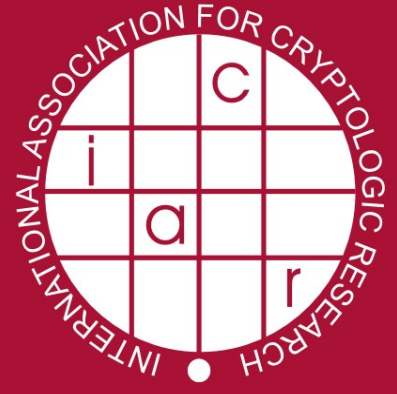
Marc Fischlin
Johannes Buchmann
Mark Manulis (Eds.)

LNCS 7293

Public Key Cryptography – PKC 2012

15th International Conference
on Practice and Theory in Public Key Cryptography
Darmstadt, Germany, May 2012, Proceedings



Springer

Lecture Notes in Computer Science 7293

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Marc Fischlin Johannes Buchmann
Mark Manulis (Eds.)

Public Key Cryptography – PKC 2012

15th International Conference
on Practice and Theory in Public Key Cryptography
Darmstadt, Germany, May 21-23, 2012
Proceedings

Springer

Volume Editors

Marc Fischlin
Darmstadt University of Technology, Department of Computer Science
Cryptography and Complexity Theory
Mornewegstr. 30, 64293 Darmstadt, Germany
E-mail: marc.fischlin@cryptoplexity.de

Johannes Buchmann
Darmstadt University of Technology, Department of Computer Science
Hochschulstraße 10, 64289 Darmstadt, Germany
E-mail: buchmann@cdc.informatik.tu-darmstadt.de

Mark Manulis
University of Surrey, Department of Computing
Guildford, Surrey, GU2 7XH, UK
E-mail: mark@manulis.eu

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-30056-1 e-ISBN 978-3-642-30057-8
DOI 10.1007/978-3-642-30057-8
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012937037

CR Subject Classification (1998): E.3, K.6.5, C.2, D.4.6, K.4.4, E.4, J.1

LNCS Sublibrary: SL 4 – Security and Cryptology

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

PKC 2012 was held at the Darmstadtium Congress Center in Darmstadt, Germany, during May 21–23, 2012. The conference was sponsored by the International Association for Cryptologic Research (IACR).

The proceedings of PKC 2012 contain 41 papers selected from 188 submissions, which corresponds to a record number of submissions in the history of PKC. Each submission was anonymized for the reviewing process and was assigned to at least three of the 30 Program Committee members. Submissions co-authored by committee members were assigned to at least five members. Committee members were allowed to submit at most one paper, or two if the second one was co-authored by a student. The committee decided to give the Best Paper Award to the paper "On Definitions of Selective Opening Security" by Florian Böhl, Dennis Hofheinz, and Daniel Kraschewski. The program also included an invited talk by David Pointcheval entitled "Password-Based Authenticated Key Exchange." On behalf of the Program Committee I would like to thank David for accepting the invitation. David was so kind to also provide a summary for the proceedings.

I would like to thank all the authors who submitted papers. I am also indebted to the Program Committee members and all external reviewers for their voluntary work, especially since the huge number of submissions meant more work for each committee member than I initially promised. The committee's work was tremendously simplified by Shai Halevi's submission software and his support. I would also like to thank the PKC Steering Committee for electing me as Chair.

Many thanks also go to the General Chairs, Johannes Buchmann and Mark Manulis, for making the event possible, and to Stanislav Bulygin and Heike Meissner for their support. Following a loose tradition of past PKC conferences you can find the General Chairs' names as co-editors of the proceedings. Unlike me, the General Chairs were in principle allowed to submit to the conference because they did not intervene in the selection process. It is clear that any objections or complaints about the program should be addressed to me.

I would have liked to say that we definitely picked the best papers among the submissions. But this would assume that there was only one best subset. And even if there was, the selection process inevitably contains some randomness, as acknowledged by chairs of other conferences before. Also, the importance of a paper has to stand the test of time, such that some uncertainty about our choice of today remains. I therefore hope the good submissions which did not make the cut for PKC eventually get accepted somewhere, and I hope that the papers we have chosen are interesting to the readers.

May 2012 Marc Fischlin

Organization

Program Chair

Marc Fischlin	Technische Universität Darmstadt, Germany

General Chairs

Johannes Buchmann	Technische Universität Darmstadt, Germany
Mark Manulis	University of Surrey, UK

Program Committee

Michel Abdalla	ENS Paris, France
Alexandra Boldyreva	Georgia Institute of Technology, USA
Colin Boyd	Queensland University of Technology, Australia
Dario Catalano	Università di Catania, Italy
Jean-Sebastien Coron	University of Luxembourg
Marc Fischlin (Chair)	Technische Universität Darmstadt, Germany
Georg Fuchsbauer	University of Bristol, UK
Rosario Gennaro	IBM Research, USA
Dov Gordon	Columbia University, USA
Matthew Green	Johns Hopkins University, USA
Jens Groth	University College London, UK
Kaoru Kurosawa	Ibaraki University, Japan
Mirosław Kutyłowski	Wrocław University of Technology, Poland
Vadim Lyubashevsky	ENS Paris, France
Adam O'Neill	Boston University, USA
Christiane Peters	Technical University of Denmark
Krzysztof Pietrzak	IST Austria
Alon Rosen	IDC Herzliya, Israel
Dominique Schröder	University of Maryland, USA
Gil Segev	Microsoft Research Silicon Valley, USA
Nicolas Sendrier	INRIA, France
Igor Shparlinski	Macquarie University, Australia
Nigel Smart	University of Bristol, UK
Ron Steinfeld	Macquarie University, Australia
Rainer Steinwandt	Florida Atlantic University, USA
Tomas Toft	Aarhus University, Denmark
Wen-Guey Tzeng	National Chiao Tung University, Taiwan
Ivan Visconti	University of Salerno, Italy
Scott Yilek	University of St. Thomas, USA
Yuliang Zheng	University of North Carolina at Charlotte, USA

PKC Steering Committee

Ronald Cramer	CWI and Universiteit Leiden, The Netherlands
Yvo Desmedt	University College London, UK
Hideki Imai	Chuo University and AIST, Japan
David Naccache	Ecole Normale Superieure, France
Tatsuaki Okamoto	NTT Labs, Japan
David Pointcheval	Ecole Normale Superieure, France
Moti Yung (Secretary)	Google Inc. and Columbia University, USA
Yuliang Zeng (Chair)	University of North Carolina at Charlotte, USA

External Reviewers

Ben Adida
Ayo Akinyele
Martin Albrecht
Daniel Augot
Manuel Barbosa
Stephanie Bayer
David Bernhard
Gaetan Bisson
Nir Bitansky
Przemyslaw Blaskiewicz
Olivier Blazy
Julia Borghoff
Elette Boyle
Zvika Brakerski
Angelo De Caro
David Cash
Nathan Chenette
Céline Chevalier
Seung Geol Choi
Ashish Choudhury
Paolo D'Arco
Morten Dahl
Yvo Desmedt
Jintai Ding
Christophe Doche
Yevgeniy Dodis
Vivien Dubois
Leo Ducas
Andrej Dujella
Stefan Dziembowski
Reza Rezaeian Farashahi
Pooya Farshim
Jean-Charles Faugere
Sebastian Faust
Serge Fehr
Dario Fiore
David Mandell Freeman
Eiichiro Fujisaki
Jun Furukawa
Philippe Gaborit
Steven Galbraith
David Galindo Chacon
Sanjam Garg
Christina Garman
Essam Ghadafi
Zbigniew Golebiewski
Domingo Gomez Perez
Adam Groce
Jaime Gutierrez
Shai Halevi
Goichiro Hanaoka
Lucjan Hanzlik
Malin Haque
Carmit Hazay
Swee-Huay Heng
Nadia Heninger
Stefan Heyse
Huseyin Hisil
Dennis Hofheinz
Susan Hohenberger
Pavel Hubacek
Abhishek Jain
Thomas P. Jakobsen
Stas Jarecki
Antoine Joux
Marc Joye
Saqib A. Kakvi
Jonathan Katz
Yutaka Kawai
Marcel Keller
Eike Kiltz
Marek Klonowski
Michal Koza
Daniel Kraschewski
Hugo Krawczyk
Stephan Krenn
Lukasz Krzywiecki
Przemyslaw Kubiak
Virendra Kumar
Abishek Kumarasubramanian
Ranjit Kumaresan
Fabien Laguillaumie
Tanja Lange
Anna Lauks-Dutka
Gregor Leander
Allison Lewko
Benoît Libert
Huijia Lin
Yehuda Lindell
Georg Lippold
Jacob Loftus
Adriana Lopez-Alt
Krzysztof Majcher
Alex Malozemoff
Mark Manulis

Takahiro Matsuda
Alexander May
Carlos Aguilar Melchor
Alexander Meurer
Daniele Micciancio
Ian Miers
Petros Mol
Dustin Moody
Francois Morain
Tal Moran
Michael Naehrig
Gregory Neven
Antonio Nicolosi
Wakaha Ogata
Miyako Ohkubo
Daniel Page
Omer Paneth
Valerio Pastro
Kenny Paterson
Chris Peikert
Ludovic Perret
Duong Hieu Phan
Le Trieu Phong
Josef Pieprzyk
David Pointcheval
Joop Van de Pol
Roberto De Prisco
Xavier Pujol
Elizabeth Quaglia
Tal Rabin
Mario Di Raimondo
Vanishree Rao
Mariana Raykova
Michal Ren
Tom Ristenpart
Mike Rosulek
Yannis Rouselakis
Yusuke Sakai
Alessandra Scafuro
Michael Schneider
Sven Schäge
Michael Scott
Reza Sepahi
Siamak F. Shahandashti
Jun Shao
Abhi Shelat
Martijn Stam
Damien Stehlé
Mario Strefler
Tomasz Struminski
Piotr Syga
Katsuyuki Takashima
Aris Tentes
Stefano Tessaro
Aishwarya Thiruvengadam
Enrico Thomae
Emmanuel Thomé
Mehdi Tibouchi
Jean-Pierre Tillich
Dominique Unruh
Yevgeniy Vahlis
Vinod Vaikuntanathan
Damien Vergnaud
Akshay Wadia
Gaven Watson
William Whyte
Daniel Wichs
Pawel Wlaz
Wojciech Wodo
Christopher Wolf
Shota Yamada
Bo-Yin Yang
Arkady Yerukhimovich
Filip Zagorski
Sarah Zakarias
Angela Zottarel

Table of Contents

Homomorphic Encryption and LWE

Signature Schemes

Code-Based and Multivariate Crypto

Public-Key Encryption: Special Properties

Identity-Based Encryption

Public-Key Encryption: Constructions

Invited Talk

Secure Two-Party and Multi-party Computations

Key Exchange and Secure Sessions

Public-Key Encryption: Relationships

DL, DDH, and More Number Theory

Beyond Ordinary Signature Schemes

Better Bootstrapping in Fully Homomorphic Encryption

Craig Gentry[1], Shai Halevi[1], and Nigel P. Smart[2]

[1] IBM T.J. Watson Research Center
[2] Dept. Computer Science, University of Bristol

Abstract. Gentry's bootstrapping technique is currently the only known method of obtaining a "pure" fully homomorphic encryption (FHE) schemes, and it may offers performance advantages even in cases that do not require pure FHE (e.g., when using the noise-control technique of Brakerski-Gentry-Vaikuntanathan).

The main bottleneck in bootstrapping is the need to evaluate homomorphically the reduction of one integer modulo another. This is typically done by emulating a binary modular reduction circuit, using bit operations on binary representation of integers. We present a simpler approach that bypasses the homomorphic modular-reduction bottleneck to some extent, by working with a modulus very close to a power of two. Our method is easier to describe and implement than the generic binary circuit approach, and we expect it to be faster in practice (although we did not implement it yet). In some cases it also allows us to store the encryption of the secret key as a single ciphertext, thus reducing the size of the public key.

We also show how to combine our new method with the SIMD homomorphic computation techniques of Smart-Vercauteren and Gentry-Halevi-Smart, to get a bootstrapping method that works in time quasi-linear in the security parameter. This last part requires extending the techniques from prior work to handle arithmetic not only over fields, but also over some rings. (Specifically, our method uses arithmetic modulo a power of two, rather than over characteristic-two fields.)

1 Introduction

Fully Homomorphic Encryption (FHE) [12,7] is a powerful technique to enable a party to compute an arbitrary function on a set of encrypted inputs; and hence obtain the encryption of the function's output. Starting from Gentry's breakthrough result [6,7], all known FHE schemes are constructed from *Somewhat Homomorphic Encryption* (SWHE) schemes, that can only evaluate functions of bounded complexity. The ciphertexts in these SWHE schemes include some "noise" to ensure security, and this noise grows when applying homomorphic operations until it becomes so large that it overwhelms the decryption algorithm and causes decryption errors. To overcome the growth of noise, Gentry used a *bootstrapping* transformation, where the decryption procedure is run homomorphically on a given ciphertext, using an encryption of the secret key that can be found in the public key,[1] resulting in a new ciphertext that encrypts the same message but has potentially smaller noise.

Over the last two years there has been a considerable amount of work on developing new constructions and optimizations [5,13,9,3,14,2,8,1,11], but all of these constructions

[1] This transformation relies on the underlying SWHE being circularly secure.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 1–16, 2012.

still have noise that keeps growing and must be reduced before it overwhelms the decryption procedure. The techniques of Brakerski et al. [1] yield SWHE schemes where the noise grows slower, only linearly with the depth of the circuit being evaluated, but for any fixed public key one can still only evaluate circuits of fixed depth. The only known way to get "pure" FHE that can evaluate arbitrary functions with a fixed public key is by using bootstrapping. Also, bootstrapping can be used in conjunction with the techniques from [1] to get better parameters (and hence faster homomorphic evaluation), as described in [1,11].

In nearly all SWHE schemes in the literature that support bootstrapping, decryption is computed by evaluating some ciphertext-dependent linear operation on the secret key, then reducing the result modulo a public odd modulus q into the range $(-q/2, q/2]$, and then taking the least significant bit of the result. Namely, denoting reduction modulo q by $[\cdot]_q$, we decrypt a ciphertext c by computing $a = [[L_c(s)]_q]_2$ where L_c is a linear function and s is the secret key. Given an encryption of the secret key s, computing an encryption of $L_c(s)$ is straightforward, and the bulk of the work in homomorphic decryption is devoted to reducing the result modulo q. This is usually done by computing encryptions of the bits in the binary representation of $L_c(s)$ and then emulating the binary circuit that reduces modulo q.

The starting point of this work is the observation that when q is very close to a power of two, the decryption formula takes a particularly simple form. Specifically, we can compute the linear function $L_c(s)$ modulo a power of two, and then XOR the top and bottom bits of the result. We then explain how to implement this simple decryption formula homomorphically, and also how the techniques of Gentry et al. from [11] can be used to compute this homomorphic decryption with only polylogarithmic overhead.

We note that applying the techniques from [11] to bootstrapping is not quite straightforward, because the input and output are not presented in the correct form for these techniques. (This holds both for the standard approach of emulating binary mod-q circuit and for our new approach.) Also, for our case we need to extend the results from [11] slightly, since we are computing a function over a ring (modulo a power of two) and not over a field.

We point out that in all work prior to [11], bootstrapping required adding to the public key many ciphertexts, encrypting the individual bits (or coefficients) of the secret key. This resulted in very large public keys, of size at least $\lambda^2 \cdot \mathrm{polylog}(\lambda)$ (where λ is the security parameter). Using the techniques from [14,1,11], it is possible to encrypt the secret key in a "packed" form, hence reducing the number of ciphertexts to $O(\log \lambda)$ (so we can get public keys of size quasi-linear in λ). Using our technique from this work, it is even possible to store an encryption of the secret key as a single ciphertext, as described in Section 4. We next outline our main bootstrapping technique in a few more details.

Our method applies mainly to "leveled" schemes that use the noise control mechanism of Brakerski-Gentry-Vaikuntanathan [1].[2] Below and throughout this paper we concentrate on the BGV ring-LWE-based scheme, since it offers the most efficient

[2] Our method can be used also with other schemes, as long as the scheme allows us to choose a modulus very close to a power of two. For example they can be used with the schemes from [3,2].

homomorphic operations and the most room for optimizations.[3] The scheme is defined over a ring $R = \mathbb{Z}[X]/F(X)$ for a monic, irreducible polynomial $F(X)$ (over the integers $\mathbb{Z}$). For an arbitrary integer modulus n (not necessarily prime) we denote the ring $R_n \stackrel{\text{def}}{=} R/nR = (\mathbb{Z}/n\mathbb{Z})[X]/F(X)$. The scheme is parametrized by the number of levels that it can handle, which we denote by L, and by a set of decreasing odd moduli $q_0 \gg q_1 \gg \cdots \gg q_L$, one for each level.

The plaintext space is given by the ring R_2, while the ciphertext space for the i'th level consists of vectors in $(R_{q_i})^2$. Secret keys are polynomials $\mathfrak{s} \in R$ with "small" coefficients, and we view $\mathfrak{s}$ as the second element of the 2-vector $\mathbf{s} = (1, \mathfrak{s})$. A level-$i$ ciphertext $\mathbf{c} = (c_0, c_1)$ encrypts a plaintext polynomial $m \in R_2$ with respect to $\mathbf{s} = (1, \mathfrak{s})$ if we have the equality over R, $[\langle \mathbf{c}, \mathbf{s} \rangle]_{q_i} = [c_0 + \mathfrak{s} \cdot c_1]_{q_i} \equiv m \pmod 2$, and moreover the polynomial $[c_0 + \mathfrak{s} \cdot c_1]_{q_i}$ is "small", i.e. all its coefficients are considerably smaller than q_i. Roughly, that polynomial is considered the "noise" in the ciphertext, and its coefficients grow as homomorphic operations are performed.[4] The crux of the noise-control technique from [1] is that a level-i ciphertext can be publicly converted into a level-$(i+1)$ ciphertext (with respect to the same secret key), and that this transformation reduces the noise in the ciphertext roughly by a factor of q_{i+1}/q_i.

Secret keys too are associated with levels, and the public key includes some additional information that (roughly speaking) makes it possible to convert a ciphertext with respect to level-i key $\mathbf{s}_i$ into a ciphertext with respect to level-$(i+1)$ key $\mathbf{s}_{i+1}$. In what follows we will only be interested in the secret keys at level L and level zero; which we will denote by $\mathbf{s}$ and $\tilde{\mathbf{s}}$ respectively to ease notation.

For bootstrapping, we have as input a level-L ciphertext (i.e. a vector $\mathbf{c} \in R/q_L R$ modulo the smallest modulus q_L). This means that the noise-control technique can no longer be applied to reduce the noise, hence (essentially) no more homomorphic operations can be performed on this ciphertext. To enable further computation, we must therefore "recrypt" the ciphertext $\mathbf{c}$, to obtain a new ciphertext that encrypts the same element of R with respect to some lower level $i < L$.

Our first observation is that the decryption at level L can be made more efficient when q_L is close to a power of two, specifically $q_L = 2^r + 1$ for an integer r, and moreover the coefficients of $Z = \langle \mathbf{c}, \mathbf{s} \rangle \bmod F(X)$ are much smaller than q_L^2 in magnitude. In particular if z is one of the coefficients of the polynomial Z then $[[z]_{q_L}]_2$ can be computed as $z\langle r \rangle \oplus z\langle 0 \rangle$, where $z\langle i \rangle$ is the i'th bit of z.

To evaluate the decryption formula homomorphically, we temporarily extend the plaintext space to polynomials modulo 2^{r+1} (rather than modulo 2). The level-L secret key is $\mathbf{s} = (1, \mathfrak{s})$, where all the coefficients of $\mathfrak{s}$ are small (in the interval $(-2^r, +2^r)$). We can therefore consider $\mathfrak{s}$ as a plaintext polynomial in $R/2^{r+1}R$, encrypt it inside a level-0 ciphertext, and keep that ciphertext in the public key. Thus, given the level-L ciphertext $\mathbf{c}$, we can evaluate the inner product $[\langle \mathbf{c}, \mathbf{s} \rangle \bmod F(X)]$ homomorphically, obtaining a level-0 ciphertext that encrypts the polynomial Z.

For simplicity, assume for now that what we get is an encryption of all the coefficients of Z separately. Given an encryption of a coefficient z of Z (which is an element

[3] Our description of the BGV cryptosystem below assumes modulo-2 plaintext arithmetic, generalizing to modulo-p arithmetic for other primes $p > 2$ is straightforward.

[4] We ignore here the encryption procedure, since it does not play any role in the current work.

in $\mathbb{Z}/2^{r+1}\mathbb{Z}$) we show in Section 3.1 how to extract (encryptions of) the zero'th and r'th bit using a data-oblivious algorithm. Hence we can finally recover a new ciphertext, encrypting the same binary polynomial at a lower level $i < L$.

To achieve efficient bootstrapping, we exploit the ability to perform operations on elements modulo 2^{r+1} in a SIMD fashion (Single Instruction Multiple Data); much like in prior work [14,1,11]. Some care must be taken when applying these techniques in our case, since the inputs and outputs of the bootstrapping procedure are not in the correct format: Specifically, these techniques require that inputs and outputs be represented using polynomial Chinese Remainders (CRT representation), whereas decryption (and therefore recryption) inherently deals with polynomials in coefficient representation. We therefore must use explicit conversion to CRT representation, and ensure that these conversions are efficient enough. See details in Section 4.

Also, the techniques from prior work must be extended somewhat to be usable in our case: Prior work demonstrated that SIMD operations can be performed homomorphically when the underlying arithmetic is over a field, but in our case we have operations over the ring $\mathbb{Z}/2^{r+1}\mathbb{Z}$, which is not a field. The algebra needed to extend the SIMD techniques to this case is essentially an application of the theory of local fields [4]. We prove many of the basic results that we need in the full version [10], and refer the reader to [4] for a general introduction and more details.

Notations. Throughout the paper we denote by $[z]_q$ the reduction of z mod q into the interval $(-\frac{q}{2}, \frac{q}{2}]$. We also denote the i'th bit in the binary representation of the integer z by $z\langle i\rangle$. Similarly, when a is an integer polynomial of degree d with coefficients $(a_0, a_1, \ldots, a_d)$, we denote by $a\langle i\rangle$ the 0-1 degree-d polynomial whose coefficients are all the i'th bits $(a_0\langle i\rangle, a_1\langle i\rangle, \ldots, a_d\langle i\rangle)$. If $\mathbf{c}, \mathbf{s}$ are two same-dimension vectors, then $\langle \mathbf{c}, \mathbf{s}\rangle$ denotes their inner product.

Organization. We begin by presenting the simplified decryption formula in Section 2 and explain how to evaluate it homomorphically in Section 3. Then in Section 4 we recall some algebra and explain how to use techniques similar to [11] to run bootstrapping in time quasi-linear in the security parameter. Some of the proofs are omitted here, these are found in the full version of this work [10].

2 A Simpler Decryption Formula

When the small modulus q_L has a special form – i.e. when it equals $u \cdot 2^r + v$ for some integer r and for some small positive odd integers u, v – then the mod-q_L decryption formula can be made to have a particularly simple form. Below we focus on the case of $q_L = 2^r + 1$, which suffices for our purposes.

So, assume that $q_L = 2^r + 1$ for some integer r and that we decrypt by setting $a \leftarrow [[\langle \mathbf{c}, \mathbf{s}\rangle \bmod F(X)]_{q_L}]_2$. Consider now the coefficients of the integer polynomial $Z = \langle \mathbf{c}, \mathbf{s}\rangle \bmod F(X)$, without the reduction mod q_L. Since $\mathbf{s}$ has small coefficients (and we assume that reduction mod-$F(X)$ does not increase the coefficients by much) then all the coefficients of Z are much smaller than q_L^2. Consider one of these integer coefficients, denoted by z, so we know that $|z| \ll q_L^2 \approx 2^{2r}$. We consider the binary

representation of z as a $2r$-bit integer, and assume for now that $z \geq 0$ and also $[z]_{q_L} \geq 0$. We claim that in this case, the bit $[[z]_{q_L}]_2$ can be computed simply as the sum of the lowest bit and the r'th bit of z, i.e., $[[z]_{q_L}]_2 = z\langle r\rangle \oplus z\langle 0\rangle$. (Recall that $z\langle i\rangle$ is the i'th bit of z.)

Lemma 1. *Let $q = 2^r + 1$ for a positive integer r, and let z be a non-negative integer smaller than $\frac{q^2}{2} - q$, such that $[z]_q$ is also non-negative, $[z]_q \in [0, \frac{q}{2}]$. Then $[[z]_q]_2 = z\langle r\rangle \oplus z\langle 0\rangle$.*

Proof. Let $z_0 = [z]_q \in [0, \frac{q}{2}]$, and consider the sequence of integers $z_i = z_0 + iq$ for $i = 0, 1, 2, \cdots$. Since we assume that $z \geq 0$ then z can be found in this sequence, say the k'th element $z = z_k = z_0 + kq$. Also since $z < \frac{q^2}{2} - q$ then $k = \lfloor z/q \rfloor < \frac{q}{2} - 1$. The bit that we want to compute is $[[z]_q]_2 = z_0\langle 0\rangle$. We claim that $z_0\langle 0\rangle = z_k\langle 0\rangle + z_k\langle r\rangle \pmod 2$. This is because $z_k = z_0 + kq = z_0 + k(2^r + 1) = (z_0 + k) + k2^r$, which in particular means that $z_k\langle 0\rangle = z_0\langle 0\rangle + k\langle 0\rangle \pmod 2$. But since $0 \leq z_0 \leq q/2$ and $0 \leq k < q/2 - 1$ then $0 \leq z_0 + k < q - 1 = 2^r$, so there is no carry bit from the addition $z_0 + k$ to the r'th bit position. It follows that the r'th bit of z_k is equal to the 0'th bit of k (i.e., $z_k\langle r\rangle = k\langle 0\rangle$), and therefore $z_k\langle 0\rangle = z_0\langle 0\rangle + k\langle 0\rangle = z_0\langle 0\rangle + z_k\langle r\rangle \pmod 2$, which implies that $z_0\langle 0\rangle = z_k\langle 0\rangle + z_k\langle r\rangle \pmod 2$, as needed. □

We note that the proof can easily be extended for the case $q = u2^r + v$, if the bound on z is strengthened by a factor of v. To remove the assumption that both z and $[z]_q$ are non-negative, we use the following easy corollary:

Corollary 1. *Let $r \geq 3$ and $q = 2^r + 1$ and let z be an integer with absolute value smaller than $\frac{q^2}{4} - q$, such that $[z]_q \in (-\frac{q}{4}, \frac{q}{4})$. Then $[[z]_q]_2 = z\langle r\rangle \oplus z\langle r-1\rangle \oplus z\langle 0\rangle$.*

Proof. Denoting $z' = z + (q^2-1)/4 = z + (q+1)(q-1)/4 = \left(z + \frac{q-1}{4}\right) + q \cdot \frac{q-1}{4}$, we have $z' \equiv z + \frac{q-1}{4} \pmod q$ (since $\frac{q-1}{4} = 2^{r-2}$ is an integer). Moreover since $[z]_q \in (-\frac{q}{4}, \frac{q}{4}]$ then $[z]_q + \frac{q-1}{4} \in [0, q/2]$, hence $[z']_q = [z]_q + \frac{q-1}{4}$ (over the integers), and as $\frac{q-1}{4}$ is an even integer then $[z]_q = [z']_q \pmod 2$, or in other words $[[z]_q]_2 = [[z']_q]_2$. Since $z > -\frac{q^2}{4}$ and z is an integer then $z \geq -\frac{q^2-1}{4}$ and therefore $z' = z + \frac{q^2-1}{4} \geq 0$. Thus z' satisfies all the conditions set in Lemma 1, so applying that lemma we have $[[z]_q]_2 = [[z']_q]_2 = z'\langle r\rangle \oplus z'\langle 0\rangle$.

We next observe that $z' = z + (q+1)(q-1)/4 = z + (2^r+2)2^{r-2} = z + 2^{r-1} + 2^{2r-2}$. Since $2r - 2 > r$, this means that the bits 0 through r in the binary representation of z' are determined by $z + 2^{r-1}$ alone, so we have:

$$\begin{aligned} z'\langle i\rangle &= z\langle i\rangle \text{ for } i = 0, 1, \ldots, r-2 \\ z'\langle r-1\rangle &= 1 - z\langle r-1\rangle \\ z'\langle r\rangle &= \left\{ \begin{matrix} z\langle r\rangle & \text{if } z\langle r-1\rangle = 0 \\ 1 - z\langle r\rangle & \text{if } z\langle r-1\rangle = 1 \end{matrix} \right\} \;=\; z\langle r\rangle \oplus z\langle r-1\rangle \end{aligned}$$

Putting it all together, we get $[[z]_q]_2 = [[z']_q]_2 = z'\langle r\rangle \oplus z'\langle 0\rangle = z\langle r\rangle \oplus z\langle r-1\rangle \oplus z\langle 0\rangle$. □

Using Corollary 1 we can get our simplified decryption formula. First, we set our parameters such that $q_L = 2^r + 1$ and all the coefficients of the integer polynomial $Z = \langle \mathbf{c}, \mathbf{s} \rangle \bmod F(X)$ are smaller than $\frac{q_L^2}{4} - 1$ in absolute value, and moreover they are all less than $\frac{q_L - 1}{4}$ away from a multiple of q_L. Given a two-element ciphertext $\mathbf{c} = (c_0, c_1) \in ((\mathbb{Z}/q_L\mathbb{Z})[X]/F(X))^2$, then compute $Z \leftarrow \langle \mathbf{c}, \mathbf{s} \rangle \bmod F(X)$ over the integers (without reduction mod q_L), and finally recover the plaintext as $Z\langle r \rangle + Z\langle r-1 \rangle + Z\langle 0 \rangle$. Ultimately, we obtain the plaintext polynomial $a \in \mathbb{F}_2[X]/F(X)$, where each coefficient in a is obtained as the XOR of bits 0, $r-1$, and r of the corresponding coefficient in Z.

Working Modulo 2^{r+1}. Since we are only interested in the contents of bit positions 0, $r-1$, and r in the polynomial Z, we can compute Z modulo 2^{r+1} rather than over the integers. Our simplified decryption of a ciphertext vector $\mathbf{c} = (c_0, c_1)$ proceeds as follows:

1. Compute $Z \leftarrow [\langle \mathbf{c}, \mathbf{s} \rangle \bmod F(X)]_{2^{r+1}}$;
2. Recover the 0-1 plaintext polynomial $a = [Z\langle r \rangle + Z\langle r-1 \rangle + Z\langle 0 \rangle]_2$.

3 Basic Homomorphic Decryption

To get a homomorphic implementation of the simplified decryption formula from above, we use an instance of our homomorphic encryption scheme with underlying plaintext space $\mathbb{Z}_{2^{r+1}}$. Namely, denoting by $\tilde{\mathbf{s}}$ the level-0 secret-key and by q_0 the largest modulus, a ciphertext encrypting $a \in (\mathbb{Z}/2^{r+1}\mathbb{Z})[X]/F(X)$ with respect to $\tilde{\mathbf{s}}$ and q_0 is a 2-vector $\tilde{\mathbf{c}}$ over $(\mathbb{Z}/q_0\mathbb{Z})[X]/F(X)$ such that $|[\langle \tilde{\mathbf{c}}, \tilde{\mathbf{s}} \rangle \bmod F(X)]_{q_0}| \ll q_0$ and $[\langle \tilde{\mathbf{c}}, \tilde{\mathbf{s}} \rangle \bmod F(X)]_{q_0} \equiv a \pmod{2^{r+1}}$.

Recall that the ciphertext before bootstrapping is with respect to secret key $\mathbf{s}$ and modulus $q_L = 2^r + 1$. In this section we only handle the simple case where the public key includes an encryption of each coefficient of the secret-key $\mathbf{s}$ separately. Namely, denoting $\mathbf{s} = (1, \mathfrak{s})$ and $\mathfrak{s}(X) = \sum_{j=0}^{d-1} \mathfrak{s}_j X^j$, we encode for each j the coefficient $\mathfrak{s}_j$ as the constant polynomial $\mathfrak{s}_j \in (\mathbb{Z}/2^{r+1}\mathbb{Z})[X]/F(X)$. (I.e., the degree-$d$ polynomial whose free term is $\mathfrak{s}_j \in [-2^r + 1, 2^r]$ and all the other coefficients are zero.) Then for each j we include in the public key a ciphertext $\tilde{\mathbf{c}}_j$ that encrypts this constant polynomial $\mathfrak{s}_j$ with respect to $\tilde{\mathbf{s}}$ and q_0. Below we abuse notations somewhat, using the same notation to refer both to a constant polynomial $z \in (\mathbb{Z}/2^r\mathbb{Z})[X]/F(X)$ and the free term of that polynomial $z \in (\mathbb{Z}/2^r\mathbb{Z})$.

Computing Z Homomorphically. Given the q_L-ciphertext $\mathbf{c} = (c_0, c_1)$ (that encrypts a plaintext polynomial $a \in \mathbb{F}_2[X]/F(X)$), we use the encryption of $\mathbf{s}$ from the public key to compute the simple decryption formula from above. Computing an encryption of $Z = [\langle \mathbf{c}, \mathbf{s} \rangle \bmod F(X)]_{2^{r+1}}$ is easy, since the coefficients of Z are just affine functions (over $(\mathbb{Z}/2^{r+1}\mathbb{Z})$) of the coefficients of $\mathfrak{s}$, which we can compute from the encryption of the $\mathfrak{s}_j$'s in the public key.

3.1 Extracting the Top and Bottom Bits

Now that we have encryptions of the coefficients of Z, we need to extract the relevant three bits in each of these coefficients and add them (modulo 2) to get encryptions of the plaintext coefficients. In more details, given a ciphertext $\tilde{\mathbf{c}}$ satisfying $[\langle \tilde{\mathbf{c}}, \tilde{\mathbf{s}} \rangle \bmod F(X)]_{q_0} \equiv z \pmod{2^{r+1}}$ where z is some constant polynomial, we would like to compute another ciphertext $\tilde{\mathbf{c}}$ satisfying $[\langle \tilde{\mathbf{c}}, \tilde{\mathbf{s}} \rangle \bmod F(X)]_{q_0} \equiv z\langle 0 \rangle + z\langle r-1 \rangle + z\langle r \rangle \pmod 2$ (with $[\langle \tilde{\mathbf{c}}, \tilde{\mathbf{s}} \rangle \bmod F(X)]_{q_0}$ still much smaller then q_0 in magnitude). To this end, we describe a procedure to compute for all $i = 0, 1, \ldots, r$ a ciphertext $\tilde{\mathbf{c}}_i$ satisfying $[\langle \tilde{\mathbf{c}}_i, \tilde{\mathbf{s}} \rangle \bmod F(X)]_{q_0} \equiv z\langle i \rangle \pmod 2$. Clearly, we can immediately set $\tilde{\mathbf{c}}_0 = \tilde{\mathbf{c}}$, we now describe how to compute the other $\tilde{\mathbf{c}}_i$'s.

The basic observation underlying this procedure is that modulo a power of 2, the second bit of $z - z^2$ is the same as that of z, but the LSB is zero-ed out. Thus setting $z' = (z - z^2)/2$ (which is an integer), we get that the LSB of z' is the second bit of z. More generally, we have the following lemma:

Lemma 2. *Let z be an integer with binary representation $z = \sum_{i=0}^{r} 2^i z\langle i \rangle$. Define $w_0 \stackrel{\text{def}}{=} z$, and for $i \geq 1$ define*

$$w_i \stackrel{\text{def}}{=} \frac{z - \sum_{j=0}^{i-1} 2^j w_j^{2^{i-j}} \bmod 2^{r+1}}{2^i} \quad \textit{(division by } 2^i \textit{ over the rationals).} \tag{1}$$

Then the w_i's are integers and we have $w_i\langle 0 \rangle = z\langle i \rangle$ for all i.

Proof. The lemma clearly holds for $i = 0$. Now fix some $i \geq 1$, assume that the lemma holds for all $j < i$, and we prove that it holds also for i. It is easy to show by induction that for any integer u and all $j \leq r$ we have

$$u^{2^j} \bmod 2^{r+1} = u\langle 0 \rangle + 2^{j+1} t \text{ for some integer } t.$$

Namely, the LSB of $u^{2^j} \bmod 2^{r+1}$ is the same as the LSB of u, and the next j bits are all zero. This means that the bit representation of $v_j \stackrel{\text{def}}{=} 2^j w_j^{2^{i-j}} \bmod 2^{r+1}$ has bits $0, 1, \ldots, j-1$ all zero (due to the multiplication by 2^j), then $v_j\langle j \rangle = w_j\langle 0 \rangle = z\langle j \rangle$ (by the induction hypothesis), and the next $i - j$ bits are again zero (by the observation above). In other words, the lowest $i + 1$ bits of v_j are all zero, except the j'th bit which is equal to the j'th bit of z.

This means that the lowest i bits of the sum $\sum_{j=0}^{i-1} v_j$ are the same as the lowest i bits of z, and the $i + 1$'st bit of the sum is zero. Hence the lowest i bits of $z - \sum_{j=0}^{i-1} v_j$ are all zero, and the $i + 1$'st bit is $z\langle i \rangle$. Hence $z - \sum_{j=0}^{i-1} v_j$ is divisible by 2^i (over the integers), and the lowest bit of the result is $z\langle i \rangle$, as needed. □

Our procedure for computing the ciphertexts $\tilde{\mathbf{c}}_i$ mirrors Lemma 2. Specifically, we are given the ciphertext $\tilde{\mathbf{c}} = \tilde{\mathbf{c}}_0$ that encrypts $z = w_0 \bmod 2^{r+1}$, and we iteratively compute ciphertexts $\tilde{\mathbf{c}}_1, \tilde{\mathbf{c}}_2, \ldots$ such that $\tilde{\mathbf{c}}_i$ encrypts $w_i \bmod 2^{r-i+1}$. Eventually we get $\tilde{\mathbf{c}}_r$ that encrypts $w_r \bmod 2$, which is what we need (since the LSB of w_r is the r'th bit of z).

Note that most of the operations in Lemma 2 are carried out in $(\mathbb{Z}/2^{r+1}\mathbb{Z})$, and therefore can be evaluated homomorphically in our $(\mathbb{Z}/2^{r+1}\mathbb{Z})$-homomorphic cryptosystem. The only exception is the division by 2^i in Equation (1), and we now show how this division can also be evaluated homomorphically. To implement division we begin with an arbitrary ciphertext vector $\tilde{\mathbf{c}}$ that encrypts a plaintext element $a \in (\mathbb{Z}/2^j\mathbb{Z})[X]/F(X)$ (for some j) with respect to the level-0 key $\tilde{\mathbf{s}}$ and modulus q_0. Namely, we have the equality over $\mathbb{Z}[X]$:

$$(\langle \tilde{\mathbf{c}}, \tilde{\mathbf{s}} \rangle \bmod F(X)) = a + 2^j \cdot S + q_0 \cdot T$$

for some polynomials $S, T \in \mathbb{Z}[X]/F(X)$, where the norm of $a + 2^j S$ is much smaller than q_0. Assuming that a is divisible by 2 over the integers (i.e., all its coefficients are even) consider what happens when we multiply $\tilde{\mathbf{c}}$ by the integer $(q_0+1)/2$ (which is the inverse of 2 modulo q_0). Then we have

$$\begin{aligned}
(\langle \tfrac{q_0+1}{2} \cdot \tilde{\mathbf{c}}, \tilde{\mathbf{s}} \rangle \bmod F(X)) &= \tfrac{q_0+1}{2} \cdot (\langle \tilde{\mathbf{c}}, \tilde{\mathbf{s}} \rangle \bmod F(X)) \\
&= \frac{(q_0+1)\cdot a}{2} + \frac{(q_0+1)\cdot 2^j \cdot S}{2} + \frac{q_0 \cdot (q_0+1) \cdot T}{2} \\
&= (q_0+1)\cdot(a/2) \;+\; (q_0+1)\cdot 2^{j-1} S \;+\; q_0 \cdot \tfrac{q_0+1}{2} \cdot T \\
&= a/2 \;+\; 2^{j-1} \cdot S \;+\; q_0 \cdot \left(a/2 + 2^{j-1} S + \tfrac{q_0+1}{2} T\right)
\end{aligned}$$

Clearly the coefficients of $a/2 + 2^{j-1}S$ are half the size of those of $a + 2^j S$, hence they are much smaller than q_0. It follows that $\tilde{\mathbf{c}}' = [\tilde{\mathbf{c}} \cdot (q_0+1)/2]_{q_0}$ is a valid ciphertext that encrypts the plaintext $a/2 \in (\mathbb{Z}/2^{j-1}\mathbb{Z})[X]/F(X)$ with respect to secret key $\tilde{\mathbf{s}}$ and modulus q_0.

The same argument shows that if a is divisible by 2^i over the integers (for some $i < j$) then $[\tilde{\mathbf{c}} \cdot ((q_0+1)/2)^i]_{q_0}$ is a valid ciphertext encrypting $a/2^i \in (\mathbb{Z}/2^{j-i}\mathbb{Z})[X]/F(X)$. Combining this division-by-two procedure with homomorphic exponentiation mod 2^{r+1}, the resulting homomorphic bit-extraction procedure is described in Figure 1.

3.2 Packing the Coefficients

Now that we have encryption of all the coefficients of a, we just need to "pack" all these coefficients back in one polynomial. Namely, we have encryption of the constant polynomials $a_0, a_1, \ldots$, and we want to get an encryption of the polynomial $a(X) = \sum_i a_i X^i$. Since a is just a linear combination of the a_i's (with the coefficient of each a_i being the "scalar" $X^i \in (Z/2Z)[X]/\Phi_m$), we can just use the additive homomorphism of the cryptosystem to compute an encryption of a from the encryptions of the a_i's.

3.3 Lower-Degree Bit Extraction

As described in Figure 1, extracting the r'th bit requires computing polynomials of degree upto 2^r, here we describe a simple trick to lower this degree. Recall our simplified decryption process: we set $Z \leftarrow [\langle \mathbf{c}, \mathbf{s} \rangle \bmod \Phi_m(X)]_{2^{r+1}}$, and then recover $a = [Z\langle r \rangle + Z\langle r-1 \rangle + Z\langle 0 \rangle]_2$.

Bit-Extraction$(\tilde{\mathbf{c}}, r, q_0)$:

Input: A ciphertext $\tilde{\mathbf{c}}$ encrypting a constant $b \in (\mathbb{Z}/2^{r+1}\mathbb{Z})$ w.r.t. secret key $\tilde{\mathbf{s}}$ and modulus q_0.
Output: A ciphertext $\tilde{\mathbf{c}}'$ encrypting $b\langle 0\rangle \oplus b\langle r-1\rangle \oplus b\langle r\rangle \in \mathbb{F}_2$ w.r.t. secret key $\tilde{\mathbf{s}}$ and modulus q_0.

1. Set $\tilde{\mathbf{c}}_0 \leftarrow \tilde{\mathbf{c}}$ // $\tilde{\mathbf{c}}$ encrypt z w.r.t. $\tilde{\mathbf{s}}$
2. For $i = 1$ to r
3. Set $\mathbf{acc} \leftarrow \tilde{\mathbf{c}}$ // $\mathbf{acc}$ is an accumulator
4. For $j = 0$ to $i-1$ // Compute $z - \sum_j 2^j w_j^{i-1}$
5. Set $\mathbf{tmp} \leftarrow \mathsf{HomExp}(\tilde{\mathbf{c}}_j, 2^{i-j})$ // Homomorphic exponentiation to the power 2^{i-j}
6. Set $\mathbf{acc} \leftarrow \mathbf{acc} - 2^j \cdot \mathbf{tmp} \bmod q_0$
7. Set $\tilde{\mathbf{c}}_i \leftarrow \mathbf{acc} \cdot ((q_0+1)/2)^i \bmod q_0$ // $\tilde{\mathbf{c}}_i$ encrypts $z\langle i\rangle$
8. Output $\tilde{\mathbf{c}}_0 + \tilde{\mathbf{c}}_{r-1} + \tilde{\mathbf{c}}_r \bmod q_0$

$\mathsf{HomExp}(\tilde{\mathbf{c}}, n)$ *uses native homomorphic multiplication to multiply* $\tilde{\mathbf{c}}$ *by itself* n *times. To aid exposition, this code assumes that the modulus and secret key remain fixed, else modulus-switching and key-switching should be added (and the level increased correspondingly to some* $i > 0$*).*

Fig. 1. A Homomorphic Bit-Extraction Procedure

Consider what happens if we add q_L to all the odd coefficients in $\mathbf{c}$, call the resulting vector $\mathbf{c}'$: On one hand, now all the coefficients of $\mathbf{c}'$ are even. On the other hand, the coefficients of $Z' = \langle \mathbf{c}', \mathbf{s}\rangle \bmod \Phi_m(X)$ are still small enough to use Lemma 1 (since they are at most $c_m \cdot q \cdot \|\mathbf{s}\|_1$ larger than those of Z itself, where c_m is the ring constant of mod-$\Phi_m(X)$ arithmetic and $\|s\|_1$ is the l_1-norm of $\mathbf{s}$). Since $\mathbf{c}' = \mathbf{c} \pmod{q_L}$ then we have

$$[[\langle \mathbf{c}, \mathbf{s}\rangle \bmod \Phi_m(X)]_{q_L}]_2 = [[\langle \mathbf{c}', \mathbf{s}\rangle \bmod \Phi_m(X)]_{q_L}]_2 = Z'\langle r\rangle + Z'\langle r\rangle - 1 + Z'\langle 0\rangle$$

However, since $\mathbf{c}'$ is even then so is Z'. This means that $Z'\langle 0\rangle = 0$, and if we divide Z' by two (over the integers), $Z'' = Z'/2$, then we have $[[\langle \mathbf{c}, \mathbf{s}\rangle \bmod \Phi_m(X)]_{q_L}]_2 = Z''\langle r-1\rangle \oplus Z''\langle r-2\rangle$. We thus have a variation of the simple decryption formula that only needs to extract the $r-1$'st and $r-2$'nd bits, so it can be realized using polynomials of degree upto 2^{r-1}. Note that we can implement this variant of the decryption formula homomorphically, because Z' is even so an q_0-encryption of Z' can be easily converted into an encryption of $Z'/2$ (by multiplying by $\frac{q_0+1}{2}$ modulo q_0 as described in Section 3.1).

This technique can be pushed a little further, adding to $\mathbf{c}$ multiples of q so that it is divisible by 4, 8, 16, etc., and reducing the required degree correspondingly to 2^{r-2}, 2^{r-3}, 2^{r-4}, etc. The limiting factor is that we must maintain that $\langle \mathbf{c}', \mathbf{s}\rangle$ has coefficients sufficiently smaller than q_L^2, in order to be able to use Lemma 1. Clearly, if $\mathbf{c}' = \mathbf{c} + q\kappa$ where all the coefficients of κ are smaller than some bound B (in absolute value), then the coefficients of $\langle \mathbf{c}', \mathbf{s}\rangle$ can be larger than the coefficients of $Z = \langle \mathbf{c}, \mathbf{s}\rangle$ (in absolute value) by at most $c_m \cdot q \cdot B \cdot \|\mathbf{s}\|_1$. (Heuristically we expect the difference to depend on the l_2 norm of $\mathbf{s}$ more than its l_1 norm.)

If we choose our parameters such that the l_1-norm of $\mathbf{s}$ is below m, and work over a ring with $c_m = O(1)$, then the coefficients of Z can be made as small as $c_m \cdot m \cdot q$, and we can make the coefficients of κ as large as $B \approx q/(4c_m \cdot m)$ in absolute value while maintaining the invariant that the coefficients of Z' are smaller than $q^2/4$ (which

is what we need to be able to use Lemma 1). By choosing an appropriate κ, we can ensure that the least significant $\lfloor \log(q/(4c_m m)) \rfloor = r - \lceil \log(4c_m m) \rceil$ bits of $\mathbf{c}'$ are all zero. This means that we can implement bit extraction using only polynomials of degree at most $2^{\lceil \log(4c_m m) \rceil} < 8c_m m = O(m)$. (Heuristically, we should even be able to get polynomials of degree $O(\sqrt{m})$ since the l_2 norm of $\mathbf{s}$ is only $O(\sqrt{m})$.) Moreover if we assume that ring-LWE is hard even with a very sparse secret, then we can use a secret key with even smaller norm and get the same reduction in the degree of the bit-extraction routine.

4 Homomorphic Decryption with Packed Ciphertexts

The homomorphic decryption procedure from Section 3 is rather inefficient, mostly because we need to repeat the bit-extraction procedure from Figure 1 for each coefficient separately. Instead, we would like to pack many coefficients in one ciphertext and extract the top bits of all of them together. To this end we employ a batching technique, similar to [1, 11, 14], using Chinese remaindering over the ring of polynomials to pack many "plaintext slots" inside a single plaintext polynomial.

Recall that the BGV scheme is defined over a polynomial ring $R = \mathbb{Z}[X]/F(X)$. If the polynomial $F(X)$ factors modulo two into distinct irreducible polynomials $F_0(X) \times \cdots \times F_{\ell-1}(X)$, then, by the Chinese Remainder Theorem, the plaintext space factors into a product of finite fields $R_2 \cong \mathbb{F}_2[X]/F_0(X) \times \cdots \times \mathbb{F}_2[X]/F_{\ell-1}(X)$.

This factorization is used in [14, 1, 11] to "pack" a vector of ℓ elements (one from each $\mathbb{F}_2[X]/F_i(X)$) into one plaintext polynomial, which is then encrypted in one ciphertext; each of the ℓ components called a plaintext slot. The homomorphic operations (add/mult) are then applied to the different slots in a SIMD fashion. When $F(X)$ is the m-th cyclotomic polynomial, $F(X) = \Phi_m(X)$, then the field $\mathbb{Q}[X]/F(X)$ is Galois (indeed Abelian) and so the polynomials $F_i(X)$ all have the same degree (which we will denote by d). It was shown in [11] how to evaluate homomorphically the application of the Galois group on the slots, and in particular this enables homomorphically performing arbitrary permutations on the vector of slots in time quasi-linear in m. This, in turn, is used in [11] to evaluate arbitrary arithmetic circuits (of average width $\tilde{\Omega}(\lambda)$) with overhead only polylog(λ).

However, the prior work only mentions the case of plaintext spaces taken modulo a prime (in our case two), i.e. R_2. In this work we will need to also consider plaintext spaces which are given by a power of a prime, i.e. $R_{2^{r+1}}$ for some positive integer r. (We stress that by R_{2^t} we really do mean $(\mathbb{Z}/2^t\mathbb{Z})[X]/F(X)$ and not $\mathbb{F}_{2^t}[X]/F(X)$.) In the full version [10] we show how the techniques from [11] extends also to this case. The "high brow" way of seeing this is to consider the message space modulo 2^{r+1} as the precision $r+1$ approximation to the 2-adic integers; namely we need to consider the localization of the field $K = \mathbb{Q}[X]/F(X)$ at the prime 2.

4.1 Using SIMD Techniques for Bootstrapping

Using the techniques from [11] for bootstrapping is not quite straightforward, however. The main difficulty is that the input and output of are not presented in a packed form:

The input is a single q_L-ciphertext that encrypts a single plaintext polynomial a (which may or may not have many plaintext elements packed in its slots), and similarly the output needs to be a single ciphertext that encrypts the same polynomial a, but with respect to a larger modulus. (We stress that this is not an artifact of our "simpler decryption formula", we would need to overcome the same difficulty also if we tried to use these "SIMD techniques" to speed up bootstrapping under the standard approach of emulating the binary mod-q_L circuit.) Our "packed bootstrapping" procedure consists of the following steps:

1. Using the encryption of the q_L-secret-key with respect to the modulus q_0, we convert the initial q_L-ciphertext into a q_0-ciphertext encrypting the polynomial $Z \in (\mathbb{Z}/2^{r+1}\mathbb{Z})[X]/\Phi_m(X)$.
2. Next we apply a homomorphic inverse-DFT transformation to get encryption of polynomials that have the coefficients of Z in their plaintext slots.
3. Now that we have the coefficients of Z in the plaintext slots, we apply the bit extraction procedure to all these slots in parallel. The result is encryption of polynomials that have the coefficients of a in their plaintext slots.
4. Finally, we apply a homomorphic DFT transformation to get back a ciphertext that encrypts the polynomial a itself.

Below we describe each of these steps in more detail. We note that the main challenge is to get an efficient implementation of Steps 2 and 4.

4.2 Encrypting the q_L-Secret-Key

As in Section 3, we use an encryption scheme with underlying plaintext space modulo 2^{r+1} to encrypt the q_L-secret-key $\mathbf{s}$ under the q_0-secret-key $\tilde{\mathbf{s}}$. The q_L-secret-key is a vector $\mathbf{s} = (1, \mathfrak{s})$, where $\mathfrak{s} \in \mathbb{Z}[X]/\Phi_m(X)$ is an integer polynomial with small coefficients. Viewing these small coefficients as elements in $\mathbb{Z}/2^{r+1}\mathbb{Z}$, we encrypt $\mathfrak{s}$ as a q_0-ciphertext $\tilde{\mathbf{c}} = (\tilde{\mathfrak{c}}_0, \tilde{\mathfrak{c}}_1)$ with respect to the q_0-secret-key $\tilde{\mathbf{s}} = (1, \tilde{\mathfrak{s}})$, namely we have

$$[\langle \tilde{\mathbf{c}}, \tilde{\mathbf{s}} \rangle \bmod \Phi_m]_{q_0} = [\tilde{\mathfrak{c}}_0 + \tilde{\mathfrak{c}}_1 \cdot \tilde{\mathfrak{s}} \bmod \Phi_m]_{q_0} = 2^{r+1}\tilde{k} + \mathbf{s} \quad \text{(equality over } \mathbb{Z}[X])$$

for some polynomial $\tilde{k}$ with small coefficients.

4.3 Step One: Computing Z Homomorphically

Given a q_L-ciphertext $\mathbf{c} = (c_0, c_1)$ we recall from the public key the q_0 ciphertext $\tilde{\mathbf{c}} = (\tilde{\mathfrak{c}}_0, \tilde{\mathfrak{c}}_1)$ that encrypts $\mathfrak{s}$, then compute the mod-2^{r+1} inner product homomorphically by setting

$$\tilde{\mathbf{z}} = \big([c_0 + c_1\tilde{\mathfrak{c}}_0 \bmod \Phi_m]_{q_0},\ [c_1\tilde{\mathfrak{c}}_1 \bmod \Phi_m]_{q_0} \big). \tag{2}$$

We claim that $\tilde{\mathbf{z}}$ is a q_0-ciphertext encrypting our Z with respect to the secret key $\tilde{\mathbf{s}}$ (and plaintext space modulo 2^{r+1}). To see that, recall that we have the following two equalities over $\mathbb{Z}[X]$,

$$(c_0 + c_1\mathfrak{s} \bmod \Phi_m) = 2^{r+1}k + Z \quad \text{and} \quad (\tilde{\mathfrak{c}}_0 + \tilde{\mathfrak{c}}_1\tilde{\mathfrak{s}} \bmod \Phi_m) = q_0\tilde{k} + 2^{r+1}\tilde{k}' + \mathfrak{s},$$

where $k, \tilde{k}, \tilde{k}' \in \mathbb{Z}[X]/\Phi_m$, the coefficients of $2^{r+1}k + Z$ are smaller than $2q_L^2 \ll q_0$, and the coefficients of $2^{r+1}\tilde{k}' + \mathfrak{s}$ are also much smaller than q_0. It follows that:

$$\begin{aligned}
(\langle \tilde{\mathbf{z}}, \tilde{\mathbf{s}} \rangle \bmod \Phi_m) &= [c_0' + c_1\tilde{\mathfrak{c}}_0 \bmod \Phi_m]_{q_0} + (\tilde{\mathfrak{s}} \cdot [c_1\tilde{\mathfrak{c}}_1 \bmod \Phi_m]_{q_0} \bmod \Phi_m) \\
&= (c_0' + c_1(\tilde{\mathfrak{c}}_0 + \tilde{\mathfrak{c}}_1\tilde{\mathfrak{s}}) \bmod \Phi_m) + q_0\kappa \\
&= (c_0' + c_1(2^{r+1}\tilde{k}' + \mathfrak{s}) \bmod \Phi_m) + q_0\kappa' \\
&= (c_0' + c_1\mathfrak{s} \bmod \Phi_m) + q_0\kappa' + 2^{r+1}(c_1 \cdot \tilde{k}' \bmod \Phi_m) \\
&= q_0\kappa' + 2^{r+1}(k + c_1\tilde{k}' \bmod \Phi_m) + Z \qquad \text{(equality over } \mathbb{Z}[X])
\end{aligned}$$

for some $\kappa, \kappa' \in \mathbb{Z}[X]/\Phi_m$. Moreover, since the coefficients of c_1 are smaller than $q_L \ll q_0$ then the coefficients of $2^{r+1}(k + c_1\tilde{k}' \bmod \Phi_m) + Z$ are still much smaller than q_0. Hence $\tilde{\mathbf{z}}$ is decrypted under $\tilde{\mathbf{s}}$ and q_0 to Z, with plaintext space 2^{r+1}.

4.4 Step Two: Switching to CRT Representation

Now that we have an encryption of the polynomial Z, we want to perform the homomorphic bit-extraction procedure from Figure 1. However, this procedure should be applied to each coefficient of Z separately, which is not directly supported by the native homomorphism of our cryptosystem. (For example, homomorphically squaring the ciphertext yields an encryption of the polynomial $Z^2 \bmod \Phi_m$ rather than squaring each coefficient of Z separately.) We therefore need to convert $\tilde{\mathbf{z}}$ to CRT-based "packed" ciphertexts that hold the coefficients of Z in their plaintext slots.

The system parameter m was chosen so that $m = \tilde{\Theta}(\lambda)$ and $\Phi_m(X)$ factors modulo 2 (and therefore also modulo 2^{r+1}) as a product of degree-d polynomials with $d = O(\log m)$, $\Phi_m(X) = \prod_{j=0}^{\ell-1} F_j(X) \pmod{2^{r+1}}$. This allows us to view the plaintext polynomial $Z(X)$ as having ℓ slots, with the j'th slot holding the value $Z(X) \bmod (F_j(X), 2^{r+1})$. This way, adding/multipliying/squaring the plaintext polynomials has the effect of applying the same operation on each of the slots separately.

In our case, we have $\phi(m)$ coefficients of $Z(X)$ that we want to put in the plaintext slots, and each ciphertext has only $\ell = \phi(m)/d$ slots, so we need d ciphertexts to holds them all. The transformation from the single ciphertext $\tilde{\mathbf{z}}$ that encrypts Z itself to the collection of d ciphertexts that hold the coefficients of Z in their slots is described in Section 4.7 below. (We describe that step last, since it is the most complicated and it builds on machinery that we develop for Step Four in Section 4.6.)

4.5 Step Three: Extracting the Relevant Bits

Once we have the coefficients of Z in the plaintext slots, we can just repeat the procedure from Figure 1. The input to the the bit-extraction procedure is a collection of some d ciphertexts, each of them holding $\ell = \phi(m)/d$ of the coefficients of Z in its ℓ plaintext slots. (Recall that we chose $m = \tilde{O}(\lambda)$ such that $d = O(\log m)$.) Applying the procedure from Figure 1 to these ciphertexts will implicitly apply the bit extraction of Lemma 2 to each plaintext slot, thus leaving us with a collection of d ciphertexts, each holding ℓ of the coefficients of a in its plaintext slots.

4.6 Step Four: Switching Back to Coefficient Representation

To finally complete the recryption process, we need to convert the d ciphertexts holding the coefficients of a in their plaintext slots into a single ciphertext that encrypts the polynomial a itself. For this transformation, we appeal to the result of Gentry et al. [11], which says that every depth-L circuit of average-width $\tilde{\Omega}(\lambda)$ and size T can be evaluated homomorphically in time $O(T) \cdot \text{poly}(L, \log \lambda)$, provided that the inputs and outputs are presented in a packed form. Below we show that the transformation we seek can be computed on cleartext by a circuit of size $T = \tilde{O}(m)$ and depth $L = \text{polylog}(m)$, and hence (since $m = \tilde{\Theta}(\lambda)$) it can be evaluated homomorphically in time $\tilde{O}(m) = \tilde{O}(\lambda)$.

To use the result of Gentry et al. we must first reconcile an apparent "type mismatch": that result requires that both input and output be presented in a packed CRT form, whereas we have input in CRT form but output in coefficient form. We therefore must interpret the output as "something in CRT representation" before we can use the result from [11]. The solution is obvious: since we want the output to be a in coefficient representation, then it is a polynomial that holds the value $A_j = a \bmod F_j$ in the j'th slot for all j.

Hence the transformation that we wish to compute takes as input the coefficients of the polynomials $a(X)$, and produces as output the polynomials $A_j = a \bmod F_j$ for $j = 0, 1, \ldots, \ell - 1$. It is important to note that our output consists of ℓ values, each of them a degree-d binary polynomial. Since this output is produced by an arithmetic circuit, then we need a circuit that operates on degree-d binary polynomials, in other words an arithmetic circuit over $\mathbb{GF}(2^d)$. This circuit has $\ell \cdot d$ inputs (all of which happen to be elements of the base field $\mathbb{F}_2$), and ℓ outputs that belong to the extension field $\mathbb{GF}(2^d)$.

Theorem 1. *Fix $m \in \mathbb{Z}$, let $d \in \mathbb{Z}$ be the smallest such that $m | 2^d - 1$, denote $\ell = \phi(m)/d$ and let $G \in \mathbb{F}_2[X]$ be a degree-d irreducible polynomial over $\mathbb{F}_2$ (that fixes a particular representation of $\mathbb{GF}(2^d)$). Let $F_0(X), F_1(X), \ldots, F_{\ell-1}(X)$ be the irreducible (degree-d) factors of the m-th cyclotomic polynomial $\Phi_m(X)$ modulo 2.*

Then there is an arithmetic circuit Π_m over $\mathbb{F}_2[X]/G(X) = \mathbb{GF}(2^d)$ with $\phi(m)$ inputs $a_0, a_1, \ldots, a_{\phi(m)-1}$ and ℓ outputs $z_0, z_1, \ldots, z_{\ell-1}$, for which the following conditions hold:

- *When the inputs are from the base field ($a_i \in \mathbb{F}_2 \ \forall i$) and we denote $a(X) = \sum_i a_i X^i \in \mathbb{F}_2[X]$, then the outputs satisfy $z_j = a(X) \bmod (F_j(X), 2) \in \mathbb{F}_2[X]/G(X)$.*
- *Π_m has depth $O(\log m)$ and size $O(m \log m)$.*

The proof is in the full version. An immediate corollary of Theorem 1 and the Gentry et al. result [11, Thm. 3], we have:

Corollary 2. *There is an efficient procedure that given d ciphertexts, encrypting d polynomials that hold the coefficients of a in their slots, computes a single ciphertext encrypting a. The procedure works in time $O(m) \cdot \text{polylog}(m)$ (and uses at most $\text{polylog}(m)$ levels of homomorphic evaluation).*

4.7 Details of Step Two

The transformation of Step Two is roughly the inverse of the transformation that we described above for Step Four, with some added complications. In this step, we have the polynomial $Z(X)$ over the ring $\mathbb{Z}/2^{r+1}\mathbb{Z}$, and we view it as defining ℓ plaintext slots with the j'th slot containing $B_j \stackrel{\text{def}}{=} Z \bmod (F_j, 2^{r+1})$. Note that the B_j's are degree-d polynomials, and we consider them as elements in the "extension ring" $R_{2^{r+1}}^d \stackrel{\text{def}}{=} \mathbb{Z}[X]/(G(X), 2^{r+1})$ (where G is some fixed irreducible degree-d polynomial modulo 2^{r+1}).

Analogous to Theorem 1, we would like to argue that there is an arithmetic circuit over $R_{2^{r+1}}^d$ that get as input the B_j's (as elements of $R_{2^{r+1}}^d$), and outputs all the coefficients of Z (which are elements of the base ring $\mathbb{Z}/2^{r+1}\mathbb{Z}$). Then we could apply again to the result of Gentry et al. [11] to conclude that this circuit can be evaluated homomorphically with only polylog overhead.

For the current step, however, the arithmetic circuit would contain not only addition and multiplication gates, but also Frobenius map gates. Namely, gates $\rho_k(\cdot)$ (for $k \in \{1, 2, \ldots, d-1\}$) computing the functions

$$\rho_k\big(u(X)\big) \;=\; u(X^{2^k}) \bmod (G(X), 2^{r+1}).$$

It was shown in [11] that arithmetic circuits with Frobenius map gates can also be evaluated homomorphically with only polylog overhead. The Frobenius operations being simply an additional automorphism operation which can be applied homomorphically to ciphertexts.

Theorem 2. *Fix $m, r \in \mathbb{Z}$, let $d \in \mathbb{Z}$ be the smallest such that $m | 2^d - 1$, denote $\ell = \phi(m)/d$ and let $G(X)$ be a degree-d irreducible polynomial over $\mathbb{Z}/2^{r+1}\mathbb{Z}$ (that fixes a particular representation of $R_{2^{r+1}}^d$). Let $F_0(X), F_1(X), \ldots, F_{\ell-1}(X)$ be the irreducible (degree-d) factors of the m-th cyclotomic polynomial $\Phi_m(X)$ modulo 2^{r+1}.*

Then there is an arithmetic circuit $\Psi_{m,r}$ with Frobenius-map gates over $R_{2^{r+1}}^d$ that has ℓ input $B_0, B_1, \ldots, B_{\ell-1}$ and $\phi(m)$ outputs $Z_0, Z_1, \ldots, Z_{\phi(m)-1}$, for which the following conditions hold:

- *On any inputs $B_0, \ldots, B_{\ell-1} \in R_{2^{r+1}}^d$, the outputs of $\Psi_{m,r}$ are all in the base ring, $Z_i \in \mathbb{Z}/2^{r+1}\mathbb{Z}\ \forall i$. Moreover, denoting $Z(X) = \sum_i Z_i X^i$, it holds that $Z(X) \bmod (F_j(X), 2^{r+1}) = B_j$ for all j.*
- *Π_m has depth $O(\log m + d)$ and size $O(m(d + \log m))$.*

The proof is in the full version. As before, a corollary of Theorem 2 and the result from [11], is the following:

Corollary 3. *There is an efficient procedure that given a single ciphertext encrypting Z' outputs d ciphertexts encrypting d polynomials that hold the coefficients of Z' in their plaintext slots. The procedure works in time $\tilde{O}(m)$ (and uses at most polylog(m) levels of homomorphic evaluation).*

4.8 An Alternative Variant

The procedure from Section 4.7 works in time $\tilde{O}(m)$, but it is still quite expensive. One alternative is to put in the public key not just one ciphertext encrypting the q_L-secret-key $\mathfrak{s}$, but rather d ciphertexts encrypting polynomials that hold the coefficients of $\mathfrak{s}$ in their plaintext slots. Then, rather than using the simple formula from Equation (2) above, we evaluate homomorphically the inner product of $\mathbf{s} = (1, \mathfrak{s})$ and $\mathbf{c} = (c_0, c_1)$ modulo $\Phi_m(X)$ and 2^{r+1}. This procedure will be even faster if instead of the coefficients of $\mathfrak{s}$ we encrypt their transformed image under length-m DFT. Then we can compute the DFT of c_1 (in the clear), multiply it homomorphically by the encrypted transformed $\mathfrak{s}$ (in SIMD fashion) and then homomorphically compute the inverse-DFT and the reduction modulo Φ_m. Unfortunately this procedure still requires that we compute the reduction mod-$\Phi_m(X)$ homomorphically, which is likely to be the most complicated part of bootstrapping. Finding a method that does not require this homomorphic polynomial modular reduction is an interesting open problem.

Acknowledgments. The first and second authors are sponsored by DARPA and ONR under agreement number N00014-11C-0390. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, or the U.S. Government. Distribution Statement "A" (Approved for Public Release, Distribution Unlimited).

The third author is sponsored by DARPA and AFRL under agreement number FA8750-11-2-0079. The same disclaimers as above apply. He is also supported by the European Commission through the ICT Programme under Contract ICT-2007-216676 ECRYPT II and via an ERC Advanced Grant ERC-2010-AdG-267188-CRIPTO, by EPSRC via grant COED–EP/I03126X, and by a Royal Society Wolfson Merit Award. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the European Commission or EPSRC.

References

1. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: Fully homomorphic encryption without bootstrapping. In: Innovations in Theoretical Computer Science, ITCS 2012 (2012), http://eprint.iacr.org/2011/277
2. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: FOCS 2011. IEEE Computer Society (2011)
3. Brakerski, Z., Vaikuntanathan, V.: Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 505–524. Springer, Heidelberg (2011)
4. Cassels, J.W.S.: Local Fields. LMS Student Texts, vol. 3. Cambridge University Press (1986)
5. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully Homomorphic Encryption over the Integers. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24–43. Springer, Heidelberg (2010), Full version available on-line from http://eprint.iacr.org/2009/616

6. Gentry, C.: A fully homomorphic encryption scheme. PhD thesis, Stanford University (2009), http://crypto.stanford.edu/craig
7. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher, M. (ed.) STOC, pp. 169–178. ACM (2009)
8. Gentry, C., Halevi, S.: Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. In: FOCS 2011. IEEE Computer Society (2011)
9. Gentry, C., Halevi, S.: Implementing Gentry's Fully-Homomorphic Encryption Scheme. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 129–148. Springer, Heidelberg (2011)
10. Gentry, C., Halevi, S., Smart, N.P.: Better bootstrapping for fully homomorphic encryption (2011), http://eprint.iacr.org/2011/680
11. Gentry, C., Halevi, S., Smart, N.P.: Fully Homomorphic Encryption with Polylog Overhead. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 465–482. Springer, Heidelberg (2012), Full version at http://eprint.iacr.org/2011/566
12. Rivest, R., Adleman, L., Dertouzos, M.L.: On data banks and privacy homomorphisms. In: Foundations of Secure Computation, pp. 169–180 (1978)
13. Smart, N.P., Vercauteren, F.: Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 420–443. Springer, Heidelberg (2010)
14. Smart, N.P., Vercauteren, F.: Fully homomorphic SIMD operations (2011) (manuscript), http://eprint.iacr.org/2011/133

Polly Cracker, Revisited, Revisited

Gottfried Herold*

Ruhr-Universität Bochum
Horst Görtz Insitute for IT-Security
Bochum, Germany
gottfried.herold@rub.de

Abstract. In this paper, we consider the Polly Cracker with Noise (PCN) cryptosystem by Albrecht, Farshim, Faugère, and Perret (Asiacrypt 2011), which is a public-key cryptosystem based on the hardness of computing Gröbner bases for noisy random systems of multivariate equations. We examine four settings, covering all possible parameter ranges of PCN with zero-degree noise. In the first setting, the PCN cryptosystem is known to be equivalent to Regev's LWE-based scheme. In the second, it is known to be at most as secure as Regev's scheme. We show that for one other settings it is equivalent to a variants of Regev's with less efficiency and in the last setting it is completely insecure and we give an efficient key-recovery attack. Unrelated to the attack, we also fix some flaws in the security proofs of PCN.

Keywords: Polly Cracker with Noise, Learning with Errors, Gröbner bases, Cryptanalysis.

1 Introduction

Background. By the term Polly Cracker-type cryptosystem, we mean a family of cryptosystems starting from the early 1990s that propose to base their security on the difficulty of computing Gröbner bases ([8,2]). In its public key version and the most simple form, the public key is an ideal I in a polynomial ring (given by sufficiently many polynomials of degree b from I) and the secret key is a Gröbner basis for I consisting of polynomials of degree $d \leq b$. These systems mostly lack a formal treatment of security and almost all of them have been broken due fundamental limitations in the construction([2,1]). See [7] for a good survey on various instantiations and attacks.

Recently, at Asiacrypt 2011, Albrecht, Farshim, Faugère, and Perret [1] proposed a new cryptosystem called Polly Cracker with Noise (PCN) that tries to overcome these limitations. Their cryptosystem can be seen both as a high-dimensional generalization of Regev's LWE-based scheme [12] and a noisy generalization of the Polly Cracker-style cryptosystems. They also give a formal proof of security, based on the hardness of computational problems related to Gröbner

* Due to space limitations, this version does not contain the proofs of Thm. 3. These are contained in the full version, available on eprint.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 17–33, 2012.

bases and ideals in multivariate polynomial rings. Note that this paper refers mainly to the full version of [1] on eprint, which contains more material than the proceedings version.

One of the appealing features of the PCN cryptosystem comes from its ideal-theoretic framework. In this framework it is prominently visible that the PCN cryptosystem, which contains LWE as a special case, is both multiplicatively and additively homomorphic for a limited number of operations. For the special case of LWE, the recent fully homomorphic scheme by Brakerski and Vaikuntanathan from FOCS 2011 [4] can be represented in this framework.

Our Contributions. Our first result is that the Polly Cracker with Noise cryptosystem with zero-degree noise is either insecure or does not offer any security benefit (although still a conceptual one) compared to Regev's scheme. For $b > d > 1$, we present an efficient attack that recovers the secret key from the public key. For $d = 1$, the security of the PCN cryptosystem is at most that of Regev's scheme by [1]. For $d = b > 1$, PCN has the same security as Regev's schem, but with less efficiency. The only remaining case $b = d = 1$ is exactly Regev's scheme by [1].

Note that zero-degree noise is used for the homomorphic properties claimed in [1], cf. Sect. 2.3.

As a second result, we point out flaws in the security proofs of [1], giving counterexamples to the statements claimed therein. We then give corrected proofs for $d = 1$, thereby showing their security proofs only work for $d = 1$. Note that the attack against $b > d > 1$ is unrelated to these flaws. Due to space limitations, the proofs are only contained in the full version, available on eprint.

Organization of this Work. This work is organized as follows: In Section 2, we start by introducing some notation and recalling the Polly Cracker with Noise cryptosystem and its security assumptions. In Section 3, we relate the PCN cryptosystem to Regev's scheme for $b = d$ and for $d = 1$.

In Section 4, we give counterexamples to the security proofs of [1] and give corrected statements for $d = 1$.

In Section 5, we present our key-recovery attack for $b > d > 1$.

2 The Polly Cracker with Noise Cryptosystem

2.1 Gröbner Bases

In this section, we introduce some notation and recall some facts regarding Gröbner bases [5]. For a more detailed exposition, see e.g. [6].

Let $P = \mathbb{F}_q[X_1, X_2, \ldots, X_n]$ be a polynomial ring and $<$ be a fixed monomial ordering for its monomials. For a subspace $Q \subset P$, we denote by $Q_{<k}, Q_{=k}, Q_{\leq k}$ the restriction of Q to polynomials of total degree $< k, = k, \leq k$, respectively. We shall always assume that q is odd, for simplicity prime, and that the monomial ordering is compatible with the total degree of monomials (e.g. `deglex` or `degrevlex`), i.e. $\deg f < \deg g$ implies $f < g$ for all monomials $f, g \in P$,

where deg denotes total degree. W.l.o.g. we may assume $X_1 < X_2 < \ldots < X_n$. For a polynomial $f \in P$, let $\mathrm{LC}(f), \mathrm{LM}(f), \mathrm{LT}(f)$ denote the leading coefficient, monomial and term, respectively. We always represent polynomials $f \in P, f = \sum_{m \leq \mathrm{LM}(f)} f_m \cdot m$ by their dense coefficient representation, i.e. the list of the f_m. Note that for degree-compatible $<$, the length of this list is at most $\dim P_{\leq \deg f} = \binom{n+\deg f}{\deg f}$, which is polynomial in n for fixed $\deg f$.

Definition 1. Gröbner basis
Let $I \subset P$ be an ideal. A finite set $G = \{g_1, \ldots, g_l\}$ is called a Gröbner basis for I if G generates I as an ideal and if for every $f \in I$, there is a $g_i \in G$ such that $\mathrm{LM}(g_i) \mid \mathrm{LM}(f)$.

If additionally, $\mathrm{LC}(g_i) = 1$ *for all i and no term of g_i is divisible by* $\mathrm{LC}(g_j)$ *for $i \neq j$, we call G a* reduced *Gröbner basis.*

Every ideal $I \subset P$ has a Gröbner basis G. If one additionally insists on G being reduced, G is unique. For any $f \in P$, we can use the multivariate polynomial division algorithm to compute the remainder, denoted $f \bmod G$. The central property of a Gröbner basis G is that $f \bmod G$ is unique. We use this property to identify P/I with the set of remainders, thus viewing $P/I \subset P$. As a vector space, P/I is generated by those monomials not divisible by any $\mathrm{LM}(g_i)$ and we always have $P = (P/I) \oplus I$.

2.2 Polly Cracker with Noise

In this section, we briefly recall the (symmetric key variant of the) Polly Cracker with Noise(PCN) cryptosystem.

The secret key of this cryptosystem is a Gröbner basis G for some ideal $I \subset P$. Ciphertexts are noisy samples from I, where the message is appropriately embedded in the noise. More precisely, we encrypt a message bit $M \in \{0, 1\}$ as $f + 2e + M$, where $f \leftarrow_\$ I$ and $e \leftarrow_\$ \mathcal{X}$ from some noise distribution $\mathcal{X}$ on P/I. We can decrypt c by computing $M = (c \bmod G) \bmod 2$, provided the noise e is small enough.

In more detail, let us consider $P = \mathbb{F}_q[X_1, \ldots, X_n]$ and $<$ as above. We will also need to fix some integers $0 < d \leq b$, which will denote the degree of the Gröbner basis polynomials and the message polynomials, respectively. The parameters $q = q(\lambda), n = n(\lambda)$ will be implicitly functions of the security parameter λ, with $\log q = \mathrm{poly}(\lambda)$ (sometimes even $q = \mathrm{poly}(\lambda)$), $n = \mathrm{poly}(\lambda)$ and $n^d = \Omega(\lambda)$ (so $\mathrm{poly}(n) = \mathrm{poly}(\lambda)$). Note that we assume b, d *not* to depend on the security parameter λ.

The secret key of our cryptosystem will be a (reduced) Gröbner basis $G = \{g_1, \ldots, g_n\}$ for some ideal I, so we need an algorithm to generate Gröbner bases. In general, we require $\mathbf{Gen}(1^\lambda)$ to be a ppt algorithm outputting a reduced Gröbner basis $G = \{g_1, \ldots, g_k\}$ for an ideal $I \subsetneq P$ with $\deg g_i \leq d$.

For definiteness, we will restrict our attention in Sect. 5 to the key generation algorithm suggested in [1] called $\mathrm{GBGen}_{\mathrm{dense}}$.

Algorithm 1. $\mathrm{GBGen}_{\mathrm{dense}}$

function $\mathrm{GBGen}_{\mathrm{dense}}(1^\lambda)$:
  for $i = 1$ to n **do**
    $g_i \leftarrow X_i^d$
    for all monomials $m \in P_{\leq d}$ with $m < X_i^d$ and $m \neq X_j^d$ for any j **do**
      $g_{i,m} \leftarrow_\$ F_q$ uniformly
      $g_i \leftarrow g_i + g_{i,m} \cdot m$
  return $G = \{g_1, \ldots, g_n\}$

Writing each g_i as $g_i = \sum_m g_{i,m} \cdot m$ where m runs over the possible monomials of P, $\mathrm{GBGen}_{\mathrm{dense}}$ sets the leading term of g_i to be X_i^d. The coefficients of smaller monomials are chosen uniformly and independently at random.

Buchberger's first criterion (cf. [3, Lemma 5.66, p. 222] or [6, section 2.9, pp. 99–108]) guarantees that this is indeed a Gröbner basis for its generated ideal $I = (g_1, \ldots, g_n)$. Setting all coefficients of X_j^d in g_i to be 0 for $i \neq j$ guarantees that G is a *reduced* Gröbner basis. Note that sampling these coefficients at random as well and then reducing the Gröbner basis afterward, as originally done in [1], gives the same output distribution.

We denote by $\mathcal{Q} = P/I$ the quotient ring and identify it with a subspace $\mathcal{Q} \subset P$ as above, such that $P = I \oplus \mathcal{Q}$.

With G generated by $\mathrm{GBGen}_{\mathrm{dense}}$, $\mathcal{Q}$ is always finite-dimensional and a basis is given by $\left\{X_1^{t_1} \cdots X_n^{t_n} \mid t_i < d\right\}$. Note that this does not depend on the randomness of $\mathrm{GBGen}_{\mathrm{dense}}$ and for simplicity we shall always assume that $\mathcal{Q} \subset P$ is finite-dimensional and a basis for $\mathcal{Q}_{\leq b}$ is publicly known, even for general **Gen**. It follows that for $d = 1$, $\mathcal{Q} = \mathbb{F}_q$ is just the field of constants in P. In the case $d > 1$, the full quotient $\mathcal{Q}$ has exponential dimension $\dim \mathcal{Q} = d^n$, essentially due to the lack of a fixed bound on *total* degree. In this case, our cryptosystem will only make use of the polynomially-dimensional subspace $\mathcal{Q}_{\leq b} \subset \mathcal{Q}$.

Let $\mathcal{X}$ be an efficiently sampleable noise distribution on $\mathcal{Q}_{\leq b}$. The distributions we will later be concerned with will be either uniform or discrete Gaussian distributions on vector sub-spaces. In the case of Gaussians, this will mean we independently sample each coefficient of $e \leftarrow \mathcal{X}$ in a particular basis from a discrete Gaussian distribution.

By the *support* S of a probability distribution Φ on a finite set Ω, we mean those elements of Ω that are assigned a non-zero probability by Φ.

Using the Gröbner basis G for I, we can obtain noisy samples from $I_{\leq b} + \mathcal{X}$ by applying algorithm 2.[1]

By SampleI without subscript, we denote the special case of noiseless sampling from $I_{\leq b}$ (i.e. with $e = 0$ above).

Following [1], we note that SampleI actually samples uniformly from $I_{\leq b}$ and also give an alternative sampling algorithm, whose equivalence we will need later on:

[1] Identifying a set with the uniform distribution on it, $I_{\leq b} + \mathcal{X}$ actually is the output distribution of the algorithm.

Algorithm 2. $\text{SampleI}_{\mathcal{X}}$

1: **function** $\text{SampleI}_{\mathcal{X}}$(G,b):
2: $f \leftarrow_{\$} P_{\leq b}$ uniformly
3: $e \leftarrow_{\$} \mathcal{X}$
4: $f := f - (f \bmod G) + e$
5: **return** f.

Lemma 1. *For any Gröbner basis $G = (g_1, \ldots, g_m)$ for I,* $\text{SampleI}(G, b)$ *yields uniform samples from $I_{\leq b}$.*

Furthermore, if $\deg g_i = d_i \leq b$ *for all g_i and the underlying monomial ordering is compatible with* deg*, we have the following alternative sampling algorithm, which gives the same distribution:*

Let $t_i \leftarrow_{\$} P_{\leq b-d_i}$ uniformly for $i \in \{1, \ldots, m\}$ and sample $f \in I$ as $f = \sum_{i=1}^{m} t_i \cdot g_i$.

Proof. Clearly, both ways of sampling give us polynomials from $I_{\leq b}$. We observe that both $f \mapsto f \bmod G$ and $(t_1, \ldots t_m) \mapsto f = \sum_{i=1}^{m} t_i \cdot g_i$ are $\mathbb{F}_q$-linear maps. Since surjective linear maps preserve uniform distributions, both resulting distributions are uniform on their respective supports.

For SampleI, the support is clearly all of $I_{\leq b}$, since we may choose any element from $I_{\leq b}$ in step 2 of the algorithm.

For the alternative sampling, we note that for $f \in I_{\leq b}$, the multivariate polynomial division algorithm for $f \bmod G$ gives us a (typically non-unique) representation $f = \sum_i t_i g_i$. Since $<$ is compatible with deg, the intermediate results in that computation have degree $\leq b$, which ensures that $\deg t_i \leq b - d_i$. This already proves the claim.

To encrypt a message bit $M \in \{0, 1\}$, we proceed as follows:

Algorithm 3. Enc_G

1: **function** Enc_G(M):
2: $f \leftarrow \text{SampleI}(G)$
3: $e \leftarrow_{\$} \mathcal{X}$
4: $c := f + 2e + M$
5: **return** c

Accordingly, decryption of a ciphertext $c \in P_{\leq b}$ is performed by the following algorithm, where for $f \in P$, $f_{=0}$ denotes the constant coefficient of f:

Algorithm 4. Dec_G

1: **function** Dec_G(c):
2: $M := (c \bmod G)_{=0} \in \{-\lfloor \frac{q}{2} \rfloor, \ldots, \lfloor \frac{q}{2} \rfloor\}$
3: **return** $M \bmod 2$

Decryption is correct, provided that for the noise $2e \leftarrow_\$ 2\mathcal{X}$ we have $|2e_{=0}| < \lfloor \frac{q}{2} \rfloor$. If $\mathcal{X}$ is a sufficiently narrow discrete Gaussian distribution, this will be the case with overwhelming probability.

Remark 1. Embedding the message in the noise
In algorithm Enc_G above, the message M is merely one bit and is embedded only in the degree 0 term of the noise. Hence, in algorithm Dec_G, we also take only the degree 0-coefficient $(c \bmod G)_{=0}$. In particular, this means that fake ciphertexts c not generated by Enc_G still decrypt to a bit, even if $c \bmod G$ is not in the support of $2\mathcal{X} + \{0, 1\}$. Alternatively, we could output an error in the latter case.[2]

In fact, in [1] it is implicitly assumed (and also implemented that way in the reference implementation) that the noise is completely contained in degree 0. Unfortunately, these issues are not addressed in [1] and we will show in Sect. 5 that for $d > 1$ this choice renders the system insecure for $b > d$. For $b = d$ or $d = 1$, compare the following Sect. 3, where we show that these choice offer no benefit compared to $b = d = 1$. For $b = d = 1$, the PCN cryptosystem is a reformulation of Regev's scheme.

Actually, if the message is contained only in degree 0, the coefficients belonging to the monomials of $\mathcal{Q}_{\leq d}$ other than the constant term of a ciphertext polynomial c are completely irrelevant for decryption (cf. Prop. 1, which is a special case of that).

So unless one wants to detect fake ciphertexts as mentioned above or make use of the multiplicative homomorphic properties (cf. Sect. 2.3), one should really use uniform noise for those coefficients (or just leave those coefficients out of the ciphertext altogether).

In this work we will consider the more general setting, where the message is contained in degree 0, but the noise distribution $\mathcal{X}$ on $\mathcal{Q}_{\leq b}$ is arbitrary. When we assume that the noise is concentrated in degree 0, we will explicitly state that.

2.3 Homomorphic Properties and Public Key Version

One of the appealing aspects of the PCN cryptosystem is that it is somewhat homomorphic:

$P \to \mathcal{Q}, f \mapsto f \bmod G$ is actually a ring map. This means that for ciphertexts $c_1 = f_1 + 2e_1 + M_1, c_2 = f_2 + 2e_2 + M_2$, with $f_i \in I_{\leq b}, e_i \in \mathcal{Q}_{\leq b}, M_i \in \{0, 1\}$, we have

$$c_1 + c_2 = (f_1 + f_2) + 2(e_1 + e_2) + M_1 + M_2$$

and

$$c_1 \cdot c_2 = g + 2\,(2e_1e_2 + e_1M_2 + e_2M_1 \bmod G) + (M_1M_2 \bmod G),$$

where $g \in I_{\leq 2b}$

[2] Note that if the support of $2\mathcal{X} + \{0, 1\}$ is a vector space, a CPA-attacker can check for this error himself, so this does not affect security.

From this, we get $\mathrm{Dec}_G(c_1) \dot{+} \mathrm{Dec}_G(c_2) = \mathrm{Dec}(c_1 \dot{+} c_2)$, provided that the noise of the sum/product does not grow too large.

For sums, this implies that for a sufficiently narrow Gaussian $\mathcal{X}$, the cryptosystem supports a limited number of homomorphic additions at the cost of increased noise, and still decrypts correctly with overwhelming probability.

Note that this also holds in the case that we embed several bits into one ciphertext, provided the noise is narrow coefficient-wise. Via the usual generic construction [14], these additive somewhat homomorphic properties allow to convert the secret key cryptosystem into a public key cryptosystem by publishing a sufficient amount of encryptions of 0 as the public key. Note that the same applies to Regev's scheme [12] described below in Section 3. For simplicity, in this work we deal with the secret key versions of both schemes, but it is easy to see that everything carries over directly to the public-key setting.

For multiplications, if the noise is concentrated in degree 0, we get that the noise is approximately multiplied for each multiplication of ciphertexts[3], so we can also perform a limited number of homomorphic multiplications.

If $d > 1$ and $\mathcal{X}$ is not supported in degree 0, this will actually fail if done naïvely. The reason is that even if all coefficients of e_1, e_2 are small, $e_1 \cdot e_2 \bmod G$ might have large coefficients due to reduction mod G.

This is the case even if the coefficients of the Gröbner basis polynomials are small; take for example the reduced Gröbner basis $G = (g_1, \ldots, g_{2n}) \subset \mathbb{F}_q[X_1, \ldots X_{2n}]$ with

$g_1 = X_1^2 - a_1, \quad g_2 = X_2^2 - a_2,$
$g_{2i} = X_{2i}^2 - a_{2i} X_{2i-2} X_{2i-3},$
$g_{2i+1} = X_{2i+1}^2 - a_{2i+1} X_{2i-2} X_{2i-1}$ for $i \geq 1$ and $a_i \in \mathbb{F}_q$ small.

Then for $e_1 = e_2 = X_{2n} X_{2n-1} \in \mathcal{Q}_{\leq 2}$, we have $e_1 \cdot e_2 \bmod G = \prod_{i=1}^{2n} a_i$, which is exponentially large.

This observation makes it highly desirable to concentrate the noise and message in degree 0. Unfortunately, this renders the system insecure (cf. Sect. 5) unless $d = 1$ or $b = d$. By the results of Section 3, in the latter cases, we should rather use $b = d = 1$.

2.4 Security Assumptions

[1] introduced the following three security problems related to the PCN cryptosystem:

Definition 2. *The* Gröbner basis with noise (GBN) problem $\mathrm{GBN}_{n,\mathbf{Gen},d,b,\mathcal{X}}$ *for parameters as above is defined as follows:*

Let $G \leftarrow \mathbf{Gen}(1^\lambda)$ *be a reduced Gröbner basis. Given access to a sampling oracle for* $\mathrm{Sample}\mathrm{I}_{\mathcal{X}}$*, the task is to find* G*. The advantage for a (ppt) algorithm* A *in solving the* $\mathrm{GBN}_{n,\mathbf{Gen},d,b,\mathcal{X}}$ *problem ist given as*

$$\mathbf{Adv}^{gbn}_{n,\mathbf{Gen},d,b,\mathcal{X},A}(\lambda) = \mathbf{Pr}[A \text{ solves the } \mathrm{GBN}_{n,\mathbf{Gen},d,b,\mathcal{X}}\text{-}problem] - \frac{1}{|\mathcal{G}|},$$

[3] Note that this also increases the total degree, which can be adressed by reencryption techniques[4], but this will not be important for us here.

where $\mathcal{G}$ is the set of possible secret keys and the probability is over the coins of **Gen**, SampleI *and* A.

Note that we always assume that $|\mathcal{G}|$ is exponential.

Definition 3. *The* Ideal remainder with noise problem $\mathrm{IRN}_{n,\mathbf{Gen},d,b,\mathcal{X}}$ *for parameters as above is defined as follows:*

Let $G \leftarrow \mathbf{Gen}(1^\lambda)$ and a uniformly random challenge $x \leftarrow_\$ P_{\leq b}$. Given x and access to a sampling oracle for $\mathrm{SampleI}_{\mathcal{X}}$, the task is to find $x \bmod G \in \mathcal{Q}_{\leq b}$. The advantage for a ppt algorithm B for this problem is given as

$$\mathbf{Adv}^{irn}_{n,\mathbf{Gen},d,b,\mathcal{X},B}(\lambda) = \mathbf{Pr}[B \text{ solves the } \mathrm{IRN}_{n,\mathbf{Gen},d,b,\mathcal{X}}\text{-problem}] - \frac{1}{|\mathcal{Q}_{\leq b}|},$$

where the probability is over the coins of **Gen**, SampleI, B *and the uniform choice of the challenge x.*

Note that this definition of advantage implicitly assumes that $\mathcal{Q}_{\leq b}$ is known to the attacker.

Definition 4. *The* Ideal membership with noise (IMN) problem $\mathrm{IMN}_{n,\mathbf{Gen},d,b,\mathcal{X}}$ *for parameters as above is defined as follows:*

Let $G \leftarrow \mathbf{Gen}(1^\lambda)$. Given access to a sampling oracle for $\mathrm{SampleI}_{\mathcal{X}}$, the task is to distinguish a challenge polynomial x drawn either as $x \leftarrow_\$ \mathrm{SampleI}_{\mathcal{X}}$ or as a uniform $x \in_R P_{\leq b}$. The advantage for a ppt algorithm C for this is given as

$$\mathbf{Adv}^{imn}_{n,\mathbf{Gen},d,b,\mathcal{X},C}(\lambda) = \mathbf{Pr}[C^{\mathrm{SampleI}_{\mathcal{X}}()}(x) = 1] - \mathbf{Pr}[C^{\mathrm{SampleI}_{\mathcal{X}}()}(u) = 1]$$

where $x \leftarrow_\$ \mathrm{SampleI}_{\mathcal{X}}, u \in_R \mathcal{Q}_{\leq b}$ and the probability is over the coins of **Gen**, SampleI, C *and choices of x or u. Note that we differ by a factor 2 from [1].*

The security assumption made in [1] is that for appropriate choice of parameters, namely $b \leq d \leq 1$ arbitrary, $\mathbf{Gen} = \mathrm{GBGen}_{\mathrm{dense}}$ and $\mathcal{X}$ a sufficiently broad discrete Gaussian distribution on $\mathbb{F}_q$, the advantage for any ppt algorithm is negligible for GBN / IRN / IMN.

Also, it was claimed in [1] that all of these assumptions and the IND-CPA-security of PCN are essentially equivalent:

1. The GBN problem is hard iff the IRN problem is hard.
2. For polynomially-sized $\mathcal{Q}_{\leq b}$, IRN is hard iff IMN is hard.
3. If IMN is hard, the PCN cryptosystem is IND-CPA-secure.

As their proofs of 1 and 2 contain errors (amongst other things, the reduction presents the wrong distributions to the algorithms), we will redo the proofs for 1 and 2 in Sect. 4.

Unfortunately, we will have to make additional assumptions compared to [1], most importantly we have to assume $d = 1$ for the $\Rightarrow$ direction in the first proof and for the $\Leftarrow$ direction of the second. We will also give a counterexample indicating that these additional assumptions are necessary.

3 Relations to LWE and Regev's Scheme

We will now relate the PCN cryptosystem to LWE and show that the cases $b = d$ and $d = 1$ both reduce to Regev's LWE-based scheme. Let us briefly recall the LWE distribution, the LWE assumption and Regev's scheme from [12], which has a reduction to the LWE assumption:

Definition 5. *Learning with Errors (LWE)*
Let Φ be some noise distribution on a finite field $\mathbb{F}_q$ and $n \in \mathbb{N}$ and $s \in \mathbb{F}_q^n$. The LWE distribution $\mathcal{L}_{s,\Phi}$ on $\mathbb{F}_q^n \times \mathbb{F}_q$ is obtained by sampling $a_1, \ldots, a_n \in \mathbb{F}_q$ uniformly random, $e \leftarrow_\$ \Phi$ and outputting $(a_1, \ldots, a_n, \sum a_i s_i + e)$.

The computational LWE problem $\mathrm{LWE}_{n,q,\Phi}$ is the following problem: For uniformly random $s \in \mathbb{F}_q^n$, compute s when given oracle access to $\mathcal{L}_{s,\Phi}$.

The decisional LWE problem $\mathrm{DLWE}_{n,q,\Phi}$ is the following problem: For uniformly random $s \in \mathbb{F}_q^n$, distinguish $x \in_R \mathbb{F}_q^{n+1}$ from $x \leftarrow_\$ \mathcal{L}_{s,\Phi}$ when given oracle access to $\mathcal{L}_{s,\Phi}$.

The LWE assumption (for $q, \mathcal{X}$ given functions of n) states that any ppt algorithm can only solve these problems with negligible advantage.

Definition 6. *Regev's scheme*
Let Φ be some noise distribution on a finite field $\mathbb{F}_q$. In its secret key version[4], Regev's scheme generates a secret key $s = (s_1, \ldots, s_n) \in \mathbb{F}_q^n$ uniformly. We encrypt a message $M \in \{0, 1\}$ by sampling $a = (a_1, \ldots, a_n) \in \mathbb{F}_q^n$ randomly, $e \leftarrow_\$ \Phi$ and defining the ciphertext as $(a, \langle a, s\rangle + 2e + M)$, where $\langle a, s\rangle = \sum a_i s_i$ is the scalar product.

Decryption recovers $2e + M$ and, from that, M itself, provided e is small enough.

As already noted in [1], Regev's scheme is equivalent to the PCN cryptosystem for $d = b = 1$. To see that, we can identify Regev's secret s with the Gröbner basis $G = (X_1 + s_1, \ldots, X_n + s_n)$. We identify ciphertexts (a, b) with linear polynomials $\sum a_i X_i + b$ and Φ with $\mathcal{X}$.

In fact, such a relationship also holds for $b = d > 1$ and for $b > d = 1$, where the cases with $d = 1$ were already discussed in [1]:

Theorem 1. *Relationship of PCN with LWE for $b = d$ or $d = 1$*

- *For $b = d$, the IND-CPA-security of PCN (with parameters $q, \mathcal{X}, b, d, n$) is equivalent to the IND-CPA-security of Regev's scheme (with parameters $q, \mathcal{X}, n$).*
- *For $d = 1$, there exists a tight security reduction from the IND-CPA-security of PCN (with parameters $q, \mathcal{X}, b, d, n$) to the IND-CPA-security of Regev's scheme (with parameters $q, \mathcal{X}, \binom{n+b}{b}$).*
- *For $b = d = 1$, the PCN cryptosystem is a reformulation of Regev's scheme.*

[4] The public-key version is obtained by using Rothblum's construction [14] just as with PCN and all observations carry over directly to the public-key versions of both schemes.

Proof. For $b = d$, this follows from proposition 1 below, showing that in this case the PCN cryptosystem is a redundant version of Regev's scheme. For $d = 1$, this follows from proposition 2 below, showing that in this case the PCN cryptosystem is a structured version of Regev's scheme. The case $b = d = 1$ was already discussed above.

Regarding ciphertext length, recall that the PCN-ciphertexts are $\binom{n+b}{b}$ elements from $\mathbb{F}_q$. As a consequence, for $b = d > 1$ we have a loss in efficiency, but no gain in security. For $b > 1, d = 1$, we have no gain in efficiency (apart from a shorter secret key compared to Regev's) and potentially a loss in security. Therefore, there is little point in using the PCN cryptosystem for $b = d$ or $d = 1$ unless $b = d = 1$.

Proposition 1. *Relation of PCN with LWE for $b = d$*
Consider the case $b = d$ and assume that $\mathcal{X}$ outputs $e \leftarrow_\$ \mathcal{X}, e = \sum_m e_m \cdot m$, where the sum runs over the monomials and the e_m are chosen independently, their distribution possibly depending on m (This is the case if the noise is contained in degree 0). Then the PCN cryptosystem is essentially[5] a reformulation of (the secret key version) of the amortized[6] variant [11] of Regev's scheme, where each monomial m of $\mathcal{Q}_{\leq b}$ corresponds to one parallel instance of Regev's original scheme.

To see this, consider a PCN-ciphertext c. By lemma 1, c is of the form $c = \sum t_i \cdot g_i + 2e + M$ for $e \leftarrow_\$ \mathcal{X}$ with $t_i \in \mathbb{F}_q$. Let us write $c = \sum_m c_m \cdot m$ for the monomials m of c. Then for $1 \neq m \in \mathcal{Q}_{\leq b}$, the coefficients of the ciphertext are $c_m = \sum t_i \cdot g_{i,m} + 2e_m$ and $c_1 = \sum t_i \cdot g_{i,1} + 2e_1 + M$. These are noisy random linear combinations of the secret $g_{i,m}$ as in Regev's scheme. The other $m \notin \mathcal{Q}_{\leq b}$ are $m = X_i^d$ and there we have $c_{X_i^d} = t_i$. It follows that the ciphertexts are exactly as in the amortized variant of Regev's.

When taking that point of view for general $b = d > 1$, beware that by construction, for some $m \in \mathcal{Q}_{=b}$ and some $j \in \{1, \ldots, n\}$ we can have $m \not\leq X_j^d$, so $g_{m,X_j^d} = 0$. In that case, the corresponding LWE-instance has a secret key from $\mathbb{F}_q^{n'}$ for some $n' < n$. In particular, for $\mathrm{GBGen}_{\mathrm{dense}}$ and $m = X_n^{b-1} X_{n-1}$ we have $n' = 1$. Of course, since the message is contained in degree 0, only the Regev-instance for the constant monomial $m = 1$ is relevant and the above is not an issue. The other coefficients (apart from the X_i^d) are superfluous, not only for the ciphertexts but also for the secret key, since these coefficients are independent of the $g_{i,1}$ and the message. It follows that for $b = d$, the security of the PCN cryptosystem does not depend on d at all, but the efficiency degrades with d. Note that if the e_m are not independent, this might only help the attacker.

[5] The only difference is that for some of the parallel instances, the secret key has fewer coordinates.

[6] This amortized variant just runs parallel instances of Regev's, where the random coefficients a of the noisy linear combinations $\langle a, s \rangle + e$ are shared between instances.

Proposition 2. *Reduction from PCN to LWE for $d = 1$, b arbitrary*
Consider the case $d = 1$, b arbitrary. Then the PCN cryptosystem can be viewed as a structured version of Regev's scheme. There is a reduction from the (IND-CPA-)security of PCN to the (IND-CPA-)security of Regev's original scheme, as already noted in [1].

To see this, first observe that for $d = 1$, the secret Gröbner basis of the PCN cryptosystem is necessarily of the form $G = (X_1 - s_1, \ldots, X_n - s_n)$ for $s = (s_1, \ldots, s_n)$. We then have $f \bmod G = f(s)$, so SampleI(G, b) just gives us polynomials $f - f(s) \cdot 1$ for $f \in P_{\leq b}$ uniformly. For a monomial $m \neq 1$ of $P_{\leq b}$, let $\widetilde{s_m} := m(s) \in \mathbb{F}_q$. It follows that PCN-ciphertexts are of the form $\sum_{m \neq 1} a_m \cdot m - (\sum_{m \neq 1} a_m \widetilde{m_s}) \cdot 1$, where the $a_m \in \mathbb{F}_q$ are uniform and the sums run over the monomials m of $P_{\leq b}$ (except the constant one). This implies that PCN-instances are nothing but Regev-instances with a structured secret key $\widetilde{m_s}$.

Our reduction just has to remove that structure from the key. This can be done as in [13] by rerandomizing the secret:

Our reduction chooses $t_m \in \mathbb{F}_q$ uniformly for $m \neq 1$ monomial of $\mathbb{F}_q$. Then we bijectively transform any PCN-ciphertext $c = \sum_{m \neq 1} a_m m + b$ into a Regev-ciphertext $T_t(c) = (\boldsymbol{a}, b + \sum_{m \neq 1} t_m a_m)$. These ciphertexts are distributed as Regev-ciphertexts with uniform secret $\widetilde{\boldsymbol{s}} + \boldsymbol{t}$ with the same a_i and the same noise $e \leftarrow_\$ \mathcal{X}$.

4 Security Proofs

In this section, we clarify the relationships between the different security assumptions we recalled in Sect. 2.4 and the security of the PCN cryptosystem. We will first give counterexamples, showing that, under the LWE assumption, the GBN, IRN and IMN problems are not equivalent for general $d > 1$, refuting the claims from [1]. We will then give corrected proofs for $d = 1$.

In order to make the proofs for $d = 1$ work, we need to impose the following technical restriction on $\mathcal{X}$:

Definition 7. *We call a noise distribution $\mathcal{X}$ on $\mathcal{Q}_{\leq b}$ recognizable with noise, if for every $p' = \mathrm{poly}(\lambda)$ there exists a ppt algorithm D that, given oracle access to $\mathcal{X}_{a,p}$ with $p \leq p'$, outputs a with overwhelming probability for uniform $a \leftarrow_\$ \mathcal{Q}_{\leq b}$. Hereby, $\mathcal{X}_{a,p}$ is defined as a distribution that, with probability $\left(1 - \frac{1}{p(\lambda)}\right)$, outputs a uniform $x \leftarrow_\$ \mathcal{Q}_{\leq b}$, and otherwise (with probability $\frac{1}{p(\lambda)}$) outputs $x = e + a$ for $e \leftarrow_\$ \mathcal{X}$.*

We remark that a discrete Gaussian distribution with polynomial standard deviation is recognizable with noise (using as D the majority vote).

Theorem 2. IRN hard $\Leftrightarrow$ GBN hard, IRN hard $\Leftrightarrow$ IMN hard
Assume that the LWE assumption holds for some $q = \mathrm{poly}(n)$ and some noise distribution Φ on $\mathbb{F}_q$ that is recognizable with noise. Then there exists an instantiation for **Gen** *with $\mathcal{X}$ recognizable with noise (and, in particular, distinguishable*

from uniform), such that the IRN problem is easy, but both the GBN problem and the IMN problem are hard, contradicting the proofs from [1].

Proof. Consider the case $b = d = 2$ and let $q = \mathrm{poly}(n)$ and Φ be such that the LWE assumption holds for q and Φ. We consider **Gen** that outputs reduced Gröbner bases of the form $G = (X_1^2, X_2 + s_2 X_1, X_3 + s_3 X_1, \ldots, X_n + s_n X_1)$. Then $\mathcal{Q} = \mathcal{Q}_{\leq b}$ is generated by X_1 and 1 as a vector space. For the noise distribution $e_1 X_1 + e_2 \leftarrow_\$ \mathcal{X}$, we take $e_1 \leftarrow_\$ \Phi$ and e_2 uniform from $\mathbb{F}_q$.

By construction, the constant coefficient of all Gröbner base polynomials is 0, so for any $f \in I_{\leq b}$ we have $f \bmod G = f_{=0} + r(f) X_1$ for some $r(f) \in \mathbb{F}_q$. This already implies that we can guess the remainder by guessing $r(f)$ with noticeable probability $\frac{1}{q}$, compared to $|\mathcal{Q}_{\leq b}| = q^2$, giving a non-negligible advantage for the IRN problem.

Now let $f \leftarrow_\$ \mathrm{SampleI}_{\mathcal{X}}$ with $f = f^{(2)} + f^{(1)} + f^{(0)}$ be the homogenous parts of degree 2, 1 and 0. Since $P_{\geq 2} \subset I$ and the noise in degree 0 is uniform, we get that $f^{(2)}$ and $f^{(0)}$ are independently uniform and independent of $f^{(1)}$. Let us write $f = e + \sum_i t_i g_i$ with $g_i \in G, t_i \in P, e \in \mathcal{Q}$. Since $\deg g_i \geq 1$, $f^{(1)}$ only depends on e and the degree-0 part of the t_i.

It follows that $f^{(1)} = bX_1 + a_2 X_2 + \ldots a_n X_n$ with a_i uniform and $b = \sum a_i s_i + e_1$ with $e_1 \leftarrow_\$ \Phi$, i.e. $f^{(1)}$ is distributed as $\mathcal{L}_{s,\Phi}$. It follows that the IMN problem is equivalent to the $\mathrm{DLWE}_{n-1,q,\Phi}$-problem and the GBN problem is equivalent to the $\mathrm{LWE}_{n-1,q,\Phi}$-problem, both of which we assumed to be hard.

Remark 2. Separation of IRN and GBN.
There is also a separation between IRN and GBN, if we assume that the LWE-assumption holds for some q and some Gaussian noise. Namely, take $b = d = 2$ and let **Gen** output Gröbner bases of the form $X_1^2 - s_1, \ldots X_n^2 - s_n$ with $s_i \in \mathbb{F}_q$ independent and uniformly. Note that there are no linear terms here. As noise distribution choose Gaussian noise, concentrated in degree 0. Then the GBN problem is is hard if the LWE assumption holds (cf. Prop. 1). However, IMN is easy, because noisy samples from the ideal have no linear terms.

Note that we assumed Φ to be recognizable with noise to satisfy the requirements from [1] and all requirements from Thm 3, apart from $d = 1$, below. Without that restriction on the noise, we may take Φ to be uniform and get an information-theoretical variant of Thm. 2 without the need for an LWE assumption.

For $d = 1$, the statements from [1] actually hold. The reason why we can make the proof work only in that case is that we need an amplification step, for which our rerandomization strategy only works for $d = 1$.

Theorem 3. IRN hard $\Leftrightarrow$ GBN hard $\Leftrightarrow$ IMN hard for $d = 1, q = \mathrm{poly}(n)$ and $\mathcal{X}$ recognizable with noise
For any **Gen**, $\mathcal{X}, b \leq d$, *we have:*

1. *If the IMN problem is hard, the PCN cryptosystem is IND-CPA-secure.*
2. *If $d = 1$, $\mathcal{X}$ recognizable with noise, then the IRN problem is hard iff the GBN problem is hard.*

3. *If $q = \mathrm{poly}(n), d = 1, \mathcal{X}$ distinguishable from uniform, then then IRN problem is hard iff the IMN problem is hard.*

Proof. Statement 1 is proven in [1]. Statement 2 and 3 are proven in the appendix of the full version.

5 Attack on Low-Dimensional Noise

In this section, we present our main contribution. We will present a polynomial time CPA-attack against the PCN cryptosystem that recovers the secret key, if $b > d > 1$, using that the noise is contained in degree 0. Note that all concrete parameter choices of [1] use $d = 1$, but this attack still violates the explicit security assumption, which is stated for general d.

Throughout this section we assume that $d > 1$ and that $\mathcal{X}$ is supported in degree $\leq k$. Furthermore, we assume for simplicity that we are using $\mathbf{Gen} = \mathrm{GBGen}_{\mathrm{dense}}$. Using the notation from Alg. 1 above, let us write the secret key as $G = (g_1, \ldots, g_n)$ with $g_i = X_i^d + \sum_m g_{i,m} \cdot m$. Our attack will derive linear equations for the $g_{i,m}$.

The intuition behind the attack is the following:

Since the support of $\mathcal{X}$ is contained in $\mathcal{Q}_{\leq k}$, all ciphertexts are contained in a vector sub-space $N := I_b \oplus \mathcal{Q}_{\leq k} \subsetneqq P_{\leq b}$. We can recover this vector space N via a CPA-oracle (In the public-key variant, N is directly given by the public key). Note that the dimension of N is known, namely, it is $\dim N = \dim P_{\leq b} - \dim \mathcal{Q}_{\leq b} + \dim \mathcal{Q}_{\leq k} = O(n^b)$.

Of course, since $g_i \in I_{\leq b} \subset N$, the secret Gröbner base polynomials must also lie in this subspace. If the inclusion is proper, this directly translates into linear equations for the $g_{i,m}$. Unfortunately (for the attacker), these equations do not yet determine the g_i: We may add any error term $h \in \mathcal{Q}_{\leq k}$ to g_i and we still have $g_i + h \in N$.

To overcome this, we make use of the fact that I is an ideal, so $t \cdot g_i \in I_{\leq b} \subset N$ for any polynomial t with $\deg t \leq b - d$. Roughly speaking, this effectively also multiplies the error term by t and we will use this to move the error out of $\mathcal{Q}_{\leq k}$. In order to move the error completely out of $\mathcal{Q}_{\leq k}$, we need to multiply by polynomials t of degree $> k$, so we expect our attack to work whenever $b - d > k$. In particular, for $k = 0$, this strategy will recover the secret key for $b > d$.

Note that we will also cover the case $d > k$:

Remember that for $d > 1$, $\mathcal{Q}$ has exponential (in n) dimension and contains polynomials of degree $> d$ for n large. In the case $d > k$, we will not get any useful information from $g_i \in N$. But for $k < b$ we still have $\mathcal{Q}_{\leq k} \subsetneqq \mathcal{Q}_{\leq b} \subsetneqq \mathcal{Q}$ for n large. This means we will get some useful equations from $t \cdot g_i \in N$ for polynomials t with $b - d \geq \deg t \geq d - k$

We now present the actual algorithm and then we will give a rigorous analysis.

Algorithm 5. ppt attack against PCN cryptosystem with low-degree noise

Input: $1^n, k, b, d$, access to a CPA oracle
Output: Secret key $g_{i,m}$

1: $N :=$ a vector space basis of $\mathcal{Q}_{\leq k}$
2: **repeat**
3: Create $f \in I_{\leq b} \oplus \mathcal{Q}_{\leq k}$ an encryption of 0
4: $N := N \cup \{f\}$
5: **until** $\dim(\operatorname{Span} N) = \dim(I_{\leq b} \oplus \mathcal{Q}_{\leq k})$
▷ We now have $\operatorname{Span}(N) = I_{\leq b} \oplus \mathcal{Q}_{\leq k}$
6: Write N as a matrix and perform Gaussian elimination to obtain $\operatorname{Span}(N) = \ker A$ for linear $A : P_{\leq b} \to \mathbb{F}_q^{\dim \mathcal{Q}_{\leq b} - \dim \mathcal{Q}_{\leq k}}$.
7: **for** $i = 1$ to n **do**
8: Let $E_i := \emptyset$ be the set of equations for the $g_{i,m}$.
9: **for all** monomials $t \in P_{\leq b-d}$ **do**
10: Add the inhomogenous linear equations $A(t \cdot g_i) = 0$ in the variables $g_{i,m}$ to E_i.
11: Solve the system of equations E_i
12: **return** A solution $\overline{g_{i,m}}$ for each of the E_i

Theorem 4. Algorithm 5 is Correct and Runs in Polynomial Time with Overwhelming Probability

With overwhelming probability, algorithm 5 runs in polynomial time $O(n^{2b+d+1})$.

If $n > k, d > 1$ *and* $b - d > k$, *the algorithm outputs the secret key.*

In particular, for $k = 0$, *that is, for noise concentrated in degree 0, the algorithm gives an efficient key-recovery attack for* $b > d > 1$.

More precisely, we claim that, if $n > k$, *we have* $\overline{g_{i,m}} = g_{i,m}$ *whenever* $\deg m > k - (b - d)$.

For any other m *with* $\deg m \leq k - (b - d)$, $\overline{g_{i,m}}$ *may be chosen arbitrarily by the algorithm (the solution of the* E_i *is not unique if such monomials exist).*

Proof. Let us start with the running time:
In line 2 to 5, we use the CPA-oracle to obtain $f \in I_{\leq b} \oplus \mathcal{Q}_{\leq k}$ (Note here that if the message is embedded only within $\mathcal{Q}_{\leq k}$ as well, *any* ciphertext will do).

Since the $I_{\leq b}$-component of f is uniform, after $O(\dim I_{\leq b}) = O(n^b)$ steps, we will eventually obtain all of $I_{\leq b} \oplus \mathcal{Q}_{\leq k}$ with overwhelming probability.

After that, the running time of the algorithm is dominated by solving the E_i. Each E_i consists of $\dim P_{\leq b-d} \cdot (\dim \mathcal{Q}_{\leq b} - \dim \mathcal{Q}_{\leq k}) = O(n^{2b-d})$ equations in at most $\dim P_{\leq d} = O(n^d)$ unknowns. Since $2b - d > d$, this gives a running time of $O(n^{2b-d}) \cdot O(n^d) \cdot O(n^d) = O(n^{2b+d})$ for solving each E_i, hence a total running time of $O(n^{2b+d+1})$ to solve all the E_i (cf. Rmk 3 below).

We now turn to the correctness statement:

By construction, $\operatorname{Span}(N) = I_{\leq b} \oplus \mathcal{Q}_{\leq k}$, starting from line 6.

Since $g_i \in I_{\leq b} \subset I_{\leq b} \oplus \mathcal{Q}_{\leq k}$, we have $A(g_i) = 0$. Making use of the fact that I is an ideal, we also have $t \cdot g_i \in I_{\leq b}$ and hence $A(t \cdot g_i) = 0$ for $\deg t \leq b - d$.

It follows that the equations we derive for the $g_{i,m}$ are correct, that is, the $g_{i,m}$ satisfy the equations E_i.

Note that in line 10, we rewrite the linear equation $A(tg_i)$ as an equation in the $g_{i,m}$. Implicitly, we add the equations $g_{i,m} = 0$ for $m > X_i^d$, $g_{i,X_j^d} = 0$ for $i \neq j$ and $g_{i,X_i^d} = 1$ at this point. Since we set the coefficient of X_i^d to be 1 in that last equation, the resulting system of equations E_i is a system of inhomogeneous linear equations.

Now, the E_i might have more than one solution, apart from the secret key $g_{i,m}$.

To show that the coefficients for monomials m with $\deg m \leq k - (b - d)$ are undetermined, we first observe that $P_{\leq k} \bmod G = \mathcal{Q}_{\leq k}$, so $I_{\leq b} \oplus \mathcal{Q}_{\leq k} = I_{\leq b} + P_{\leq k}$. Consequently, by Lemma 1 $I_{\leq b} \oplus \mathcal{Q}_{\leq k}$ are exactly all elements of the form $f = e + \sum t_i g_i$ with arbitrary $e \in P_{\leq k}$, $t_i \in P_{\leq (b-d)}$. The coefficients of the g_i of degree $\leq k - (b - d)$ then only affect the coefficients of f of degree $\leq k$, which are uniform due to e. So $I_{\leq b} + P_{\leq k}$ does not depend on the coefficients of g_i of degree $\leq k - (b - d)$, which implies that these coefficients span a subspace of the kernel of the E_i.

To show that for $n \geq k + 1, \deg m > k - (b - d)$, we have $\overline{g_{i,m}} = g_{i,m}$, let $\overline{g_i}, \overline{g_i}'$ be 2 solutions for E_i and $h = \overline{g_i} - \overline{g_i}'$. We need to show $\deg h \leq k - (b - d)$ (which means $h = 0$ if the right-hand side is negative).

By construction of the E_i, we know that $A(t \cdot h) = 0$, or equivalently $t \cdot h \in I_{\leq b} \oplus \mathcal{Q}_{\leq k}$, for all $t \in P_{\leq b-d}$. The other equations coming from the restrictions on the set of monomials that can appear in the $\overline{g_i}, \overline{g_i}'$ imply that h can only contain coefficients for the set of monomials $\{m \mid m < X_i^d, m \neq X_j^d \text{ for any } j\}$. This implies that $h \in \mathcal{Q}_{\leq d}$, in particular, $\deg h \leq d$.

We will show that $\deg h \leq k - \alpha$ for $0 \leq \alpha \leq b - d$, using induction on α. For $\alpha = b - d$, the claim then follows.

For the base case $\alpha = 0$, we already observed that $h \in \mathcal{Q}_{\leq d}$. Setting $t = 1$ in $A(th) = 0$ yields $h \in I_{\leq b} \oplus \mathcal{Q}_{\leq k}$. Together, these give $h \in \mathcal{Q}_{\leq k} \cap \mathcal{Q}_{\leq d}$, so $\deg h \leq k$ as desired.

For the inductive step, assume $\deg h \leq k - \alpha$ for $\alpha < (b - d)$. Assume w.l.o.g. that $h \neq 0$, since otherwise we are done. Let $H = \mathrm{LT}(h)$ be the leading term. Since the monomial order is degree-compatible, $\deg H = \deg h$. We need to show that $\deg H < k - \alpha$.

For this, choose a monomial t of degree $\deg t = \alpha + 1 \leq b - d$ such that $t \cdot H \in \mathcal{Q} = \mathrm{Span}\{X_1^{v_1} \cdots X_n^{v_n} \mid v_i < d \text{ for all } i\}$. This can be accomplished for $d > 1$ and $n \geq k + 1$ by choosing $t = X_{i_1} \cdots X_{i_{\alpha+1}}$ a product of $\alpha + 1$ pairwise different variables, disjoint from those of H.[7] By the properties of a monomial order, $\mathrm{LT}(t \cdot h) = t \cdot H$. Since $t \cdot H \in \mathcal{Q}$, this is not reduced modulo G, so we have $\mathrm{LT}((t \cdot h) \bmod G) = t \cdot H$. Since $A(t \cdot h) = 0$, we have $t \cdot h \in I_{\leq b} \oplus \mathcal{Q}_{\leq k}$. This implies $(t \cdot h) \bmod G \in \mathcal{Q}_{\leq k}$, in particular $(t \cdot h) \bmod G$ has degree at most k. It follows that H has degree at most $k - \deg t = k - \alpha - 1$. This finally proves the theorem.

Remark 3. Algorithm 5 was optimized for simplicity of analysis. We can get a better running time by using the highly structured nature of the equations on

[7] Note that if $k < d$, any t of degree $\alpha + 1$ will do without the restriction on n.

the E_i. In particular, we don't need all $t \in P_{\leq b-d}$, as the proof above shows and we also don't need to solve the E_i separately for $1 \leq i \leq n$.

Also, we would like to remark that Algorithm 5 also gives an attack to the underlying GBN, IRN and IMN problems; in particular the existence of this attack is not related to the flaws in security proof of [1] we pointed out in section 4.

6 Conclusion and Open Problems

We have seen that for $d > 1$, the security reductions from [1] will no longer work and there arise problems in choosing a noise distribution. Concentrating the noise in low degree makes the scheme insecure unless $b = d$, so the obvious way to go is to spread the noise over the full quotient. We remark that it might be possible to retain the homomorphic properties by using a different strategy to generate the Gröbner basis, allowing multiplicative homomorphic properties in Ring-LWE [10] style. We leave this as an open problem.

Acknowledgements. We would like to thank Martin Albrecht for valuable discussions and helpful comments.

References

1. Albrecht, M.R., Farshim, P., Faugère, J.-C., Perret, L.: Polly Cracker, Revisited. In: Lee, D.H. (ed.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 179–196. Springer, Heidelberg (2011); Cryptology ePrint Archive, Report 2011/289, http://eprint.iacr.org/
2. Barkee, B., Can, D.C., Ecks, J., Moriarty, T., Ree, R.F.: Why you cannot even hope to use Gröbner bases in public key cryptography: An open letter to a scientist who failed and a challenge to those who have not yet failed. J. of Symbolic Computations 18(6), 497–501 (1994)
3. Becker, T., Weispfenning, V.: Gröbner bases: a computational approach to commutative algebra. Graduate Texts in Mathematics. Springer (1993)
4. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. To appear in FOCS 2011 (2011); Cryptology ePrint Archive, Report 2011/344 (2011), http://eprint.iacr.org/
5. Buchberger, B.: Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal. PhD thesis, Universität Innsbruck (1965)
6. Cox, D., Little, J., O'Shea, D.: Ideals, Varieties, and Algorithms, 3rd edn. Springer (2005)
7. dit Vehel, F.L., Marinari, M.G., Perret, L., Traverso, C.: A survey on Polly Cracker systems. In: Gröbner Bases. Coding and Cryptography, pp. 285–305. Springer (2009)
8. Fellows, M., Koblitz, N.: Combinatorial cryptosystems galore! In: Finite Fields: Theory, Applications, and Algorithms. Contemporary Mathematics, vol. 168, pp. 51–61. AMS (1994)

9. Gentry, C.: A fully homomorphic encryption scheme. PhD thesis, Stanford University (2009)
10. Lyubashevsky, V., Peikert, C., Regev, O.: On Ideal Lattices and Learning with Errors over Rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (2010)
11. Peikert, C., Vaikuntanathan, V., Waters, B.: A Framework for Efficient and Composable Oblivious Transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008)
12. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing (STOC 2005), pp. 84–93. ACM (2005)
13. Regev, O.: The learning with errors problem (invited survey). In: IEEE Conference on Computational Complexity, pp. 191–204. IEEE Computer Society Press (2010)
14. Rothblum, R.: Homomorphic Encryption: From Private-Key to Public-Key. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 219–234. Springer, Heidelberg (2011)

Ring-LWE in Polynomial Rings

Léo Ducas and Alain Durmus*

ENS, Dépt. Informatique, 45 rue d'Ulm, 75005 Paris, France

Abstract. The Ring-LWE problem, introduced by Lyubashevsky, Peikert, and Regev (Eurocrypt 2010), has been steadily finding many uses in numerous cryptographic applications. Still, the Ring-LWE problem defined in [LPR10] involves the fractional ideal $R^\vee$, the dual of the ring R, which is the source of many theoretical and implementation technicalities. Until now, getting rid of $R^\vee$, required some relatively complex transformation that substantially increase the magnitude of the error polynomial and the practical complexity to sample it. It is only for rings $R = \mathbb{Z}[X]/(X^n + 1)$ where n a power of 2, that this transformation is simple and benign.

In this work we show that by applying a different, and much simpler transformation, one can transfer the results from [LPR10] into an "easy-to-use" Ring-LWE setting (*i.e.* without the dual ring $R^\vee$), with only a very slight increase in the magnitude of the noise coefficients. Additionally, we show that creating the correct noise distribution can also be simplified by generating a Gaussian distribution over a particular extension ring of R, and then performing a reduction modulo $f(X)$. In essence, our results show that one does not need to resort to using any algebraic structure that is more complicated than polynomial rings in order to fully utilize the hardness of the Ring-LWE problem as a building block for cryptographic applications.

1 Introduction

Since its recent introduction, the *Ring-LWE* problem [LPR10] has already been used as a building block for numerous cryptographic applications. In addition to its original functionality as the basis of efficient lattice-based cryptosystems [LPR10], it has since been used as a hardness assumption in the constructions of efficient signature schemes [MP11, Lyu11], fully-homomorphic encryption schemes [BV11b, BV11a, BGV11, GHS11], pseudo-random functions [BPR11], protocols for doing secure multi-party computation [DPSZ11, LATV11], and also gives an explanation for the hardness of the NTRU cryptosystem [SS11].

A very natural way in which one would like to be able to define the (decisional) Ring-LWE problem is as follows: for a polynomial ring $R_q = \mathbb{Z}_q[X]/(f(X))$ and a random polynomial $w \in R_q$, it is computationally hard to distinguish the uniform distribution over $R_q \times R_q$ from ordered pairs of the form $(a_i, a_i w + e_i)$, where a_i are uniformly distributed in R_q and e_i are polynomials in R whose coefficients are independently distributed Gaussians. Unfortunately, the results from [LPR10]

* This work was partially supported by the European Research Council.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 34–51, 2012.

do not directly imply that the above problem is hard based on the worst-case hardness of lattice problems, except in the one case when $f(X) = X^n + 1$ for n a power of 2, and thus most papers that use the Ring-LWE problem only use this one specific ring. The reason for this limitation is that the problem statement in [LPR10] requires w to be in the *dual* ring of R (which is a fractional ideal) and for the distribution of the noise to be a spherical Gaussian in the *embedding* representation of R. And it is only in the case that $R = \mathbb{Z}[X]/(X^n + 1)$ that the dual ring is simply a scaling of R (thus, one can simply multiply by the scaling and end up in R) and the embedding is just a rigid rotation and a scaling (thus the spherical Gaussian distribution is not affected by the transformation). For *all* other cyclotomic polynomials, while it is possible to transform the problem that was proved hard in [LPR10] to the one described above, the transformation between the polynomial and embedding representations involves multiplication by a skewed matrix, and the dual of R is a (possibly very) skewed fractional ideal of R. Therefore there is no obvious way to generate the noise directly in the ring R, nor work entirely in the ring R without utilizing a transformation that can substantially increase the magnitude of the error polynomials.

A natural question to ask at this point is whether there is ever a reason to use a ring other than $R_q = \mathbb{Z}_q[X]/(X^n + 1)$. While it's true that this ring has some very nice features, and we believe that it should be used whenever possible, there are situations where an alternative may be preferable. Since $X^n + 1$ is only irreducible when n is a power of 2, these polynomials are scarce. Thus it is conceivable that to achieve a certain security level, it may be advantageous to try to find a polynomial of some particular degree rather than round up to the next power of 2. A different, and a probably even stronger reason to use a different ring, is that other cyclotomic polynomials may have a more desirable structure for the task at hand. An example of this is the recent result of Gentry, Halevi, and Smart [GHS11] who show that there are particular cyclotomic polynomials that allow for much faster (at least asymptotically) instantiations of fully-homomorphic encryption. Their hardness assumption is that the Ring-LWE problem, instantiated with polynomial rings as in our description above, is a difficult problem. Using the result of our current paper, it can actually be shown that their scheme has tight connections to worst-case lattice problems (modulo a small change in the way the errors are generated, but this can be easily remedied).

1.1 Our Results

Our main result (Theorem. 2) essentially shows that for any cyclotomic polynomial $\Phi_m(X)$, one can work entirely in the ring $\mathbb{Z}[X]/(\Phi_m)$, and generate the noise distribution without resorting to complex embeddings.

Our analysis (Sect. 5) shows that for primes m (and even wider class) our simplification comes at almost no cost in term of algorithmic simplicity, tightness and efficiency compared to the scarce class of m that are powers of 2 as used for practical application in [LPR10]; thus increasing the density of usable $m < M$ from $\mathcal{O}(\log(M)/M)$ to $\mathcal{O}(1/log(M))$.

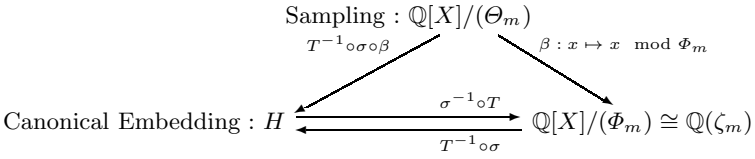


Fig. 1. Mappings Between Different Representations (see Sect. f2 or formal definitions. The polynomial Θ_m is defined to be $X^m - 1$ if m is odd, and $X^{m/2} + 1$ when m is even.

Our main result is a consequence of two theorems with surprisingly elementary proofs. The first theorem (see Section 4) states that every cyclotomic ring of integers $R = \mathbb{Z}[X]/(\Phi_m) \cong \mathbb{Z}[\zeta_m]$ contains $mR^\vee$, where $R^\vee$ is its dual (if m is even, it actually contains $\frac{m}{2}R^\vee$). What this means is that one can scale everything that is in $R^\vee$ by a factor of m (or $m/2$) and end up in the ring R. Similarly, if something were uniform, either statistically or computationally, modulo $R^\vee$, then m times it will be uniform modulo $mR^\vee$ and thus uniform modulo R, since $mR^\vee$ is an additive subgroup of R. This transformation is not completely tight (except in the case that $\Phi_m(X) = X^{m/2} + 1$) because we end up with something that is uniform modulo a subgroup of R, whereas we only use the randomness modulo R. This loss of tightness, however, is very small, resulting in the noise being at most $\sqrt{m/\phi(m)}$ "larger than necessary" (see the discussion after Theorem 2).

Our second theorem (see Section 5) deals with the noise generation. In the Ring-LWE definition of [LPR10], the noise needs to be a spherical Gaussian in the *canonical embedding* representation of the ring $\mathbb{Q}[X]/(\Phi_m)$ (see Figure 1), and to convert it to the polynomial representation, one needs to perform transformation $\sigma^{-1}\circ T$, where σ^{-1} is the multiplication by the inverse of a complex Vandermonde matrix (and T is a multiplication by a very simple matrix). Ideally, one would like to avoid working with the complex numbers and generate the noise by simply drawing it from the ring $\mathbb{Q}[X]/(\Phi_m)$; but unfortunately this method does not lead to the correct distribution in the embedding representation. What we show is that an almost equally simple way of generating the noise does lead to the correct distribution. We consider the ring $\mathbb{Q}[X]/(\Theta_m)$, where $\Theta_m(X) = X^m - 1$ if m is odd, and $X^{m/2} + 1$ if m is even (notice that Φ_m is a factor of Θ_m). We then show that the transformation denoted by $T^{-1} \circ \gamma$ from $\mathbb{Q}[X]/(\Theta_m)$ to the embedding representation actually preserves the spherical Gaussian distribution! This means that one can sample in $\mathbb{Q}[X]/(\Theta_m)$ by picking each coefficient independently from a continuous Gaussian distribution (rounded to $\mathbb{Q}$, see details in 2), and it will be the correct distribution required by [LPR10]. Then to move the noise from $\mathbb{Q}[X]/(\Theta_m)$ to $\mathbb{Q}[X]/(\Phi_m)$, one simply performs the transformation β, which is just a reduction modulo Φ_m.

In addition to making our noise generation much simpler to implement, the reduction modulo Φ_m is also simpler to analyze than $\sigma^{-1}\circ T$. This allows us to make several improvements in constructions that use rings other than $\mathbb{Z}[X]/(X^n + 1)$

(for rings $\mathbb{Z}[X]/(X^n+1)$, the mapping β is just the identity, and so there is nothing to analyze). As realized in previous works that used ideal lattices (e.g. [LM06, Gen10, GHS11]), multiplication in polynomial rings increases the size of the coefficients by a factor that depends on the size of the coefficients in the multiplicands, and also on the ring itself, and the ring in which the coefficients grow the least is $\mathbb{Z}_q[X]/(X^n+1)$. As a consequence, if one were to, for example, implement the encryption scheme from [LPR10] in the ring $\mathbb{Z}[X]/(\Phi_p)$ for some prime p, one would observe that the noise grows by a factor of approximately $\sqrt{2}$ larger than in the ring $\mathbb{Z}[X]/(X^n+1)$. We show that by analyzing the noise in the ring $\mathbb{Q}[X]/(\Phi_p)$, one can actually remove some of the noise that is introduced by the reduction modulo Φ_m; it seems that our strategy makes the coefficients grow only $(1+o(1))$ times as much (see Section 6).

2 Preliminaries

Cyclotomic Ring. Let ζ_m be a primitive m^{th} root of unity and the cyclotomic polynomial $\Phi_m(X) \in \mathbb{Q}[X]$ be its minimal monic polynomial. Thus m is the smallest integer for which $\zeta_m^m = 1$ and Φ_m is the rational polynomial with the smallest degree of which ζ_m is a root. It is known that $\Phi_m \in \mathbb{Z}[X]$ and the other roots of Φ_m (the conjugates of ζ_m) are the elements of the set $\{\zeta_m^k | k \in \mathbb{Z}_m^*\}$. Thus, Φ_m has degree $\phi(m)$, the totient of m. So, the number field $\mathbb{Q}(\zeta_m)$, which we will call the m^{th} cyclotomic field, has degree $\phi(m)$ and its power basis is $\left\{1, \zeta_m, \cdots, \zeta_m^{\phi(m)-1}\right\}$.

Extension of the Cyclotomic Ring. For a given each integer m we define the polynomial $\Theta_m(X)$ as X^m-1 if m is odd, and $X^{m/2}+1$ when m is even. It gives a natural ring extension $\mathbb{Z}[X]/(\Theta_m)$ of the cyclotomic ring $\mathbb{Z}[X]/(\Phi_m)$: as Φ_m is a factor of Θ_m, the reduction modulo Φ_m, noted β is a ring morphism (it preserve both sum and product). The power basis of $\mathbb{Z}[X]/(\Theta_m)$ is $\left\{1, \zeta_m, \cdots, \zeta_m^{m-1}\right\}$ when m is odd and $\left\{1, \zeta_m, \cdots, \zeta_m^{\frac{m}{2}-1}\right\}$ when m is even.

Ring of Integers. The ring of integers of $\mathbb{Q}(\zeta_m)$ is $\mathbb{Z}[\zeta_m] \cong \mathbb{Z}[X]/(\Phi_m)$. According to the following theorem from [Con09, Theorem 3.7], the dual (or co-different ideal) of $\mathbb{Z}[\zeta_m]$, denoted by $\mathbb{Z}[\zeta_m]^\vee$, is the fractional ideal $\frac{1}{\Phi'_m(\zeta_m)}\mathbb{Z}[\zeta_m]$, where Φ'_m is the derivative of Φ_m. While the dual has many nice properties and is extensively used in the proof of the hardness of Ring-LWE in [LPR10], in the current paper we only need its definition.

Embeddings of Cyclotomic Fields. The field $\mathbb{Q}(\zeta_m) \simeq \mathbb{Q}[X]/(\Phi_m)$ has exactly $\phi(m)$ embeddings $(\sigma_k)_{k\in\mathbb{Z}_m^*}$, defined by $\sigma_k : x \mapsto x(\zeta_m^k)$, for $k \in \mathbb{Z}_m^*$. The canonical embedding $\sigma : \mathbb{Q}(\zeta_m) \to \mathbb{C}^{\phi(m)}$ is defined as the direct sum of all the embeddings : $\sigma(x) = \bigoplus_{k\in\mathbb{Z}_m^*} \sigma_k(x)$. Note that that for each $k \in \mathbb{Z}_m^*$ and any

$x \in \mathbb{Q}(\zeta_m)$, we have $\sigma_{-k}(x) = \overline{\sigma_k(x)}$. Thus for a proper indexation of $\mathbb{Z}_m^*$ the image H of σ is the $\mathbb{Q}$ vector space generated by the columns of $\sqrt{2} \cdot T$ where :

$$T = \frac{1}{\sqrt{2}} \begin{pmatrix} \mathrm{Id}_{\phi(m)/2} & \mathbf{i}\,\mathrm{Id}_{\phi(m)/2} \\ \mathrm{Id}_{\phi(m)/2} & -\mathbf{i}\,\mathrm{Id}_{\phi(m)/2} \end{pmatrix} \quad \text{with } \mathbf{i} = \sqrt{-1}$$

In other words, for any element $x \in \mathbb{Q}(\zeta_m)$, there exists a vector $v \in \mathbb{Q}^{\phi(m)}$ such that $\sigma(x) = \sqrt{2}Tv$, and vice versa. For the rest of the paper, we will consider the column vectors of T as the canonical basis for the embedding space H.

Gaussian Distributions. By ψ_s we denote the Gaussian distribution with mean 0 and standard deviation s over $\mathbb{R}$; and by ψ_s^d the spherical Gaussian distribution over $\mathbb{R}^d$ of the vector $(v_1, \ldots, v_d)$ where each coordinate is drawn independently from ψ_s.

For our purpose, one would like the Gaussian distributions to be defined over $\mathbb{Q}$ rather than $\mathbb{R}$, so that an element drawn from $\psi_s^{\phi(m)}$ may be seen as element of the field $\mathbb{Q}(\zeta_m)$. The theoretical solution to that issues is to work with the tensor product $\mathbb{Q}(\zeta_m) \otimes_{\mathbb{Q}} \mathbb{R}$ as done in [LPR10].

However, in practice elements needs to be represented finitely, typically using floating points numbers of a fixed mantissa. For simplicity we choose this solution: we consider that output of Gaussian distribution ψ_s^d are rounded off to rational numbers using a fine enough grid so that all our results go through except with a negligibly small probability.

3 The Main Result

In this section we give the main result of this paper. We describe a distribution over $R_q \times R_q$, where $R_q = \mathbb{Z}_q[X]/(\Phi_m)$ which is computationally indistinguishable from the uniform distribution over $R_q \times R_q$ based on the worst-case hardness of the approximate shortest vector problem in ideal lattices. The proof of our theorem will use results that we later prove in Sections 4 and 5 that will aid us in transforming the hard Ring-LWE problem defined in [LPR10] into one in which all operations are performed in polynomial rings.

Theorem 1 ([LPR10]). *Let m be integer, and q be a prime congruent to 1 modulo m. Let denote K be the number field $\mathbb{Q}(\zeta_m)$, $R = \mathbb{Z}[\zeta_m]$ be its ring of integers, $R^\vee$ be the fractional ideal $\frac{1}{\Phi'_m(\zeta_m)}\mathbb{Z}[\zeta_m]$, q be a prime congruent to $1 (\mathrm{mod}\, m)$. Also, let k be any positive integer and $\alpha \in (0,1)$ be a real number such that $\alpha q > \omega\left(\sqrt{\log m}\right)$. If there exists an algorithm that can solve the decisional Ring-LWE problem, that is distinguish (with some advantage $1/poly(m)$) between k uniformly random samples drawn from $R/qR \times K/R^\vee$ and k samples $(a_i, \frac{a_i w}{q} + e_i) \in R/qR \times K/R^\vee$ where a_i are chosen uniformly at random from R/qR, w is chosen uniformly at random from $R^\vee/qR^\vee$, and the e_i are sampled in the embedding space H from the distribution $\psi_s^{\phi(m)}$ for $s = \alpha \cdot \left(\frac{\phi(m)k}{\log(\phi(m)k)}\right)^{1/4}$,*

then there exists a quantum algorithm that runs in time $O(q \cdot poly(m))$ that solves the approximate Shortest Vector Problem to within a factor $\tilde{O}(\sqrt{m}/\alpha)$ in any ideal of the ring $\mathbb{Z}[\zeta_m]$.

Before stating our main theorem, we believe that it would be helpful to first understand why everything turns out to be so simple and convenient when working with the ring of integers $\mathbb{Z}[\zeta_m]$ when m is a power of 2 (and not so convenient otherwise). If m is a power of 2, then $\Phi_m = x^n + 1$, where $n = m/2$, and therefore $\Phi' = nX^{n-1}$, and so $\Phi'(\zeta_m) = n\zeta_m^{n-1}$. The last equation implies that $n\zeta_m^{n-1}R^\vee = R$ (and since $\zeta_m^j R = R$ for any integer j, we have $nR^\vee = R$), which gives us a very simple way to remove the ring $R^\vee$ and work entirely in the ring R. When given a sample $(a_i, \frac{a_i w}{q} + e_i) \in R/qR \times K/R^\vee$, we can simply multiply the second element of the ordered pair by n and get $(a_i, \frac{a_i wn}{q} + e_i n) \in R/qR \times K/nR^\vee$. Now we observe that since the e_i were chosen from the distribution $\psi_s^{\phi(m)}$, the ne_i are distributed according to $\psi_{ns}^{\phi(m)}$. And since w was chosen uniformly at random from $R^\vee/qR^\vee$, we have that nw is uniformly random in R/qR. Thus the problem of distinguishing uniformly random samples in $R/qR \times K/R$ from samples $(a_i, \frac{a_i w'}{q} + e'_i) \in R/qR \times K/R$ where a_i and w' are drawn uniformly from $R/(q)$ and the e'_i are drawn according to the distribution $\psi_{ns}^{\phi(m)}$ is exactly equivalent to the problem from Theorem 1. We now turn to how one would generate the errors e'_i directly in the ζ_m power basis, without first generating them in the embedding space and then doing the transformation. The main observation here is that the linear transformation $\sigma^{-1} \circ T$ (see Figure 1) from the embedding space H to the power basis representation turns out to be a multiplication by a scaled orthogonal matrix. Therefore, the spherical Gaussian distribution in H remains a spherical Gaussian distribution in the power basis representation, and can therefore be sampled directly in the latter domain.

On the other hand, if ζ_m is a primitive root of unity for any other m except a power of 2, then neither of the above-described conditions hold. It is still possible to multiply elements in $R^\vee$ by $\Phi'(\zeta_m)$ in order to take them into R, but this transformation does not result in "nice" distributions in the power basis of R. It is known that there exist cyclotomic polynomials Φ_m whose coefficients are of the order of $m^{\log m}$, and thus Φ'_m also has coefficients of that magnitude. Therefore when multiplying an element by $\Phi'(\zeta_m)$, the coefficients of the product in the power basis will also very likely have such large coefficients, and thus the noise will increase by a super-polynomial factor. And even for simple cyclotomic polynomials such as Φ_p for some prime p, its derivative will have $\Omega(p)$ coefficients of size $\Omega(p)$, and so the multiplication by Φ'_p could increase the coefficients by a factor of p^2. Additionally, if ζ_m is a primitive root of unity and m is not a power of 2, then the mapping $\sigma^{-1} \circ T$ from the embedding space H to the power basis representation is no longer an orthogonal linear map, and thus the spherical Gaussian distribution is no longer preserved.

Theorem 2 (Main Theorem). *Let m be an integer, and let R_q be the ring $\mathbb{Z}_q[X]/(\Phi_m)$ where q is a prime congruent to 1 modulo m. Also, let k be any*

positive integer, $\alpha \in (0,1)$ be a real number such that $\alpha q > \omega\left(\sqrt{\log m}\right)$, and define m' to be equal to m if m is odd and $m/2$ if m is even. If there is an algorithm that can solve the Ring-LWE problem, that is distinguish (with some advantage $1/poly(m)$) between k uniformly random samples drawn from $R_q \times R_q$ and k samples $(a_i, a_i w + e_i) \in R_q \times R_q$, where a_i and w are chosen uniformly at random from R_q and $e_i = \lceil e'_i \bmod \Phi_m \rfloor$ with $e'_i \in \mathbb{Q}[X]/(\Theta_m)$ is distributed as $\psi_s^{m'}$ for $s = \sqrt{m'}\alpha q \left(\frac{\phi(m)k}{\log(\phi(m)k)}\right)^{1/4}$; then there exists a quantum algorithm that runs in time $O(q \cdot poly(m))$ that solves the approximate Shortest Vector Problem to within a factor $\tilde{O}(\sqrt{m}/\alpha)$ in any ideal of the ring $\mathbb{Z}[\zeta_m]$.

Before we give the proof of this theorem (which uses results from Sections 4 and 5), we would like to draw the reader's attention to several things.

First, we emphasize that the error distribution is generated by sampling a polynomial $g_0 + g_1 X + \ldots + g_{d-1} X^{m'-1} \in \mathbb{Q}[X]/(\Theta_m)$ where g_i simply are independants Gaussian variables, then reducing modulo Φ_m, and only then rounding each coefficient to the nearest integer. While it would have been slightly more convenient to be able to round and then do a reduction modulo Φ_m, the two distributions are not equivalent.

Secondly, we point out that by using a Lemma similar to [ACPS09, Lemma 2], it can be shown that instead of choosing the secret w uniformly from R_q, it can be drawn from the same distribution as the error vectors e_i. The only consequence of this is that the value of k in the theorem increases by one.

A third comment is that just as in Theorem 1, the $\left(\frac{\phi(m)k}{\log(\phi(m)k)}\right)^{1/4}$ term in the standard deviation of the error is a consequence of converting elliptic distributions into spherical ones in [LPR10]. It is unclear whether having this term is actually necessary for hardness or whether the elliptical distributions in [LPR10] are an artifact of the proof, and so in practice it may be enough to just sample with standard deviation $\sqrt{m'}\alpha q$. Fortunately, most constructions involving Ring-LWE only require a small (usually a constant or a logarithmic) number of samples, and so for theoretical applications when one does not care too much about small polynomial factors, this term does not cause too much trouble.

The final comment that we would like to make is about the "tightness" of our reduction. It is natural to wonder whether our transformation from Ring-LWE in the domain in Theorem 1 to the one in the domain in Theorem 2 is tight, in the sense that one did not need to add more noise than necessary in order to obtain pseudo-randomness in $R_q \times R_q$. We now give an intuition for why the transformation is actually rather tight. Ignoring the $\left(\frac{\phi(m)k}{\log(\phi(m)k)}\right)^{1/4}$ term, which is a possibly removable artifact carried over from Theorem 1, the required noise in our new theorem is $\sqrt{m'}\alpha q$, where there is a requirement that $\alpha q > \omega(\sqrt{\log m})$. Thus the noise must have standard deviation at least $\omega(\sqrt{m' \log m})$. This is almost tight because by the result of Arora and Ge [AG11], if the standard deviation were $o(\sqrt{\phi(m)})$, then the Ring-LWE problem could be solved in sub-exponential time $2^{o(\phi(m))}$, which would them imply that the Shortest Vector

Problem could be solved in sub-exponential time as well. And since $\sqrt{m'/\phi(m)} = O(\sqrt{\log\log m})$, this is essentially the maximum tightness factor that we lose during our reduction.

Proof of Theorem 2. To prove the theorem, we will show how one can transform the samples from Theorem 1 into samples from the ring $R_q \times R_q$. Given samples of the form $(a_i, a_i w/q + e_i) \in \mathbb{R}_q \times \mathbb{Q}(\zeta_m)/R^\vee$ where a_i are chosen uniformly at random from R_q, w is chosen uniformly at random from $R_q^\vee$, and the e_i are sampled from the distribution $\psi_s^{\phi(m)}$ in the embedding space H, we scale the second element of each ordered pair by a factor of $m'q$ to obtain elements $(a_i, a_i w m' + q m' e_i) = (a_i, a_i w' + e_i') \in R_q \times \mathbb{Q}(\zeta_m)/qm'R^\vee$ where w' is distributed uniformly at random in $m'R^\vee/m'qR^\vee$, and e_i' are sampled from the distribution $\psi_{sm'q}^{\phi(m)}$. Since we did nothing but scaling at this point, it is clear that distinguishing these ordered pairs from uniform ones in $R_q \times \mathbb{Q}(\zeta_m)/qm'R^\vee$ is as hard as the original problem from Theorem 1. We now apply Theorem 3 which states that $m'R^\vee \subseteq R$ to conclude that if we reduce the second entry of the ordered pairs modulo qR to obtain elements $(a_i, a_i w' + e_i') \in R_q \times \mathbb{Q}(\zeta_m)/qR$ where w' is distributed uniformly at random in $m'R^\vee/qR$, and e_i' are sampled from the distribution $\psi_{sm'q}^{\phi(m)}$, the distinguishing problem is at least as difficult as before.

We now make the observation that instead of choosing w' uniformly at random from $m'R^\vee/qR$, we can choose it from R/qR without making the problem any easier. The reason is that given a pair $(a_i, a_i w' + e_i')$, we can choose a uniformly random $w'' \in R/qR$ and output $(a_i, a_i w' + a_i w'' + e_i') = (a_i, a_i(w' + w'') + e_i')$, and the secret $w' + w''$ is uniform in R/qR. We can also observe that if we consider the element $a_i w' + e_i'$ in the power-basis representation and round each coefficient to the nearest integer, it is equivalent to only rounding the error term e_i' to the nearest integer because the product $a_i w'$ already has integer coefficients. Thus the problem of distinguishing rounded elements $(a_i, a_i s + \lceil e_i' \rfloor) \in R_q \times R_q$ from random elements in $R_q \times R_q$ is at least as difficult as the problem from Theorem 1. The last thing we need to address is the noise generation. Currently, the e_i' are generated from the distribution $\psi_{sm'q}^{\phi(m)}$ in the embedding space H. Theorem 5 states that to obtain such a distribution, it is equivalent to sample the distribution $g_0 + g_1 X + \ldots + g_{m'-1} X^{m'-1} \in \mathbb{Q}[X]/(\Theta_m)$ where each g_i is a normally distributed random variable with mean 0 and standard deviation that is $\sqrt{m'}$ times smaller than that required in the distribution in the embedding space H. And this is exactly the distribution from which the errors come from in the statement of our Theorem. □

4 Mapping $\mathbb{Z}[\zeta_m]^\vee$ to $\mathbb{Z}[\zeta_m]$

In this section we prove that the element $\frac{m'}{\Phi_m(\zeta_m)}$, for $m' = m$ when m is odd and $m/2$ when it is even, is an element of the ring $\mathbb{Z}[\zeta_m]$, which implies that the ring $\mathbb{Z}[\zeta_m]$ contains $m'\mathbb{Z}[\zeta_m]^\vee$.

Theorem 3. *For $R = \mathbb{Z}[\zeta_m]$, we have $m'R^\vee \subseteq R$, where $m' = m$ if m is odd and $m/2$ if m is even.*

Proof: Let $\Theta_m(X)$ be the polynomial $X^m - 1$, if m is odd, and $X^{m/2} + 1$ if m is even. Then it is easily seen that $\Phi_m(X)$ is a factor of $\Theta_m(X)$, and we can write $\Theta_m(X) = \Phi_m(X)g(X)$ for some polynomial $g(X) \in \mathbb{Z}[X]$. By taking the derivative of both sides, we obtain the equation

$$m' X^{m'-1} = \Phi'_m(X)g(X) + \Phi_m(X)g'(X),$$

or equivalently,

$$m' X^{m'} = X\Phi'_m(X)g(X) + X\Phi_m(X)g'(X).$$

Evaluating both sides at ζ_m, we obtain

$$\pm m' = \zeta_m \Phi'_m(\zeta_m)g(\zeta_m) + \zeta_m \Phi_m(\zeta_m)g'(\zeta_m) = \zeta_m \Phi'_m(\zeta_m)g(\zeta_m)$$

since $\zeta_m^{m'} = 1$ when $m' = m$ and -1 when $m' = m/2$, and $\Phi_m(\zeta_m) = 0$. Now, using the definition that $R^\vee = \frac{1}{\Phi'_m(\zeta_m)} R$, we obtain

$$m' R^\vee = \frac{m'}{\Phi'_m(\zeta_m)} R = \pm \zeta_m g(\zeta_m) R \subseteq R,$$

where the last inclusion is true because $g(X) \in \mathbb{Z}[X]$, and so $g(\zeta_m) \in R$. □

We get that if we multiply the different ideal by m', we find a set included in the ring of integer. In fact, we prove in Appendix B that m' is the smallest integer which verifies this property. It mainly comes from the fact that m' is the radical of the finite group $R^\vee/R$, namely the least common multiple of orders of the elements in this group. And we get eventually this following characterization:

Theorem 4. *A integer k is such that $kR^\vee \subset R$ if and only if m' divides k.*

5 Geometry and Error Sampling

For the rest of the paper, let $m' \in \mathbb{Z}$ denote $m/2$ if m is even, and m if m is odd.

To obtain the correct distribution of the error polynomials in the Ring-LWE problem in Theorem 1, we want the noise distribution over $\mathbb{Q}[X]/(\Phi_m)$ to map to a spherical Gaussian in the embedding space H. This is not a problem if the map $T^{-1} \circ \sigma$ is a scaled-orthonormal map, which is the case when m is a power of two. For a general m, a natural solution would be to generate the noise in the space H and then map it to $\mathbb{Q}[X]/(\Phi_m)$, however this requires dealing with the inverse Vandermonde matrix of σ^{-1}, making the noise generation much less efficient.

To overcome this technical issue, we use the ring extension $\mathbb{Q}[X]/(\Theta_m)$ and show that it is a the natural ring for the error generation. First unlike $\mathbb{Q}[X]/(\Phi_m)$ the canonical embedding from this ring preserves sphericity of Gaussian distributions: thus one just needs to sample a spherical Gaussian in this extension then reduce modulo ϕ_m.

Theorem 5 (Geometry of $T^{-1} \circ \sigma \circ \beta$). *Let $v \in \mathbb{Q}[X]/(\Theta_m)$ be a random variable distributed as $\psi_s^{m'}$ in the power basis. Then the distribution of $(T^{-1} \circ \sigma \circ \beta)(v)$, seen in the canonical basis of H is the spherical Gaussian $\psi_{s\sqrt{m'}}^{\phi(m)}$.*

Secondly, for a large class of integers m the reduction modulo Φ_m has a very simple and sparse matrix representation in the power basis. The knowledge of this matrix representation simplifies the geometric analysis of the error and products of errors, leading to some better theoretical bounds for correct decryption (see lemma 7), detailed below.

5.1 Analysis of β, the Reduction Modulo Φ_m

First, if B is very sparse and structured, this reduction can be implemented in a very simple ad-hoc way, while having better practical running time than general quasi-linear reduction algorithms. We will show that it is the case when $m = 2^k p$ for a any prime p, and also when $m = 2^k p' p$ if p' is a small prime.

Secondly, error distributions in the $\mathbb{Q}[X]/(\Phi_m)$ representation depend on the geometry of B, and thus the norms of B have an impact on the relation between m, s and q : the smaller the norms are, the smaller q one may choose while ensuring correct decryption. In particular, for any $e \in \mathbb{Q}[X]/(\Theta_m)$ we have: $\|\beta(e)\|_\infty \leq \|B\|_1 \|e\|_\infty$, which is related to the expansion factor inequality [LM06]. One may indeed only deal only with the expansion factor of Φ_m, and bound the error preimage in $\mathbb{Q}[X]/(\Theta_m)$. As described later, the main part of the error that needs to be dealt with for decryption has the form $ab + cd$ where a, b, c, d are drawn according to $\beta(\psi_s^{m'})$. Considering the tailcut function $\mathcal{E}(\tau) = \tau e^{1/2-\tau^2/2}$ we have the following fact:

Fact 6 (Error Bound in the Extension Ring $\mathbb{Q}[X]/(\Theta_m)$). *Let $a, b, c, d \in \mathbb{Q}[X]/(\Theta_m)$ be distributed as $\psi_s^{m'}$. Then, $\|ab + cd\|_\infty \leq \sqrt{2m'}\,\tau\,\tau' s^2$ except with probability less than $n\mathcal{E}(\tau) + \mathcal{E}(\tau')^{2m'}$.*

Since β is ring morphism, preserving products as well as sums, this translate to $\mathbb{Q}[X]/(\Phi_m)$:

$$\|\beta(a)\beta(b) + \beta(c)\beta(d)\|_\infty = \|\beta(ab + cd)\|_\infty \leq \|B\|_1 \sqrt{2m'}\,\tau\,\tau' s^2.$$

However, the exact knowledge of B, together with the knowledge of the error distribution may lead to better bounds. While there is no simple explicit formula for B in general, some specific values of m makes B very simple. Obviously, when m is a power of two, B is the identity since $\Theta_m = \Phi_m$. When $m = 2^k p$ we have:

$$B = \left(\begin{array}{c|c} \mathrm{Id}_{p-1} & \begin{matrix}-1\\ \vdots \\ -1\end{matrix}\end{array}\right) \text{ if } k = 0; \quad B = \left(\begin{array}{c|c} \mathrm{Id}_{p-1} & \begin{matrix}-1\\ 1\\ \vdots \\ -1\\ 1\end{matrix}\end{array}\right) \otimes \mathrm{Id}_{2^{k-1}} \text{ otherwise}$$

In that case, a better bound can be proved, replacing the constant $\|B\|_1 = 2$ by $\|B\|_2 = \sqrt{2}$.

Fact 7 (Error Bound for $m = 2^k p$). *Let p be a prime number, k a positive integer and assume $m = 2^k p$. Let $a, b, c, d \in \mathbb{Q}[X]/(\Theta_m)$ be distributed as $\psi_s^{m'}$. Then, $\|\beta(ab + cd)\|_\infty \leq 2\sqrt{m'}\tau\tau' s^2$, except with probability less than $m'\left(\mathcal{E}(\tau) + 3\,\mathcal{E}(\tau')^{2\lfloor m'/3 \rfloor}\right)$.*

The proof is available in the full version of this article. This statement raises the interesting question of whether it can be generalized to other values m, *i.e.* can we replace $\|B\|_1$ by $\|B\|_2$ (while keeping the exponent of $\mathcal{E}(\tau')$ big enough) ? While such constant $\|B\|_2$ applies to Gaussian errors, its not clear if it applies in general for products of Gaussians.

Other Polynomials Φ_m. For general values of m the coefficients of B may be much bigger, and can even grow exponentially in m for product of many primes. Few is known about the behavior of the coefficients of Φ_m in terms of the prime decomposition of m, however Lam and Leung proved in [LL96] that Φ_{pq} for two primes p and q have its coefficient in $\{-1, 0, 1\}$.

A generalization of their proof gives a more detailed behavior:

Theorem 8. *If m is on the form $m = 2^k pq$ where p q are two odd primes and $k \in \mathbb{N}$, B has coefficients in $\{-1, 0, 1\}$ and $\|B\|_1 = 2\min(p, q)$.*

The proof will be detailed in the full version of this article.

Improved Decryption. Additionally, the explicit knowledge of B can suggest strategies to improve the tolerance of the decryption algorithm. Such an idea is described when $m = p$ is a prime integer in section 6.4. It seems to improve the tolerance, replacing the $\|B\|_2 = \sqrt{2}$ factor by ≈ 1.16 for dimension $m \approx 500$; and seems to be $1 + o(1)$ when the dimension grows. With that improvement the tolerance loss compared to the encryption scheme based on Φ_{2^k} can becomes marginal.

6 Ring-LWE Encryption Scheme

In this section we present an application example of our result, that is an adaptation of the [LPR10] scheme to general polynomial Φ_m, and sketch strategies to improve the decryption rate.

6.1 Definition

We consider m to be our main security parameter, and we assume it grows in an unbounded set of integer S such that $\|B\|_1$ is polynomially bounded : $\|B\|_1 \leq O(m^b)$ for some $b \geq 0$. For example, we can take $b = 0$ for the set $S = \left\{2^k p | k \in \mathbb{Z},\ p \text{ is prime}\right\}$, while $S = \left\{2^k pq | k \in \mathbb{Z},\ p, q \text{ are prime}\right\}$ gives $b = 1/2$.

Choose some small $\epsilon \in (0, 1/4)$, and set other parameters to grow as follow : the modulus $q = \Theta(m^{2+b+2\epsilon})$, and the standard deviation $s = \Theta(m^{3/4+\epsilon})$. Our encryption scheme is as follows :

- **Gen**(1^m) : Sample $w, e_1 \leftarrow \psi_s^{m'}$, and a uniformly in R_q. Set $\bar{w} = \lfloor \beta(w) \rceil$ and $\bar{e}_1 = \lfloor \beta(e_1) \rceil$ The private key $\bar{w} = \lfloor \beta(w) \rceil \in R_q$ and the public key is $(a, \bar{t})$ where $\bar{t} = a\bar{w} + \bar{e}_1 \mod q \in R_q$
- **Encrypt**$(\bar{t} \in R_q,\ \mu \in \{0,1\}^{\phi(m)})$: To encrypt the message μ under the public key $\bar{t}$, draw $r, e_2, e_3 \leftarrow \psi_s^{m'}$, and set $\bar{r} = \lfloor \beta(r) \rceil$, $\bar{e}_2 = \lfloor \beta(e_2) \rceil$ and $\bar{e}_3 = \lfloor \beta(e_3) \rceil$. Output $(u, v) \in R_q \times R_q$ where $u = a\bar{r} + \bar{e}_2 \mod q$ and $v = \bar{t}\bar{r} + \bar{e}_3 + \mu \lfloor q/2 \rfloor \mod q$.
- **Decrypt**$(\bar{w} \in R_q, (u, v) \in R_q \times R_q)$: To decrypt (u, v) with the private key $\bar{w}$, compute $d = v - u\bar{w} \in R_q$, and decrypt the i-th bit μ_i as 0 if $d_i \in [-q/4; q/4]$, and as 1 otherwise.

6.2 Security

We prove semantic security based on the hardness of the approximate Shortest Vector Problem to within a factor $\tilde{O}(m^{5/2+b+\epsilon})$. For any constant number of samples k set :

$$\alpha^{-1} = \frac{\sqrt{m'}q}{s} \left(\frac{\phi(m')k}{\log(\phi(m')k)} \right)^{1/4} = O(m^{2+b+\epsilon}).$$

To fulfill the condition of theorem 2, we verify that :

$$\alpha q = s \frac{\log(\phi(m)k)^{1/4}}{\sqrt{m'}\phi(m)k^{1/4}} \geq \Theta(m^{\epsilon}) > \omega(\sqrt{\log m}), \qquad \text{since } \frac{m}{\phi(m)} = O(\log \log m).$$

Note that we use the main theorem 2 in its modified form that replaces the uniform distribution of the secret w by the same Gaussian distribution as the error (see the discussion under the statement of theorem 2).

First, the public key distribution $(\bar{a}, \bar{t} = \bar{a}\bar{w} + \bar{e}_1)$ follows the distribution defined in theorem 2 relatively to w which is distributed according to a Gaussian distribution. Thus for $k = 2$, our theorem states that this public key $(\bar{a}, \bar{t})$ is indistinguishable from the uniform distribution over $R_q \times R_q$.

We can now assume that $(\bar{a}, \bar{t})$ is uniformly random, in which case $(a, u = a\bar{r} + \bar{e}_2)$ and $(\bar{t}, v' = \bar{t}\bar{r} + \bar{e}_2)$ are two samples following the distribution of theorem 2, where $\bar{r}$ is once again Gaussian. Using theorem 2 with $k = 3$, we deduce that (a, u) and $(\bar{t}, v')$ are also is indistinguishable from random, so is $v = v' + \mu \lfloor q/2 \rfloor$ That concludes the security proof.

6.3 Correctness

During decryption, we get :

$$\begin{aligned} d = v - u\bar{w} &= (\bar{a}\bar{w} + e_1)\bar{r} + \bar{e}_3 + \mu \lfloor q/2 \rfloor - (\bar{a}\bar{r} + \bar{e}_2) \mod q \\ &= \bar{e}_1\bar{r} + \bar{e}_3 + \bar{e}_2\bar{w} + \mu \lfloor q/2 \rfloor \mod q \end{aligned}$$

Thus, the decryption will be correct if $\|\bar{e}\|_\infty < q/4$ where $\bar{e} = \bar{e}_1\bar{r} + \bar{e}_3 + \bar{e}_2\bar{w}$. First, note that the rounding operations have a limited effect on the final result of the error $\bar{e}$: the difference between that computation with and without rounding is bounded by $\tilde{O}(B_1 m' s) = \tilde{O}(m^{7/4+b+\epsilon})$, this negligible compared to $q = \Theta(m^{2+b+2\epsilon})$. Similarly, one can neglect the contribution of $\bar{e}_3$ since $\|\bar{e}_3\|_\infty \leq \tilde{O}(\|B\|_1 s) = \tilde{O}(m^{3/4+b+\epsilon})$.

According to lemma 6, we have that $\|\bar{e}\|_\infty \leq \tilde{O}(\|B\|_1 \sqrt{m} s^2) \leq \tilde{O}(m^{2+b+\epsilon})$ except with negligible probability. On the other hand, q grows as $\Theta(m^{2+b+2\epsilon})$, thus, decryption is correct with overwhelming probability for large enough values of m.

6.4 Practical Improvements

For applications, any tricks to decrease the minimal value of q while preserving correct decryption might be worthwhile. We hereby presents two independent ideas.

Recovering Approximation of the Error Preimage $e' \in \mathbb{Z}[X]/(\Theta_m)$. This first idea concerns the decryption algorithm. For simplicity, we restrict our attention to $m = p$ a prime. In this case we have for each index $i \leq p-2$ that $\bar{e}_i = e'_i - e'_{p-1}$ where $e' = e_1 r + e_3 + e_2 w \in \mathbb{Z}[X]/(\Theta_m)$. Thus if we recover a good approximation x of e'_{p-1} we might reduce the error by adding x to each coordinate. Without warping modulo q, an approximation of e'_{p-1} may be recovered as the average $\frac{-1}{p-1}\sum_{i=0}^{p-2} \bar{e}_i$; the error should be less than $\approx \tau s^2$, using the heuristic that e' behave like a spherical Gaussian.

However, we need to consider $\bar{e}_i$ modulo $q/2$ to get rid of the message. Our heuristic algorithm proceeds as follow : for a certain constant $\alpha \in (0,1)$, find (one of) the smallest interval $[a,b]$ such that for at least $\alpha(p-1)$ indexes $i \in [p-1]$ verifies $\bar{e}_i \in ([a,b] \mod q/2)$. Consider $a_i \in \mathbb{Z}$ as the unique integer representing $\bar{e}_i \in \mathbb{Z}_{q/2}$ in $[a,b]$, compute the average x of those a_i, and output the smallest representative of $-\lfloor x \rceil$ modulo $q/2$ as an approximation of e'_{p-1}. Note that this algorithm can be implemented in quasi-linear time, by sorting the values e_i.

Our experiments indicates that such a strategy decrease the $\sqrt{2} \approx 1.41$ factor to ≈ 1.16 for $m = 503$ and $\alpha = 0.9$, and keeps decreasing when the dimension grows. We conjecture that it asymptotically decrease as $1 + o(1)$. Similar idea should apply to $m = 2^k p$. While this suggest that the quality loss compared to cryptosystem based on the Φ_{2^k} polynomial can be almost reduced to nothing, implementing such a error recovery strategy would require more study.

Rejection during Key Generation. The second idea consist of modifying the key generation algorithm **Gen** so that the couple (s, e_1) is rejected whenever $\|(s|e_1)\| \geq \sqrt{2m'}\tau'' s$, where τ'' is chosen such that $\mathcal{E}(\tau'')^{2n} \leq 1/2$; only half of them are rejected, thus the advantage of the adversary is no more than doubled. For $m' \geq 500$ this improves our bound by $\tau'/\tau'' \approx 1.4/1.05$. The same idea applies when using the tight bound of lemma 7 by rejecting $(\bar{s}, \bar{e}_1)$ depending on $\|B \cdot \mathrm{Circ}(\bar{s})\|^2 + \|B \cdot \mathrm{Circ}(\bar{e}_1)\|^2$.

References

[ACPS09] Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 595–618. Springer, Heidelberg (2009)

[AG11] Arora, S., Ge, R.: New Algorithms for Learning in Presence of Errors. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011, Part I. LNCS, vol. 6755, pp. 403–415. Springer, Heidelberg (2011)

[BGV11] Brakerski, Z., Gentry, C., Vaikuntanathan, V.: Fully homomorphic encryption without bootstrapping, Cryptology ePrint Archive, Report 2011/277 (2011); To appear at ITCS 2012

[BPR11] Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom Functions and Lattices. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 719–737. Springer, Heidelberg (2012); Cryptology ePrint Archive, Report 2011/401 (2011)

[BV11a] Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: FOCS (2011)

[BV11b] Brakerski, Z., Vaikuntanathan, V.: Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 505–524. Springer, Heidelberg (2011)

[Con09] Conrad, K.: The different ideal (2009), http://www.math.uconn.edu/~kconrad/blurbs/

[DPSZ11] Damgard, I., Pastro, V., Smart, N.P., Zakarias, S.: Multiparty computation from somewhat homomorphic encryption. Cryptology ePrint Archive, Report 2011/535 (2011)

[Gen10] Gentry, C.: Toward Basing Fully Homomorphic Encryption on Worst-Case Hardness. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 116–137. Springer, Heidelberg (2010)

[GHS11] Gentry, C., Halevi, S., Smart, N.P.: Fully Homomorphic Encryption with Polylog Overhead. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 465–482. Springer, Heidelberg (2012); Cryptology ePrint Archive, Report 2011/566 (2011)

[LATV11] Lopez-Alt, A., Tromer, E., Vaikuntanathan, V.: Cloud-assisted multiparty computation from fully homomorphic encryption. Cryptology ePrint Archive, Report 2011/663 (2011)

[LL96] Lam, T.Y., Leung, K.H.: On the cyclotomic polynomial $\phi_{pq}(x)$. The American Mathematical Monthly 103(7), 562–564 (1996)

[LM06] Lyubashevsky, V., Micciancio, D.: Generalized Compact Knapsacks Are Collision Resistant. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006, Part II. LNCS, vol. 4052, pp. 144–155. Springer, Heidelberg (2006)

[LPR10] Lyubashevsky, V., Peikert, C., Regev, O.: On Ideal Lattices and Learning with Errors over Rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (2010)

[Lyu11] Lyubashevsky, V.: Lattice Signatures without Trapdoors. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 738–755. Springer, Heidelberg (2012); Cryptology ePrint Archive, Report 2011/537 (2011)

[MP11] Micciancio, D., Peikert, C.: Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012); Cryptology ePrint Archive, Report 2011/501 (2011)

[SS11] Stehlé, D., Steinfeld, R.: Making NTRU as Secure as Worst-Case Problems Over Ideal Lattices. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 27–47. Springer, Heidelberg (2011)

[Ste05] Stein, W.: Introduction to algebraic number theory (2005), http://wstein.org/courses/

[Was97] Washington, L.C.: Introduction to cyclotomic fields. Graduate Texts in Mathematics, vol. 83. Springer, New York (1997)

A Proof of Theorem 5

Proof: We proceed by considering $G \in \mathbb{C}^{\phi(m)\times m'}$, the matrix representing the linear map γ from the power basis of $\mathbb{Z}[X]/(\Theta_m)$ to the canonical basis of $\mathbb{C}^{\phi(m)}$ and will show $G\overline{G^t} = m'\mathrm{Id}_{\phi(m)}$. Also note that T^{-1} is hermitian, that is $T^{-1} = \overline{T}^t$, and $T^{-1} \circ \gamma$ is a real linear map. Thus $E = T^{-1}G = \overline{E}$ so $EE^t = E\overline{E}^t = T^{-1}G\overline{G}^tT = m'\,\mathrm{Id}_{\phi(m)}$.

This last equation implies that if a random variable $v \in \mathbb{Q}[X]/(\Theta_m)$ has covariance $s^2\cdot\mathrm{Id}_{m'}$ then the covariance of $(T^{-1}\circ\gamma)(v)$ is $s\cdot E\cdot\mathrm{Id}_{m'}\cdot\overline{E}^t = s^2m'\cdot\mathrm{Id}_n$: the distribution of $(T^{-1}\circ\gamma)(v)$ is the spherical Gaussian $\psi^{\phi(m)}_{s\sqrt{m'}}$.

Now we show that $G\overline{G^t} = m'\mathrm{Id}_{\phi(m)}$: let $g_{i,j}$ for $(i,j) \in [m'] \times \mathbb{Z}^*_m$ denotes the coefficients of G, that is $g_{i,j} = \sigma_j(X^i) = \zeta_m^{ij}$. Let $c_{i,j}$ for $i,j \in \mathbb{Z}^*_m$ denote the coefficients of $C = G\cdot\overline{G^t}$. For all $i,j \in \mathbb{Z}^*_m$ we have:

$$c_{i,j} = \sum_{k\in[m']} \zeta_m^{ik}\overline{\zeta_m^{jk}} = \sum_{k\in[m']} \left(\zeta_m^{i-j}\right)^k = \begin{cases} m' \text{ if } i = j\ , & \text{since } \zeta_m^{i-j} = 1 \\ 0 \text{ otherwise,} & \text{since } \zeta_m^{i-j} \neq 1 \text{ is a } m\text{-th root of unity} \\ & \text{(or an } m'\text{-th root when } m \text{ is even)} \end{cases}$$

□

B Proof of Theorem 4

First of all, we remind some facts about free abelian groups of finite rank directly apply to $\mathbb{Z}[X]/(\Phi_m)$ and its different ideal. For conciseness, we will note R for the ring $\mathbb{Z}[X]/(\Phi_m) \cong \mathbb{Z}[\zeta_m]$ in all this section.

Definition 9. *Let G be a group and I a set. We says that a family of element $(e_i)_{i\in I}$ of G is a basis of G is every element of G can be written uniquely as a finite linear combination with integer coefficients of elements of this family. If I is finite, this cardinal is called the rank of G.*

Notations. For two integers k and n, the predicate $k|n$ denotes that k divides n. Also, let n be an integer and p a prime numbers. We define the order of n at p, denoted by $\mathrm{ord}_p(n)$, as the positive integer α such that $p^\alpha | n$ and $p^{\alpha+1}$ does not divide n. It is the exponent of p in the prime decomposition of n if $p|n$, and 0 otherwise.

Fact 10. *There exists a basis $(e_i)_{1\leq i\leq \phi(m)}$ for $R^\vee$ and $\phi(m)$ positive integer $(b_i)_{1\leq i\leq \phi(m)}$ such that $(b_i e_i)_{1\leq i\leq \phi(m)}$ is a basis for R. And moreover, $\forall i \leq \phi(m)-1, b_i | b_{i+1}$.*

Proof: All elements of R can be uniquely written as a linear combination with integer coefficients of $(\zeta_m^i)_{0\leq i\leq \phi(m)-1}$, as this family is rationally independent. In a similar way, we have that all elements of $R^\vee$ can be uniquely written as a linear combination with integer coefficients of $(\frac{\zeta_m^i}{\Phi_n'(\zeta_m)})_{0\leq i\leq \phi(m)-1}$. Since $R \subset R^\vee$, we can write for all i, ζ_m^i in the latter family : $\forall i \in [|0, \phi(m)-1|], \zeta_m^i = \sum_{j=0}^{\phi(m)-1} a_{i,j} \frac{\zeta_m^j}{\Phi_m'(\zeta_m)}$. We end with a square matrix $A = (a_{i,j})_{0\leq i,j\leq \phi(m)-1}$ of dimension $\phi(m)$ with integer coefficients. And we consider its Smith normal form (Proposition 2.1.5 in [Ste05]): namely, there exists two matrix U and V with integer coefficients of dimension $\phi(m)$, invertible as integer matrices, such that $UAV = D$, where D is a diagonal matrix with positive integer coefficients on the form :

$$\begin{pmatrix} b_1 & 0 & \cdots & 0 \\ 0 & \ddots & & \\ & & b_r & \\ \vdots & & 0 & \\ & & & \ddots \\ 0 & \cdots & & 0 \end{pmatrix}$$

And, we have $b_i | b_{i+1}$ for any $i < r$. Besides, in our case, $r = \phi(m)$ and $\forall i < r, b_i \neq 0$. Indeed, let's notice that A is a change-of-basis matrix for two $\mathbb{Q}$-basis of $\mathbb{Q}(\zeta_m)$, then invertible, and then its determinant is non-zero. But $\det(D) = \det(UAV) = \det(U)\det(A)\det(V) = \det(A)$, because U and V are invertible as integer matrices, and thus have their determinant equal to 1.

This decomposition of A gives us a basis $(e_i)_{1\leq i\leq \phi(m)}$ for $R^\vee$ and $\phi(m)$ integer $(b_i)_{1\leq i\leq \phi(m)}$ such that $(b_i e_i)_{1\leq i\leq \phi(m)}$ is a basis for R, where $b_i | b_{i+1}$ for any $i \leq \phi(m)-1$. □

Thanks to this result, we can state two immediate consequences.

Fact 11. *With the notation of the previous Fact 10, an integer k is such that $kR^\vee \subset R$ if and only if $b_{\phi(m)} | k$. Therefore, $b_{\phi(m)} | m'$.*

Moreover we have the following equality:

$$\prod_{i=1}^{\phi(m)} b_i = m^{\phi(m)} \Big/ \prod_{\substack{p/m \\ p \ \text{prime number}}} p^{\phi(m)/(p-1)}$$

Proof: First, an integer k is such that $kR^\vee \subset R$ if and only if

$$\forall i \leq \phi(m) \quad ke_i \in R \tag{1}$$

or equivalently, if and only if for all i, there exist $c_i \in \mathbb{N}$ such that $ke_i = a_i b_i e_i$. This can be rewritten as

$$\forall i, \quad b_i | k.$$

By Fact 10, all the b_i's divides $b_{\phi(m)}$, so condition (1) is equivalent to: $b_{\phi(m)} | k$.

The Fact 10 gives us the cardinality of the finite group $R^\vee / R$, called the index of R in $R^\vee$ and denoted $[R^\vee : R] : [R^\vee : R] = \prod_{i=1}^{\phi(m)} b_i$.

Yet, this index is known and in fact equals to the absolute value of the discriminant of the cyclotomic field (Theorem 4.6 in [Con09]), whom we know an exact expression (Proposition 2.7 of [Was97]):

$$disc(\mathbb{Q}(\zeta_m)) = (-1)^{\frac{\phi(m)}{2}} m^{\phi(m)} \Big/ \prod_{\substack{p/m \\ p \text{ prime number}}} p^{\phi(m)/(p-1)}$$

□

Using previous facts, we may now prove our main results:

Theorem 12. *With the notation above, $m' = b_{\phi(m)}$ and a integer k is such that $kR^\vee \subset R$ if and only if $m' | k$.*

Proof: First, we prove that $m' | b_{\phi(m)}$. To prove this, we work on the prime factors of m'. More precisely, we show that for all prime $p | m'$, $\mathrm{ord}_p(m') \leq \mathrm{ord}_p(b_{\phi(m)})$, it immediately follows that $m' | b_{\phi(m)}$. Let p a prime factor of m' different from 2. Then by definition of m', $\mathrm{ord}_p(m') = \mathrm{ord}_p(m)$.

We proceed by assuming that $\mathrm{ord}_p(m) > \mathrm{ord}_p(b_{\phi(m)})$, and show that it is absurd. From Fact 10 we have $b_i | b_{i+1}$ for all $i < \phi(m)$. Thus: $\mathrm{ord}_p(m) - 1 \geq \mathrm{ord}_p(b_i)$ and summing over all i we get:

$$(\mathrm{ord}_p(m) - 1)\phi(m) \geq \sum_{i=1}^{\phi(m)} \mathrm{ord}_p(b_i). \tag{2}$$

Fact 11 tells us that:

$$\prod_{i=1}^{\phi(m)} b_i = m^{\phi(m)} \Big/ \prod_{\substack{p/m \\ p \text{ prime number}}} p^{\phi(m)/(p-1)}$$

and therefore

$$\mathrm{ord}_p \left(\prod_{i=1}^{\phi(m)} b_i \right) = \mathrm{ord}_p(m)\phi(m) - \frac{\phi(m)}{p-1}$$

$$\sum_{i=1}^{\phi(m)} \mathrm{ord}_p(b_i) = \phi(m) \cdot \left(\mathrm{ord}_p(m) - \frac{1}{p-1} \right)$$

Combining with the inequality (2) we deduce:

$$(\mathrm{ord}_p(m) - 1)\phi(m) \geq \phi(m)(\mathrm{ord}_p(m) - \frac{1}{p-1})$$

which is absurd since $p > 2$. We actually get that $\mathrm{ord}_p(m') \leq \mathrm{ord}_p(b_{\phi(m)})$, if p is prime factor of m' different from 2.

If 2 is a prime factor of m', the same reasoning is similar, starting with $\mathrm{ord}_2(m') = \mathrm{ord}_2(m) - 1$.

Therefore $m'|b_{\phi(m)}$. And the Theorem 3 with the Fact 11 tell us that $b_{\phi(m)}|m'$. Thus $m' = b_{\phi(m)}$. And by the Fact 11 again, a integer k is such that $kR^\vee \subset R$ if and only if $m'|k$. □

On Homomorphic Encryption and Chosen-Ciphertext Security

Brett Hemenway[1] and Rafail Ostrovsky[2,⋆]

[1] University of Michigan
[2] UCLA

Abstract. Chosen-Ciphertext (IND-CCA) security is generally considered the right notion of security for a cryptosystem. Because of its central importance much effort has been devoted to constructing IND-CCA secure cryptosystems.

In this work, we consider constructing IND-CCA secure cryptosystems from (group) homomorphic encryption. Our main results give natural and efficient constructions of IND-CCA secure cryptosystems from any homomorphic encryption scheme that satisfies weak cyclic properties, either in the plaintext, ciphertext or randomness space. Our results have the added benefit of being simple to describe and analyze.

1 Introduction

Since the definition of security against a Chosen-Ciphertext Attack (IND-CCA) was given in [NY90], [RS91], much effort has been devoted to constructing efficient IND-CCA secure cryptosystems under a variety of cryptographic hardness assumptions.

The first construction of an IND-CCA secure cryptosystem was given by Dolev, Dwork and Naor in [DDN91]. Their construction builds on the ideas of Naor and Yung [NY90], and relies on non-interactive zero-knowledge proofs, to prove that a ciphertext was created honestly. The generic non-interactive zero-knowledge proofs used in [DDN91] are too inefficient for practical use, but the idea of including some sort of "proof of validity" in the ciphertext has strongly shaped this area of research, and many of the subsequent IND-CCA secure cryptosystems can be viewed in this light.

⋆ R. Ostrovsky, University of California Los Angeles, Department of Computer Science and Department of Mathematics, 3732D Boelter Hall, Los Angeles CA 90095-1596, U.S., email: rafail@cs.ucla.edu. Supported in part by NSF grants 0830803, 09165174, 1065276, 1118126 and 1136174, US-Israel BSF grant 2008411, OKAWA Foundation Research Award, IBM Faculty Research Award, Xerox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, and Lockheed-Martin Corporation Research Award. This material is based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11-1-0392. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 52–65, 2012.

The first IND-CCA secure cryptosystem efficient enough to be used in practice was given by Cramer and Shoup in [CS98], and the security of their construction rested on the Decisional Diffie-Hellman (DDH) assumption. Since then, there have many fairly efficient IND-CCA secure schemes proposed under a wide variety of cryptographic hardness assumptions.

Constructions based on the DDH assumption include those of [CS98],[CS02] and [PW08]. Recently, new constructions were given based on the Computational Diffie-Hellman (CDH) assumption [HJKS10], [CHK10]. IND-CCA secure cryptosystems based on the RSA assumption are given in [CHK10]. Schemes based on the Quadratic Residuosity (QR) assumption are given in [CS02]. IND-CCA secure cryptosystems based on lattice assumptions like Learning With Errors (LWE) are given in [PW08] and [Pei09]. In the pairing world, IND-CCA secure schemes can be based on the Bilinear Diffie-Hellman (BDH) assumption [CHK04],[BK05],[BCHK07], or the Decisional Linear (D-Lin) assumption [FGK+10]. Chosen-ciphertext secure cryptosystems have also been proposed based on the Syndrome Decoding problem [DMQN09], [FGK+10].

For a notion as fundamental as secure encryption, it is important to consider generic constructions as well as concrete instantiations, and in fact, many of the above constructions are best viewed as part of general frameworks for constructing IND-CCA secure encryption. In [DDN91], IND-CCA secure cryptosystems were built from any one-way trapdoor permutation. In [CS02], Cramer and Shoup gave a general construction based on universal hash proof systems, which can be viewed as an algebraic designated verifier proof system. In [CHK04],[BCHK07], Boneh, Canetti, Halevi and Katz gave a general framework for constructing IND-CCA secure encryption from any Identity-Based Encryption (IBE) scheme. In [PW08], Peikert and Waters created lossy trapdoor functions (LTDFs) as a method for constructing IND-CCA secure encryption. The notion of lossy trapdoor functions has since been relaxed to correlated product secure functions [RS09], and slightly lossy trapdoor functions [MY09], and both relaxations were shown to still be sufficient to construct IND-CCA secure encryption.

These frameworks provide many different constructions of IND-CCA secure encryption, and help to locate IND-CCA secure encryption in the cryptographic landscape. Despite their utility, these frameworks all rely on fairly complicated underlying primitives, and the search continues for the simplest primitive that can be shown to imply IND-CCA secure encryption. Perhaps the simplest primitive that could imply IND-CCA secure encryption is IND-CPA secure encryption. This, however, is widely believed to be false, and the results of Gertner, Malkin and Myers [GMM07] give partial results towards the impossibility of such a construction.

It is natural, then, to examine what additional properties of an IND-CPA secure cryptosystem are sufficient to construct an IND-CCA secure cryptosystem. One natural property, is that the IND-CPA secure cryptosystem supports a group operation on the plaintext. Such cryptosystems are called *homomorphic*. Indeed, one of the main open questions concerning homomorphic encryption is

whether homomorphic encryption implies IND-CCA encryption, and this question has attracted much attention over the years.

In this work, we will call an encryption scheme *homomorphic* if the plaintexts form a group, the ciphertexts form a group, and $E(pk, m_1, r_1) \cdot E(pk, m_2, r_2) = E(pk, m_1 + m_2, r^*)$. Unless explicitly stated, we will not assume that $r^* = r_1 + r_2$, schemes that satisfy this additional property are said to be *homomorphic over their randomness*.[1] Here we have written the group operation on the ciphertexts multiplicatively and the group operations on the plaintexts additively. This is simply a convention, but it is a natural one since it corresponds to the usual method of writing the groups corresponding to Goldwasser-Micali [GM84], Paillier [Pai99], and (additive) El-Gamal [Gam85]. We do *not* require our encryption schemes to be *fully homomorphic*, as constructed in the breakthrough work of Gentry [Gen09].

The consequences of the existence of homomorphic encryption have been well studied, and many exciting results are known. Homomorphic encryption has been show to imply Private Information Retrieval (PIR) [KO97],[Man98],[IKO05]. Since PIR implies Collision Resistant Hash Functions [IKO05], Oblivious Transfer [CMO00], and lossy encryption [HLOV11], we immediately have constructions of any of these primitives based on any homomorphic encryption. The work of [AKP10] provides a clean abstraction of homomorphic encryption and a discussion of homomorphic encryption and its relations to IND-CCA1 security.

It remains an important open question whether homomorphic encryption implies IND-CCA secure cryptosystems, and in this work we present steps towards closing the gap.

1.1 Previous Work

Chosen-ciphertext security was introduced by Rackoff and Simon in [RS91], and the first cryptosystem provably secure in this model was given in [DDN91], extending the work of [NY90]. Since that time, there has been a vast amount of work done on the topic of IND-CCA secure encryption.

Our work draws most from the works of Cramer and Shoup on universal hash proof systems [CS02], and Peikert and Waters on lossy trapdoor functions [PW08], and we briefly highlight some key ideas of their constructions below.

The first practical IND-CCA secure cryptosystem was given by Cramer and Shoup in [CS98]. In [CS02], Cramer and Shoup created Universal Hash Proof systems, generalizing their work in [CS98], and providing a framework for creating IND-CCA secure encryption. In [CS02], Cramer and Shoup defined a natural algebraic object called a *Diverse Group System*, and showed that diverse group systems imply universal hash proof systems, and diverse group systems are implied by many natural cryptographic hardness assumptions that occur in groups. The algebraic nature of diverse group systems suggests a possible connection

[1] Notice that our definition of homomorphic encryption implies that the randomness space forms a group, since the randomness space is isomorphic to the subgroup of encryptions of 0.

between homomorphic encryption and IND-CCA secure encryption, and in this work we explore this connection.

A different framework for constructing IND-CCA secure cryptosystems was proposed by Peikert and Waters in [PW08]. In their work, Peikert and Waters defined Lossy Trapdoor Functions (LTDFs), and showed that LTDFs imply IND-CCA secure cryptosystems. Roughly, a lossy trapdoor function, is a function that can operate in one of two computationally indistinguishable modes. In injective mode, it is injective and has a trapdoor. In "lossy" mode, the function statistically loses information about its input. In [PW08], Peikert and Waters leveraged the homomorphic properties of the El-Gamal cryptosystem and the Regev [Reg05] cryptosystem to create LTDFs based on the DDH and LWE assumptions. At the highest level, their construction proceeds as follows. The description of an LTDF in injective mode is simply the encryption of the identity matrix using some underlying homomorphic cryptosystem, and the description of an LTDF in lossy mode is the encryption of the zero matrix. To evaluate a function on an input x, viewed as a bit vector, we compute the matrix product of the ciphertext matrix with the input vector. By the homomorphic properties of the underlying cryptosystem, this results in either a ciphertext vector encrypting x, or a ciphertext vector encrypting the zero vector. It is easy to see that the IND-CPA security of the underlying cryptosystem implies that the injective and lossy modes are indistinguishable, and the decryption algorithm provides a trapdoor in injective mode. The difficulty is in showing that the lossy mode *statistically* loses information about its input. Let us examine this further. The output of a lossy function is the encryption of the zero vector, so it is clear that the underlying plaintexts are statistically independent of the input x (since they are all 0). It is, however, unclear whether the *randomness* of the ciphertexts statistically encodes the vector x. The constructions of LTDFs given by Peikert and Waters, modify the underlying homomorphic cryptosystems to ensure that the randomness of the resulting ciphertext vector does not leak too much information about the input x.

Both the works of [CS02] and [PW08] give an indication of the connection between homomorphic encryption and IND-CCA secure encryption, but despite significant effort, no one has, as yet, been able to bridge the gap.

In this work, we show that if we have a homomorphic cryptosystem with some natural cyclic structure, we immediately have IND-CCA secure encryption.

1.2 Our Contributions

In this work, we consider the problem of constructing an IND-CCA secure cryptosystem from homomorphic encryption schemes. By a homomorphic encryption scheme, we mean an IND-CPA secure cryptosystem, for which the plaintext space forms a group, the ciphertext space forms a group, and the group operation on ciphertexts induces a group operation on plaintexts. Cryptosystems of this type arise naturally, e.g. [Gam85, GM84, Pai99, Ben94, OU98, NS98, DJ01, BGN05].

It has been a long standing open question whether an IND-CCA secure cryptosystem can be constructed from any homomorphic encryption scheme. In this

work, we give a number of simple properties for a homomorphic encryption scheme, any one of which allows us to construct an IND-CCA secure cryptosystem.

Our results can be summarized as follows:

Lemma (Lemma 1 (informal)). *If there exists a homomorphic encryption with cyclic plaintext group X, and randomness space R, such that $|X| > |R|$, then there exist lossy trapdoor functions.*

Corollary (Corollary 1 (informal)). *If there exists a homomorphic encryption with cyclic ciphertext space, with plaintext group X, and randomness space R, such that $|X| > |R|$, then there exist lossy trapdoor functions.*

Theorem (Main Theorem (informal)). *If there exists a homomorphic encryption with cyclic ciphertext space, then there exist universal hash proof systems, and hence IND-CCA secure encryption.*

2 Preliminaries

2.1 Notation

If $f : X \to Y$ is a function, for any $Z \subset X$, we let $f(Z) = \{f(x) : x \in Z\}$. For example, if E is an encryption algorithm $E(pk, x, R) = \{E(pk, x, r) : r \in R\}$, is the set of all encryptions of x. Similarly, $E(pk, X, R) = \{E(pk, x, r) : x \in X, r \in R\}$ is the ciphertext space of E. If G is a group, and $g_1, \ldots, g_d$ are elements of G, then we use the notation $\langle g_1, \ldots, g_d \rangle$ to denote the subgroup of G generated by $g_1, \ldots, g_d$.

If A is a PPT machine, then we use $a \leftarrow A$ to denote running the machine A and obtaining an output, where a is distributed according to the internal randomness of A. If R is a set, and no distribution is specified, we use $r \leftarrow R$ to denote sampling uniformly from the uniform distribution on R.

If X and Y are families of distributions indexed by a security parameter λ, we say that X is statistically close to Y, (written $X \approx_s Y$) to mean that for all polynomials p and sufficiently large λ, we have $\sum_x |\Pr[X = x] - \Pr[Y = x]| < \frac{1}{p(\lambda)}$.

We say that X and Y are computationally close (written $X \approx_c Y$) to mean that for all PPT adversaries A, for all polynomials p, and for all sufficiently large λ, we have $|\Pr[A^X = 1] - \Pr[A^Y = 1]| < 1/p(\lambda)$.

2.2 Homomorphic Encryption

A public key cryptosystem given by algorithms (G, E, D) is called *homomorphic* if

- The plaintext space forms a group X (written with group operation $+$).
- The ciphertexts are members of a group Y.
- For all $x_0, x_1 \in X$, and for all r_0, r_1 in the randomness space R, there exists an $r^* \in R$ such that

$$E(pk, x_0 + x_1, r^*) = E(pk, x_0, r_0)E(pk, x_1, r_1).$$

Notice that we do not assume that the encryption is also homomorphic over the randomness, as is the case of most homomorphic encryption schemes, e.g. ElGamal, Paillier, and Goldwasser-Micali. We also do not assume that the image $E(pk, X, R)$ is the whole group Y, only that $E(pk, X, R) \subset Y$. Since the homomorphic property implies closure, we have that $E(pk, X, R)$ is a semi-group.[2] Notice also, that while it is common to use the word "homomorphic" to describe the cryptosystem, encryption is *not* a homomorphism in the mathematical sense (although decryption is).

We now show some basic properties from all homomorphic encryption schemes. These facts are commonly used but, since our definition is weaker than the (implicit) definitions of homomorphic encryption that appear in the literature, it is important to note that they hold under this definition as well.

- $E(pk, X, R)$ is a group.
- $E(pk, 0, R)$ is a subgroup of $E(pk, X, R)$.
- For all $x \in X$, $E(pk, x, R)$ is the coset $E(pk, x, r)E(pk, 0, R)$.
- For all $x_0, x_1 \in X$, $|E(pk, x_0, R)| = |E(pk, x_1, R)|$.
- If y is chosen uniformly from $E(pk, 0, R)$, then $yE(pk, x, r)$ is uniform in $E(pk, x, R)$.
- $E(pk, X, R)$ is such that $E(pk, X, R) \simeq X \times E(pk, 0, R)$ and decryption is the homomorphism

$$E(pk, X, R) \to E(pk, X, R)/E(pk, 0, R) \simeq X.$$

- If y is chosen uniformly from $E(pk, 0, R)$, then for any $x \in X$, $yE(pk, x, r)$ is uniformly distributed in $E(pk, X, R)$. This follows because the map

$$\begin{aligned} f : E(pk, 0, R) &\to E(pk, X, R) \\ y &\mapsto yE(pk, x, r) \end{aligned}$$

  is an injection because the group element $E(pk, x, r)$ has an inverse in Y (we do not need to assume that the inverse is a valid ciphertext). Thus by counting, we see that f is in fact a bijection. Hence if y is uniformly distributed, so is $f(y)$.

We call a public key cryptosystem a *homomorphic public key encryption scheme*, if it is IND-CPA secure and homomorphic.

2.3 Diverse Group Systems

In [CS02], Cramer and Shoup defined diverse group systems and used them as a foundation for all their constructions of Universal Hash Proof Systems. We review these definitions here.

Let Z, L, Π be finite abelian groups written additively, with $L \subsetneq Z$. Let $\mathrm{Hom}(Z, \Pi)$ be the group of homomorphisms, $\phi : Z \to \Pi$. This is also clearly an abelian group under the operation $(\phi_1 + \phi_2)(x) = \phi_1(x) + \phi_2(x)$.

[2] A semi-group satisfies the axioms of a group but is not guaranteed to have an identity element or inverses.

Definition 1 (Group System). *Let Z, L, Π be finite abelian groups with $L \subsetneq Z$. Let $\mathbf{H} \subset \mathrm{Hom}(Z, \Pi)$. We call*

$$\mathbf{G} = (\mathbf{H}, Z, L, \Pi),$$

a group system.

Definition 2 (Diverse Group System). *We call a group system $\mathbf{G} = (\mathbf{H}, Z, L, \Pi)$* diverse *if for all $z \in Z \setminus L$, there exists $\phi \in \mathbf{H}$ such that $\phi(\ell) = 0$ for all $\ell \in L$, but $\phi(z) \neq 0$.*

In [CS02] Cramer and Shoup show a natural method for constructing Universal Hash Proof Systems from Diverse Group Systems.

Definition 3 (Hash Proof System Associated to a Diverse Group System). *Let $\mathbf{G} = (\mathbf{H}, Z, L, \Pi)$ be a diverse group system, and let $g_1, \ldots, g_d \in L$ be a set of generators for L. We define the associated Hash Proof system $\mathbf{UHP} = (H, K, Z, L, \Pi, S, \alpha)$,*

- *For uniformly chosen $k \in K$, H_k is uniform on $\mathbf{H}$.*
  Without loss of generality, we may assume $K = \mathbf{H}$, and $k = \phi \in \mathbf{H}$.
  We maintain Universal Hash Proof notation to emphasize that $H_k(\cdot)$ that someone who can calculate $H_k(\cdot)$ on elements of L may not know the underlying homomorphism ϕ.
- *$S = \Pi^d$, and*

$$\begin{aligned} \alpha : K &\to S \\ k &\mapsto (H_k(g_1), \ldots, H_k(g_d)). \end{aligned}$$

2.4 Lossy Trapdoor Functions

We briefly review the notion of *Lossy Trapdoor Functions* (LTDFs) as described in [PW08].

Intuitively, a family of Lossy Trapdoor Functions is a family of functions which have two modes, injective mode, which has a trapdoor, and lossy mode which is guaranteed to have a small image size. In particular, the preimage of any element in the image will have a large size. Formally we have:

Definition 4 (Lossy Trapdoor Functions). *A tuple $(S_{\mathrm{ltdf}}, F_{\mathrm{ltdf}}, F^{-1}_{\mathrm{ltdf}})$ of PPT algorithms is called a family of (n, k)-Lossy Trapdoor Functions if the following properties hold:*

- ***Sampling Injective Functions:*** *$S_{\mathrm{ltdf}}(1^\lambda, 1)$ outputs s, t where s is a function index, and t its trapdoor. We require that $F_{\mathrm{ltdf}}(s, \cdot)$ is an injective deterministic function on $\{0,1\}^n$, and $F^{-1}_{\mathrm{ltdf}}(t, F_{\mathrm{ltdf}}(s, x)) = x$ for all x.*
- ***Sampling Lossy Functions:*** *$S_{\mathrm{ltdf}}(1^\lambda, 0)$ outputs $(s, \perp)$ where s is a function index and $F_{\mathrm{ltdf}}(s, \cdot)$ is a function on $\{0,1\}^n$, where the image of $F_{\mathrm{ltdf}}(s, \cdot)$ has size at most 2^{n-k}.*
- ***Indistinguishability:*** *The first outputs of $S_{\mathrm{ltdf}}(1^\lambda, 0)$ and $S_{\mathrm{ltdf}}(1^\lambda, 1)$ are computationally indistinguishable.*

3 Implications of Homomorphic Encryption

Much effort has been devoted studying the implications of homomorphic encryption, and many results are now known. It is known that homomorphic encryption implies Private Information Retrieval (PIR) [KO97],[Man98],[IKO05], and since PIR implies Collision Resistant Hash Functions [IKO05], Oblivious Transfer [CMO00], and lossy encryption [HLOV11], we immediately have constructions of these primitives based on any homomorphic encryption. It remains open, however, whether homomorphic encryption implies IND-CCA secure cryptosystems.

Our main contributions are a step towards resolving this long-standing open question.

3.1 Constructing Lossy Trapdoor Functions

As in Section 2.2, throughout the following section, let (G, E, D) be a homomorphic encryption with plaintext group X, and randomness space R. We write the group operation on X additively and the group operation on ciphertexts multiplicatively.

We begin by attempting to generalize the construction of lossy trapdoor functions from the Damgård-Jurik cryptosystem given by [BFOR08], [RS08] and [FGK+10].

- **Sampling Injective Functions:** $S_{\mathrm{ltdf}}(1^\lambda, 1)$, runs $(pk, sk) \leftarrow G(1^\lambda)$, and chooses $r \leftarrow R$, and sets $e = E(pk, 1, r)$. The function index $s = (pk, e)$, and the trapdoor $t = sk$.
- **Sampling Lossy Functions:** $S_{\mathrm{ltdf}}(1^\lambda, 0)$, runs $(pk, sk) \leftarrow G(1^\lambda)$, and chooses $r \leftarrow R$, and sets $e = E(pk, 0, r)$. The function index $s = (pk, e)$, and the trapdoor $t = \perp$.
- **Evaluation:** Given $s = e$ and an input $a \in \{0, 1, \ldots, B-1\}$,

$$F_{\mathrm{ltdf}}(s, a) = e^a$$

- **Inversion:** Given $t = sk$, and a value c, set $a = D(sk, c)$.

Fig. 1. Generalizing the DCR-based LTDFs

Lemma 1. *Let (G, E, D) be a homomorphic encryption such that the plaintext group X is cyclic, with $|X| \geq B > |R|$, for some publicly known bound $B \in \mathbb{Z}$, then the construction given in Figure 1 is a family of lossy trapdoor functions.*

Proof. Note that the homomorphic property of the cryptosystem ensures that the product of two ciphertexts can be computed efficiently, so e^a can be computed efficiently using the square-and-multiply algorithm. Correctness of inversion follows immediately from the correctness of decryption. The indistinguishability of

modes follows immediately from the IND-CPA security of (G, E, D). It remains only to consider the lossiness of the lossy mode.

The output of the function in lossy mode is $F_{\text{ltdf}}(s, a) = e^a$, where $e = E(pk, 0, r)$, thus $F_{\text{ltdf}}(s, a)$ is a valid encryption of 0, i.e. $F_{\text{ltdf}}(s, a) \in E(pk, 0, R)$. Since the size of $|E(pk, 0, R)| \leq |R|$, and there are B choices for a, with $B > |R|$, the function is lossy. It is clear as well that as the ratio of B to $|R|$ grows, the functions become more lossy. If the size of X is efficiently computable, then it is natural to take $B = |X|$.

Lemma 1 has an immediate corollary, that if we assume instead that the ciphertext space is cyclic, we obtain the same result.

Corollary 1. *If (G, E, D) is a homomorphic encryption such that the group $E(pk, X, R)$ is cyclic with $|X| > |R|$, then the construction in Lemma 1 is a family of lossy trapdoor functions.*

Proof. The decryption algorithm provides an isomorphism between $E(pk, X, R)/E(pk, 0, R)$ and X, and since the quotient group of a cyclic group is cyclic, we conclude that X must be cyclic, and the result follows from Lemma 1.

The construction outline in Figure 1 leaves much to be desired, the three primary drawbacks are:

1. This construction requires a known public bound B, separating the size of the plaintext and randomness spaces. This condition seems extremely mild, however, since the definition of IND-CPA security requires the plaintext space be efficiently samplable, and the group is cyclic.
2. The requirement that the messages be longer than the randomness in Lemma 1 is rather strong. In fact, the Damgård-Jurik cryptosystem is the only homomorphic cryptosystem known to have this property.[3] In the next section, we show how to remove this restriction on the size of the plaintext space.
3. Decryption involves a somewhat more subtle difficulty. A careful look at the functions in Lemma 1 shows that the input is $a \in \{0, \ldots, B-1\}$, yet the trapdoor reveals $1 \cdot a \in X$. If $a \in \mathbb{Z}$ can be recovered from $1 \cdot a \in X$ (i.e. the Discrete Log Problem is easy in X[4]), this will not be an issue.
   In the case that the discrete-log problem is hard in the plaintext group X, we can still apply F_{ltdf} on random inputs, which may be enough for some applications. To see this, notice that we can sample pairs $x, F_{\text{ltdf}}(s, x)$, by sampling $a \leftarrow \{0, 1, \ldots, B-1\}$, setting $x = 1 \cdot a \in X$, and setting $F_{\text{ltdf}}(s, x) = e^a$. With this (slightly modified) definition, inversion becomes efficient. We can no longer evaluate, $F_{\text{ltdf}}(s, \cdot)$ on given values of x, but we can sample pairs $x, F_{\text{ltdf}}(s, x)$, where x is chose (almost) uniformly. This is not a serious restriction, however, since one-wayness only makes sense when applying a function to a high min-entropy input.

[3] It is trivial to construct non-homomorphic cryptosystems that have this property by extending the randomness using a PRG.

[4] As is the case with the Damgård-Jurik cryptosystem.

Unfortunately, this is *not* enough to apply the constructions of IND-CCA secure encryption from LTDFs given in [PW08, RS09, MY09]. Although these constructions require applying $F_{\text{ltdf}}(s, \cdot)$ on a random input, the decryption algorithm requires inverting one function and then evaluating the All-But-One function to the recovered input. This second evaluation cannot be performed when the discrete-log problem is hard in the plaintext space X.

We can quantify the lossiness of the functions given in Figure 1 based on the ratio of B to $|R|$. If $\frac{B}{|R|} = \omega(\lambda)$, then we obtain strong lossy trapdoor functions, as required for the constructions in [PW08]. If we only have $B/|R| > 1 + 1/\text{poly}(\lambda)$, then we obtain slightly lossy trapdoor functions as defined by Mol and Yilek [MY09]. The results of Mol and Yilek show that this is in fact sufficient for constructing Correlated Product Secure Functions [RS09], and IND-CCA secure cryptosystems.[5]

3.2 Constructing Diverse Group Systems

The generalization of the construction of the [BFOR08, RS08, FGK+10] given in Section 3.1 leaves much to be desired. In this section, by applying a different method, we are able to obtain a stronger result[6] than in Section 3.1 under a slightly different assumption. In particular, we show that any homomorphic encryption with cyclic ciphertext space (e.g. Goldwasser-Micali, Paillier), immediately implies Diverse Group Systems as defined by Cramer and Shoup in [CS02]. This method does not suffer from many of the drawbacks of the previous method.

Theorem 1. *Let (G, E, D) be a homomorphic encryption with plaintext group X and ciphertext group Y. If the group $E(pk, X, R)$ is cyclic, then $\mathbf{G} = (\mathbf{H}, Z, L, \Pi)$ is a* Diverse Group System. *Let $\gamma = |E(pk, X, R)|$.*

- *$Z = E(pk, X, R) \subset Y$, is the group of all encryptions.*
- *$\mathbf{H}$ is the set of homomorphisms given by exponentiating in the group, i.e. for $k \in \{0, 1, \ldots, \gamma\}$, and $z \in Z$, $H_k(z) = z^k$. So $|\mathbf{H}| = |E(pk, X, R)| = |Z|$.*
- *$L = E(pk, 0, R)$ is the group of all encryptions of 0.*
- *$\Pi = Z = E(pk, X, R)$.*

Proof. To show that $\mathbf{G}$ is *diverse*, we must show that for all $z \in Z \setminus L$, there exists a $\phi \in \mathbf{H}$ such that $\phi(L) = \langle 0 \rangle$, but $\phi(z) \neq 0$.

Let $\eta = |L|$, and $\gamma = |Z|$. Since Z was assumed to be cyclic, and L is a subgroup of Z, we know that L is cyclic and $\eta = |L|$ divides $|Z| = \gamma$. Now, it is also a basic fact about cyclic groups that L is exactly the subgroup of elements of Z whose order divides η, i.e. $L = \{z : z \in Z, z^\eta = 1\}$. For any $z \in Z \setminus L$,

[5] Again, we reiterate, that to apply the results of [PW08, RS09, MY09], the discrete-log problem must be solvable in X.

[6] We will construct a Diverse Group System which is known to imply LTDFs by [HO12].

Let d be the order of z, i.e. d is the smallest positive integer such that $z^d = 1$. Since $z \notin L$, we know that d doesn't divide η. Thus we may set $k = \eta$, (or any multiple of η not divisible by d). In which case, we have $H_k(z) = z^\eta \neq 0$. But $H_k(\ell) = \ell^\eta = 0$ for all $\ell \in L$. This shows that any cyclic group (with a proper subgroup) gives rise to a Diverse Group System.

To prove security, however, we need to show that L and Z are indistinguishable. This follows easily, however, since L is the set of encryptions of 0, and Z is the set of all encryptions, they are indistinguishable by the IND-CPA security of (G, E, D).

Applying the results of [CS02], which show that Diverse Group Systems imply universal hash proof systems, and universal hash proof systems imply IND-CCA secure cryptosystems, we arrive at the following result.

Corollary 2. *Homomorphic encryption with cyclic ciphertext space implies IND-CCA secure encryption.*

Applying the results of [HO12], which show that Diverse Group Systems imply Lossy Trapdoor Functions, we have

Corollary 3. *Homomorphic encryption with cyclic ciphertext space implies Lossy Trapdoor Functions.*

Applying the results of [BFOR08], we have

Corollary 4. *Homomorphic encryption with cyclic ciphertext space implies Deterministic Encryption.*

Corollary 5. *If (G, E, D) is a homomorphic encryption with cyclic* randomness *space, and there is an element $x_0 \in X$ such that the order of x_0 in the group X is relatively prime to $|R|$, then there is an IND-CCA secure cryptosystem.*

Proof. We define a new cryptosystem (G', E', D'), with plaintext space X', and randomness space R'. We set $X' = \langle x_0 \rangle \subset X$, and $R' = R$. We define $G' = G$, $E'(pk, x, r) = E(pk, x, r)$, for $x \in X'$, and $D' = D$. We claim that the ciphertext space of (G', E', D') is cyclic. To see this, notice first that the map $R \to E(pk, 0, R)$, given by $r \mapsto E(pk, 0, r)$ is a surjective homomorphism, thus $E(pk, 0, R)$ is isomorphic to a quotient group of R. Since R is cyclic, all its quotient groups are cyclic, so we see that $E(pk, 0, R)$ is also cyclic, in addition $|E(pk, 0, R)|$ divides $|R|$. Since $E(pk, 0, R) = E'(pk, 0, R')$, we have $|E'(pk, 0, R')|$ divides $|R|$, and is thus relatively prime to the order of the cyclic group $|\langle x_0 \rangle|$, which has size equal to the order of x_0. Thus the group $\langle x_0 \rangle \times E'(pk, 0, R')$ is cyclic, but this group is isomorphic to $E'(pk, X', R')$, so we may apply Theorem 1 to construct an IND-CCA secure cryptosystem.

4 Conclusion

In this work, we examined the connection between homomorphic encryption and chosen-ciphertext (IND-CCA) secure encryption. In particular, we showed that

any homomorphic encryption with a large cyclic plaintext space implies Lossy Trapdoor Functions, and when the discrete-log problem is easy in the plaintext group, then this implies IND-CCA encryption.

More importantly, we showed that any homomorphic encryption with a cyclic ciphertext space implies universal hash proof systems, and hence both Lossy Trapdoor Functions and IND-CCA secure encryption.

Homomorphic encryption schemes arise naturally in many contexts, where the security rests on a computational hardness assumption about groups. This makes homomorphic encryption a natural candidate for creating more complex cryptographic primitives.

Our constructions of IND-CCA secure cryptosystems from homomorphic encryption over a cyclic space are efficient, and have the benefit of simple proofs of security. Our results extend what is known to follow from homomorphic encryption, and bring us one step closer to the long sought-after goal of a generic construction of IND-CCA secure encryption from any homomorphic cryptosystem.

References

[AKP10] Armknecht, F., Katzenbeisser, S., Peter, A.: Group homomorphic encryption. IACR ePrint Archive 2010/501 (2010)

[BCHK07] Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM J. Comput. 36(5), 1301–1328 (2007)

[Ben94] Benaloh, J.C.: Dense probabilistic encryption. In: Proceedings of the Workshop on Selected Areas in Cryptography, pp. 120–128 (1994)

[BFOR08] Bellare, M., Fischlin, M., O'Neill, A., Ristenpart, T.: Deterministic Encryption: Definitional Equivalences and Constructions without Random Oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 360–378. Springer, Heidelberg (2008)

[BGN05] Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF Formulas on Ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)

[BK05] Boneh, D., Katz, J.: Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity-Based Encryption. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 87–103. Springer, Heidelberg (2005)

[CHK04] Canetti, R., Halevi, S., Katz, J.: Chosen-Ciphertext Security from Identity-Based Encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)

[CHK10] Cramer, R., Hofheinz, D., Kiltz, E.: A Twist on the Naor-Yung Paradigm and Its Application to Efficient CCA-Secure Encryption from Hard Search Problems. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 146–164. Springer, Heidelberg (2010)

[CMO00] Di Crescenzo, G., Malkin, T., Ostrovsky, R.: Single Database Private Information Retrieval Implies Oblivious Transfer. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 122–138. Springer, Heidelberg (2000)

[CS98] Cramer, R., Shoup, V.: A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)

[CS02] Cramer, R., Shoup, V.: Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002); Full version available at http://eprint.iacr.org Cryptology ePrint Archive, Report 2001/085

[DDN91] Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography. In: STOC 1991, pp. 542–552 (1991)

[DJ01] Damgård, I., Jurik, M.: A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-key System. In: Kim, K.-C. (ed.) PKC 2001. LNCS, vol. 1992, pp. 119–136. Springer, Heidelberg (2001)

[DMQN09] Dowsley, R., Müller-Quade, J., Nascimento, A.C.A.: A CCA2 Secure Public Key Encryption Scheme Based on the McEliece Assumptions in the Standard Model. In: Fischlin, M. (ed.) CT-RSA 2009. LNCS, vol. 5473, pp. 240–251. Springer, Heidelberg (2009)

[FGK+10] Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More Constructions of Lossy and Correlation-Secure Trapdoor Functions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 279–295. Springer, Heidelberg (2010)

[Gam85] El Gamal, T.: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (1985)

[Gen09] Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC 2009, pp. 169–178. ACM, New York (2009)

[GM84] Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28, 270–299 (1984)

[GMM07] Gertner, Y., Malkin, T., Myers, S.: Towards a Separation of Semantic and CCA Security for Public Key Encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 434–455. Springer, Heidelberg (2007)

[HJKS10] Haralambiev, K., Jager, T., Kiltz, E., Shoup, V.: Simple and Efficient Public-Key Encryption from Computational Diffie-Hellman in the Standard Model. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 1–18. Springer, Heidelberg (2010)

[HLOV11] Hemenway, B., Libert, B., Ostrovsky, R., Vergnaud, D.: Lossy Encryption: Constructions from General Assumptions and Efficient Selective Opening Chosen Ciphertext Security. In: Lee, D.H. (ed.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 70–88. Springer, Heidelberg (2011)

[HO12] Hemenway, B., Ostrovsky, R.: Extended-DDH and Lossy Trapdoor Functions. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 627–643. Springer, Heidelberg (2012)

[IKO05] Ishai, Y., Kushilevitz, E., Ostrovsky, R.: Sufficient Conditions for Collision-Resistant Hashing. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 445–456. Springer, Heidelberg (2005)

[KO97] Kushilevitz, E., Ostrovsky, R.: Replication is not needed: Single database, computationally-private information retrieval. In: FOCS 1997, pp. 364–373. ACM, New York (1997)

[Man98] Mann, E.: Private access to distributed information. Master's thesis, Technion - Israel Institute of Technology (1998)

[MY09] Mol, P., Yilek, S.: Chosen-Ciphertext Security from Slightly Lossy Trapdoor Functions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 296–311. Springer, Heidelberg (2010)

[NS98] Naccache, D., Stern, J.: A new public key cryptosystem based on higher residues. In: CCS 1998: Proceedings of the 5th ACM Conference on Computer and Communications Security, pp. 59–66. ACM Press, New York (1998)

[NY90] Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: STOC 1990, pp. 427–437 (1990)

[OU98] Okamoto, T., Uchiyama, S.: A New Public-Key Cryptosystem as Secure as Factoring. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 308–318. Springer, Heidelberg (1998)

[Pai99] Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)

[Pei09] Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: STOC 2009: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, pp. 333–342. ACM, New York (2009)

[PW08] Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: STOC 2008: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, pp. 187–196. ACM, New York (2008)

[Reg05] Regev, O.: On lattices, learning with errors, random linear codes and cryptography. In: STOC 2005, pp. 84–93. ACM (2005)

[RS91] Rackoff, C., Simon, D.R.: Non-interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)

[RS08] Rosen, A., Segev, G.: Efficient lossy trapdoor functions based on the composite residuosity assumption. Cryptology ePrint Archive, Report 2008/134 (2008)

[RS09] Rosen, A., Segev, G.: Chosen-Ciphertext Security via Correlated Products. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 419–436. Springer, Heidelberg (2009)

Waters Signatures with Optimal Security Reduction

Dennis Hofheinz[1], Tibor Jager[1], and Edward Knapp[2]

[1] Institut für Kryptographie und Sicherheit, Karlsruhe Institute of Technology, Germany
{dennis.hofheinz,tibor.jager}@kit.edu
[2] Department of Combinatorics and Optimization, University of Waterloo
eknapp@uwaterloo.ca

Abstract. Waters signatures (Eurocrypt 2005) can be shown existentially unforgeable under chosen-message attacks under the assumption that the computational Diffie-Hellman problem in the underlying (pairing-friendly) group is hard. The corresponding security proof has a reduction loss of $O(\ell \cdot q)$, where ℓ is the bitlength of messages, and q is the number of adversarial signature queries. The original reduction could meanwhile be improved to $O(\sqrt{\ell} \cdot q)$ (Hofheinz and Kiltz, Crypto 2008); however, it is currently unknown whether a better reduction exists. We answer this question as follows:

(a) We give a simple modification of Waters signatures, where messages are encoded such that each two encoded messages have a suitably large Hamming distance. Somewhat surprisingly, this simple modification suffices to prove security under the CDH assumption with a reduction loss of $O(q)$.
(b) We also show that any black-box security proof for a signature scheme with *re-randomizable* signatures must have a reduction loss of at least $\Omega(q)$, or the underlying hardness assumption is false. Since both Waters signatures and our variant from (a) are re-randomizable, this proves our reduction from (a) optimal up to a constant factor.

Understanding and optimizing the security loss of a cryptosystem is important to derive concrete parameters, such as the size of the underlying group. We provide a complete picture for Waters-like signatures: there is an inherent lower bound for the security loss, and we show how to achieve it.

Keywords: Digital signatures, Waters signatures, provable security, black-box reductions.

1 Introduction

Waters Signatures. Waters signatures [22] form a simple and efficient digital signature scheme in pairing-friendly groups. The existential unforgeability of the scheme can be proved under the computational Diffie-Hellman (CDH) assumption. Unfortunately, the corresponding security reduction from [22] suffers from

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 66–83, 2012.

a multiplicative loss of $O(\ell \cdot q)$, where ℓ is the bitlength of signed messages, and q is the number of adversarial signing queries. In other words, every signature forger with success probability ε can only be mapped to a CDH-solver with success probability $\Omega(\varepsilon/(\ell \cdot q))$.

From the proof of [22], it is not immediately clear whether this comparatively large security gap is inherent or an artifact of the used proof technique. In fact, [13,14] used a rather different simulation setup to show the security of Waters signatures with a reduction loss of $O(\sqrt{\ell} \cdot q)$. However, it is not at all clear whether their reduction is optimal. There is no known lower bound on the reduction loss of Waters signatures.

Our Contributions. Our contributions revolve around the possibility of achieving a better security reduction for Waters (and similar) signatures. Concretely:

(a) We first give a simple modification of Waters signatures. Essentially, we simply encode each message before signing. This guarantees that any two (encoded) messages have a suitably large Hamming distance. Perhaps somewhat surprisingly, this trivial modification can be shown secure under the CDH assumption with a reduction loss of $O(q)$. The price to pay for this improved reduction is a constant-factor blowup (caused by the encoding) of the public key size and signature/verification times.

(b) Building on work of Coron [7], we proceed to show that any security proof for a signature scheme with re-randomizable signatures must have a reduction loss of at least $\Omega(q)$, or the underlying complexity assumption is false. Coron showed that statement for deterministic signature schemes. We extend the statement to schemes in which any signature can be publicly re-randomized. Since both Waters signatures and our variant from (a) are re-randomizable, this proves our reduction from (a) optimal up to a constant factor.

Of course, the *practical* impact of our results is somewhat limited. In fact, it is a bit disappointing that one can only save a reduction factor of $\sqrt{\ell}$ (compared to the proof of [13,14]), where ℓ itself is typically significantly smaller than the remaining reduction loss of $O(q)$. However, we stress that from a *conceptual* point of view, our results essentially give a complete picture: there is an inherent lower bound for the security loss of Waters-like signature schemes, *and* we show how to achieve this bound.

Other Related Work. There exist a number of tightly secure signature schemes, both with (e.g., [1,18]) and without random oracles (e.g., [3,5,9,16,20]). However, to the best of our knowledge, there is no standard-model signature scheme whose security could be *tightly* reduced to the CDH problem. In particular, the only known results about the reduction tightness of Waters (or similar) signatures are the discussed works [13,14,22]. We do mention that Guo et al. [11] give a variant of Waters signatures and claim that this variant suffers from a reduction loss of only $O(\ell)$. However, their security proof is subtly flawed [12], as we sketch briefly in Section 1.1. It is not clear if and how their argument can be fixed.

1.1 Technical Overview

Partitioning. In order to present our techniques, we briefly recall the "partitioning" proof strategy used in the context of signature schemes, e.g., by Coron [6] and Waters [22]. A "partitioning" proof simulation partitions the message space into two sets: those messages that can be signed during the simulation, and those that cannot. Let us call those messages "signable," resp. "unsignable." Any forged signature for an unsignable message can then be used to solve a computational problem (e.g., a CDH challenge). The simulation thus succeeds if (a) all adversarial signature queries correspond to signable messages, and (b) the forger forges a signature for an unsignable message. For simplicity, assume that each message is set up as signable with a certain probability p. Assume further that these probabilities are independent for different messages. Then, it is not hard to see that the probability that the simulation succeeds is $P := p^q \cdot (1-p)$, where q is the number of signature queries. This probability is maximized if we set p suitably in the order of $1 - 1/q$, in which case $P = O(1/q)$.

Coron's Results. Specifically, using a partitioning technique, the best we can hope for is a reduction with a loss of $O(q)$. In fact, Coron [6] shows how to achieve such a reduction for the RSA-FDH scheme in the (programmable) random oracle model. Furthermore, he shows that any reduction of a deterministic signature scheme must essentially be partitioning, and thus the loss of $O(q)$ is inherent. See also [17].

Waters Signatures. Waters [22] conducts a similar partitioning simulation in the standard model, for a particular CDH-based signature scheme. For this outline, we will only give a very abstract and idealized breakdown of his strategy. In his scheme, a message $m = (m_1, \ldots, m_\ell)$ to be signed selects group elements h_i (for i with $m_i = 1$) that determine an intermediate hash value

$$H(m) = h_0 \prod_{m_i=1} h_i.$$

Depending on $H(m)$, the simulation in the security proof will be able to either generate a signature for m, or use any forged signature for m to solve a given CDH-challenge. Concretely, each h_i is associated with an (information-theoretically hidden) integer a_i. A message m in turn leads to an integer $a(m) := a_0 + \sum_{m_i=1} a_i$. If $a(m) \neq 0$, then the simulation can sign m; if $a(m) = 0$, then the simulation can use any forged signature for m to solve a CDH-challenge.

The Programming of the Hash Function. Unfortunately, neither the messages that need to be signed, nor the message on which the adversary forges are known in advance. Hence, the crux in the security analysis is to set up the values a_i such that with significant probability (say, P) over the a_i,

(a) all q adversarial signature queries $m^{(1)}, \ldots, m^{(q)}$ can be answered (that is, $a(m^{(i)}) \neq 0$ for all i), and
(b) the message m^* on which the adversary forges can be used to embed a challenge (i.e., $a(m^*) = 0$).

The probabilistic argument from [22] chooses the a_i uniformly over a suitable domain that depends on q. This results in a simulation success probability of $P = \Theta(1/(\ell \cdot q))$. Hofheinz and Kiltz [13,14] show that by setting up the a_i as suitably long random walks, the success probability can be improved to $P = \Theta(1/(\sqrt{\ell} \cdot q))$.

The Problem. The reason for the somewhat annoying ℓ, resp. $\sqrt{\ell}$ terms in these analyses is a bit subtle, and we will only try to give a brief idea here. Consider what happens when the forgery message m^* is "close" to a signed message m in the sense that m^* and $m^{(i)}$ differ in only one bit. Then, $a(m^*)$ and $a(m)$ differ by only one a_i. Now the analysis requires that the conditioned probability $\Pr[a(m) = 0 \mid a(m^*) = 0]$ is $O(1/q)$. (Otherwise, it becomes difficult to prove that the probability is significant that, say, q random messages m can all be signed, given that m^* cannot.) But since $a(m^*)$ and $a(m)$ differ by only one (a-priori unknown) a_i, *each* a_i must have a distribution with min-entropy at least $\log_2 q$. (That is, the probability that a_i takes a particular value must always be $O(1/q)$.) Hence, e.g., for the all-one message $m = (1, \ldots, 1)$, we get that $a(m) = a_0 + \sum_{i=1}^{\ell} a_i$, and we would expect that $a(m)$ has a much larger min-entropy than $\log_2 q$. (In particular, if m^* is the all-one message, then $\Pr[a(m^*) = 0]$ will be much smaller than $\Theta(1/q)$.)

Our Solution. Intuitively, our solution is simply to encode all messages using a code with large minimum distance prior to signing. This avoids that two messages m^*, m that are "close" even exist. Concretely, we will ensure that any two different $(a(m^*), a(m)$ will always differ by at least *a constant fraction* of all a_i. This allows to set up the a_i with lower min-entropy than in previous analyses, and allows us to set up a simulation with success probability $P = \Theta(1/q)$.

For completeness, we note that Guo et al. describe another way to set-up the a_i in the proof of [11, Theorem 2], and claim that this set-up can used to give a tighter security reduction for Waters signatures. However, it turns out that this is not true [12]. The reason is that in the proof of [11, Theorem 2] the simulation is set up in a way that depends on the messages to be signed. (Specifically, the variables that correspond to our a_i are not statistically hidden in [11].) Thus, the view of the adversary is not independent of the event that the simulation succeeds. Concretely, the setup in [11] potentially allows adversaries who forge only signatures for messages m^* with $a(m^*) \neq 0$, in which case no solution to the CDH problem can be extracted.

Optimality of Our Solution. Naturally, one may ask whether it is possible to improve the reduction further. We answer this question in the negative. Concretely, we show that it is impossible to prove any re-randomizable signature scheme secure, using a black-box reduction to any of a large class of hardness assumptions, such that the security loss in the reduction is significantly better than $1/q$. Since both Waters signatures and our new variant are efficiently re-randomizable, this shows our reduction optimal. We stress that our impossibility result does not cover

interactive assumptions (such as the LRSW assumption [19]). In particular, our result does not contradict re-randomizable signature schemes with tight security proofs based on interactive assumptions (such as [4]).

The proof technique is based on the meta-reduction technique of Coron [7], which simulates a forger for $\mathcal{R}$ such that the simulation fails with probability at most $1/q$. For Coron's proof it is essential that the considered signature scheme is deterministic, and that for all public keys it is publicly verifiable that there exists only a single valid signature per message (as it is the case for instance for certified trapdoor permutations, cf. [17]). Since we want to consider probabilistic schemes, we lose this leverage and Coron's result does not apply.

Instead, we will show that it suffices that signatures are re-randomizable. Moreover, since deterministic signature scehemes are re-randomizable, our result can be seen as a generalization of previous work [7,17].

Let us intuitively sketch the reason why re-randomizability suffices. Basically, if signatures are efficiently re-randomizable, then the only way left to prove security is to partition the message space into messages which can be signed by the reduction, and messages from which a solution to the given problem instance can be extracted. To see this, suppose that for a random message m^* it holds with high probability that the reduction can simulate one signature for m^*, but extract a solution to a hard problem from a different signature for m^*. Then the reduction could solve the hard problem even without interacting with the forger, by generating a simulated signature σ^* for m^*, re-randomizing it to obtain some random signature σ', and finally extracting the solution to the hard problem from σ'. Since the reduction would solve the problem without any additional assumption (i.e. the existence of a signature forger), this would contradict the assumption that the underlying problem is hard.

Further Applications. We note that the analysis from Section 4 can also be applied to show that a security reduction from any hard problem to breaking Waters' identity-based encryption (IBE) scheme from [22] must lose a factor of $\Omega(q)$, if the adversary may issue q adaptive chosen-identity key queries are allowed.

However, this bound is only achievable using our techniques if one wants to prove that Waters' IBE scheme is *one-way* under adaptive chosen-identity attacks. The commonly accepted security notion for IBE is *indistinguishability* under adaptive chosen-identity attacks, and it seems that in this setting our techniques do not substantially improve on the results of [2,22]. Therefore we do not elaborate this further.

1.2 Outline

We recall some notation, standard definitions, and Waters' signature scheme in Section 2. In Section 3, we present our modified signature scheme and prove it secure with a reduction loss of $O(q)$. Finally, in Section 4, we show a lower bound of $\Omega(q)$ on the reduction loss of schemes with re-randomizable signatures.

2 Preliminaries

For $k \in \mathbb{N}$, we write 1^k for the string of k ones, and $[k]$ for $\{1, \ldots, k\}$. Moreover, $|x|$ denotes the length of a bitstring x, while $|S|$ denotes the size of a set S. Further, $s \xleftarrow{\$} S$ denotes the sampling a uniformly random element s of S. For an algorithm $\mathcal{A}$, we write $z \xleftarrow{\$} \mathcal{A}(x, y, \ldots)$ to indicate that $\mathcal{A}$ is a (probabilistic) algorithm that outputs z on input $(x, y, \ldots)$.

2.1 Digital Signatures

A digital signature scheme $\mathsf{Sig} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vfy})$ consists of three algorithms. Key generation Gen generates a keypair $(pk, sk) \xleftarrow{\$} \mathsf{Gen}(1^k)$ for a secret signing key sk and a public verification key pk. The signing algorithm Sign inputs a message and the secret signing key, and returns a signature $\sigma \xleftarrow{\$} \mathsf{Sign}(sk, m)$ of the message. The verification algorithm Vfy takes a verification key and a message with corresponding signature as input, and returns $b \leftarrow \mathsf{Vfy}(pk, m, \sigma)$, where $b \in \{0, 1\}$. We say that a signature is *valid*, if $\mathsf{Vfy}(pk, m, \sigma) = 1$. We require the usual correctness properties.

Let us recall the *existential unforgeability against chosen message attacks* (EUF-CMA) security experiment [10], played between a challenger and a forger $\mathcal{F}$.

1. The challenger runs Gen to generate a keypair (pk, sk). The forger receives pk as input.
2. The forger may ask the challenger to sign a number of messages. To query the i-th signature, $\mathcal{F}$ submits a message $m^{(i)}$ to the challenger. The challenger returns a signature σ_i under sk for this message.
3. The forger outputs a message m^* and signature σ^*.

$\mathcal{F}$ wins the game, if $1 \leftarrow \mathsf{Vfy}(pk, m^*, \sigma^*)$, that is, σ^* is a valid signature for m^*, and $m^* \neq m^{(i)}$ for all i.

Definition 1. *We say that $\mathcal{F}$ (t, q, ϵ)-breaks the EUF-CMA security of Sig, if $\mathcal{F}$ runs in time t, makes at most q signing queries, and has success probability ϵ. Furthermore, we say that Sig is EUF-CMA secure if there is no PPT forger $\mathcal{F}$ that t, q, ϵ-breaks the EUF-CMA security of Sig for polynomials t, q and a non-negligible ϵ.*

2.2 The Computational Diffie-Hellman Problem

Let $\mathbb{G}$ be a group of order p. The computational Diffie-Hellman problem is to compute the group element $g^{\alpha\beta}$, given random group elements $(g, g^\alpha, g^\beta) \in \mathbb{G}^3$.

Definition 2. *We say that algorithm $\mathcal{A}$ (ϵ, t)-solves the computational Diffie-Hellman problem in $\mathbb{G}$, if*

$$\Pr\left[\mathcal{A}(g, g^\alpha, g^\beta) = g^{\alpha\beta}\right] \geq \epsilon,$$

and $\mathcal{A}$ runs in time t.

2.3 Waters Signatures

Recall Waters' signature scheme $\mathsf{Sig}_{\mathsf{Wat}} = (\mathsf{Gen}_{\mathsf{Wat}}, \mathsf{Sign}_{\mathsf{Wat}}, \mathsf{Vfy}_{\mathsf{Wat}})$ from [22]:

$\mathsf{Gen}_{\mathsf{Wat}}(1^k)$**:** The key generation algorithm selects a group $\mathbb{G}$ of prime order $p \approx 2^{2k}$ with generator g and bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Then $h_0, h_1, \ldots, h_\ell \xleftarrow{\$} \mathbb{G}$ and $\alpha, \beta \xleftarrow{\$} \mathbb{Z}_p$ are chosen at random. The public key is defined as

$$pk := (\mathbb{G}, g, g^\alpha, g^\beta, h_0, h_1, \ldots, h_\ell),$$

and the secret key is $sk := (pk, g^{\alpha\beta})$.

In the sequel we will denote with $H : \{0,1\}^\ell \to \mathbb{G}$ the function mapping $m \mapsto h_0 \prod_{i=1}^{\ell} h_i^{m_i}$, where for $i \in [\ell]$, we denote by $m_i \in \{0,1\}$ the ith bit of m.

$\mathsf{Sign}_{\mathsf{Wat}}(sk, m)$**:** The signing algorithm takes as input a message $m \in \{0,1\}^\ell$. The algorithm samples $r \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$\sigma_1 = g^r \qquad \text{and} \qquad \sigma_2 = g^{\alpha\beta} H(m)^r.$$

Then it returns the signature $\sigma = (\sigma_1, \sigma_2)$.

$\mathsf{Vfy}_{\mathsf{Wat}}(pk, m, \sigma)$**:** The verification algorithm returns 1 if the equation

$$e(g^\alpha, g^\beta) \cdot e(\sigma_1, H(m)) = e(g, \sigma_2)$$

holds. Otherwise 0 is returned.

Waters [22] proved that the above signature scheme is EUF-CMA secure under the computational Diffie-Hellman assumption in $\mathbb{G}$. The original reduction from [22] is not very tight. Concretely, it loses a factor of $(16(\ell+1)q)$, where ℓ is the bit-length of the message and q is (an upper bound on) the number of signature queries issued by the forger. The original analysis was slightly improved in [13], which gives the following theorem.

Theorem 1 ([22,13]). *Suppose there exists a forger $\mathcal{F}$ that (t, q, ϵ)-breaks the EUF-CMA security of $\mathsf{Sig}_{\mathsf{Wat}}$. Then there exists an algorithm $\mathcal{A}$ (ϵ', t')-solving the computational Diffie-Hellman problem in $\mathbb{G}$ in time $t' \approx t$ with success probability $\epsilon' \geq \epsilon \cdot O(\frac{1}{\sqrt{\ell q}})$.*

3 A Variant of Waters' Signature Scheme

As mentioned in Section 1, our only modification of Waters' scheme will be to encode messages prior to signing. For each security parameter k, we will therefore assume a code $\mathcal{C} = \mathcal{C}_k$ over $\mathbb{F}_2$ of dimension k, length ℓ, and minimum distance $d \geq \gamma \cdot \ell$ for a fixed $\gamma > 0$. (For instance, one can use a family of expander codes with suitable parameters [21,23].)

We will apply $\mathcal{C}$ to k-bit messages, and we assume that each encoded message has Hamming weight at least d. (For instance, one could simply forbid any message that leads to an all-zero output.)

Our scheme $\mathsf{Sig}_{\mathsf{tight}} = (\mathsf{Gen}_{\mathsf{tight}}, \mathsf{Sign}_{\mathsf{tight}}, \mathsf{Vfy}_{\mathsf{tight}})$ is almost identical to the one by Waters (see Section 2.3):

$\mathsf{Gen}_{\mathsf{tight}}(1^k)$ outputs $pk := (\mathbb{G}, g, g^\alpha, g^\beta, h_1, \ldots, h_\ell)$ and $sk := (pk, g^{\alpha\beta})$ just like $\mathsf{Gen}_{\mathsf{Wat}}$, but without h_0. Now pk defines a hash function $H(M) := h_0 \prod_i h_i^{M_i}$ for $M = (M_1, \ldots, M_\ell) \in \{0,1\}^\ell$.

$\mathsf{Sign}_{\mathsf{tight}}(sk, m)$ (for $m \in \{0,1\}^k$) first computes $M := \mathcal{C}_k(m) \in \{0,1\}^\ell$ and then outputs $\sigma := (\sigma_1, \sigma_2) := (g^r, g^{\alpha\beta} H(M)^r)$.

$\mathsf{Vfy}_{\mathsf{tight}}(pk, m, \sigma)$ sets $M := \mathcal{C}_k(m)$ and then checks

$$e(g^\alpha, g^\beta) \cdot e(\sigma_1, H(M)) \stackrel{?}{=} e(g, \sigma_2).$$

Obviously, this defines a signature scheme. We also claim:

Theorem 2. *Suppose there exists a forger $\mathcal{F}$ that (t, q, ϵ)-breaks the EUF-CMA security of $\mathsf{Sig}_{\mathsf{tight}}$. Then there exists an algorithm $\mathcal{A}$ (ϵ', t')-solving the computational Diffie-Hellman problem in $\mathbb{G}$ in time $t' \approx t$ with success probability $\epsilon' \geq \epsilon \cdot \Theta(\frac{1}{q})$.*

The rest of this section will be devoted to proving Theorem 2.

3.1 A Better Bound on the Success Probability of the Simulation

We start with our abstract setup and the analysis of the crucial variables a_i for our simulation. In the next subsection, we then proceed to outline how this setup is embedded in a simulation of the signature scheme.

In the following let $\ell, w \in \mathbb{N}$. In the simulation, ℓ will be the bitlength of (encoded) messages, and w will be an integer that determines how long each random walk a_i will be. For $i \in [\ell], j \in [w]$, let $a_{i,j}$ be independently and uniformly distributed random variables over $\{-1, 0, 1\}$. Let $a_i := \sum_{j=1}^{w} a_{i,j}$. Furthermore, for $S \subseteq [\ell]$, let $a(S) := \sum_{i \in S} a_i$. Note that $a(S)$ is a random walk (with $\{-1, 0, 1\}$-steps) of length $|S| \cdot w$. Hence, the following standard result about random walks applies:

Theorem 3. *There exist $\lambda, \Lambda \in \mathbb{R}$ that do not depend on ℓ, w, such that for any $S \subseteq [\ell]$ of size $s := |S|$, we have*

$$\frac{\lambda}{\sqrt{s \cdot w}} \quad \leq \quad \Pr[a(S) = 0] \quad \leq \quad \frac{\Lambda}{\sqrt{s \cdot w}}.$$

Furthermore, for any ℓ, w, S, the probability $\Pr[a(S) = i]$ is maximized for $i = 0$.

Proof. Although this is a standard fact about random walks (see [8,15] for a thorough introduction), [14, Theorems 17 and 18] provide a direct proof of the theorem adjusted to our setting.

We can now use Theorem 3 to derive the main technical lemma for the analysis of our variant of Waters' signature scheme. This result uses and extends techniques of [13,14] to a setting in which there is a guaranteed "minimum distance" between two random walks. (Later, this "minimum distance" will correspond to the Hamming distance between two encoded messages to be signed.)

Lemma 1. *Let $X, Y \subseteq [\ell]$ such that $|X|, |Y| \geq d$, and $|(X \setminus Y) \cup (Y \setminus X)| \geq d$ for $d \geq 1$. Then, we have*

$$\Pr[a(Y) = 0 \mid a(X) = 0] \quad \leq \quad C \cdot \frac{\sqrt{\ell}}{d \cdot \sqrt{w}} \tag{1}$$

for a fixed constant C that does not depend on ℓ, w, d, X, Y.

Proof. We distinguish the two cases $|X \setminus Y| \geq d/2$ and $|Y \setminus X| \geq d/2$:
Case $|Y \setminus X| \geq d/2$:

$$\begin{aligned}
\Pr[a(Y) = 0 \mid a(X) = 0] &\overset{(a)}{\leq} \max_i \Pr[a(Y) = 0 \mid a(Y \cap X) = i] \\
&\overset{(b)}{\leq} \max_i \Pr[a(Y \setminus X) = -i \mid a(Y \cap X) = i] \\
&\overset{(c)}{=} \max_i \Pr[a(Y \setminus X) = -i] \\
&\overset{(d)}{=} \Pr[a(Y \setminus X) = 0] \\
&\overset{(e)}{\leq} \frac{\sqrt{2} \cdot \Lambda}{\sqrt{d \cdot w}} \overset{(f)}{\leq} \sqrt{2} \cdot \Lambda \cdot \frac{\sqrt{\ell}}{d \cdot \sqrt{w}}
\end{aligned}$$

Here, (a) holds because $a(Y)$ only depends on $a(Y \cap X)$ but not on $a(X \setminus Y)$; (b) uses $a(Y) = a(Y \setminus X) + a(Y \cap X)$; (c) uses that $a(Y \setminus X)$ and $a(Y \cap X)$ are independent; (d) and (e) apply Theorem 3, using that $a(Y \setminus X)$ is a random walk of length at least $(d/2) \cdot w$; finally, (f) uses $d \leq \ell$.
Case $|X \setminus Y| \geq d/2$:

$$\begin{aligned}
\Pr[a(Y) = 0 \mid a(X) = 0] &\overset{(a)}{=} \Pr[a(X) = 0 \mid a(Y) = 0] \cdot \frac{\Pr[a(Y) = 0]}{\Pr[a(X) = 0]} \\
&\overset{(b)}{\leq} \frac{\sqrt{2} \cdot \Lambda}{\sqrt{d \cdot w}} \cdot \frac{\Pr[a(Y) = 0]}{\Pr[a(X) = 0]} \\
&\overset{(c)}{\leq} \frac{\sqrt{2} \cdot \Lambda}{\sqrt{d \cdot w}} \cdot \frac{\sqrt{2} \cdot \Lambda}{\sqrt{d \cdot w}} \cdot \frac{\sqrt{\ell \cdot w}}{\sqrt{2} \cdot \lambda} \\
&= \frac{\sqrt{2} \cdot \Lambda^2}{\lambda} \cdot \frac{\sqrt{\ell}}{d \cdot \sqrt{w}}
\end{aligned}$$

Here, (a) uses Bayes' theorem; (b) uses what we have proved for the case $|Y \setminus X| \geq d/2$ (with swapped X, Y); (c) apply Theorem 3, using that $a(X)$ and $a(Y)$ are random walks of length at least $d \cdot w$ and at most $\ell \cdot w$.
Since we have $\Lambda \geq \lambda$, setting $C := \sqrt{2} \cdot (\Lambda^2/\lambda)$ proves (1).

Next, we can plug Lemma 1 into the existing analysis of Waters' scheme [22]. First, this means proving the following technical claim, which essentially bounds the probability that all signing queries can be answered, while the adversary's forgery solves a computational challenge. This claim roughly corresponds to [22, Claim 2].

Lemma 2. *Let $X, Y_1, \ldots, Y_q \subseteq [\ell]$ such that $|X|, |Y_i| \geq d$ and $|(X \setminus Y_i) \cup (Y_i \setminus X)| \geq d$ for some $d \geq 1$ and all i. Then, we have*

$$\Pr[a(X) = 0 \wedge \forall i \in [q] : a(Y_i) \neq 0] \quad \geq \quad \left(1 - C \cdot q \cdot \frac{\sqrt{\ell}}{d \cdot \sqrt{w}}\right) \cdot \frac{D}{\sqrt{d \cdot w}} \tag{2}$$

for fixed constants C, D that do not depend on ℓ, w, d, q, X, and the Y_i.

Proof. We have

$$\begin{aligned}
\Pr[a(X) = 0 \wedge \forall i : a(Y_i) \neq 0] &= \Pr[\forall i : a(Y_i) \neq 0 \mid a(X) = 0] \cdot \Pr[a(X) = 0] \\
&\overset{(a)}{\geq} \Pr[\forall i : a(Y_i) \neq 0 \mid a(X) = 0] \cdot \frac{\lambda}{\sqrt{d \cdot w}} \\
&= (1 - \Pr[\exists i : a(Y_i) = 0 \mid a(X) = 0]) \cdot \frac{\lambda}{\sqrt{d \cdot w}} \\
&\overset{(b)}{\geq} \left(1 - \sum_{i=1}^{q} \Pr[a(Y_i) = 0 \mid a(X) = 0]\right) \cdot \frac{\lambda}{\sqrt{d \cdot w}} \\
&\overset{(c)}{\geq} \left(1 - q \cdot C \cdot \frac{\sqrt{\ell}}{d \cdot \sqrt{w}}\right) \cdot \frac{\lambda}{\sqrt{d \cdot w}}
\end{aligned}$$

Here, (a) applies Theorem 3, using that $a(X)$ is a random walk of length at least $d \cdot w$; (b) uses a union bound; (c) denotes a q-wise application of Lemma 1. Setting $D := \lambda$ yields (2).

Note that if we set $d = \gamma \cdot \ell$ and $w = (2Cq/\gamma)^2/\ell$ (for some $\gamma > 0$) in (2), a quick calculation gives

$$\Pr[a(X) = 0 \wedge \forall i \in [q] : a(Y_i) \neq 0] \quad \geq \quad \frac{D\sqrt{\gamma}}{4C} \cdot \frac{1}{q}. \tag{3}$$

Hence, if γ is a constant, then this probability lies in the order of $1/q$.

3.2 The Full Simulation

We now briefly sketch how to use Lemma 2 to prove Theorem 2. We are very brief because except for Lemma 2 and a few syntactic differences, the proof is identical to the one from [22].

Our goal is to build a CDH adversary $\mathcal{A}$ from an EUF-CMA forger $\mathcal{F}$ on $\mathsf{Sig}_{\mathsf{tight}}$ that makes at most $q = q(k)$ signature queries. Our CDH adversary $\mathcal{A}$ gets as input a CDH challenge (g, g^α, g^β) for a group $\mathbb{G}$ of order p with pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, and is supposed to output $g^{\alpha\beta}$.

Public Key. The first task of $\mathcal{A}$ is to prepare a $\mathsf{Sig}_{\mathsf{tight}}$ public key for $\mathcal{F}$. In order to do so, $\mathcal{A}$ sets $w = (2Cq/\gamma)^2/\ell$ for the parameter γ of the code $\mathcal{C}$, and the constant C from Section 3.1. Then, $\mathcal{A}$ prepares random variables $a_1, \ldots, a_\ell$

as random walks (over $\{-1, 0, 1\}$) of length w, just as in Section 3.1. Finally, $\mathcal{A}$ chooses uniformly blinding exponents $b_1, \ldots, b_\ell \leftarrow [p]$ and sets

$$h_i := (g^\alpha)^{a_i}\, g^{b_i} \quad (\text{for } i = 1, \ldots, \ell)$$
$$pk := (\mathbb{G}, g, g^\alpha, g^\beta, h_1, \ldots, h_\ell).$$

This results in a public key that is distributed exactly as in $\mathsf{Sig}_{\mathsf{tight}}$.

Signing Queries. Next, $\mathcal{A}$ runs $\mathcal{F}$ on pk, and answers $\mathcal{F}$'s signing queries as follows. Suppose $\mathcal{F}$ asks for the signature of a message $m \in \{0,1\}^k$ that induces an encoded message $M = \mathcal{C}(m) \in \{0,1\}^\ell$. Let us view M as a subset of $[n]$, such that $i \in M$ iff the i-th bit of M is set. Write $a(M) := \sum_{i \in M} a_i$ and $b(M) := \sum_{i \in M} b_i$. Note that we can always write $H(M) = (g^\alpha)^{a(M)} g^{b(M)}$. Hence, valid signatures have the form

$$(g^r, g^{\alpha\beta} \cdot H(M)^r) \quad = \quad (g^r, g^{\alpha\beta + r\cdot(\alpha\cdot a(M) + b(M))})$$

In particular, if we set $g^r = \left(g^\beta\right)^x g^y$, then valid signatures are of the form

$$\begin{aligned} &(g^{x\beta+y}, g^{\alpha\beta+(x\beta+y)\cdot(\alpha\cdot a(M)+b(M))}) \\ &= \left(\left(g^\beta\right)^x g^y, \left(g^{\alpha\beta}\right)^{1+x\cdot a(M)} (g^\alpha)^{y\cdot a(M)} \left(g^\beta\right)^{x\cdot b(M)} g^{y\cdot b(M)}\right). \end{aligned} \tag{4}$$

Thus, depending on $a(M)$, we now distinguish two cases:

- if $a(M) \neq 0$, then the simulation can generate properly distributed valid signatures via (4) by setting $x = -a(M)^{-1} \bmod p$ and choosing y uniformly (notice that the $g^{\alpha\beta}$ term in (4) then vanishes);
- if $a(M) = 0$, then the simulation cannot generate a signature for m, and the simulation fails.

Extraction. Suppose that eventually, $\mathcal{F}$ generates a valid forged signature σ^* for a fresh message m^* with associated encoding $M^* := \mathcal{C}(m^*)$. Again, we can distinguish two cases:

- if $a(M^*) = 0$, then the simulation can extract $g^{\alpha\beta}$ by using

$$\begin{aligned} \sigma^* \;=\; (g^{r^*}, g^{\alpha\beta} \cdot H(M^*)^{r^*}) \;&=\; \left(g^{r^*}, g^{\alpha\beta} \cdot \left(g^{b(M^*)}\right)^{r^*}\right) \\ &= \left(g^{r^*}, g^{\alpha\beta} \cdot \left(g^{r^*}\right)^{b(M^*)}\right) \end{aligned}$$

  for some unknown r^* but known $b(M^*)$;
- if $a(M^*) \neq 0$, then the extraction fails.

Simulation Success. Let `fail` denote the event that the simulation fails (either because $a(M_i) = 0$ for a signature query, or because $a(M^*) \neq 0$). Then Lemma 2 immediately gives an upper bound of $1 - \Theta(1/q)$ on $\Pr[\texttt{fail}]$. Indeed, if we set $X := M^*$ and $Y_i := M_i$, then any two different encoded messages differ in at least $d = \gamma \cdot \ell$ bits. In particular, $|(X \setminus Y_i) \cup (Y_i \setminus X)| \geq d$. Substituting

$d = \gamma \cdot \ell$ and $w = (2Cq/\gamma)^2/\ell$ in (2) yields (3), and thus a lower bound of $\Theta(1/q)$ on $\neg\texttt{fail}$. Furthermore, the a_i are information-theoretically hidden from $\mathcal{F}$, so conditioning on $\neg\texttt{fail}$ does not change $\mathcal{F}$'s success in the EUF-CMA experiment. Theorem 2 follows.

4 Lower Tightness Bounds for Re-randomizable Signatures

In this section we show that it is impossible to prove security of a signature scheme with significantly smaller security loss than $\Omega(q)$, if the signature scheme is efficiently re-randomizable. To this end, we first define re-randomizable signatures. Then we give abstract definitions of computational problems, and reductions that reduce solving a given computational problem to breaking the security of a given signature scheme. All these results are generic, in the sense that they apply to any re-randomizable signature scheme. Finally, we show that both Waters' signature scheme from [22] and our modified scheme from Section 3 are efficiently re-randomizable, which implies that the reduction from Section 3 is optimal.

4.1 Re-randomizable Signatures

The intuition behind re-randomizable signatures is the property that, given only the public key pk and a valid signature σ for some message m, one can efficiently generate a new signature σ' that is distributed uniformly over the set of all possible signatures for m.

Let $\mathsf{Sig} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vfy})$ be a signature scheme. For any string pk (which may either be a honestly generated public key, or a fake public key generated by a simulator in a security proof) let us denote with

$$\Sigma(pk, m) = \{\sigma : \mathsf{Vfy}(pk, m, \sigma) = 1\}$$

the set of signatures σ for message m that verify correctly under public key pk.

Definition 3. *We say that* Sig *is* t-re-randomizable*, if there exists an algorithm* ReRand *running in time at most* t*, such that for all* (pk, m, σ) *with* $\mathsf{Vfy}(pk, m, \sigma) = 1$ *holds that the output distribution of*

$$\mathsf{ReRand}(pk, m, \sigma)$$

is identical to the uniform distribution over $\Sigma(pk, m)$*.*

4.2 Computational Problems and Reductions

The definitions in this section follow [7].

Definition 4. *A* computational problem $\Pi = (C, \mathcal{S})$ *consists of a set* C *and a family of sets* $\mathcal{S} = (S_c)_{c \in C}$. *We say that* C *is the set of* challenges *of* Π, *and for each* $c \in C$ *set* S_c *is the set of* solutions *for* c. *We say that an algorithm* $\mathcal{A}$ $(\epsilon_{\mathcal{A}}, t_{\mathcal{A}})$-solves Π, *if* $\mathcal{A}$ *runs in time* $t_{\mathcal{A}}$ *and*

$$\Pr[\mathcal{A}(c) \in S_c : c \xleftarrow{\$} C] \quad \geq \quad \epsilon_{\mathcal{A}}.$$

As an example consider a group $\mathbb{G}$ of prime order p. Then the computational Diffie-Hellman problem in $\mathbb{G}$ is the problem $\Pi = (C, \mathcal{S})$ with $C = \mathbb{G} \times \mathbb{G} \times \mathbb{G}$ and where for each $c = (g, g^a, g^b) \in C$ we have $S_c = \{g^{ab}\}$.

Definition 5. $\mathcal{R}$ *is a* $(t_{\mathcal{F}}, \epsilon_{\mathcal{F}}, q, \epsilon_{\mathcal{R}}, t_{\mathcal{R}})$-reduction *from problem* Π *to breaking the security of signature scheme* Sig, *if for* any *forger* $\mathcal{F}$ *that* $(t_{\mathcal{F}}, \epsilon_{\mathcal{F}}, q)$-*breaks the EUF-CMA security of* Sig *in the sense of Definition 1, algorithm* $\mathcal{R}$ $(\epsilon_{\mathcal{R}}, t_{\mathcal{R}})$-*solves* Π.

Note that we require that the $\mathcal{R}$ works for any forger $\mathcal{F}$, in particular if $\mathcal{F}$ is given as a black-box.

For instance, Section 3 gives an example for an algorithm $\mathcal{R}$ that $(t_{\mathcal{F}}, \epsilon_{\mathcal{F}}, q, \epsilon_{\mathcal{F}} \cdot \Theta(1/q), t_{\mathcal{R}})$-reduces solving the computational Diffie-Hellman problem to breaking the security of Waters' signature scheme with $t_{\mathcal{R}} \approx t_{\mathcal{F}}$.

4.3 Lower Tightness Bound for Re-randomizable Signature Schemes

In this section we consider reductions that run Forger $\mathcal{F}$ only once, and show that any such reduction loses a factor of at least q. A generalization to reductions that run $\mathcal{F}$ repeatedly is straightforward, see Section 4.4.

Theorem 4. *Let* Sig *be a* t_{ReRand}-*re-randomizable signature scheme and let* Π *be a computational problem in the sense of Definition 4. If there exists an* $(t_{\mathcal{F}}, \epsilon_{\mathcal{F}}, q, \epsilon_{\mathcal{R}}, t_{\mathcal{R}})$-*reduction* $\mathcal{R}$ *that runs* $\mathcal{F}$ *once and reduces* Π *to breaking* Sig, *then there exists an algorithm* $\mathcal{A}$ *that* $(\epsilon_{\mathcal{A}}, t_{\mathcal{A}})$-*solves* Π *with* $t_{\mathcal{A}} \approx 2t_{\mathcal{R}} + t_{\mathsf{ReRand}}$ *and*

$$\epsilon_{\mathcal{A}} \geq \epsilon_{\mathcal{R}} - \exp(-1)/q.$$

We will use the following lemma, which is due to Coron [7].

Lemma 3. *Let* $\mathcal{M}$ *be a set and let* Q *be a set of sequences of at most* q *elements of* $\mathcal{M}$, *such that for any sequence* $(m_1, \ldots, m_j) \in Q$ *we have* $(m_1, \ldots, m_{j-1}) \in Q$. *Let* $i \xleftarrow{\$} [q]$ *and* $(m_1, \ldots, m_q, m^*) \xleftarrow{\$} \mathcal{M}^{q+1}$ *be uniformly random. Then*

$$\Pr\left[(m_1, \ldots, m_q) \in Q \wedge (m_1, \ldots, m_{i-1}, m^*) \notin Q\right] \leq \exp(-1)/q.$$

See [7, Appendix D] for the proof.

Proof (Proof of Theorem 4). Consider an (imaginary) forger $\mathcal{F}$ that $(t_{\mathcal{F}}, \epsilon_{\mathcal{F}}, q)$-breaks the EUF-CMA security of a given signature scheme Sig with some success probability $\epsilon_{\mathcal{F}}$ in some time $t_{\mathcal{F}}$. Forger $\mathcal{F}$ works as follows.

1. $\mathcal{F}$ receives as input a public key pk from the challenger.
2. It selects $q+1$ random pairwise different messages $(m^{(1)}, \ldots, m^{(q)}, m^*)$ from the message space of Sig.
3. Then $\mathcal{F}$ queries the challenger for signatures of messages $(m^{(1)}, \ldots, m^{(q)})$.
4. $\mathcal{F}$ computes a valid signature σ^* for message m^*, such that σ^* is distributed uniformly over $\Sigma(pk, m^*)$. (Forger $\mathcal{F}$ may be inefficient, since it needs to forge a signature. However, we will later show how to simulate $\mathcal{F}$ efficiently.)
5. Finally $\mathcal{F}$ tosses a (biased) coin $b \xleftarrow{\$} \{0,1\}$ with $\Pr[b=1] = \epsilon_{\mathcal{F}}$.
   (a) If $b = 1$ then it outputs σ^*.
   (b) Otherwise it outputs error symbol $\perp$.

Note that any $(t_{\mathcal{F}}, \epsilon_{\mathcal{F}}, q, \epsilon_{\mathcal{R}}, t_{\mathcal{R}})$-reduction from some computational problem Π to breaking the security of Sig can use Forger $\mathcal{F}$ to $(\epsilon_{\mathcal{R}}, t_{\mathcal{R}})$-solve Π. In the sequel we will apply the rewinding technique of Coron [7] to show how to simulate $\mathcal{F}$, if Sig is re-randomizable.

Consider an algorithm $\mathcal{A}$ that uses $\mathcal{R}$ as follows.

1. $\mathcal{A}$ receives as input an instance c of Π, and starts $\mathcal{R}$ on input c.
2. $\mathcal{R}$ outputs a public key pk.
3. $\mathcal{A}$ selects a random integer $i \in [q]$ and $q+1$ random pairwise different messages $(m^{(1)}, \ldots, m^{(q)}, m^*)$.
4. It queries $\mathcal{R}$ for a signature for each message in $M_0 = (m^{(1)}, \ldots, m^{(i-1)}, m^*)$. If $\mathcal{R}$ aborts, then so does $\mathcal{A}$.
5. Then $\mathcal{A}$ rewinds $\mathcal{R}$ to the state after it has output the public key (i.e. the state after Step 2).
6. Now $\mathcal{A}$ queries a signature for each message in $M_1 = (m^{(1)}, \ldots, m^{(q)})$. Again, if $\mathcal{R}$ aborts, then $\mathcal{A}$ aborts too.
7. Then $\mathcal{A}$ computes $\sigma' = \mathsf{ReRand}(pk, m^*, \sigma^*)$ and tosses a coin $b \xleftarrow{\$} \{0,1\}$ with $\Pr[b=1] = \epsilon_{\mathcal{F}}$.
   (a) If $b = 1$ then it submits σ' to $\mathcal{R}$.
   (b) Otherwise it submits error symbol $\perp$.
   Finally $\mathcal{A}$ returns outputs whatever $\mathcal{R}$ returns

Fix the internal coins of $\mathcal{R}$, and let Q be the set of (ordered) message sequences M of size at most q, such that $\mathcal{R}$ aborts when asked to sign the messages in M. Let $\mathcal{E}$ denote the event that $M_0 \notin Q$ and $M_1 \in Q$. (In other words, $\mathcal{E}$ occurs when $\mathcal{R}$ does not abort before the rewinding, but does abort after the rewinding by $\mathcal{A}$.) Note that, due to the re-randomizability of Sig, $\mathcal{A}$ outputs a uniformly random signature σ' from the set $\Sigma(pk, m^*)$ of all valid signatures for m^* and public key pk. Therefore $\mathcal{A}$ simulates $\mathcal{F}$ perfectly (after the rewind), and thus can use $\mathcal{R}$ to solve Π, unless $\mathcal{E}$ occurs. By applying Lemma 3, we obtain that the success probability of $\mathcal{A}$ is at least

$$\epsilon_{\mathcal{A}} \geq \epsilon_{\mathcal{R}} - \Pr[\mathcal{E}] = \epsilon_{\mathcal{R}} - \Pr[M_0 \notin Q \wedge M_1 \in Q] \geq \epsilon_{\mathcal{R}} - \exp(-1)/q.$$

$\mathcal{A}$ essentially runs $\mathcal{R}$ twice and performs one re-randomization, therefore the running time of $\mathcal{A}$ is $t_{\mathcal{A}} \approx 2t_{\mathcal{R}} + t_{\mathsf{ReRand}}$.

The above theorem directly gives rise to the following corollary.

Corollary 1. *Let Π be a $(\epsilon, 2t + t_{\mathsf{ReRand}})$-hard computational problem. Then the success probability $\epsilon_{\mathcal{R}}$ of any security reduction from Π to breaking a re-randomizable signature scheme that runs in time t is at most*

$$\epsilon_{\mathcal{R}} \leq \exp(-1)/q + \epsilon.$$

In particular, if ϵ is close to zero and signatures are efficiently re-randomizable, then this gives an upper bound on the success probability of the reduction of $\epsilon_{\mathcal{R}} \lessapprox \exp(-1)/q$ for all reductions running in time t.

Note that in principle any (probabilistic) signature scheme is re-randomizable, though not necessarily efficiently. However, the running time of the simulated forger depends on the running time of the re-randomization algorithm. Thus, in order to get a meaningful result, we need to require that signatures are *efficiently* re-randomizable.

4.4 Reductions That Run $\mathcal{F}$ More Than Once

So far we have only considered reductions that run the forger once. While the reduction from [22] is of this type, it may be possible that there exist a tighter reduction that runs $\mathcal{F}$ several times with different public keys. Fortunately, following [7,17] it is very simple to generalize the result of Section 4.3 to reductions that run $\mathcal{F}$ repeatedly.

Theorem 5. *Let* Sig *be a t_{ReRand}-re-randomizable signature scheme and let Π be a computational problem as in Definition 4. If there exists a $(t_{\mathcal{F}}, \epsilon_{\mathcal{F}}, q, \epsilon_{\mathcal{R}}, t_{\mathcal{R}})$-reduction $\mathcal{R}$ that runs $\mathcal{F}$ at most r times and reduces Π to breaking* Sig*, then there exists an algorithm $\mathcal{A}$ that $(\epsilon_{\mathcal{A}}, t_{\mathcal{A}})$-solves Π with $t_{\mathcal{A}} \approx 2 \cdot t_{\mathcal{R}} + r \cdot t_{\mathsf{ReRand}}$ and*

$$\epsilon_{\mathcal{A}} \geq \epsilon_{\mathcal{R}} - (r \cdot \exp(-1))/q.$$

The proof is very similar to the proof of Theorem 4, the only difference is that now $\mathcal{A}$ needs to simulate r executions of $\mathcal{F}$. Consider an adversary $\mathcal{A}$ which proceeds exactly like in the proof of Theorem 4, and let $\mathcal{E}_i$ denote the event that the simulation of $\mathcal{F}$ fails in the i-th execution. Then we have

$$\begin{aligned}\epsilon_{\mathcal{A}} &\geq \epsilon_{\mathcal{R}} - \sum_{i=1}^{r} \Pr[\mathcal{E}_i] = \epsilon_{\mathcal{R}} - \sum_{i=1}^{r} \Pr[M_{i,0} \notin Q \wedge M_{i,1} \in Q] \\ &\geq \epsilon_{\mathcal{R}} - (r \cdot \exp(-1))/q,\end{aligned}$$

where $M_{i,0}$ and $M_{i,1}$ are the sequences of chosen-message queries issued by the simulated forger in the i-th execution of $\mathcal{F}$.

4.5 Waters Signatures are Re-randomizable

To show that any reduction from a computationally hard problem to the (t, q, ϵ)-EUF-CMA security of Waters signatures loses at least a factor $1/q$, it remains to show that Waters signatures are efficiently re-randomizable.

Note that the original Waters scheme from [22] and the variant from Section 3 differ only in the way the hash value $H(m) \in \mathbb{G}$ is computed. The following considerations do not depend on a specific function H. Therefore we consider a Waters signature scheme that uses some abstract hash function H in the sequel, which makes the analysis applicable to both schemes (and other similar constructions) simultaneously.

Lemma 4. *Waters signatures are t-re-randomizable, where t amounts to two exponentiations in $\mathbb{G}$ plus some minor additional operations.*

Proof. Let $pk = (\mathbb{G}, g, g^\alpha, g^\beta, H)$ be a given public key, and let m and $\sigma = (\sigma_1, \sigma_2)$ be a given message with valid Waters signature, i.e., σ satisfies

$$e(g^\alpha, g^\beta) \cdot e(\sigma_1, H(m)) = e(g, \sigma_2). \tag{5}$$

Since σ_1 is a group element, we can write $\sigma_1 = g^r$ for some integer $r \in \mathbb{Z}_p$, where $p = |\mathbb{G}|$ is the order of $\mathbb{G}$. Then Equation 5 implies that we can write σ_2 as $\sigma_2 = g^{\alpha\beta} H(m)^r$. The set of all (σ_1, σ_2) satisfying Equation 5 is therefore identical to the set

$$\Sigma(pk, m) = \{(g^r, g^{\alpha\beta} H(m)^r) : r \in \mathbb{Z}_p\}.$$

It remains to show that there exists an efficient algorithm ReRand that produces uniformly random elements of $\Sigma(pk, m)$ given only the public key pk, message m, and a valid signature $\sigma = (\sigma_1, \sigma_2)$. Consider algorithm ReRand taking as input pk, signature $(\sigma_1, \sigma_2) = (g^r, g^{\alpha\beta} H(m)^r)$ for some r, and message m. The algorithm samples $s \xleftarrow{\$} \mathbb{Z}_p$ and computes and returns (σ_1', σ_2') where

$$\sigma_1' := \sigma_1 \cdot g^s = g^{r+s} \qquad \text{and} \qquad \sigma_2' := \sigma_2 \cdot H(m)^s = g^{\alpha\beta} H(m)^{r+s}.$$

Since s is uniformly distributed over $\mathbb{Z}_p$, the resulting signature (σ_1', σ_2') is distributed uniformly over $\Sigma(sk, m)$, as required.

Combining the above lemma with Theorem 4 yields the following result.

Theorem 6. *Let Π be a computational problem according to Definition 4. If there exists a $(t_\mathcal{F}, \epsilon_\mathcal{F}, q, \epsilon_\mathcal{R}, t_\mathcal{R})$-reduction $\mathcal{R}$ that reduces solving Π to breaking Waters signatures, then there exists an algorithm $\mathcal{A}$ that $(\epsilon_\mathcal{A}, t_\mathcal{A})$-solves Π with $t_\mathcal{A} \approx 2t_\mathcal{R}$ and*

$$\epsilon_\mathcal{A} \geq \epsilon_\mathcal{R} - \exp(-1)/q.$$

Thus, a reduction from any computational problem Π to breaking Waters signatures that runs in time t with success probability significantly better than $1/q$ implies that there exists an algorithm solving Π in time $\approx 2t$ with significant success probability.

Acknowledgements. We are grateful to Daniel Kraschewski for pointing us to expander codes, to Eike Kiltz for insightful discussions, and to Brent Waters for helpful remarks on our paper.

References

1. Attrapadung, N., Furukawa, J., Gomi, T., Hanaoka, G., Imai, H., Zhang, R.: Efficient Identity-Based Encryption with Tight Security Reduction. In: Pointcheval, D., Mu, Y., Chen, K. (eds.) CANS 2006. LNCS, vol. 4301, pp. 19–36. Springer, Heidelberg (2006)
2. Bellare, M., Ristenpart, T.: Simulation without the Artificial Abort: Simplified Proof and Improved Concrete Security for Waters' IBE Scheme. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 407–424. Springer, Heidelberg (2009)
3. Bernstein, D.J.: Proving Tight Security for Rabin-Williams Signatures. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 70–87. Springer, Heidelberg (2008)
4. Camenisch, J.L., Lysyanskaya, A.: Signature Schemes and Anonymous Credentials from Bilinear Maps. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 56–72. Springer, Heidelberg (2004)
5. Chevallier-Mames, B., Joye, M.: A Practical and Tightly Secure Signature Scheme Without Hash Function. In: Abe, M. (ed.) CT-RSA 2007. LNCS, vol. 4377, pp. 339–356. Springer, Heidelberg (2007)
6. Coron, J.-S.: On the Exact Security of Full Domain Hash. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 229–235. Springer, Heidelberg (2000)
7. Coron, J.-S.: Optimal Security Proofs for PSS and Other Signature Schemes. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 272–287. Springer, Heidelberg (2002)
8. Feller, W.: An Introduction to Probability Theory and Its Applications, 3rd edn., vol. 1. Wiley (1968)
9. Gennaro, R., Halevi, S., Rabin, T.: Secure Hash-and-Sign Signatures without the Random Oracle. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 123–139. Springer, Heidelberg (1999)
10. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing 17(2), 281–308 (1988)
11. Guo, F., Mu, Y., Susilo, W.: How to Prove Security of a Signature with a Tighter Security Reduction. In: Pieprzyk, J., Zhang, F. (eds.) ProvSec 2009. LNCS, vol. 5848, pp. 90–103. Springer, Heidelberg (2009)
12. Guo, F., Mu, Y., Susilo, W.: Personal communication (November 2011)
13. Hofheinz, D., Kiltz, E.: Programmable Hash Functions and Their Applications. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 21–38. Springer, Heidelberg (2008)
14. Hofheinz, D., Kiltz, E.: Programmable hash functions and their applications. Journal of Cryptology 25(3), 484–527 (2012), http://eprint.iacr.org/2011/270
15. Hughes, B.D.: Random Walks and Random Environments. Random Walks, vol. 1. Oxford University Press (1995)
16. Joye, M.: An Efficient On-Line/Off-Line Signature Scheme without Random Oracles. In: Franklin, M.K., Hui, L.C.K., Wong, D.S. (eds.) CANS 2008. LNCS, vol. 5339, pp. 98–107. Springer, Heidelberg (2008)
17. Kakvi, S.A., Kiltz, E.: Optimal Security Proofs for Full Domain Hash, Revisited. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 537–553. Springer, Heidelberg (2012)

18. Katz, J., Wang, N.: Efficiency improvements for signature schemes with tight security reductions. In: Jajodia, S., Atluri, V., Jaeger, T. (eds.) ACM CCS 2003: 10th Conference on Computer and Communications Security, October 27-30, pp. 155–164. ACM Press, Washington, D.C. (2003)
19. Lysyanskaya, A., Rivest, R.L., Sahai, A., Wolf, S.: Pseudonym Systems (Extended Abstract). In: Heys, H.M., Adams, C.M. (eds.) SAC 1999. LNCS, vol. 1758, pp. 184–199. Springer, Heidelberg (2000)
20. Schäge, S.: Tight Proofs for Signature Schemes without Random Oracles. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 189–206. Springer, Heidelberg (2011)
21. Sipser, M., Spielman, D.A.: Expander codes. IEEE Transactions on Information Theory 42(6), 1710–1722 (1996)
22. Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
23. Zémor, G.: On expander codes. IEEE Transactions on Information Theory 47(2), 835–837 (2001)

Strong Security from Probabilistic Signature Schemes

Sven Schäge*

University College London
s.schage@ucl.ac.uk

Abstract. We introduce a new and very weak security notion for signature schemes called target randomness security. In contrast to previous security definitions we focus on signature schemes with (public coin) probabilistic signature generation where the randomness used during signature generation is exposed as part of the signature. To prove practical usefulness of our notion we present a new signature transformation for mapping target randomness secure signature schemes to weakly secure signature schemes. It is well-known that, using chameleon hash functions, the resulting weakly secure scheme can then be turned into a fully secure one. Our transformation outputs signature schemes that in general produce signatures with l elements, where l is the bit length of the input randomness. We present an instantiation of a target randomness secure signature scheme based on the RSA assumption and show that after applying our new signature transformation to this scheme, we can *accumulate* the l signature elements into a single element. This results in a new efficient RSA-based signature scheme. In contrast to traditional security definitions, all signature schemes obtained with our transformation enjoy *strong* security, i.e. they remain secure even if the adversary outputs a new signature on a previously queried message. In our proofs, we rely on the prefix-based technique introduced by Hohenberger and Waters at Crypto'09. However, using a precise analysis we are able decrease the security loss in proofs relying on the prefix-based technique. This result may be of independent interest.

Keywords: digital signature, signature scheme, RSA, accumulation, target randomness, transformation, prefix, tightness.

1 Introduction

Signature transformations that map signature schemes with weak security guarantees to schemes which fulfill the standard notion of security have proven very useful in the past. This is because it is much more easy to design a signature scheme which only fulfills a very weak notion of security than a fully secure scheme. Many of the existing signature schemes like [25,5,21,6] have been developed in this spirit by first specifying a signature scheme with weak guarantees and then applying a corresponding signature transformation to construct a

* Supported by EPSRC grant number EP/G013829/1.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 84–101, 2012.

scheme with stronger properties. This is also true for the recent signature scheme by Hohenberger and Waters (HW) from Crypto'09. In their work, Hohenberger and Waters present the first RSA-based hash-and-sign signature scheme in the standard model and so solved a long-standing open problem [21]. To this end, they tackle the problem that the challenger can predict the message $\overline{M}$ that the adversary will generate a signature on (in the following called target message) only with negligible probability. Their solution is to force the adversary to not only process $\overline{M}$ but also, *independently*, all prefixes of $\overline{M}$. (In the following, we refer to this approach as 'prefix-based technique'.) This greatly reduces the complexity for the simulator, because now it can guess with non-negligible probability one of the prefixes of $\overline{M}$. The remaining task of the challenger is to embed the RSA challenge such that it can extract a solution from the attacker's forgery if its guess was correct.

As Hohenberger-Waters and Brakerski and Tauman Kalai (BTK) [6] pointed out, the HW scheme gives rise to a new transformation[1] that step-wisely transforms signature schemes which only guarantee a very weak form of security – security against universal (message) forgeries under generic chosen (message) attacks (UMUF-GMA)[2], to weakly secure schemes, i.e. schemes secure against existential (message) forgeries under generic chosen message attacks (EMUF-GMA). (For formal definitions we refer to Section 2.1.) The UMUF-GMA security game is equivalent to the definition of EMUF-GMA security, except that the attacker *is given* the message it has to produce a forgery on – subsequently called 'target message' – in the first move of the security experiment. The transformation essentially grasps the ideas behind the prefix-based technique by HW. The key idea is to use a UMUF-GMA secure scheme to sign the l prefixes of the message M as in the HW scheme. The final signature consists of all the signatures on these prefixes. However, apparently the HW scheme cannot be obtained as a direct application of this transformation; HW signatures have constant size and do not grow with the message length as one might expect given the BTK transformation description. There must be some additional structure that the HW scheme exploits.

In this work we analyze public coin probabilistic signature schemes, where the randomness used in the signature generation is also sent to the verifier (as part of the signature). We show how public coin signatures schemes that only fulfill very weak notions of security can be used to construct efficient fully *and strongly* secure signature schemes.

Contribution. We extend the existing work on signature schemes in the standard model. In particular we

– define a new and very weak security notion called *target randomness security* that defines 'existential message universal randomness unforgeability against generic chosen message and randomness attacks' (EMURUF-GMRA) for

[1] This transformation was implicitly given in [21] and made explicit in [6].
[2] [6] use the term *a-priori-message unforgeability* to refer to UMUF-GMA security.

signatures with public coin probabilistic signature generation. Signature schemes that are EMURUF-GMRA secure are always secure in the strong sense.
- present a new *general* transformation from EMURUF-GMRA secure signature schemes to weakly secure signature schemes. Signatures of the resulting scheme consist of l elements and are strongly secure as well.
- present a new and efficient target randomness secure RSA-based signature scheme with a probabilistic signing algorithm.
- show that when applying our transformation to our RSA-based scheme the signature elements of the resulting weakly secure signature scheme can be accumulated into a single group element. This results in a new and efficient RSA-based signature scheme with constant-size signatures. Slightly modified our technique makes obvious why the size of the HW signatures does not grow with the message size.
- improve the loss of tightness in prefix-based security reductions. This improvement transfers to all proofs that rely on the prefix-based techniques like HW [21], BTK [6], and the recent signature scheme by Hofheinz, Jager and Kiltz (HJK) [18].

To obtain signature schemes secure under the standard definition – security against existential message unforgeability under adaptive chosen message attacks (EMUF-AMA) [17] or full security – we can apply the well-known Shamir-Tauman transformation to generate EMUF-AMA secure schemes from EMUF-GMA secure schemes [31]. In Section 2.2 we extend this result by showing that if the chameleon hash function also guarantees a certain form of strong security, the resulting signature scheme is strongly secure too. We stress that most chameleon hash function are also secure in this strong sense. This is advantageous in scenarios where strong security is required, as we do not have to apply an additional transformation like the Bellare-Shoup [3] or Huang *et al.* [22] transformation to turn EMUF-AMA secure schemes into strongly secure ones. These transformations increase the signature size and make the signing and verification algorithms less efficient.

Related Work. Our work is related to the existing standard model signature schemes. Most of the factoring-based hash-and-sign signature schemes were proven secure under the Strong RSA (SRSA) assumption [16,13,25,34,35,7,15,19,30]. Until 2009, it was an open problem to design an efficient signature scheme that is secure solely under the RSA assumption. Hohenberger and Waters stepwisely solved this problem by first presenting a stateful RSA-based signature scheme at EUROCRYPT'09 [20] and, in the same year, a stateless RSA-based signature scheme at CRYPTO'09 [21]. Recently, Hofheinz, Jager, and Kiltz (HJK) presented a new signature scheme that is secure under the sole RSA assumption [18]. It essentially relies on programmable hash functions as introduced in [19] and results in very small signature sizes while having a relatively large number of public key elements. Technically, the authors also rely on the prefix-based proof technique and similar to our result, they also use all the prefixes of the *randomness* to sign a message. Our RSA-based scheme differs from their scheme in the following way. 1) Signatures are

longer as we need two random values for signature generation. However, in 2006 Mironov showed how to re-use the second random value as a key to a target collision resistant (TCR) hash function [24]. Target collision resistant hash functions can be used as an alternative to collision-resistant (CR) hash functions for domain extension of the message space. In contrast to CRs, TCRs do only rely on the existence of one-way functions. At the same time they are much more efficient than provably secure CRs in groups of hidden (composite) order [9]. Thus our instantiation allows to efficiently sign long messages without relying on additional security assumptions. Similar arguments hold if we compare our scheme with the HW scheme, which also only uses a single random value, see Section 6.1. 2) Our public key is much smaller and comparable to that of the HW scheme. We stress that besides these issues our focus is much more general. Our main result consist in a new security definition together with an appropriate transformation to weakly secure schemes. We aim at showing that probabilistic signature schemes, even with very weak security properties, provide interesting starting points for the construction of strongly secure signature schemes. At the same time our tightness improvements hold for prefix-based security proofs *in general* and independent of the underlying security assumption.

By now there exist several security definitions and corresponding transformations for signature schemes. In 1989 Even, Goldreich and Micali showed 1) how to construct fully secure signature schemes from schemes that are secure under known message attacks and 2) how to construct *practical* fully secure signature schemes from schemes that are chosen message secure [14]. Cramer, Damgård, and Pedersen presented an alternative construction for 1) that features a much smaller signature size ($O(\kappa)$ instead of $O(\kappa^2)$ in the Even *et. al.* transformation where κ is the security parameter) [12]. The signature scheme by Naccache *et al.* can be interpreted as an application of this transformation to an SRSA-based known message secure signature scheme [25]. In 2001, Shamir and Tauman presented an improved transformation for 2) that maps weakly secure signature schemes to fully secure schemes using chameleon hash functions [31]. Due to the efficiency of the resulting signature schemes this transformation is very popular and an essential ingredient in several signature schemes like [5,21]. In 2007, Bellare and Shoup [3] and independently Huang *et. al.* [22] presented a generic transformation to construct strongly secure signature schemes. In contrast to the standard security notion, in the attack game of strongly secure signature schemes the adversary is also allowed to output *a new signature on a previously queried message*. Figure 1 gives an overview of the existing (and new) notions and transformations of chosen-message security. We have ignored selective security, where the adversary may choose the target message/randomness, as universally secure schemes can trivially be transformed into selectively secure schemes under generic chosen message attacks (see HW [21] and BTK [6]). A random element X is simply added to the public key and in the first step of the signature generation the message is XORed with X. This technique is implicit in HW and the signature transformation of Section 3.

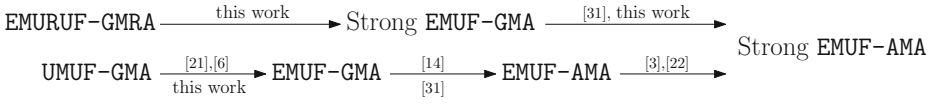


Fig. 1. Security Notions and Transformations of Chosen-Message Secure Signature Schemes. Arrows from A to B indicate that an efficient transformation exists that constructs a secure instantiation of B from a secure instantiation of A. References below arrows point to improved transformations.

With respect to tightness improvements, our work follows the line of work initiated by Bellare and Rogaway who showed tight security of PSS [1]. We concentrate on tightness improvements for existing signature schemes (*without* introducing additional modifications). In 2000, Coron provided tighter proofs for Full-Domain Hash [10,11]. Eight years later Bernstein [4] presented the first tight proofs for the Rabin-William's signature schemes. All these result hold in the random oracle model. In 2008, Hofheinz and Kiltz [19] presented asymptotical tightness improvements for the Computational Diffie-Hellman based signature scheme by Waters [33] that is secure without random oracles. Recently, Schäge presented tight proofs for (new and) existing SRSA and Strong Diffie-Hellman based signature schemes in the standard model like the Cramer-Shoup [13], Fischlin [15], Zhu [34,35], and Camenisch-Lysyanskaya [8] scheme.

2 Preliminaries and Notation

The security parameter is denoted as $\kappa \in \mathbb{N}$. We write 1^κ to describe the string that consist of κ ones and let $l = l(\kappa)$ and $q = q(\kappa)$ be polynomials. For a set S, we use $x \xleftarrow{\$} S$ to denote that x is drawn from S uniformly at random and $|S|$ to denote the cardinality of S. If s is a string, we write $|s|$ to denote its bit-length.We let $\perp$ denote the empty string. If $M \in \{0,1\}^l$ we let $M = m_1 m_2 \dots m_l$ with $m_j \in \{0,1\}$ for $j \in \{1,\dots,l\}$ be the binary representation of M. We use M^i to denote the prefix of M which consist of the first $i \in [1;l]$ bits: $M^i = m_1 m_2 \dots m_i$. For an algorithm $\mathcal{A}$ we write $\mathcal{A}(i_1, i_2, \dots)$ to denote that $\mathcal{A}$ has input parameters $i_1, i_2, \dots$. Similarly, we denote with $y \leftarrow \mathcal{A}(i_1, i_2, \dots)$ that $\mathcal{A}$ outputs y when running on inputs $i_1, i_2, \dots$. We write PPT (probabilistic polynomial time) to refer to randomized algorithms that run in polynomial time. As usual $\gcd(a,b)$ with $a, b \in \mathbb{Z}$ denotes the greatest common divisor of a and b. Our new signature scheme will be secure under the well-known RSA assumption [27].

Definition 1 (RSA assumption (RSA)). *Given an RSA modulus $n = pq$, where p, q are sufficiently large primes, a prime $\alpha < \phi(n)$ with $\gcd(\alpha, \phi(n)) = 1$, and an element $u \in \mathbb{Z}_n^*$, we say that the $(t_{RSA}, \epsilon_{RSA})$-RSA assumption holds if for all t_{RSA}-time adversaries $\mathcal{A}$*

$$Pr[(x) \leftarrow \mathcal{A}(n, u, \alpha),\ x \in \mathbb{Z}_n^*,\ x^\alpha = u \bmod n] \leq \epsilon_{RSA}.$$

The probability is over the random choices of u, n, α and the random coins of $\mathcal{A}$.

2.1 Signature Scheme

In a digital signature scheme $\mathcal{S} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ the PPT algorithm KeyGen generates the key material: secret key SK and public key PK. The algorithm $\mathsf{Sign}(SK, M)$ uses SK and a message M from the message space $\{0,1\}^l$ to output a signature σ. If the signing algorithm is probabilistic we use $R \xleftarrow{\$} \{0,1\}^l$ to denote the randomness used for the signature generation. In this case we also write $\mathsf{Sign}(SK, M, R)$. The verification algorithm $\mathsf{Verify}(PK, M, \sigma)$ processes PK, a message M, and a purported signature σ on M and outputs 1 if σ is a legitimate signature on M and 0 otherwise. If Sign is probabilistic $\mathsf{Verify}(PK, M, \sigma, R)$ additionally processes randomness R that was used for the signature generation. Usually one may regard this randomness as a part of the signature. For clarity we deviate from this convention and make R explicit.

We restrict ourself to signature schemes where the randomness R is generated in the signing phase and directly given to the verifier (as part of the signature). Most of the existing signature schemes have this property, examples are [16,13,25,34,35,7,15,20,21,5,8,19,29]. In the following two security definitions we consider the most general case of forgeries – existential forgeries. We use the terminology of Goldwasser, Micali and Rivest [17].

Full Security: Existential Message Unforgeability under Adaptive Chosen Message Attacks (`EMUF-AMA`). The standard notion of security for signature schemes is called existential message unforgeability under adaptive chosen message attacks [17]. Here the adversary is given access to a signing oracle $\mathcal{O}_{SK}(\cdot)$ to adaptively query signatures.

Setup. In the setup phase, $\mathsf{KeyGen}(1^\kappa)$ is run and the public key PK is given to the adversary.

Signature queries. The adversary adaptively queries the signing oracle $\mathcal{O}_{SK}(\cdot)$ with q messages $M_1, \ldots, M_q \in \{0,1\}^l$ of his choice and obtains q signatures $\sigma_1, \ldots, \sigma_q$ with $\mathsf{Verify}(PK, M_i, \sigma_i) = 1$ for $i \in \{1, \ldots, q\}$.

Output. The attacker outputs $(\overline{M}, \overline{\sigma})$ such that $\overline{M} \notin \{M_1, \ldots, M_q\}$ and at the same time $\mathsf{Verify}(PK, \overline{M}, \overline{\sigma}) = 1$.

Weak Security: Existential Unforgeability under Generic Chosen Message Attacks (`EMUF-GMA`). In this attack model, the attacker specifies all signature queries *before it receives the public key.*

Signature queries. At first the adversary outputs a list of q signature queries $M_1, \ldots, M_q \in \{0,1\}^l$.

Public Key Generation and Signature Output. In the next phase, the public key PK is given to the adversary together with q signatures $\sigma_1, \ldots, \sigma_q$ such that $\mathsf{Verify}(PK, M_i, \sigma_i) = 1$ for $i \in \{1, \ldots, q\}$.

Output. The attacker outputs $(\overline{M}, \overline{\sigma})$ such that $\overline{M} \notin \{M_1, \ldots, M_q\}$ and at the same time $\mathsf{Verify}(PK, \overline{M}, \overline{\sigma}) = 1$.

Both of the above security notions are well-known. As sketched above, the only difference between the definition of `UMUF-GMA`-security and weak security is that

the attacker is given the 'target message' $\overline{M}$ in the first step of the security game and only has to output $\overline{\sigma}$. Let us now present our new security definition for signature schemes with a probabilistic signing algorithm.

Target Randomness Security: Existential Message Universal Randomness Unforgeability under Generic Chosen Message & Randomness Attacks (EMURUF-GMRA). In contrast to the previous security definitions we exploit the randomness R used in probabilistic public coin signing algorithms; both the challenger and the attacker can now also specify the randomness used for the signature generation. The adversary is given the target randomness in the first step of the security experiment. Informally we refer to this notion as target randomness security.

Target Randomness. At first the attacker is given the (target) randomness $\overline{R}$.
Signature queries. The adversary outputs q pairs of message/randomness as $(M_1, R_1) \ldots, (M_q, R_q)$ with $M_i, R_i \in \{0,1\}^l$. For at most one pair (M_i, R_i) it may hold that $R_i = \overline{R}$.
Public Key Generation and Signature Output. Next, the public key PK is given to the adversary together with q signatures $\sigma_1, \ldots, \sigma_q$ such that $\mathsf{Verify}(PK, M_i, \sigma_i, R_i) = 1$.
Output. The attacker outputs $\overline{M}, \overline{\sigma}$ such that $\mathsf{Verify}(PK, \overline{M}, \overline{\sigma}, \overline{R}) = 1$ and $\overline{M}, \overline{R}$ is not among the signature queries.

We denote the success probability of an adversary $\mathcal{A}$ (taken over the random coins of the challenger and the adversary) to win the i-security game as $Adv_{\mathcal{S},\mathcal{A},i}$ where $i \in \{$EMUF-AMA, EMUF-GMA, EMURUF-GMRA$\}$.

Definition 2 (Secure signature scheme). *An adversary $\mathcal{A}$ is said to (q, t, ϵ)-break the i-security ($i \in \{$EMUF-AMA, EMUF-GMA, EMURUF-GMRA$\}$) of a signature scheme $\mathcal{S}$ if $\mathcal{A}$ has success probability $Adv_{\mathcal{S},\mathcal{A},i} = \epsilon$ after generating at most q queries and running in time t. $\mathcal{S}$ is said to be (q, ϵ, t)-secure if there exists no PPT adversary that (q, ϵ, t)-breaks the existential unforgeability of $\mathcal{S}$. A signature scheme is called* strongly secure *if in the above games $\mathcal{A}$ may also output a forgery with $\overline{M} \in \{M_1, \ldots, M_q\}$ but $(\overline{M}, \overline{\sigma}) \notin \{(M_1, \sigma_1), \ldots, (M_q, \sigma_q)\}$ (or $(\overline{M}, \overline{\sigma}, \overline{R}) \notin \{(M_1, \sigma_1, R_1), \ldots, (M_q, \sigma_q, R_q)\}$ in case of probabilistic signatures).*

Discussion. In our new security definition the adversary is still allowed to output messages $\overline{M}$ of his choice (as in the first two security definitions), i.e. existential message forgeries (in contrast to UMUF-GMA security). Observe that previously queried messages may be re-used in the forgery what makes our definition inherently strongly secure. In the new security game, the adversary can now explicitly specify the random values R_i. To the best of our knowledge no existing security definition for signature schemes allows similar attack capabilities. It does not only give the adversary control over the messages to be signed but also *specifies* the randomness used for signature generation. In most signature schemes it is essential for security that the randomness is not controlled by the adversary. In our case, the only restriction is that the forgery must verify under randomness $\overline{R}$,

while $\overline{R}$ has been queried at most once before. However, due to this freedom it is technically more difficult to construct a target randomness secure scheme than a UMUF-GMA secure scheme. Informally, when constructing a UMUF-GMA secure scheme one might directly use an 'all-but-one assumption' where the simulator can easily invert a function f for all but a single output value. This is possible because we give the adversary the target forgery $\overline{M}$ that *cannot be sent to the signing oracle* as the security definition requires $\overline{M} \notin \{M_1, \ldots, M_q\}$. The signature must simply be set up such that the target message $\overline{M}$ exactly corresponds to the one value that the simulator cannot invert. We cannot transfer this technique to target randomness secure schemes because here the adversary might *re-use the target randomness* $\overline{R}$ in his signature queries. On the one hand, the simulator *must be able* to answer all signature queries even the one with $R_i = \overline{R}$. On the other hand it *must not be able* to construct the forgery by itself as it wants to extract a solution to an underlying problem from it. We must setup the parameters such that the simulator is not required to invert this function for the signature query (M_i, R_i) with $R_i = \overline{R}$. The idea is to make the inversion also depend on the message M such that only for M_i the simulator can produce a signature without actually inverting. For all other $M \neq M_i$ this must not be possible.

2.2 From Weakly to Fully Secure Schemes

There exists a well-known transformation by Shamir and Tauman [31] for constructing fully secure schemes from weakly secure signature schemes to using chameleon hash functions [23].[3] It will be applied to our final weakly secure signature scheme to yield a fully secure one. The basic idea is to first use the message as input to a (randomized) chameleon hash function and then sign the output using the weakly secure signature scheme. Signatures consist of the so produced signature and the randomness used for the computation of the chameleon hash. There exist chameleon hash functions that are secure under the RSA [21] or the factoring assumption [31]. Since the factoring assumption is weaker than the RSA assumption we can utilize this transformation without making additional complexity assumptions. In both cases, the overhead amounts to an additional element in the secret key and the public key. We can easily extend the Shamir-Tauman result by showing that if 1) the chameleon hash function has slightly stronger security guarantees than required by the standard definition and 2) the weakly secure signature scheme is strongly secure then the resulting fully secure scheme is strongly secure too. We again stress that most of the existing chameleon hash functions are secure in this strong sense. The proof of the following theorem is straight-forward and, for space reasons, appears in the full version.

Theorem 1. *If the underlying weakly secure signature scheme is strongly secure and the chameleon hash function also guarantees that it is hard (given only the*

[3] In 1999, Gennaro-Halevi-Rabin also proposed a similar but less general solution for their signature scheme [16].

public parameters) to compute two distinct random values that make any message map to the same output value, then the Shamir-Tauman transformation produces fully and strongly *secure signature schemes.*

3 A New Transformation to Weakly Secure Signature Schemes

Let us now present our new transformation that maps EMURUF-GMRA secure signature schemes to EMUF-GMA secure schemes. In Section 4 we then present a new RSA-based EMURUF-GMRA secure signature scheme. First, we fix some additional notation.

Encoding Function. In the following we will regularly produce signatures on strings $S \in \{0,1\}^l$ and on their prefixes. These strings will naturally be interpreted as integers. Now, if $s_l = 0$, S^l and S^{l-1} obviously map to the same integer. However, our proof technique requires that these strings map to different values. To accomplish this we apply an injective and invertible encoding function $\mathsf{enc} : \{0,1\}^{\leq l} \to \{0,1\}^{l+1} \setminus \{0^{k+1}, 0^k 1\}$ that maps to fixed-size outputs. Given an input string $S \in \{0,1\}^{\leq l}$, enc first prepends a 1 and subsequently leading zeros until the result has length $l+1$: $\mathsf{enc}(S) = 0^{l-|S|}1s_1 \dots s_{|S|}$. In the following, we will denote with $R^{\underline{i}} \in \{0,1\}^{l+1}$, $i \in [1;l]$ the string $R^{\underline{i}} = \mathsf{enc}(R^i) = \mathsf{enc}(r_1 \dots r_i)$. It is easy to see that we now always have $R^{\underline{i}} \neq R^{\underline{j}}$ for $j \in [1;l]$ and $i \neq j$.

Target Randomness Secure Signature Scheme => Weakly Secure Signature Scheme. The final signature σ consist of l distinct signatures $\sigma = (\sigma_1, \dots, \sigma_l)$. Each of the single signatures is *on the same message* M. We use *the prefixes of the randomness* R to modify the signature generation, such that $\sigma_i = \mathsf{Sign}(SK, M, R^{\underline{i}} \oplus X)$ for all $i \in [1;l]$. Let us go into more detail.

Let $\mathcal{S}_{\text{tRand}} = (\mathsf{KeyGen}_{\text{tRand}}, \mathsf{Sign}_{\text{tRand}}, \mathsf{Verify}_{\text{tRand}})$ be a probabilistic and EMURUF-GMRA secure signature scheme. Then we can construct the EMUF-GMA secure signature scheme $\mathcal{S} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ as follows.

- $\mathsf{KeyGen}(1^\kappa)$: run $\mathsf{KeyGen}_{\text{tRand}}(1^\kappa)$ and obtain a key pair $(PK_{\text{tRand}}, SK_{\text{tRand}})$ for the signature scheme. Next, draw a random $X \in \{0,1\}^{l+1}$. The scheme's secret key is $SK = SK_{\text{tRand}}$, its public key is $PK = (PK_{\text{tRand}}, X)$.
- $\mathsf{Sign}(SK, M, R)$: to obtain a signature on message $M \in \{0,1\}^l$ using randomness $R \in \{0,1\}^l$, compute the l-tuple $\sigma = (\sigma_1, \dots, \sigma_l)$ s.t. for each $i \in [1;l]$ the i-th component is computed as $\sigma_i = \mathsf{Sign}_{\text{tRand}}(SK_{\text{tRand}}, M, R^{\underline{i}} \oplus X)$.
- $\mathsf{Verify}(PK, M, \sigma, R)$: parse σ as $\sigma_1, \dots, \sigma_l$. If it holds that $M, R \in \{0,1\}^l$ and $\bigwedge_{i=1}^{l} \mathsf{Verify}(PK, M, \sigma_i, R^{\underline{i}} \oplus X) = 1$ output 1, otherwise 0.

Theorem 2. *Let $\mathcal{S}_{tRand}$ be a (q', t', ϵ')-secure probabilistic signature scheme that is secure under* EMURUF-GMRA *attacks. Then the application of the above transformation produces a* EMUF-GMA *signature scheme $\mathcal{S}$ that is (q, t, ϵ)-secure under generic chosen message attacks provided that*

$$q = q',\ t \approx t',\ \epsilon \leq 2(q(l - \lfloor \log(q) \rfloor) + 2^{\lfloor \log(q) \rfloor + 1} - 1)\epsilon' + q^2/2^l$$

for $q \geq 1$ *and*

$$q = q',\ t \approx t',\ \epsilon \leq 2\epsilon'$$

for $q = 0$. *Moreover, if* $\mathcal{S}_{tRand}$ *is strongly secure so is* $\mathcal{S}$.

Theorem 2 already makes use of our improved security analysis. To prove Theorem 2 we first need to analyze the set of prefixes of q l-bit strings.

Definition 3 (Prefix-Closure). *Let* S *be an* l*-bit string. We use* $\mathsf{precl}(S) = \{S^j |\ j \in [0;l]\} \cup \perp$ *to denote the prefix-closure of* S*, i.e. the set of all prefixes of* S (including the empty string $\perp$). If R is a set of q l-bit strings, i.e. $R = \{R_1, \ldots, R_q\} \subseteq \{0,1\}^l$ for $q \in \mathbb{N}$ and $q > 0$, we call $\mathsf{precl}(R) = \bigcup_{i=1}^{q} \mathsf{precl}(R_i)$ the prefix-closure of R. It is the set of all the prefixes of all the R_i. In case $q = 0$, we define $\mathsf{precl}(R) = \{\perp\}$.

Definition 4 (Co-Path of Prefix-Closure). *Let* $R = \{R_1, \ldots, R_q\} \subseteq \{0,1\}^l$ *for* $q \in \mathbb{N}, q > 0$ *and* $\mathsf{precl}(R)$ *be the prefix-closure of* R*. Let* $\mathcal{Z}_{\mathsf{precl}(R)}$ *denote the set of all strings* $z = z_1 \ldots z_k$ *with* $k \in [1;l]$ *such that* $z_1 \ldots z_{k-1} \in \mathsf{precl}(R)$ *but* $z \notin \mathsf{precl}(R)$*. We say that* $\mathcal{Z}_{\mathsf{precl}(R)}$ *is the co-path of* $\mathsf{precl}(R)$*. For* $q = 0$*, we define* $\mathcal{Z}_{\mathsf{precl}(R)} = \{0,1\}$.

In prefix-based security proofs it is essential to bound the *maximal* size of the co-path of the prefix-closure. Roughly, in the proof the simulator chooses an element z of $\mathcal{Z}_{\mathsf{precl}(R)}$ uniformly at random. This element is used to embed the complexity challenge. With probability $\geq 1/|\mathcal{Z}_{\mathsf{precl}(R)}|$ z will be a prefix of the randomness in the forgery[4]. Thus the simulator's success probability is $\geq 1/|\mathcal{Z}_{\mathsf{precl}(R)}|$. This accounts for the security loss in prefix-based proofs. The existing results upper bound $|\mathcal{Z}_{\mathsf{precl}(R)}|$ simply as $|\mathcal{Z}_{\mathsf{precl}(R)}| \leq ql$. Subsequently, in Theorem 3, we present a more precise analysis. But first we need to analyze the worst-case size of the prefix-closure of R.

Lemma 1. *Let* $q, l \in \mathbb{N}$ $q, l > 0$*. Let* $R = \{R_1, \ldots, R_q\} \subseteq \{0,1\}^l$ *be an arbitrary set of* q l*-bit strings. Then it holds that*

$$\max_R |\mathsf{precl}(R)| = q(l - \lfloor \log(q) \rfloor) + 2^{\lfloor \log(q) \rfloor + 1} - 1$$

Proof. We will show which properties R must fulfill to have a maximum size prefix-closure. For convenience we partition $\mathsf{precl}(R) = \mathsf{precl}_{\leq}(R) \cup \mathsf{precl}_{>}(R)$ depending on q. We consider the set $\mathsf{precl}_{\leq}(R)$ of prefixes with length smaller than or equal to $\lfloor \log(q) \rfloor$ and the set of prefixes $\mathsf{precl}_{>}(R)$ that are longer than $\lfloor \log(q) \rfloor$-bits separately. If we maximize both sets, $|\mathsf{precl}(R)| = |\mathsf{precl}_{\leq}(R)| + |\mathsf{precl}_{>}(R)|$ is maximal too.

For $q > 1$ it is clear that there are 'prefix-collisions', i.e. there are $R_j, R_{j'}$ with $R_j \neq R_{j'}$ such that $\mathsf{precl}(R_j) \cap \mathsf{precl}(R_{j'}) \neq \emptyset$. (In fact the pigeon-hole principle shows that for $q > 2^w$ and $w \geq 0$ there is always a pair of distinct indices $j, j' \in [1;q]$ with $|\mathsf{precl}(R_j) \cap \mathsf{precl}(R_{j'})| = w + 1$.) The size of $\mathsf{precl}_{\leq}(R)$ is maximal if i) despite of these collisions the prefixes of the R_i cover all prefixes lower

[4] In HW and BTK, z must be a prefix of the forgery's message.

than or equal to $\lfloor \log(q) \rfloor$, i.e. $\mathsf{precl}_{\leq}(R) = \{0,1\}^{\leq \lfloor \log(q) \rfloor}$. Thus, the maximal size of $\mathsf{precl}_{\leq}(R)$ is $|\mathsf{precl}_{\leq}(R)| \leq \sum_{i=0}^{\lfloor \log(q) \rfloor} = 2^{\lfloor \log(q) \rfloor + 1} - 1$. To analyze $\mathsf{precl}_{>}(R)$ observe that if $2^{\lfloor \log(q) \rfloor + 1} > q$ ii) there need not be any prefix-collisions among the prefixes of length greater than $\lfloor \log(q) \rfloor$. In this case every $\mathsf{precl}(R_i)$ adds the maximal number, $l - \lfloor \log(q) \rfloor$, of additional prefixes to $\mathsf{precl}(R)$. Thus $|\mathsf{precl}_{>}(R)| \leq q(l - \lfloor \log(q) \rfloor)$ is maximal and $|\mathsf{precl}(R)| \leq 2^{\lfloor \log(q) \rfloor + 1} - 1 + q(l - \lfloor \log(q) \rfloor)$.

Theorem 3. *Let $q, l \in \mathbb{N}$ $q, l > 0$. Let $R = \{R_1, \ldots, R_q\} \subseteq \{0,1\}^l$ be an arbitrary set of q l-bit strings. Then it holds that*

$$q(l - \lfloor \log(q) \rfloor) + 2^{\lfloor \log(q) \rfloor} \leq \max_R \left| \mathcal{Z}_{\mathsf{precl}(R)} \right| \leq q(l - \lfloor \log(q) \rfloor) + 2^{\lfloor \log(q) \rfloor + 1} - 1.$$

Proof. To prove the upper-bound observe that for $q \geq 1$ we always have that $|\mathsf{precl}(R)| \geq |\mathcal{Z}_{\mathsf{precl}(R)}|$. This directly follows from the definition of $\mathsf{precl}(R)$. Lemma 1 gives the maximal size of $\mathsf{precl}(R)$. To show the lower bound recall the construction of sets $R = \{R_1, \ldots, R_q\}$ with maximal sized prefix-closure. For $\mathsf{precl}(R)$ to have maximal cardinality we must have that all prefixes lower than $\lfloor \log(q) \rfloor$ are in $\mathsf{precl}(R)$, i.e. $\{0,1\}^{\leq \lfloor \log(q) \rfloor} \subset \mathsf{precl}(R)$. However, this means that for all prefixes with length i strictly lower than $\lfloor \log(q) \rfloor$ we cannot find a corresponding element in $\mathcal{Z}_{\mathsf{precl}(R)}$. This is because all prefixes of length $i+1$ are already in $\mathsf{precl}(R)$ and by definition cannot also be in $\mathcal{Z}_{\mathsf{precl}(R)}$. Thus, for a maximum size prefix-closure we have for the size of the corresponding co-path:

$$|\mathcal{Z}_{\mathsf{precl}(R)}| \geq \max_R \{|\mathsf{precl}(R)|\} - \sum_{j=0}^{\lfloor \log(q) \rfloor - 1} 2^j = q(l - \lfloor \log(q) \rfloor) + 2^{\lfloor \log(q) \rfloor}.$$

Now, the maximum size of $|\mathcal{Z}_{\mathsf{precl}(R)}|$ over all R must be at least as large as $q(l - \lfloor \log(q) \rfloor) + 2^{\lfloor \log(q) \rfloor}$. This concludes the proof of Theorem 3.

Discussion. Since $|\mathcal{Z}_{\mathsf{precl}(R)}| \leq q(l - \lfloor \log(q) \rfloor) + 2^{\lfloor \log(q) \rfloor + 1} - 1 \leq q(l - \lfloor \log(q) \rfloor + 2)$ and both l and q are polynomials in the security parameter we obtain a clear improvement, in particular for large q. Still q is the dominant factor of the security loss. In their paper, HW also gave a new prefix-based security proof for the CDH-based Waters signature scheme [33]. Before that, Hofheinz and Kiltz (HK) have already proposed *asymptotically* better bounds for the security loss in the Waters scheme. They give a new security analyses of the Waters hash function showing that the security loss is in $O(q\sqrt{l})$ [19]. The original reduction by Waters accounts for a security loss of $8q(l+1)$. The HK improvement relies on random walks and essentially exploits that the message bits of the prefixes can be processed independently as variables of a linear function in the exponents of the hash function's group elements. However, there does not seem to be a general way to transfer their approach to signature scheme like the HW scheme where prefixes are first mapped to prime numbers in a non-linear way. Although our improvements are quantitatively smaller our result 1) provides a *non-asymptotic*, i.e. concrete, bound on the security loss and 2) works for prefix-based technique

in general and thus also applies to the (original) Waters, Hohenberger-Waters, and Hofheinz-Jager-Kiltz scheme and the general transformation presented by Brakerski and Tauman Kalai.

3.1 Proof of Theorem 2

We are now ready to prove the security of our transformation.

Proof. Assume $\mathcal{S}$ is not secure and let $\mathcal{A}$ be a successful attacker against $\mathcal{S}$. Then we can build a simulator $\mathcal{B}$ that uses $\mathcal{A}$ in a black box manner to break the security of $\mathcal{S}_{\text{tRand}}$. Let $\tilde{R}$ be the first message (the target randomness) received from $\mathcal{B}$'s challenger and let $M_1 \ldots, M_q$ be $\mathcal{A}$'s signature queries. In the first step $\mathcal{B}$ draws q random values $R_1, \ldots, R_q \in \{0,1\}^l$. With overwhelming probability these values are all distinct: a simple union bound shows that a collision occurs with probability $\leq q^2/2^l$.

In the next step, $\mathcal{B}$ draws a random coin $y \xleftarrow{\$} \{0,1\}$ indicating whether $\mathcal{A}$ will re-use any of the R_i as randomness in the forgery. According to y, $\mathcal{B}$ will setup the public parameters in two different ways.

If $y = 0$, $\mathcal{B}$ assumes that $\mathcal{A}$ will not re-use any of the R_i in the forgery. Observe that $\overline{R} \notin \{R_1, \ldots, R_q\}$ implies that there must be at least one prefix $\overline{R}^j$ with $\overline{R}^j \notin \mathsf{precl}(R)$. By construction of $\mathcal{Z}_{\mathsf{precl}(R)}$ this means that there exists a prefix $\overline{R}^v$ with $\overline{R}^v \in \mathcal{Z}_{\mathsf{precl}(R)}$. In the first step, $\mathcal{B}$ guesses this $\overline{R}^v$ upfront by drawing a random string $z \xleftarrow{\$} \mathcal{Z}_{\mathsf{precl}(R)}$. Then $\mathcal{B}$ computes $X \in \{0,1\}^{l+1}$ as $X = \mathsf{enc}(z) \oplus \tilde{R}$. Next, $\mathcal{B}$ sends ql signature queries $\{(M_i, T_i^j)\}_{i\in[1;q],j\in[1;l]}$ to the challenger with $T_i^j = R_i^j \oplus X$ for all $i \in [1;q]$. The challenger answers with a public key PK_{tRand} and ql signatures $\{\sigma_{i,j}\}_{i\in[1;q],j\in[1;l]}$ on the given messages such that $\mathsf{Verify}_{\text{tRand}}(PK_{\text{tRand}}, M_i, \sigma_{i,j}, T_i^j) = 1$ for all $i \in [1;q], j \in [1;l]$. In the next step $\mathcal{B}$ sends $PK_{\text{weak}} = (PK_{\text{tRand}}, X)$ and q randomness/signature pairs $(\Sigma_1, R_1), \ldots, (\Sigma_q, R_q)$ to $\mathcal{A}$. Each Σ_i consist of l signatures of the target randomness secure scheme: $\Sigma_i = (\sigma_{i,1}, \ldots, \sigma_{i,l})$ for all $i \in [1;q]$. By assumption $\mathcal{A}$ then outputs a forgery $(\overline{M}, \overline{\Sigma} = (\overline{\sigma_1}, \ldots, \overline{\sigma_l}), \overline{R})$ with $\overline{R} \notin \{R_1, \ldots, R_q\}$. Now with probability $\geq q(l - \lfloor \log(q) \rfloor) + 2^{\lfloor \log(q) \rfloor + 1} - 1$ $\mathcal{B}$'s guess of z is right. In this case there exists a $v \in [1;l]$ with $\overline{R}^v = z$ and it holds that $\mathsf{Verify}_{\text{tRand}}(PK_{\text{tRand}}, \overline{M}, \overline{\sigma_v}, \overline{R}^v \oplus X) = 1$. Since by definition we have that $\overline{R}^v \oplus X = \mathsf{enc}(z) \oplus X = \tilde{R}$, $(\overline{M}, \overline{\sigma_v})$ breaks the security of the target randomness secure signature scheme.

In case $y = 1$, $\mathcal{B}$ assumes that $\mathcal{A}$ will re-use any of the R_i in the forgery and guesses the signature index of the corresponding randomness upfront by drawing $w \xleftarrow{\$} [1;q]$. Next, $\mathcal{B}$ computes $X = R_w^l \oplus \tilde{R}$ and T_i^j as $T_i^j = R_i^j \oplus X$ for all $i \in [1;q], j \in [1;l]$. Then $\{(M_i, T_i^j)\}_{i\in[1;q],j\in[1;l]}$ is given to the challenger who in turn answers with the public key PK_{tRand} and ql signatures $\{\sigma_{i,j}\}_{i\in[1;q],j\in[1;l]}$. By assumption it must always be the case that $\mathsf{Verify}_{\text{tRand}}(PK_{\text{tRand}}, M_i, \sigma_{i,j}, R_i^j \oplus X) = 1$ with $i \in [1;q], j \in [1;l]$. As before $\mathcal{B}$ now gives $PK_{\text{weak}} = (PK_{\text{tRand}}, X)$

and q signatures $\Sigma_1, \ldots, \Sigma_l$ with $\Sigma_i = (\sigma_{i,1}, \ldots, \sigma_{i,l}, R_i)$, $i \in [1;q]$ to $\mathcal{A}$. However, this time $\mathcal{A}$ outputs a forgery $(\overline{M}, \overline{\Sigma} = (\overline{\sigma_1}, \ldots, \overline{\sigma_l}), \overline{R})$ with $\overline{R} \in \{R_1, \ldots, R_q\}$. With probability $\geq q$, we have that $\overline{R} = R_w$. Then $(\overline{M}, \overline{\sigma_l})$ breaks the security of the underlying EMURUF-GMRA secure signature scheme since $\overline{R}^{\underline{l}} \oplus X = R_w^{\underline{l}} \oplus X = R$. Observe that only in case $y = 1$ we need that the R_i are all distinct to comply with the requirement of the EMURUF-GMRA security game, i.e. the target randomness is only queried at most once to the signature oracle.

With probability $\geq 1/2$, $\mathcal{B}$'s guess of y is correct. Observe that by construction $\mathcal{A}$ cannot tell apart the values produced by $\mathcal{B}$ from those of the original attack game. This concludes the proof of Theorem 2. Also note that at no point in the proof we rely on the fact that $\overline{M} \notin \{M_1, \ldots, M_q\}$. Thus the resulting signature scheme is strongly secure.

4 A Target Randomness Secure Signature Scheme Based on the RSA Assumption

Let us now present our new RSA-based EMURUF-GMRA secure signature scheme. For simplicity we focus on the practically most relevant case of balanced, safe RSA moduli with prime public exponent but we stress that the signature scheme can be instantiated in general RSA groups as well.

- $\mathsf{KeyGen}(1^\kappa)$: Choose two large balanced and safe primes $\hat{p} = 2\overline{p}+1, \hat{q} = 2\overline{q}+1$ with $|\overline{p}| = |\overline{q}|, \overline{p} > \overline{q}$ and $2^{l+1} < \overline{p} \cdot \overline{q}$. Set $n = \hat{p}\hat{q}$ and $SK = \hat{p}, \hat{q}$. Next choose a random $X' \xleftarrow{\$} \{0,1\}^{l+1}$. We will also need a target collision-resistant function $p : \{0,1\}^{l+1} \rightarrow \mathsf{Primes}_{l+1}$ that maps strings to the set of odd prime numbers $\mathsf{Primes}_{l+1} \subset [1; 2^{l+1} - 1]$. We use the following instantiation. Choose a key s' for a pseudo-random function $t : \{0,1\}^* \rightarrow \{0,1\}^{l+1}$. Whenever we want to evaluate p on input $R^{\underline{i}} \in \{0,1\}^{l+1}$ we continually increment *the resolving index* $ind \in \{0,1\}^*$, which is initially set to $ind := 0$, until $t(ind||R^{\underline{i}}) \oplus X$ is prime. For a proof that p indeed is collision-free for the first polynomial many input values we refer to [21] or [18].[5] Finally, choose two random generators a, b of a subgroup S of $\mathbb{Z}_n^*$ with $\langle a \rangle = \langle b \rangle = S \subset \mathbb{Z}_n^*$ and $|S| = \overline{p} \cdot \overline{q}$. Publish $PK = (a, b, n, s', X')$.
- $\mathsf{Sign}(SK, M, R)$: To sign a message $M \in \{0,1\}^{l+1}$, choose a random $R \in \{0,1\}^{l+1}$ and compute $s = (ab^M)^{1/p(R)} \bmod n$. Output σ and R.
- $\mathsf{Verify}(PK, M, \sigma, R)$: If $\sigma^{p(R)} = ab^M \bmod n$ output 1, else output 0.

Theorem 4. *If the RSA assumption is secure then the above construction yields a target randomness secure probabilistic signature scheme. Moreover, the security reduction tightly reduces to the RSA assumption.*

[5] Basically, one first shows that with overwhelming probability one can always find a prime while $ind \leq (l+1)^2$ – otherwise the PRF could be distinguished from a truly random function. In the second step, one argues that the probability to find a collision only after $q(l+1)^2$ evaluations of the PRF must be negligible as otherwise the PRF construction could again easily be distinguished from a truly random function.

Proof. Assume the simulator $\mathcal{B}$ is given an RSA challenge (n, u, α). At first $\mathcal{B}$ draws a random key s' for the pseudo-random permutation and a random $R \in \{0,1\}^l$. Next it constructs X' such that $\alpha = t(ind||R^{\underline{l}}) \oplus X' = p(R)$. Then $\mathcal{B}$ sends R to the forger $\mathcal{A}$. As an answer $\mathcal{B}$ receives q signature queries $(M_1, R_1), \ldots, (M_q, R_q)$ from $\mathcal{A}$. With these values $\mathcal{B}$ can now set up the rest of the public key. To this end, $\mathcal{B}$ draws $h_a, h_b \in [1; (n-1)/4]$. In the following, $\mathcal{B}$ considers two cases.

- If there is no R_i with $R_i = R$ (Case 1), $\mathcal{B}$ computes $a = u^{h_a \prod_{i=1}^{q} p(R_i)}$ and $b = u^{h_b \prod_{i=1}^{q} p(R_i)}$.
- If there exists such a value (Case 2) so let $j \in [1; q]$ be the corresponding index with that $R_j = R$. Now, $\mathcal{B}$ computes $a = u^{h_a \prod_{i=1}^{q} p(R_i) + h_b M_j \prod_{i=1, i\neq j}^{q} p(R_i)}$ and $b = u^{-h_b \prod_{i=1, i\neq j}^{q} p(R_i)}$.

Observe that in both cases a and b are distributed almost like in the original attack game. This is because h_a, h_b are almost distributed uniformly in $[1; \phi(n)]$. The probability for a value $h \xleftarrow{\$} [1; (n-1)/4]$ not to be in $[1; \overline{p} \cdot \overline{q}]$ is

$$\Pr[h \xleftarrow{\$} [1; (n-1)/4], h \notin [1; \overline{p} \cdot \overline{q}]] \leq (\overline{p} + \overline{q})/(2\overline{pq} + \overline{p} + \overline{q}) < 1/(\overline{q} + 1) < 1/2^{|\overline{p}|-2}$$

and thus negligible.

$\mathcal{B}$ can now easily answer the signature queries by computing signatures σ_k on (M_k, R_k) for all $k \in [1; q]$ as follows.

- In Case 1, $\mathcal{B}$ computes $\sigma_k = u^{h_a \prod_{i=1, i\neq k}^{q} p(R_i)} \cdot u^{M_k h_b \prod_{i=1, i\neq k}^{q} p(R_i)}$.
- If it holds that $R_k \neq R$ in Case 2, $\mathcal{B}$ computes the queried signature σ_k as follows: $\sigma_k = u^{h_a \prod_{i=1, i\neq k}^{q} p(R_i) + M_j h_b \prod_{i=1, i\neq j,k}^{q} p(R_i)} \cdot u^{-M_k h_b \prod_{i=1, i\neq j,k}^{q} p(R_i)}$.
- If $R_k = R$ (or $k = j$) in Case 2, $\mathcal{B}$ computes $\sigma_k = u^{h_a \prod_{i=1, i\neq k}^{q} p(R_i)}$.

Observe that these values perfectly simulate the original attack game.

The last case solves the problem that we are faced with when designing EMURUF-GMRA secure schemes (see the discussion in Section 2.1). If $M_k \neq M_j$ and $R_k = R$, $\mathcal{B}$ cannot compute a signature since for the exponent e it holds that $e = h_a \prod_{i=1}^{q} p(R_i) + h_b(M_j - M_k) \prod_{i=1, i\neq k}^{q} p(R_i)$ which implies $p(R) \not| e$ and thus $\mathcal{B}$ would need to compute $p(R)$ roots what, by the RSA assumption, is not feasible. However, if $M_j = M_k$ we get that $e = h_a \prod_{i=1}^{q} p(R_i)$ and clearly $p(R)|e$. This time $\mathcal{B}$ can generate a signature by just exponentiating.

Now when $\mathcal{A}$ outputs the forgery $(\overline{M}, \overline{\sigma}, R)$, $\mathcal{B}$ can extract a solution to the RSA challenge as follows.

In Case 1, the verification equation gives us $\overline{\sigma}^{p(R)} = u^{(h_a + \overline{M} h_b) \prod_{i=1}^{q} p(R_i)}$. Let us analyze the probability for the event $p(R)|(h_a + \overline{M} h_b) \prod_{i=1}^{q} p(R_i)$. Since p is target collision-resistant the $p(R_i)$ are all distinct from $p(R)$ and we only must analyze whether $p(R)|(h_a + \overline{M} h_b)$. Since $\mathcal{A}$ never gets to see u in the clear, h_a and h_b are perfectly hidden from her view. As 3 is the smallest prime number $p(R)$ can take on, the probability for $\mathcal{A}$ to output $\overline{M}$ with $p(R)|(h_a + \overline{M} h_b)$ is at most $1/3$. We so have with probability $\geq 2/3$ that $\gcd(p(R), (h_a + \overline{M} h_b) \prod_{i=1}^{q} p(R_i)) = 1$.

In this case, we can easily compute two integers w_1 and w_2 (using Euclidean algorithm) such that $w_1 p(R) + w_2(h_a + \overline{M}h_b)\prod_{i=1}^{q} p(R_i) = 1$. It then holds that $u = u^{w_1 p(R)+w_2(h_a+\overline{M}h_b)\prod_{i=1}^{q} p(R_i)} = u^{w_1 \cdot p(R)} \cdot \overline{\sigma}^{w_2 \cdot p(R)}$. Therefore the final solution to the RSA challenge is $u^{1/p(R)} = u^{1/\alpha} = u^{w_1} \cdot \overline{\sigma}^{w_2}$.

In Case 2, we can show with the same arguments as above that with probability at least 2/3 we have $p(R) \nmid \left(h_a \prod_{i=1}^{q} p(R_i) + (M_j - \overline{M})h_b \prod_{i=1, i\neq j}^{q} p(R_i)\right)$. Using the same techniques as before $\mathcal{B}$ finds the corresponding values w_1 and w_2 such that $w_1 p(R) + w_2 \left(h_a \prod_{i=1}^{q} p(R_i) + (M_j - \overline{M})h_b \prod_{i=1, i\neq j}^{q} p(R_i)\right) = 1$. The final solution to the RSA challenge is $u^{1/\alpha} = u^{w_1} \cdot \overline{\sigma}^{w_2}$.

This concludes the proof of Theorem 4.

5 Accumulation of Signature Schemes

When applying our transformation from target randomness secure signature schemes to weakly secure schemes to the above signature scheme we get signatures that consist of l group elements.

We now show how to accumulate signatures of the type $\sigma' = (\sigma'_1, \ldots, \sigma'_l)$ where for each $i \in [1; q]$ $\sigma'_i = (ab^M)^{1/p(R^{\underline{i}})} \bmod n$ to a *single* element $\Sigma' = (ab^M)^{1/\prod_{i=1}^{l} p(R^{\underline{i}})} \bmod n$. The accumulation technique used here is a direct application of the extended Euclidean algorithm. Observe that our accumulation technique does not require knowledge of the secret key.

Lemma 2. *Let PK, M, R, and $\sigma'_1 = (ab^M)^{1/p(R^{\underline{1}})}, \ldots, \sigma'_l = (ab^M)^{1/p(R^{\underline{l}})} \in \mathbb{Z}_n^*$ with $l \in \mathbb{N}$ be given. Then we can easily compute $\Sigma' = (ab^M)^{1/\prod_{i=1}^{l} p(R^{\underline{i}})} \bmod n$.*

Proof. Since for $i \neq j$ it holds that $R^{\underline{i}} \neq R^{\underline{j}}$ and because of the properties of $p(\cdot)$ we have that the $e_1 = p(R^{\underline{1}}), \ldots, e_l = p(R^{\underline{l}})$ are distinct primes. Let $\bar{e}_i = \prod_{j=1, j\neq i}^{l} e_i$ and $e = \prod_{i=1}^{l} e_i$. By construction we have $\gcd(\bar{e}_1, \ldots, \bar{e}_l) = 1$. Next we can use extended Euclidean algorithm to find $a_1, \ldots, a_l \in \mathbb{Z}$ such that $\gcd(\bar{e}_1, \ldots, \bar{e}_l) = \sum_{i=1}^{l} a_i \bar{e}_i = 1$. We have $(ab^M) = (ab^M)^{\sum_{i=1}^{l} a_i \bar{e}_i} \bmod n$. As it holds for all e_i that $\gcd(e_i, \phi(n)) = 1$ we can finally find the e-th root of ab^M as

$$\Sigma' = (ab^M)^{1/\prod_{i=1}^{l} e_i} = \prod_{i=1}^{l} (ab^M)_i^{a_i/e_i} = \prod_{i=1}^{l} \sigma'^{a_i} \bmod n.$$

Lemma 3. *Given PK, M, R, and $\Sigma' = (ab^M)^{1/(\prod_{i=1}^{l} p(R^{\underline{i}}))} \in \mathbb{Z}_n^*$ we can easily compute $\sigma'_1 = (ab^M)^{1/p(R^{\underline{1}})}, \ldots, \sigma'_l = (ab^M)^{1/p(R^{\underline{l}})} \in \mathbb{Z}_n^*$.*

Proof. For all $j \in \{1, \ldots, l\}$, if we want to find the j-th component of the basic scheme we simply compute

$$\sigma'_j = \Sigma'^{\prod_{i=1, i\neq j}^{l} p(R^{\underline{i}})} \bmod n.$$

The previous two lemmas show that both signature descriptions, the l-element signature and the accumulated signature, are equivalent. The above technique can easily be adapted to accumulate a variant of the Gennaro-Halevi-Rabin [16] signature scheme by Brakerski-Tauman-Kalai [6]. The result is the Hohenberger-Waters signature. In the above we simply have to set $(ab^M) := u$ and $R := M$.

6 Final RSA Signature Scheme

Like the Hohenberger-Waters scheme, our final RSA-based EMUF-GMA secure signature scheme features a built-in accumulation process.

- $\mathsf{KeyGen}(1^\kappa)$: The key generation algorithm is exactly the same as in the scheme of Section 4.
- $\mathsf{Sign}(SK, M, R)$: To sign a message $M \in \{0,1\}^l$ choose a random $R \in \{0,1\}^l$ and compute $\sigma = (ab^M)^{1/(\prod_{i=1}^{l} p(R^i))} \bmod n$. Output (σ, R).
- $\mathsf{Verify}(PK, M, \sigma, R)$: If $\sigma^{\prod_{i=1}^{l} p(R^i)} = ab^M \bmod n$ output 1, else output 0.

Theorem 5. *Applying the Shamir-Tauman transformation to the above signature scheme as presented in Section 2.2 gives us a strongly and fully secure signature scheme under the RSA assumption.*

6.1 Comparison with the Hohenberger-Waters Scheme

Our RSA-based signature scheme presents an alternative to the Hohenberger-Waters signature scheme. The times for signature generation and verification are comparable. As a drawback, the size of our signatures is longer than the Hohenberger-Waters signature. Besides a group element in $\mathbb{Z}_n^*$, our scheme additionally contains a random string R where $|R|$ is ≈ 160. However, when signing long messages our scheme requires weaker security assumptions than the Hohenberger-Waters scheme. Let us explain this in more detail. To extend the input domain of a signature scheme, one usually applies a collision-resistant hash function and signs the hash value of the input message. Alternatively, one can also use a primitive called target collision resistant hash function (TCR) (or universal hash function) [26]. TCRs are fundamentally weaker primitives than collision-resistant hash functions, since on the one hand there exist efficient constructions of TCRs from one-way functions [28,26,17] but on the other hand collision resistant hash function cannot be constructed from one-way functions using black-box constructions [32]. There exists a standard transformation for signature schemes by Bellare and Rogaway that allows to exchange the collision-resistant hash function with a target collision-resistant function when signing long messages [2]. Usually this would require an additional random element to be embedded in the signature – the key of the TCR. However, following similar arguments as Mironov [24] we can re-use the message-independent randomness R of our signature scheme for this purpose. Therefore, our RSA-based signature scheme can use target collision resistant hash functions without any modification for domain extension.

References

1. Bellare, M., Rogaway, P.: The Exact Security of Digital Signatures - How to Sign with RSA and Rabin. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 399–416. Springer, Heidelberg (1996)
2. Bellare, M., Rogaway, P.: Collision-Resistant Hashing: Towards Making UOWHFs Practical. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 470–484. Springer, Heidelberg (1997)
3. Bellare, M., Shoup, S.: Two-Tier Signatures, Strongly Unforgeable Signatures, and Fiat-Shamir Without Random Oracles. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 201–216. Springer, Heidelberg (2007)
4. Bernstein, D.J.: Proving Tight Security for Rabin-Williams Signatures. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 70–87. Springer, Heidelberg (2008)
5. Boneh, D., Boyen, X.: Short signatures without random oracles and the SDH assumption in bilinear groups. J. Cryptology 21(2), 149–177 (2008)
6. Brakerski, Z., Kalai, Y.T.: A framework for efficient signatures, ring signatures and identity based encryption in the standard model. Cryptology ePrint Archive, Report 2010/086 (February 2010), http://eprint.iacr.org/ (version from February 13, 2010)
7. Camenisch, J.L., Lysyanskaya, A.: A Signature Scheme with Efficient Protocols. In: Cimato, S., Galdi, C., Persiano, G. (eds.) SCN 2002. LNCS, vol. 2576, pp. 268–289. Springer, Heidelberg (2003)
8. Camenisch, J.L., Lysyanskaya, A.: Signature Schemes and Anonymous Credentials from Bilinear Maps. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 56–72. Springer, Heidelberg (2004)
9. Contini, S., Lenstra, A.K., Steinfeld, R.: VSH, an Efficient and Provable Collision-Resistant Hash Function. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 165–182. Springer, Heidelberg (2006)
10. Coron, J.-S.: On the Exact Security of Full Domain Hash. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 229–235. Springer, Heidelberg (2000)
11. Coron, J.-S.: Optimal Security Proofs for PSS and Other Signature Schemes. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 272–287. Springer, Heidelberg (2002)
12. Cramer, R., Damgård, I., Pedersen, T.P.: Efficient and Provable Security Amplifications. In: Lomas, M. (ed.) Security Protocols 1996. LNCS, vol. 1189, pp. 101–109. Springer, Heidelberg (1997)
13. Cramer, R., Shoup, V.: Signature schemes based on the Strong RSA assumption. ACM Trans. Inf. Syst. Secur. 3(3), 161–185 (2000)
14. Even, S., Goldreich, O., Micali, S.: On-line/off-line digital signatures. J. Cryptology 9(1), 35–67 (1996)
15. Fischlin, M.: The Cramer-Shoup Strong-RSA Signature Scheme Revisited. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 116–129. Springer, Heidelberg (2003)
16. Gennaro, R., Halevi, S., Rabin, T.: Secure Hash-and-Sign Signatures without the Random Oracle. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 123–139. Springer, Heidelberg (1999)
17. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. 17(2), 281–308 (1988)

18. Hofheinz, D., Jager, T., Kiltz, E.: Short Signatures From Weaker Assumptions. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 647–666. Springer, Heidelberg (2011)
19. Hofheinz, D., Kiltz, E.: Programmable Hash Functions and Their Applications. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 21–38. Springer, Heidelberg (2008)
20. Hohenberger, S., Waters, B.: Realizing Hash-and-Sign Signatures under Standard Assumptions. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 333–350. Springer, Heidelberg (2009)
21. Hohenberger, S., Waters, B.: Short and Stateless Signatures from the RSA Assumption. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 654–670. Springer, Heidelberg (2009)
22. Huang, Q., Wong, D.S., Li, J., Zhao, Y.: Generic transformation from weakly to strongly unforgeable signatures. J. Comput. Sci. Technol. 23(2), 240–252 (2008)
23. Krawczyk, H., Rabin, T.: Chameleon signatures. In: NDSS. The Internet Society (2000)
24. Mironov, I.: Collision-Resistant No More: Hash-and-Sign Paradigm Revisited. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 140–156. Springer, Heidelberg (2006)
25. Naccache, D., Pointcheval, D., Stern, J.: Twin signatures: an alternative to the hash-and-sign paradigm. In: ACM Conference on Computer and Communications Security, pp. 20–27 (2001)
26. Naor, M., Yung, M.: Universal one-way hash functions and their cryptographic applications. In: STOC, pp. 33–43. ACM (1989)
27. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21(2), 120–126 (1978)
28. Rompel, J.: One-way functions are necessary and sufficient for secure signatures. In: STOC, pp. 387–394. ACM (1990)
29. Schäge, S.: Twin Signature Schemes, Revisited. In: Pieprzyk, J., Zhang, F. (eds.) ProvSec 2009. LNCS, vol. 5848, pp. 104–117. Springer, Heidelberg (2009)
30. Schäge, S.: Tight Proofs for Signature Schemes without Random Oracles. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 189–206. Springer, Heidelberg (2011)
31. Shamir, A., Tauman, Y.: Improved Online/Offline Signature Schemes. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 355–367. Springer, Heidelberg (2001)
32. Simon, D.R.: Findings Collisions on a One-Way Street: Can Secure Hash Functions Be Based on General Assumptions? In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 334–345. Springer, Heidelberg (1998)
33. Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
34. Zhu, H.: New digital signature scheme attaining immunity to adaptive-chosen message attack. Chinese Journal of Electronics 10(4), 484–486 (2001)
35. Zhu, H.: A formal proof of Zhu's signature scheme. Cryptology ePrint Archive, Report 2003/155 (2003), http://eprint.iacr.org/

Space Efficient Signature Schemes from the RSA Assumption

Shota Yamada[1,⋆] Goichiro Hanaoka[2], and Noboru Kunihiro[1]

[1] The University of Tokyo
{yamada@it,kunihiro@}k.u-tokyo.ac.jp
[2] National Institute of Advanced Industrial Science and Technology (AIST)
hanaoka-goichiro@aist.go.jp

Abstract. Signature schemes from the RSA assumption are very important because of their highly reliable security. Despite their importance, only a few digital signature schemes from the RSA assumption are currently known. Thus, improvement of efficiency in this area seems to be very important. In this paper, we propose various signature schemes from the RSA assumption. First, we propose a scheme that simultaneously provides the shortest signatures and public key length among the known schemes. Compared with the known best schemes, the signature size is the same as that of the scheme proposed recently by Hofheinz, Jager, and Kiltz, whereas the public key size is about the half that of the Hohenberger-Waters scheme. The drawback of the scheme is its heavy signing and verification algorithms. Second, we also propose a scheme whose public key is longer than our first scheme, but the signing and verification cost is more efficient. The scheme can be seen as a generalization of our first scheme and the Hofheinz-Jager-Kiltz scheme. Finally, we propose a scheme whose signing and verification algorithms are more efficient than our first and second schemes, whereas the signature size is longer. All these schemes are constructed based on a new observation about the relation between m-time signature schemes and short signature schemes.

Keywords: Short signature, m-time signature, RSA assumption.

1 Introduction

1.1 Background

Construction of a digital signature scheme with existential unforgeability under chosen message attack (EUF-CMA) [9] in the standard model is a main research topic in cryptography. In particular, the construction of a short signature from a mild assumption has been extensively studied. Earlier studies proposed various efficient signature schemes in the standard model from various assumptions, such as the strong q-DH assumption [2,11], the q-DH assumption [10,20], the strong

⋆ The first author is supported by a JSPS Research Fellowship for Young Scientists.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 102–119, 2012.

RSA assumption [8,3,7,11], and the CDH assumption [19]. Many of these schemes rely on the q-type assumption or the strong type assumption, except for Waters' scheme [19]. Even though these assumptions seem reasonable, it is desired to construct a signature scheme from a better studied, weaker assumption (such as the RSA assumption), to obtain high confidentiality in security. A digital signature scheme from the RSA assumption whose signatures are short enough, is not known even in the stateful setting until the recent work by Hohenberger and Waters [12]. Subsequently, they proposed a signature scheme from the RSA assumption in the stateless setting using a new technique [13]. Very recently, Hofheinz, Jager, and Kiltz showed that even shorter signature schemes can be obtained using a programmable hash function [10]. Despite of their importance, no (stateless) signature schemes from the RSA assumption are known, except for the schemes proposed in the above two papers. Improvement in efficiency for RSA based signature schemes seems very important as a step to obtaining a truly efficient, reliable signature scheme in the future. In this paper, we propose various novel signature schemes from the RSA assumption. For example, we propose a scheme that achieves the shortest signature size and public key size simultaneously.

1.2 Our Approach

As an approach to constructing short signature schemes, we focus on the fact that a one-time signature scheme and a weakly secure signature scheme yield a fully-fledged signature scheme. This is a variant of the generic construction proposed in [18]. As we will discuss in a later section, the idea can be (informally) generalized to the combination of an m-time signature scheme and a weakly secure scheme. This is the first time that this idea has been explicitly discussed. Even though the idea is not formal, the idea is conceptually of interest since it often leads to constructions of short signature schemes. For example, recent generic constructions of short signature schemes from the programmable hash function [10] and its variant [20] can be seen as the realizations of the idea. Based on this idea, we construct various novel signature schemes from the RSA assumption. Conceptually, we take two steps to construct a scheme. First, we construct an m-time signature scheme from the RSA assumption. We then combine it with a weakly secure signature scheme from the RSA assumption proposed by Hohenberger-Waters [13]. According to this strategy, we obtain various new schemes from the RSA assumption.

More concretely, we obtain three signature schemes based on the approach described above. In section 4, we propose a signature scheme that provides the shortest signature size and public key size simultaneously. Compared to currently known best schemes, the signature size of our scheme is 1074 bits, which is the same as that of $\mathsf{Sig}_{\mathsf{RSA}}[\mathsf{H}_{\mathsf{cfs}}]$ in [10], whereas the public key size is about 2000 bits, which is about half the size of the scheme by Hohenberger and Waters [13]. The drawback of the scheme is its heavy signing and verification algorithms. To compensate it, in section 5, we also propose another scheme whose signature size is the same as our first scheme, and the public key is longer than for that scheme,

but the signing and verification cost is more efficient. The scheme is equipped with parameters u_1, u_2 and we can adjust the trade-off between public key length and signing and verification cost. The scheme can be seen as a generalization of our first scheme and $\mathsf{Sig}_{\mathsf{RSA}}[\mathsf{H}_{\mathsf{cfs}}]$ in [10]. In fact, we can obtain a slight variant of these schemes as a special case of the scheme. Finally, in section 6, we propose a scheme whose signing and verification cost is more efficient than the our first and second schemes whereas the public key and signature sizes are larger than for our first scheme. The structure of the scheme can be seen as a hybrid of our first scheme and $\mathsf{Sig}_{\mathsf{RSA}}[\mathsf{H}_{\mathsf{rand}}]$ in [10].

Finally, we note that we have also constructed a stateful version of our second and third schemes. The scheme is more efficient than corresponding stateless version of the scheme except for the signature size, which is slightly larger.

2 Preliminaries

For $\lambda \in \mathbb{N}$, 1^λ denotes the string of λ ones, with λ expressing the security parameter throughout this paper. $[d]$ denotes the set $\{1, 2, \ldots, d\}$. Moreover, $|x|$ and $|S|$ denote, respectively, the length of bitstring x, and the size of set S. If S is a set, $s \xleftarrow{\$} S$ denotes the action of uniform randomly selecting an element of S. Given algorithm $\mathcal{A}$, we write $z \xleftarrow{\$} \mathcal{A}(x, y, \ldots)$ to indicate that $\mathcal{A}$ is a (probabilistic) algorithm that outputs z on input $(x, y, \ldots)$.

2.1 Digital Signature and Its EUF-CMA Security

A digital signature scheme is defined by the three algorithms, Gen, Sign, and Verify. The key generation algorithm Gen generates a keypair $(PK, sk) \xleftarrow{\$} \mathsf{Gen}(1^\lambda)$ for a secret key sk and a public key PK. The signing algorithm Sign inputs a message and the secret key, and returns a signature $\sigma \xleftarrow{\$} \mathsf{Sign}(sk, M)$ of the message. The verification algorithm Verify takes a public key and a message with a corresponding signature as input, and returns $\top$ or $\bot$, indicating "accept" or "reject", respectively. We require the usual correctness properties.

We recall the EUF-CMA experiment played by a challenger and a forger $\mathcal{F}$. First, the challenger runs $(PK, sk) \xleftarrow{\$} \mathsf{Gen}(1^\lambda)$ and $\mathcal{F}$ is given PK. Proceeding adaptively, $\mathcal{F}$ requests signatures on messages $M_1, \ldots, M_q \in \{0,1\}^*$ under PK. The challenger responds to each query with a signature $\sigma_i \xleftarrow{\$} \mathsf{Sign}(sk, M_i)$. Eventually, $\mathcal{F}$ outputs the pair (M^*, σ^*). We say that the adversary wins the game if $\mathsf{Verify}(M^*, \sigma^*, PK) = \top$ and $M^* \notin \{M_1, \ldots, M_q\}$. We say that $\mathcal{F}$ (t, q, ϵ)-breaks the EUF-CMA security of the signature if $\mathcal{F}$ runs in time t, makes at most q signing queries, and has success probability ϵ. We say that the signature scheme is EUF-CMA secure if ϵ is negligible for any probabilistic polynomial-time algorithm $\mathcal{F}$.

2.2 Prime Numbers, the RSA-Assumption, and Generalized Birthday Bounds

For $x \in \mathbb{N}$ let $\pi(x)$ denote the number of primes between 0 and x. The following lemma is a direct consequence of Chebyshev's bounds on $\pi(x)$.

Lemma 1. $\frac{x}{\log_2 x} < \pi(x) < \frac{2x}{\log_2 x}$

We say that a prime p is a safe prime, if $p = 2p' + 1$ and p' is also prime. Let p and q be two randomly chosen r'-bit safe primes, and let $N = pq$. Let $e \in \mathbb{Z}_{\phi(N)}$ be a random odd prime with $e \neq p', q'$. We say that an algorithm $\mathcal{A}$ (t, ϵ)-breaks the RSA assumption, if $\mathcal{A}$ runs in time t and $\Pr[y^{1/e} \xleftarrow{\$} \mathcal{A}(N, e, y)] \geq \epsilon$. As discussed in the previous papers [12,13,10], the definition above is equivalent to a more standard version of the RSA assumption where $e \in \mathbb{Z}_{\phi(N)}$ is a random integer relatively prime to $\phi(N)$ with only polynomial loss in reduction cost. We say that an algorithm $\mathcal{A}$ (t, ϵ)-breaks the RSA assumption, if $\mathcal{A}$ runs in time t and non-negligible ϵ.

We denote with $\mathbb{QR}_N$ the group of quadratic residues modulo N. We recall the following lemmas which is needed for the security proof of our constructions.

Lemma 2. *([17],[3]) There is an efficient algorithm that, on input $y, z \in \mathbb{Z}_N$ and integer $e, f \in \mathbb{Z}$ such that $\gcd(e, f) = 1$ and $z^e \equiv y^f \mod N$, computes $x \in \mathbb{Z}_N$ satisfying $x^e \equiv y \mod N$.*

Lemma 3. *([10]) Let A be a set with $|A| = a$. Let $X_1, \ldots, X_q$ be q independent random variables, taking uniformly random values from A. Then the probability that there exists $m + 1$ pairwise distinct indices $i_1, \ldots, i_{m+1}$ such that $X_{i_1} = \cdots = X_{i_{m+1}}$ is upper bounded by $\frac{q^{m+1}}{a^m}$.*

3 Overview of the Idea of Our Constructions

Here, we explain an underlying idea of our constructions. It is known that the combination of a weakly secure signature scheme and a one-time signature scheme yields an EUF-CMA secure signature scheme. It can be seen as a variant of the generic construction of an EUF-CMA secure signature from a weakly secure signature scheme and a chameleon hash function [14,18]. It would be interesting to consider what would happen if we used an m-time signature scheme instead of a one-time signature scheme in the above. Even in this case, we can obtain an analogous construction of a signature scheme as we explain below.

The public key of the scheme is $(pk_w, vk_m^{(1)}, \ldots, vk_m^{(2^\eta)})$ where pk_w is the public key of the weakly secure signature scheme and all $vk_m^{(s)}$ $(s \in [2^\eta])$ are verification keys of the m-time signature. The secret key of the scheme is $(sk_w, sk_m^{(1)}, \ldots, sk_m^{(2^\eta)})$ where sk_w is the secret key corresponding to pk_w and $sk_m^{(s)}$ $(s \in [2^\eta])$ are secret keys corresponding to $vk_m^{(s)}$. To sign a message M, a signer first picks a random bit string s with length η by $s \xleftarrow{\$} [2^\eta]$. Then, the signer computes signature σ_w on "message" s by the signing algorithm of the weakly secure signature scheme. The signer also computes signature σ_m on M for $vk_m^{(s)}$ using $sk_m^{(s)}$. The final signature is $\sigma = (\sigma_w, \sigma_m, s)$. The verification algorithm simply checks the validity of σ_w and σ_m.

In fact, the above idea does not work without change. This is simply because 2^η is exponentially large and the above construction needs an exponential number of public keys. Nevertheless, the idea has a potential advantage over the previous generic constructions. That is, if we take larger m (for example $m = 4$), then we can take smaller η. Concretely, we can take $\eta = \lg(q) + \lambda/m$ where q is the upper bound of the number of signing queries issued by the adversary and λ is the security parameter [10]. Since the size of the signature is $|s|(= \eta)+|\sigma_w|+|\sigma_m|$, this considerably reduces the size of the signature.

The generic construction of short signature schemes from a $(m, 1)$-programmable hash function proposed by [11,10] can be seen as a realization of the above informal idea. In fact, one can obtain an m-time signature from $(m, 1)$-programmable hash function as suggested in [10]. Since m-time signatures form a wider class than $(m, 1)$-programmable hash functions, we can obtain various short signature schemes that cannot be captured by the generic construction by [11,10].

Based on the above idea, we construct three short signature schemes from the RSA assumption which are presented in section 4, 5, and 6. Specifically, signature size of our first scheme in section 4 is the same as that of the best known scheme [10], and furhtermore its public key size is significantly shorter than that of [10]. Moreover, our second scheme in section 5 yields the same signature size and better computational efficiency by admitting larger public key size, and our third scheme in section 6 yields further better computational efficiency by only slightly increasing signature size and public key size (compared with our first scheme).

4 Our First Scheme

4.1 Basic Idea

As we discussed in the precious section, one possible approach to constructing a (fully-fledged) short signature scheme is to combine an m-time signature scheme and weakly secure signature scheme. We use the weakly secure signature scheme proposed by [13] in this paper.

One possible choice of m-time signature would be the RSA-based m-time signature considered in [4]. In fact, the construction of $\mathsf{Sig}_{\mathsf{RSA}}[\mathsf{H}_{\mathsf{cfs}}]$ proposed in [10] is closely related to the m-time signature in [4]. Since this choice of an m-time signature leads to a signature scheme with huge public key size, we do not use the scheme here. Instead, we construct a new m-time signature and propose a fully-fledged short signature scheme based on it.

Here, we explain our m-time signature. The verification key of our scheme vk_m consists of the odd primes $e_1, \ldots, e_d$, the product of large two primes $N = pq$, and $h \in \mathbb{Z}_N^*$. Let S be a map $S : \mathcal{M} \to 2^{[d]}$ where $\mathcal{M} = \{0,1\}^l$ is the message space. We assume that for all $M^*, M_1, \ldots, M_m \in \mathcal{M}$, it holds that $S(M^*) \not\subseteq \cup_{i=1}^m S(M_i)$ if $M^* \notin \{M_1, \ldots, M_m\}$. We remark that a map S with such a property can be constructed from an m-cover free family [6,15]. The signature on a message M is $\sigma_m = h^{1/\prod_{i \in S(M)} e_i}$. The verification algorithm checks whether $\sigma_m^{\prod_{i \in S(M)} e_i} \stackrel{?}{=} h$.

We now explain how to combine our m-time signature with weakly secure scheme in [13]. We need an exponentially large number of verification keys $vk_m^{(1)}, \ldots, vk_m^{(2^\eta)}$ where $vk_m^{(s)} = (e_1^{(s)}, \ldots, e_d^{(s)})$ if we apply the idea described in the previous section straightforwardly. We can resolve this problem by a technique from [12,13]. That is, we prepare a pseudorandom function F that can be computed publicly and let $e_j^{(s)} = \mathsf{F}(s||j)$ where $s||j$ is the concatenation of s and j. Since all $\{e_1^{(s)}, \ldots, e_d^{(s)}\}_{s \in [2^\eta]}$ can be computed from F, we do not need these elements in the public key. The public key size of our scheme becomes very short by this idea. Another problem to consider is that the signature $\sigma = (\sigma_m, \sigma_w, s)$ is still longer than that of previous schemes. In the construction below, we reduce the signature length by using the algebraic structure of σ_m and σ_w. As a result, we obtain a signature scheme that achieves the shortest signature length and public key length simultaneously among the signature schemes from the RSA assumption [13,10].

4.2 Construction

Let S be a map $S : \mathcal{M} \to 2^{[d]}$ where $\mathcal{M} = \{0,1\}^l$ is the message space. We assume that for all $M^*, M_1, \ldots, M_m \in \mathcal{M}$ it holds that $S(M^*) \not\subseteq \cup_{i=1}^m S(M_i)$ if $M^* \notin \{M_1, \ldots, M_m\}$. Such S can be constructed using a cover free family [6,15] if $d \geq 16m^2 l$. See Appendix A for the details. We define the scheme as follows.

$\mathsf{Gen}(1^\lambda)$**:** It picks two large safe r'-bit primes p and q, and sets $N = pq$. Then it chooses a random key K for the pseudorandom function $\mathsf{PRF} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^r$ and picks $c \xleftarrow{\$} \{0,1\}^r$, where $r = \lfloor \log_2 N \rfloor - 1$. These values define a function F as $\mathsf{F}(z) = \mathsf{PRF}_K(\mu, z) \oplus c$ where μ, called the resolving index of z, denotes the smallest positive integer such that $\mathsf{PRF}_K(\mu, z) \oplus c$ is an odd prime. Here $\oplus$ denotes the bit-wise XOR operation, and we interpret r-bit string returned by F as an integer in the obvious way. Finally, it picks $h \xleftarrow{\$} \mathbb{Z}_N^*$. The public key is $PK = (N, h, K, c)$, the secret key is $sk = (PK, p, q)$.

In the following, we define $\mathsf{P} : \{0,1\}^\eta \to \mathbb{N}$ as $\mathsf{P}(s)$ for $\mathsf{P}(s) = \prod_{i=1}^\eta \mathsf{F}(s|_i)$ where $s|_i$ is the i-th prefix of s, i.e., the bit string consisting of the first i bits of s. We also define $s|_0 = \emptyset$, where $\emptyset$ is the empty string, for technical reasons. We define another function $\mathsf{Q} : \{0,1\}^\eta \times 2^{[d]} \to \mathbb{N}$ as $\mathsf{Q}(s, S) = \prod_{i \in S} \mathsf{F}(s||i)$ where S is a subset of $[d]$ and $s||i$ denotes concatenation of a bit string s and $i \in [d]$. In this case, we regard i as a bit string.

$\mathsf{Sign}(sk, M)$**:** It first picks random $s \xleftarrow{\$} \{0,1\}^\eta$ and computes $\mathsf{F}(t)$ for $t \in (\cup_{i \in [\eta]} \{s|_i\}) \cup (\cup_{i \in S(M)} \{s||i\})$. If the resolving index of t is more than r^2 or $\mathsf{F}(t)$ divides $\phi(N)$ for some $t \in (\cup_{i \in [\eta]} \{s|_i\}) \cup (\cup_{i \in S(M)} \{s||i\})$, then it outputs $((p,q), s)$.[1] Otherwise it computes

$$\sigma = h^{1/\mathsf{P}(s)\mathsf{Q}(s, S(M))},$$

[1] The probability of these events happen is negligible as proven in the security proof of the scheme. Thus this step can be ignored in practice.

where inverse of $\mathsf{P}(s)\mathsf{Q}(s, S(M))$ is computed modulo the order $\phi(N) = (p-1)(q-1)$ of the multiplicative group $\mathbb{Z}_N^*$. The signature is (σ, s).

$\mathsf{Verify}(M, (\sigma, s), PK)$**:** Given a signature (σ, s), it first checks whether resolving index of t is more than r^2 or $2\mathsf{F}(t)+1$ divides N (which is equivalent to $\mathsf{F}(t)$ divides $\phi(N)$) for some $t \in (\cup_{i\in[\eta]}\{s|_i\}) \cup (\cup_{i\in S(M)}\{s||i\})$. If it holds, it outputs $\top$ if $\sigma = (p, q)$ and otherwise $\bot$. Next, it returns $\top$ if

$$\sigma^{\mathsf{P}(s)\mathsf{Q}(s,S(M))} = h.$$

Otherwise it returns $\bot$.

Correctness. The correctness can be verified by the following equation:

$$\sigma^{\mathsf{P}(s)\mathsf{Q}(s,S(M))} = h^{\mathsf{P}(s)\mathsf{Q}(s,S(M))/\mathsf{P}(s)\mathsf{Q}(s,S(M))} = h.$$

4.3 Security

In this subsection, we prove the following theorem which establishes the security of the scheme.

Theorem 1. *Let* PRF *be a* (t'', ϵ'')*-secure pseudo-random function. Suppose there exists a forger* $\mathcal{F}$ *who* (t, q, ϵ)*-breaks the EUF-CMA security of the above scheme. Then there exists an adversary* $\mathcal{A}$ *that* (t', ϵ')*-breaks the RSA assumption with* $t \approx t'$ *and* $\epsilon \leq (q+1)\eta d\big(4r^2\epsilon' + 3\epsilon'' + \frac{r(q+1)^2(\eta+d)^2}{2^{r-1}}\big) + \frac{q^{m+1}}{2^{m\eta}}$.

In the following, let M_k denote the k-th query to the signing oracle, and let (σ_k, s_k) denote the reply. Let (M^*, σ^*, s^*) be the forgery output of $\mathcal{F}$. We distinguish between two types of forgers. A type1 forger returns (M^*, σ^*, s^*) such that $s^* = s_k$ for some $k \in [q]$. A type2 forger returns (M^*, σ^*, s^*) such that $s^* \neq s_k$ for all $k \in [q]$.

The following lemma proves security against Type1 forger.

Lemma 4. *Let* $\mathcal{F}$ *be a Type1 forger that* (t, q, ϵ)*-breaks the existential unforgeability of our scheme. Then there exists an adversary* $\mathcal{A}$ *that* (t', ϵ')*-breaks the RSA assumption with* $t \approx t'$ *and* $\epsilon' \geq \frac{1}{4r^2}\big(\frac{1}{qd}(\epsilon - \frac{q^{m+1}}{2^{m\eta}}) - 3\epsilon'' - \frac{q(\eta+d)(2r+1+rq(\eta+d))}{2^r}\big)$.

Proof. In the following let X_i denote the probability that $\mathcal{F}$ is successful in Game i and the challenger does not abort.

Game 0. We define Game 0 as the EUF-CMA experiment between the challenger and the forger $\mathcal{F}$. By definition we have $\Pr[X_0] = \epsilon$.

Game 1. In this game, the challenger aborts if there exist at least $m+1$ indices $k_1, \ldots, k_{m+1} \in [q]$ such that $s_k = s_{k'}$ for all $k, k' \in \{k_1, \ldots, k_{m+1}\}$. We denote this event by $\mathsf{Abort}_{\mathsf{mColl}}$. We know $\Pr[\mathsf{Abort}_{\mathsf{mColl}}] \leq \frac{q^{m+1}}{2^{m\eta}}$ from Lemma 3. Thus we have $\Pr[X_1] \geq \Pr[X_0] - \frac{q^{m+1}}{2^{m\eta}}$.

Game 2. In this game, the challenger chooses randomness $s_1, \ldots, s_q$ and guesses $k^* \xleftarrow{\$} [q]$ such that $s_{k^*} = s^*$ at the beginning of the game. The challenger aborts if $\mathcal{F}$ outputs a forgery (M^*, σ^*, s^*) with $s_{k^*} \neq s^*$. Since $s^* \in \{s_i\}_{i=1}^q$, we have $\Pr[X_2] \geq \Pr[X_1]/q$.

Game 3. In this game, the challenger chooses $j^* \xleftarrow{\$} [d]$ before setting the public key and aborts if $j^* \notin S(M^*)$ or $j^* \in S(M_k)$ for some $k \in \{k \mid s_k = s_{k^*}\}$. Recall that $\{k \mid s_k = s_{k^*}\} \leq m$, so $S(M^*) \not\subseteq \cup_{k \in \{k|s_k = s_{k^*}\}} S(M_k)$ from the property of S. Thus there exists at least one $j' \in [d]$ such that $j' \in S(M^*)$ and $j' \notin S(M_k)$ for all $k \in \{k \mid s_k = s_{k^*}\}$. We have $\Pr[X_3] \geq \Pr[X_2]/d$.

Game 4. Let E_{all} be $E_{all} = (\cup_{(i,j)\in[q]\times[\eta]}\{s_i|_j\}) \cup (\cup_{(i,j)\in[q]\times[d]}\{s_i||j\})$ in the following. The challenger in this game proceeds just like the challenger in the previous game, except that we add an abort condition. The challenger aborts if for some $t \in E_{all}$, the resolving index μ is greater than r^2. We denote this event with Abort_μ. Let us assume PRF_K is replaced with a truly random function, and let us consider the probability of not finding a prime by evaluating the random function r^2 times and computing the exclusive or with c. This is equivalent to sampling r^2 uniform r-bit strings. Lemma 1 tells us that the probability of finding a prime by sampling r random bits is at least $1/r$, thus the probability of not finding a prime in r^2 trials is at most $(1-1/r)^{r^2}$. Since the challenger has to compute F at most $q(\eta+d)$ times, so we can therefore construct an adversary distinguishing PRF_K from a random function with probability at least $\epsilon_\mu \geq \mathsf{Abort}_\mu - q(\eta+d)(1-1/r)^{r^2} \geq \mathsf{Abort}_\mu - q(\eta+d)/2^r$, where the latter inequality uses that $(1-1/r)^r \leq 1/2$ for all $r \in \mathbb{N}$. Since we must have $\epsilon_\mu \leq \epsilon''$, this implies $\Pr[X_4] \geq \Pr[X_3] - \epsilon'' - q(\eta+d)/2^r$.

Game 5. In this game, the challenger aborts if there exists $t \in E_{all}$ such that $\mathsf{F}(t)$ divides $\phi(N)$. We denote this event by $\mathsf{Abort}_{\mathsf{div}}$. Recall that $\phi(N) = 4p'q'$ and that F returns only odd primes. Again replacing PRF_K with a truly random function, the probability that one out of at most $q(\eta+d)$ randomly chosen odd r-bit primes equals one of the two odd primes dividing $\phi(N)$ is at most $(q(\eta+d)2r)/2^r$ by Lemma 1. Now consider the case where the truly random function is instantiated with PRF_K, and suppose that a collision occurs with probability $\Pr[\mathsf{Abort}_{\mathsf{div}}]$. Then this would allow an attack distinguishing PRF_K from a random function with probability at least $\epsilon_{\mathsf{div}} \geq \Pr[\mathsf{Abort}_{\mathsf{div}}] - (q(\eta+d)2r)/2^r$. Since we have $\epsilon_{\mathsf{div}} \leq \epsilon''$, this implies $\Pr[X_5] \geq \Pr[X_4] - \Pr[\mathsf{Abort}_{\mathsf{div}}] \geq \Pr[X_4] - \epsilon'' - (q(\eta+d)2r)/2^r$.

Game 6. In the following, let $E = \cup_{i=1}^{q}\{s_i\}$, $E^* = E\backslash\{s_{k^*}\}$. In this game the challenger picks $y \xleftarrow{\$} \mathbb{Z}_N^*$ and sets h by

$$h = y^{\mathsf{P}(s_{k^*})\mathsf{Q}(s_{k^*},[d]\backslash\{j^*\})\cdot\prod_{t\in E^*}\mathsf{P}(t)\mathsf{Q}(t,[d])}.$$

The distribution of the public key is unchanged from the previous game. This change is only conceptual, so we have $\Pr[X_6] = \Pr[X_5]$.

Game 7. Now the challenger computes a signature σ_k on some chosen-message M_k as

$$\sigma_k = \begin{cases} y^{\mathsf{Q}(s_{k^*},[d]\backslash\{S(M_k)\cup\{j^*\}\})\cdot\prod_{t\in E^*}\mathsf{P}(t)\mathsf{Q}(t,[d])} & s_k = s_{k^*} \\ y^{\mathsf{P}(s_{k^*})\mathsf{Q}(s_{k^*},[d]\backslash\{j^*\})\cdot\mathsf{Q}(s_k,[d]\backslash S(M_k))\cdot\prod_{t\in E_k^*}\mathsf{P}(t)\mathsf{Q}(t,[d])} & s_k \neq s_{k^*} \end{cases}$$

where $E_k^* = E^* \backslash \{s_k\}$. It is easy to check that $\sigma_k = h^{1/\mathsf{P}(s_k)\mathsf{Q}(s_k,S(M_k))}$. In the above, we used the fact that $j^* \notin S(M_k)$ if $s_k = s_{k^*}$. This change is only conceptual, so we have $\Pr[X_7] = \Pr[X_6]$.

Game 8. The challenger in this game aborts if there exists $t, t' \in E_{all}$ such that $\mathsf{F}(t) = \mathsf{F}(t')$ and $t \neq t'$. This event is denoted with $\mathsf{Abort}_{\mathsf{col}}$. Recall that $\mathsf{F}(z) = \mathsf{PRF}_K(\mu, z) \oplus c$, where μ is incremented until $\mathsf{PRF}_K(\mu, z) \oplus c$ is prime. Let us again assume PRF_K is replaced with a truly random function. Then evaluating F is equivalent to sampling a uniformly random r-bit prime. There are at least $2^r/r$ such primes by Lemma 1, and at most $q(\eta + d)$ primes are sampled. Applying Lemma 3, we conclude that the collision probability for a truly random function is at most $rq^2(\eta + d)^2 \cdot 2^{-r}$. Now consider the case where the truly random function is instantiated with PRF_K, and suppose that a collision occurs with probability $\Pr[\mathsf{Abort}_{\mathsf{col}}]$. Then this would allow an attack distinguishing PRF_K from a random function with probability at least $\epsilon_{\mathsf{col}} \geq \Pr[\mathsf{Abort}_{\mathsf{col}}] - rq^2(\eta+d)^2/2^r$. Since we have $\epsilon_{\mathsf{col}} \leq \epsilon''$, this implies $\Pr[X_8] = \Pr[X_7] - \Pr[\mathsf{Abort}_{\mathsf{col}}] \geq \Pr[X_7] - \epsilon'' - rq^2(\eta + d)^2/2^r$.

Game 9. In this game, the challenger chooses $\mu^* \xleftarrow{\$} [r^2]$ in advance and aborts if μ^* is not resolving index of $s_{k^*}||j^*$. Due to the changes introduced in the Game 4 we know that the resolving index of $s_{k^*}||j^*$ lies in the interval $[1, r^2]$. Thus we have $\Pr[X_9] \geq \Pr[X_8]/r^2$.

Game 10. Recall that c is uniformly distributed, and we abort if μ^* is not the resolving index of $s_{k^*}||j^*$. The latter implies that $\mathsf{PRF}(\mu^*, s_{k^*}||j^*)$ is prime, thus e has the distribution of uniformly random prime. In this game, the challenger determines c differently. Instead of sampling c at random, the challenger sets $c = \mathsf{PRF}(\mu^*, s_{k^*}||j^*) \oplus e$, where e is the random r-bit prime the challenger chooses. Observe that this defines $\mathsf{F}(s_{k^*}||j^*) = e$. The distribution of μ^*, c, and e is not altered. Thus We have $\Pr[X_{10}] = \Pr[X_9]$.

The RSA Adversary. We replace the challenger in Game 10 with RSA adversary $\mathcal{A}$. $\mathcal{A}$ receives a RSA challenge (N', e', y') as input and aborts if e' is not an odd prime or $e' > 2^r$. Otherwise $\mathcal{A}$ sets $N = N', e = e'$ and proceeds like the challenger in Game 10. Recall that $s_{k^*} = s^*$, $\mathsf{F}(s^*||j^*) = e$ and $j^* \in S(M^*)$. Otherwise $\mathcal{A}$ aborts as the challenger does in Game 10. Since we have set $r = \lfloor \log_2 N \rfloor - 1$, the probability that $e \geq 2^r$ is at most $1/4$. Thus, the success probability of $\mathcal{A}$ is at least $\Pr[X_{10} \wedge e < 2^r] \geq \frac{1}{4}\Pr[X_{10}]$.

Answering the Signing Queries. Due to the changes introduced in the Games 4 to 7, $\mathcal{A}$ can answer signing queries without the knowledge of the factorization of N.

Extracting the Solution to the RSA Challenge. Eventually, $\mathcal{F}$ returns a forgery (M^*, σ^*, s^*), from which $\mathcal{A}$ extracts the solution to the RSA challenge as follows. First observe that

$$\begin{aligned}\sigma^* &= h^{1/\mathsf{P}(s^*)\mathsf{Q}(s^*,S(M^*))} = y^{\left(\prod_{t\in E^*}\mathsf{P}(t)\mathsf{Q}(t,[d])\right)\cdot\mathsf{Q}(s^*,[d]\backslash\{j^*\})/\mathsf{Q}(s^*,S(M^*))} \\ &= y^{\left(\prod_{t\in E^*}\mathsf{P}(t)\mathsf{Q}(t,[d])\right)\cdot\mathsf{Q}(s^*,[d]\backslash S(M^*))/\mathsf{F}(s^*||j^*)} \\ &= y^{\left(\prod_{t\in E^*}\mathsf{P}(t)\mathsf{Q}(t,[d])\right)\cdot\mathsf{Q}(s^*,[d]\backslash S(M^*))/e} = y^{f/e}\end{aligned}$$

where $f = \mathsf{Q}(s^*, [d] \backslash S(M^*)) \cdot \prod_{t \in E^*} \mathsf{P}(t)\mathsf{Q}(t, [d])$. Then we can see that $(\sigma^*)^e = y^f$ holds from the above equation. Furthermore, $\gcd(e, f) = 1$ by Game 8. Thus we can apply Lemma 2 and extract $y^{1/e}$ which is answer to the RSA challenge from σ^*.

The following lemma proves security against Type2 forger.

Lemma 5. *Let $\mathcal{F}$ be a Type2 forger that (t, q, ϵ)-breaks the existential unforgeability of our scheme. Then there exists an adversary $\mathcal{A}$ that (t', ϵ')-breaks the RSA assumption with $t \approx t'$ and $\epsilon' \geq \frac{1}{4r^2}\left(\frac{1}{(q+1)\eta}\left(\epsilon - 2\epsilon'' - \frac{q(\eta+d)(2r+1)}{2^r}\right) - 2\epsilon'' - \frac{r(q+1)^2(\eta+d)^2+1}{2^r}\right)$.*

Proof. Let X_i denote the probability that $\mathcal{F}$ is successful in Game i and the challenger does not abort.

Game 0. We define Game 0 as the EUF-CMA experiment between the challenger and the forger $\mathcal{F}$. By definition we have $\Pr[X_0] = \epsilon$.

Game 1. Let E_{all} be $E_{all} = (\cup_{(i,j) \in [q] \times [\eta]} \{s_i|_j\}) \cup (\cup_{(i,j) \in [q] \times [d]} \{s_i||j\})$ in the following. In this game, the challenger aborts if for some $t \in E_{all}$, the resolving index μ is greater than r^2. As the proof of Lemma 4, we have $\Pr[X_1] \geq \Pr[X_0] - \epsilon'' - q(\eta + d)/2^r$.

Game 2. In this game, the challenger aborts if there exists $t \in E_{all}$ such that $\mathsf{F}(t)$ divides $\phi(N)$. As the proof of Lemma 4, we have $\Pr[X_2] \geq \Pr[X_1] - \epsilon'' - (q(\eta + d)2r)/2^r$.

Game 3. In this game, the challenger chooses the randomness $s_1, \ldots, s_q$ in advance. Let $E = \cup_{i=1}^{q} \{s_i\}$. The challenger picks $y \xleftarrow{\$} \mathbb{Z}_N^*$ and sets public key h by

$$h = y^{\prod_{t \in E} \mathsf{P}(t)\mathsf{Q}(t,[d])}.$$

The distribution of the public key is unchanged from the previous game. This change is only conceptual, so we have $\Pr[X_3] = \Pr[X_2]$.

Game 4. Now the challenger computes a signature σ_k on some chosen-message M_k as

$$\sigma_k = y^{\mathsf{Q}(s_k,[d]\backslash S(M_k)) \prod_{t \in E_k} \mathsf{P}(t)\mathsf{Q}(t,[d])}$$

where $E_k = E \backslash \{s_k\}$. It is easy to check that $\sigma_k = h^{1/\mathsf{P}(s_k)\mathsf{Q}(s_k, S(M_k))}$. This change is only conceptual, so we have $\Pr[X_4] = \Pr[X_3]$.

Game 5. In this game the challenger guesses the shortest prefix of s^* that differs from all prefixes of $s_1, \ldots, s_q$. Note that this prefix must exist, because the Type2-forger will return a forgery (M^*, σ^*, s^*) with $s^* \notin \{s_1, \ldots, s_q\}$. To this end, the challenger proceeds as follows. If $q = 0$, it samples a bit $\psi \xleftarrow{\$} \{0, 1\}$ at random, and aborts if the forger returns s^* with $s^*|_1 \neq \psi$. If $q \geq 0$, the challenger picks $i \in [q]$ and $j \in [\eta]$ and sets $\psi = s_i|_{j-1}||b$, where b is the complement of the j-th bit of s_i. (Recall that we defined the 0-th prefix as the empty string $\emptyset$, thus $s_i|_0 = \emptyset$.) The challenger aborts if either ψ is a prefix of some $s_i \in \{s_1, \ldots, s_q\}$, that is, there exists (i', j') such that $\psi = s_{i'}|_{j'}$, or if the forger returns (M^*, σ^*, s^*) such that ψ is not a prefix of s^*. If $q = 0$,

then the challenger aborts with probability 1/2. Otherwise there are at most $q\eta$ possible choices of ψ. Thus we have $\Pr[X_5] = \Pr[X_4]/(q+1)\eta$.

Game 6. We add an abort condition. If $\mathsf{F}(\psi)|\prod_{t\in E}\mathsf{P}(t)\mathsf{Q}(t,[d])$, or (equivalently) $\mathsf{F}(\psi) = \mathsf{F}(t)$ for some $t \in E_{all}$, then the challenger aborts. Note that $\psi \neq t$ for all $t \in E_{all}$. As the proof of Lemma 4, we have $\Pr[X_6] \geq \Pr[X_5] - \epsilon'' - r(q+1)^2(\eta+d)^2/2^r$.

Game 7. We introduce a number of changes to the challenger.

- The challenger aborts if the resolving index of ψ is greater than r^2.
- The challenger guesses resolving index of ψ as $\mu^* \xleftarrow{\$} [r^2]$ and aborts if μ^* is not the resolving index of ψ.
- Instead of sampling c at random, we set $c = \mathsf{PRF}(\mu^*, \psi) \oplus e$, where e is the uniformly random r-bit prime that the challenger chooses.

With the same arguments as the proof of Lemma 4 we have $\Pr[X_7] \geq 1/r^2(\Pr[X_6] - \epsilon'' - 1/2^r)$.

The RSA Adversary. We replace the challenger in Game 7 with the RSA adversary $\mathcal{A}$. $\mathcal{A}$ receives an RSA challenge (N', e', y') as input and aborts if e' is not an odd prime or $e' > 2^r$. Otherwise $\mathcal{A}$ sets $N = N', e = e'$ and proceeds like the challenger in Game 7. Recall that we have $\mathsf{F}(\psi) = e$ now. As the proof of Lemma 4, the success probability of $\mathcal{A}$ is at least $\frac{1}{4}\Pr[X_7]$.

Answering the Signing Queries. Due to the changes introduced in the Game 1 to 4, $\mathcal{A}$ can answer signing queries without the knowledge of the factorization of N.

Extracting the Solution to the RSA Challenge. Eventually, $\mathcal{F}$ returns a forgery (M^*, σ^*, s^*), from which $\mathcal{A}$ extracts the solution to the RSA challenge as follows. In the case where resolving index of t is more than r^2 or $\mathsf{F}(t)|\phi(N)$ for some $t \in (\cup_{i\in[\eta]}\{s^*|_i\}) \cup (\cup_{i\in S(M^*)}\{s^*||i\})$, $\sigma = (p, q)$ if $\mathcal{F}$ is successful. Thus $\mathcal{A}$ can efficiently compute $y^{1/e}$ from the output of $\mathcal{F}$ in this case. Otherwise,

$$\begin{aligned}\sigma^* &= h^{1/\mathsf{P}(s^*)\mathsf{Q}(s^*,S(M^*))} = y^{\left(\prod_{t\in E}\mathsf{P}(t)\mathsf{Q}(t,[d])\right)/\mathsf{P}(s^*)\mathsf{Q}(s^*,S(M^*))} \\ &= y^{\left(\prod_{t\in E}\mathsf{P}(t)\mathsf{Q}(t,[d])\right)/\left(z\cdot\mathsf{F}(\psi)\right)} = y^{\left(\prod_{t\in E}\mathsf{P}(t)\mathsf{Q}(t,[d])\right)/ez}\end{aligned}$$

where $z = \mathsf{Q}(s^*, S(M^*)) \cdot \prod_{\{i\in[\eta]|s^*|_i\neq\psi\}}\mathsf{F}(s^*|_i)$ holds. Thus we have $\left((\sigma^*)^z\right)^e = y^{\prod_{t\in E}\mathsf{P}(t)\mathsf{Q}(t,[d])}$. Since $\gcd(e, \prod_{t\in E}\mathsf{P}(t)\mathsf{Q}(t,[d])) = 1$ by Game 6, we can apply Lemma 2 and extract $y^{1/e}$.

5 Our Second Scheme

Our first scheme suffers from its heavy signing and verification algorithms while providing very short public key size. This is because the signing and verification algorithms need the generation of a large number of primes. On the other hand, $\mathsf{Sig}_{\mathsf{RSA}}[\mathsf{H}_{\mathsf{cfs}}]$ in [10] has an opposite property. That is, the signing and verification algorithms are more efficient than our scheme, but the public key size is huge.

In this section, we propose a generalized version of these schemes. The scheme has parameters u_1 and u_2 with condition $u_1 \cdot u_2 = d$ where d is some constant depending on m and λ. If we take u_1 smaller, the public key size becomes smaller and the signing and verification algorithms become heavier. On the other hand, if we take u_1 larger, the public key size becomes larger and the signing and verification algorithm becomes more efficient. Especially, if we set $(u_1, u_2) = (1, d)$, then the scheme can be seen as a slight variant of our first scheme. Similarly, if we set $(u_1, u_2) = (d, 1)$, then we can obtain a slight variant of $\mathsf{Sig}_{\mathsf{RSA}}[\mathsf{H}_{\mathsf{cfs}}]$ in [10]. Furthermore, we can also construct a stateful version of the scheme. The scheme is more efficient than the above scheme except for its slightly larger signature size.

CONSTRUCTION. Let S be a map $S : \mathcal{M} \to 2^{[u_1]\times[u_2]}$ where $\mathcal{M} = \{0,1\}^l$ is the message space. We assume that for all $M^*, M_1, \ldots, M_m \in \mathcal{M}$ it holds that $S(M^*) \not\subseteq \cup_{i=1}^m S(M_i)$ if $M^* \notin \{M_1, \ldots, M_m\}$. Such S can be constructed using cover free family [6,15] as in [20] if $u_1 u_2 \geq 16m^2 l$. See Appendix A for the details. We define the scheme as follows.

$\mathsf{Gen}(1^\lambda)$**:** It picks two large safe r'-bit primes p and q, and sets $N = pq$. Then it chooses a random key K for the pseudorandom function $\mathsf{PRF} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^r$ and picks $c \xleftarrow{\$} \{0,1\}^r$, where $r = \lfloor \log_2 N \rfloor - 1$. These values define functions $\mathsf{F} : \{0,1\}^* \to \mathbb{N}$, $\mathsf{P} : \{0,1\}^\eta \to \mathbb{N}$, and $\mathsf{Q} : \{0,1\}^\eta \times 2^{[u_2]} \to \mathbb{N}$ as in section 4. Finally, it picks $h', h_1, \ldots, h_{u_1} \xleftarrow{\$} \mathbb{QR}_N$. The public key is $PK = (N, h', h_1, \ldots, h_{u_1}, K, c)$, the secret key is $sk = (PK, p, q)$.

$\mathsf{Sign}(sk, M)$**:** It first picks random $s \xleftarrow{\$} \{0,1\}^\eta$ and computes $\mathsf{F}(t)$ for $t \in (\cup_{i\in[\eta]}\{s|_i\}) \cup (\cup_{j\in S'(M)}\{s||j\})$ where $S'(M) = \{j | \exists i, (i,j) \in S(M)\}$. Let $e_j = \mathsf{F}(s||j)$. If the resolving index of t is more than r^2 or $\mathsf{F}(t)$ divides $\phi(N)$ for some $t \in (\cup_{i\in[\eta]}\{s|_i\}) \cup (\cup_{j\in S'(M)}\{s||j\})$, then it outputs $((p,q), s)$. If $\gcd(\mathsf{P}(s), \mathsf{Q}(s, S'(M))) \neq 1$, then it outputs $((p,q), s)$. [2] Otherwise it computes

$$\sigma = (h')^{1/\mathsf{P}(s)} \cdot \prod_{(i,j)\in S(M)} h_i^{1/e_j}$$

where inverse of $\mathsf{P}(s)$ and e_j is computed modulo the order $\phi(N) = (p-1)(q-1)$ of the multiplicative group $\mathbb{Z}_N^*$. The signature is (σ, s).

$\mathsf{Verify}(M, (\sigma, s), PK)$**:** Given a signature (σ, s), it first checks whether resolving index of t is more than r^2 or $2\mathsf{F}(t)+1$ divides N for some $t \in (\cup_{i\in[\eta]}\{s|_i\}) \cup (\cup_{j\in S'(M)}\{s||j\})$, or $\gcd(\mathsf{P}(s), \mathsf{Q}(s, S'(M))) \neq 1$. If one of the above holds, it outputs $\top$ if $\sigma = (p,q)$ and otherwise $\bot$. Next, it returns if

$$\sigma^{\mathsf{P}(s)\mathsf{Q}(s,S'(M))} = (h')^{\mathsf{Q}(s,S'(M))} \cdot \prod_{(i,j)\in S(M)} h_i^{\mathsf{P}(s)\mathsf{Q}(s,S'(M))\setminus\{j\})}.$$

Otherwise it returns $\bot$.

[2] Similarly to our first scheme, this step can be ignored in practice.

Correctness. The correctness can be verified by the following equation:

$$\sigma^{\mathsf{P}(s)\mathsf{Q}(s,S'(M))} = \Big((h')^{1/\mathsf{P}(s)} \cdot \prod_{(i,j)\in S(M)} h_i^{1/\mathsf{F}(s||j)}\Big)^{\mathsf{P}(s)\mathsf{Q}(s,S'(M))}$$
$$= (h')^{\mathsf{Q}(s,S'(M))} \cdot \prod_{(i,j)\in S(M)} h_i^{\mathsf{P}(s)\mathsf{Q}(s,S'(M)\setminus\{j\})}$$

The following theorem establishes the security of the scheme. The theorem can be proven by a similar argument to the proof of Theorem 1 and [10]. We omit the proof due to a lack of space.

Theorem 2. *Let* PRF *be a* (t'', ϵ'')*-secure pseudo-random function. Suppose there exists a forger* $\mathcal{F}$ *who* (t, q, ϵ)*-breaks the EUF-CMA security of the above scheme. Then there exists an adversary* $\mathcal{A}$ *that* (t', ϵ')*-breaks the RSA assumption with* $t \approx t'$ *and* $\epsilon \leq (q+1)\eta u_1 u_2\big(4r^2\epsilon' + 3\epsilon'' + \frac{r(q+1)^2(\eta+u_2)^2}{2^{r-1}}\big) + \frac{q^{m+1}}{2^{m\eta}}$.

Stateful Version of The Scheme. We can also construct a stateful version of the above scheme. The scheme is more efficient than the above scheme except for the signature size, which is slightly larger than the above scheme. There are two reasons why we can obtain a more efficient scheme in the stateful setting. The first reason is that we can remove the computation of $\mathsf{P}(s)$ from the above. Conceptually, it is equivalent to removing the weakly secure signature scheme in [13] from the above construction. Instead, we use a trick from [12], which leads to a more efficient scheme. The second reason is that we can use a one-time signature instead of the m-time signature ($m \geq 2$) in the stateful setting. This reduces the public key and signing and verification cost. See the full version of this paper for the details.

6 Our Third Scheme

While providing a short signature size, the signing and verification algorithms of our first and second schemes are heavy if we want the public key size to be small. In this section, we propose another signature scheme that provides a shorter public key and a more efficient signing and verification algorithm using a chameleon hash-like technique [7,10]. A signature of the scheme is longer than that of our first and second schemes, but still much shorter than that of the scheme in [13]. The structure of the scheme can be seen as a hybrid of our first scheme and $\mathsf{Sig}_{\mathsf{RSA}}[\mathsf{H}_{\mathsf{rand}}]$ in [10]. Compared with $\mathsf{Sig}_{\mathsf{RSA}}[\mathsf{H}_{\mathsf{rand}}]$ in [10], the scheme has the same size of the signatures. The public key size of the scheme is smaller than that of the other scheme, while the signing and verification algorithms are slightly heavier than the other scheme. We can also construct a stateful version of the scheme. The scheme is more efficient than the above scheme except for its slightly larger signature size.

CONSTRUCTION. We define the scheme as follows. In the following, let $[X]_{2^l} \in \mathbb{Z}$ denote a canonical interpretation of a field element $X \in \mathbb{F}_{2^l}$ as an integer between 0 and $2^l - 1$. We assume that X and $[X]_{2^l}$ are efficiently computable from one another. The message space of the scheme is $\mathcal{M} = \{0,1\}^l$.

$\mathsf{Gen}(1^\lambda)$**:** It picks two large safe r'-bit primes p and q, and sets $N = pq$. Then it chooses a random key K for the pseudorandom function $\mathsf{PRF} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^r$ and picks $c \xleftarrow{\$} \{0,1\}^r$, where $r = \lfloor \log_2 N \rfloor - 1$. These values define functions $\mathsf{F} : \{0,1\}^* \to \mathbb{N}$, $\mathsf{P} : \{0,1\}^\eta \to \mathbb{N}$, and $\mathsf{Q} : \{0,1\}^\eta \times 2^{[2m]} \to \mathbb{N}$ as in section 4. Finally, it picks $h', h_1, \ldots, h_m \xleftarrow{\$} \mathbb{QR}_N$. The public key is $PK = (N, h', h_1, \ldots, h_m, K, c)$, the secret key is $sk = (PK, p, q)$.

$\mathsf{Sign}(sk, M)$**:** It first picks random $s \xleftarrow{\$} \{0,1\}^\eta$, $\rho \xleftarrow{\$} \{0,1\}^l$ and computes $\mathsf{F}(t)$ for $t \in (\cup_{i\in[\eta]}\{s|_i\}) \cup (\cup_{j\in[2m]}\{s||j\})$. Let $e_i = \mathsf{F}(s||i)$. If the resolving index of t is more than r^2 or $\mathsf{F}(t)$ divides $\phi(N)$ for some $t \in (\cup_{i\in[\eta]}\{s|_i\}) \cup (\cup_{j\in[2m]}\{s||j\})$, then it outputs $((p,q), s, \rho)$. If $\gcd(\mathsf{P}(s), \mathsf{Q}(s,[2m])) \neq 1$, then it outputs $((p,q), s, \rho)$. [3] Otherwise it computes

$$\sigma = (h')^{1/\mathsf{P}(s)} \cdot \prod_{i\in[2m]} \Big(h_0 \prod_{j\in[m]} h_j^{([iM+\rho]_{2^l})^j}\Big)^{1/e_i}$$

where inverse of $\mathsf{P}(s)$ and e_i is computed modulo the order $\phi(N) = (p-1)(q-1)$ of the multiplicative group $\mathbb{Z}_N^*$. The signature is (σ, s, ρ).

$\mathsf{Verify}(M, (\sigma, s), PK)$**:** Given a signature (σ, s, ρ), it first checks whether resolving index of t is more than r^2 or $2\mathsf{F}(t)+1$ divides N for some $t \in (\cup_{i\in[\eta]}\{s|_i\}) \cup (\cup_{j\in[2m]}\{s||j\})$, or $\gcd(\mathsf{P}(s), \mathsf{Q}(s,[2m])) \neq 1$. If one of the above holds, it returns $\top$ if $\sigma = (p,q)$ and otherwise $\bot$. Next, it returns $\top$ if

$$\sigma^{\mathsf{P}(s)\mathsf{Q}(s,[2m])} = (h')^{\mathsf{Q}(s,[2m])} \cdot \prod_{i\in[2m]} \Big(h_0 \prod_{j\in[m]} h_j^{([iM+\rho]_{2^l})^j}\Big)^{\mathsf{P}(s)\mathsf{Q}(s,[2m]\setminus\{i\})}.$$

Otherwise it returns $\bot$.

CORRECTNESS. The correctness can be verified by the following equation:

$$\begin{aligned}\sigma^{\mathsf{P}(s)\mathsf{Q}(s,[2m])} &= \Big((h')^{1/\mathsf{P}(s)} \cdot \prod_{i\in[2m]} \Big(h_0 \prod_{j\in[m]} h_j^{([iM+\rho]_{2^l})^j}\Big)^{1/\mathsf{F}(s||i)}\Big)^{\mathsf{P}(s)\mathsf{Q}(s,[2m])} \\ &= (h')^{\mathsf{Q}(s,[2m])} \cdot \prod_{i\in[2m]} \Big(h_0 \prod_{j\in[m]} h_j^{([iM+\rho]_{2^l})^j}\Big)^{\mathsf{P}(s)\mathsf{Q}(s,[2m]\setminus\{i\})}\end{aligned}$$

The following theorem establishes the security of the scheme. The theorem can be proven by a similar argument to the proof of Theorem 1 and [10]. We omit the proof due to a lack of space.

Theorem 3. *Let* PRF *be a* (t'', ϵ'')*-secure pseudo-random function. Suppose there exists a forger* $\mathcal{F}$ *who* (t, q, ϵ)*-breaks the EUF-CMA security of the above scheme. Then there exists an adversary* $\mathcal{A}$ *that* (t', ϵ')*-breaks the RSA assumption with* $t \approx t'$ *and* $\epsilon \leq 4r^2(q+1)\eta\big(\epsilon' + 3\epsilon'' + \frac{m}{2^{r/2}} + \frac{r(q+1)^2(\eta+u_2)^2}{2^{r-1}}\big) + \frac{q^{m+1}}{2^{m\eta}}$.

[3] Similarly to our first scheme, this step can be ignored in practice.

Table 1. Comparison of signature schemes based on the RSA assumption

Signature scheme	Signature size (bits)	Public key size (bits)	Efficiency
Hohenberger-Waters *	$2 \times \lvert\mathbb{Z}_N\rvert = 2048$	$\lvert\mathbb{Z}_N\rvert + \lvert c\rvert + \lvert pk_{ch}\rvert = 4$k	$160 \times \mathsf{P}_{1024}$
$\mathsf{Sig}_{RSA}[\mathsf{H}_{\mathsf{Wat}}]$, [10]	$\lvert\mathbb{Z}_N\rvert + \lvert s\rvert = 1094$	$l \times \lvert\mathbb{Z}_N\rvert + \lvert c\rvert = 161$k	$70 \times \mathsf{P}_{1024}$
$\mathsf{Sig}_{RSA}[\mathsf{H}_{\mathsf{cfs}}]$, $(m = 4)$ [10]	$\lvert\mathbb{Z}_N\rvert + \lvert s\rvert = 1074$	$16m^2 l \times \lvert\mathbb{Z}_N\rvert + \lvert c\rvert = 40$m	$50 \times \mathsf{P}_{1024}$
Ours in Sec. 4, $(m = 4)$	$\lvert\mathbb{Z}_N\rvert + \lvert s\rvert = 1074$	$\lvert\mathbb{Z}_N\rvert + \lvert c\rvert = 2$k	$2610 \times \mathsf{P}_{1024}$
Ours in Sec. 5, $(m = 4)$ $(u_1, u_2) = (410, 100)$	$\lvert\mathbb{Z}_N\rvert + \lvert s\rvert = 1074$	$u_1 \times \lvert\mathbb{Z}_N\rvert + \lvert c\rvert = 411$k	$150 \times \mathsf{P}_{1024}$
$\mathsf{Sig}_{RSA}[\mathsf{H}_{\mathsf{rand}}]$, $(m = 4)$ [10]	$\lvert\mathbb{Z}_N\rvert + \lvert s\rvert + \lvert\rho\rvert = 1234$	$(2m^2 + 1) \times \lvert\mathbb{Z}_N\rvert + \lvert c\rvert = 34$k	$50 \times \mathsf{P}_{1024}$
Ours in Sec. 6, $(m = 4)$	$\lvert\mathbb{Z}_N\rvert + \lvert s\rvert + \lvert\rho\rvert = 1234$	$(m + 1) \times \lvert\mathbb{Z}_N\rvert + \lvert c\rvert = 6$k	$58 \times \mathsf{P}_{1024}$

* The RSA-based chameleon hash function from [12] was used (adding $1 \times \mathbb{Z}_N$ and $2 \times \mathbb{Z}_N$ to signature size and public key size respectively).

The chosen parameters are $\lambda = 80$, $q = 2^{30}, l = 2\lambda = 160$. We also set $\eta = |s| = \log q + \lambda/m = 50$ so that the term $q^{m+1}/2^{ml}$ is at most $1/2^{\lambda}$ as in [10]. Signatures are instantiated with a modulus of $|N| = 1024$ bits. The description of modulus N and key for PRF are not counted in the public key. We assume l-bit messages with $l = 2\lambda = 160$ in order to provide λ bits of security (to sign longer messages, we can apply a collision resistant hash function first.) The efficiency column counts the dominant operations for signing. $k \times \mathsf{P}_{\mu}$ counts the number of random μ-bit primes that need to be generated in the signing and verification algorithms. (For $\mu >> 60$, $1 \times \mathsf{P}_{\mu}$ takes more time than one exponentiation over modulus N.)

Stateful Version of The Scheme. As our second scheme, we can consider a stateful version of the above scheme. The scheme is more efficient than the above scheme except for slightly longer signature size. The structure of the scheme can be seen as a combination of (randomized) $(1, 1)$-programmable hash function [11] in [7] with RSA based stateful signature scheme in [12]. See the full version of this paper for the details.

7 Comparison

In the Table 1, we compare our schemes with other RSA based signature schemes under appropriately chosen parameters. We ignore the penalty imposed on the modulus size due to the non-tight reduction in the table. The signature size of our first scheme is the same as that of $\mathsf{Sig}_{RSA}[\mathsf{H}_{\mathsf{cfs}}]$, which is currently the shortest signature scheme. As for the public key size, our first scheme is about 2000 bits, which is about $1/20000$ of that of $\mathsf{Sig}_{RSA}[\mathsf{H}_{\mathsf{cfs}}]$, and about half compared with Hohenberger-Waters scheme. However, as we can see, our first scheme requires generation of about 2600 primes, which is impractical. For our second scheme, the public key size is about $1/100$ of that of $\mathsf{Sig}_{RSA}[\mathsf{H}_{\mathsf{cfs}}]$ while its signing and verification cost is about 3 times higher than for their scheme. The second scheme indicates that we can considerably reduce the public key size of $\mathsf{Sig}_{RSA}[\mathsf{H}_{\mathsf{cfs}}]$ at the cost of relatively small increase in computational efficiency. We remark that other choices of parameters are also possible for this scheme. For our third scheme, compared with $\mathsf{Sig}_{RSA}[\mathsf{H}_{\mathsf{rand}}]$, the public key is reduced to less than

1/5 whereas the increase in the computational cost is less than 20 percent. We remark that the reduction cost of our first and second scheme is essentially the same as that of $\mathsf{Sig}_{RSA}[\mathsf{H}_{\mathsf{cfs}}]$, and that of our third scheme is also essentially the same as $\mathsf{Sig}_{RSA}[\mathsf{H}_{\mathsf{rand}}]$. We also remark that we can obtain more efficient schemes in the stateful setting. Especially, stateful version of our third scheme is at least as efficient as Hohenberger-Waters scheme [12] in all aspects. See the full version for the details.

References

1. Bellare, M., Miner, S.K.: A Forward-Secure Digital Signature Scheme. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 431–448. Springer, Heidelberg (1999)
2. Boneh, D., Boyen, X.: Short Signatures Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004)
3. Cramer, R., Shoup, V.: Signature schemes based on the strong RSA assumption. In: ACM Conference on Computer and Communications Security, pp. 46–51 (1999)
4. Dodis, Y., Haitner, I., Tentes, A.: On the (in)security of rsa signatures. Cryptology ePrint Archive, Report 2011/087 (2011), http://eprint.iacr.org/
5. Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography (extended abstract). In: STOC, pp. 542–552 (1991)
6. Erdös, P.L., Frankl, P., Füredi, Z.: Families of finite sets in which no set is covered by the union of two others. J. Comb. Theory, Ser. A 33(2), 158–166 (1982)
7. Fischlin, M.: The Cramer-Shoup Strong-RSA Signature Scheme Revisited. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 116–129. Springer, Heidelberg (2003)
8. Gennaro, R., Halevi, S., Rabin, T.: Secure Hash-and-Sign Signatures without the Random Oracle. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 123–139. Springer, Heidelberg (1999)
9. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. 17(2), 281–308 (1988)
10. Hofheinz, D., Jager, T., Kiltz, E.: Short Signatures From Weaker Assumptions. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 647–666. Springer, Heidelberg (2011)
11. Hofheinz, D., Kiltz, E.: Programmable Hash Functions and Their Applications. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 21–38. Springer, Heidelberg (2008)
12. Hohenberger, S., Waters, B.: Realizing Hash-and-Sign Signatures under Standard Assumptions. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 333–350. Springer, Heidelberg (2009)
13. Hohenberger, S., Waters, B.: Short and Stateless Signatures from the RSA Assumption. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 654–670. Springer, Heidelberg (2009)
14. Krawczyk, H., Rabin, T.: Chameleon signatures. In: NDSS (2000)
15. Kumar, R., Rajagopalan, S., Sahai, A.: Coding Constructions for Blacklisting Problems without Computational Assumptions. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 609–623. Springer, Heidelberg (1999)
16. Rompel, J.: One-way functions are necessary and sufficient for secure signatures. In: STOC, pp. 387–394 (1990)

17. Shamir, A.: On the generation of cryptographically strong pseudorandom sequences. ACM Trans. Comput. Syst. 1(1), 38–44 (1983)
18. Shamir, A., Tauman, Y.: Improved Online/Offline Signature Schemes. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 355–367. Springer, Heidelberg (2001)
19. Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
20. Yamada, S., Hanaoka, G., Kunihiro, N.: Two-Dimensional Representation of Cover Free Families and Its Applications: Short Signatures and More. In: Dunkelman, O. (ed.) CT-RSA 2012. LNCS, vol. 7178, pp. 260–277. Springer, Heidelberg (2012)

A Construction of Map S

In many of our schemes, we use a map S with a special property. We describe how we realize this map S from cover free family [6,15]. Although the idea we describe in this section is not new, we include this section in this paper for completeness.

We begin by recalling the definition of cover-free families. Let S_1, S_2 be sets. We say that S_2 does not cover S_1 if $S_1 \not\subseteq S_2$. Let d, m, α be integers, and let $F = (F_\mu)_{\mu\in[\alpha]}$ be a family of α subsets of $[d]$. We say that F is m-cover free if for any set I containing (up to) m indices $I = \{\mu_1, \ldots, \mu_m\} \subseteq [\alpha]$, it holds that $F_\nu \not\subseteq \cup_{\mu\in I} F_\mu$ for any ν that is not contained in I. In other words, if $|I| \leq m$, then the union $\cup_{\mu\in I} F_\mu$ does not cover F_ν for all $\nu \in [\alpha]\backslash I$. We say that F is w-uniform if $|F_\mu| = w$ for all $\mu \in [\alpha]$. Throughout this paper, we use a parameter in the following lemma.

Lemma 6. *([6,15]) There is a deterministic polynomial-time algorithm that, on input of integers $m, \alpha = 2^n$, returns $d \in \mathbb{N}$ and the set family $F = (F_\mu)_{\mu\in[\alpha]}$, such that F is m-cover free over $[d]$ and w-uniform, where $d \leq 16m^2n$ and $w = d/4m$.*

Note that in the case of $m = 1$, we have a cover-free family with smaller parameters. That is, $\alpha = 2^n$, $d = 2n$, and $w = n$. F_μ is defined as $F_\mu = \{2i - 1 + b_i | i \in [n]\}$ where we regard μ as a concatenation of bit strings in a natural way as $\mu = b_1||\cdots||b_n$ with $b_i \in \{0, 1\}$ for $i \in [n]$. This cover-free family is used in many cryptographic protocols explicitly or implicitly, for example [16,5].

For Our First Scheme. In our first scheme, we associate a message $M \in \mathcal{M}$ with a subset of $[d]$ by a map $S : \mathcal{M} \to 2^{[d]}$. S should satisfy the following property: "For all $M^*, M_1, \ldots, M_m \in \mathcal{M}$, it holds that $S(M^*) \not\subseteq \cup_{i=1}^m S(M_i)$ if $M^* \notin \{M_1, \ldots, M_m\}$." We can construct a map S with this property by defining S as $S(M) \stackrel{def}{=} F_{H(M)} \subseteq [d]$ where $H : \mathcal{M} \to [\alpha]$ is an injective (or hash) function.

For Our Second Scheme. In our second scheme, we associate a message $M \in \mathcal{M}$ with a subset of $[u_1] \times [u_2]$ by a map $S : \mathcal{M} \to 2^{[u_1]\times[u_2]}$. S should satisfy the following property: "For all $M^*, M_1, \ldots, M_m \in \mathcal{M}$, it holds that $S(M^*) \not\subseteq \cup_{i=1}^m S(M_i)$ if $M^* \notin \{M_1, \ldots, M_m\}$." To construct such a map, we

first regard $[d]$ as $[u_1] \times [u_2]$, where u_1 and u_2 are integers satisfying $u_1 \geq u_2$ and $u_1 u_2 \geq d$. (The case for $u_1 \leq u_2$ is analogous.) We regard $i \in [d]$ as an element of $[u_1] \times [u_2]$ by associating it with $(i - u_1(\lceil i/u_1 \rceil - 1), \lceil i/u_1 \rceil)$. Then, all F_μ can be seen as a subset of $[u_1] \times [u_2]$ in a natural way and $(F_\mu)_{\mu \in \alpha}$ can be seen as an m-cover free family over $[u_1] \times [u_2]$. Then we define S as $S(M) \stackrel{def}{=} F_{H(M)} \subseteq [u_1] \times [u_2]$ where $H : \mathcal{M} \to [\alpha]$ is an injective (or hash) function.

In the constructions, we treat H (and S) as an injective function for simplicity, but it is enough to assume that H is a collision resistant hash for our schemes to be secure. To avoid a birthday attack, we typically set $n = 2\lambda$. Besides, if we require F to be w-uniform, then $|S(M)| = w$ for all $M \in \mathcal{M}$.

The Construction of Ambiguous Optimistic Fair Exchange from Designated Confirmer Signature without Random Oracles*

Qiong Huang[1], Duncan S. Wong[2], and Willy Susilo[3]

[1] South China Agricultural University, Guangzhou 510642, China
[2] City University of Hong Kong, Hong Kong S.A.R., China
[3] University of Wollongong, Wollongong, NSW 2522, Australia
csqhuang@alumni.cityu.edu.hk, duncan@cityu.edu.hk, wsusilo@uow.edu.au

Abstract. Ambiguous Optimistic Fair Exchange (AOFE), introduced by Huang *et al.* in ASIACRYPT 2008, is an extension of OFE that enhances the fairness of the two communicating parties in the exchange of signatures. The first scheme was proven secure without random oracles while its partial signature contains dozens of group elements. Recently, interactive AOFE was introduced and the construction is more practical, where one partial signature only contains three group elements. It is based on the existence of Designated Confirmer Signature (DCS) with a special property where one is able to sample a confirmer signature efficiently from a signer's signature space. Nevertheless, we note that there are only a few DCS schemes that have this special property. Security of the interactive AOFE construction relies on the q-Computational and Decisional Hidden Strong Diffie-Hellman assumptions. In this paper, we propose a new construction of interactive AOFE from DCS, where the underlying DCS is standard and does not require any special property. We also propose a new DCS construction. By applying our transformation from DCS to interactive AOFE, we build a concrete interactive AOFE which is secure under more standard number-theoretic assumptions, namely Strong Diffie-Hellman and Decision Linear assumptions, without random oracles. A partial signature of the interactive AOFE contains six group elements, while a full signature contains two only.

Keywords: Optimistic fair exchange, ambiguity, designated confirmer signature, standard model.

1 Introduction

How to exchange items between parties so that either both the parties get their counterpart's item or none of them does, is an important problem in e-commerce.

* This work is supported by the National Natural Science Foundation of China (No. 61103232), the Research Fund for the Doctoral Program of Higher Education of China (No. 20114404120027), and the Foundation for Distinguished Young Talents in Higher Education of Guangdong, China (No. LYM11033). D. S. Wong is supported by a grant from the RGC of the HKSAR, China (Project No. CityU 123511). W. Susilo is supported by ARC Future Fellowship FT0991397.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 120–137, 2012.

Optimistic Fair Exchange (OFE), introduced by Asokan, Schunter and Waidner [1], is a kind of protocols for exchanging items between two parties say, Alice and Bob, in a fair manner. There is an arbitrator, which is semi-trusted by Alice and Bob and gets involved only when a party attempts to cheat the other or simply crashes. Asokan, Shoup and Waidner later proposed an OFE protocol for exchanging digital signatures [2]. In a typical run of OFE, Alice sends a *partial signature* to Bob, who in turn sends back his *full signature*, which triggers Alice to complete the protocol by releasing her full signature to Bob. If everything goes well, Alice and Bob should get each other's full signatures. However, if Alice refuses or fails to respond in the third move, Bob then resorts to the arbitrator for resolving Alice's partial signature into a full one. Since the introduction, OFE has attracted the attention of many researchers, i.e. [3, 13, 14, 23, 30].

In OFE, Alice's partial signature is generally self-authenticating and indicates her commitment to some message already. This may allow Bob to convince others that Alice has already committed herself to the message; while Alice obtains nothing. This could be unfair to Alice. Huang *et al.* [22] addressed this problem and proposed the notion of *ambiguous optimistic fair exchange* (AOFE), which is similar to the notion of *abuse-free optimistic contract signing* introduced by Garay *et al.* [15]. Different from the traditional OFE, AOFE enjoys the property of *signer ambiguity*. That is, Bob is able to produce partial signatures which are indistinguishable to those produced by Alice. Because of this property, given a valid partial signature from Alice, Bob cannot transfer its conviction to others any more.

1.1 Our Contributions

In this paper, we propose a new efficient and yet generic construction of AOFE from a primitive called designated confirmer signature (DCS) [10]. Compared with previous work on the construction of AOFE from DCS, e.g. [20, 21], our construction makes use of standard security properties of the underlying primitive, rather than any special property, e.g. *samplability* [20, 21]. Our AOFE protocol is interactive in the sense that the partial signature generation needs an interaction between the signer and the verifier. Below we give an intuition.

To partially sign a message, the signer produces a confirmer signature on it and then carries out a zero-knowledge proof with the verifier to show that the confirmer signature belongs to either the signer or the verifier. Thanks to the anonymity of the DCS scheme and the zero-knowledge property of the proof, a third party (except the arbitrator) cannot tell who is the real signer of the signature. We show that the resulting interactive AOFE protocol is secure in the registered-key model[1] [4] without random oracles if the underlying DCS scheme is secure and the proof is sound and zero-knowledge.

To instantiate our construction of AOFE, we present a concrete and efficient DCS scheme, which is secure based on Strong Diffie-Hellman assumption [6] and Decision Linear assumption [7] without random oracles. It has short signatures and keys, and the confirmation/disavowal protocol is practically efficient.

[1] In this model the adversary has to prove its knowledge of the secret key before using a public key.

Compared with [20, 21], our scheme has much shorter signer public key and its security relies on relatively more standard assumptions. Compared with [33], the signature of our scheme is shorter, and the confirmation/disavowal protocol is more efficient. In Table 1 we give a comparison of our scheme with some existing DCS schemes in terms of sizes of confirmer public key **cpk**, confirmer secret key **csk**, signer public key **spk**, signer secret key **ssk**, confirmer signature σ and standard signature ζ, the underlying assumptions and the need of random oracles for security.

Table 1. Comparison with some existing DCS schemes

	[8]	[31]	[33]	[20, 21]	Ours
cpk	$5\mathbb{Z}_p$	$1\mathbb{G}$	$2\|N\|+1\|n\|+4\|\mathbb{Z}_{n^2}\|$	$1\mathbb{G}$	$4\mathbb{G}$
csk	$5\mathbb{Z}_q$	$1\mathbb{Z}_p$	$3(\kappa+\kappa_r)$	$1\mathbb{Z}_p$	$2\mathbb{Z}_p$
spk	$1\mathbb{Z}_N + k$	$1\mathbb{G}$	$\|N_0\|+\|N_1\|+1SQ_{N_0}+1SQ_{N_1}$	$163\mathbb{G}$	$2\mathbb{G}$
ssk	$1\mathbb{Z}_N$	$1\mathbb{Z}_p$	2κ	$1\mathbb{Z}_p$	$2\mathbb{Z}_p$
σ	$4\mathbb{Z}_p \approx 4K$	$2\mathbb{G}+4\mathbb{Z}_p \approx 0.95K$	$9\mathbb{Z}_{n^2}+1\mathbb{Z}_{N_0}+1\mathbb{Z}_{N_1} \approx 22K$	$3\mathbb{G} \approx 0.47K$	$5\mathbb{G}+1\mathbb{Z}_p \approx 0.96K$
ζ	$1\mathbb{Z}_N \approx 1K$	$2\mathbb{G}+6\mathbb{Z}_p \approx 1.2K$	$3SQ_{n^2}+1\mathbb{Z}^*_{N_0}+1\mathbb{Z}^*_{N_1} \approx 10K$	$3\mathbb{G} \approx 0.47K$	$1\mathbb{G}+1\mathbb{Z}_p \approx 0.32K$
Asmp	RSA+DDH	DDH+EUF-CMA	SRSA+DCRA+DDH	HSDH+DHSDH	SDH+DLIN
ROM	no	yes	no	no	no

Legends:

DLIN	: Decision Linear Assumption
DDH	: Decision Diffie-Hellman
SRSA	: Strong RSA Assumption
DCRA	: Decision Composite Residuosity Assumption
HSDH	: Hidden Strong Diffie-Hellman Assumption
DHSDH	: Decision Hidden Strong Diffie-Hellman Assumption
EUF-CMA	: Existential unforgeability (under chosen message attacks) of the underlying signature scheme

Security Parameters:
$\kappa = |n| = |N| = 1024$, $|N_0| = |N_1| = 2048$, $\kappa_r = 50$, $|\mathbb{G}| \approx 163$, $|\mathbb{Z}_p| \approx 163$ (for [8], we choose $|\mathbb{Z}_p| \approx 1024$).

1.2 Paper Organization

In the next section we review the related works on designated confirmer signature and optimistic fair exchange. The definition and security models of ambiguous optimistic fair exchange are given in Section 3. Our construction of interactive AOFE is then proposed in Section 4, followed by a section which gives the security analysis. In Section 6 we propose an efficient construction of designated confirmer signature, the security of which does not rely on the random oracle model. In Section 7 we compare our AOFE scheme with some existing schemes. The paper is concluded in Section 8.

2 Related Work

Designated Confirmer Signature. The notion of designated confirmer signature was proposed by Chaum [10] to alleviate the burden of the signer in

undeniable signature [9]. In DCS, the signer designates a confirmer to confirm or disavow signatures for him, and the verifier cannot verify signatures alone. If a DCS scheme is *convertible*, the confirmer has the ability to extract the signer's standard signature from a valid confirmer signature. There have been a lot works on DCS since its introduction, e.g. [8, 11, 16, 17, 20, 21, 27, 28, 31, 32, 33, 34]. Readers can refer to [20, 21] for a brief review of the previous works.

The *de facto* security properties of a DCS scheme include *unforgeability* and *anonymity*. The former requires that no one but the signer is able to produce valid signatures; while the latter says that given a confirmer signature, no verifier is able to distinguish the identity of the signer. A popular approach in the design of DCS is known as the '*sign-then-encrypt*' paradigm. Intuitively, the confirmer holds a key pair $(\mathtt{Pk}_E, \mathtt{Sk}_E)$ for an encryption scheme E and the signer holds a key pair $(\mathtt{Pk}_S, \mathtt{Sk}_S)$ for a signature scheme Σ. To sign a message M w.r.t. the confirmer, the signer computes a standard signature ζ on M using $\mathtt{Sk}_E$, and encrypts ζ under the confirmer's public key $\mathtt{Pk}_S$ to obtain the ciphertext C. Its confirmer signature is set to be C. To convert a confirmer signature C, the confirmer decrypts it to ζ using $\mathtt{Sk}_E$, and outputs ζ if it is valid under $\mathtt{Pk}_S$. In the confirmation (resp. disavowal) protocol, the confirmer proves to the verifier (interactively) that the confirmer signature C can (resp. cannot) be decrypted to a valid signature of the signer on message M. The unforgeability of the DCS scheme simply follows that of Σ. On the other hand, the (chosen ciphertext) security of E guarantees that given a ciphertext C, anyone who does not known $\mathtt{Sk}_E$, including the signer, is not able to tell C contains which signer's signature. Thus we have the anonymity.

Many DCS schemes follow this paradigm, e.g. [8, 16, 17]. The difficulty of implementing the paradigm is in the design of confirmation and disavowal protocols so that the scheme is efficient enough for practical use. It is known that the protocols can be constructed in general, using complex NP reduction. However, the efficiency is a big issue. As far as we know, there are only a few DCS schemes which have efficient confirmation and disavowal protocols, e.g. [16, 20, 21, 31, 33, 34].

Wikström [33] revisited the aforementioned paradigm of constructing DCS, and proposed a similar generic construction, which makes use of a weak variant of CCA-secure cryptosystem, a signature scheme, and a weak form of zero-knowledge proofs. A concrete instantiation was also presented in [33], which is built from Cramer-Shoup version of Paillier encryption [29] and a twin-moduli signature. The confirmation and disavowal protocols, although do not involve any NP-reduction, are not efficient enough. The prover and the verifier have to carry out a bunch of proofs of knowledge, and both of them need perform more than 150 exponentiation evaluations.

Huang *et al.* [20, 21] proposed a new variant of DCS, in which both the signer and the confirmer are able to not only confirm but also disavow signatures efficiently. They also presented a concrete construction, the security of which is based on new number-theoretic assumptions (e.g. Hidden Strong Diffie-Hellman assumption and Decision Hidden Strong Diffie-Hellman assumption) without random oracles. The new variant is useful in applications in which the signer

prefers to retain the ability to disavow signatures, whereas there are still some cases in which the signer only wants to keep the ability of confirmation.

Ambiguous Optimistic Fair Exchange. Garay *et al.* [15] for the first time addressed the problem with the non-repudiation of a partial signature, and proposed an efficient abuse-free contract signing protocol, in which no one but the arbitrator can distinguish who produced which signature. The protocol makes use of a type of signatures called '*private contract signatures*', which is similar to but different from DCS. Their private contract signature scheme is built from designated-verifier signature [24], and is secure based on DDH assumption in the random oracle model [5] and the registered-key model [4], in which the adversary has to show its knowledge of the secret key before using a public key.

Huang *et al.* [22] proposed an efficient construction of AOFE based on the group signature scheme in [18]. Their scheme uses (the weakly secure) Boneh-Boyen signature [6] and Groth-Sahai non-interactive proof techniques [19]. The scheme is secure based on Strong Diffie-Hellman assumption [6] and Decision Linear assumption [7] in the chosen-key model [23, 26] without random oracles, in which the adversary is allowed to use public keys arbitrarily. However, the scheme suffers from long signatures, which consist of more than 40 group elements.

Very recently, Huang *et al.* [20, 21] proposed a new approach to constructing *interactive* AOFE, in which the signer interacts with the verifier to produce the partial signature. Their construction applies to a specific class of DCS schemes, in which anyone is able to sample confirmer signatures from the signer's signature space efficiently, e.g. in polynomial time. However, not many DCS schemes enjoy this property, and thus limiting the application of Huang *et al.*'s construction. They also instantiated the construction using the DCS scheme proposed in the same paper. The resulting interactive protocol is secure without random oracles in the registered-key model.

3 Ambiguous Optimistic Fair Exchange

3.1 Definition

Essentially, AOFE is a variant of the traditional OFE, in which both of the exchanging parties can produce indistinguishable signatures on the same message. An AOFE scheme consists of the following probabilistic polynomial time algorithms/protocols:

PMGen. It takes 1^k as input where k is the security parameter and outputs the system parameter PM.

Setup$^{\mathbf{TTP}}$. It takes as input the system parameter PM and outputs a key pair for the arbitrator. We denote it by $(\mathtt{Apk}, \mathtt{Ask}) \leftarrow \mathsf{Setup}^{\mathsf{TTP}}(1^k)$.

Setup$^{\mathbf{User}}$. It takes the system parameter PM (and optionally $\mathtt{Apk}$) as input and outputs a key pair for the user. We denote it by $(\mathtt{Pk}, \mathtt{Sk}) \leftarrow \mathsf{Setup}^{\mathsf{User}}(1^k, \mathtt{Apk})$.

PSig. This is the partial signature generation algorithm. It takes as input a message M, the signer's secret key $\mathtt{Sk}_i$, the signer's public key $\mathtt{Pk}_i$, the verifier's

public key $\mathtt{Pk}_j$ and the arbitrator's public key $\mathtt{Apk}$, and outputs a partial signature σ. We denote it by $\sigma \leftarrow \mathsf{PSig}(M, \mathtt{Sk}_i, \mathtt{Pk}_i, \mathtt{Pk}_j, \mathtt{Apk})$.

PVer. This is for the verification of a partial signature. It can be either an algorithm or a protocol, depending on whether the verification requires the interaction between the signer and the verifier or not. The (common) input consists of $(M, \sigma, \mathtt{Pk}_i, \mathtt{Pk}_j, \mathtt{Apk})$. If the verification is interactive, the signer has private input $\mathtt{Sk}_i$. We denote it by $b \leftarrow \mathsf{PVer}(M, \sigma, \mathtt{Pk}_i, \mathtt{Pk}_j, \mathtt{Apk})$, where b is the output of the verifier, which is 1 for acceptance and 0 for rejection.

Sig. This is the full signature generation algorithm. It takes as input $(M, \mathtt{Sk}_i, \mathtt{Pk}_i, \mathtt{Pk}_j, \mathtt{Apk})$ and outputs a full signature ζ. We denote it by $\zeta \leftarrow \mathsf{Sig}(M, \mathtt{Sk}_i, \mathtt{Pk}_i, \mathtt{Pk}_j, \mathtt{Apk})$.

Ver. This is for the verification of a full signature. It takes as input $(M, \zeta, \mathtt{Pk}_i, \mathtt{Pk}_j, \mathtt{Apk})$ and outputs a bit b which is 1 if ζ is a valid full signature of $\mathtt{Pk}_i$ and 0 otherwise. We denote it by $b \leftarrow \mathsf{Ver}(M, \zeta, \mathtt{Pk}_i, \mathtt{Pk}_j, \mathtt{Apk})$.

Res. This is for resolving a partial signature. It takes as input $(M, \mathtt{Ask}, \sigma, \mathtt{Pk}_i, \mathtt{Pk}_j)$ and outputs ζ if ζ is a valid full signature of $\mathtt{Pk}_i$, and $\perp$ otherwise.

The AOFE introduced in [22] is *non-interactive* in the sense that all the signature generation and verification algorithms are non-interactive. However, in this work we consider *interactive* AOFE (iAOFE in short), in which the partial signature verification is an interactive protocol between the signer and the verifier. For simplicity we treat PVer as a protocol universally for both interactive and non-interactive AOFE. If the scheme is non-interactive, then in the PVer protocol (which should be an algorithm) the signer does nothing and the verifier makes the decision alone.

3.2 Security Models

The security of AOFE was originally defined in the *chosen-key* model [22], in which the adversary is allowed to use any public key arbitrarily without showing its knowledge of the corresponding secret key. While in this work we consider AOFE in the *registered-key* model [4], which is weaker than the chosen-key model yet still practical.

Registered-Key Model. In this model the adversary has to prove its knowledge of the corresponding secret key before using a public key. Although this model puts limits to the adversary on using public keys, it is still a practical model, and has been considered in many works, such as [13, 25]. Usually, an adversary in this model conducts a proof of knowledge of the secret key to the game challenger or simply submits the key pair or even the randomness used in key generation. In the rest of the paper we assume that the adversary has access to a key registration oracle O_{KR}, which takes as input a key pair $(\mathtt{Pk}, \mathtt{Sk})$, and returns $\mathtt{Pk}$ if the pair is a valid output of the key generation algorithm and $\perp$ otherwise.

Let $\mathcal{Q}(O)$ be the set of queries that the adversary submits to oracle O, where O could be any of the oracles below.

- O_{KR} is the key registration oracle.
- O_{PSig} takes as input $(M, \mathtt{Pk}_i)$ and returns a partial signature σ of the signer with public key $\mathtt{Pk}_A$, which is valid on M under $\mathtt{Pk}_A$, $\mathtt{Pk}_i$. The oracle then starts an execution of PVer with the adversary to show the validity of σ.
- O_{FakePSig} takes as input $(M, \mathtt{Pk}_i)$ and returns a partial signature σ generated using $\mathtt{Sk}_B$ and valid under $\mathtt{Pk}_i$, $\mathtt{Pk}_B$. The oracle then starts an execution of the PVer protocol with the adversary to show the validity of σ.
- O_{Res} takes as input $(M, \sigma, \mathtt{Pk}_i, \mathtt{Pk}_j)$ and outputs ζ if it is a valid (standard) signature on M under $\mathtt{Pk}_i$, and $\perp$ otherwise.

If the public key submitted to any of O_{PSig}, O_{FakePSig} and O_{Res} was not ever submitted to O_{KR}, these oracles would simply return nothing to the adversary.

Signer Ambiguity. The signer ambiguity says that after obtaining the valid partial signature from the signer S, the verifier V cannot transfer the conviction to any third party. We require that V is able to produce signatures indistinguishable from those by S. Formally, we consider the game $\mathbb{G}_{\mathrm{sa}}$ depicted in Figure 1 (page 127), where Υ is $\mathcal{D}$'s state information. Note that after sending σ^* to $\mathcal{D}$ in the game, the challenger also starts an execution of the PVer protocol with $\mathcal{D}$ to show the validity of σ^* under $\mathtt{Pk}_A, \mathtt{Pk}_B$. The advantage of $\mathcal{D}$, denoted by $\mathtt{Adv}^{\mathrm{sa}}_{\mathcal{D}}(k)$, is defined to be the gap between its success probability in the game and one half, i.e. $\mathtt{Adv}^{\mathrm{sa}}_{\mathcal{D}}(k) = |\Pr[\mathcal{D}\ \mathrm{Succ}] - 1/2|$.

Definition 1 (Signer Ambiguity). *An AOFE scheme is* signer ambiguous *if there is no PPT distinguisher $\mathcal{D}$ such that $\mathtt{Adv}^{\mathrm{sa}}_{\mathcal{D}}(k)$ is non-negligible in k.*

Security Against Signers. It requires that (malicious) signer $\mathcal{A}$ cannot produce a partial signature, which looks good to V but cannot be resolved to a full signature by the honest arbitrator, ensuring the fairness for verifiers. V should always be able to obtain the full commitment of the signer if the signer has committed to a message. Formally, we consider the game $\mathbb{G}_{\mathrm{sas}}$ depicted in Figure 1. The advantage of $\mathcal{A}$ in the game, denoted by $\mathtt{Adv}^{\mathrm{sas}}_{\mathcal{A}}(k)$, is defined as its success probability.

Definition 2 (Security Against Signers). *An AOFE scheme is* secure against signers *if there is no PPT adversary $\mathcal{A}$ such that $\mathtt{Adv}^{\mathrm{sas}}_{\mathcal{A}}(k)$ is non-negligible in k.*

Security Against Verifiers. It requires that any efficient verifier $\mathcal{B}$ should not be able to convert a partial signature into a full one with non-negligible probability if it obtains no help from the signer or the arbitrator. This ensures the fairness for the arbitrator and the signer. Formally, we consider the game $\mathbb{G}_{\mathrm{sav}}$ depicted in Figure 1. The advantage of $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ in the game, denoted by $\mathtt{Adv}^{\mathrm{sav}}_{\mathcal{B}}(k)$, is defined as its success probability.

Definition 3 (Security Against Verifiers). *An AOFE scheme is* secure against verifiers *if there is no probabilistic polynomial-time adversary $\mathcal{B}$ such that $\mathtt{Adv}^{\mathrm{sav}}_{\mathcal{B}}(k)$ is non-negligible in k.*

Game $\mathbb{G}_{\mathrm{sa}}$:

$\mathsf{PM} \leftarrow \mathsf{PMGen}(1^k), \quad (\mathtt{Apk}, \mathtt{Ask}) \leftarrow \mathsf{Setup}^{\mathsf{TTP}}(\mathsf{PM}),$

$(\mathtt{Pk}_A, \mathtt{Sk}_A) \leftarrow \mathsf{Setup}^{\mathsf{User}}(\mathsf{PM}, \mathtt{Apk}), \quad (\mathtt{Pk}_B, \mathtt{Sk}_B) \leftarrow \mathsf{Setup}^{\mathsf{User}}(\mathsf{PM}, \mathtt{Apk}),$

$(M^*, \Upsilon) \leftarrow \mathcal{D}^{O_{\mathsf{KR}}, O_{\mathsf{Res}}}(\mathtt{Apk}, (\mathtt{Pk}_A, \mathtt{Sk}_A), (\mathtt{Pk}_B, \mathtt{Sk}_B)), \quad b \leftarrow \{0,1\},$

$$\sigma^* \leftarrow \begin{cases} \mathsf{PSig}(M^*, \mathtt{Sk}_A, \mathtt{Pk}_A, \mathtt{Pk}_B, \mathtt{Apk}) & \text{if } b = 0 \\ \mathsf{FakePSig}(M^*, \mathtt{Sk}_B, \mathtt{Pk}_A, \mathtt{Pk}_B, \mathtt{Apk}) & \text{otherwise} \end{cases},$$

$b' \leftarrow \mathcal{D}^{O_{\mathsf{KR}}, O_{\mathsf{Res}}}(\Upsilon, \sigma^*),$

Succ. of $\mathcal{D} := [b' = b \wedge (M^*, \sigma, \{\mathtt{Pk}_A, \mathtt{Pk}_B\}) \notin \mathcal{Q}(O_{\mathsf{Res}})]$.

Game $\mathbb{G}_{\mathrm{saa}}$:

$\mathsf{PM} \leftarrow \mathsf{PMGen}(1^k), \quad (\mathtt{Apk}, \mathtt{Ask}) \leftarrow \mathsf{Setup}^{\mathsf{TTP}}(\mathsf{PM}),$

$(\mathtt{Pk}_A, \mathtt{Sk}_A) \leftarrow \mathsf{Setup}^{\mathsf{User}}(\mathsf{PM}, \mathtt{Apk}), \quad (M^*, \mathtt{Pk}_B, \zeta^*) \leftarrow \mathcal{C}^{O_{\mathsf{KR}}, O_{\mathsf{PSig}}}(\mathtt{Ask}, \mathtt{Apk}, \mathtt{Pk}_A),$

Succ. of $\mathcal{C} := [\mathsf{Ver}(M^*, \zeta, \mathtt{Pk}_A, \mathtt{Pk}_B, \mathtt{Apk}) = 1 \wedge (M^*, \mathtt{Pk}_B) \notin \mathcal{Q}(\mathcal{O}_{\mathsf{PSig}})]$.

Game $\mathbb{G}_{\mathrm{sas}}$:

$\mathsf{PM} \leftarrow \mathsf{PMGen}(1^k), \quad (\mathtt{Apk}, \mathtt{Ask}) \leftarrow \mathsf{Setup}^{\mathsf{TTP}}(\mathsf{PM}), \quad (\mathtt{Pk}_B, \mathtt{Sk}_B) \leftarrow \mathsf{Setup}^{\mathsf{User}}(\mathsf{PM}, \mathtt{Apk}),$

$(M^*, \mathtt{Pk}_A, \sigma^*) \leftarrow \mathcal{A}^{O_{\mathsf{KR}}, O_{\mathsf{FakePSig}}, O_{\mathsf{Res}}}(\mathtt{Apk}, \mathtt{Pk}_B), \quad \zeta^* \leftarrow \mathsf{Res}(M^*, \sigma^*, \mathtt{Ask}, \mathtt{Pk}_A, \mathtt{Pk}_B),$

Succ. of $\mathcal{A} := [\mathsf{PVer}(M^*, \sigma^*, \{\mathtt{Pk}_A, \mathtt{Pk}_B\}, \mathtt{Apk}) = 1$

$\wedge\, \mathsf{Ver}(M^*, \zeta^*, \mathtt{Pk}_A, \mathtt{Pk}_B, \mathtt{Apk}) = 0 \wedge (M^*, \mathtt{Pk}_A) \notin \mathcal{Q}(O_{\mathsf{FakePSig}})]$.

Game $\mathbb{G}_{\mathrm{sav}}$:

$\mathsf{PM} \leftarrow \mathsf{PMGen}(1^k), \quad (\mathtt{Apk}, \mathtt{Ask}) \leftarrow \mathsf{Setup}^{\mathsf{TTP}}(\mathsf{PM}),$

$(\mathtt{Pk}_A, \mathtt{Sk}_A) \leftarrow \mathsf{Setup}^{\mathsf{User}}(\mathsf{PM}, \mathtt{Apk}), \quad (\mathtt{Pk}_B, \mathtt{Sk}_B) \leftarrow \mathsf{Setup}^{\mathsf{User}}(\mathsf{PM}, \mathtt{Apk}),$

$(M^*, \Upsilon) \leftarrow \mathcal{B}_1^{O_{\mathsf{KR}}, O_{\mathsf{PSig}}, O_{\mathsf{Res}}}(\mathtt{Apk}, \mathtt{Pk}_A, \mathtt{Pk}_B, \mathtt{Sk}_B),$

$\sigma^* \leftarrow \mathsf{PSig}(M^*, \mathtt{Sk}_A, \mathtt{Pk}_A, \mathtt{Pk}_B, \mathtt{Apk}), \quad \zeta^* \leftarrow \mathcal{B}_2^{O_{\mathsf{KR}}, O_{\mathsf{PSig}}, O_{\mathsf{Res}}}(\Upsilon, \sigma^*),$

Succ. of $\mathcal{B} := [\mathsf{Ver}(M^*, \zeta^*, \mathtt{Pk}_A, \mathtt{Pk}_B, \mathtt{Apk}) = 1 \wedge (M^*, \cdot, \{\mathtt{Pk}_A, \mathtt{Pk}_B\}) \notin \mathcal{Q}(O_{\mathsf{Res}})]$.

Fig. 1. Security models of AOFE

Security Against the Arbitrator. This is for ensuring the unforgeability of the signer's signatures. It says that no efficient adversary $\mathcal{C}$, even the arbitrator, is able to generate with non-negligible probability a valid full signature without explicitly asking the signer for generating one. Formally, we consider the game $\mathbb{G}_{\mathrm{saa}}$ depicted in Figure 1. The advantage of $\mathcal{C}$ in this game, denoted by $\mathtt{Adv}^{\mathrm{saa}}_{\mathcal{C}}(k)$, is defined as its success probability.

Definition 4 (Security Against the Arbitrator). *An AOFE scheme is* secure against the arbitrator *if there is no PPT adversary $\mathcal{C}$ such that $\mathtt{Adv}^{\mathrm{saa}}_{\mathcal{C}}(k)$ is non-negligible in k.*

Remark 1. Our definitions of signer ambiguity and security against verifiers are slightly weaker than those considered in [20, 21, 22]. In our definition of signer ambiguity, the two challenge public keys $(\mathtt{Pk}_A, \mathtt{Pk}_B)$ (along with their

corresponding secret keys) are given to the adversary, rather than the adversary chooses one of them as [20, 21] does. Similarly, in our definition of security against verifiers, the two public keys are also chosen by the challenger and given to the adversary. Nevertheless, it is this slight weakening in the security which enables us to construct AOFE protocols from DCS schemes with standard security properties instead of any special property like samplability [20, 21], in a general way, as we shall see in Section 4.

Similar to [22], it is straightforward to establish the relation between security against verifiers and signer ambiguity. We have the following lemma and therefore, we need not consider security against verifiers in proving the security of an AOFE scheme.

Lemma 1. *If an AOFE scheme is both signer ambiguous (Definition 1) and secure against the arbitrator (Definition 4), it is secure against verifiers (Definition 3) as well.*

Definition 5 (Secure AOFE). *An AOFE scheme is said to be* secure in the multi-user setting and registered-key model *(or simply,* secure*), if it satisfies signer ambiguity (Definition 1), security against signers (Definition 2), and security against the arbitrator (Definition 4).*

4 Our Construction of iAOFE

In this part we present a construction of interactive AOFE based on a designated confirmer signature (DCS) scheme. Before describing our construction, let us give a brief introduction of DCS first.

In DCS, there are a signer S, a verifier V and a confirmer C. S and C run algorithms SKg and CKg to produce their public/secret key pairs, respectively. The signer can run Sig algorithm to produce its standard signatures, which can be verified by V by calling the Ver algorithm. S can also run DCSig to produce confirmer signatures by designating C as the confirmer, which could not be verified by the verifier alone. To prove the validity/invalidity of a confirmer signature, C runs $\mathsf{Confirm}/\mathsf{Disavow}$ protocol with V. There are two confirmation protocols, $\mathsf{ConfirmS}$ and $\mathsf{ConfirmC}$, run by S and C to prove the validity of a confirmer signature, respectively. Besides, C is able to convert (valid) confirmer signatures to standard ones.

The Construction. Below we present our construction of interactive AOFE. Compared with previous work on the construction of AOFE from DCS, e.g. [20, 21], our construction makes use of the standard security properties of the underlying DCS, rather than any special property, e.g. *samplability* [20, 21]. Intuitively, in our construction of interactive AOFE, U_i's partial signature σ on a message M is simply its confirmer signature. Since the DCS scheme is anonymous, no one but the confirmer is able to tell σ was produced by U_i or U_j. Let Σ be a DCS scheme. Our AOFE scheme works as below, where U_i is the signer and U_j is the verifier.

PMGen. It generates all the necessary system parameters for Σ.

Setup$^{\mathbf{TTP}}$. The arbitrator computes $(\mathtt{Cpk}, \mathtt{Csk}) \leftarrow \Sigma.\mathsf{CKg}(1^k)$, and sets its key pair as $(\mathtt{Apk}, \mathtt{Ask}) := (\mathtt{Cpk}, \mathtt{Csk})$.

Setup$^{\mathbf{User}}$. Each user computes $(\mathtt{Spk}, \mathtt{Ssk}) \leftarrow \Sigma.\mathsf{SKg}(1^k)$, and sets its key pair as $(\mathtt{Pk}, \mathtt{Sk}) := (\mathtt{Spk}, \mathtt{Ssk})$.

PSsig. To partially sign a message M for U_j, U_i computes $\sigma \leftarrow \Sigma.\mathsf{DCSig}(\mathtt{Sk}_i, \hat{M})$, where $\hat{M} = M \| \mathtt{Pk}_j$, and sends σ to U_j.

PVer. Given a partial signature σ, U_i and U_j carry out an execution of a zero-knowledge proof Π which is the OR combination of two independent copies of $\Sigma.\mathsf{ConfirmS}$, to show that σ is a valid confirmer signature on $\hat{M}$ of either U_i or U_j. U_i plays the role of the prover in the proof. U_j outputs 1 if it accepts at the end of the proof, and 0 otherwise.

Sig. To fully sign a message M, U_i computes $\zeta \leftarrow \Sigma.\mathsf{Sig}(\mathtt{Sk}_i, \hat{M}, \mathtt{Apk})$, and sends it to U_j.

Ver. Given a full signature ζ, U_j outputs $\Sigma.\mathsf{Ver}(\hat{M}, \zeta, \mathtt{Pk}_i, \mathtt{Apk})$.

Res. Given $(M, \sigma, \mathtt{Pk}_i, \mathtt{Pk}_j)$, the arbitrator computes and returns $\zeta \leftarrow \Sigma.\mathsf{Ext}(\mathtt{Ask}, \hat{M}, \sigma, \mathtt{Pk}_i)$ to U_j if $1 \leftarrow \Sigma.\mathsf{Ver}(\hat{M}, \zeta, \mathtt{Pk}_i, \mathtt{Apk})$ and $\perp$ otherwise.

Remark 2. As we can see from the construction above, the resulting interactive AOFE protocol is solely based on the underlying DCS scheme. The proof run between the signer and the verifier is an OR composition of two copies of the confirmation protocol of the DCS scheme. There are standard technique of composing (the Σ-protocol [12] version of) the confirmation protocol.

The correctness of the construction above is obvious, and we skip the details here. In the next section we analyze the security of the construction under the models given in Fig. 1.

5 Security Analysis

Since our AOFE protocol is built from a DCS scheme, before proving the security of our AOFE scheme (under the models given in Sec. 3.2), let us briefly describe the security models of DCS.

A secure DCS scheme satisfies two security properties. One is unforgeability, which requires that no one but the signer be able to generate valid (standard) signatures. Even the confirmer could not forge either. The other property is anonymity, which requires no one but the confirmer be able to tell a given confirmer signature was generated by which signer. If the signer does not store the signatures it ever produced, it cannot distinguish either. Due to the page limit we defer the detailed security definitions of DCS into the full version.

Now we begin to analyze the security of our construction of interactive AOFE. We have the following theorem.

Theorem 1. *The interactive AOFE scheme above is secure (Definition 5) provided that Σ is secure and the proof Π is sound and zero-knowledge.*

It follows the following lemmas immediately.

Lemma 2. *The interactive AOFE scheme is signer-ambiguous if* Σ *is anonymous and the proof* Π *is zero-knowledge.*

Proof. To simulate U_i's partial signature on a message M, U_j computes $\sigma' \leftarrow \Sigma.\mathsf{DCSig}(\mathtt{Sk}_j, \hat{M})$ where $\hat{M} = M \| \mathtt{Pk}_j$, and outputs σ' as the simulated partial signature. Guaranteed by the anonymity of Σ, we know that σ' looks indistinguishable from U_i's partial signature on M. Below we prove that the simulated signature is indistinguishable from the output of a real signer.

Let $\mathcal{D}$ be a distinguisher which can tell U_j's simulated signatures apart from U_i's real signatures with probability $1/2 + \epsilon$, where ϵ is non-negligible. We use it to build another algorithm $\mathcal{D}'$ for breaking the anonymity of Σ.

Given the system parameters, two key pairs $(\mathtt{Spk}_0, \mathtt{Ssk}_0), (\mathtt{Spk}_1, \mathtt{Ssk}_1)$ and a confirmer public key $\mathtt{Cpk}$, $\mathcal{D}'$ sets $\mathtt{Apk} := \mathtt{Cpk}$, $(\mathtt{Pk}_A, \mathtt{Sk}_A) := (\mathtt{Spk}_0, \mathtt{Ssk}_0)$ and $(\mathtt{Pk}_B, \mathtt{Sk}_B) := (\mathtt{Spk}_1, \mathtt{Ssk}_1)$, and invokes $\mathcal{D}$ on input $(\mathtt{Apk}, (\mathtt{Pk}_A, \mathtt{Sk}_A), (\mathtt{Pk}_B, \mathtt{Sk}_B))$. The oracles are simulated by $\mathcal{D}'$ as follows:

O_{KR}**.** Given a key pair $(\mathtt{Pk}, \mathtt{Sk})$, if it is not well-formed, $\mathcal{D}'$ returns $\perp$; otherwise, it stores the pair and returns $\mathtt{Pk}$.

O_{Res}**.** Given $(M, \sigma, \mathtt{Pk}_i, \mathtt{Pk}_j)$, $\mathcal{D}'$ sets $\hat{M} := M \| \mathtt{Pk}_j$ and forwards $(\hat{M}, \sigma, \mathtt{Pk}_i)$ to its extraction oracle, which returns ζ. It returns $\perp$ to the distinguisher if $\zeta = \perp$ or $0 \leftarrow \Sigma.\mathsf{Ver}(\hat{M}, \zeta, \mathtt{Pk}_i, \mathtt{Apk})$, and ζ otherwise.

When $\mathcal{D}$ submits a challenge message M^*, $\mathcal{D}'$ forwards $\hat{M}^* := M^* \| \mathtt{Pk}_B$ to its own challenger, which tosses a coin b and returns a confirmer signature σ^* on $\hat{M}^*$ valid under $\mathtt{Spk}_b$. It then sends σ^* to $\mathcal{D}$, and runs the simulator of protocol Π to prove that σ^* is a valid confirmer signature under either $\mathtt{Pk}_A$ or $\mathtt{Pk}_B$. The disgintuisher continues to issuing queries, which are handled by $\mathcal{D}'$ as above. Finally, $\mathcal{D}'$ outputs the bit b' that $\mathcal{D}$ outputs.

Assume that $\mathcal{D}$ wins its game, and thus it did not send a query on input $(M^*, \sigma^*, \{\mathtt{Pk}_A, \mathtt{Pk}_B\})$ to the resolution oracle. Hence, $\mathcal{D}'$ did not make an extraction query on $(\hat{M}^*, \sigma^*, \mathtt{Spk}_0)$ nor $(\hat{M}^*, \sigma^*, \mathtt{Spk}_1)$, and wins its own game as well.

The view of $\mathcal{D}$ in this simulated game is the same as that in a real attack, except that the proof of the validity of σ^*. However, since the protocol Π is zero-knowledge, the simulated proof causes only a negligible difference to the view of $\mathcal{D}$, denoted by δ. Therefore, if $\mathcal{D}$ breaks the signer ambiguity with non-negligible advantage ϵ, $\mathcal{D}'$ breaks the anonymity of Σ with advantage at least $\epsilon - \delta$, which is non-negligible as well. □

Lemma 3. *The interactive AOFE scheme is secure against signers if* Σ *is sound and unforgeable and* Π *is sound.*

Proof. Let $\mathcal{A}$ be a malicious signer which can break the security against signers with non-negligible probability. We make use of it to construct another algorithm $\mathcal{A}'$ to break the unforgeability of Σ.

Given $(\mathtt{Cpk}, \mathtt{Csk}, \mathtt{Spk}^*)$, algorithm $\mathcal{A}'$ sets $(\mathtt{Apk}, \mathtt{Ask}) := (\mathtt{Cpk}, \mathtt{Csk})$ and $\mathtt{Pk}_B := \mathtt{Spk}^*$, and invokes the adversary $\mathcal{A}$ on input $(\mathtt{Apk}, \mathtt{Pk}_B)$. It then begins to simulate the oracles for $\mathcal{A}$ as below:

O_{KR}. Same as in the proof of Lemma 2.

O_{FakePSig}. Given $(M, \mathtt{Pk}_i)$, if $\mathtt{Pk}_i \neq \mathtt{Pk}_B$, $\mathcal{A}'$ computes the simulated signature σ using its knowledge of $\mathtt{Sk}_i$, since we are working in the registered-key model. Otherwise, it forwards $M \| \mathtt{Pk}_B$ to its signing oracle, and obtains a confirmer signature σ. In either case, $\mathcal{A}'$ returns σ to $\mathcal{A}$.

O_{Res}. Given $(M, \sigma, \mathtt{Pk}_i, \mathtt{Pk}_j)$, $\mathcal{A}'$ perfectly computes the answer using its knowledge of $\mathtt{Ask}$.

Finally, $\mathcal{A}$ outputs $(M^*, \mathtt{Pk}_A, \sigma^*)$, and starts an execution of Π with $\mathcal{A}'$ to show that σ^* is a valid confirmer signature on $\hat{M}^* := M^* \| \mathtt{Pk}_B$ under either $\mathtt{Pk}_A$ or $\mathtt{Pk}_B$. $\mathcal{A}'$ then computes

$$\zeta_A^* \leftarrow \Sigma.\mathsf{Ext}(\mathtt{Ask}, \hat{M}^*, \sigma^*, \mathtt{Pk}_A) \quad \text{and} \quad \zeta_B^* \leftarrow \Sigma.\mathsf{Ext}(\mathtt{Ask}, \hat{M}^*, \sigma^*, \mathtt{Pk}_B).$$

Suppose $\mathcal{A}$ wins the game. By the soundness of Π, we have that with overwhelming probability σ^* is indeed a valid confirmer signature under either $\mathtt{Pk}_A$ or $\mathtt{Pk}_B$, but ζ^* is not a valid standard signature under $\mathtt{Pk}_A$. Therefore, it holds that ζ_B^* is a valid standard signature under $\mathtt{Pk}_B$. $\mathcal{A}'$ then outputs $(\hat{M}^*, \zeta_B^*)$, and wins the game. If $\mathcal{A}$ succeeds in breaking the security against signers with non-negligible probability, so does $\mathcal{A}'$ in breaking the unforgeability of Σ. □

Lemma 4. *The interactive AOFE scheme is secure against the arbitrator if* Σ *is unforgeable and* Π *is zero-knowledge.*

Proof. Let $\mathcal{C}$ be a malicious arbitrator. Below we show how to use it to build an algorithm $\mathcal{C}'$ to break the unforgeability of Σ.

Given $(\mathtt{Spk}^*, \mathtt{Cpk}, \mathtt{Csk})$, $\mathcal{C}'$ sets $\mathtt{Pk}_A = \mathtt{Spk}^*$ and $(\mathtt{Apk}, \mathtt{Ask}) := (\mathtt{Cpk}, \mathtt{Csk})$, and invokes $\mathcal{C}$ on input $(\mathtt{Ask}, \mathtt{Apk}, \mathtt{Pk}_A)$. The oracle queries are answered by $\mathcal{C}'$ as below:

O_{KR}. Same as in the proof of Lemma 2.

O_{PSig}. Given $(M, \mathtt{Pk}_j)$, $\mathcal{C}'$ forwards $\hat{M} := M \| \mathtt{Pk}_j$ to its signing oracle and obtains a confirmer signature σ. It sends σ to $\mathcal{C}$ and then runs the simulator to prove to $\mathcal{C}$ that σ is a valid confirmer signature under either $\mathtt{Pk}_A$ or $\mathtt{Pk}_j$.

Finally, $\mathcal{C}$ outputs $(M^*, \mathtt{Pk}_B, \zeta^*)$. Suppose that it wins the game. We have that $1 \leftarrow \Sigma.\mathsf{Ver}(\hat{M}^*, \zeta^*, \mathtt{Spk}, \mathtt{Cpk})$, where $\hat{M}^* := M^* \| \mathtt{Pk}_B$. By the hypothesis, $\mathcal{C}$ did not issue a partial signing query on input $(M^*, \mathtt{Pk}_B)$, and hence $\mathcal{C}'$ did not send $\hat{M}^*$ to its signing oracle for a confirmer signature. If $\mathcal{C}$ succeeds in breaking the security against the arbitrator, so does $\mathcal{C}'$ in breaking the unforgeability of Σ with at least the same advantage. □

6 A New Construction of Designated Confirmer Signature

The unforgeability of DCS requires that no one but the signer can produce valid signatures, while the anonymity requires no one but the designated confirmer can tell the validity of a given confirmer signatures. Hence it is very natural to construct a DCS from a standard signature scheme Σ and a public key encryption scheme E, which is also known as the '*sign-then-encrypt*' paradigm. Many constructions of DCS follow this paradigm, such as [16, 17, 33] and etc. The difficulty of implementing the paradigm is in the design of confirmation and disavowal protocol. It is known that the protocols can be constructed generally, using complex NP reduction. However, the efficiency is a big issue. The resulting protocols may not be useful in practice.

Intuitively, in the paradigm, the confirmer holds a key pair $(\mathtt{Pk}_E, \mathtt{Sk}_E)$ for E and the signer holds a key pair $(\mathtt{Pk}_S, \mathtt{Sk}_S)$ for Σ. To sign a message M with respect to the confirmer, the signer first computes a standard signature ζ on M using $\mathtt{Sk}_E$, and then encrypts ζ under the confirmer's public key $\mathtt{Pk}_S$ to obtain the ciphertext c. Its confirmer signature is set to be c. Given c, the confirmer uses $\mathtt{Sk}_E$ to decrypt it to obtain ζ, and outputs it if it is valid under $\mathtt{Pk}_S$. In the confirmation (resp. disavowal) protocol, the confirmer proves to the verifier (interactively) that a confirmer signature c can (resp. cannot) be decrypted to the signer's valid signature on M. The unforgeability of the DCS scheme simply follows that of Σ. On the other hand, the chosen ciphertext security of E guarantees that given a ciphertext c, anyone who does not known $\mathtt{Sk}_E$, including the signer, is not able to tell the signature hidden in c belongs to which signer. Thus we have the anonymity.

Below we present a concrete and efficient instantiation of the above paradigm, which is based on Boneh-Boyen signature [6] and a variant of the linear encryption [7]. The confirmation and disavowal protocols in the construction are simple and efficient, and do not use any complex reduction. In the scheme we assume that the message space is $\mathbb{Z}_p$ for simplicity. The space can be extended to $\{0,1\}^*$ by applying a collision-resistant hash function to the message before signing.

6.1 The Construction

Let $\mathbb{G}, \mathbb{G}_T$ be two cyclic multiplicative groups of prime order p, and g a random generator of $\mathbb{G}$. Let $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be an admissible bilinear pairing and $\mathtt{H} : \mathbb{G}^3 \to \mathbb{Z}_p$ a collision-resistant hash function. Our first DCS scheme, denoted by Σ, works as follows:

Ckg. The confirmer chooses at random $F, G, K, L \in \mathbb{G}$ so that $F^{\xi_1} = G^{\xi_2} = g$ for some known $\xi_1, \xi_2 \in \mathbb{Z}_p$. It then sets $\mathtt{Apk} = (F, G, K, L)$ and $\mathtt{Ask} = (\xi_1, \xi_2)$.

SKg. The signer chooses at random $x, y \in \mathbb{Z}_p$ and computes $X = g^x$, $Y = g^y$. It sets $\mathtt{Spk} = (X, Y)$ and $\mathtt{Ssk} = (x, y)$.

Sig. To sign a message M, the signer selects at random $r \in \mathbb{Z}_p$ and computes $S = g^{1/(x+M+yr)}$. In case that $x + M + yr = 0 \bmod p$, it chooses another r and repeats the computation. Its signature on M is $\zeta = (S, r)$.

Ver. Given (M, ζ) where $\zeta = (S, r)$, the verifier checks if $\hat{e}(S, Xg^M Y^r) = \hat{e}(g, g)$. It accepts if the equation holds, and rejects otherwise.

DCSig. Given a message M, the signer randomly selects $r, s, t \in \mathbb{Z}_p$ and computes

$$S = g^{1/(x+M+yr)}, \quad \sigma_1 = F^s, \quad \sigma_2 = G^t, \quad \sigma_3 = S \cdot g^{s+t},$$
$$\sigma_4 = (g^\alpha K)^s \quad \text{and} \quad \sigma_5 = (g^\alpha L)^t,$$

where $\alpha = \mathtt{H}(\sigma_1, \sigma_2, \sigma_3)$. Again, if $x + M + yr = 0 \bmod p$, the signer chooses another r and repeats the process. Its confirmer signature on M is $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, r)$.

Ext. Given (M, σ) where $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, r)$, the confirmer computes

$$S = \sigma_3 / (\sigma_1^{\xi_1} \sigma_2^{\xi_2}) \quad \text{and} \quad \alpha = \mathtt{H}(\sigma_1, \sigma_2, \sigma_3).$$

If either of the following equations does not hold, it returns $\perp$; otherwise, it returns $\zeta = (S, r)$:

$$\hat{e}(\sigma_4, F) = \hat{e}(\sigma_1, g^\alpha K) \tag{1}$$
$$\hat{e}(\sigma_5, G) = \hat{e}(\sigma_2, g^\alpha L) \tag{2}$$
$$\hat{e}(g, g) = \hat{e}(S, Xg^M Y^r) \tag{3}$$

ConfirmS. To prove the validity of a confirmer signature $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, r)$ on a message M that it ever generated, the signer makes use of the randomness (s, t) used in the signature generation to carry out the following proof of knowledge with the verifier

$$\text{PoK}\left\{(s, t) : F^s = \sigma_1 \wedge G^t = \sigma_2 \wedge \hat{e}(\sigma_3 g^{-s-t}, Xg^M Y^r) = \hat{e}(g, g)\right\} \tag{4}$$

if both equations (1) and (2) hold, and does nothing otherwise.

ConfirmC. Given (M, σ) where $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, r)$, the confirmer and the verifier carry out the following (zero-knowledge) proof of knowledge

$$\text{PoK}\left\{(\xi_1, \xi_2) : F^{\xi_1} = g \wedge G^{\xi_2} = g \wedge \hat{e}(\sigma_3 \sigma_1^{-\xi_1} \sigma_2^{-\xi_2}, Xg^M Y^r) = \hat{e}(g, g)\right\} \tag{5}$$

if both equations (1) and (2) hold, and do nothing otherwise.

Disavow. Given (M, σ) where $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, r)$, the confirmer and the verifier carry out an execution of the following (zero-knowledge) proof of knowledge

$$\text{PoK}\left\{(\xi_1, \xi_2) : F^{\xi_1} = g \wedge G^{\xi_2} = g \wedge \hat{e}(\sigma_3 \sigma_1^{-\xi_1} \sigma_2^{-\xi_2}, Xg^M Y^r) \neq \hat{e}(g, g)\right\} \tag{6}$$

if both equations (1) and (2) hold, and do nothing otherwise.

The correctness and extraction ambiguity of the DCS scheme above can be verified trivially. The following theorem shows that the DCS scheme above is secure under the models given in Sec. 5.

Theorem 2. *The DCS scheme Σ is secure if Strong Diffie-Hellman assumption and Decision Linear assumption hold, and the hash function* $\mathtt{H}$ *is collision-resistant.*

Due to the page limit we defer the detailed proof of the theorem and the definitions of the assumptions into the full version.

6.2 Non-interactive AOFE

Huang *et al.*'s non-interactive AOFE is obtained by applying Fiat-Shamir heuristic to their interactive AOFE protocol, specifically, to the confirmation proof of the signature's validity. Via the same technique, we can obtain a non-interactive AOFE protocol as well.

7 Comparison

In Table 2 we compare the interactive AOFE protocol instantiated with the DCS scheme proposed in Section 6, with previous AOFE protocols. The second column shows if the protocol require interaction between the signer and the verifier in order to verify a partial signature. The third and fourth columns show the size of a partial signature and that of a full signature, respectively. The fifth column indicates whether the protocol works under the registered-key model or chosen-key model. The sixth column lists the basic number-theoretic assumptions used for guaranteeing the security. The last column shows whether the security of the protocols rely on the random oracle model or not.

Both of the interactive AOFE protocol proposed in [20, 21] and ours are built from a DCS scheme, and are secure in the standard model. The protocol in [20, 21] requires a special property of DCS, named *samplability*, while our protocol only makes use of standard security properties of the underlying DCS scheme. In the comparison, we consider that the signer's partial signature merely consists of its confirmer signature on the message, while leave the proof of the validity of it to the verification part. Compared with [20, 21], our protocol has longer partial signature, but smaller standard signature. In addition, the security of our protocol relies on relatively more standard assumptions, while the protocol in [20, 21] relies on newly proposed assumptions.

Table 2. Comparison with existing AOFE protocols

	interact?	Pk	Apk	PSig	Sig	PK Model	Asmp	ROM
[15]	no	$1\mathbb{G}$	$1\mathbb{G}$	$2\mathbb{G}+8\mathbb{Z}_p$	$2\mathbb{G}+12\mathbb{Z}_p$	registered	DDH	yes
[22]	no	$1\mathbb{G}$	$10\mathbb{G}$	$45\mathbb{G}+1\mathbb{Z}_p$	$46\mathbb{G}+1\mathbb{Z}_p$	chosen	SDH+DLIN	no
[20, 21]	no	$2\mathbb{G}$	$1\mathbb{G}$	$3\mathbb{G}+4\mathbb{Z}_p$	$3\mathbb{G}$	registered	HSDH+DHSDH	no
[20, 21]	yes	$163\mathbb{G}$	$1\mathbb{G}$	$3\mathbb{G}$	$3\mathbb{G}$	registered	HSDH+DHSDH	yes
Ours	yes	$2\mathbb{G}$	$4\mathbb{G}$	$5\mathbb{G}+1\mathbb{Z}_p$	$1\mathbb{G}+1\mathbb{Z}_p$	registered	SDH+DLIN	no

8 Conclusion

In this paper we showed how to build an interactive ambiguous optimistic fair exchange protocol using a designated confirmer signature scheme with slight modifications. The resulting protocol is almost as efficient as the underlying DCS scheme. It makes use of standard security properties of the underlying DCS, and is secure without random oracles. We also proposed a concrete and efficient construction of designated confirmer signature, which is secure based on Strong Diffie-Hellman assumption and decision linear assumption without random oracles, and to the best of our knowledge has the shortest standard signature.

References

1. Asokan, N., Schunter, M., Waidner, M.: Optimistic protocols for fair exchange. In: CCS, pp. 7–17. ACM (1997)
2. Asokan, N., Shoup, V., Waidner, M.: Optimistic Fair Exchange of Digital Signatures (Extended Abstract). In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 591–606. Springer, Heidelberg (1998)
3. Asokan, N., Shoup, V., Waidner, M.: Optimistic fair exchange of digital signatures. IEEE Journal on Selected Areas in Communication 18(4), 593–610 (2000)
4. Barak, B., Canetti, R., Nielsen, J.B., Pass, R.: Universally composable protocols with relaxed set-up assumptions. In: FOCS 2004, pp. 186–195. IEEE Computer Society (2004)
5. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: CCS, pp. 62–73. ACM (1993)
6. Boneh, D., Boyen, X.: Short Signatures Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004)
7. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
8. Camenisch, J., Michels, M.: Confirmer Signature Schemes Secure against Adaptive Adversaries (Extended Abstract). In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 243–258. Springer, Heidelberg (2000)
9. Chaum, D.: Zero-Knowledge Undeniable Signatures. In: Damgård, I.B. (ed.) EUROCRYPT 1990. LNCS, vol. 473, pp. 458–464. Springer, Heidelberg (1991)
10. Chaum, D.: Designated Confirmer Signatures. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 86–91. Springer, Heidelberg (1995)
11. Chen, L.: Efficient Fair Exchange with Verifiable Confirmation of Signatures. In: Ohta, K., Pei, D. (eds.) ASIACRYPT 1998. LNCS, vol. 1514, pp. 286–299. Springer, Heidelberg (1998)
12. Damgård, I.: On Σ-protocols. Course on Cryptologic Protocol Theory. Aarhus University (2009), http://www.daimi.au.dk/~ivan/Sigma.pdf
13. Dodis, Y., Lee, P.J., Yum, D.H.: Optimistic Fair Exchange in a Multi-user Setting. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 118–133. Springer, Heidelberg (2007); also at Cryptology ePrint Archive, Report 2007/182

14. Dodis, Y., Reyzin, L.: Breaking and repairing optimistic fair exchange from PODC 2003. In: DRM 2003, pp. 47–54. ACM (2003)
15. Garay, J.A., Jakobsson, M., MacKenzie, P.: Abuse-free optimistic contract signing. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 449–466. Springer, Heidelberg (1999)
16. Gentry, C., Molnar, D., Ramzan, Z.: Efficient Designated Confirmer Signatures Without Random Oracles or General Zero-Knowledge Proofs (Extended Abstract). In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 662–681. Springer, Heidelberg (2005)
17. Goldwasser, S., Waisbard, E.: Transformation of Digital Signature Schemes into Designated Confirmer Signature Schemes. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 77–100. Springer, Heidelberg (2004)
18. Groth, J.: Fully Anonymous Group Signatures Without Random Oracles. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 164–180. Springer, Heidelberg (2007); also at Cryptology ePrint Archive, Report 2007/186
19. Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
20. Huang, Q., Wong, D.S., Susilo, W.: A New Construction of Designated Confirmer Signature and Its Application to Optimistic Fair Exchange - (Extended Abstract). In: Joye, M., Miyaji, A., Otsuka, A. (eds.) Pairing 2010. LNCS, vol. 6487, pp. 41–61. Springer, Heidelberg (2010)
21. Huang, Q., Wong, D.S., Susilo, W.: Efficient designated confirmer signature and DCS-based ambiguous optimistic fair exchange. IEEE Transactions on Information Forensics and Security 6(4), 1233–1247 (2011)
22. Huang, Q., Yang, G., Wong, D.S., Susilo, W.: Ambiguous Optimistic Fair Exchange. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 74–89. Springer, Heidelberg (2008)
23. Huang, Q., Yang, G., Wong, D.S., Susilo, W.: Efficient Optimistic Fair Exchange Secure in the Multi-user Setting and Chosen-Key Model without Random Oracles. In: Malkin, T. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 106–120. Springer, Heidelberg (2008)
24. Jakobsson, M., Sako, K., Impagliazzo, R.: Designated Verifier Proofs and Their Applications. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 143–154. Springer, Heidelberg (1996)
25. Lu, S., Ostrovsky, R., Sahai, A., Shacham, H., Waters, B.: Sequential Aggregate Signatures and Multisignatures Without Random Oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 465–485. Springer, Heidelberg (2006)
26. Lysyanskaya, A., Micali, S., Reyzin, L., Shacham, H.: Sequential Aggregate Signatures from Trapdoor Permutations. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 74–90. Springer, Heidelberg (2004)
27. Michels, M., Stadler, M.: Generic Constructions for Secure and Efficient Confirmer Signature Schemes. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 406–421. Springer, Heidelberg (1998)
28. Okamoto, T.: Designated Confirmer Signatures and Public-Key Encryption Are Equivalent. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 61–74. Springer, Heidelberg (1994)
29. Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)

30. Park, J.M., Chong, E.K., Siegel, H.J.: Constructing fair-exchange protocols for e-commerce via distributed computation of RSA signatures. In: PODC 2003, pp. 172–181. ACM (2003)
31. Wang, G., Baek, J., Wong, D.S., Bao, F.: On the Generic and Efficient Constructions of Secure Designated Confirmer Signatures. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 43–60. Springer, Heidelberg (2007)
32. Wang, G., Xia, F.: A pairing based designated confirmer signature scheme with unified verification. Technical report, School of Computer Science, University of Birmingham (December 2009)
33. Wikström, D.: Designated confirmer signatures revisited. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 342–361. Springer, Heidelberg (2007)
34. Zhang, F., Chen, X., Wei, B.: Efficient designated confirmer signature from bilinear pairings. In: ASIACCS 2008. pp. 363–368. ACM (2008)

Efficient Implementation of a CCA2-Secure Variant of McEliece Using Generalized Srivastava Codes

Pierre-Louis Cayrel[1], Gerhard Hoffmann[2], and Edoardo Persichetti[3]

[1] Université Jean Monnet, Saint-Etienne, France
[2] Technische Universität Darmstadt, Germany
[3] University of Auckland, New Zealand

Abstract. In this paper we present efficient implementations of McEliece variants using quasi-dyadic codes. We provide secure parameters for a classical McEliece encryption scheme based on quasi-dyadic generalized Srivastava codes, and successively convert our scheme to a CCA2-secure protocol in the random oracle model applying the Fujisaki-Okamoto transform. In contrast with all other CCA2-secure code-based cryptosystems that work in the random oracle model, our conversion does not require a constant weight encoding function. We present results for both 128-bit and 80-bit security level, and for the latter we also feature an implementation for an embedded device.

1 Introduction

The McEliece and Niederreiter public-key encryption schemes are based on error-correcting codes. One drawback are the large public keys. There have been few implementations reported; we cite for instance [29] and [30] for 32-bit software implementations. An alternative scheme, called HyMES (Hybrid McEliece cryptosystem), was implemented by Sendrier and Biswas [11], combining ideas from both the previous schemes.

Recently, implementations of the McEliece and Niederreiter cryptosystems for embedded devices have been presented, respectively by Eisenbarth et al. in [13] and by Heyse in [18], with the disadvantage of an external memory requirement for storing the key. A first proposal to deal with this issue from an implementational point of view is to make use of the quasi-dyadic variant of Misoczki and Barreto [25]. This was done by Heyse in [19], along with the extension to a CCA2-secure protocol. Unfortunately, the fields underlying the Goppa codes chosen are still too big to fit on the flash memory of the embedded device and this has repercussions in the speed of the implementation, since the use of tower field arithmetic becomes necessary.

In our paper, we provide an alternative construction based on the more general framework of generalized Srivastava codes described by Persichetti in [27]. We then convert the encryption scheme into a CCA2-secure protocol with the help of the Fujisaki-Okamoto transform [17]. To the best of our knowledge, a scheme

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 138–155, 2012.

based on this family of codes has never been implemented before; moreover, we use McEliece with a twist, and we don't require any constant weight encoding function [32] for our conversion. This is also a novelty, and it allows to simplify the construction and save computational costs at the same time. The finite fields in use are much smaller than previous proposals, and fit completely on the flash memory, with the result that our implementation is much faster.

We note that there exist schemes, such as Dowsley et al. [12] and Freeman et al. [22], that provide CCA2-secure encryption based on coding theory in the standard model, but these schemes are completely impractical.

The paper is organized as follows: in Section 2 the McEliece and Niederreiter encryption schemes are introduced, along with an overview of constructions based on structured matrices. Security definitions such as IND-CCA2 and their instantiations are discussed in Section 3, and the technical details about the implementations with the respective timings are provided in Section 4, both for a C++ code, and for implementation on an embedded device. Finally, we conclude in Section 5.

2 Code-Based Public-Key Encryption Schemes

2.1 The McEliece Cryptosystem

The first cryptosystem based on coding theory was introduced in 1978 by Robert J. McEliece [23] and, for an appropriate choice of parameters, is still unbroken. In the original proposal, binary Goppa codes are used as a basis for the construction, and the security comes from the hardness of the General Decoding Problem (GDP).

Definition 1 (GDP). *Let C be an $[n, k]$ linear code over $\mathbb{F}_q$ and let y be a vector of $\mathbb{F}_q^n$.*

Find the codeword closest to y, i.e. find $c \in \mathsf{C}$ such that $d(c, y)$ is minimal.

This corresponds to correcting a certain number of errors occurred on the codeword c, represented by an error vector e, that is $y = c + e$. A unique solution exists if the weight of e is less than or equal to $w = \lfloor \frac{d-1}{2} \rfloor$, where d is the *minimum distance* of the code C.

This problem is well known and was proved to be NP-complete [7]. Moreover, GDP is believed to be hard on average, and not just on the worst-case instances. The general framework proceeds as follows:

Key Generation: Pick a $k \times n$ generator matrix G for a w-error correcting linear code with an efficient decoding algorithm over the finite field $\mathbb{F}_q$, a $k \times k$ invertible matrix S and an $n \times n$ permutation matrix P at random, then compute $G' = SGP$, which is another valid generator matrix. The private key consists of G, S, P, and the public key is G'.The system parameters n, k, w are also public.

Encryption: To encrypt a plaintext $x \in \mathbb{F}_q^k$, compute the corresponding codeword xG' and add a random error vector e of weight at most w, obtaining the ciphertext $y = xG' + e$.

Decryption: Given a ciphertext y, calculate $yP^{-1} = xG'P^{-1} + eP^{-1} = xSG + eP^{-1}$, and since the weight of eP^{-1} is still the same, it is enough to apply the decoding algorithm for the code to retrieve xS and consequently x.

The other computational assumption underlying the security is that the $k \times n$ matrix G' so obtained is computationally indistinguishable from a uniform matrix of the same size, hence an attacker that does not know the private key is faced with solving GDP.

Remark. The encryption process is dominated by the cost of computing xG', which requires at most $k \times n$ field multiplications. Hence this is fast. On the other hand, decryption requires performing a decoding algorithm and is not usually so fast. Therefore, McEliece is most suitable for applications where encryption is required to be fast. This is analogous to RSA using small encryption exponents.

2.2 The Niederreiter Cryptosystem

A first alternative version of the McEliece cryptosystem has been proposed by Niederreiter [26] in 1986, and has been proved to be equivalent in terms of security. It is often considered as a "dual" version, as the trapdoor is given by the parity-check matrix rather than the generator matrix. The underlying hard problem is the Syndrome Decoding Problem.

Definition 2 (SDP). *Let H be an $r \times n$ matrix over $\mathbb{F}_q$, s a vector of $\mathbb{F}_q^r$ and $w > 0$.*

Find a vector e in $\mathbb{F}_q^n$ of weight $\leq w$ such that $He^T = s$.

If H is the parity-check matrix for an $[n, k]$ linear code C, then $r = n - k$ and it is immediate to see that the two problems are equivalent: in fact, for $y = c + e$ we have $Hy^T = Hc^T + He^T$ but $Hc^T = 0$ since c is a codeword so $Hy^T = He^T = s$, which means that SDP in this case corresponds, again, to finding an error vector of weight less or equal to w.

This is a description of Niederreiter's scheme:

Key Generation: Pick an $(n - k) \times n$ parity-check matrix H for a w-error correcting linear code with an efficient decoding algorithm over the finite field $\mathbb{F}_q$, an $(n - k) \times (n - k)$ invertible matrix S and an $n \times n$ permutation matrix P at random, then evaluate $H' = SHP$, which is another valid parity-check matrix. The private key consists of H, S, P, and the the public key is H'.The system parameters n, k, w are also public.

Encryption: A plaintext here is a vector $e \in \mathbb{F}_q^n$ of weight at most w; to encrypt, compute the corresponding syndrome, obtaining the ciphertext $y = H'e^T$.

Decryption: Given a ciphertext y, calculate first $S^{-1}y = HPe^T$, and then apply the decoding algorithm for the code to retrieve Pe^T and consequently e.

2.3 Structured Matrices

Definition 3. *Given a ring R (in our case the finite field $\mathbb{F}_{q^m}$) and a vector $\bar{h} = (h_0, \ldots, h_{n-1}) \in R^n$, the dyadic matrix $\Delta(\bar{h}) \in R^{n \times n}$ is the symmetric matrix with components $\Delta_{ij} = h_{i \oplus j}$, where $\oplus$ stands for bitwise exclusive-or on the binary representations of the indices. The sequence $\bar{h}$ is called its signature. Moreover, $\Delta(t, \bar{h})$ denotes the matrix $\Delta(\bar{h})$ truncated to its first t rows. Finally, we call a matrix quasi-dyadic if it is a block matrix whose component blocks are $t \times t$ dyadic submatrices.*

If n is a power of 2, then every $2^k \times 2^k$ dyadic matrix can be described recursively as

$$M = \begin{pmatrix} A & B \\ B & A \end{pmatrix}$$

where each block is a $2^{k-1} \times 2^{k-1}$ dyadic matrix (and where any 1×1 matrix is dyadic).

Definition 4. *Given two sequences $\bar{x} = (x_1, \ldots, x_n)$, $\bar{y} = (y_1, \ldots, y_n) \in \mathbb{F}_q^n$, a* Generalized Reed-Solomon (GRS) code *of order ℓ is defined by a parity-check matrix related to the Vandermonde form, i.e. the matrix with components $H_{ij} = y_j x_j^{i-1}$:*

$$H = \begin{pmatrix} y_1 & \cdots & y_n \\ y_1 x_1 & \cdots & y_n x_n \\ \vdots & \vdots & \vdots \\ y_1 x_1^{\ell-1} & \cdots & y_n x_n^{\ell-1} \end{pmatrix}.$$

If the resulting code is then restricted to $\mathbb{F}_q$ it is called an Alternant code.

Definition 5. *For $m, n, s, t \in \mathbb{N}$ and a prime power q, let $\bar{\alpha} = (\alpha_1, \ldots, \alpha_n)$, $\bar{w} = (w_1, \ldots, w_s)$ be $n + s$ distinct elements of $\mathbb{F}_{q^m}$, and $(z_1, \ldots, z_n)$ be nonzero elements of $\mathbb{F}_{q^m}$. The* Generalized Srivastava (GS) code *of order st and length n is defined by a parity-check matrix of the form:*

$$H = \begin{pmatrix} H_1 \\ H_2 \\ \vdots \\ H_s \end{pmatrix}$$

where each block is

$$H_i = \begin{pmatrix} \dfrac{z_1}{\alpha_1 - w_i} & \cdots & \dfrac{z_n}{\alpha_n - w_i} \\ \dfrac{z_1}{(\alpha_1 - w_i)^2} & \cdots & \dfrac{z_n}{(\alpha_n - w_i)^2} \\ \vdots & \vdots & \vdots \\ \dfrac{z_1}{(\alpha_1 - w_i)^t} & \cdots & \dfrac{z_n}{(\alpha_n - w_i)^t} \end{pmatrix}.$$

The parameters for such a code are the length $n \leq q^m - s$, dimension $k \geq n - mst$ and minimum distance $d \geq st + 1$.

GS codes are part of the family of Alternant codes, and therefore benefit of an efficient decoding algorithm. More information about this class of codes can be found in [21, Ch. 12, §6].

2.4 Secure Parameters

Both the previous schemes share some common traits: a very fast and efficient encryption procedure, and very big public keys. Our proposal to deal with these issues is to use structured codes, and in particular, quasi-dyadic codes. See Appendix B for a summary of the key generation process.

Misoczki and Barreto in [25] give an assessment of the hardness of decoding quasi-dyadic codes, providing a reduction to the Syndrome Decoding Problem.

Keeping in mind the scope of the paper, the parameters proposed in [27, Table 3] seem to fit our proposal best; we report the table here for completeness.

Table 1. Quasi-dyadic GS codes [27, Table 3]. The column "Size" indicates the size of the public key, while in the column "Security level" are reported the approximate cost of general decoding attacks ($\log_2$ of binary operations).

Base Field	m	n	k	s	t	Errors	Size (bytes)	Security level[1]
$\mathbb{F}_{2^5}$	2	992	416	2^5	9	144	4680	128
$\mathbb{F}_{2^4}$	3	768	432	2^4	7	56	4536	80
$\mathbb{F}_{2^5}$	2	512	256	2^4	2^3	64	2560	80

3 CCA-Secure Schemes

Until now, we have been considering only the weakest notion of security for a public-key encryption scheme, that is, One-Way Encryption (OWE). The following are formal definitions of public-key encryption and one-way security.

Definition 6. *A Public-Key Encryption (PKE) scheme consists of a 6-tuple* $(\mathcal{K}, \mathcal{P}, \mathcal{C}, \mathcal{G}, \mathcal{E}, \mathcal{D})$ *defined as follows:*

- $\mathcal{K} = \mathcal{K}_{publ} \times \mathcal{K}_{priv}$ *is the* key space.
- $\mathcal{P}$ *is the set of messages to be encrypted, or* plaintext space.
- $\mathcal{C}$ *is the set of the messages transmitted over the channel, or* ciphertext space.
- $\mathcal{G}$ *is a probabilistic key generation algorithm that takes as input a security parameter* 1^δ *and outputs a public key* $pk \in \mathcal{K}_{publ}$ *and a private key* $sk \in \mathcal{K}_{priv}$.

[1] http://www2.mat.dtu.dk/people/C.Peters/isdfq.html

- $\mathcal{E}$ *is a (possibly probabilistic) encryption algorithm that receives as input a public key* $pk \in \mathcal{K}_{publ}$ *and a plaintext* $x \in \mathcal{P}$ *and returns a ciphertext* $\psi \in \mathcal{C}$.
- $\mathcal{D}$ *is a deterministic decryption algorithm that receives as input a private key* $sk \in \mathcal{K}_{priv}$ *and a ciphertext* $\psi \in \mathcal{C}$ *and outputs either a plaintext* $x \in \mathcal{P}$ *or the failure symbol* $\perp$.

Definition 7 (One-Way). *A One-Way adversary is a polynomial-time algorithm* $\mathcal{A}$ *that takes as input a public key* $pk \in \mathcal{K}_{publ}$ *and a ciphertext* $\psi \in \mathcal{C}$. *We say that a PKE is* One-Way Secure *if the probability of success of any adversary* $\mathcal{A}$ *is negligible in the security parameter, i.e.*

$$Pr[pk \leftarrow \mathcal{K}_{publ}, x \leftarrow \mathcal{P} : \mathcal{A}(pk, \mathcal{E}_{pk}(x)) = x] \in negl(\delta)$$

The standard definitions for *Indistinguishability*, and the attack models CPA and CCA2 are omitted here due to space requirements.

3.1 CCA2 Security Conversions

There are standard ways to obtain an IND-CCA2 secure encryption scheme from one that only has OW-CPA, for example the Fujisaki-Okamoto transform [17]. The construction achieves CCA2-security by integrating an asymmetric encryption scheme with a symmetric scheme.

Definition 8. *A Symmetric Encryption (SE) scheme consists of a 5-tuple* $(\mathcal{K}, \mathcal{P}, \mathcal{C}, \mathcal{E}, \mathcal{D})$ *defined as follows:*

- $\mathcal{K}$ *is the* key space.
- $\mathcal{P}$ *is the set of messages to be encrypted, or* plaintext space.
- $\mathcal{C}$ *is the set of the messages transmitted over the channel, or* ciphertext space.
- $\mathcal{E}$ *is a deterministic encryption algorithm that receives as input a key* $\chi \in \mathcal{K}$ *and a plaintext* $x \in \mathcal{P}$ *and returns a ciphertext* $\psi \in \mathcal{C}$.
- $\mathcal{D}$ *is a deterministic decryption algorithm that receives as input a key* $\chi \in \mathcal{K}$ *and a ciphertext* $\psi \in \mathcal{C}$ *and outputs a plaintext* $x \in \mathcal{P}$.

The Fujisaki-Okamoto conversion requires an additional property of the encryption scheme called γ-$uniformity$. We define it here.

Definition 9. *Let* Π *be a PKE defined as above and let's call* $\mathcal{R}$ *the set where the randomness to be used in the (probabilistic) encryption is chosen. For given* $(pk, sk) \in \mathcal{K}$, $x \in \mathcal{P}$ *and a string* y, *we define*

$$\gamma(x, y) = Pr[r \xleftarrow{\$} \mathcal{R} : y = \mathcal{E}_{pk}(x, r)]$$

where the notation $\mathcal{E}_{pk}(x, r)$ *makes explicit the role of the randomness* r. *We say that* Π *is* γ**-uniform** *if, for any* $(pk, sk) \in \mathcal{K}$, *any* $x \in \mathcal{P}$ *and any* y, $\gamma(x, y) \leq \gamma$ *for a certain* $\gamma \in \mathbb{R}$.

Table 2. The Fujisaki-Okamoto conversion. $\mathcal{H}_1$ and $\mathcal{H}_2$ are hash functions.

Encryption of x	Decryption of ψ
$\sigma \xleftarrow{\$} \mathcal{P}^{PKE}$	$\psi := (\psi_1 \| \psi_2)$
$r := \mathcal{H}_1(\sigma, x)$	$\hat{\sigma} := \mathcal{D}_{sk}^{PKE}(\psi_1)$ (return $\perp$ if decryption fails)
$\psi_1 := \mathcal{E}_{pk}^{PKE}(\sigma, r)$	$\hat{x} := \mathcal{D}_{\mathcal{H}_2(\hat{\sigma})}^{SE}(\psi_2)$ (return $\perp$ if decryption fails)
$\psi_2 := \mathcal{E}_{\mathcal{H}_2(\sigma)}^{SE}(x)$	$\hat{r} := \mathcal{H}_1(\hat{\sigma}, \hat{x})$
	if $\mathcal{E}_{pk}^{PKE}(\hat{\sigma}, \hat{r}) == \psi_1$ return $x := \hat{x}$
return $\psi := (\psi_1 \| \psi_2)$	else return $\perp$

In a successive paper [20], Kobara and Imai proposed three alternative constructions in a similar fashion, tailored specifically for the McEliece cryptosystem rather than a general OWE encryption scheme. The biggest contribution of the new constructions is that the amount of overhead data (i.e. difference between the bit-length of the ciphertext and the bit-length of the plaintext) is considerably reduced.

While this is certainly an important issue for some applications, in the common cryptographic practice it will never constitute a serious concern. In fact, the aim of public key cryptography is not to encrypt a whole, large plaintext, but rather to encrypt just a small (e.g. 128 or 256 bits) key for a more efficient symmetric scheme, that will be then used to encrypt the message. From a computational point of view the Kobara-Imai encryption process seems to be more expensive; in fact, the whole construction is rather complex.

Table 3. The Kobara-Imai hybrid conversion γ for the McEliece (McE) public-key encryption scheme. $\mathcal{H}$ is a hash function, Gen a random number generator, $Conv$ a constant weight encoding function and $Const$ a (predetermined) public constant.

Encryption of x	Decryption of ψ
$r \xleftarrow{\$} \{0,1\}^*$	$\psi := (y_5 \| y')$
$y_1 := Gen(r) \oplus (x \| Const)$	$y_3 := \mathcal{D}_{sk}^{McE}(y')$
$y_2 := r \oplus \mathcal{H}(y_1)$	$y_3 G' \oplus y'$
$(y_5 \| y_4 \| y_3) := (y_2 \| y_1)$	$y_4 := Conv^{-1}(z)$
$z := Conv(y_4)$	$(y_2 \| y_1) := (y_5 \| y_4 \| y_3)$
	$r := y_2 \oplus \mathcal{H}(y_1)$
	$(\hat{x} \| Const') := y_1 \oplus Gen(r)$
	if $Const' == Const$ return $x := \hat{x}$
return $\psi := (y_5 \| \mathcal{E}_{pk}^{McE}(y_3, z))$	else return $\perp$

Note that the Fujisaki-Okamoto decryption process includes an encoding operation in the final check. This makes decryption slower. The cost of the process, though, is still dominated by the decoding operation rather than the

matrix-vector multiplication. Moreover, as we already remarked, we argue that the distinctive feature of the McEliece scheme is the fast encryption process, and the Fujisaki-Okamoto conversion preserves fast encryption better than the Kobara-Imai approach.

3.2 Applying Fujisaki-Okamoto to McEliece

We give here a new way to use McEliece together with the Fujisaki-Okamoto transform. Previous approaches always needed a constant weight encoding function to convert $\mathcal{H}_1(\sigma, x)$ into an error vector. Our idea is to swap the message and the error in the McEliece scheme, with a technique similar to the one used by Micciancio in [24]. This means that we interpret $\mathcal{E}_{G'}^{McE}(x, r) = rG' + x$, encoding the message in the error vector rather than in the codeword. This is possible because, unlike other PKE's, the decryption process of McEliece, consisting mainly of decoding, returns both x and r, allowing to recover, in addition to the plaintext, also the randomness used. With this simple trick, we avoid having to use a (costly) constant weight encoding function and we simplify the encryption process considerably.

For simplicity we take the symmetric encryption scheme to be the one-time pad with an ephemeral key generated as $\mathcal{H}_2(\sigma)$ where $\mathcal{H}_2$ is a random oracle with arbitrary length output. This symmetric encryption scheme satisfies the Find-Guess security property. In practice, one might use a block cipher in CBC mode.

Table 4. The Fujisaki-Okamoto transform applied to McEliece

Encryption of x	Decryption of ψ
$\sigma \xleftarrow{\$} \mathcal{W}_{n,w}$	$\psi := (\psi_1 \| \psi_2)$
$r := \mathcal{H}_1(\sigma \| x)$	$\hat{\sigma} := \mathcal{D}_G^{McE}(\psi_1)$ (return $\perp$ if decoding fails)
$\psi_1 := rG' + \sigma$	$\hat{x} = \mathcal{H}_2(\hat{\sigma}) \oplus \psi_2$
$\psi_2 := \mathcal{H}_2(\sigma) \oplus x$	$\hat{r} := \mathcal{H}_1(\hat{\sigma} \| \hat{x})$
	if $\hat{r}G' + \hat{\sigma} == \psi_1$ return $x := \hat{x}$
return $\psi := (\psi_1 \| \psi_2)$	else return $\perp$

The following lemma is fundamental to prove that our scheme enjoys the γ-uniformity required by the conversion.

Lemma 1. *The McEliece encryption scheme described above is γ-uniform for* $\gamma = \dfrac{1}{q^k}$.

Proof. Let G' be a public key that is a generator matrix for the code C; in our setting, y is a generic string in $\mathbb{F}_q^n$. Then clearly:

$$\gamma(\sigma, y) = Pr[r \xleftarrow{\$} \mathbb{F}_q^k : y = rG' + \sigma] = \begin{cases} 0 \text{ if } y - \sigma \notin \mathsf{C} \\ \\ \dfrac{1}{q^k} \text{ if } y - \sigma \in \mathsf{C} \end{cases}$$

and that concludes the proof. □

Theorem 1. *If the assumptions of indistinguishability and decoding hardness of the McEliece PKE hold, the encryption scheme described in Table 4 is IND-CCA2 secure.*

Proof. The scheme enjoys one-way security because of the computational assumptions in the hypothesis. Moreover, Lemma 1 provides the γ-uniformity as required. Finally, the symmetric scheme used (one-time pad) satisfies the required security property (Find-Guess). It is then possible to apply [17, Th. 12]. □

4 Efficient Implementation

The implementation was done in C++ and is based on the library *SBCrypt* (Syndrome-Based Cryptography Library) by Barreto, Misoczki and Villas Boas [3].

We subsequently converted our code to run on an embedded device, namely the microcontroller ATxmega256A3 from the AVR XMEGA family. It has 264 Kbytes of Flash memory, 16 Kbytes of SRAM memory and is running at a clock frequency of 32 MHz.

To represent the finite fields we used exponential/antilog tables [21, Ch. 4, §5], which is possible as our extension fields are small enough to fit completely in the available memory (apart from the first code, for which the private trapdoor would be too big). This is a key feature of our scheme and one of the main reasons to choose GS codes over Goppa codes. In fact, when using GS codes, it is possible to choose secure parameters even for codes defined over relatively small extension fields. See Appendix C for a summary of the security discussion. More information can be found in [27].

As for the hash functions $\mathcal{H}_1$ and $\mathcal{H}_2$, we opted for the Keccak family [10], one of the five remaining SHA-3 finalists, with assigned output length equal to k, in the first instance, or equal to the plaintext length (128 bits in our case), in the second. Its flexibility also allows for using it as stream cipher, and we deployed it for randomly choosing error vectors of weight w.

The procedure to generate error vectors for encryption is as follows: at first, the error vector is initialized to zero. Next, we ask Keccak for $\beta = \lceil \log_2 n \rceil$ bits and interpret the result as an index into the error vector. If the interval is greater than n then we reject and re-sample. Now, in case this index is still a zero entry, we ask Keccak for additional bits to be read as a field element. Otherwise, we

ask Keccak for the next bits to be interpreted as the next index to be examined. This simple procedure is iterated until the error vector has the desired weight.

It is clear that this process samples uniformly from $\mathcal{W}_{n,w}$.

The test results for the C++ code have been executed on an Intel(R) Core (TM) 2 Duo CPU E8400@3.00GHz running Ubuntu/Linux 2.6.32, where the source has been compiled with gcc 4.4.3. Similar results have been obtained using the Intel compiler icpc/icc. As for the embedded microcontroller, the code has been simulated on AVR Studio 5.0 [1].

McEliece Based on GS Codes. We have measured two different operations: the encoding step $xG + e$ for $x \in \mathbb{F}_q^k$ and the decoding of a ciphertext $y \in \mathbb{F}_q^n$. Results are presented in Table 5 (timings expressed in milliseconds (ms)).

Table 5. Profiling results for McEliece using GS codes

Code Name	Base Field	m	n	k	s	t	Errors	Encoding	Decoding
$\mathfrak{A}$	$\mathbb{F}_{2^5}$	2	992	416	2^5	9	144	0.287	5.486
$\mathfrak{B}$	$\mathbb{F}_{2^4}$	3	768	432	2^4	7	56	0.179	1.578
$\mathfrak{C}$	$\mathbb{F}_{2^5}$	2	512	256	2^4	2^3	64	0.093	1.234

It is easy to see that the decoding process dominates the runtime.

The following tables report the results obtained when running the same operations on the microcontroller, for the last two codes. The costs displayed are

Table 6. Details of the costs of encryption and decryption steps for codes $\mathfrak{B}$ and $\mathfrak{C}$

Operation	**Code $\mathfrak{B}$**	**Code $\mathfrak{C}$**
Generate error vector e	313,114	316,568
Load the plaintext x	4,313	2,553
Encode xG	3,418,292	1,603,854
Add e	8,818	5,944
Encoding total	3,744,537	1,928,919

Operation	**Code $\mathfrak{B}$**	**Code $\mathfrak{C}$**
Compute syndrome Hy^T	6,910,742	5,440,245
Solve key equation	955,597	1,192,400
Compute error positions	2,061,066	1,571,689
Compute error values	611,898	794,463
Correct the errors	8,641	5,121
Decoding total	10,547,944	9,003,918

in clock cycles; for a conversion to the standard time units, keep in mind that the device runs at 32MHz, hence we have 32 million cycles per second.

Note on Decoding. In our scheme, we have implemented a standard alternant decoder (see for example [21, Ch. 12, §9]). That consists of extrapolating the key equation from the syndrome and then solve it and compute the error positions as the roots of the error locator polynomial. To find the roots, we use the Horner scheme in the sense that we directly evaluate the polynomial on the support. More sophisticated root-finding algorithms are available, for instance Berlekamp's trace algorithm [6]. However, our codes are punctured codes, and, as also stated in [19], Berlekamp's trace algorithm is not designed for such a case. Moreover, although Berlekamp's algorithm does find the roots of the polynomial, there is an additional step necessary to find them in the support sequence, which is not the case when using the Horner scheme and direct evaluation. Finally, one can see from the timings of the decoding operation, that the by far dominating part is the syndrome computation. For the time being, we therefore refrained from implementing Berlekamp's algorithm, opting for the much simpler Horner scheme instead.

CCA2-McEliece Based on GS Codes. The performances of the scheme are given in Table 7 and Table 8, respectively for the C++ code and for the microcontroller.

Table 7. Profiling results for CCA2-McEliece using GS codes

Code Name	Base Field	m	n	k	s	t	Errors	Encryption	Decryption
$\mathfrak{A}$	$\mathbb{F}_{2^5}$	2	992	416	2^5	9	144	0.323	5.914
$\mathfrak{B}$	$\mathbb{F}_{2^4}$	3	768	432	2^4	7	56	0.213	1.814
$\mathfrak{C}$	$\mathbb{F}_{2^5}$	2	512	256	2^4	2^3	64	0.114	1.382

Table 8. Details of the costs of the encryption and decryption steps of CCA2-McEliece

Operation	**Code $\mathfrak{B}$**	**Code $\mathfrak{C}$**
Generate error vector σ	322,109	321,812
Load the plaintext x	1,019	1,019
Hash $r = \mathcal{H}(\sigma, x)$	282,285	281,497
Encode rG	3,426,700	1,591,031
Add σ	1,103	1,314
Hash $\mathcal{K}(\sigma)$	137,704	137,720
Pad $\mathcal{K}(\sigma) \oplus x$	1,814	1,811
Encryption total	4,171,734	2,336,204

Table 8. *(Continued)*

Operation	**Code $\mathfrak{B}$**	**Code $\mathfrak{C}$**
Compute syndrome $H\psi_1^T$	7,029,985	5,425,696
Solve key equation	954,522	1,202,032
Compute error positions	2,031,514	1,561,946
Compute error values	611,944	794,524
Correct the errors	1,108	5,112
Hash $\mathcal{K}(\hat{\sigma})$	147,822	144,768
Pad $\mathcal{K}(\hat{\sigma}) \oplus \psi_2$	1,585	1,586
Hash $\hat{r} = \mathcal{H}(\hat{\sigma}, \hat{x})$	282,066	282,278
Encode $\hat{r}G$	3,426,721	1,591,049
Add $\hat{\sigma}$	1,113	1,273
Check equality	9,207	6,135
Decryption total	14,497,587	11,016,399

Comparing the results in Table 5 and Table 7 (as well as Table 6 and Table 8), we see that indeed the computational overhead is quite low.

For simplicity, the comparison of the total timings for both cases is reported in Tables 9 and 10.

Table 9. Summary of the timings (ms) for the C++ code

Code	Encoding	CCA2 Encryption	Decoding	CCA2 Decryption
$\mathfrak{A}$	0.287	0.323	5.486	5.914
$\mathfrak{B}$	0.179	0.213	1.578	1.814
$\mathfrak{C}$	0.093	0.114	1.234	1.382

Table 10. Summary of the timings (clock cycles) for the embedded device

Code	Encoding	CCA2 Encryption	Decoding	CCA2 Decryption
$\mathfrak{B}$	3,744,537	4,171,734	10,547,944	14,497,587
$\mathfrak{C}$	1,928,919	2,336,204	9,003,918	11,016,399

5 Conclusions

In this paper we propose the implementation of a construction based on quasi-dyadic generalized Srivastava codes. We first implement a plain McEliece encryption scheme, and then convert it to a CCA2-secure scheme using the Fujisaki-Okamoto transform. The results are initially given for a C++ implementation, and successively for an embedded device.

An independent work proposing a CCA2-secure scheme based on quasi-dyadic Goppa codes has been recently presented at PQCrypto 2011 by Stefan Heyse

[19]. The performance indicated for encryption and decryption on the embedded device are slower than our results (the simulator program is the same, AVR Studio, although in a slightly older version). Part of the reason is due to the use a constant weight encoding function (more than three times as costly as hashing) that we avoid thanks to the particular configuration of our scheme. However, the major difference comes from the fact that our vector-matrix multiplication, despite performing operations over non-binary fields, is at least two times faster, and this is the dominating part in the encryption process and is also a very high cost in the decryption process. This is a direct consequence of the structure of the scheme. In fact, the construction in [19] makes use of binary Goppa codes, which for security reasons [14] need to be defined over the extension field $\mathbb{F}_{2^{16}}$: this is too big to fit the corresponding log/antilog tables on the flash memory of the device. The result is that, in order to avoid using additional, external memory, the tables for $\mathbb{F}_{2^8}$ are represented instead, and operations are performed using tower field arithmetic, which is much slower. For example, a multiplication over a tower $\mathbb{F}_{(2^8)^2}$ is equivalent to performing 5 multiplications over $\mathbb{F}_{2^8}$.

Another disadvantage is constituted by the fact that the public key G' is computed as SG like in the original McEliece (P is supposed to be implicit into the support of the code), and the scramble matrix S occupies a great amount of memory (131,072 bytes, see [19, Table 3]). This is completely redundant, as the reduction to the systematic form is enough to mask the trapdoor and provide one-way security [11].

On the other hand, the length of the encrypted plaintext is about 10 times the length of our plaintext (1288 bits, as opposed to 128 bits); however, we stress again that, in a "real-world" scenario, public-key encryption would only be used for encrypting a small amount of data, for obvious reasons. So if a large number of bits needs to be encrypted, with every probability a PKE would be used to exchange a small key (usually 128 or 256 bits) and then the plaintext would be encrypted with a symmetric encryption scheme.

If we follow this approach in our case, the timings that we obtain strongly support our claim. The latest benchmark speed indicated for AES-128 is about 16 cycles per byte[2]. Hence, if we want to encrypt, for a comparison, a plaintext of length 1288 bits = 161 bytes, it would take only 2,576 clock cycles; even on an embedded device, this number is very small compared to the rest of the encryption process. In total, our encryption process ranges from around 1.5 to 2.7 times faster than [19].

Table 11. Cost of encrypting a plaintext of length 1288 bits

Code	Cost (clock cycles)
Goppa + Kobara-Imai	6,358,952
Code $\mathfrak{B}$	4,174,310
Code $\mathfrak{C}$	2,338,780

A similar argument holds for decryption.

[2] http://www.cryptopp.com/benchmarks.html

Finally, we would like to highlight that we are using Keccak to represent both our hash functions and a random number generator; the flexibility that it provides is evident. Other SHA-3 competitors like the function Blue Midnight Wish (BMW) used in [19] have been proved to be faster [16], but do not reach the same level of security, and for this have been discarded: although, as noted in the announcement of the finalists, "none of these candidates was clearly broken", several attacks have been presented[3].

Further investigation is certainly still required, but for a totally detailed analysis probably even a comparison at source code level would become necessary, and that falls beyond the scope of this paper.

Acknowledgments. We would like to thank Steven Galbraith for many fruitful discussions and his constant support throughout the development of the paper.

References

1. Atmel Corporation, "AVR Studio 5.0", http://www.atmel.com/avrstudio
2. Barreto, P.S.L.M., Cayrel, P.-L., Misoczki, R., Niebuhr, R.: Quasi-Dyadic CFS Signatures. In: Lai, X., Yung, M., Lin, D. (eds.) Inscrypt 2010. LNCS, vol. 6584, pp. 336–349. Springer, Heidelberg (2011)
3. Barreto, P.S.L.M., Misoczki, R., Villas Boas, L.B.: SBCRYPT - Syndrome-Based Cryptography Library
4. Berger, T.P., Cayrel, P.-L., Gaborit, P., Otmani, A.: Reducing Key Length of the McEliece Cryptosystem. In: Preneel, B. (ed.) AFRICACRYPT 2009. LNCS, vol. 5580, pp. 77–97. Springer, Heidelberg (2009)
5. Berger, T.P., Loidreau, P.: How to mask the structure of codes for a cryptographic use. Design, Codes and Cryptography 35, 63–79 (2005)
6. Berlekamp, E.R.: Factoring polynomials over finite fields. Bell System Technical Journal 46, 1853–1859 (1967)
7. Berlekamp, E.R., McEliece, R.J., van Tilborg, H.C.A.: On the inherent intractability of certain coding problems. IEEE Transactions on Information Theory 24, 384–386 (1978)
8. Bernstein, D.J., Lange, T., Peters, C.: Attacking and Defending the McEliece Cryptosystem. In: Buchmann, J., Ding, J. (eds.) PQCrypto 2008. LNCS, vol. 5299, pp. 31–46. Springer, Heidelberg (2008)
9. Bernstein, D.J., Lange, T., Peters, C., van Tilborg, H.C.A.: Explicit bounds for generic decoding algorithms for code-based cryptography. In: Pre-proceedings of WCC 2009, pp. 168–180 (2009)
10. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: The Keccak sponge function family, http://keccak.noekeon.org/
11. Biswas, B., Sendrier, N.: McEliece Cryptosystem Implementation: Theory and Practice. In: Buchmann, J., Ding, J. (eds.) PQCrypto 2008. LNCS, vol. 5299, pp. 47–62. Springer, Heidelberg (2008)
12. Dowsley, R., Müller-Quade, J., Nascimento, A.C.A.: A CCA2 Secure Public Key Encryption Scheme Based on the McEliece Assumptions in the Standard Model. In: Fischlin, M. (ed.) CT-RSA 2009. LNCS, vol. 5473, pp. 240–251. Springer, Heidelberg (2009)

[3] http://ehash.iaik.tugraz.at/wiki/Blue_Midnight_Wish

13. Eisenbarth, T., Güneysu, T., Heyse, S., Paar, C.: MicroEliece: McEliece for Embedded Devices. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 49–64. Springer, Heidelberg (2009)
14. Faugère, J.-C., Otmani, A., Perret, L., Tillich, J.-P.: Algebraic Cryptanalysis of McEliece Variants with Compact Keys. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 279–298. Springer, Heidelberg (2010)
15. Faugère, J.C., Otmani, A., Perret, L., Tillich, J.P.: Algebraic Cryptanalysis of Compact McEliece's Variants - Toward a Complexity Analysis. In: International Conference on Symbolic Computation and Cryptography, SCC 2010, pp. 45–56 (2010)
16. Fleischmann, E., Forler, C., Gorski, M.: Classification of the SHA-3 Candidates, http://drops.dagstuhl.de/volltexte/2009/1948/pdf/09031.ForlerChristian.Paper.1948.pdf
17. Fujisaki, E., Okamoto, T.: Secure Integration of Asymmetric and Symmetric Encryption Schemes. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999)
18. Heyse, S.: Low-Reiter: Niederreiter Encryption Scheme for Embedded Microcontrollers. In: Sendrier, N. (ed.) PQCrypto 2010. LNCS, vol. 6061, pp. 165–181. Springer, Heidelberg (2010)
19. Heyse, S.: Implementation of McEliece Based on Quasi-dyadic Goppa Codes for Embedded Devices. In: Yang, B.-Y. (ed.) PQCrypto 2011. LNCS, vol. 7071, pp. 143–162. Springer, Heidelberg (2011)
20. Kobara, K., Imai, H.: Semantically Secure McEliece Public-Key Cryptosystems-Conversions for McEliece PKC. In: Kim, K.-C. (ed.) PKC 2001. LNCS, vol. 1992, pp. 19–35. Springer, Heidelberg (2001)
21. MacWilliams, F.J., Sloane, N.J.: The theory of error-correcting codes. North Holland, Amsterdam (1977)
22. Mandell Freeman, D., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More Constructions of Lossy and Correlation-Secure Trapdoor Functions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 279–295. Springer, Heidelberg (2010)
23. McEliece, R.J.: A Public-Key System Based on Algebraic Coding Theory. In: DSN Progress Report 44, pp. 114–116. Jet Propulsion Lab (1978)
24. Micciancio, D.: Improving Lattice Based Cryptosystems Using the Hermite Normal Form. In: Silverman, J.H. (ed.) CaLC 2001. LNCS, vol. 2146, pp. 126–145. Springer, Heidelberg (2001)
25. Misoczki, R., Barreto, P.S.L.M.: Compact McEliece Keys from Goppa Codes. In: Jacobson Jr., M.J., Rijmen, V., Safavi-Naini, R. (eds.) SAC 2009. LNCS, vol. 5867, pp. 376–392. Springer, Heidelberg (2009)
26. Niederreiter, H.: A Public-Key Cryptosystem Based on Shift Register Sequences. In: Pichler, F. (ed.) EUROCRYPT 1985. LNCS, vol. 219, pp. 35–39. Springer, Heidelberg (1986)
27. Persichetti, E.: Compact McEliece keys based on Quasi-Dyadic Srivastava codes. IACR Cryptology ePrint Archive, (2011) (preprint)
28. Peters, C.: Information-Set Decoding for Linear Codes over $\mathbb{F}_q$. In: Sendrier, N. (ed.) PQCrypto 2010. LNCS, vol. 6061, pp. 81–94. Springer, Heidelberg (2010)
29. Preneel, B., Bosselaers, A., Govaerts, R., Vandewalle, J.: A software implementation of the McEliece public-key cryptosystem. In: Proceedings of the 13th Symposium on Information Theory in the Benelux, Werkgemeenschap voor Informatieen Communicatietheorie, pp. 119–126. Springer (1992)

30. Prometheus. Implementation of McEliece cryptosystem for 32-bit microprocessors (c-source), http://www.eccpage.com/
31. Schechter, S.: On the inversion of certain matrices. Mathematical Tables and Other Aids to Computation 13(66), 73–77 (1959)
32. Sendrier, N.: Encoding information into constant weight words. In: IEEE Conference, ISIT 2005, pp. 435–438 (September 2005)

A Additional Definitions

We present here some additional definitions needed for the key generation process.

Definition 10. *Given two disjoint sequences* $\bar{v} = (v_1, \ldots, v_\ell) \in \mathbb{F}_q^\ell$ *and* $\bar{L} = (L_1, \ldots, L_n) \in \mathbb{F}_q^n$*, the* Cauchy matrix $C(\bar{v}, \bar{L})$ *is the matrix with components* $C_{ij} = \dfrac{1}{v_i - Lj}$*, i.e.*

$$C(\bar{v}, \bar{L}) = \begin{pmatrix} \dfrac{1}{v_1 - L_1} & \cdots & \dfrac{1}{v_1 - L_n} \\ \vdots & \vdots & \vdots \\ \dfrac{1}{v_\ell - L_1} & \cdots & \dfrac{1}{v_\ell - L_n} \end{pmatrix}.$$

Cauchy matrices have the property that all of their submatrices are invertible [31].

Definition 11. *Fix a finite field* $\mathbb{F}_q$ *and an integer* $m > 1$*. Choose a polynomial* $g(z)$ *in* $\mathbb{F}_{q^m}[z]$ *of degree* $t < n/m$ *and a sequence of distinct elements* $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_{q^m}$ *such that* $g(\alpha_i) \neq 0$ *for all* i*. The polynomial* $g(z)$ *is called the* Goppa polynomial*. The set of words* $\bar{c} = (c_1, \ldots, c_n) \in \mathbb{F}_{q^m}^n$ *with* $\sum_{i=1}^n \frac{c_i}{z - \alpha_i} \equiv 0 \pmod{g(z)}$ *defines an* $[n, n-t]$ *linear code over* $\mathbb{F}_{q^m}$*. The corresponding* Goppa code *is the restriction of this code to* $\mathbb{F}_q$*, i.e. the set of elements* $\bar{c} = (c_1, \ldots, c_n) \in \mathbb{F}_q^n$ *which satisfy the above condition.*

Alternatively (and usually) a Goppa code is defined by means of its parity-check matrix, which is of the form:

$$H = \begin{pmatrix} \dfrac{1}{g(\alpha_1)} & \cdots & \dfrac{1}{g(\alpha_n)} \\ \vdots & \vdots & \vdots \\ \dfrac{\alpha_1^{t-1}}{g(\alpha_1)} & \cdots & \dfrac{\alpha_n^{t-1}}{g(\alpha_n)} \end{pmatrix}$$

It is clear then that a Goppa code has dimension $k \geq n - mt$. The minimum distance is $t + 1$, or $2t + 1$ in the special binary case ($q = 2$).

Goppa codes are a particular instance of Alternant codes, with $x_i = \alpha_i$, $y_i = 1/g(\alpha_i)$.

B Quasi-Dyadic Key Generation

Misoczki and Barreto in [25] first introduced a scheme based on quasi-dyadic Goppa codes, making use of codes simultaneously in dyadic [25, Th. 2] and Cauchy form [21, Ch. 12, Pr. 5]. Necessary conditions are that the generator polynomial has to be monic and without multiple zeros, and that the code needs to be defined over a field of characteristic 2, with a dyadic signature satisfying

$$\frac{1}{h_{i\oplus j}} = \frac{1}{h_i} + \frac{1}{h_j} + \frac{1}{h_0}. \tag{1}$$

The scheme was subsequently extended and generalized to the case of GS codes [27], with multiple benefits including security improvements (described in the next section). Since it can be easily proved that every generalized Srivastava code with $t = 1$ is a Goppa code, the two cases are in fact just two instances of the same scheme. For the construction, we follow the steps presented in [27, Section 4].

Equation (1) is the core of the key generation algorithm. The procedure takes input parameters n, s, t such that $n = n_0 s$, $mst < n$ for s a power of 2 and a finite field $\mathbb{F}_{q^m} = \mathbb{F}_{2^u}$ where $q = 2^\lambda$, $u = m\lambda$, then assigns distinct values at random to the elements h_{2^j} for $j = 1, \ldots, \log_2(n-1)$, in the meantime fixing the elements between h_{2^j} and $h_{2^{j+1}}$ by using (1).

An initial block in dyadic form is formed from the signature $\bar{h}$ just built; this is equivalent to a Goppa code. In case $t > 1$, the other blocks are computed by successive powering, up to the power of t. The parity-check matrix eventually obtained is projected onto the base field and finally, we retain the non-trivial part of its systematic form to be used as trapdoor.

We refer to [27] for a fully detailed description of the construction process.

C Resistance to Structural Attacks

The main threat against quasi-dyadic schemes is represented by the so-called FOPT attack [14]. It relies on the fundamental property $H \cdot G^T = 0$ to build an algebraic system, using then Gröbner bases techniques to solve it. The special properties of codes in quasi-dyadic form are of key importance, as they contribute to considerably reduce the number of unknowns of the system. Also, the parameters m and t come into account as they define the dimension of the solution space.

The aim is to find a valid parity-check matrix for the code, that is, a matrix H in Alternant form, $H = \{y_j x_j^i\}$; these elements are represented by two sets of unknowns $\{X_i\}$ and $\{Y_i\}$. The first step of the attack is then generating the following system of equations:

$$\left\{ g_{i,0} Y_0 X_0^j + \cdots + g_{i,n-1} Y_{n-1} X_{n-1}^j = 0 \mid i = 0, \ldots, k-1, j = 0, \ldots, \ell - 1 \right\}. \tag{2}$$

As is easy to see, the case $j = 0$ produces a set of linear equations involving only the Y_i. These can be further reduced with the help of some properties derived from the dyadicity and the key-generation algorithm [14, Pr. 5]; in particular, we have that $Y_{is+j} = Y_{is}$ for each block, i.e. $i = 0, \ldots, n_0 - 1$, $j = 1, \ldots s$ (a proof is given for the case $t = 1$; for the adaptation to the case $t > 1$ see [27]). This results in having only $n_0 - 1$ unknowns Y_i, since we can arbitrarily choose one of them. Moreover, the linear equations are identical for all the rows of each dyadic block, hence only $n_0 - mt$ distinct equations remain after eliminating the redundant ones.

As in any linear system, the difference between these two numbers gives the number of *free variables* of the system: in this case, $mt - 1$. If it is possible to recover the free variables (if the number of those is very small, even just by guessing) it is possible to reduce (2) to a simplified system involving only the X_i. Once the reduction is done, a linearization trick is used to solve and retrieve the remaining unknowns.

Hence, it is crucial to keep the dimension of the solution space (number of free variables) high enough to prevent the attack to succeed; the authors in [15] indicate that this number should be not smaller than 20. In this case in fact, the computational effort required to solve the system is too high: experimental results indicate a complexity of approximately 2^{128} bit operations.

Additional security comes from another phenomenon that occurs when the base field is $\mathbb{F}_2$. In this case the Gröbner basis necessary to solve the system is easy to compute, but somehow "trivial" (reduced to one equation) and doesn't provide enough information, hence the attack cannot be completed.

Solving Underdetermined Systems of Multivariate Quadratic Equations Revisited

Enrico Thomae and Christopher Wolf

Horst Görtz Institute for IT-security
Faculty of Mathematics
Ruhr-University of Bochum, 44780 Bochum, Germany
enrico.thomae@rub.de, chris@Christopher-Wolf.de

Abstract. Solving systems of m $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic ($\mathcal{MQ}$) equations in n variables is one of the main challenges of algebraic cryptanalysis. Although the associated $\mathcal{MQ}$-problem is proven to be NP-complete, we know that it is solvable in *polynomial time* over fields of even characteristic if either $m \geq n(n-1)/2$ (*overdetermined*) or $n \geq m(m+1)$ (*underdetermined*). It is widely believed that $m = n$ has worst case complexity. Actually in the overdetermined case Gröbner Bases algorithms show a gradual decrease in complexity from $m = n$ to $m \geq n(n-1)/2$ as more and more equations are available. For the underdetermined case no similar behavior was known. Up to now the best way to deal with the case $m < n < m(m+1)$ was to randomly guess variables until $m = n$. This article shows how to smartly use additional variables and thus obtain a gradual change of complexity over even characteristics also for the underdetermined case. Namely, we show how a linear change of variables can be used to reduce the overall complexity of solving a $\mathcal{MQ}$-system with m equations and $n = \omega m$ variables for some $\omega \in \mathbb{Q}_{>1}$ to the complexity of solving a $\mathcal{MQ}$-system with only $(m - \lfloor \omega \rfloor + 1)$ equations and variables, respectively. Our algorithm can be seen as an extension of the previously known algorithm from Kipnis-Patarin-Goubin (extended version of Eurocrypt '99) and improves an algorithm of Courtois *et al.* which eliminates $\lfloor \log_2 \omega \rfloor$ variables. For small ω we also adapt our algorithm to fields of odd characteristic. We apply our result to break current instances of the Unbalanced Oil and Vinegar public key signature scheme that uses $n = 3m$ and hence $\omega = 3$.

Keywords: Underdetermined Multivariate Equations, UOV Signature Scheme.

1 Introduction

It is well known that algebraic equations can be an Achilles' heel for cryptographic systems. Whether stream ciphers [5, 13], hash functions [19] or block ciphers [16], they all can be expressed through a system of equations over a finite field $\mathbb{F}$ with a solution that yields the private key. For asymmetric schemes the importance is even more obvious. For example variants of McEliece [12] or

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 156–171, 2012.

$\mathcal{M}$ultivariate $\mathcal{Q}$uadratic ($\mathcal{MQ}$) schemes such as Hidden Field Equations [11] were broken using algebraic techniques. So it is fair to say that solving systems of $\mathcal{MQ}$ equations is one of the main challenges of algebraic cryptanalysis. However, as the underlying $\mathcal{MQ}$-problem is proven to be NP-complete [14], we cannot hope to find an efficient algorithm for all instances. In particular, if the number of equations m equals the number of unknowns n, all known empirical algorithms are exponential on random instances of the $\mathcal{MQ}$-problem. Nevertheless we know that the problem becomes easy for fields of characteristic 2 if either $m \geq n(n-1)/2$ or $n \geq m(m+1)$. In the first case, we replace each monomial by a new variable and solve a linear system in $n(n-1)/2$ equations and variables. The second case is covered by an algorithm of Kipnis-Patarin-Goubin [15, Sec. 7] and will be further explored in this article.

Until now, research mainly covered the *overdetermined* case $m \geq n$. There are many algorithms like F_4, F_5 and XL that benefit of additional equations [8, 9, 10]. So for $m = n$ even guessing one or two variables can help to reduce the complexity dramatically [2]—and thus make a big difference in practice. In contrast none of the algorithms benefits in the same way of the *underdetermined* case $n > m$ (cf. Section 1.1). In particular, their complexity is exponentially linked to the number of variables. Hence, having more variables will dramatically increase their running time (and also space requirements). As finding *one* solution often suffices for cryptographic purpose, the best way of "using" more variables today, is to fix them to random values and thus receive a hard instance with $n = m$ and one solution on average. This is not very sophisticated and in a sense similar to throw away additional equations in the overdetermined case and only work with the remaining ones. This article shows how to us additional variables and hence closes the complexity gap between $n = m$ and $n \geq m(m+1)$. Our main result applies to fields of even characteristic. In section 6 we discuss a generalization to arbitrary characteristics.

1.1 Related Work

The best treatment of the overdetermined case $m \geq n$ is covered by XL or Gröbner bases algorithms like F_4 or its successor F_5. The overall complexity is well understood [1] and becomes gradually easier if more and more equations are available. In particular for $m \geq n(n-1)/2$ over $\mathbb{F}_{2^k}$ and $m \geq n(n+1)/2$ over $\mathbb{F}_{p^k}$ for p an odd prime, the overall problem can be solved in polynomial time by *Linearization*. For the underdetermined case not much is known. Basically, all research so far has centered around two cases: $n = m$ and $n \geq m(m+1)$. The first has *exponential*, the latter *polynomial* time complexity. In particular, an algorithm from Kipnis-Patarin-Goubin [15, Sec. 7] can efficiently solve the latter case in $\mathbb{F}_{2^k}$. Courtois *et al.* [6] extended this result to arbitrary fields $\mathbb{F}_{p^k}$ and showed that the problem becomes polynomial as soon as $n \geq 2^{\frac{m}{7}}(m+1)$. Furthermore they showed how to eliminate $\lfloor \log_2 \omega \rfloor$ variables and thus get a system of $m - \lfloor \log_2 \omega \rfloor$ variables and equations (cf. Prop. 1 in [6]). We extend this result, using a tight analysis of the technique of Kipnis-Patarin-Goubin, to obtain a system of $m - \lfloor \omega \rfloor + 1$ variables and equations.

1.2 Achievement and Organization

We close an important gap in understanding the underdetermined case especially for $\mathbb{F}_{2^k}$. In particular we show that there is a gradual change from *exponential* running time to *polynomial* running time if n gets larger than m. This improves the cryptanalysis of the Unbalanced Oil and Vinegar Signature scheme (UOV) [17] and therefore forces a change of parameter sets (cf. section 5).

The organization of the paper is as follows. Section 2 gives some notation. Section 3 shows how to describe the transformation of variables we are using, shortly repeats the algorithm of Kipnis-Patarin-Goubin and introduce our new algorithm. Section 4 is the most important, as it gives a theoretical analysis of the correctness of our algorithm and also the one of Kipnis-Patarin-Goubin. Section 5 gives a complexity analysis and shows that parameters of UOV have to be increased. In section 6 we adapt our algorithm to the general case $\mathbb{F}_{p^k}$ for small ω and motivate future research on this question.

2 Notation

A $\mathcal{MQ}$-system of equations over a finite field $\mathbb{F}_q$ with q elements is given by m equations $p^{(k)} = 0$ for polynomial functions $p^{(k)} : \mathbb{F}_q^n \to \mathbb{F}_q$ for $1 \leq k \leq m$ and $\gamma_{ij}^{(k)}, \beta_i^{(k)}, \alpha^{(k)} \in \mathbb{F}_q$ according to

$$p^{(k)}(x_1, \ldots, x_n) := \sum_{1 \leq i \leq j \leq n} \gamma_{ij}^{(k)} x_i x_j + \sum_{1 \leq i \leq n} \beta_i^{(k)} x_i + \alpha^{(k)} . \tag{1}$$

If we speak of *solving* such an $\mathcal{MQ}$-system, we always mean finding *one* solution. For cryptanalytic purposes, this is actually sufficient in most cases. We call $p^{(k)}$ as defined by (1) *inhomogeneous*. The *homogeneous* case consists only of terms in $x_i x_j$ for $1 \leq i \leq j \leq n$ and is thus defined by

$$p^{(k)}(x_1, \ldots, x_n) := \sum_{1 \leq i \leq j \leq n} \gamma_{ij}^{(k)} x_i x_j .$$

The corresponding $\mathcal{MQ}$-map $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ is defined by $\mathcal{P} := \left(p^{(1)}, \ldots, p^{(m)}\right)^\intercal$. To ease notation, we restrict to homogeneous systems in the sequel. Note that our algorithm also works for inhomogeneous systems without introducing a homogenization variable.

Let $\pi^{(k)}$ be the coefficient vector of $p^{(k)}(x_1, \ldots, x_n)$ in lexicographic order, *i.e.*

$$\pi^{(k)} = (\gamma_{11}^{(k)}, \gamma_{12}^{(k)}, \ldots, \gamma_{1n}^{(k)}, \gamma_{22}^{(k)}, \gamma_{23}^{(k)}, \ldots, \gamma_{nn}^{(k)}).$$

Note that our algorithm also works with other monomial orderings. However, for the ease of explanation, we have fixed lexicographic ordering throughout this paper. The corresponding coefficient matrix Π is defined by $\Pi := \left(\pi^{(1)}, \ldots, \pi^{(m)}\right)^\intercal$.

3 Transformation of Variables

Let $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ be an $\mathcal{MQ}$-map with m equations and $n = \omega m$ variables $x_1, \ldots, x_n$ for some $\omega \in \mathbb{Q}_{>1}$. To make parts of the arguments easier, we will sometimes change to the notation $n = m+v$ with $v = (\omega-1)m$. The current way to find a solution of this system is to fix v variables at random [2, 3, 7] and solve the remaining system of m equations and m variables using a $\mathcal{MQ}$-solver such as F_5 or XL. Kipnis-Patarin-Goubin [15, Sec. 7] were the first who took benefit of the additional v variables and showed that the system is solvable in *polynomial time* for $n \geq m(m+1)$. In a nutshell they applied a linear transformation $S \in \mathrm{GL}_n(\mathbb{F}_q)$ of variables to obtain a new $\mathcal{MQ}$-system $\mathcal{F} := \mathcal{P} \circ S$ with coefficient matrix Φ. The transformation matrix S is calculated in polynomial time such that fixing v variables in $\mathcal{F}$ provides a linear system in the remaining m variables for fields of characteristic 2. We will investigate this approach in more detail in section 3.2.

To understand how S operates on the coefficients of $\mathcal{P}$ and $\mathcal{F}$, we introduce the transformation Σ such that $\Sigma\Pi^\intercal = \Phi^\intercal$. This transformation was previously used to determine short key variants of UOV [18].

3.1 How to Determine Σ

We can write every equation $p^{(i)}$ of $\mathcal{P}$ as a quadratic form $p^{(i)} = x^\intercal \mathfrak{P}^{(i)} x$ for $x = (x_1, \ldots, x_n)$ and a matrix $\mathfrak{P}^{(i)} \in \mathbb{F}_q^{n \times n}$ consisting of the coefficients of $p^{(i)}$. Note that this matrix is *not* symmetric if $\mathbb{F}$ is of characteristic 2. Applying the change of variables, *i.e.* $y = S^{-1}x$, we obtain a new $\mathcal{MQ}$-system $\mathcal{F}$ with $f^{(i)} = y^\intercal S^\intercal \mathfrak{P}^{(i)} S y$ for $y = (y_1, \ldots, y_n)$. The coefficients of this new map are determined by $S^\intercal \mathfrak{P}^{(i)} S =: \mathfrak{F}^{(i)}$. Or in other words $f^{(k)}(y_1, \ldots, y_n) := \sum_{1 \leq i \leq j \leq n} \widetilde{\gamma}_{ij}^{(k)} y_i y_j$ and $x_i = \sum_{p=1}^{n} s_{ip} y_p$. Comparison of coefficients in the following equation reveals an explicit formula for Σ.

$$\begin{aligned} \sum_{1 \leq i \leq j \leq n} \widetilde{\gamma}_{ij}^{(k)} y_i y_j &= \sum_{1 \leq i \leq j \leq n} \gamma_{ij}^{(k)} x_i x_j \\ &= \sum_{1 \leq i \leq j \leq n} \gamma_{ij}^{(k)} \Big(\sum_{p=1}^{n} s_{ip} y_p\Big)\Big(\sum_{p=1}^{n} s_{jp} y_p\Big) \end{aligned}$$

Let $s_{i\cdot} \in \mathbb{F}_q^n$ be the i-th row of S and $D^{ij} := s_{i\cdot}^\intercal s_{j\cdot}$ the dyadic product of the i-th and j-th row of S. Now we can express $x_i x_j$ by

$$x_i x_j = \Big(\sum_{p=1}^{n} s_{ip} y_p\Big)\Big(\sum_{p=1}^{n} s_{jp} y_p\Big) = \sum_{1 \leq l \leq n} D_{ll}^{ij} y_l^2 + \sum_{1 \leq l < p \leq n} (D_{lp}^{ij} + D_{pl}^{ij}) y_l y_p.$$

Let $I := ((a_i, b_i) \mid 1 \leq a_i \leq b_i \leq n)$ be the ordered index set of all quadratic monomials. We have chosen lexicographic order of the monomials, *i.e.*

$$I = ((1,1), (1,2), \ldots, (1,n), (2,2), (2,3), \ldots, (n,n)).$$

For (a_i, b_i), $(a_j, b_j) \in I$ we obtain $\Sigma := (\sigma_{ij})$ with

$$\sigma_{ij} := \begin{cases} s_{a_j a_i} \cdot s_{b_j b_i} & \text{for } a_i = b_i, \\ s_{a_j a_i} \cdot s_{b_j b_i} + s_{a_j b_i} \cdot s_{b_j a_i} & \text{for } a_i \neq b_i, \end{cases} \tag{2}$$

by collecting the appropriate entries of all the dyadic products. The matrix Σ, obtained by comparing coefficients, maps the coefficients of $p^{(k)}$ to the coefficients of $f^{(k)}$. Denoting $\tau := |I| = \frac{n(n+1)}{2}$ (for $q > 2$) and $\tau := |I| = \frac{n(n-1)}{2}$ (for $q = 2$) the number of monomials, this leads to

$$\Sigma \cdot \begin{pmatrix} \gamma^{(k)}_{a_1,b_1} \\ \vdots \\ \gamma^{(k)}_{a_\tau,b_\tau} \end{pmatrix} = \begin{pmatrix} \widetilde{\gamma}^{(k)}_{a_1,b_1} \\ \vdots \\ \widetilde{\gamma}^{(k)}_{a_r,b_r} \end{pmatrix}. \tag{3}$$

3.2 Algorithm of Kipnis-Patarin-Goubin

We briefly recap the algorithm of Kipnis-Patarin-Goubin for $n \geq m(m+1)$, cf. [15, Sec. 7] for details. In section 3.3 we will generalize this technique to $n \leq m(m+1)$ and show that we can force enough elements of Φ, *i.e.* coefficients of $\mathcal{F}$, to be zero, such that we obtain $(\lfloor \omega \rfloor - 1)$ linear equations. The first idea is to split the variables $y_1, \ldots, y_n$ into two sets $V := \{y_{m+1}, \ldots, y_n\}$ and $O := \{y_1, \ldots, y_m\}$. Here V denotes the set of variables we want to fix and O the set of variables we want to determine. Due to the strong connection to the Oil and Vinegar Signature Scheme, we call V the *vinegar variables* and O the *oil variables*. The aim of Kipnis-Patarin-Goubin was to find S such that most coefficients of $\mathcal{F}$ are zero and thus the new $\mathcal{MQ}$-system is easily solvable *e.g.* by Linearization. The overall idea to find such a linear transformation S efficiently is the following. First all equations of (3) are quadratic in s_{ij}. But if we fix certain elements of S at random, some of the equations become linear. Solving this linear equations enable us to fix some coefficients of $\mathcal{F}$ to zero. More precisely Kipnis-Patarin-Goubin aimed at solving the quadratic equations in s_{ij} of (3) we obtain by setting

$$\widetilde{\gamma}^{(k)}_{i,j} = 0 \text{ for } 1 \leq i, j, k \leq m, i \neq j\,. \tag{4}$$

To ease notation we label (4) by (i, j, k) or just (i, j) if we want to denote all equations $(i, j, 1)$ to (i, j, m). As all these equations are quadratic, Kipnis-Patarin-Goubin fixed the first column of S to random values. Note that regarding to (2) all monomials in equation (i, j) consists of one variable of the i-th and one variable of the j-th column of S. This means $\widetilde{\gamma}^{(k)}_{1,1}$ is fixed to a random value and equations $(1, 2)$ to $(1, n)$ become linear. $(1, 2)$ gives us m linear equations in the s_{i2} and after randomly fixing the superfluous variables, we can determine them such that $\widetilde{\gamma}^{(k)}_{1,2} = 0$ for $1 \leq k \leq m$. Now that the second column of S is determined, we obtain additional linear equations $(2, 3)$ to $(2, n)$. Using the $2m$ linear equations of $(1, 3)$ and $(2, 3)$, we can determine s_{i3}. If the first k columns of S are determined, we solve the km linear equations $(1, k + 1)$ to $(k, k + 1)$ to

determine the $(k+1)$-th column of S. The algorithm continues until columns 1 to m of S are determined. At each level, more and more of the equations become linear. For the last step we have to solve the linear equations $(1, m)$ to $(m-1, m)$ in the unknowns s_{1m} to s_{nm} and thus $n \geq m(m-1)$ must hold.

After this transformation we obtain m equations $1 \leq j \leq m$ of the form

$$\sum_{i=1}^{m} \beta_{i,j} y_i^2 + y_1 L_{1,j}(y_{m+1}, \ldots, y_{m+v}) + \ldots \\ + y_m L_{m,j}(y_{m+1}, \ldots, y_{m+v}) + Q_j(y_{m+1}, \ldots, y_{m+v}) \,. \tag{5}$$

The terms $L_{i,j}$ denote some linear functions in the V variables we want to fix and Q_j denotes some quadratic function in these variables. Now Kipnis-Patarin-Goubin determined $y_{m+1}, \ldots, y_{m+v}$ by Gaussian Elimination such that $L_{i,j} = 0$ for all $1 \leq i, j \leq m$. This is possible for $v \geq m^2$ and thus we obtain the condition $n \geq m(m+1)$. For fields of characteristic 2 the remaining system in (5) is linear in the O variables and can thus be easily solved. This is due to the Frobenius Homomorphism $x \mapsto x^2$ which effectively allows us to treat monomials of the form y_i^2 as linear variables.

In the next section we provide a tight analysis for $n \leq m(m+1)$ and show that solving a $\mathcal{MQ}$-system $\mathcal{P}$ with m equations and $n = \omega m$ variables is roughly as hard as solving a $\mathcal{MQ}$-system of $(m - \lfloor \omega \rfloor + 1)$ equations in $(m - \lfloor \omega \rfloor + 1)$ variables.

3.3 Tight Analysis for $n = \omega m$ and Improvement

To obtain linear equations we also fix the first column of S at random. This step is similar to Kipnis-Patarin-Goubin. As we are in the case $m < n < m(m+1)$ we cannot fulfill all equations (4) and have to adjust our strategy accordingly. In particular, we have to reduce the number of equations during the intermediate steps, *i.e.* due to a lack of variables in S we can only solve equations (i, j, k) for some fixed bound b_j and $1 \leq i < j$, $1 \leq k \leq b_j$.

The overall process is depicted in figure 1. Remember, lines of $\Phi^\intercal$ denote coefficients and columns denote polynomials $f^{(1)}$ to $f^{(m)}$. The dotted areas are of no interest for us, as all the corresponding monomials in $\mathcal{F}$ vanish after fixing the variables in V. Hence, setting them to a specific value is no use. The gray part are arbitrary values. To make the interesting part of $\Phi^\intercal$ clearer, we reordered the rows in the right block of figure 1, what is indicated by the ordered pairs labeling the rows.

Let $B_j := \{(i, j, k) \,|\, 1 \leq k \leq b_j, 1 \leq i < j\}$ be the j-th block of coefficients for which we want to gain $\widetilde{\gamma}_{i,j}^{(k)} = 0$ in Φ where b_j is some bound to be determined later (cf. section 4) and $\mathcal{V}(B_j) := (j-1)b_j$ the volume of such a block or in other words the number of zero coefficients. As depicted in figure 1 we are able to eliminate all $O \times O$ coefficients with $i \neq j$ in the first b_m columns of Φ. Solving the linear system in the $\widetilde{\gamma}_{i,i}^{(k)} y_i^2$ for $i \in \{1, \ldots, m\}$ allows us to replace a total

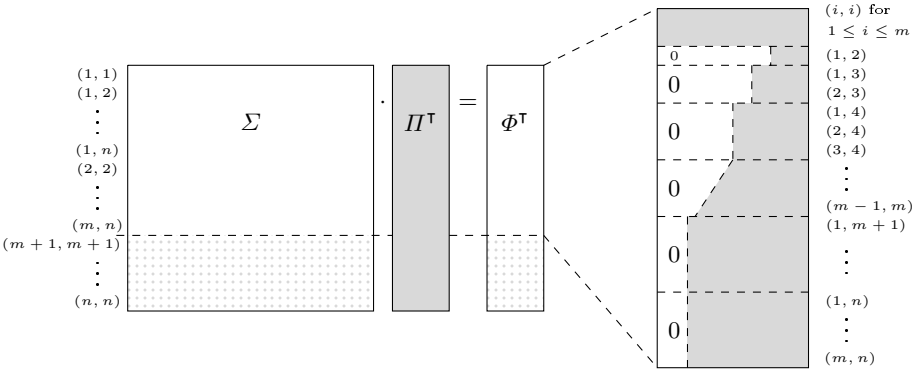


Fig. 1. Overview of coefficients in Φ^{T} we want to fix to zero

of b_m variables in the remaining equations (see Section 4.2). This leads effectively to a new system of $(m - b_m)$ equations and variables. The crucial point is to determine the correct value of b_m. One might be inclined to choose $b_j \leq m$ maximal such that $\mathcal{V}(B_j)$ is less than the number of variables n in the j-th column of S in order to produce as much zeros in Φ as possible. However note that $\mathcal{V}(B_j)$ cannot be equal to n as we need one more variable than equations in each block, as the system is homogeneous. Hence, by having as many variables as linear equations, we only would obtain the all-zero vector. In section 4 we will show that the naive approach will not work in general, as if b_j is to large the obtained solution S will not be regular. This question did not come up in the Kipnis-Patarin-Goubin approach, as they fixed enough variables at random to trivially assure regularity of S.

Algorithm 1. High-level description of our algorithm.

Fix the last $n - m$ columns of matrix S to linearly independent vectors.
for $i = 1 \rightarrow m$ **do**
  $A := \emptyset$
  for $j = 1 \rightarrow i - 1$ **do**
    Append linear equations $(i, j, 1)$ to (i, j, ω) to A.
  end for
  Solve A and include its solution into S.
end for
Apply linear transformation S to $\mathcal{P}$.
$A := \emptyset$
for $i = 1 \rightarrow \omega$ **do**
  Append linear equations $L_{1,i}$ to $L_{m,i}$ to A.
end for
Solve A and derive corresponding vinegar variables.
Substitute ω linear equations in remaining $\mathcal{MQ}$-polynomials.

4 Equivalent Solutions S and Their Impact on b_j

Up to this point our approach is a straightforward enhancement of Kipnis-Patarin-Goubin idea. This section covers the main idea of our approach and gives new insights to the theory of solving underdetermined systems of equations. Kipnis-Patarin-Goubin claimed that all the linear equations provide at least one solution in general and that S is regular with high probability. Due to the large n this is actually true for their approach. But as we want to use as many s_{ij} as possible to fix elements in Φ to zero, it is not clear at all, how many equations $\widetilde{\gamma}_{i,j}^{(k)} = 0$ we are able to solve in order to obtain a regular solution S. We use the theory of equivalent keys [20, 21] for the Unbalanced Oil and Vinegar Scheme as a toolkit to show that for every solution S there is an equivalent solution S' with a special structure. The number of variables in S' will upper bound the number of equations $\widetilde{\gamma}_{i,j}^{(k)} = 0$ that yield a regular solution.

4.1 Equivalent Solutions

First let us determine b_m and thus the number of zeros in Φ for $n \geq m(m+1)$, *i.e.* for the original algorithm of Kipnis-Patarin-Goubin.

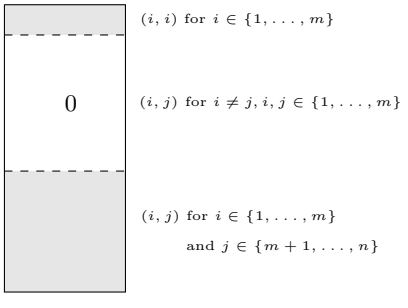


Fig. 2. Upper part of $\Phi^{\intercal}$ for $n \geq m(m+1)$

Denote $\mathbb{S}$ the subset of equations of $\Sigma\Pi^{\intercal} = \Phi^{\intercal}$ labeled by (i,j) for $i,j \in \{1,\ldots,m\}$ and $i \neq j$, *i.e.* the zero part in figure 2. Let S be a solution to $\mathbb{S}$. We call S' an *equivalent solution*, if it preserves the structure of Φ, *i.e.* if S' also fulfills all equations of $\mathbb{S}$. Every element of the equivalence class of such solutions solves our problem. To determine an upper bound on b_j we search for a small (linear) family of matrices $\widetilde{S}$ such that every equivalence class has a representative in this family. We call this a minimal representative. Or loosely speaking these are solutions with large fixed parts for arbitrary Π or a matrix $\widetilde{S}$ with minimal number of variables.

Obviously all equations in $\mathbb{S}$ remain zero if we map every variable $\{y_1,\ldots,y_m\}$ to itself or some permutation and any variable within $\{y_{m+1},\ldots,y_n\}$ to sums of these variables. The only two things we are not allowed to do is mapping

variables of V to variables of O as this would lead to quadratic terms in the O variables and mapping O variables to a sum of O variables, as this would also lead to quadratic terms in the O variables due to $\widetilde{\gamma}_{i,i}^{(k)} \neq 0$ for $i \in \{1, \ldots, m\}$. So if S is a solution to $\mathbb{S}$ then $S\Omega^{-1}$ with

$$\Omega := \begin{pmatrix} \Omega_{(m\times m)}^{(1)} & \Omega_{(m\times v)}^{(2)} \\ 0 & \Omega_{(v\times v)}^{(3)} \end{pmatrix}$$

for $\Omega_{(m\times m)}^{(1)}$ some regular diagonal matrix and $\Omega_{(v\times v)}^{(3)}$ some regular matrix is also a solution as $x = S\Omega^{-1}\Omega y$ holds and Ωy preserves $\widetilde{\gamma}_{i,j}^{(k)} = 0$ for $i \neq j, 1 \leq i, j \leq m$. Note that Ω^{-1} has the same form as Ω, *i.e.*

$$\Omega^{-1} = \begin{pmatrix} {\Omega_{(m\times m)}^{(1)}}^{-1} & \widetilde{\Omega}_{(m\times v)}^{(2)} \\ 0 & {\Omega_{(v\times v)}^{(3)}}^{-1} \end{pmatrix} \text{ with } \widetilde{\Omega}^{(2)} := -{\Omega^{(1)}}^{-1}\Omega^{(2)}{\Omega^{(3)}}^{-1}.$$

Thus we are able to choose Ω^{-1} such that

$$S\Omega^{-1} = \begin{pmatrix} S_{(m\times m)}^{(1)} & S_{(m\times v)}^{(2)} \\ S_{(v\times m)}^{(3)} & S_{(m\times m)}^{(4)} \end{pmatrix} \Omega^{-1} = \begin{pmatrix} \widetilde{S}_{(m\times m)}^{(1)} & 0 \\ \widetilde{S}_{(v\times m)}^{(3)} & I \end{pmatrix} \tag{6}$$

under the condition that $S^{(1)}\widetilde{\Omega}^{(2)} + S^{(2)}{\Omega^{(3)}}^{-1} = I$ and $S^{(3)}\widetilde{\Omega}^{(2)} + S^{(4)}{\Omega^{(3)}}^{-1} = 0$. Note that this is always the case because S is regular and thus $S^{(1)}||S^{(2)}$ has full rank. As ${\Omega^{(1)}}^{-1}$ is just a diagonal matrix, we are only able to fix the first row in $\widetilde{S}^{(1)}$ and thus the remaining number of free variables per column is $d_i = n - 1$ for $1 \leq i \leq m$ and $d_i = 0$ for $m + 1 \leq i \leq n$.

Corollary 1. *For $n \geq m(m+1)$ the Kipnis-Patarin-Goubin approach is upper bounded by $\mathcal{V}(B_i) = n - 1$ for $1 \leq i \leq m$ and $\mathcal{V}(B_i) = 0$ for $m + 1 \leq i \leq n$. This leads to $b_i \leq (n-2)/(i-1)$ for $1 \leq i \leq m$ and $b_i = 0$ for $m + 1 \leq i \leq n$.*

As $\mathcal{V}(B_1) \leq \ldots \leq \mathcal{V}(B_m) = m(m-1) < m(m+1) - 1 \leq n - 1$ hold, Kipnis-Patarin-Goubin were right in assuming that their system of linear equations is solvable. But as $\mathcal{V}(B_i) = 0$ for $m + 1 \leq i \leq n$ they indeed have to use (5) to eliminate the $O \times V$ coefficients.

Let us now come back to our case of figure 1. Our approach eliminates as many $O \times O$ coefficients $\widetilde{\gamma}_{i,j}^{(k)}$ with $i \neq j$ as possible using a linear transformation S of variables. After applying (5) we obtain equations that are linear in fields of characteristic two and thus can be used to substitute variables in the remaining equations. This introduces new $O \times O$ coefficients in the remaining equations and thus we skip eliminating them beforehand (see figure 3).

In order to preserve the structure of Φ, the only transformation of variables Ω applicable is mapping O variables to itself or some permutation and V variables to sums of V variables. In contrast to corollary 1 we are not allowed to map O

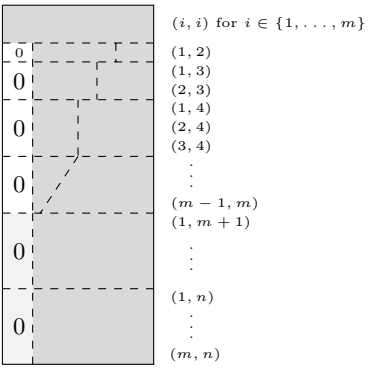


Fig. 3. Upper part of $\Phi^\intercal$ fixing only significant coefficients

variables to V variables as this would introduce new $O \times V$ monomials due to $\widetilde{\gamma}_{i,i}^{(k)} \neq 0$. Thus Ω^{-1} is of form

$$\Omega^{-1} = \begin{pmatrix} {\Omega_{(m\times m)}^{(1)}}^{-1} & 0 \\ 0 & {\Omega_{(v\times v)}^{(3)}}^{-1} \end{pmatrix},$$

with ${\Omega_{(m\times m)}^{(1)}}^{-1}$ a diagonal matrix. This leads to a minimal representative of every equivalence class of the form

$$S\Omega^{-1} = \begin{pmatrix} \widetilde{S}_{(m\times m)}^{(1)} & \widetilde{S}_{(m\times v)}^{(2)} \\ \widetilde{S}_{(v\times m)}^{(3)} & I \end{pmatrix}, \tag{7}$$

where the first row of $\widetilde{S}_{(m\times m)}^{(1)}$ is fixed.

Corollary 2. *For $n = \omega m$ our approach pictured in figure 3 is upper bounded by $\mathcal{V}(B_i) = n - 1$ for $1 \leq i \leq m$ and $\mathcal{V}(B_i) = m$ for $m + 1 \leq i \leq n$. This leads to $b_i = (n-1)/(i-1)$ for $1 \leq i \leq m$ and $b_i = 1$ for $m + 1 \leq i \leq n$.*

Claim. We claim that corollary 1 as well as corollary 2 also yields a lower bound and thus are sharp. The crucial question is, if the $S\Omega^{-1}$ given by (6) or (7) is minimal for all columns i with $\mathcal{V}(B_i) = d_i$, *i.e.* we cannot find a representative $\widetilde{S}$ with less free variables in those columns we are actually using all the free variables. This is the case, if these columns are uniquely defined for some generic Π. For fixed columns 1 to $(i-1)$ this is obviously the case. Although intuitively clear, a rigorous mathematical proof seems to require stronger tools. Nevertheless, experiments prove that there are no systematic dependencies and thus corollary 1 as well as corollary 2 are tight (cf. appendix 5).

The volume of the zero blocks $B_{m+1}, \ldots, B_n$ in figure 3 is

$$mb_m = m\frac{n-1}{m-1} > n-1 > m$$

i.e. larger than the number of independent variables for $m+1 \leq i \leq n$. Thus the solution S would be singular. In order to eliminate the $O \times V$ coefficients, we also have to calculate $y_{m+1}, \ldots, y_{m+v}$ by Gaussian Elimination such that $L_{i,j} = 0$ for all $1 \leq i, j \leq m$ (see equation 5). Corollary 2 leads to $b_m = (n-1)/(m-1) > \lfloor \omega \rfloor$ for $\omega \geq 2$, *i.e.* we are able to reduce the $\mathcal{MQ}$-system to $(m - \lfloor \omega \rfloor)$ equations and variables in this case.

But in order for the $b_m m$ equations $L_{i,j} = 0$ in $(\omega - 1)m$ variables to be solvable, we have to choose $b_m = \lfloor \omega \rfloor - 1$ and thus we are only able to reduce the $\mathcal{MQ}$-system to $(m - \lfloor \omega \rfloor + 1)$ equations and variables, respectively. For $\lfloor \omega \rfloor \mid m$ our algorithm in the next section will merge both strategies, *i.e.* first eliminating some $O \times V$ coefficients and then use equation (5). This allows us to reduce to an $\mathcal{MQ}$-system with $(m - \lfloor \omega \rfloor)$ equations and variables.

4.2 Our Algorithm in Its Most General Form

For a very tight analysis, which give a further improvement if $\lfloor \omega \rfloor \mid m$, we first use algorithm 1 to eliminate the $O \times O$ coefficients $\widetilde{\gamma}_{i,j}^{(k)}$ with $i \neq j$ in the first $\lfloor \omega \rfloor$ equations. Corollary 2 ensures that this is possible. Next we eliminate the $O \times V$ coefficients $\widetilde{\gamma}_{i,j}^{(k)}$ for $m+1 \leq j \leq n$, $1 \leq k \leq \lfloor \omega \rfloor$ and $1 \leq i \leq (m/\lfloor \omega \rfloor)$. See figure 4 for illustration.

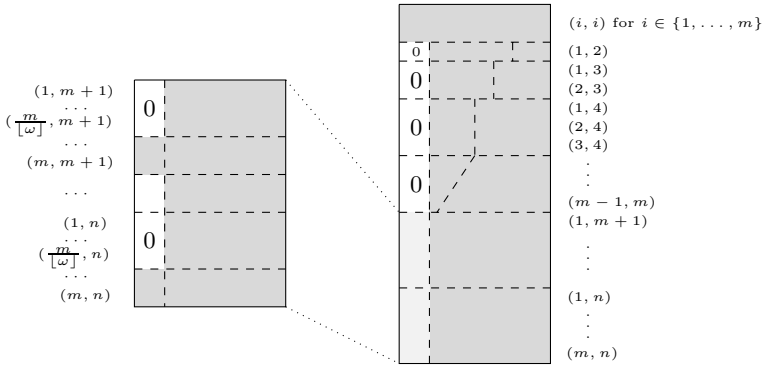


Fig. 4. Upper part of $\Phi^{\intercal}$ for tight analysis

This is possible as the number of coefficients $\widetilde{\gamma}_{i,j}^{(k)}$ for $1 \leq i \leq (m/\lfloor \omega \rfloor)$ equals the number of independent variables in the j-th column of S due to $m = d_j$ (cf. corollary 2). To eliminate the linear terms $L_{i,j}$ (cf. equation (5)) we have to solve $\omega\,(m - (m/\omega)) = (\omega - 1)m$ equations, which equals the number of variables and thus yields a solution. We obtain ω equations of the form (8). Using the Frobenius Homomorphism several times $x \mapsto x^{2^{p-1}}$ over $\mathbb{F}_{2^p}$ leads to equation (8).

$$\sum_{i=1}^{m} \beta_{i,j} y_i^2 + c_j = 0 \qquad \text{for } j \in \{1, \ldots, \omega\}$$

$$\Leftrightarrow \sum_{i=1}^{m} \beta_{i,j}^{2^{p-1}} y_i + c_j^{2^{p-1}} = 0 \tag{8}$$

After using equation (8) to eliminate $\lfloor \omega \rfloor$ variables in the remaining $(m - \lfloor \omega \rfloor)$ equations we obtain a $\mathcal{MQ}$-system of $(m - \lfloor \omega \rfloor)$ variables and equations. Note that if $\frac{m}{\lfloor \omega \rfloor} \notin \mathbb{N}$ this very tight analysis fails and we are only able to eliminate $(\lfloor \omega \rfloor - 1)$ instead of $\lfloor \omega \rfloor$ variables.

5 Complexity Analysis

The complexity of our approach is on the one hand the complexity of the preprocessing step, *i.e.* applying the transformation of variables and on the other hand the complexity of solving the obtained $\mathcal{MQ}$-system by some algorithm like F_5. In the case of $\frac{m}{\lfloor \omega \rfloor} \notin \mathbb{N}$ we would have to solve $(m-1)$ systems of linear equations of different dimensions to eliminate the $O \times O$ coefficients. Deleting the $O \times V$ coefficients requires solving another linear system of size $(\lfloor \omega \rfloor - 1)m$. The overall complexity is

$$\mathcal{O}\left(\sum_{i=1}^{m} \left(\left(\lfloor \omega \rfloor - 1\right) i\right)^3\right) = \mathcal{O}\left(m(\lfloor \omega \rfloor\, m)^3\right).$$

In the tight case of $\frac{m}{\lfloor \omega \rfloor} \in \mathbb{N}$ we have to solve $(m-1)$ systems of linear equations of different dimension to eliminate the $O \times O$ coefficients and another $(\lfloor \omega \rfloor - 1)m$ systems of dimension m to delete some of the $O \times V$ coefficients. Deleting the remaining coefficients using (5) requires solving another linear system of size $(\lfloor \omega \rfloor - 1)m$. The overall complexity is

$$\mathcal{O}\left(m(\lfloor \omega \rfloor - 1)m^3 + \left((\lfloor \omega \rfloor - 1)m\right)^3 + \sum_{i=1}^{m-1} \left(\lfloor \omega \rfloor\ i\right)^3\right) = \mathcal{O}\left(m(\lfloor \omega \rfloor\, m)^3\right).$$

To determine the complexity of solving a $\mathcal{MQ}$-system using F_5 we refer to [1]. In a nutshell, we first have to calculate the degree of regularity. For semi-regular sequences, which generic systems are assumed to be, the degree of regularity is the index of the first non-positive coefficient in the Hilbert series $S_{m,n}$ with

$$S_{m,n} = \frac{\prod_{i=1}^{m}(1 - z^{d_i})}{(1-z)^n},$$

where d_i is the degree of the i-th equation. Then the complexity of solving a zero-dimensional (semi-regular) system using F_5 [2, Prop. 2.2] is

$$\mathcal{O}\left(\left(m\binom{n + d_{reg} - 1}{d_{reg}}\right)^{\alpha}\right),$$

with $2 \leq \alpha \leq 3$ the linear algebra constant. We use $\alpha = 2$ as the equations are sparse and to be comparable to the results of [2], who gave the currently best attack against UOV.

Table 1. Attack complexity against UOV, comparing the improved attack with the previously known best attack using the hybrid approach of [2], *i.e.* guessing g variables beforehand. The previously secure value for $m = 26$ is marked in **bold**. Rows where m is divisible by $\omega = 3$ are marked with "←".

	direct attack [2]			our approach					improvement	
m	g	d_{reg}	$\log_2$ complexity	$\log_2$ Gauss	g	d_{reg}	$\log_2$ complexity	overall	$\log_2$	
24	1	13	78.0	19.8	1	11	68.2	68.2	9.8	←
25	1	13	79.5	18.1	1	12	73.6	73.6	5.9	
26	1	14	**83.7**	18.3	1	13	78.1	**78.1**	5.6	
27	1	14	85.7	20.4	1	13	78.1	78.1	7.6	←
28	1	15	89.4	18.7	1	14	83.7	83.7	5.7	
29	1	15	90.6	18.9	1	14	85.1	85.1	5.5	
30	1	16	95.0	21.0	1	14	85.1	85.1	9.9	←

Table 1 give some examples of the complexity of our algorithm applied to attack UOV. The underlying field is $\mathbb{F}_{2^8}$ and nowadays parameters are $n = 78$ variables and $m = 26$ equations, *i.e.* $\omega = 3$ [2]. We use the HybridF$_5$ algorithm and thus g denote the optimal number of variables to guess. Referring to table 1, we see that today's parameter of UOV are insecure (row with **bold** values). Based on our analysis, we suggest UOV with $m = 28$ for $n = 3m$.

Table 2. Experimental complexities of our approach (TW) in seconds [s]

			$\mathbb{F}_{2^6}$		$\mathbb{F}_{2^8}$	
ω	m	n	standard	TW	standard	TW
1	3	6	0	0	0	0
1	4	8	0.01	0	0	0
1	5	10	0.03	0.02	0.03	0.02
1	6	12	0.25	0.06	0.25	0.06
1	7	14	2.94	0.33	3.10	0.34
1	8	16	33	3.51	36	3.60
1	9	18	460	43	479	45
2	3	9	0	0	0	0
2	4	12	0.01	0	0	0.03
2	5	15	0.05	0.07	0.05	0.08
2	6	18	0.35	0.29	0.36	0.28
2	7	21	3.40	0.80	3.45	0.80
2	8	24	38	2.19	41	2.25
2	9	27	520	8.34	546	8.10

We have implemented our algorithm using the software system Magma V2.16-1 [4] and found it to be in line with the theoretical predictions. All experiments were

performed on a Intel Xeon X33502.66GHz (Quadcore) with 8 GB of RAM using only one core. Table 2 compares the time complexities of the standard approach of guessing v variables and solve the remaining $\mathcal{MQ}$-system in m variables and our algorithm for various parameter sets. The source code of our implementation can be found on the homepage of the first author.

6 Odd Cases

In this section we outline some ideas to extend our results to fields of *odd* characteristic. Hence we are now working over $\mathbb{F}_{p^k}$ for some prime $p \neq 2$ and $k \in \mathbb{N}_{>0}$. Unfortunately there is no straightforward extension of our ideas. The main problem is that equations (8) are not longer linear and thus we are not able to eliminate variables in the remaining equations. Nevertheless, Gröbner algorithms are empirically faster on systems containing equations (8), but it is hard to quantify the gain from a theoretical perspective. An argument that this task is inherently difficult is also the odd-characteristic algorithm of Courtois *et al.* [6]. It extended the algorithm for even characteristics by Kipnis-Patarin-Goubin [15] to the odd case. However, it requires now $n \geq 2^{\frac{m}{7}}(m+1)$—which is infeasible in practice.

However, for *small* values of ω, we can actually adapt our algorithm from even to odd characteristics. This coincides with the cryptanalytically interesting case of UOV, where we have $\omega \approx 3$ for efficiency reasons. Our main concern is to obtain some equations $y_i = g(y_{\omega+1}, \ldots, y_m)$ for $1 \leq i \leq \omega$ and some polynomial function g of low degree from the ω equations given by (8). They will be used to eliminate the variables $y_1, \ldots, y_\omega$. Therefore, we need to determine coefficients $(\beta_{\omega+1,i}, \ldots, \beta_{m,i})$ such that they are linearly dependent on $(\beta_{\omega+1,1}, \ldots, \beta_{m,1})$ for every $i \in \{2, \ldots, \omega\}$. This way we could eliminate these parts in equations 2 to ω by Gaussian Elimination. Producing an upper triangular form on these equations leads to $y_i^2 = c_i$ for $2 \leq i \leq \omega$, which is efficiently solvable for finite fields of size p^k. Still, the question remains how to determine the coefficients β_{ij}. Fixing $\widetilde{\gamma}_{i,i}^{(j)} = \beta_{ij}$ to some value for $2 \leq j \leq \omega$ leads to a *quadratic* system of $(\omega - 1)$ equations and variables s_{ij} (cf. figure 1)—so we seem to be back on square one. However, if ω is sufficiently small, *i.e.* smaller than 20, we can use any $\mathcal{MQ}$-solver, such as Gröbner algorithms for this task.

7 Conclusions and Open Questions

In this article we showed a more "gradual" change between exponential running time in the determined case ($n = m$) and the polynomial running time in massively underdetermined case ($n \geq m(m+1)$). Previously, this change was abrupt (Kipnis-Goubin-Patarin), *i.e.* there was a polynomial time algorithm in one case, and a fully exponential algorithm in the other. The situation is depicted in figure 5.

Our algorithm can be used as a general preprocessing step for further applications. Applied to UOV we would have to raise parameters from $m = 26$ to $m = 28$ in order to make the scheme secure.

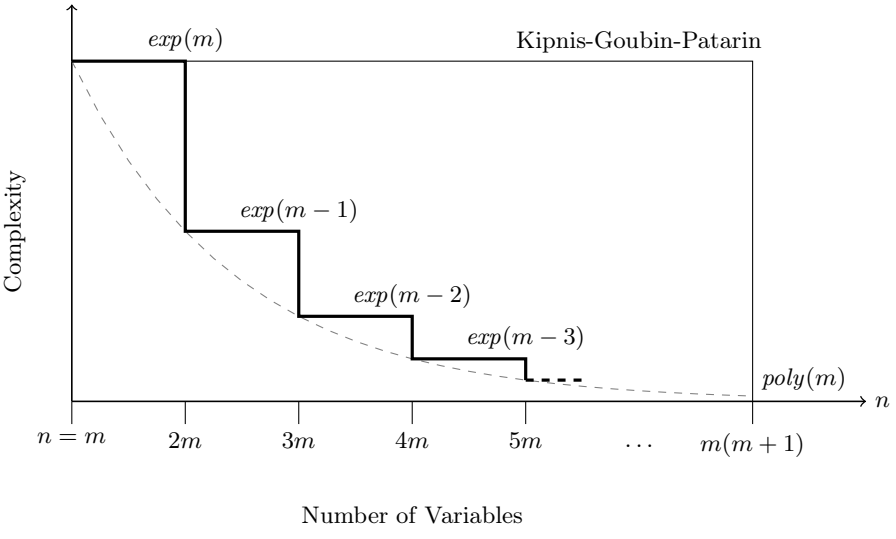


Fig. 5. Achievement of this paper (solid), compared with Kipnis-Goubin-Patarin (thin) for fixed m

Acknowledgments. We want to thank Gottfried Herold (Bochum) for fruitful discussions and helpful remarks. Furthermore we thank the reviewers for helpful comments.

The authors were supported by the German Science Foundation (DFG) through an Emmy Noether grant where the second author is principal investigator. All authors were in part supported by the European Commission through the IST Programme under contract *ICT-2007-216676 Ecrypt II*.

References

[1] Bardet, M., Faugère, J.-C., Salvy, B., Yang, B.-Y.: Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems. In: Gianni, P. (ed.) MEGA 2005, Sardinia, Italy (2005)

[2] Bettale, L., Faugère, J.-C., Perret, L.: Hybrid approach for solving multivariate systems over finite fields. Journal of Mathematical Cryptology 3, 177–197 (2009)

[3] Braeken, A., Wolf, C., Preneel, B.: A Study of the Security of Unbalanced Oil and Vinegar Signature Schemes. In: Menezes, A.J. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 29–43. Springer, Heidelberg (2005), http://eprint.iacr.org/2004/222/

[4] Computational Algebra Group, University of Sydney. The MAGMA Computational Algebra System for Algebra, Number Theory and Geometry, http://magma.maths.usyd.edu.au/magma/

[5] Courtois, N.T.: Higher Order Correlation Attacks, XL Algorithm and Cryptanalysis of Toyocrypt. In: Lee, P.J., Lim, C.H. (eds.) ICISC 2002. LNCS, vol. 2587, pp. 182–199. Springer, Heidelberg (2003)

[6] Courtois, N., Goubin, L., Meier, W., Tacier, J.-D.: Solving Underdefined Systems of Multivariate Quadratic Equations. In: Naccache, D., Paillier, P. (eds.) PKC 2002. LNCS, vol. 2274, pp. 211–227. Springer, Heidelberg (2002)
[7] Courtois, N.T., Daum, M., Felke, P.: On the Security of HFE, HFEv- and Quartz. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 337–350. Springer, Heidelberg (2003), http://eprint.iacr.org/2002/138
[8] Courtois, N.T., Klimov, A.B., Patarin, J., Shamir, A.: Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 392–407. Springer, Heidelberg (2000), Extended Version: http://www.minrank.org/xlfull.pdf
[9] Faugère, J.-C.: A new efficient algorithm for computing Gröbner bases (F_4). Journal of Pure anNational Institute of Standards and Technologyd Applied Algebra 139, 61–88 (1999)
[10] Faugère, J.-C.: A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5). In: International Symposium on Symbolic and Algebraic Computation — ISSAC 2002, pp. 75–83. ACM Press (July 2002)
[11] Faugère, J.-C., Joux, A.: Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 44–60. Springer, Heidelberg (2003)
[12] Faugère, J.-C., Otmani, A., Perret, L., Tillich, J.-P.: Algebraic Cryptanalysis of McEliece Variants with Compact Keys. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 279–298. Springer, Heidelberg (2010)
[13] Fischer, S., Meier, W.: Algebraic Immunity of S-Boxes and Augmented Functions. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 366–381. Springer, Heidelberg (2007)
[14] Garey, M.R., Johnson, D.S.: Computers and Intractability — A Guide to the Theory of NP-Completeness. W.H. Freeman and Company (1979)
[15] Kipnis, A., Patarin, J., Goubin, L.: Unbalanced Oil and Vinegar signature schemes — extended version, 17 pages (June 11, 2003), http://www.citeseer/231623.html
[16] Murphy, S., Robshaw, M.J.: Essential Algebraic Structure within the AES. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 1–16. Springer, Heidelberg (2002)
[17] Patarin, J.: The oil and vinegar signature scheme. Presented at the Dagstuhl Workshop on Cryptography (September 1997), transparencies
[18] Petzoldt, A., Thomae, E., Bulygin, S., Wolf, C.: Small Public Keys and Fast Verification for $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic Public Key Systems. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 475–490. Springer, Heidelberg (2011)
[19] Sugita, M., Kawazoe, M., Perret, L., Imai, H.: Algebraic Cryptanalysis of 58-Round SHA-1. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 349–365. Springer, Heidelberg (2007)
[20] Wolf, C., Preneel, B.: Equivalent Keys in HFE, C*, and Variations. In: Dawson, E., Vaudenay, S. (eds.) Mycrypt 2005. LNCS, vol. 3715, pp. 33–49. Springer, Heidelberg (2005); Extended version, http://eprint.iacr.org/2004/360/, 15 pages
[21] Wolf, C., Preneel, B.: Equivalent keys in multivariate quadratic public key systems. Journal of Mathematical Cryptology 4(4), 375–415 (2011)

Public-Key Identification Schemes Based on Multivariate Cubic Polynomials

Koichi Sakumoto

Sony Corporation 5-1-12 Kitashinagawa Shinagawa-ku,
Tokyo 141-0001, Japan
Koichi.Sakumoto@jp.sony.com

Abstract. Solving a system of multivariate polynomials over a finite field is a promising problem in cryptography. Recently, Sakumoto et al. proposed public-key identification schemes based on the quadratic version of the problem, which is called the MQ problem. However, it is still an open question whether or not it is able to build efficient constructions of public-key identification based on multivariate polynomials of degree greater than two. In this paper, we tackle the *cubic* case of this question and construct public-key identification schemes based on the *cubic* version of the problem, which is called the MC problem. The MQ problem is a special case of the MC problem. Our schemes consist of a protocol which is zero-knowledge argument of knowledge for the MC problem under the assumption of the existence of a non-interactive commitment scheme. For a practical parameter choice, the efficiency of our scheme is highly comparable to that of the schemes based on the MQ problem. Furthermore, the parallel version of our scheme also achieves the security under active attack with some additional cost.

Keywords: public-key identification scheme, zero knowledge, MQ problem, MC problem.

1 Introduction

Diversity of underlying mathematical problems is important for cryptography. Although the ones widely used today are factorization and a discrete logarithm problem, there are other various problems which are used for cryptography. Among them, a problem of solving a system of multivariate polynomials over a finite field is a promising problem. In particular the quadratic case of the problem is called the MQ problem. Even in the quadratic case, the associated decision problem is known to be NP-complete [14, 23], and a random instance of the problem is widely believed to be intractable. Naturally, the problem of degree greater than two is expected to be equally or more intractable than the quadratic one. The generic attacks on the MQ problem using Gröbner basis are known to have exponential complexity in time and space [3, 11], and there is no known polynomial-time quantum algorithm to solve the MQ problem in contrast to factorization or a discrete logarithm problem.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 172–189, 2012.

Over the past few decades, many studies have been made on cryptographic primitives based on multivariate polynomials. Most of them deal with quadratic polynomials [5, 18, 19, 21, 26], and some of them deal with polynomials of degree greater than two [6, 10, 12, 22, 32]. In symmetric cryptography, Berbain et al. proposed QUAD, which is a stream cipher with provable security based on the MQ problem [5]. In asymmetric cryptography, several public-key schemes have been proposed, which are known as multivariate public-key cryptography (MPKC) [18, 19, 21].

Recently, Sakumoto et al. proposed public-key identification schemes based on the MQ problem [26]. A remarkable advantage of their schemes is that they have provable security based on the conjectured intractability of the MQ problem under the assumption of the existence of a non-interactive commitment scheme. In fact, their schemes do *not* depend either on the Isomorphism of Polynomials (IP) problem or on the Functional Decomposition (FD) problem, while the other schemes in MPKC depend on the IP problem [18, 19, 21] or the FD problem [22]. Their new cut-and-choose techniques are specialized for the quadratic case and are based on the bilinearity of the map $(\mathbf{x}, \mathbf{y}) \mapsto \mathbf{F}_2(\mathbf{x} + \mathbf{y}) - \mathbf{F}_2(\mathbf{x}) - \mathbf{F}_2(\mathbf{y})$, where $\mathbf{F}_d$ is a function consisting of multivariate polynomials of degree d. In fact, their techniques do not work in the case of degree $d > 2$, because the map $(\mathbf{x}, \mathbf{y}) \mapsto \mathbf{F}_d(\mathbf{x} + \mathbf{y}) - \mathbf{F}_d(\mathbf{x}) - \mathbf{F}_d(\mathbf{y})$ where $d > 2$ is not linear either in $\mathbf{x}$ or in $\mathbf{y}$. Thus it is an interesting question whether or not it is able to build efficient constructions of public-key identification based on multivariate polynomials of degree greater than two.

In this paper, we tackle the *cubic* case of this question and construct public-key identification schemes based on the MC problem, which is a problem of solving a system of multivariate *cubic* polynomials over a finite field. The MQ problem is a special case of the MC problem, and we have less perspective on solving the MC problem compared to the MQ problem even considering the state-of-the-art algorithms [7, 8, 11]. It is important for higher security to be based on such a more intractable problem even though the MQ problem is very hard. A function consisting of multivariate cubic polynomials is also called an MC function.

We present two concrete protocols, a three-pass protocol and a five-pass one, which are statistical zero-knowledge argument of knowledge for the MC problem. Our schemes consisting of the protocol have provable security based on the conjectured intractability of the MC problem under the assumption of the existence of a non-interactive commitment scheme. Concretely, the identification schemes consisting of the *sequential* composition and the *parallel* composition of our protocol are secure against impersonation under *active* attack and under *passive* attack, respectively. Moreover, the parallel version of our scheme is also secure under active attack if its underlying MC function is substantially compressing (e.g., mapping 160 bits to 80 bits). These levels of provable security are the same as those of the identification schemes based on the MQ problem. Of course, our schemes do *not* depend either on the IP problem or on the FD problem.

Efficiency of our five-pass protocol is highly comparable to that of the MQ-based schemes for a practical parameter choice. The size of communication data

in our five-pass protocol is 26,697 bits when the impersonation probability is less than 2^{-30}, while those in the three-pass protocol and in the five-pass protocol of [26] are 29,640 bits and 26,565 bits, respectively. Our five-pass protocol also has the small sizes of a public key and a secret key, 88 bits and 132 bits for 80-bit security, respectively. Although our schemes have the relatively large size of the system parameter, it can be reduced to a small seed, e.g., 128 bits, by employing a pseudo-random number generator. The technique is also used in the implementation of QUAD [2]. We note that cubic systems with only 33 variables and 22 equations over $\mathbb{F}_{2^4}$ achieve 80-bit security, while quadratic systems over $\mathbb{F}_{2^4}$ require 45 variables and 30 equations. This evaluation is derived from the way of [7] of selecting the minimum parameters for 80-bit security and contributes to the efficiency of our five-pass scheme.

Techniques for our constructions. First, we briefly review the techniques for the MQ-based construction. They employ the cut-and-choose approach, where a prover first divides her secret into shares and then proves the correctness of some shares depending on the choice of a verifier without revealing the secret itself.

Let $\mathbf{F}_{\mathtt{MQ}}$ be a function $(x_1, \ldots, x_n) \mapsto (y_1, \ldots, y_m)$ where $y_l = \sum_{i,j} a_{l,i,j} x_i x_j + \sum_i b_{l,i} x_i$. The function $\mathbf{F}_{\mathtt{MQ}}$ is called an MQ function. The associated bilinear form of $\mathbf{F}_{\mathtt{MQ}}$ is defined as $\mathbf{G}_{\mathtt{MQ}}(\mathbf{x}, \tilde{\mathbf{x}}) = \mathbf{F}_{\mathtt{MQ}}(\mathbf{x}+\tilde{\mathbf{x}}) - \mathbf{F}_{\mathtt{MQ}}(\mathbf{x}) - \mathbf{F}_{\mathtt{MQ}}(\tilde{\mathbf{x}})$. It is easy to see the bilinearity of the function $\mathbf{G}_{\mathtt{MQ}}(\mathbf{x}, \tilde{\mathbf{x}})$, since it maps $(x_1, \ldots, x_n, \tilde{x}_1, \ldots, \tilde{x}_n) \mapsto (z_1, \ldots, z_m)$ where $z_l = \sum_{i,j} a_{l,i,j}(x_i \tilde{x}_j + \tilde{x}_i x_j)$. Let $\mathbf{s}$ be a secret key and $\mathbf{v} = \mathbf{F}_{\mathtt{MQ}}(\mathbf{s})$ the corresponding public key. When the secret key is divided as $\mathbf{s} = \mathbf{r}_0 + \mathbf{r}_1$, the public key $\mathbf{v} = \mathbf{F}_{\mathtt{MQ}}(\mathbf{r}_0 + \mathbf{r}_1)$ can be represented as $\mathbf{v} = \mathbf{F}_{\mathtt{MQ}}(\mathbf{r}_0) + \mathbf{F}_{\mathtt{MQ}}(\mathbf{r}_1) + \mathbf{G}_{\mathtt{MQ}}(\mathbf{r}_0, \mathbf{r}_1)$. This representation still contains the term $\mathbf{G}_{\mathtt{MQ}}(\mathbf{r}_0, \mathbf{r}_1)$ which depends on both $\mathbf{r}_0$ and $\mathbf{r}_1$. Then, the two vectors $\mathbf{r}_0$ and $\mathbf{F}_{\mathtt{MQ}}(\mathbf{r}_0)$ are also divided as $\mathbf{r}_0 = \mathbf{t}_0 + \mathbf{t}_1$ and $\mathbf{F}_{\mathtt{MQ}}(\mathbf{r}_0) = \mathbf{e}_0 + \mathbf{e}_1$. Accordingly, the public key can also be represented as $\mathbf{v} = \mathbf{e}_0 + \mathbf{e}_1 + \mathbf{F}_{\mathtt{MQ}}(\mathbf{r}_1) + \mathbf{G}_{\mathtt{MQ}}(\mathbf{t}_0, \mathbf{r}_1) + \mathbf{G}_{\mathtt{MQ}}(\mathbf{t}_1, \mathbf{r}_1)$, due to the bilinearity of $\mathbf{G}_{\mathtt{MQ}}$. As a result, it yields the following equations:

$$\mathbf{r}_0 - \mathbf{t}_0 = \mathbf{t}_1, \quad \mathbf{F}_{\mathtt{MQ}}(\mathbf{r}_0) - \mathbf{e}_0 = \mathbf{e}_1, \quad \text{and}$$
$$\mathbf{v} - \mathbf{G}_{\mathtt{MQ}}(\mathbf{t}_1, \mathbf{r}_1) - \mathbf{F}_{\mathtt{MQ}}(\mathbf{r}_1) - \mathbf{e}_1 = \mathbf{e}_0 + \mathbf{G}_{\mathtt{MQ}}(\mathbf{t}_0, \mathbf{r}_1).$$

Each side of each of the three equations can be checked by using some one of three tuples $(\mathbf{r}_0, \mathbf{t}_0, \mathbf{e}_0)$, $(\mathbf{r}_1, \mathbf{t}_1, \mathbf{e}_1)$, and $(\mathbf{r}_1, \mathbf{t}_0, \mathbf{e}_0)$, while no information on the secret key $\mathbf{s}$ can be obtained from one out of the three tuples. As described above the bilinearity of $\mathbf{G}_{\mathtt{MQ}}$ plays an important role in their dividing technique.

Then we consider the case of the MC function $\mathbf{F}_{\mathtt{MC}} : (x_1, \ldots, x_n) \mapsto (y_1, \ldots, y_m)$ where $y_l = \sum_{i,j,k} a_{l,i,j,k} x_i x_j x_k + \sum_{i,j} b_{l,i,j} x_i x_j + \sum_i c_{l,i} x_i$. Unfortunately, the mapping $(\mathbf{x}, \tilde{\mathbf{x}}) \mapsto \mathbf{F}_{\mathtt{MC}}(\mathbf{x} + \tilde{\mathbf{x}}) - \mathbf{F}_{\mathtt{MC}}(\mathbf{x}) - \mathbf{F}_{\mathtt{MC}}(\tilde{\mathbf{x}})$ is *not* bilinear, since it maps $(x_1, \ldots, x_n, \tilde{x}_1, \ldots, \tilde{x}_n) \mapsto (z_1, \ldots, z_m)$ where $z_l = \sum_{i,j,k} a_{l,i,j,k}(x_i x_j \tilde{x}_k + x_i \tilde{x}_j x_k + x_i \tilde{x}_j \tilde{x}_k + \tilde{x}_i x_j x_k + \tilde{x}_i x_j \tilde{x}_k + \tilde{x}_i \tilde{x}_j x_k) + \sum_{i,j} b_{l,i,j}(x_i \tilde{x}_j + \tilde{x}_i x_j)$. Thus the dividing technique using the mapping does not work in the cubic case. We also note that there is a trivial construction derived from the MQ-based scheme, because it is always possible to express degree three terms $x_i x_j x_k$ as degree two terms $w_{i,j} x_k$

by introducing auxiliary variables $w_{i,j}$ and equations $w_{i,j} - x_i x_j = 0$. However, this reduction makes the numbers of variables and equations much larger, and the construction becomes inefficient.

Therefore, in the cubic case, we introduce another function which is associated with $\mathbf{F}_{\mathtt{MC}}$. Concretely, we define a function $\mathbf{G}_{\mathtt{MC}} : (x_1, \ldots, x_n, \tilde{x}_1, \ldots, \tilde{x}_n) \mapsto (z_1, \ldots, z_m)$ where $z_l = \sum_{i,j,k} a_{l,i,j,k}(x_i\tilde{x}_j\tilde{x}_k + \tilde{x}_i x_j \tilde{x}_k + \tilde{x}_i\tilde{x}_j x_k) + \sum_{i,j} b_{l,i,j} x_i \tilde{x}_j$. The function $\mathbf{G}_{\mathtt{MC}}(\mathbf{x}, \tilde{\mathbf{x}})$ is linear in one argument $\mathbf{x}$. In this paper we call $\mathbf{G}_{\mathtt{MC}}$ the associated linear-in-one-argument (LOA) form of $\mathbf{F}_{\mathtt{MC}}$. By using the function $\mathbf{G}_{\mathtt{MC}}$, it is able to divide $\mathbf{F}_{\mathtt{MC}}(\mathbf{x} + \tilde{\mathbf{x}}) - \mathbf{F}_{\mathtt{MC}}(\mathbf{x}) - \mathbf{F}_{\mathtt{MC}}(\tilde{\mathbf{x}})$ into two parts $\mathbf{G}_{\mathtt{MC}}(\mathbf{x}, \tilde{\mathbf{x}})$ and $\mathbf{G}_{\mathtt{MC}}(\tilde{\mathbf{x}}, \mathbf{x})$ which are linear in $\mathbf{x}$ and in $\tilde{\mathbf{x}}$, respectively. In fact, it is seen that $\mathbf{G}_{\mathtt{MC}}(\mathbf{x}, \tilde{\mathbf{x}}) + \mathbf{G}_{\mathtt{MC}}(\tilde{\mathbf{x}}, \mathbf{x}) = \mathbf{F}_{\mathtt{MC}}(\mathbf{x} + \tilde{\mathbf{x}}) - \mathbf{F}_{\mathtt{MC}}(\mathbf{x}) - \mathbf{F}_{\mathtt{MC}}(\tilde{\mathbf{x}})$.

With this associated LOA form $\mathbf{G}_{\mathtt{MC}}$, our new dividing techniques for $\mathbf{F}_{\mathtt{MC}}$ are briefly described as follows. Let $\mathbf{s}$ be a secret key and $\mathbf{v} = \mathbf{F}_{\mathtt{MC}}(\mathbf{s})$ the corresponding public key. When the secret key is divided as $\mathbf{s} = \mathbf{r}_0 + \mathbf{r}_1$, the public key $\mathbf{v} = \mathbf{F}_{\mathtt{MC}}(\mathbf{r}_0 + \mathbf{r}_1)$ can be represented as $\mathbf{v} = \mathbf{F}_{\mathtt{MC}}(\mathbf{r}_0) + \mathbf{F}_{\mathtt{MC}}(\mathbf{r}_1) + \mathbf{G}_{\mathtt{MC}}(\mathbf{r}_0, \mathbf{r}_1) + \mathbf{G}_{\mathtt{MC}}(\mathbf{r}_1, \mathbf{r}_0)$. This representation still contains the terms $\mathbf{G}_{\mathtt{MC}}(\mathbf{r}_0, \mathbf{r}_1)$ and $\mathbf{G}_{\mathtt{MC}}(\mathbf{r}_1, \mathbf{r}_0)$ which depend on both $\mathbf{r}_0$ and $\mathbf{r}_1$. Then, the two vectors $\mathbf{r}_0$ and $\mathbf{F}_{\mathtt{MC}}(\mathbf{r}_0) + \mathbf{G}_{\mathtt{MC}}(\mathbf{r}_1, \mathbf{r}_0)$ are also divided as $\mathbf{r}_0 = \mathbf{t}_0 + \mathbf{u}$ and $\mathbf{F}_{\mathtt{MC}}(\mathbf{r}_0) + \mathbf{G}_{\mathtt{MC}}(\mathbf{r}_1, \mathbf{r}_0) = \mathbf{e}_0 + \mathbf{e}_1$ similarly to the quadratic case. However, the latter equation also contains the term $\mathbf{G}_{\mathtt{MC}}(\mathbf{r}_1, \mathbf{r}_0)$ depending on both $\mathbf{r}_0$ and $\mathbf{r}_1$ in contrast to the case of $\mathbf{F}_{\mathtt{MQ}}(\mathbf{r}_0) = \mathbf{e}_0 + \mathbf{e}_1$. Thus $\mathbf{r}_1$ is further divided as $\mathbf{r}_1 = \mathbf{t}_1 + \mathbf{u}$. Accordingly, the terms depending on both $\mathbf{r}_0$ and $\mathbf{r}_1$ are divided as $\mathbf{G}_{\mathtt{MC}}(\mathbf{r}_0, \mathbf{r}_1) = \mathbf{G}_{\mathtt{MC}}(\mathbf{t}_0, \mathbf{r}_1) + \mathbf{G}_{\mathtt{MC}}(\mathbf{u}, \mathbf{r}_1)$ and $\mathbf{G}_{\mathtt{MC}}(\mathbf{r}_1, \mathbf{r}_0) = \mathbf{G}_{\mathtt{MC}}(\mathbf{t}_1, \mathbf{r}_0) + \mathbf{G}_{\mathtt{MC}}(\mathbf{u}, \mathbf{r}_0)$, due to the linearity in one argument. As a result, it yields the following equations:

$$\begin{gathered}
\mathbf{r}_0 - \mathbf{u} = \mathbf{t}_0, \quad \mathbf{r}_1 - \mathbf{u} = \mathbf{t}_1, \\
\mathbf{G}_{\mathtt{MC}}(\mathbf{u}, \mathbf{r}_1) + \mathbf{e}_1 = \mathbf{v} - \mathbf{F}_{\mathtt{MC}}(\mathbf{r}_1) - \mathbf{G}_{\mathtt{MC}}(\mathbf{t}_0, \mathbf{r}_1) - \mathbf{e}_0, \quad \text{and} \\
\mathbf{G}_{\mathtt{MC}}(\mathbf{u}, \mathbf{r}_0) - \mathbf{e}_0 = \mathbf{e}_1 - \mathbf{F}_{\mathtt{MC}}(\mathbf{r}_0) - \mathbf{G}_{\mathtt{MC}}(\mathbf{t}_1, \mathbf{r}_0).
\end{gathered}$$

Each side of each of the four equations can be checked by using some one of four tuples $(\mathbf{r}_0, \mathbf{u}, \mathbf{e}_0)$, $(\mathbf{r}_0, \mathbf{t}_1, \mathbf{e}_1)$, $(\mathbf{r}_1, \mathbf{u}, \mathbf{e}_1)$, and $(\mathbf{r}_1, \mathbf{t}_0, \mathbf{e}_0)$, while no information on the secret key $\mathbf{s}$ can be obtained from one out of the four tuples. We note that using the common $\mathbf{u}$ in dividing $\mathbf{r}_0 = \mathbf{t}_0 + \mathbf{u}$ and $\mathbf{r}_1 = \mathbf{t}_1 + \mathbf{u}$ does not damage the zero-knowledge property, since each of the four tuples contains only one out of $\mathbf{t}_0$, $\mathbf{t}_1$, and $\mathbf{u}$.

Related work. Identification schemes based on Permuted Kernels (PK) [27], binary Syndrome Decoding (SD) [28, 30], Constrained Linear Equations (CLE) [29], Permuted Perceptrons (PP) [24, 25], and q-ary SD [9] have some features similar to the MQ-based schemes [26] and ours as follows. First, these schemes depend on the hardness of a random instance of each of the problems whose associated decision version is known to be NP-complete. Second, their protocols have perfect correctness. Finally, assuming the existence of a non-interactive commitment scheme, the sequential version and the parallel version of the schemes are secure against impersonation under active attack and passive attack, respectively. However, it is not explicitly known that the parallel versions of these

schemes achieve the security under active attack. The efficiency of our scheme is highly comparable to that of these schemes. Indeed, the data sizes of a public key of the schemes of [9, 24, 25, 27–30] are between 245 bits and 2,450 bits, and those of communication are between 27,234 bits and 120,652 bits.

Paper Organization. The remainder of this paper is organized as follows. In Section 2 we define some notions related to the MC function and evaluate the intractability of the function. In Section 3 and Section 4, our 3-pass construction and 5-pass one are presented, respectively. In Section 5 we discuss their security and efficiency for a practical parameter choice. In Section 6 we study the security of the parallel composition of our scheme at the expense of the efficiency. Finally, we close with some extensions, open problems, and conclusion.

2 Multivariate Cubic Functions

In this section we define a family of MC functions $\mathcal{MC}(n, m, \mathbb{F}_q)$ and study its parameters achieving 80-bit security.

Definition 1. *We denote by $\mathcal{MC}(n, m, \mathbb{F}_q)$ a family of functions $\{\mathbf{F} = (f_1, \ldots, f_m)\}$ such that, for $l = 1, \ldots, m$, $f_l(x_1, \ldots, x_n) = \sum_{i,j,k} a_{l,i,j,k} x_i x_j x_k + \sum_{i,j} b_{l,i,j} x_i x_j + \sum_i c_{l,i} x_i$ where $a_{l,i,j,k}, b_{l,i,j}, c_{l,i} \in \mathbb{F}_q$. We call $\mathbf{F} \in \mathcal{MC}(n, m, \mathbb{F}_q)$ an MC function.*

For the simplicity, constant terms are omitted without any security loss. The MQ function is a special case of the MC function, where the coefficients $a_{l,i,j,k}$ are all zero. For the MC function $\mathbf{F}$, we define a binary relation $R_{\mathbf{F}} = \{(\mathbf{v}, \mathbf{x}) \in \mathbb{F}_q^m \times \mathbb{F}_q^n : \mathbf{v} = \mathbf{F}(\mathbf{x})\}$. and a set $R_{\mathbf{F}}(\mathbf{v}) = \{\mathbf{x} : (\mathbf{v}, \mathbf{x}) \in R_{\mathbf{F}}\}$. Given an instance $\mathbf{F} \in \mathcal{MC}(n, m, \mathbb{F}_q)$ and a vector $\mathbf{v} \in \mathbb{F}_q^m$, the MC problem is finding a solution $\mathbf{s} \in R_{\mathbf{F}}(\mathbf{v})$. The associated linear-in-one-argument (LOA) form of the MC function is defined as follows.

Definition 2. *Let $\mathbf{F} = (f_1, \ldots, f_m) \in \mathcal{MC}(n, m, \mathbb{F}_q)$ and $f_l(x_1, \ldots, x_n) = \sum_{i,j,k} a_{l,i,j,k} x_i x_j x_k + \sum_{i,j} b_{l,i,j} x_i x_j + \sum_i c_{l,i} x_i$. Then a function $\mathbf{G} = (g_1, \ldots, g_m)$ is called the associated linear-in-one-argument (LOA) form of $\mathbf{F}$ if, for $l = 1, \ldots, m$, $g_l(x_1, \ldots, x_n, y_1, \ldots, y_n) = \sum_{i,j,k} a_{l,i,j,k}(x_i y_j y_k + y_i x_j y_k + y_i y_j x_k) + \sum_{i,j} b_{l,i,j} x_i y_j$.*

When $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ are vectors of n variables, the associated LOA form $\mathbf{G}(\mathbf{x}, \mathbf{y})$ is linear with respect to the first argument $\mathbf{x}$. Moreover, it satisfies that $\mathbf{F}(\mathbf{x} + \mathbf{y}) = \mathbf{F}(\mathbf{x}) + \mathbf{G}(\mathbf{x}, \mathbf{y}) + \mathbf{G}(\mathbf{y}, \mathbf{x}) + \mathbf{F}(\mathbf{y})$.

Then, we study the intractability of the MC function. An intractability assumption for a random instance of $\mathcal{MC}(n, m, \mathbb{F}_q)$ is defined as follows.

Definition 3. *For polynomially bounded functions $n = n(\lambda)$, $m = m(\lambda)$, and $q = q(\lambda)$, it is said that $\mathcal{MC}(n, m, \mathbb{F}_q)$ is intractable if there is no polynomial-time algorithm that takes $(\mathbf{F}, \mathbf{v})$ generated via $\mathbf{F} \in_R \mathcal{MC}(n, m, \mathbb{F}_q)$, $\mathbf{s} \in_R \mathbb{F}_q^n$, and $\mathbf{v} \leftarrow \mathbf{F}(\mathbf{s})$ and finds a preimage $\mathbf{s}' \in \mathbb{F}_q^n$ such that $\mathbf{F}(\mathbf{s}') = \mathbf{v}$ with non-negligible probability $\epsilon(\lambda)$.*

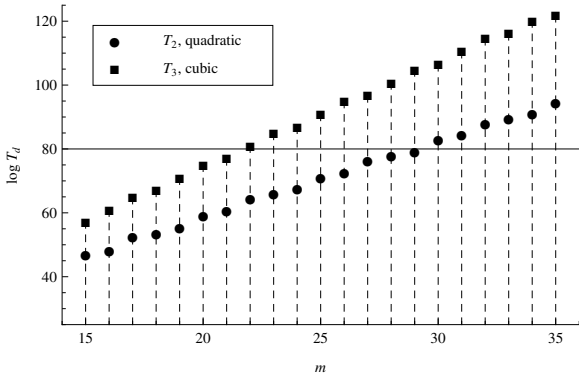


Fig. 1. The complexity of the hybrid approach where $n = m$, $q = 2^4$, and $w = 2$

All the state-of-the-art solving techniques have exponential complexity to break the intractability [7, 8, 11]. In particular, it is known that complexity of generic attacks using Gröbner basis is exponential in time and space [3, 11]. In this paper we use two sets of parameters $\mathcal{MC}(84, 80, \mathbb{F}_2)$ and $\mathcal{MC}(33, 22, \mathbb{F}_{2^4})$ for 80-bit security. It is easy to see that the former achieves 80-bit security, because even a *quadratic* system with 84 variables and 80 equations over $\mathbb{F}_2$ satisfies 80-bit security [26]. In fact, the complexity of the improved exhaustive search algorithm [8] and the F_5 algorithm [11] to break $\mathcal{MC}(84, 80, \mathbb{F}_2)$ is more than 2^{80}. On the other hand, the latter requires more detailed analysis. The hybrid approach which is proposed by Bettale et al. [7] is the best known algorithm for solving multivariate cubic systems over $\mathbb{F}_{2^4}$. We follow their evaluation method of [7] to select the minimal parameters for 80-bit security and obtain the parameter set $\mathcal{MC}(33, 22, \mathbb{F}_{2^4})$ as follows.

Let $D(n, m, d)$ be the degree of regularity of a semi-regular system with m equations of degree d in n variables. The complexity of solving a semi-regular system with n variables and m equations of degree d over $\mathbb{F}_q$ is estimated as $\min_{0 \le k \le n}(q^k \cdot (m \cdot \binom{n-k-1+D(n-k,m,d)}{D(n-k,m,d)})^w)$ where $2 \le w \le 3$. They stated that $D(n, m, d)$ corresponds to the index i of the first non-positive coefficient c_i of the series $\sum_{i>0} c_i \cdot z^i = \frac{(1-z^d)^m}{(1-z)^n}$. Let $T_d(m)$ be the complexity of the hybrid approach where $n = m$, $q = 2^4$, and $w = 2$. Figure 1 shows the comparison of $T_2(m)$ and $T_3(m)$. The complexity $T_3(m)$ increases faster than $T_2(m)$. In particular, $\min\{m | T_3(m) > 2^{80}\} = 22$ and $T_3(22) \approx 2^{81}$. Finally, the number of variables n is conservatively chosen as $n = \frac{3}{2}m$. Thus we can see that $\mathcal{MC}(33, 22, \mathbb{F}_{2^4})$ achieves 80-bit security.

3 A 3-Pass Protocol

This section describes our 3-pass protocol which is statistical zero-knowledge argument of knowledge for $R_{\mathbf{F}}$ with knowledge error 3/4, assuming the existence

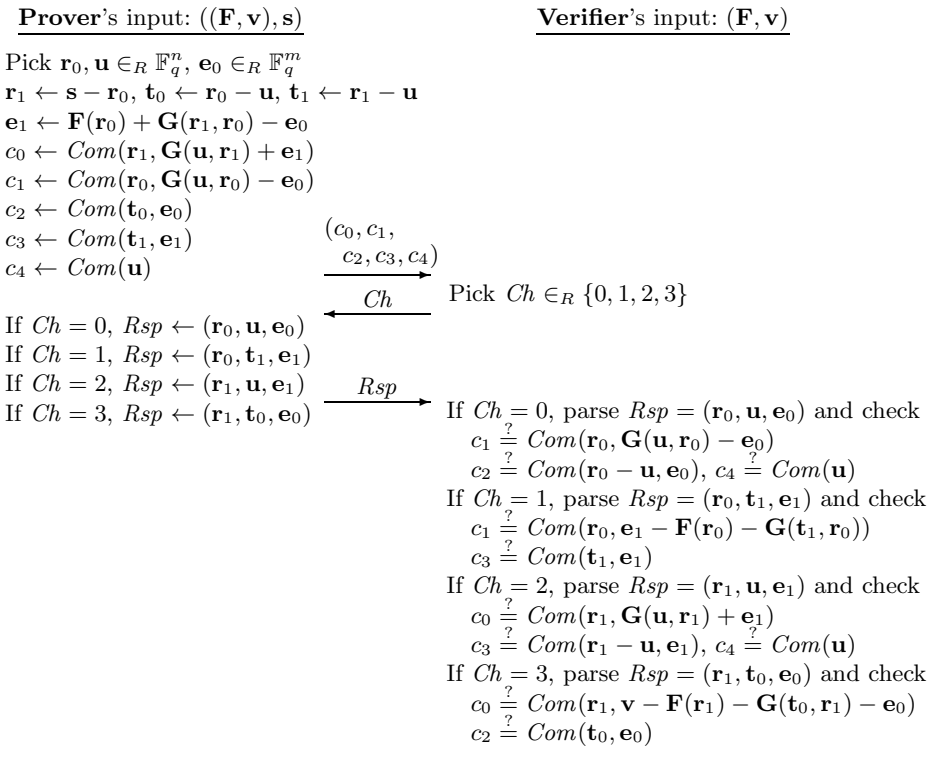


Fig. 2. Our 3-pass protocol

of a non-interactive commitment scheme Com which is statistically hiding and computationally binding.

We begin with describing a setup algorithm and a key-generation algorithm. Let λ be a security parameter. Let $n = n(\lambda)$, $m = m(\lambda)$, and $q = q(\lambda)$ be polynomially bounded functions. The setup algorithm `Setup` takes 1^λ and outputs a system parameter $\mathbf{F} \in_R \mathcal{MC}(n, m, \mathbb{F}_q)$ which consists of m-tuple of random multivariate cubic polynomials. The key-generation algorithm `Gen` takes $\mathbf{F}$. After choosing a random vector $\mathbf{s} \in_R \mathbb{F}_q^n$, `Gen` computes $\mathbf{v} \leftarrow \mathbf{F}(\mathbf{s})$, then outputs $(pk, sk) = (\mathbf{v}, \mathbf{s})$.

The basic idea for our 3-pass construction is that a prover proves that she has a tuple $(\mathbf{r}_0, \mathbf{r}_1, \mathbf{u}, \mathbf{t}_0, \mathbf{t}_1, \mathbf{e}_0, \mathbf{e}_1)$ satisfying

$$\mathbf{G}(\mathbf{u}, \mathbf{r}_1) + \mathbf{e}_1 = \mathbf{v} - \mathbf{F}(\mathbf{r}_1) - \mathbf{G}(\mathbf{t}_0, \mathbf{r}_1) - \mathbf{e}_0, \tag{1}$$

$$\mathbf{t}_0 = \mathbf{r}_0 - \mathbf{u}, \tag{2}$$

$$\mathbf{t}_1 = \mathbf{r}_1 - \mathbf{u}, \tag{3}$$

$$\text{and} \quad \mathbf{G}(\mathbf{u}, \mathbf{r}_0) - \mathbf{e}_0 = \mathbf{e}_1 - \mathbf{F}(\mathbf{r}_0) - \mathbf{G}(\mathbf{t}_1, \mathbf{r}_0), \tag{4}$$

since if the tuple satisfies (1), (2), (3), and (4) then $\mathbf{v} = \mathbf{F}(\mathbf{r}_0 + \mathbf{r}_1)$. Note that $\mathbf{G}$ is the associated LOA form of $\mathbf{F}$. Then, corresponding to a challenge

$Ch \in \{0,1,2,3\}$ of a verifier, the prover reveals one out of four tuples $(\mathbf{r}_0, \mathbf{u}, \mathbf{e}_0)$, $(\mathbf{r}_0, \mathbf{t}_1, \mathbf{e}_1)$, $(\mathbf{r}_1, \mathbf{u}, \mathbf{e}_1)$, and $(\mathbf{r}_1, \mathbf{t}_0, \mathbf{e}_0)$. The verifier can check each side of each of the equations (1), (2), (3), and (4) by using some one of the four tuples. Such vectors $\mathbf{r}_0, \mathbf{r}_1, \mathbf{u}, \mathbf{t}_0, \mathbf{t}_1, \mathbf{e}_0, \mathbf{e}_1$ are produced by using the dividing techniques described in Section 1. Thus, when $\mathbf{r}_0$, $\mathbf{u}$, and $\mathbf{e}_0$ are randomly chosen, the verifier can obtain no information on the secret key $\mathbf{s}$ from only one out of the four tuples.

The 3-pass protocol is described in Figure 2. For the simplicity, a random string ρ in Com is not written explicitly. The verifier finally outputs 1 if all of the checks "$\stackrel{?}{=}$" are passed, otherwise outputs 0. This is denoted by $0/1 \leftarrow Dec(\mathbf{F}, \mathbf{v}; (c_0, c_1, c_2, c_3, c_4), Ch, Rsp)$. It is easy to see that the verifier always accepts an interaction with the honest prover. Thus the 3-pass protocol has perfect correctness.

Now we show two properties of the protocol in Theorem 4 and Theorem 5 as follows.

Theorem 4. *The 3-pass protocol is statistically zero knowledge when the commitment scheme Com is statistically hiding.*

Proof sketch. Let $\mathcal{S}$ be a simulator which takes $\mathbf{F}$ and $\mathbf{v}$ without knowing $\mathbf{s}$, and interacts with a cheating verifier $\mathcal{CV}$. We show that the simulator $\mathcal{S}$ can impersonate the honest prover with probability 3/4.

The simulator $\mathcal{S}$ randomly chooses a value $Ch^* \in_R \{0,1,2,3\}$ and vectors $\mathbf{s}', \mathbf{r}'_0, \mathbf{u}' \in_R \mathbb{F}_q^n$, $\mathbf{e}'_0 \in_R \mathbb{F}_q^m$, where Ch^* is a prediction of what value the cheating verifier $\mathcal{CV}$ will *not* choose. Then, it computes $\mathbf{r}'_1 \leftarrow \mathbf{s}' - \mathbf{r}'_0$, $\mathbf{t}'_0 \leftarrow \mathbf{r}'_0 - \mathbf{u}'$, and $\mathbf{t}'_1 \leftarrow \mathbf{r}'_1 - \mathbf{u}'$. If $Ch^* \in \{0,1\}$ then it computes $\mathbf{e}'_1 \leftarrow \mathbf{v} - \mathbf{F}(\mathbf{r}'_1) - \mathbf{G}(\mathbf{r}'_0, \mathbf{r}'_1) - \mathbf{e}'_0$, else $\mathbf{e}'_1 \leftarrow \mathbf{F}(\mathbf{r}'_0) + \mathbf{G}(\mathbf{r}'_1, \mathbf{r}'_0) - \mathbf{e}'_0$. If $Ch^* = 2$ then it computes $c'_0 \leftarrow Com(\mathbf{r}'_1, \mathbf{v} - \mathbf{F}(\mathbf{r}'_1) - \mathbf{G}(\mathbf{t}'_0, \mathbf{r}'_1) - \mathbf{e}'_0)$, else $c'_0 \leftarrow Com(\mathbf{r}'_1, \mathbf{G}(\mathbf{u}', \mathbf{r}'_1) + \mathbf{e}'_1)$. If $Ch^* = 0$ then it computes $c'_1 \leftarrow Com(\mathbf{r}'_0, \mathbf{e}'_1 - \mathbf{F}(\mathbf{r}'_0) - \mathbf{G}(\mathbf{t}'_1, \mathbf{r}'_0))$, else $c'_1 \leftarrow Com(\mathbf{r}'_0, \mathbf{G}(\mathbf{u}', \mathbf{r}'_0) - \mathbf{e}'_0)$. It computes $c'_2 \leftarrow Com(\mathbf{t}'_0, \mathbf{e}'_0)$, $c'_3 \leftarrow Com(\mathbf{t}'_1, \mathbf{e}'_1)$, and $c'_4 \leftarrow Com(\mathbf{u}')$ and sends $(c'_0, c'_1, c'_2, c'_3, c'_4)$ to $\mathcal{CV}$.

Due to the statistically hiding property of Com, a challenge Ch from $\mathcal{CV}$ is different from Ch^* with probability 3/4. If $Ch \neq Ch^*$ then $(\mathbf{r}'_0, \mathbf{u}', \mathbf{e}'_0)$, $(\mathbf{r}'_0, \mathbf{t}'_1, \mathbf{e}'_1)$, $(\mathbf{r}'_1, \mathbf{u}', \mathbf{e}'_1)$, and $(\mathbf{r}'_1, \mathbf{t}'_0, \mathbf{e}'_0)$ are accepted responses to $Ch = 0$, 1, 2, and 3, respectively. Note that if $Ch^* \in \{0,1\}$ and $Ch = 3$ then it is seen that $\mathbf{v} - \mathbf{F}(\mathbf{r}'_1) - \mathbf{G}(\mathbf{t}'_0, \mathbf{r}'_1) - \mathbf{e}'_0 = \mathbf{e}'_1 + \mathbf{G}(\mathbf{r}'_0 - \mathbf{t}'_0, \mathbf{r}'_1) = \mathbf{e}'_1 + \mathbf{G}(\mathbf{u}', \mathbf{r}'_1)$, since $\mathbf{e}'_1 = \mathbf{v} - \mathbf{F}(\mathbf{r}'_1) - \mathbf{G}(\mathbf{r}'_0, \mathbf{r}'_1) - \mathbf{e}'_0$ and $\mathbf{t}'_0 = \mathbf{r}'_0 - \mathbf{u}'$. Note that if $Ch^* \in \{2,3\}$ and $Ch = 1$ then it is seen that $\mathbf{e}'_1 - \mathbf{F}(\mathbf{r}'_0) - \mathbf{G}(\mathbf{t}'_1, \mathbf{r}'_0) = \mathbf{G}(\mathbf{r}'_1 - \mathbf{t}'_1, \mathbf{r}'_0) - \mathbf{e}'_0 = \mathbf{G}(\mathbf{u}', \mathbf{r}'_0) - \mathbf{e}'_0$, since $\mathbf{e}'_1 = \mathbf{F}(\mathbf{r}'_0) + \mathbf{G}(\mathbf{r}'_1, \mathbf{r}'_0) - \mathbf{e}'_0$ and $\mathbf{t}'_1 = \mathbf{r}'_1 - \mathbf{u}'$. □

Theorem 5. *The 3-pass protocol is argument of knowledge for $R_{\mathbf{F}}$ with knowledge error 3/4 when the commitment scheme Com is computationally binding.*

Proof sketch. For $i \in \{0,1,2,3\}$, let $((c_0, c_1, c_2, c_3, c_4), Ch_i, Rsp_i)$ be a transcript such that $Dec(\mathbf{F}, \mathbf{v}; (c_0, c_1, c_2, c_3, c_4), Ch_i, Rsp_i) = 1$ and $Ch_i = i$. Then, by using the four transcripts, it is shown to be able to either break the binding property of Com or extract a solution for $\mathbf{v}$. Consider the situation where the responses are parsed as $Rsp_0 = (\tilde{\mathbf{r}}_0^{(0)}, \tilde{\mathbf{u}}^{(0)}, \tilde{\mathbf{e}}_0^{(0)})$, $Rsp_1 = (\tilde{\mathbf{r}}_0^{(1)}, \tilde{\mathbf{t}}_1^{(1)}, \tilde{\mathbf{e}}_1^{(1)})$,

$Rsp_2 = (\tilde{\mathbf{r}}_1^{(2)}, \tilde{\mathbf{u}}^{(2)}, \tilde{\mathbf{e}}_1^{(2)})$, and $Rsp_3 = (\tilde{\mathbf{r}}_1^{(3)}, \tilde{\mathbf{t}}_0^{(3)}, \tilde{\mathbf{e}}_0^{(3)})$. Then, it is seen that

$$\begin{aligned} c_0 &= Com(\tilde{\mathbf{r}}_1^{(2)}, \mathbf{G}(\tilde{\mathbf{u}}^{(2)}, \tilde{\mathbf{r}}_1^{(2)}) + \tilde{\mathbf{e}}_1^{(2)}) \\ &= Com(\tilde{\mathbf{r}}_1^{(3)}, \mathbf{v} - \mathbf{F}(\tilde{\mathbf{r}}_1^{(3)}) - \mathbf{G}(\tilde{\mathbf{t}}_0^{(3)}, \tilde{\mathbf{r}}_1^{(3)}) - \tilde{\mathbf{e}}_0^{(3)}), \qquad (5) \\ c_1 &= Com(\tilde{\mathbf{r}}_0^{(0)}, \mathbf{G}(\tilde{\mathbf{u}}^{(0)}, \tilde{\mathbf{r}}_0^{(0)}) - \tilde{\mathbf{e}}_0^{(0)}) \\ &= Com(\tilde{\mathbf{r}}_0^{(1)}, \tilde{\mathbf{e}}_1^{(1)} - \mathbf{F}(\tilde{\mathbf{r}}_0^{(1)}) - \mathbf{G}(\tilde{\mathbf{t}}_1^{(1)}, \tilde{\mathbf{r}}_0^{(1)})), \qquad (6) \\ c_2 &= Com(\tilde{\mathbf{r}}_0^{(0)} - \tilde{\mathbf{u}}^{(0)}, \tilde{\mathbf{e}}_0^{(0)}) = Com(\tilde{\mathbf{t}}_0^{(3)}, \tilde{\mathbf{e}}_0^{(3)}), \qquad (7) \\ c_3 &= Com(\tilde{\mathbf{t}}_1^{(1)}, \tilde{\mathbf{e}}_1^{(1)}) = Com(\tilde{\mathbf{r}}_1^{(2)} - \tilde{\mathbf{u}}^{(2)}, \tilde{\mathbf{e}}_1^{(2)}), \quad \text{and} \qquad (8) \\ c_4 &= Com(\tilde{\mathbf{u}}^{(0)}) = Com(\tilde{\mathbf{u}}^{(2)}). \qquad (9) \end{aligned}$$

If the two pairs of the arguments of *Com* are distinct on any one of the above equations, the binding property of *Com* is broken. Otherwise, the equation (5) yields $\mathbf{v} = \mathbf{G}(\tilde{\mathbf{u}}^{(2)} + \tilde{\mathbf{t}}_0^{(3)}, \tilde{\mathbf{r}}_1^{(2)}) + \mathbf{F}(\tilde{\mathbf{r}}_1^{(2)}) + \tilde{\mathbf{e}}_1^{(2)} + \tilde{\mathbf{e}}_0^{(3)}$. By combining it with the equations (6), (7), and (8), it is seen that $\mathbf{v} = \mathbf{F}(\tilde{\mathbf{r}}_0^{(0)}) + \mathbf{G}(\tilde{\mathbf{r}}_0^{(0)} + \tilde{\mathbf{u}}^{(2)} - \tilde{\mathbf{u}}^{(0)}, \tilde{\mathbf{r}}_1^{(2)}) + \mathbf{G}(\tilde{\mathbf{r}}_1^{(2)} + \tilde{\mathbf{u}}^{(0)} - \tilde{\mathbf{u}}^{(2)}, \tilde{\mathbf{r}}_0^{(0)}) + \mathbf{F}(\tilde{\mathbf{r}}_1^{(2)})$. Finally, putting it together with the equation (9), we obtain $\mathbf{v} = \mathbf{F}(\tilde{\mathbf{r}}_0^{(0)}) + \mathbf{G}(\tilde{\mathbf{r}}_0^{(0)}, \tilde{\mathbf{r}}_1^{(2)}) + \mathbf{G}(\tilde{\mathbf{r}}_1^{(2)}, \tilde{\mathbf{r}}_0^{(0)}) + \mathbf{F}(\tilde{\mathbf{r}}_1^{(2)}) = \mathbf{F}(\tilde{\mathbf{r}}_0^{(0)} + \tilde{\mathbf{r}}_1^{(2)})$. It means that a solution $\tilde{\mathbf{r}}_0^{(0)} + \tilde{\mathbf{r}}_1^{(2)}$ for $\mathbf{v}$ is extracted. □

Extension. A standard trick for saving the communication data size can be applied to our 3-pass protocol. The trick employs a collision resistant hash function H. Let $c_a = H(c_0, c_2)$ and $c_b = H(c_1, c_3)$ be hash values. In the first pass, a prover sends one hash value $c = H(c_a, c_b, c_4)$ instead of five commitments $(c_0, c_1, c_2, c_3, c_4)$. In the third pass, the pairs of the hash values (c_0, c_3), (c_a, c_4), (c_1, c_2), and (c_b, c_4) are appended to prover's responses *Rsp* for $Ch = 0$, 1, 2, and 3, respectively. Finally, a verifier checks $c = H(c_a, c_b, c_4)$. We note that the hash values c_a, c_b, and c_4 can be obtained from the prover's response *Rsp* in every case of $Ch = 0$, 1, 2, and 3. As a result, the number of hash values sent is reduced from 5 to 3. The modified version of our 3-pass protocol is also shown to be zero-knowledge argument of knowledge with knowledge error 3/4.

4 A 5-Pass Protocol

This section describes our 5-pass protocol which is statistical zero-knowledge argument of knowledge for $R_{\mathbf{F}}$ with knowledge error $1/2 + 1/2q$, assuming the existence of a non-interactive commitment scheme *Com* which is statistically hiding and computationally binding. The knowledge error of the 5-pass protocol is smaller than that of the 3-pass protocol when $q \geq 3$. The setup algorithm and the key-generation algorithm for the 5-pass protocol are identical to those for the 3-pass protocol.

In the 5-pass protocol, a prover also divides the secret key $\mathbf{s}$ and the public key $\mathbf{F}(\mathbf{s})$ as $\mathbf{s} = \mathbf{r}_0 + \mathbf{r}_1$ and $\mathbf{F}(\mathbf{s}) = \mathbf{F}(\mathbf{r}_0 + \mathbf{r}_1) = \mathbf{F}(\mathbf{r}_0) + \mathbf{F}(\mathbf{r}_1) + \mathbf{G}(\mathbf{r}_0, \mathbf{r}_1) + \mathbf{G}(\mathbf{r}_1, \mathbf{r}_0)$,

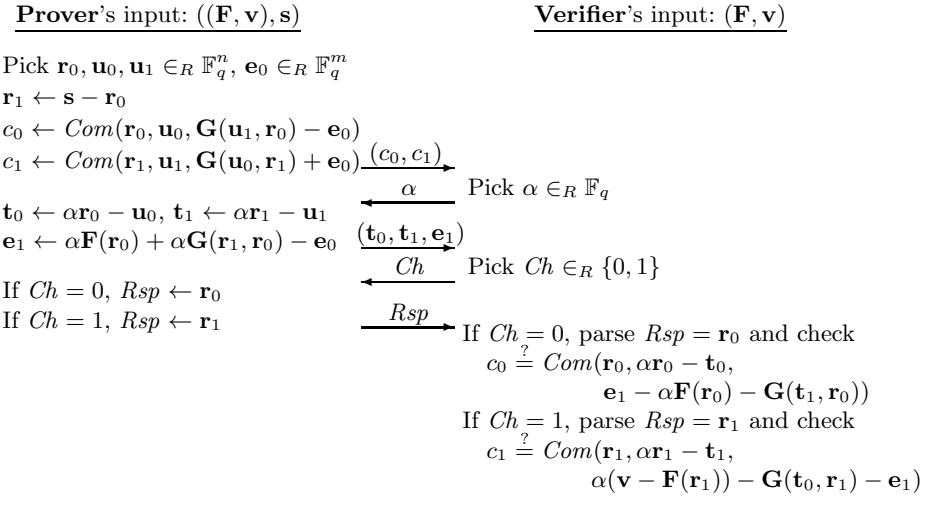


Fig. 3. Our 5-pass protocol

respectively. The difference from the 3-pass protocol is that $\mathbf{r}_0$, $\mathbf{r}_1$, and $\mathbf{F}(\mathbf{r}_0) + \mathbf{G}(\mathbf{r}_1, \mathbf{r}_0)$ are divided as $\alpha\mathbf{r}_0 = \mathbf{t}_0 + \mathbf{u}_0$, $\alpha\mathbf{r}_1 = \mathbf{t}_1 + \mathbf{u}_1$, and $\alpha\mathbf{F}(\mathbf{r}_0) + \alpha\mathbf{G}(\mathbf{r}_1, \mathbf{r}_0) = \mathbf{e}_0 + \mathbf{e}_1$ where $\alpha \in \mathbb{F}_q$ is a choice of a verifier. In particular, we note that $\mathbf{r}_0$ and $\mathbf{r}_1$ are divided by using two independent vectors $\mathbf{u}_0$ and $\mathbf{u}_1$. The reason is that the prover of the 5-pass protocol sends *both* $\mathbf{t}_0$ and $\mathbf{t}_1$, while that of the 3-pass protocol sends *either* $\mathbf{t}_0$ or $\mathbf{t}_1$. After sending $(\mathbf{t}_0, \mathbf{t}_1, \mathbf{e}_1)$ to the verifier, corresponding to a challenge $Ch \in \{0, 1\}$ of the verifier, the prover reveals one out of two vectors $\mathbf{r}_0$ and $\mathbf{r}_1$. When $\mathbf{r}_0$, $\mathbf{u}_0$, $\mathbf{u}_1$, and $\mathbf{e}_0$ are randomly chosen, the verifier can obtain no information on the secret key $\mathbf{s}$ from only one out of the two vectors $\mathbf{r}_0$ and $\mathbf{r}_1$. On the other hand, the argument-of-knowledge property comes from that, for more than one choice of $\alpha \in \mathbb{F}_q$, an impersonator cannot response both of verifier's challenges $Ch = 0$ and $Ch = 1$ unless the impersonator has a solution $\mathbf{s}$ for $\mathbf{v}$.

The 5-pass protocol is described in Figure 3 where $\mathbf{G}$ is the associated LOA form of $\mathbf{F}$. The verifier finally outputs 1 if the check of "$\stackrel{?}{=}$" is passed, otherwise outputs 0. This is denoted by $0/1 \leftarrow Dec(\mathbf{F}, \mathbf{v}; (c_0, c_1), \alpha, (\mathbf{t}_1, \mathbf{e}_1), Ch, Rsp)$. It is easy to see that the verifier always accepts an interaction with the honest prover. Thus the 5-pass protocol has perfect correctness.

Now we show two properties of the protocol in Theorem 6 and Theorem 7 as follows.

Theorem 6. *The 5-pass protocol is statistically zero knowledge when the commitment scheme Com is statistically hiding.*

Proof sketch. Let $\mathcal{S}$ be a simulator which takes $\mathbf{F}$ and $\mathbf{v}$ without knowing $\mathbf{s}$, and interacts with a cheating verifier $\mathcal{CV}$. We show that the simulator $\mathcal{S}$ can impersonate the honest prover with probability 1/2. The simulator $\mathcal{S}$ randomly chooses a value $Ch^* \in_R \{0, 1\}$ and vectors $\mathbf{s}', \mathbf{r}'_0, \mathbf{u}'_0, \mathbf{u}'_1 \in_R \mathbb{F}_q^n$, $\mathbf{e}'_0 \in_R \mathbb{F}_q^m$,

where Ch^* is a prediction of what value the cheating verifier $\mathcal{CV}$ *will* choose. Then, it computes $\mathbf{r}'_1 \leftarrow \mathbf{s}' - \mathbf{r}'_0$, $c'_0 \leftarrow Com(\mathbf{r}'_0, \mathbf{u}'_0, \mathbf{G}(\mathbf{u}'_1, \mathbf{r}'_0) - \mathbf{e}'_0)$, and $c'_1 \leftarrow Com(\mathbf{r}'_1, \mathbf{u}'_1, \mathbf{G}(\mathbf{u}'_0, \mathbf{r}'_1) + \mathbf{e}'_0)$. It sends (c'_0, c'_1) to $\mathcal{CV}$. Receiving a challenge α from $\mathcal{CV}$, it computes $\mathbf{t}'_0 \leftarrow \alpha\mathbf{r}'_0 - \mathbf{u}'_0$ and $\mathbf{t}'_1 \leftarrow \alpha\mathbf{r}'_1 - \mathbf{u}'_1$. If $Ch^* = 0$ then it computes $\mathbf{e}'_1 \leftarrow \alpha\mathbf{F}(\mathbf{r}'_0) + \alpha\mathbf{G}(\mathbf{r}'_1, \mathbf{r}'_0) - \mathbf{e}'_0$, else $\mathbf{e}'_1 \leftarrow \alpha(\mathbf{v} - \mathbf{F}(\mathbf{r}'_1) - \mathbf{G}(\mathbf{r}'_0, \mathbf{r}'_1)) - \mathbf{e}'_0$. It sends $(\mathbf{t}'_0, \mathbf{t}'_1, \mathbf{e}'_1)$ to $\mathcal{CV}$. Due to the statistically hiding property of Com, a challenge Ch from $\mathcal{CV}$ is equal to Ch^* with probability $1/2$. If $Ch = Ch^*$ then $\mathbf{r}'_0$ and $\mathbf{r}'_1$ are accepted responses to $Ch = 0$ and 1, respectively. Note that the case of $\alpha = 0$ does not spoil the zero-knowledge property. □

Theorem 7. *The 5-pass protocol is argument of knowledge for $R_{\mathbf{F}}$ with knowledge error $1/2 + 1/2q$ when the commitment scheme Com is computationally binding.*

Proof sketch. Let $\alpha_0, \alpha_1 \in \mathbb{F}_q$ such that $\alpha_0 \neq \alpha_1$. For $(i, j) \in \{(0,0), (0,1), (1,0), (1,1)\}$, let $((c_0, c_1), \alpha_i, (\tilde{\mathbf{t}}_0^{(i)}, \tilde{\mathbf{t}}_1^{(i)}, \tilde{\mathbf{e}}_1^{(i)}), Ch_j, Rsp^{(i,j)})$ be a transcript such that $Dec(\mathbf{F}, \mathbf{v}; (c_0, c_1), \alpha_i, (\tilde{\mathbf{t}}_0^{(i)}, \tilde{\mathbf{t}}_1^{(i)}, \tilde{\mathbf{e}}_1^{(i)}), Ch_j, Rsp^{(i,j)}) = 1$ and $Ch_j = j$. By using the four transcripts, it is shown to be able to either break the binding property of Com or extract a solution for $\mathbf{v}$. Consider that the responses are parsed as $Rsp^{(0,0)} = \tilde{\mathbf{r}}_0^{(0)}$, $Rsp^{(0,1)} = \tilde{\mathbf{r}}_1^{(0)}$, $Rsp^{(1,0)} = \tilde{\mathbf{r}}_0^{(1)}$, and $Rsp^{(1,1)} = \tilde{\mathbf{r}}_1^{(1)}$. Then, it is seen that

$$\begin{aligned} c_0 &= Com(\tilde{\mathbf{r}}_0^{(0)}, \alpha_0\tilde{\mathbf{r}}_0^{(0)} - \tilde{\mathbf{t}}_0^{(0)}, \tilde{\mathbf{e}}_1^{(0)} - \alpha_0\mathbf{F}(\tilde{\mathbf{r}}_0^{(0)}) - \mathbf{G}(\tilde{\mathbf{t}}_1^{(0)}, \tilde{\mathbf{r}}_0^{(0)})) \\ &= Com(\tilde{\mathbf{r}}_0^{(1)}, \alpha_1\tilde{\mathbf{r}}_0^{(1)} - \tilde{\mathbf{t}}_0^{(1)}, \tilde{\mathbf{e}}_1^{(1)} - \alpha_1\mathbf{F}(\tilde{\mathbf{r}}_0^{(1)}) - \mathbf{G}(\tilde{\mathbf{t}}_1^{(1)}, \tilde{\mathbf{r}}_0^{(1)})) \quad \text{and} \qquad (10) \\ c_1 &= Com(\tilde{\mathbf{r}}_1^{(0)}, \alpha_0\tilde{\mathbf{r}}_1^{(0)} - \tilde{\mathbf{t}}_1^{(0)}, \alpha_0(\mathbf{v} - \mathbf{F}(\tilde{\mathbf{r}}_1^{(0)})) - \mathbf{G}(\tilde{\mathbf{t}}_0^{(0)}, \tilde{\mathbf{r}}_1^{(0)}) - \tilde{\mathbf{e}}_1^{(0)}) \\ &= Com(\tilde{\mathbf{r}}_1^{(1)}, \alpha_1\tilde{\mathbf{r}}_1^{(1)} - \tilde{\mathbf{t}}_1^{(1)}, \alpha_1(\mathbf{v} - \mathbf{F}(\tilde{\mathbf{r}}_1^{(1)})) - \mathbf{G}(\tilde{\mathbf{t}}_0^{(1)}, \tilde{\mathbf{r}}_1^{(1)}) - \tilde{\mathbf{e}}_1^{(1)}). \qquad (11) \end{aligned}$$

If the two tuples of the arguments of Com are distinct on either of the above equations, the binding property of Com is broken. Otherwise, it is seen that $(\alpha_0 - \alpha_1)(\mathbf{v} - \mathbf{F}(\tilde{\mathbf{r}}_1^{(0)})) = \mathbf{G}(\tilde{\mathbf{t}}_0^{(0)} - \tilde{\mathbf{t}}_0^{(1)}, \tilde{\mathbf{r}}_1^{(0)}) + \tilde{\mathbf{e}}_1^{(0)} - \tilde{\mathbf{e}}_1^{(1)}$ and $\tilde{\mathbf{t}}_1^{(0)} - \tilde{\mathbf{t}}_1^{(1)} = (\alpha_0 - \alpha_1)\tilde{\mathbf{r}}_1^{(0)}$ from the equation (11). Combining them with the equation (10) yields $(\alpha_0 - \alpha_1)(\mathbf{v} - \mathbf{F}(\tilde{\mathbf{r}}_1^{(0)})) = \mathbf{G}(\tilde{\mathbf{t}}_0^{(0)} - \tilde{\mathbf{t}}_0^{(1)}, \tilde{\mathbf{r}}_1^{(0)}) + (\alpha_0 - \alpha_1)\mathbf{F}(\tilde{\mathbf{r}}_0^{(0)}) + \mathbf{G}(\tilde{\mathbf{t}}_1^{(0)} - \tilde{\mathbf{t}}_1^{(1)}, \tilde{\mathbf{r}}_0^{(0)}) = (\alpha_0 - \alpha_1)(\mathbf{G}(\tilde{\mathbf{r}}_0^{(1)}, \tilde{\mathbf{r}}_1^{(0)}) + \mathbf{F}(\tilde{\mathbf{r}}_0^{(0)}) + \mathbf{G}(\tilde{\mathbf{r}}_1^{(0)}, \tilde{\mathbf{r}}_0^{(0)}))$. Thus, we obtain $\mathbf{v} = \mathbf{F}(\tilde{\mathbf{r}}_1^{(0)}) + \mathbf{G}(\tilde{\mathbf{r}}_0^{(1)}, \tilde{\mathbf{r}}_1^{(0)}) + \mathbf{G}(\tilde{\mathbf{r}}_1^{(0)}, \tilde{\mathbf{r}}_0^{(0)}) + \mathbf{F}(\tilde{\mathbf{r}}_0^{(0)}) = \mathbf{F}(\tilde{\mathbf{r}}_1^{(0)} + \tilde{\mathbf{r}}_0^{(0)})$, since $\alpha_0 \neq \alpha_1$. It means that a solution $\tilde{\mathbf{r}}_1^{(0)} + \tilde{\mathbf{r}}_0^{(0)}$ for $\mathbf{v}$ is extracted. □

5 Security and Efficiency

This section we summarize the security of our identification schemes which is easily obtained from results in Section 3 and 4, and evaluate the efficiency of our schemes for a practical parameter choice.

5.1 Security

Here we briefly mention the security of each of the sequential and the parallel compositions in the same way as [26]. Let $(\mathtt{P}, \mathtt{V})$ be our 3-pass protocol or 5-pass protocol and ϵ its knowledge error. Let $N = \omega(\log \lambda)$. Then identification protocols which consist of repeating $(\mathtt{P}, \mathtt{V})$ N-times in sequential and in parallel are denoted by $(\mathtt{P}_N^{(s)}, \mathtt{V}_N^{(s)})$ and $(\mathtt{P}_N^{(p)}, \mathtt{V}_N^{(p)})$, respectively. When $\mathcal{MC}(n, m, \mathbb{F}_q)$ is intractable and the commitment scheme *Com* is statistically hiding and computationally binding, the security of our identification schemes $(\mathtt{Setup}, \mathtt{Gen}, \mathtt{P}_N^{(s)}, \mathtt{V}_N^{(s)})$ and $(\mathtt{Setup}, \mathtt{Gen}, \mathtt{P}_N^{(p)}, \mathtt{V}_N^{(p)})$ is evaluated as follows.

The former $(\mathtt{P}_N^{(s)}, \mathtt{V}_N^{(s)})$ is statistically zero-knowledge argument of knowledge with knowledge error ϵ^N due to the sequential composition lemma [15] and Stern's proof techniques of [29, 30]. Thus the identification scheme $(\mathtt{Setup}, \mathtt{Gen}, \mathtt{P}_N^{(s)}, \mathtt{V}_N^{(s)})$ is secure against impersonation under *active* attack. On the other hand, the parallel repetition of $(\mathtt{P}, \mathtt{V})$ reserves zero-knowledge with respect to an *honest verifier*. By combining it with Pass and Venkitasubramaniam's result [20], the latter $(\mathtt{P}_N^{(p)}, \mathtt{V}_N^{(p)})$ is also honest-verifier zero-knowledge argument of knowledge with a negligible knowledge error. Therefore, the identification scheme $(\mathtt{Setup}, \mathtt{Gen}, \mathtt{P}_N^{(p)}, \mathtt{V}_N^{(p)})$ is secure against impersonation under *passive* attack. In addition, for a certain parameter choice, the parallel version of our scheme is also secure under *active* attack as shown in Section 6.

5.2 Efficiency

The efficiency of the schemes consisting of our 5-pass protocol is highly comparable to that of the schemes based on binary SD, q-ary SD, CLE, PP, PK, and MQ, even though our 3-pass protocol is not so efficient. Here we evaluate the data sizes of system parameters, a public key, a secret key, and a transcript of our schemes. The numbers of arithmetic operations, computing permutations, and computing hash functions are also estimated as computational cost. These are evaluated according to [9, 26]. In this paper $\mathcal{MC}(84, 80, \mathbb{F}_2)$ and $\mathcal{MC}(33, 22, \mathbb{F}_{2^4})$ are used for our 3-pass protocol and for our 5-pass one, respectively.

First, we consider the schemes consisting of each of the 3-pass protocols. Table 1 compares our scheme with the schemes based on binary SD, CLE, PP, and MQ when each protocol is sequentially repeated until impersonation probability is less than 2^{-30}. In this comparison we consider the case where each scheme uses techniques for saving the communication data size such as the trick mentioned at the end of Section 3.

Second, consider the 5-pass protocols. Table 2 compares our scheme with the schemes based on binary SD, q-ary SD, CLE, PK, PP, and MQ when each protocol is sequentially repeated until impersonation probability is less than 2^{-30}. The data sizes of a public key and a secret key of our scheme are smaller than those of the other schemes. The communication data size is almost the smallest in Table 2. Although the size of system parameter of our scheme is relatively large, it can be reduced to some small seed, e.g. 128 bits, if a pseudo-random

Table 1. Comparison of 3-pass schemes on 80-bit security against key-recovery attack when the impersonation probability is less than 2^{-30}

	SD [30]	CLE [29]	PP [25]	MQ [26]	Our
round	52	52	73	52	73
system parameter (bit)	122,500	4,608	28,497	285,600	7,908,320
public key (bit)	350	288	245	80	80
secret key (bit)	700	192	177	84	84
communication (bit)	59,800	45,517	100,925	29,640	53,290
arithmetic ops. (times/field)	2^{24} / $\mathbb{F}_2$	2^{16} / $\mathbb{F}_{257}$	2^{22} / $\mathbb{F}_{127}$	2^{26} / $\mathbb{F}_2$	2^{32} / $\mathbb{F}_2$
permutations[*1] (times/size)	$2/S_{700}$	$4/S_{24}$	$2/S_{161}, S_{177}$	NO	NO
hash function (times)	4	4	8	4	8
best known key-recovery attack	2^{87}	2^{84}	$> 2^{74}$	2^{80}	2^{80}

Table 2. Comparison of 5-pass schemes on 80-bit security against key-recovery attack when the impersonation probability is less than 2^{-30}

	SD [30]	SD [9]	PK [27]	CLE [29]	PP [24, 25]	MQ [26]	Our
round	31	31	31	31	52	33	33
system parameter (bit)	122,500	32,768	4,608	4,608	28,497	129,600[*2]	581,768
public key (bit)	2450	512	384	288	245	120	88
secret key (bit)	4900	1024	203	192	177	180	132
communication (bit)	120,652	61,783	27,234	27,528	105,060	26,565	26,697
arithmetic ops. (times/field)	$2^{23}/\mathbb{F}_2$	$2^{18}/\mathbb{F}_{256}$	$2^{15}/\mathbb{F}_{251}$	$2^{15}/\mathbb{F}_{257}$	$2^{21}/\mathbb{F}_{127}$	$2^{22}/\mathbb{F}_{2^4}$	$2^{27}/\mathbb{F}_{2^4}$
permutations[*1] (times/size)	$8/S_{700}$	$2/S_{128}$	$3/S_{48}$	$4/S_{24}$	$2/S_{161}, S_{177}$	NO	NO
hash function (times)	2	2	2	2	5	2	2
best known key-recovery attack	2^{87}	2^{87}	2^{85}	2^{84}	$> 2^{74}$	2^{83}	2^{81}

[*1] This shows the number of times of computing permutations and the size of the permutation, where S_n means a permutation over $\{1, \ldots, n\}$.

[*2] This is the correct size of the system parameters, although it is stated as 259,200 bits in the original paper [26].

number generator is used as the implementation of QUAD [2]. Although the cost of arithmetic operations of our scheme is relatively high, it is still reasonable.

6 On the Security against Active Attack in Parallel Repetition

In this section we focus on the case of $n = m + k$ and $k = \omega(\log \lambda)$. For example, the MC function $\mathbf{F} \in \mathcal{MC}(2m, m, \mathbb{F}_q)$ satisfies the requirement where $m = \omega(\log \lambda)$. In this case, $(\mathtt{Setup}, \mathtt{Gen}, \mathtt{P}_N^{(p)}, \mathtt{V}_N^{(p)})$ is shown to be secure against impersonation under *active* attack, although the data sizes of the secret key and the communication increase at most double compared to those of Section 5.2. The security can be shown in almost the same way as that of the MQ-based scheme. Although we consider the scheme consisting of our 3-pass protocol in this section, the same argument can also be applied to that of our 5-pass protocol.

We begin with defining the preimage resistance and the second-preimage resistance of the MC function. Note that the difference between the preimage

resistance and the intractability of Definition 3 is only in the distribution of the challenge $\mathbf{v}$, and both of them are widely believed.

Definition 8. *For polynomially bounded functions $n = n(\lambda)$, $m = m(\lambda)$, and $q = q(\lambda)$, it is said that $\mathcal{MC}(n, m, \mathbb{F}_q)$ is preimage resistant if there is no polynomial-time algorithm that takes $(\mathbf{F}, \mathbf{v})$ generated via $\mathbf{F} \in_R \mathcal{MC}(n, m, \mathbb{F}_q)$ and $\mathbf{v} \in_R \mathbb{F}_q^m$ and finds a preimage $\mathbf{s} \in \mathbb{F}_q^n$ such that $\mathbf{F}(\mathbf{s}) = \mathbf{v}$ with non-negligible probability $\epsilon(\lambda)$. On the other hand, it is said that $\mathcal{MC}(n, m, \mathbb{F}_q)$ is second-preimage resistant if there is no polynomial-time algorithm that takes $(\mathbf{F}, \mathbf{x})$ generated via $\mathbf{F} \in_R \mathcal{MC}(n, m, \mathbb{F}_q)$ and $\mathbf{x} \in_R \mathbb{F}_q^n$ and finds a second preimage $\mathbf{x}' \in \mathbb{F}_q^n$ such that $\mathbf{F}(\mathbf{x}') = \mathbf{F}(\mathbf{x})$ and $\mathbf{x}' \neq \mathbf{x}$ with non-negligible probability $\epsilon(\lambda)$.*

Then we present the following lemma.

Lemma 9. *If there exists an algorithm that breaks the second-preimage resistance of $\mathcal{MC}(n+1, m, \mathbb{F}_q)$ with advantage ϵ, then there exists an algorithm that breaks the preimage resistance of $\mathcal{MC}(n, m, \mathbb{F}_q)$ with advantage $\epsilon/(q-1)(n+1)$. That is, if $\mathcal{MC}(n, m, \mathbb{F}_q)$ is preimage resistant, then $\mathcal{MC}(n+1, m, \mathbb{F}_q)$ is second-preimage resistant.*

Proof sketch. Let $\mathcal{A}$ be an algorithm that breaks the second-preimage resistance of $\mathcal{MC}(n+1, m, \mathbb{F}_q)$. Let $\mathbf{F} = (f_1, \ldots, f_m) \in_R \mathcal{MC}(n, m, \mathbb{F}_q)$ and $\mathbf{v} = (v_1, \ldots, v_m) \in_R \mathbb{F}_q^m$, where $f_l(x_1, \ldots, x_n) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} a_{l,i,j,k}x_ix_jx_k + \sum_{i=1}^{n}\sum_{j=1}^{n} b_{l,i,j}x_ix_j + \sum_{i=1}^{n} c_{l,i}x_i$. We show that, given $\mathbf{F}$ and $\mathbf{v}$, a preimage $\mathbf{x}$ satisfying $\mathbf{v} = \mathbf{F}(\mathbf{x})$ can be found by using the algorithm $\mathcal{A}$. For the simplicity, suppose that the algorithm $\mathcal{A}$ takes $\tilde{\mathbf{F}} = (\tilde{f}_1, \ldots, \tilde{f}_m) \in \mathcal{MC}(n+1, m, \mathbb{F}_q)$ and $\mathbf{t} = (t_1, \ldots, t_{n+1}) \in \mathbb{F}_q^{n+1}$ and outputs a second preimage $\mathbf{t} + \Delta$ such that $\tilde{\mathbf{F}}(\mathbf{t} + \Delta) = \tilde{\mathbf{F}}(\mathbf{t})$ and $\Delta = (d_1, \ldots, d_n, 1)$, where $\tilde{f}_l(x_1, \ldots, x_{n+1}) = \sum_{i=1}^{n+1}\sum_{j=1}^{n+1}\sum_{k=1}^{n+1} \tilde{a}_{l,i,j,k}x_ix_jx_k + \sum_{i=1}^{n+1}\sum_{j=1}^{n+1} \tilde{b}_{l,i,j}x_ix_j + \sum_{i=1}^{n+1} \tilde{c}_{l,i}x_i$. Note that in the full proof it is necessary to guess an index ξ and a value d_ξ of a non-zero element in Δ, but in this proof sketch we suppose $\xi = n+1$ and $d_\xi = 1$. In this case, the equation $\tilde{\mathbf{F}}(\mathbf{t} + \Delta) - \tilde{\mathbf{F}}(\mathbf{t}) = \mathbf{0}$ is expanded as follows:

$$
\begin{aligned}
&\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} \tilde{a}_{l,i,j,k}d_id_jd_k \\
&+ \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\begin{array}{l}\displaystyle\sum_{k=1}^{n+1}(\tilde{a}_{l,k,j,i} + \tilde{a}_{l,i,k,j} + \tilde{a}_{l,j,i,k})t_k \\ \qquad +\tilde{b}_{l,i,j} + \tilde{a}_{l,i,j,n+1} + \tilde{a}_{l,n+1,i,j} + \tilde{a}_{l,i,n+1,j}\end{array}\right) d_id_j \\
&+ \sum_{i=1}^{n}\left(\begin{array}{l}\displaystyle\sum_{j=1}^{n+1}\sum_{k=1}^{n+1}(\tilde{a}_{l,k,j,i} + \tilde{a}_{l,j,i,k} + \tilde{a}_{l,i,k,j})t_jt_k \\ \displaystyle+\sum_{k=1}^{n+1}\left(\begin{array}{l}\tilde{a}_{l,k,i,n+1} + \tilde{a}_{l,n+1,k,i} + \tilde{a}_{l,n+1,i,k} \\ +\tilde{a}_{l,k,n+1,i} + \tilde{a}_{l,i,k,n+1} + \tilde{a}_{l,i,n+1,k} + \tilde{b}_{l,k,i} + \tilde{b}_{l,i,k}\end{array}\right)t_k \\ +\tilde{a}_{l,n+1,i,n+1} + \tilde{a}_{l,i,n+1,n+1} + \tilde{a}_{l,n+1,n+1,i} + \tilde{b}_{l,n+1,i} + \tilde{b}_{l,i,n+1} + \tilde{c}_{l,i}\end{array}\right) d_i
\end{aligned}
$$

$$+\left(\begin{array}{l}\sum_{j=1}^{n+1}\sum_{k=1}^{n+1}(\tilde{a}_{l,k,j,n+1}+\tilde{a}_{l,j,n+1,k}+\tilde{a}_{l,n+1,k,j})t_jt_k\\ +\sum_{k=1}^{n+1}(\tilde{a}_{l,k,n+1,n+1}+\tilde{a}_{l,n+1,k,n+1}+\tilde{a}_{l,n+1,n+1,k}+\tilde{b}_{l,k,n+1}+\tilde{b}_{l,n+1,k})t_k\\ +\tilde{a}_{l,n+1,n+1,n+1}+\tilde{b}_{l,n+1,n+1}+\tilde{c}_{l,n+1}\end{array}\right)=0$$

for $l = 1, \ldots, m$. From the above equation, we can see that the output $\mathbf{t} + \Delta$ of $\mathcal{A}$ satisfies $\mathbf{v} = \mathbf{F}(d_1, \ldots, d_n)$ if the input $(\tilde{\mathbf{F}}, \mathbf{t})$ of $\mathcal{A}$ is produced as follows.

- The vector $\mathbf{t}$ is generated via $\mathbf{t} \in_R \mathbb{F}_q^{n+1}$.
- For $1 \le i, j, k \le n$ do $\tilde{a}_{l,i,j,k} \leftarrow a_{l,i,j,k}$, otherwise $\tilde{a}_{l,i,j,k} \in_R \mathbb{F}_q$.
- For $1 \le i, j \le n$ do $\tilde{b}_{l,i,j} \leftarrow b_{l,i,j} - \sum_{k=1}^{n+1}(\tilde{a}_{l,k,j,i} + \tilde{a}_{l,i,k,j} + \tilde{a}_{l,j,i,k})t_k - (\tilde{a}_{l,i,j,n+1} + \tilde{a}_{l,n+1,i,j} + \tilde{a}_{l,i,n+1,j})$, otherwise $\tilde{b}_{l,i,j} \in_R \mathbb{F}_q$.
- For $1 \le i \le n$ do $\tilde{c}_{l,i} \leftarrow c_{l,i} - \sum_{j=1}^{n+1}\sum_{k=1}^{n+1}(\tilde{a}_{l,k,j,i} + \tilde{a}_{l,j,i,k} + \tilde{a}_{l,i,k,j})t_jt_k - \sum_{k=1}^{n+1}(\tilde{a}_{l,k,i,n+1} + \tilde{a}_{l,n+1,k,i} + \tilde{a}_{l,n+1,i,k} + \tilde{a}_{l,k,n+1,i} + \tilde{a}_{l,i,k,n+1} + \tilde{a}_{l,i,n+1,k} + \tilde{b}_{l,k,i} + \tilde{b}_{l,i,k})t_k - (\tilde{a}_{l,n+1,i,n+1} + \tilde{a}_{l,i,n+1,n+1} + \tilde{a}_{l,n+1,n+1,i} + \tilde{b}_{l,n+1,i} + \tilde{b}_{l,i,n+1})$.
- $\tilde{c}_{l,n+1} \leftarrow -v_l - \sum_{j=1}^{n+1}\sum_{k=1}^{n+1}(\tilde{a}_{l,k,j,n+1} + \tilde{a}_{l,j,n+1,k} + \tilde{a}_{l,n+1,k,j})t_jt_k - \sum_{k=1}^{n+1}(\tilde{a}_{l,k,n+1,n+1} + \tilde{a}_{l,n+1,k,n+1} + \tilde{a}_{l,n+1,n+1,k} + \tilde{b}_{l,k,n+1} + \tilde{b}_{l,n+1,k})t_k - (\tilde{a}_{l,n+1,n+1,n+1} + \tilde{b}_{l,n+1,n+1})$.

The details of the proof of Lemma 9 are described in the full paper. □

Moreover, the following lemma is shown.

Lemma 10. *Let $n = m + k$, $k = \omega(\log \lambda)$, and $N = \omega(\log \lambda)$. Suppose that $\mathcal{MC}(n, m, \mathbb{F}_q)$ is second-preimage resistant. Then, $(\mathtt{P}_N^{(p)}, \mathtt{V}_N^{(p)})$ achieves the security against impersonation under* active *attack when Com is statistically hiding and computationally binding.*

Proof sketch. The proof of this lemma is described in the full paper, since it is similar to that of Lemma 8 of [26]. □

Finally, combining Lemma 9 and Lemma 10 yields the following theorem.

Theorem 11. *Let $n = m + k$, $k = \omega(\log \lambda)$, and $N = \omega(\log \lambda)$. Suppose that $\mathcal{MC}(n - 1, m, \mathbb{F}_q)$ is preimage resistant. Then, $(\mathtt{P}_N^{(p)}, \mathtt{V}_N^{(p)})$ achieves the security against impersonation under* active *attack when Com is statistically hiding and computationally binding.*

7 Concluding Remarks

In this section we mention some extensions and an open problem.

Extensions. The Fiat-Shamir method transforms an identification scheme into a signature scheme which is secure against chosen-message attack in the random oracle model, if the underlying identification scheme is secure against impersonation under passive attack [1, 13]. According to it, a signature scheme based on the conjectured intractability of the MC problem can be obtained from the parallel composition of our 3-pass protocol. Using the signature scheme, we can also extend our identification/signature scheme to an identity-based one in a natural way [4].

An open problem. Efficient constructions based on multivariate polynomials of degree $d \geq 4$ remain as an open problem. However, it might be difficult to construct them by using techniques similar to those of [26] or of ours. This is because, for a multivariate polynomial $f(\mathbf{x})$ of degree $d \geq 4$, the polynomial $f(\mathbf{x}+\mathbf{y}) - f(\mathbf{x}) - f(\mathbf{y})$ contains terms which are not linear either in $\mathbf{x}$ or in $\mathbf{y}$.

8 Conclusion

We proposed an efficient construction of zero-knowledge argument of knowledge for the MC problem, and showed that the MC function is useful for public-key identification as well as the MQ function. In particular the efficiency of our scheme is highly comparable to the identification schemes based on another problem including PK, SD, CLE, PP, and MQ.

Acknowledgements. We thank Taizo Shirai and Harunaga Hiwatari for their generous support, and Marc Fischlin and the anonymous reviewers for useful comments.

References

1. Abdalla, M., An, J.H., Bellare, M., Namprempre, C.: From Identification to Signatures via the Fiat-Shamir Transform: Minimizing Assumptions for Security and Forward-Security. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 418–433. Springer, Heidelberg (2002)
2. Arditti, D., Berbain, C., Billet, O., Gilbert, H.: Compact FPGA Implementations of QUAD. In: Bao, F., Miller, S. (eds.) ASIACCS, pp. 347–349. ACM (2007)
3. Bardet, M., Faugère, J.-C., Salvy, B.: Complexity of Gröbner Basis Computation for Semi-regular Overdetermined Sequences over F_2 with Solutions in F_2. Research Report RR-5049, INRIA (2003)
4. Bellare, M., Namprempre, C., Neven, G.: Security Proofs for Identity-Based Identification and Signature Schemes. J. Cryptology 22(1), 1–61 (2009)
5. Berbain, C., Gilbert, H., Patarin, J.: QUAD: A Practical Stream Cipher with Provable Security. In: Vaudenay (ed.) [31], pp. 109–128

6. Bettale, L., Faugère, J.-C., Perret, L.: Security Analysis of Multivariate Polynomials for Hashing. In: Yung, M., Liu, P., Lin, D. (eds.) Inscrypt 2008. LNCS, vol. 5487, pp. 115–124. Springer, Heidelberg (2009)
7. Bettale, L., Faugère, J.-C., Perret, L.: Hybrid Approach for Solving Multivariate Systems over Finite Fields. Journal of Mathematical Cryptology 3(3), 177–197 (2009)
8. Bouillaguet, C., Chen, H.-C., Cheng, C.-M., Chou, T., Niederhagen, R., Shamir, A., Yang, B.-Y.: Fast Exhaustive Search for Polynomial Systems in F_2. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 203–218. Springer, Heidelberg (2010)
9. Cayrel, P.-L., Véron, P., El Yousfi Alaoui, S.M.: A Zero-Knowledge Identification Scheme Based on the q-ary Syndrome Decoding Problem. In: Biryukov, A., Gong, G., Stinson, D.R. (eds.) SAC 2010. LNCS, vol. 6544, pp. 171–186. Springer, Heidelberg (2011)
10. Ding, J., Yang, B.-Y.: Multivariates Polynomials for Hashing. In: Pei, D., Yung, M., Lin, D., Wu, C. (eds.) Inscrypt 2007. LNCS, vol. 4990, pp. 358–371. Springer, Heidelberg (2008)
11. Faugère, J.C.: A New Efficient Algorithm for Computing Gröbner Bases without Reduction to Zero (F5). In: Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation, ISSAC 2002, pp. 75–83. ACM, New York (2002)
12. Faugère, J.-C., Perret, L.: Cryptanalysis of 2R$^-$ Schemes. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 357–372. Springer, Heidelberg (2006)
13. Fiat, A., Shamir, A.: How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987)
14. Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness. W.H. Freeman & Co., New York (1979)
15. Goldreich, O.: Foundations of Cryptography. Basic Tools, vol. I. Cambridge University Press, Cambridge (2001)
16. Han, Y., Okamoto, T., Qing, S. (eds.): ICICS 1997. LNCS, vol. 1334. Springer, Heidelberg (1997)
17. Johnson, D.S., Feige, U. (eds.): Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13. ACM (2007)
18. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced Oil and Vinegar Signature Schemes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 206–222. Springer, Heidelberg (1999)
19. Matsumoto, T., Imai, H.: Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption. In: Günther, C.G. (ed.) EUROCRYPT 1988. LNCS, vol. 330, pp. 419–453. Springer, Heidelberg (1988)
20. Pass, R., Venkitasubramaniam, M.: An Efficient Parallel Repetition Theorem for Arthur-Merlin Games. In: Johnson, Feige (eds.) [17], pp. 420–429
21. Patarin, J.: Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In: Maurer, U. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 33–48. Springer, Heidelberg (1996)
22. Patarin, J., Goubin, L.: Asymmetric Cryptography with S-Boxes. In: Han, et al. (eds.) [16], pp. 369–380
23. Patarin, J., Goubin, L.: Trapdoor One-Way Permutations and Multi- variate Polynominals. In: Han, et al. (eds.) [16], pp. 356–368

24. Pointcheval, D.: A New Identification Scheme Based on the Perceptrons Problem. In: Guillou, L.C., Quisquater, J.-J. (eds.) EUROCRYPT 1995. LNCS, vol. 921, pp. 319–328. Springer, Heidelberg (1995)
25. Pointcheval, D., Poupard, G.: A New NP-Complete Problem and Public-key Identification. Des. Codes Cryptography 28(1), 5–31 (2003)
26. Sakumoto, K., Shirai, T., Hiwatari, H.: Public-Key Identification Schemes Based on Multivariate Quadratic Polynomials. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 706–723. Springer, Heidelberg (2011)
27. Shamir, A.: An Efficient Identification Scheme Based on Permuted Kernels (Extended Abstract). In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 606–609. Springer, Heidelberg (1990)
28. Stern, J.: A New Identification Scheme Based on Syndrome Decoding. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 13–21. Springer, Heidelberg (1994)
29. Stern, J.: Designing Identification Schemes with Keys of Short Size. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 164–173. Springer, Heidelberg (1994)
30. Stern, J.: A New Paradigm for Public Key Identification. IEEE Transactions on Information Theory, 13–21 (1996)
31. Vaudenay, S. (ed.): EUROCRYPT 2006. LNCS, vol. 4004. Springer, Heidelberg (2006)
32. Ye, D.-F., Lam, K.-Y., Dai, Z.-D.: Cryptanalysis of "2R" Schemes. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 315–325. Springer, Heidelberg (1999)

Public-Key Cryptography from New Multivariate Quadratic Assumptions

Yun-Ju Huang[1,3], Feng-Hao Liu[2], and Bo-Yin Yang[3]

[1] Faculty of Mathematics, Kyushu University, Japan
[2] Computer Science, Brown University, USA
[3] Institute of Information Science, Academia Sinica, Taiwan

Abstract. In this work, we study a new multivariate quadratic (MQ) assumption that can be used to construct public-key encryptions. In particular, we research in the following two directions:

- We establish a precise *asymptotic* formulation of a family of hard MQ problems, and provide empirical evidence to confirm the hardness.
- We construct public-key encryption schemes, and prove their security under the hardness assumption of this family. Also, we provide a new *perspective* to look at MQ systems that plays a key role to our design and proof of security.

As a consequence, we construct the *first* public-key encryption scheme that is *provably secure* under the MQ assumption. Moreover, our public-key encryption scheme is efficient in the sense that it only needs a ciphertext length $L + \text{poly}(k)$ to encrypt a message $M \in \{0,1\}^L$ for any un-prespecified polynomial L, where k is the security parameter. This is essentially *optimal* since an additive overhead is the best we can hope for.

1 Introduction

Exploring different types of assumptions has been an important direction in the agenda of cryptography research. For robustness, this reduces the risk of a new mathematical/algorithmic/hardware breakthrough that breaks a particular assumption and renders all its following constructions insecure; for versatility, different assumptions usually have advantages for different applications. However, over the past 30 years, only a few candidates of computational problems are built as foundations on which more exciting cryptographic applications can build; for example, some well-structured algebraic, coding, or geometric problems (and their variants): DDH[17], Pairing (some are instantiated by elliptic curves)[10], RSA[46], McEliece[38], LWE [1, 43, 45], and some recent works for combinatorial problems[2].

This work is in a step of this agenda. We study a new type of assumption inspired from the field of solving multivariate quadratic (MQ) equations. In particular, we give the first asymptotic formulation of a family of MQ problems that enjoy some good mathematical structures and hardness. Thus one can use

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 190–205, 2012.

this formulation as a base to construct more interesting crypto primitives, such as public-key encryption schemes. Our assumption considers a family of problems that can be viewed as solving MQ equations described as the followings (informally) :

Definition 1 (The Hard Task (Informal)). *Let $\mathbb{F}_q$ be a finite field, and H be some subset of $\mathbb{F}_q$. Let S be a multivariate quadratic system with n variables and m polynomials whose coefficients are sampled from some distribution χ.*

Then a solver A, given $(S, \boldsymbol{y} = S(\boldsymbol{x}))$ where $\boldsymbol{x}$ is sampled uniformly from H^n, is asked to output some $\boldsymbol{x}'$ such that $S(\boldsymbol{x}') = \boldsymbol{y}$.

Actually, solving systems of non-linear equations is not a new topic, for it has been studied in commutative algebra and algebraic geometry, at least since Francis Sowerby Macaulay [36] (1902). Around the turn of the millennium, these techniques [14] were also found that they can be used as a cryptanalytic step. Claims (e.g. **XSL** [15]) concerning such techniques, today called "algebraic cryptanalysis", were often over-optimistic, but equation-solvers over different finite fields such as **XL** [14], **F4, F5** [23, 24] are now significant topics for crypto.

The fundamental reason that algebraic cryptanalysis is not all-powerful is that solving systems of non-linear equations does not scale well with the parameters even with Moore's Law. Theoretically, solving multivariate non-linear systems, or even just multivariate quadratic (MQ) equations has been proven to be NP-hard [25, 41] in the worst case, and practically, all the proposed solvers fail to solve the systems efficiently (i.e. in polynomial-time) for *most* non-trivial distributions [4, 35].

The above approach hints at inherent hardness in solving MQ equations, and consequently MQ could be a good choice as a base for designing crypto systems. Although this direction in fact has been considered for the last 20 years, however, it has had a rocky history. Many schemes were proposed, broken, sometimes patched, and sometimes broken again (see [18, 20, 21, 37, 39, 42], and [5, 6, 12, 40]). One objection frequently voiced is that the security of these systems is often ad-hoc, and thus hard to evaluate. Fundamentally, these approaches mostly were designed with a practical goal in mind. As a result, they considered concrete and fixed-parameter constructions, with a design security of, e.g., 2^{80}, with specialization to signatures with 160-bit hashes and optimizing for speed. Since MQ was examined not as a hardness basis but only as the most obvious attack or even some sanity check, the designers' mindsets were not focusing on how to construct a reduction for their security proof, nor about extending their schemes in an asymptotic way. Thus, it seems that using the hardness to construct crypto construction remains an interesting open direction.

Berbain, Gilbert, and Patarin [4] explored this and constructed efficient pseudorandom generators (PRGs) based on the hardness of solving MQ equations. Berbain *et al.* considered fixed and concrete-parameter constructions, yet an asymptotic formulation of hard problems is implicit in their work. Consequently, many primitives such as pseudorandom functions (PRFs), symmetric encryptions, etc., in the Minicrypt world (i.e., one way functions exist) [33] can be constructed based on this formulation of hard problems. For the more sophisticated

Cryptomania world (i.e., public-key crypto systems exist) [33], the possibilities have not yet been explored in the MQ literature. This line of research will be our main focus in the rest of this paper.

Our Main Results. In this work, we study a new MQ assumption that can be used to construct more sophisticated primitives such as public-key encryptions in the Cryptomania world [33]. In particular, we research in the following two directions:

- On the one hand, we establish a precise *asymptotic* formulation of a family of hard problems, and provide empirical evidence to confirm the hardness. Since there are many practical solvers studied and implemented during the studies of algebraic attacks, we use these to examine the hardness of the problems.
- On the other hand, we construct public-key encryption schemes, and prove their security under the hardness assumption of the said family. Also, we provide a new *perspective* to look at MQ systems that plays a key role to our design and proof of security.

As a consequence, we construct the first public-key encryption scheme that is *provably secure* under the MQ assumption. Moreover, our public-key encryption scheme is efficient in the sense that it only needs a ciphertext length $L + \mathrm{poly}(k)$ to encrypt a message $M \in \{0,1\}^L$ for any un-prespecified polynomial L.[1] This is essentially *optimal* since an additive overhead is the best we can hope for.

The MQ assumption has some interesting properties for its potential. In the following, we will discuss that the MQ problems share some structures with the learning with error (LWE) problems [26, 44, 45]. Thus the MQ assumption may also enjoys the versatility as LWE. On the other hand, there are many experiences or fast implementations under a variety of hardwares [4, 9, 11] in the MQ literature, and thus this can be a good basis for practical applications.

Note: we are unaware of any reductions between our MQ assumption or indeed any MQ-type assumptions and lattice-related ones such as LWE. Furthermore, lattice problems have been studied for a much shorter period of time than equation-solving, and new methods such as BKZ 2.0 [13] are still proposed. So it is difficult to compare PKC constructions based on lattice-related hard problems and MQ problems. The comparison is a very interesting research direction but outside the scope of this paper. This paper will simply focus on the MQ assumption and its consequent constructions. In section 3.1, we give a brief remark on the difference between MQ and LWE assumptions. More detailed discussions will appear in the full version of this paper.

A Closer Look at Our Assumption. In the following, we take a closer look at our assumption and techniques, and still maintain a high-level perspective for intuitions. First, we give some notation for convenience of exposition. Let $\mathbb{F}_q$ be a field which we use in the following discussion, and let S describe a multivariate

[1] k is the security parameter.

quadratic system with n variables and m polynomials. For example, the following system is one with 3 variables and 2 polynomials, and for a concrete explanation we set $q = 13$.

$$S \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \stackrel{\text{def}}{=} \begin{cases} x_1x_3 + x_2^2 + 3x_1 + 2 \\ x_1x_2 + 2x_1 + 2x_2 + 7 \end{cases} \tag{1}$$

In addition to viewing S as a set of polynomials, we can view the above system S as a function mapping from $\mathbb{F}_q^3$ to $\mathbb{F}_q^2$. For example, $S([1,2,3]^T) = [12,2]^T$, where T denotes transposes of vectors. In the rest of the paper, we use $S[\cdot]$ to denote a system of polynomials, and $S(\cdot)$ to denote the corresponding function. Now we are ready to describe the hard problem of our assumption with more details (still informally). Note that here the system S includes quadratic terms, linear terms and constant terms. Throughout the paper, we will use S to denote a system with all quadratic, linear and constant terms.

Definition 2 (The Hard Task (Informal)). *Let q be a large enough prime, and H be some small subset of $\mathbb{F}_q$. Let S be a multivariate quadratic system with n variables and $m = \Theta(n)$ polynomials sampled from a distribution where the coefficients of linear and constant terms are uniformly random, and the quadratic terms come from independent Gaussian distributions with means 0 and moderately large standard deviations.*

Then a solver A, given $(S, \boldsymbol{y} = S(\boldsymbol{x}))$ where $\boldsymbol{x}$ is sampled uniformly from H^n, is asked to output some $\boldsymbol{x}'$ such that $S(\boldsymbol{x}') = \boldsymbol{y}$.

To make the seemingly intimidating parameters more reader-friendly, we give an intuitive-level discussion as follows. First, we observe that depending on the parameters, solving MQ equations can be easy or hard. As discussed in [4], when m is significantly larger or smaller than n, solving the problem is easy. The interesting hard instances fall on the cases when m is close to n, as stated in the above definition that $m = \Theta(n)$. Moreover, the problem is believed to be not only hard in the worst case, but hard on average over random instance of S, and random input $\boldsymbol{x}$. Under a series of empirical studies and theoretical studies [3, 16, 47, 48] for the best known solvers, the best known algorithms still remain exponential-time.

Previously, [35] observed (from experiments) that even if the instance S is drawn from a biased distribution (whose quadratic coefficients are not uniform but instead sparse), solving the problem is still hard. This result hints at an intuition that MQ problems are hard for most (non-trivial) distributions from which S is drawn. In this work, we further test this intuition by investigating the case that the instance S is drawn from a distribution whose quadratic coefficients come from Gaussian distributions with moderately large standard deviation, and the input $\boldsymbol{x}$ is drawn from a smaller subset H^n. Our experiment results (in the full version of this paper) confirm our intuition that the problem does not become significantly easier. In the following paragraphs, we explain how and why this type of assumption and hardness help our design.

We remark that here we only give a structural description of the problem, and leave the precise quantitative statement in Section 3. Before going to the detailed calculation of numbers, we first focus on the structural properties of the hard problem and maintain a high-level perspective.

Overview of Our Construction. Inspired by the recent constructions of public-key crypto systems by learning with error (LWE) problems [45], we observe that the problem in Definition 2 also shares the same structure with LWE. We can take advantage of this similarity for our construction of public-key encryption schemes. This is a new perspective of how we can view MQ equations.

First, let us take a look at the LWE problem, which can be stated as the following: let $A \in \mathbb{F}_q^{m \times n}$ be a matrix, and $\boldsymbol{b}$ be a vector $\boldsymbol{b} = A \cdot \boldsymbol{s} + \boldsymbol{e}$, where $\boldsymbol{s} \in \mathbb{F}_q^n$ is some secret, and $\boldsymbol{e}$ comes from some error distribution. The task of the LWE problem is to find out $\boldsymbol{s}$ given a random A, and an induced $\boldsymbol{b}$.

We highlight the similarity by way of the following observation: recall that the task of the problem in Definition 2 is to invert $\boldsymbol{y} = S(\boldsymbol{x})$ given $S, \boldsymbol{y}$. We can rewrite $\boldsymbol{y}$ into $S(\boldsymbol{x}) = L \cdot \boldsymbol{x} + \boldsymbol{d} + R(\boldsymbol{x})$, where L is the matrix of the terms of linear coefficients, $\boldsymbol{d}$ is the coefficient vector of constant terms, and $R(\boldsymbol{x})$ are the mapping by the quadratic terms. Take Equation 1 for example, we can rewrite the expression of $S(\boldsymbol{x})$ as:

$$S(\boldsymbol{x}) = \begin{Bmatrix} x_1x_3 + x_2^2 \\ x_1x_2 \end{Bmatrix} + \begin{matrix} 3x_1 \\ 2x_1 + 2x_2 \end{matrix} + \begin{matrix} 2 \\ 7 \end{matrix} = R(\boldsymbol{x}) + \begin{pmatrix} 3 & 0 & 0 \\ 2 & 2 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 2 \\ 7 \end{pmatrix}$$

In this expression, $S(\boldsymbol{x})$ is a combination of an affine transformation $(L \cdot \boldsymbol{x} + \boldsymbol{d})$ plus some quadratic mapping $R(\boldsymbol{x})$. We remark that without loss of generality, we can assume $\boldsymbol{d} = 0$, since solving a multivariate system with all 0s for the constant coefficients is equivalent to solving that with random constant coefficients.[2] Then if we view the quadratic terms as *noise* (analogous to the vector $\boldsymbol{e}$), the shared structure becomes apparent. Thus, the ideas that com from using LWE may be translated into candidates of constructions by MQ problems.

However, to bridge the two problems, we need to deal with some subtleties. In the LWE problems, the noise (error vector $\boldsymbol{e}$) comes from a Gaussian distribution that has "moderately" large standard deviation. Intuitively, if the standard deviation is too small, then the problems become easier; on the other hand, if it is too large, then the ciphertexts (constructed from LWE) become undecryptable. Thus, in this series or works [26, 44, 45], certain ranges of parameters for stds have been identified such that both the hardness of the problems and the correctness of the decryption hold simultaneously.

When MQ problems are viewed in this way, we also need to argue that the noise $R(\boldsymbol{x})$ is also "moderate." To achieve this, we use the structure of the assumption that the coefficients of each quadratic term come from Gaussian

[2] There is a simple reduction showing that solving $(S, \boldsymbol{y} = S(\boldsymbol{x}))$ for S contains random constant coefficients is equivalent to solving $(S', \boldsymbol{y}' = S'(\boldsymbol{x}))$, where S' has the same distribution as S, except for the 0 constant coefficients.

distributions with moderately large standard deviations, and the input $\boldsymbol{x}$ comes from a small subset $H^n \subseteq \mathbb{F}_q^n$. That property allows us to bound the size of the noise $R(\boldsymbol{x})$. On the other hand, we need to examine the hardness of the problem for these parameters. To do so, we conduct experiments under what to our knowledge the best quadratic equation solver. Our experiment results confirm our intuitions that MQ problems do not become significantly easier under any (non-trivial) particular distribution of the inputs S and $\boldsymbol{x}$. This particularly gives us evidence of the hardness of the problem in Definition 2, which we can use to construct public-key encryptions.

Our First Construction of Encryption for Bits. In our first attempt, we construct a public-key encryption scheme for bits. This construction is similar in spirit to those LWE-based constructions [26, 44, 45]. Because of the similarity, here we omit discussions of intuitions and refer the curious readers to [26, 44, 45]. Here we give an informal outline of the construction:

- In key generation, the algorithm samples an MQ system S with n variables and $m = c \cdot n$ polynomials, and $\boldsymbol{x} \in H^n$. Then it sets the public key to be $(S, \boldsymbol{y} = S(\boldsymbol{x}))$, and the secret key to be $\boldsymbol{x}$.
- To encrypt a bit b, the encryption algorithm samples $\boldsymbol{r} \in H^m$, and computes $(c_1, c_2) = (\boldsymbol{r}^T \cdot L,\ \boldsymbol{r}^T \cdot (\boldsymbol{y} - \boldsymbol{d}) + b \cdot \lfloor q/2 \rfloor)$. Recall that L is an $m \times n$ matrix, and $m > n$. Thus, given $\boldsymbol{r}^T \cdot L$, $\boldsymbol{r}$ is still hidden information theoretically.
- To decrypt, the algorithm computes $t = c_2 - c_1^T \cdot \boldsymbol{x}$. It outputs 1 if and only if $|t - q/2| \leq q/4$.

Security Proof. The key to the security proof of the bit-encryption scheme is based on a proof that relates the hardness of the assumption to some pseudorandom distribution. Namely, suppose the problem in Definition 2 is hard, then $(S, S(\boldsymbol{x}))$ is indistinguishable from (S, U_m) where U_m is uniform over $\mathbb{F}_q^m$. Moreover, we prove a more general theorem that suppose there exist a distribution over the quadratic terms of S, and a subset $H \subseteq \mathbb{F}_q$ such that the problem is hard, then $(S, S(\boldsymbol{x}))$ is indistinguishable from (S, U_m). The crux of our proof is a new application of the new version of Goldreich-Levin Theorem by Dodis *et. el* [19].

We remark that this general theorem also, as a consequence, implies Theorem 2 plus 3 in [4], and Proposition 5 plus 6 in [35] as its special cases.[3]

Improving Efficiency Using KEM. Feasibility results for bit-encryptions are nice but not quite satisfactory. One general technique to improve efficiency is to use *Key Encapsulation Mechanism* (KEM). We know that to use KEM, it is sufficient to have an efficient symmetric encryption scheme or a pseudorandom generator (PRG). (Note that a pseudorandom generator implies an efficient symmetric

[3] We present our theorem and assumption in asymptotic forms, and both [4, 35] presented their theorems in concrete parameters.

encryption scheme.) Although there are many implementations of PRGs and thus symmetric encryptions as well[7, 8, 22, 30–32, 34], the constructions are either not practically efficient, or require some additional assumption(s).

Here we further observe that the MQ assumption (Definition 2) already gives us an efficient construction of a certain form of PRG[4] that is sufficient to implement the KEM technique. As a consequence, in the resulting scheme, we are able to achieve a public-key encryption scheme that only needs a ciphertext length $L + \mathrm{poly}(k)$ to encrypt a message $M \in \{0,1\}^L$ for any un-prespecified polynomial L, where k is the security parameter. This is essentially *asymptotically optimal* since we know the ciphertext length must be at least as large as the message (otherwise there will be decryption errors), and an additive overhead in the security parameter is the the best we can hope for.

2 Preliminary

2.1 Notation

All vectors are assumed to be column vectors. Unless stated otherwise, all scalar and vector operations are performed modulo q. We use arrow notation to represent a vector, and subscripts to represent the corresponding element, i.e. $\boldsymbol{r} \in \mathbb{F}_q^n$ means $\boldsymbol{r}$ is a vector of n elements in $\mathbb{F}_q$ and $\boldsymbol{r}_i$ means the i-th element of the vector. We denote the transpose of a vector $\boldsymbol{r}$ as $\boldsymbol{r}^T$.

For simplicity we will assume that q is an odd prime. We represent elements in $\mathbb{F}_q$ by integers within the range $[-(q-1)/2, (q-1)/2]$. We denote the inner product of $\boldsymbol{a}$ and $\boldsymbol{b}$ as $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$, or $\boldsymbol{a}^T \cdot \boldsymbol{b}$.

Let m, n, q be numbers. Though out the paper, we will use $S = (R, L, \boldsymbol{d})$ to denote a MQ system with n variables and m equations, where $R \in \mathbb{F}_q^{m \times n \times n}$ denotes the quadratic coefficients, and $L \in \mathbb{F}_q^{m \times n}$ denotes the linear coefficients and $\boldsymbol{d} \in \mathbb{F}_q^m$ denotes the constant coefficients. In particular $R_{i,j,k}$ denotes the coefficient of $x_j x_k$ in the i-th equation, and $L_{i,j}$ denotes the coefficient of x_j in the i-th equation, and $\boldsymbol{d}_i$ denotes the constant coefficient in the i-th equation.

Definition 3 (Multivariate Quadratic Problems). *Let $n, m, q \in \mathbb{N}$ be parameters such that q is a prime, let χ be a distribution between $\mathbb{F}_q^{m \times n \times n}$, and let $H \subseteq \mathbb{F}_q$. The goal for a solver A to the (average-case) multivariate quadratic problem $MQ(n, m, q, \chi, H)$ is that A on a random instance $(S, S(\boldsymbol{x}))$ tries to output some $\boldsymbol{x}' \in \mathbb{F}_q^n$ such that $S(\boldsymbol{x}') = S(\boldsymbol{x})$, where $S = (R, L, \boldsymbol{d})$ with $R \leftarrow \chi$, $L \leftarrow \mathbb{F}_q^{m \times n}$, $\boldsymbol{d} \leftarrow \mathbb{F}_q^m$, and $\boldsymbol{x} \leftarrow H^n$. If A does so, we say it successfully solves the instance.*

[4] The PRG constructed by the MQ assumption is somewhat non-standard but is sufficient for KEM. See Section 5 for further discussions.

Definition 4 (Hardness of a MQ Family).[5] *Let k be the security parameter, $n, m, q : \mathbb{N} \to \mathbb{N}$ be efficiently computable and polynomially bounded such that q is an odd prime. Let χ be a distribution over $\mathbb{F}_q^{m \times n \times n}$ and $H \subseteq \mathbb{F}_q$. We say that the family $MQ(n, m, q, \chi, H)$ is hard to solve if for every* PPT *solver A, there exists some negligible function $\mathsf{ngl}(\cdot)$ such that the following holds for all sufficiently large k:*

$$\Pr_{\substack{S \leftarrow MQ(n,m,q,\chi,H) \\ \boldsymbol{x} \leftarrow H^n}} [\boldsymbol{x}' \leftarrow A(S, S(\boldsymbol{x})) : S(\boldsymbol{x}') = S(\boldsymbol{x})] < \mathsf{ngl}(k).$$

3 Public-Key Encryption Schemes for Bits

In this section, we show a construction of public-key encryption schemes (for bits) under the hardness of some specialized MQ problem. We present our results in the following order: (1) the hardness assumption, (2) the construction of the scheme, and (3) the analysis.

3.1 The Assumption

Definition 5 (MQ Hardness Assumption). *Let k be the security parameter. For every constant $c > 1 \in \mathbb{N}$, every efficiently computable and polynomially bounded $n, m, q : \mathbb{N} \to \mathbb{N}$, $\alpha : \mathbb{N} \to [-q/2, q/2]$ and every $0 < \beta \leq [q/2]$ such that (1) $m = cn$, (2) q is prime, (3) $\alpha = O(1)$, let Φ_α be the distribution of $m \times n \times n$ identical independent discrete Gaussian distribution D_α's with mean 0, standard deviation α, namely, each D_α samples $z \leftarrow N(0, \alpha^2)$ (normal distribution with mean 0, and standard deviation α), and then outputs $\lfloor z \rceil \pmod q$, and let $H_\beta = \{-\beta, -\beta + 1 \ldots, \beta - 1, \beta\}$.*

Then the problem $MQ(n, m, q, \Phi_\alpha, H_\beta)$ is hard to solve.

As discussed in the introduction, we need to choose the parameters α such that $|R(\boldsymbol{x})|$ is "moderate" for two aspects. First, α cannot be too large, otherwise there will be decryption errors. On the other hand, if α is too small, then with high probability, most coefficients are 0, so the system becomes sparse. There are known attacks for sparse systems where there are only $o(1)$ non-zero coefficients, so in our assumption, the α cannot fall into this region. In our setting, $\alpha \geq O(1)$ implies that each quadratic terms has at least a constant probability not being zero, and thus there will be $O(n^2)$ quadratic terms in expectation. In the full version of this paper, we will discuss more details about the parameters and how they influence the hardness of the problem.

[5] To lend more credence to our contention that our family is hard, we attach logarithmic plots in the appendix in which we compare the behavior under MAGMA-2.17 of systems with $m/n = 2$ in cases (A) random systems in GF(3) and GF(5); (B) systems in larger fields but with variables restricted to $\{-1, 0, 1\}$ and the equations $x_i^3 = x_i$ included for every i; (C) systems in larger fields but with variables restricted to $\{-2, -1, 0, 1, 2\}$ and the equations $x_i(x_i^2 - 1)(x_i^2 - 4) = 0$ included for every i. The trend looks quite exponential. For more discussion see the full version of this paper.

Remark 1. As we discussed in the introduction, the MQ assumption has a similar structure to the LWE assumption. Here we do a brief comparison of the two assumptions for different range of parameters.
For q being superpolynomial, we can show that an MQ instance $(S, S(\boldsymbol{x}))$ can be transformed to (L, b) that is statistically close to an LWE instance. The transformation just sets L as the linear part of S, and sets $b = S(\boldsymbol{x}) + \boldsymbol{e}'$, where each coordinate of $\boldsymbol{e}'$ comes from some i.i.d. Gaussian with a small std. For $q = \mathsf{superpoly}(k)$, one can show that b is statistically close to $L \cdot \boldsymbol{x} + \boldsymbol{e}''$ where each coordinate of $\boldsymbol{e}''$ comes from i.i.d. Gaussian with a slightly bigger std. Thus, (L, b) is statistically close to an LWE instance, and consequently, there is a simple reduction from MQ to LWE.

In this paper, we need $q = \mathrm{poly}(k)$ for our construction. For this range of parameters, the above argument does not work. In fact, an MQ instance and an LWE instance can be statistically far. Thus, a straitforward reduction from MQ to LWE does not work. We are not aware of any other reduction from any one to the other, and leave this issue as an interesting open question.

Under the above assumption, we are able to obtain the following lemma, which is a key to the security proof of our construction of public-key encryption scheme. In the following section, we are going to prove a more general result as Theorem 2, which directly implies this lemma. Thus, we only put the statement of the lemma.

Lemma 1. *Let k be the security parameter, and assuming $MQ(n, m, q, \Phi_\alpha, H_\beta)$ be the hard problem as stated in Definition 5. Then $(S, S(\boldsymbol{x}))$ is computationally indistinguishable from (S, U_m), where $S \leftarrow MQ(n, m, q, \Phi_\alpha, H_\beta)$, $\boldsymbol{x} \leftarrow H_\beta^n$, U_m is the uniform distribution over $\mathbb{F}_q^m$.*

Here we remark that the MQ hardness assumption in Definition 5 can be generalized in the following sense.

Remark 2. Actually all we need for our construction is to bound the quantity $R(\boldsymbol{x})$.Thus any distribution of S, and $\boldsymbol{x}$ that has the following properties (1) the problem of equation solving is hard, and (2) we are able to bound $R(\boldsymbol{x})$, are sufficient for us to construct public-key encryptions. Here for concreteness, we present study Φ_α and H_β^n as a candidate for the hard problem.

3.2 Construction of a Public-Key Encryption Scheme for Bits

In this section we present our construction of a public-key bit-encryption scheme.

Construction of the Scheme $\mathcal{E} = (\mathrm{KeyGen}(\cdot), \mathrm{Enc}(\cdot), \mathrm{Dec}(\cdot))$:

- $\mathrm{KeyGen}(1^k)$: choose public parameters n, m, q, α, β, and $\lambda \in \mathbb{N}$ satisfying the following constraints:
  1. $k \cdot \alpha \cdot n^{(2+\lambda)} \cdot m \cdot \beta^2 \leq q/4$.
  2. $m \cdot \log(2n^\lambda + 1) \geq (n+1) \cdot \log q + 2k$.
  3. n, m, q, α, β satisfy the condition in the MQ assumption such that $MQ(n, m, q, \Phi_\alpha, H_\beta)$ is hard to solve.

Then it samples a random instance $(S, S(\boldsymbol{x})) \leftarrow MQ(n, m, q, \Phi_\alpha, H_\beta)$, and deontes $\boldsymbol{y} = S(\boldsymbol{x})$. Then it sets $\mathsf{pk} = (S, \boldsymbol{y}) = ((R, L, d), \boldsymbol{y})$, $\mathsf{sk} = \boldsymbol{x}$.

- $\mathrm{Enc}(b)$ for $b \in \{0, 1\}$: sample $\boldsymbol{r} \in H^m_{n^\lambda}$, and outputs $(c_1, c_2) = (\boldsymbol{r}^T \cdot L,\ \boldsymbol{r}^T \cdot (\boldsymbol{y} - \boldsymbol{d}) + b \cdot [q/2])$.
- $\mathrm{Dec}(c_1, c_2)$: compute $t = c_2 - c_1^T \cdot \boldsymbol{x}$. If $|t - q/2| \leq q/4$ then output 1, otherwise 0.

The intuition of the construction and analysis of security are similar to the case of the work [45]. Thus we only state the theorem and leave the discussions in the full version of this paper.

Theorem 1. *Assume the MQ assumption holds for the above parameters. Then the scheme $\mathcal{E}$ is a semantically secure encryption scheme.*

4 Hardness of MQ Problems Implies Pseudorandom Distributions

Recall that in the previous section, we claimed that the hardness of some family of MQ problems implies a pseudorandom distribution (Lemma 1). In this section, we are going to show that the hardness of more general families of MQ problems also implies a pseudorandom distribution. In particular, we obtain the following theorem.

Theorem 2. *Let k be the security parameter, n, m, q be efficiently computable and polynomially bounded such that q is an odd prime, χ is a distribution over $\mathbb{F}_q^{m \times n \times n}$, and $H \subseteq \mathbb{F}_q$.*

Suppose for these parameters the problem $MQ(n, m, q, \chi, H)$ is hard to solve, then the following two distributions are computationally indistinguishable. $D_1 = (S, S(\boldsymbol{x}))$, $D_2 = (S, U_m)$, where $S \leftarrow MQ(n, m, q, \chi, H)$, $\boldsymbol{x} \leftarrow H^n$, and U_m is a uniform distribution over $\mathbb{F}_q^m$.

If we set H to be H_β, and χ to be Φ_α as the setting in Definition 5, then this version of the theorem directly becomes Lemma 1.

We prove the theorem by contradiction. For intuition, first we state our high level ideas and then delve into details. Suppose there exists a distinguisher A that distinguishes D_1 and D_2, from here we want to construct an inverter B that solves the MQ problem $(S, S(\boldsymbol{x}))$, which leads to a contradiction. We achieve this goal using the following strategy:

- First we show that from A, we can construct another algorithm A' that distinguishes $D'_1 = (S, S(\boldsymbol{x}), \boldsymbol{r}, \langle \boldsymbol{r}, \boldsymbol{x} \rangle)$ and $D'_2 = (S, S(\boldsymbol{x}), \boldsymbol{r}, U)$ where $\boldsymbol{r} \in \mathbb{F}_q^n$ is a random vector, and U is uniform over $\mathbb{F}_q$. For any $\boldsymbol{r} \in \mathbb{F}_q^n$, we can view $\langle \boldsymbol{r}, \boldsymbol{x} \rangle$ as the $\boldsymbol{r}$'s location of the (Hadamard) encoding of $\boldsymbol{x}$. The ability to distinguish D'_1 and D'_2 gives us a somewhat corrupted codeword of $\boldsymbol{x}$, i.e., the codeword is correct in at least a noticeable fraction of places over all $\boldsymbol{r}$'s.

– Then from A', we construct an inverter B that applies the list-decoding algorithm by the Goldreich-Levin Theorem to recover $\boldsymbol{x}$. We remind the reader that the Goldreich-Levin Theorem is essentially a decoding algorithm for the Hadamard code, which says (informally) that if given $f(\boldsymbol{x})$, for random $\boldsymbol{r}$'s one can distinguish $\langle \boldsymbol{r}, \boldsymbol{x} \rangle$ from a uniform element with noticeable probability, then one can invert f with noticeable probability (for any function f).

However, when applying the Goldreich-Levin Theorem here, we encountered some subtleties. First the classical theorem [28] deals with the boolean field only (i.e. $q = 2$); thus it is not applicable in general cases. A generalized version of [29] handles the case for large q's, but it works only for the case where the input $\boldsymbol{x} \in \mathbb{F}_q^n$. It remains unclear for the case where $\boldsymbol{x}$ comes from a subset $H_\beta^n \subseteq \mathbb{F}_q^n$. Recently, Dodis *et al.* [19] proved a new version of the theorem that is essentially what we need in our setting. With it, we are able to implement the list-decoding algorithm in the second bullet above, and this completes the proof. The formal proof will appear in the full version of this paper.

5 Key Encapsulation Mechanism

In the previous section, we constructed a public-key encryption for bits. However, this approach is not satisfactory when we want to encrypt a long message $M \in \{0,1\}^L$ for some large L. As discussed in the introduction, we can use a key encapsulation mechanism (KEM) to achieve better efficiency.

First, we recall how we can achieve this by the KEM technique: let Enc be any public-key encryption scheme for bits, and let $G : \{0,1\}^k \to \{0,1\}^{k+t}$ be a pseudorandom generator. To encrypt a long message $M \in \{0,1\}^L$, we first sample a seed $s \in \{0,1\}^k$ for the PRG, and then stretch the generator G[6] to get a pseudorandom string $G'(s) \in \{0,1\}^L$. Then we encrypt the seed by the public-key scheme and use the pseudorandom string as a one-time pad to XOR M. The resulting ciphertext becomes $(\mathrm{Enc}_{\mathsf{pk}}(s), G'(s) \oplus M)$.

In this paper, we observe that the MQ assumption implies a certain form of PRG. Thus, we can implement KEM under the same assumption as the one from which we construct the public-key encryption scheme. However, this type of PRG is somewhat non-standard, so we avoid using this term formally. We will discuss this issue in the full version of this paper.

In the next section, we are going to show how we can obtain the desired long pseudorandom string $G'(s)$, and then present the entire scheme in Section 5.2. Finally we sketch the proof of security, which follows from the folklore.

Remark 3. We remark that KEM is a generic way to construct efficient public-key encryption schemes. As discussed in the introduction and the above, we know that a PRG plus any bit-encryption public encryption scheme is sufficient to achieve the task. In this paper, we observe that the MQ assumption implies an efficient constructions of PRGs and a public-key bit-encryption scheme, so we can obtain an efficient public-key encryption under one single assumption.

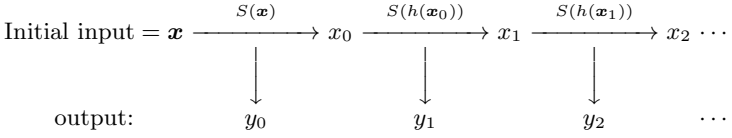
[6] We refer the readers to [27] for the details of how to stretch a PRG.

5.1 Longer Pseudorandom Strings

Recall that Lemma 1 states that $(S, S(\boldsymbol{x})) \approx_c (S, U_m)$. This means we can get a pseudorandom string $S(\boldsymbol{x}) \in \mathbb{F}_q^m$ by only sampling a shorter seed $\boldsymbol{x} \in H_\beta^n$. Note: $m > n$, and $H \subseteq \mathbb{F}_q$. To get a longer pseudorandom string, we can use the following iterative method (analogous to how we can stretch a PRG.)

Definition 6. *Let $S \leftarrow MQ(n, m, q, \Phi_\alpha, H_\beta)$. For $\boldsymbol{x} \in H_\beta^n$, and let $(x_0, y_0) = S(\boldsymbol{x})$ where $\boldsymbol{x}_0 \in \mathbb{F}_q^n$, $\boldsymbol{y}_0 \in \mathbb{F}_q^{m-n}$ be the prefix n elements and the suffix $m - n$ elements of $S(\boldsymbol{x})$ respectively.*

Let $h : \mathbb{F}_q^n \to H_\beta^n$ be a hash function, and for $i \in \mathbb{N}$, we recursively define $(\boldsymbol{x}_i, \boldsymbol{y}_i) = S(h(\boldsymbol{x}_{i-1}))$ where $\boldsymbol{x}_i \in \mathbb{F}_q^n, \boldsymbol{y}_i \in \mathbb{F}_q^{m-n}$ (representing the prefix and suffix of $S(h(\boldsymbol{x}_{i-1}))$ respectively). Then we define $S_h^i(\boldsymbol{x}) = (\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_i)$.



Then we are going to argue that for any $i \leq \text{poly}(k)$, we have $(S, S_h^i(\boldsymbol{x})) \approx_c (S, U_{(m-n)\cdot(i+1)})$, given $(S, S(\boldsymbol{x})) \approx_c (S, U_m)$. This means, we can get an arbitrarily long (polynomially bounded) pseudorandom string $S_h^i(\boldsymbol{x})$ from an initial random seed $\boldsymbol{x}$.

The proof of security follows from a hybrid argument, and it is similar to that of QUAD in the work [4]. We remark that here we need the hash function for some technical reason. The only property we require is that $h(U_n)$ outputs a (statistically close) uniformly random element in H_β^n. The hash function h does not need to be collision resistant nor one-way. We can view h as a reinterpretation from elements in $\mathbb{F}_q^n$ to elements in H_β^n, and thus there are many simple constructions.

Then we are able to achieve the following theorem.

Theorem 3. *Let k be the security parameter. Assuming the MQ problem $MQ(n, m, q, \Phi_\alpha, H_\beta)$ is hard, and let $h : \mathbb{F}_q^n \to H_\beta^n$ be a (randomized) hash function such that $h(\boldsymbol{z})$ maps a uniformly random $\boldsymbol{z} \in \mathbb{F}_q^n$ to a uniformly random $\boldsymbol{y} \in H_\beta^n$.*

Then for any $i = \text{poly}(k)$, $(S, S_h^i(\boldsymbol{x}))$ is computationally indistinguishable to $(S, U_{(m-n)\cdot(i+1)})$, where $S \leftarrow MQ(n, m, q, \Phi_\alpha, H_\beta)$, $\boldsymbol{x} \leftarrow H_\beta^n$, and $U_{(m-n)\cdot(i+1)}$ is uniform over $\mathbb{F}_q^{(m-n)\cdot(i+1)}$.

The proof will appear in the full version of this paper.

5.2 Construction of the KEM Scheme

In previous sections, we have constructed the bit encryption scheme $\mathcal{E} = (\text{KeyGen}(\cdot), \text{Enc}(\cdot), \text{Dec}(\cdot))$ described in section 3.2, and the pseudorandom generator above. Here we describe a KEM scheme $\mathcal{E}_{\mathsf{KEM}} =$

$(\mathrm{KeyGen}_{\mathsf{KEM}}(\cdot), \mathrm{Enc}_{\mathsf{KEM}}(\cdot), \mathrm{Dec}_{\mathsf{KEM}}(\cdot))$ that can encrypt messages with unprespecified lengths (polynomially bounded).

- $\mathrm{KeyGen}_{\mathsf{KEM}}(1^k)$: run $\mathrm{KeyGen}(1^k)$. In particular, the algorithm chooses public parameters $n, m, q, \Phi_\alpha, H_\beta$ in the range as stated in the MQ assumption, and also a hash function $h : \mathbb{F}_q^n \to H_\beta^n$ with the property $h(U_n)$ being uniform over H_β^n as discussed in the above section. Then it samples a random instance $(S, S(\boldsymbol{x})) \leftarrow MQ(n, m, q, \Phi_\alpha, H_\beta)$, and deontes $\boldsymbol{y} = S(\boldsymbol{x})$. Then it sets $\mathsf{pk} = (S, \boldsymbol{y})$, $\mathsf{sk} = \boldsymbol{x}$.
- For any $L = \mathrm{poly}(k)$, and any message $M \in \mathbb{F}_q^L$, $\mathrm{Enc}_{\mathsf{KEM}}(M)$ does the following: the algorithm samples $\boldsymbol{s} \in H_\beta^n$, and computes $c_i = \mathrm{Enc}(\mathsf{pk}, \boldsymbol{s}_i)$ for $i \in [n]$. Then let $t = \lceil L/(m-n) \rceil$, and compute $c^* = M \oplus S_h^t(\boldsymbol{s})$.[7] The resulting ciphertext will be $c = (c_1, c_2, \ldots, c_n, c^*)$.
- $\mathrm{Dec}_{\mathsf{KEM}}(c)$: the algorithm computes $\boldsymbol{s}$ by running $\mathrm{Dec}(\mathsf{sk}, c_i)$ for $i \in [n]$. Then it outputs $M = c^* \oplus S_h^t(\boldsymbol{s})$.

Then we are able to obtain the following theorem.

Theorem 4. *The scheme above $\mathcal{E}_{\mathsf{KEM}}$ is a semantically secure encryption scheme.*

5.3 Concrete Parameters

Our goal here is to instantiate Theorem 4 with concrete parameters. Here, we exhibit two sets of parameters (for proven security levels 2^{80} and 2^{128}) based on a conservative estimate of the hardness of MQ systems (i.e., assuming the general applicability of sparse matrix solvers in XL [47]), and no particular effort in optimization.

Our security level aims for time 2^{80} (and 2^{128}), and $\varepsilon = 2^{-10}$ for plaintext length $L = 2^{20}$ (1 Mb), i.e., no adversary within running time 2^{80} (and 2^{128}) can distinguish two ciphertexts with advantage better than 2^{-10}. Since our construction uses the KEM mechanism, we need parameters for (1) $(S, S_h^t(\boldsymbol{x}))$ to be a PRG some length L, and (2) $\mathcal{E}$ to be a semantically secure bit-encryption scheme. It follows from a standard argument that the KEM security achieves this level (with a slight loss) once both the underlying PRG and the encryption scheme achieve this level of security. In particular, we instantiate the scheme with the following parameters:

Case	k	n	m	α	β	q
1	12	200	400	10	2	$1803131754697263278851 9 \approx 2^{74}$
2	12	256	512	10	2	$5232440279576267872 4873 \approx 2^{76}$

And we approximate the hardness in the following table:

Case	Hardness of MQ	Security of Enc	Security of PRG	Security of KEM
1	2^{156}, 2^{-100}	2^{87}, 2^{-11}	2^{85}, 2^{-11}	2^{85}, 2^{-10}
2	2^{205}, 2^{-104}	2^{130}, 2^{-11}	2^{134}, 2^{-11}	2^{130}, 2^{-10}

[7] Here $\oplus$ means we add two vectors component-wise. That is, let $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{F}_q^L$, then we say $\boldsymbol{a} \oplus \boldsymbol{b} = [\boldsymbol{a}_1 + \boldsymbol{b}_1, \boldsymbol{a}_2 + \boldsymbol{b}_2, \cdots, \boldsymbol{a}_L + \boldsymbol{b}_L]^T$.

We remark the tuple (T, ε) in each cell means for any adversary running in time T has advantage (or success probability) less than ε.

In the full version of this paper, we will explain our methodology of the experimental studies, and provide the data. Due to space limit, we omit most details for proofs and experiments.

Acknowledgement. The authors would like to thank Kai-Min Chung for his valuable comments. Feng-Hao Liu is supported by National Science Foundation for partial support under grant CNS-0347661 and CNS-0831293. Yun-Ju Huang and Bo-Yin Yang thank the Taiwan National Science Council and the Academia Sinica for partial support under grant NSC-100-2218-E-001-002 and the AS Career Award.

References

1. Ajtai, M., Dwork, C.: A public-key cryptosystem with worst-case/average-case equivalence. In: STOC, pp. 284–293 (1997)
2. Applebaum, B., Barak, B., Wigderson, A.: Public-key cryptography from different assumptions. In: STOC (2010)
3. Ars, G., Faugère, J.-C., Imai, H., Kawazoe, M., Sugita, M.: Comparison Between XL and Gröbner Basis Algorithms. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 338–353. Springer, Heidelberg (2004)
4. Berbain, C., Gilbert, H., Patarin, J.: QUAD: A Practical Stream Cipher with Provable Security. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 109–128. Springer, Heidelberg (2006)
5. Bettale, L., Faugère, J.-C., Perret, L.: Cryptanalysis of Multivariate and Odd-Characteristic HFE Variants. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 441–458. Springer, Heidelberg (2011)
6. Billet, O., Patarin, J., Seurin, Y.: Analysis of intermediate field systems. In: SCC (2008)
7. Blum, L., Blum, M., Shub, M.: A simple unpredictable pseudo-random number generator. SIAM J. Comput. 15 (1986)
8. Blum, M., Micali, S.: How to generate cryptographically strong sequences of pseudo-random bits. SIAM J. Comput. (1984)
9. Bogdanov, A., Eisenbarth, T., Rupp, A., Wolf, C.: Time-Area Optimized Public-Key Engines: $\mathcal{MQ}$-Cryptosystems as Replacement for Elliptic Curves? In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 45–61. Springer, Heidelberg (2008)
10. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
11. Chen, A.I.-T., Chen, M.-S., Chen, T.-R., Cheng, C.-M., Ding, J., Kuo, E.L.-H., Lee, F.Y.-S., Yang, B.-Y.: SSE Implementation of Multivariate PKCs on Modern x86 CPUs. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 33–48. Springer, Heidelberg (2009)
12. Chen, C.-H.O., Chen, M.-S., Ding, J., Werner, F., Yang, B.-Y.: Odd-char multivariate hidden field equations. Cryptology ePrint Archive, Report 2008/543 (2008)

13. Chen, Y., Nguyen, P.Q.: BKZ 2.0: Better Lattice Security Estimates. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 1–20. Springer, Heidelberg (2011)
14. Courtois, N., Klimov, A., Patarin, J., Shamir, A.: Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 392–407. Springer, Heidelberg (2000)
15. Courtois, N.T., Pieprzyk, J.: Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 267–287. Springer, Heidelberg (2002)
16. Diem, C.: The XL-Algorithm and a Conjecture from Commutative Algebra. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 323–337. Springer, Heidelberg (2004)
17. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Transactions on Information Theory (1976)
18. Ding, J., Dubois, V., Yang, B.-Y., Chen, O.C.-H., Cheng, C.-M.: Could SFLASH be Repaired? In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 691–701. Springer, Heidelberg (2008)
19. Dodis, Y., Goldwasser, S., Kalai, Y.T., Peikert, C., Vaikuntanathan, V.: Public-Key Encryption Schemes with Auxiliary Inputs. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 361–381. Springer, Heidelberg (2010)
20. Dubois, V., Fouque, P.-A., Shamir, A., Stern, J.: Practical Cryptanalysis of SFLASH. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 1–12. Springer, Heidelberg (2007)
21. Dubois, V., Fouque, P.-A., Stern, J.: Cryptanalysis of SFLASH with Slightly Modified Parameters. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 264–275. Springer, Heidelberg (2007)
22. Farashahi, R.R., Schoenmakers, B., Sidorenko, A.: Efficient Pseudorandom Generators Based on the DDH Assumption. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 426–441. Springer, Heidelberg (2007)
23. Faugère, J.C.: A new efficient algorithm for computing gröbner bases without reduction to zero (f5). In: ISSAC, New York, NY, USA (2002)
24. Faugère, J.-C., Joux, A.: Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 44–60. Springer, Heidelberg (2003)
25. Fraenkel, A.S., Yesha, Y.: Complexity of solving algebraic equations. Inf. Process. Lett. (1980)
26. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC (2008)
27. Goldreich, O.: Foundations of Cryptography. Basic tools. Cambridge University Press (2001)
28. Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: STOC, pp. 25–32 (1989)
29. Goldreich, O., Rubinfeld, R., Sudan, M.: Learning polynomials with queries: The highly noisy case. In: FOCS (1995)
30. Haitner, I., Reingold, O., Vadhan, S.P.: Efficiency improvements in constructing pseudorandom generators from one-way functions. In: STOC (2010)
31. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM J. Comput. 28(4), 1364–1396 (1999)

32. Holenstein, T.: Pseudorandom Generators from One-Way Functions: A Simple Construction for Any Hardness. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 443–461. Springer, Heidelberg (2006)
33. Impagliazzo, R.: A personal view of average-case complexity. In: Structure in Complexity Theory Conference, pp. 134–147 (1995)
34. Kaliski Jr., B.S.: A Pseudo-random Bit Generator Based on Elliptic Logarithms. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 84–103. Springer, Heidelberg (1987)
35. Liu, F.-H., Lu, C.-J., Yang, B.-Y.: Secure PRNGs from Specialized Polynomial Maps over Any $\mathbb{F}_q$. In: Buchmann, J., Ding, J. (eds.) PQCrypto 2008. LNCS, vol. 5299, pp. 181–202. Springer, Heidelberg (2008)
36. Macaulay, F.S.: On some formulae in elimination. Proceedings of the London Mathematical Society (1902)
37. Matsumoto, T., Imai, H.: Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption. In: Günther, C.G. (ed.) EUROCRYPT 1988. LNCS, vol. 330, pp. 419–453. Springer, Heidelberg (1988)
38. McEliece, R.: A public-key cryptosystem based on algebraic coding theory. DSN Progress Report (1978)
39. Patarin, J.: Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt '88. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 248–261. Springer, Heidelberg (1995)
40. Patarin, J.: Asymmetric Cryptography with a Hidden Monomial. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 45–60. Springer, Heidelberg (1996)
41. Patarin, J., Goubin, L.: Asymmetric Cryptography with S-boxes. Is it easier than expected to design efficient asymmetric cryptosystem? In: Han, Y., Quing, S. (eds.) ICICS 1997. LNCS, vol. 1334, pp. 369–380. Springer, Heidelberg (1997)
42. Patarin, J., Goubin, L., Courtois, N.: C^*_{-+} and HM: Variations Around Two Schemes of T. In: Ohta, K., Pei, D. (eds.) ASIACRYPT 1998. LNCS, vol. 1514, pp. 35–50. Springer, Heidelberg (1998)
43. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: STOC (2009)
44. Peikert, C., Vaikuntanathan, V., Waters, B.: A Framework for Efficient and Composable Oblivious Transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008)
45. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM (2009)
46. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21 (1978)
47. Yang, B.-Y., Chen, O.C.-H., Bernstein, D.J., Chen, J.-M.: Analysis of `QUAD`. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 290–308. Springer, Heidelberg (2007)
48. Yang, B.-Y., Chen, J.-M.: All in the XL Family: Theory and Practice. In: Park, C.-S., Chee, S. (eds.) ICISC 2004. LNCS, vol. 3506, pp. 67–86. Springer, Heidelberg (2005)

Anonymous Broadcast Encryption: Adaptive Security and Efficient Constructions in the Standard Model

Benoît Libert[1], Kenneth G. Paterson[2], and Elizabeth A. Quaglia[2]

[1] Université Catholique de Louvain, ICTEAM Institute, Belgium
[2] Information Security Group, Royal Holloway, University of London, U.K.

Abstract. In this paper we consider *anonymity* in the context of Broadcast Encryption (BE). This issue has received very little attention so far and *all but one* of the currently available BE schemes fail to provide anonymity. Yet, we argue that it is intrinsically desirable to provide anonymity in standard applications of BE and that it can be achieved at a moderate cost. We provide a security definition for Anonymous Broadcast Encryption (ANOBE) and show that it is achievable assuming only the existence of IND-CCA secure public key encryption (PKE). Focusing on reducing the size of ciphertexts, we then give two generic constructions for ANOBE. The first is from any anonymous (key-private) IND-CCA secure PKE scheme, and the second is from any IBE scheme that satisfies a weak security notion in the multi-TA setting. Furthermore, we show how randomness re-use techniques can be deployed in the ANOBE context to reduce computational and communication costs, and how a new cryptographic primitive – anonymous hint systems – can be used to speed up the decryption process in our ANOBE constructions. All of our results are in the standard model, achieving fully collusion-resistant ANOBE schemes secure against *adaptive* IND-CCA adversaries.

Keywords: Broadcast Encryption, Anonymity.

1 Introduction

Anonymity. In a world that is increasingly relying on digital technologies, addressing the issue of protecting users' privacy is of crucial importance. This is reflected by the great attention given to *anonymity* in all the main fields of modern cryptography. In the area of Public-Key Encryption (PKE), anonymity is often referred to as key-privacy [6]. This notion captures the property that an eavesdropper is not able to tell under which one of several public keys a ciphertext was created. The analogous concept in the ID-based setting was studied in [1]. The benefit of preserving receivers' privacy is relevant in more elaborate systems involving for example Hierarchical IBE [12], Attribute-Based Encryption (ABE) or Predicate Encryption [26], where achieving anonymity guarantees becomes increasingly challenging. Furthermore, in the context of digital signatures, a number of primitives effectively *rely* on anonymity: group signatures [16] and anonymous credentials [15] are well-known examples of this.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 206–224, 2012.

Broadcast Encryption. Broadcast Encryption (BE) addresses the issue of confidentially broadcasting a message to an arbitrary subset drawn from a universe of users. We will call the universe of n users U and the target, or privileged, set S, where $S \subseteq U$. Since its introduction in 1993 by Fiat and Naor [22], various flavours of BE have been introduced: the scheme can be in a symmetric or asymmetric setting; the set of receivers could be static or dynamic; revocation and traitor-tracing algorithms could be integrated into the system, users' keys might or might not be updated and then forward secrecy may be achieved. We refer to some of the relevant work in the area and the references therein [22,32,19,39,9,18,17,24,36]. One of the fundamental properties of a BE scheme is *collusion resistance* in the sense that no coalition of users in $U \setminus S$ should be able to recover the message. In the literature we can find several schemes that resist collusion attacks mounted by coalitions of at most $t < n$ users; only some schemes are *fully* collusion-resistant, *i.e.* they can tolerate attacks by coalitions of any size. For the purpose of this paper, we will consider systems that are *public-key*, allow *stateless receivers* (users that are not required to update their private keys) and are *fully* collusion-resistant. These are by now standard objectives for a BE scheme in the public-key setting.

Several additional practical aspects need to be taken into consideration, especially in view of the real-life applications of BE: strength of security notions, public and private storage requirements, ciphertext length, and computational costs. The specific nature of the primitive has led researchers to focus in particular on solutions having ciphertexts that are as short as possible. In this respect, the results of [9] and [24] are nearly optimal. However, designing BE schemes for real-life applications to broadcasting should not only involve efficiency and confidentiality issues. In particular, the privacy of users should be protected as much as possible. We believe that, to date, this aspect has not been adequately dealt with. Our study of the literature reveals that anonymity in BE has only been considered in a single paper [5], in the context of encrypted file systems[1]. Surprisingly, almost all subsequent work on BE has ignored the issue of anonymity. Moreover, as we shall explain below, state-of-the-art BE schemes are inherently incapable of providing any kind of anonymity.

Anonymity in Broadcast Encryption. According to commonly accepted definitions [24,10,17], a BE scheme consists of four algorithms: `Setup`, `KeyGen`, `Enc` and `Dec`. Each user in the system can obtain his private key from the `KeyGen` algorithm, and the sender can choose an *arbitrary* target set of users S to which he wishes to broadcast a message. To decrypt, a legitimate user, i.e. a user in S, has to run the decryption algorithm on input the ciphertext, his private key *and* a description of the target set S. This set S is required specifically as an input to `Dec` in the existing definitions of BE. Hence the user needs to somehow know to which set S the message was broadcast, otherwise he cannot decrypt. Unfortunately, solving this problem is not just a matter of removing this requirement from the model, as current schemes explicitly *rely* on S as an

[1] We observe that [25] addresses the issue of hiding the identity of the *sender* in a broadcast protocol, which is *not* what we intend by anonymous broadcast encryption.

input to `Dec` for decryption to work. Thus these schemes cannot provide any anonymity.

This limitation in the existing BE model and schemes clearly causes serious privacy issues: imagine we deploy a BE scheme, as defined above, for television broadcasting. Suppose the privileged set is the set of all users who have paid a subscription to a certain channel. Each customer should have access to that channel using his private key. The problem is that, to decrypt, he will have to know who *else* has paid for the specific subscription! Not only is this requirement very inconvenient for the practical deployment of BE schemes, it is also a severe violation of the individual subscriber's privacy. Ideally, a BE scheme should protect users' privacy by guaranteeing that ciphertexts do not leak any information about the privileged set S.

Current BE schemes such as those in [24,10,17] do not account for the cost of broadcasting a description of S when calculating the size of ciphertexts. In the most general usage scenario intended for BE, where S is dynamic and may be unpredictable from message to message, the ciphertexts in such schemes must effectively include a description of S as part of the ciphertexts themselves. This means that the true ciphertext size in these schemes is linear in n rather than constant-size, as a cursory examination of the schemes might suggest[2]. However, achieving linear-sized ciphertext is already an impressive achievement, since there is a simple counting argument showing that, for a universe of n users in which every possible subset S should be reachable by secure broadcast, ciphertexts must contain at least n bits.

Further Details on Related Work. As mentioned above, the only prior work addressing the issue of anonymity in BE appears to be that of Barth *et al.* [5] (there, it is called *privacy*). In [5], several BE systems used in practice were examined with respect to anonymity. In addition, a generic construction for a BE scheme using a key-private, IND-CCA secure PKE scheme was given, with the scheme achieving anonymity and IND-CCA security against static adversaries. The construction encrypts the message for each intended receiver using the PKE scheme, and then ties together the resulting ciphertexts using a strongly secure one-time signature. Barth *et al.* [5] also provided a technique which can be used to speed-up decryption, but this technique was only analysed in the Random Oracle Model.

In very recent work [21] that builds on [5] and this paper, the authors have given constructions for anonymous broadcast encryption schemes with compact ciphertexts, but using a much weaker notion of anonymity that does not seem to relate very closely to real-world requirements.

In [11] the authors provide a private linear broadcast encryption (PLBE) scheme to realise a fully collusion-resistant traitor-tracing scheme. A PLBE, however, is a BE system with limited capabilities (i.e. it cannot address arbitrary

[2] This does not rule the use of compact encodings of S being transmitted with ciphertexts in more restrictive usage scenarios, for example, only sending the difference in S when the set S changes only slowly from message to message.

sets of users) and hence this work does not provide a solution to the problem considered so far.

There is much work, both cryptographic and non-cryptographic, on pseudonymous systems. In principle, pseudonyms could be used to enhance the anonymity of BE schemes: now users would not be identifiable directly, since a certificate would link a public key to a pseudonym rather than a real name. However, ciphertexts would still be linkable, in the sense that it would be possible to detect if two ciphertexts were intended for the same set of recipients or not. The approach we take here offers much stronger levels of privacy, removing ciphertext linkability in particular.

Our Contributions. Despite its importance, anonymous broadcast encryption has not received much attention since the initial work of Barth *et al.* [5]. This paper aims to raise the profile of this neglected primitive.

We start by giving a unified security definition for Anonymous Broadcast Encryption (ANOBE). Instead of separating anonymity and confidentiality as in [5], we use a combined security notion for ANOBE which helps to streamline our presentation and proofs. In addition, we strengthen the model to allow the adversary to make *adaptive* corruptions, *with all of our constructions achieving security in this setting.* In contrast, the definition of [5] is static, requiring the adversary to choose whom to corrupt before seeing the public keys in the system. As a first step, we show that our enhanced security definition is satisfiable: adaptively secure ANOBE can be built based only on the existence of IND-CCA secure PKE (without requiring the base PKE scheme to have anonymity properties itself). This construction results in a very efficient (constant-time) decryption procedure but has ciphertexts whose size is linear in n, the number of users in the universe U.

Our second contribution is to show that the generic construction for ANOBE suggested by Barth *et al.* [5] actually possesses adaptive security, and not merely static security as was established in [5]. This construction starts from any weakly robust (in the sense of [2]), key-private PKE scheme with chosen-ciphertext security. In comparison with our first generic construction, this result imposes stronger requirements on the underlying encryption scheme. However, it achieves shorter ciphertexts, with the size being linear in the size of the target set S. We also provide a variant of this construction that replaces the IND-CCA secure PKE component with an identity-based encryption (IBE) scheme having suitable security properties. This alternative further increases the set of components that can be used to obtain ANOBE.

One major drawback of the latter constructions is that decryption takes linear time in the size of the set S. Our third result is a technique allowing for constant decryption cost and which we prove secure in the standard model (*i.e.*, without random oracles) using our enhanced security definition. So far, the only known technique – put forth by Barth *et al.* [5] – enabling constant-time decryption requires the random oracle heuristic in the security analysis. To eliminate the random oracle, we introduce a new primitive, which we call an *anonymous hint system.* In essence, this primitive provides a way for an encrypter to securely

tell receivers which ciphertext component is intended for them, allowing them to ignore all but one ciphertext component and so decrypt more efficiently. The hint primitive, for which we provide an implementation based on the Decision-Diffie-Hellman (DDH) assumption, is defined and realized in such a way that its integration with our generic ANOBE constructions maintains compatibility with our proofs of adaptive security.

Our fourth contribution is to show how randomness re-use techniques originally developed for PKE in [28,8,7] can be modified for secure deployment in the ANOBE setting. In particular, we identify a slightly stronger notion of reproducibility that we call *key-less reproducibility*. We show that if our base PKE scheme has this property (in addition to the other properties needed in our generic construction) then it can be used with the same randomness across all ciphertext components in our main ANOBE construction. This not only allows the size of ciphertexts to be reduced further (by eliminating repeated ciphertext elements) but also reduces the sender's computational overhead.

In the full version of the paper [30], we establish that the Kurosawa-Desmedt (KD) [29] hybrid encryption scheme can be tweaked to have all the properties that are needed of the base PKE scheme in our constructions. The KD scheme is an ideal starting point since it is one of most efficient PKE schemes with IND-CCA security in the standard model.

Tying everything together and using KD* as the base scheme, we obtain the *currently most efficient instantiation* of an ANOBE scheme, for which ciphertexts contain only 2 group elements and $|S|$ symmetric ciphertexts (plus a signature and a verification key). Decryption can be achieved in constant time by combining this scheme with our DDH-based hint system, with an additional $2|S|+1$ group elements in the ciphertext.

As can be seen from the details of our constructions, achieving anonymity does not add *any* cost to the encryption process compared to non-anonymous schemes (for example, [9,24]): in our ANOBE schemes, encryption requires a number of group operations that is linear in $|S|$. As for decryption, our speed-up technique allows the legitimate user to recover the message in constant time. Our ciphertext size is linear in $|S|$ (and thus linear in n and of the same order of magnitude as the *true* ciphertext size in existing BE schemes). Thus one interpretation of our results is that anonymity does not "cost" anything in an asymptotic sense. Naturally, the constants matter in practice, and reducing the constant in the ciphertext size for ANOBE to something closer to what can be achieved in the non-anonymous setting is a major open problem. However, we reiterate that reducing the *true* size of ciphertexts below linear in n in either the anonymous or non-anonymous setting is impossible.

2 Anonymous Broadcast Encryption

We define a model of public-key Broadcast Encryption, where algorithms are specified to allow for anonymity (similarly to [5]) and they are general enough to include the identity-based variant of BE introduced in [17].

Definition 1. *Let $U = \{1, ..., n\}$ be the universe of users. A* broadcast encryption (BE) scheme *is defined by four algorithms and has associated message space $\mathcal{MSP}$ and ciphertext space $\mathcal{CSP}$.*

`BE.Setup`(λ, n)**:** *This algorithm takes as input the security parameter λ and the number of users in the system n. It outputs a master public key BE-MPK and a master secret key BE-MSK.*

`BE.Key-Gen`(BE-MPK, BE-MSK, i)**:** *This algorithm takes as input BE-MPK, BE-MSK and an index $i \in U$ and outputs the private key sk_i for user i.*

`BE.Enc`(BE-MPK, m, S)**:** *This algorithm takes as input BE-MPK, a message $m \in \mathcal{MSP}$ and a subset $S \subseteq U$, the broadcast target set. It outputs a ciphertext $c \in \mathcal{CSP}$.*

`BE.Dec`(BE-MPK, sk_i, c)**:** *This algorithm takes as input BE-MPK, a private key sk_i and a ciphertext $c \in \mathcal{CSP}$. It outputs either a message $m \in \mathcal{MSP}$ or a failure symbol $\perp$.*

For all $S \subseteq U$ and $i \in U$, if $c =$ `BE.Enc`$(BE\text{-}MPK, m, S)$ and sk_i is the private key for $i \in S$, then `BE.Dec`$(BE\text{-}MPK, sk_i, c) = m$ with overwhelming probability.

We observe that this definition no longer requires the set S as an input to the decryption algorithm. This is crucial in developing the notion of anonymous broadcast encryption (ANOBE), for which we next provide an appropriate security model for the case of *adaptive* adversaries.

Definition 2. *We define the ANO-IND-CCA security game for BE as follows.*
Setup. *The challenger $\mathcal{C}$ runs `BE.Setup`(λ, n) to generate the master key pair $(BE\text{-}MPK, BE\text{-}MSK)$ and gives BE-MPK to the adversary $\mathcal{A}$.*
Phase 1. *$\mathcal{A}$ can issue queries to a private key extraction oracle for any index $i \in U$. The oracle will respond by returning $sk_i =$ `BE.Key-Gen`$(BE\text{-}MPK, BE\text{-}MSK, i)$. $\mathcal{A}$ can also issue decryption queries of the form (c, i), where $i \in U$, and the oracle will return the decryption `BE.Dec`$(BE\text{-}MPK, sk_i, c)$.*
Challenge. *$\mathcal{A}$ selects two equal-length messages m_0, $m_1 \in \mathcal{MSP}$ and two distinct sets $S_0, S_1 \subseteq U$ of users. We require that S_0 and S_1 be of equal size and also impose the restriction that $\mathcal{A}$ has not issued key queries for any $i \in S_0 \triangle S_1 = (S_0 \setminus S_1) \cup (S_1 \setminus S_0)$. Further, if there exists an $i \in S_0 \cap S_1$ for which $\mathcal{A}$ has queried the key, then we require that $m_0 = m_1$. The adversary $\mathcal{A}$ passes m_0, m_1 and S_0, S_1 to $\mathcal{C}$. The latter picks a random bit $b \in \{0, 1\}$ and computes $c^* =$ `BE.Enc`$(BE\text{-}MPK, m_b, S_b)$ which is returned to $\mathcal{A}$.*
Phase 2. *$\mathcal{A}$ continues to make queries to the private key extraction oracle with the restrictions that $i \notin S_0 \triangle S_1$ and that, if $i \in S_0 \cap S_1$, then $m_0 = m_1$. $\mathcal{A}$ may continue issuing decryption queries (c, i) with the restriction that if $c = c^*$ then either $i \notin S_0 \triangle S_1$ or $i \in S_0 \cap S_1$ and $m_0 = m_1$.*
Guess. *The adversary outputs its guess b' for b.*

Definition 3. *We say that a BE scheme is* anonymous and semantically secure against chosen-ciphertext attacks *(ANO-IND-CCA) if all polynomial-time adaptive adversaries $\mathcal{A}$ have at most negligible advantage in the above game, where $\mathcal{A}$'s advantage is defined as $Adv_{\mathcal{A},BE}^{ANO\text{-}IND\text{-}CCA}(\lambda) = \left|\Pr[b' = b] - \frac{1}{2}\right|$.*

Like the definition of [5], Definition 2 does not require the ANOBE ciphertext to hide the number of receivers. However, specific schemes (such as the one in Section 3.1) can also conceal the cardinality of S.

We will next show that this notion is indeed *feasible* by presenting a generic construction that relies solely on the existence of IND-CCA secure PKE schemes. We will then improve its performance by giving alternative generic constructions whose underlying primitives require additional security properties.

3 Generic Constructions for ANOBE from PKE

3.1 ANOBE from Minimal Assumptions

Since our aim is to provide a formal treatment of anonymous broadcast encryption, we begin by showing that ANOBE *can be achieved*. Indeed, by simply assuming the existence of an IND-CCA secure PKE scheme we can construct an ANOBE scheme as follows.

Let $\pi^{\mathrm{pke}} = (\mathsf{Gen}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ be a PKE scheme with message space $\mathcal{M} = \{0,1\}^m$. Here, algorithm Gen takes as input a security parameter and outputs public parameters par, used by KeyGen to generate a key pair (pk, sk). Let $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ be a one-time signature scheme consisting of a key generation algorithm $\mathcal{G}$, a signing algorithm $\mathcal{S}$ and a verification algorithm $\mathcal{V}$. We assume that the key space of Σ is $\mathcal{K} = \{0,1\}^v$, for some $v \in \mathsf{poly}(\lambda)$. We use π^{pke} and Σ to generically instantiate a BE scheme, with message space $\{0,1\}^{m-v}$. In the description hereafter, we include the symbol ε as a valid but distinguished message in $\{0,1\}^{m-v}$: in other words, all the messages that receivers accept as legal plaintexts are different from ε.

$\mathtt{BE.Setup}(\lambda, n)$**:** Generate $par \leftarrow \mathsf{Gen}(\lambda)$ and, for $i = 1$ to n, generate $(sk_i, pk_i) \leftarrow \mathsf{Keygen}(par)$. The master private key is BE-MSK $= \{sk_i\}_{i=1}^n$ and the master public key consists of BE-MPK $= \big(par,\ \Sigma,\ \{pk_i\}_{i=1}^n\big)$.

$\mathtt{BE.Key\text{-}Gen}$(BE-MPK, BE-MSK, i)**:** parse the master secret key BE-MSK as $\{sk_i\}_{i=1}^n$ and output sk_i.

$\mathtt{BE.Enc}$(BE-MPK, M, S)**:** to encrypt M for a receiver set $S \subseteq \{1, \ldots, n\}$, generate a one-time key pair $(\mathsf{SK}, \mathsf{VK}) \leftarrow \mathcal{G}(\lambda)$. For each $j = 1$ to n, compute $C_j = \mathsf{Encrypt}(par, pk_j, M||\mathsf{VK})$ if $j \in S$ and $C_j = \mathsf{Encrypt}(par, pk_j, \varepsilon||\mathsf{VK})$ if $j \notin S$. Finally, output $C = \big(C_1, \ldots, C_n, \sigma\big)$, where $\sigma = \mathcal{S}\big(\mathsf{SK}, (C_1, \ldots, C_n)\big)$.

$\mathtt{BE.Dec}$(BE-MPK, sk_i, C)**:** given the ANOBE ciphertext $C = \big(C_1, \ldots, C_n, \sigma\big)$, compute $M' = \mathsf{Decrypt}(sk_i, C_i)$. If $M' \neq \perp$, parse M' as $M' = M||\mathsf{VK}$ for some bitstrings $M \in \{0,1\}^{m-v}$ and $\mathsf{VK} \in \{0,1\}^v$. Then, if it holds that $\mathcal{V}\big(\mathsf{VK}, (C_1, \ldots, C_n), \sigma\big) = 1$ and $M \neq \varepsilon$ return M. Otherwise, output $\perp$.

The correctness follows directly from the correctness of π^{pke} and Σ. This construction is reminiscent of generic constructions of chosen-ciphertext-secure multiple encryption [20] and it is easily seen to yield a secure ANOBE. A proof of the following theorem is available in the full version of the paper [30].

Theorem 1. *Let π^{pke} be an IND-CCA secure PKE scheme and let Σ be a strongly unforgeable one-time signature scheme. The BE scheme constructed above is ANO-IND-CCA secure against adaptive adversaries.*

We have described an ANOBE scheme from minimal assumptions. We note that encryption time is linear in n but decryption is performed in *constant* time, since a user simply selects the ciphertext component to decrypt according to its index. However, the ciphertext size is *linear* in n, as we encrypt to each user in the universe. It is desirable to improve on this and achieve a realization of ANOBE with more compact ciphertexts.

We will next see how to modify this first generic construction, obtaining an ANOBE scheme whose ciphertext size is linear in the size of the *target set* S.

3.2 Adaptively Secure ANOBE from Robust, Anonymous PKE

A simple solution to the broadcast problem is to encrypt the message under the public key of each user in the privileged set. This naive approach, so often discarded in most BE literature due to efficiency reasons, turns out to provide another generic construction for ANOBE, which differs from the previous one as now we deploy a public-key encryption scheme only to encrypt the *message* to the users *in the target set*.

For this approach, the underlying PKE scheme has to be key-private (or IK secure [6]), in that the ciphertext does not leak under which public key it was created. We also require the PKE scheme to be weakly robust, in the sense of [2], not only for correctness but also for consistency in the CCA security proof simulation. This property can be generically achieved [2] for any PKE scheme using a simple redundancy-based transformation.

This is essentially the construction that was already suggested by Barth, Boneh and Waters [5]. We now prove that it is actually *adaptively* secure, rather than just statically secure, as was established in [5].

Let $\pi^{\text{pke}} = (\mathsf{Gen}, \mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ be a PKE scheme and $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ be a one-time signature. Our ANOBE scheme, $\text{ANOBE}^{\pi^{\text{pke}}, \Sigma}$, is as follows.

`BE.Setup`(λ, n)**:** Run $\mathsf{Gen}(\lambda, n)$ to obtain public parameters par. For $i = 1$ to n, run $\mathsf{Keygen}(par)$ to generate (sk_i, pk_i). The master private key is BE-MSK $= \{sk_i\}_{i=1}^n$ and the master public key is BE-MPK $= \big(par,\ \Sigma,\ \{pk_i\}_{i=1}^n\big)$.

`BE.Key-Gen`(BE-MPK, BE-MSK, i)**:** given BE-MSK $= \{sk_i\}_{i=1}^n$, output sk_i.

`BE.Enc`(BE-MPK, M, S)**:** to encrypt M for a receiver set $S = \{i_1, \ldots, i_\ell\} \subseteq \{1, \ldots, n\}$ of size $\ell = |S|$, generate a signature key pair $(\mathsf{SK}, \mathsf{VK}) \leftarrow \mathcal{G}(\lambda)$. For $j = 1$ to ℓ, compute $C_j = \mathsf{Encrypt}(par, pk_{i_j}, M||\mathsf{VK})$. The ANOBE ciphertext is $C = \big(\mathsf{VK}, C_{\tau(1)}, \ldots, C_{\tau(\ell)}, \sigma\big)$, where $\sigma = \mathcal{S}\big(\mathsf{SK}, C_{\tau(1)}, \ldots, C_{\tau(\ell)}\big)$ and $\tau : \{1, \ldots, \ell\} \to \{1, \ldots, \ell\}$ is a random permutation.

`BE.Dec`(BE-MPK, sk_i, C)**:** parse C as a tuple $\big(\mathsf{VK}, C_1, \ldots, C_\ell, \sigma\big)$. Return $\perp$ if $\mathcal{V}\big(\mathsf{VK}, C_1, \ldots, C_\ell, \sigma\big) = 0$. Otherwise, repeat these steps for $j = 1$ to ℓ.

1. Compute $M' = \mathsf{Decrypt}(sk_i, C_j)$. If $M' \neq \perp$ and can moreover be parsed as $M' = M||\mathsf{VK}$ for some M of appropriate length, return M.
2. If $j = \ell$ output $\perp$.

The correctness of ANOBE$^{\pi^{\mathrm{pke}},\Sigma}$ follows directly from the correctness and weak robustness of π^{pke}.

Theorem 2. *ANOBE$^{\pi^{\mathrm{pke}},\Sigma}$ is adaptively ANO-IND-CCA secure assuming that: (i) π^{pke} is key-private and IND-CCA (AI-CCA) secure and weakly robust under chosen-ciphertext attacks (as defined in [2]); (ii) Σ is a strongly unforgeable one-time signature scheme.*

In our proof (given in the full version of the paper) we make use of a sequence of hybrid arguments where ciphertext components are gradually modified at each step and each hybrid argument requires the reduction to guess upfront the identity of an uncorrupted user.

In terms of efficiency, from this construction we will obtain secure ANOBE schemes with typically very small (constant) private key storage requirements and ciphertexts which are $|S|$ times the size of the ciphertext of the underlying PKE scheme. Encryption and decryption have both cost linear in the size of S.

If we look at recent efficient instantiations of BE, for example that of Gentry-Waters [24], we have private keys whose size is linear in the number of users, and ciphertexts which consist of n bits plus 3 group elements (if we include the cost of transmitting a description of S as part of the ciphertext). It is clear that in general the solution of [24] is more efficient in terms of ciphertext size. The key point though is that it is not anonymous.

4 Generic Construction for ANOBE from IBE

An IBE scheme I typically consists of four algorithms $(\mathsf{Setup}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Dec})$, where Setup and KeyExt are run by a trusted authority (TA). Our construction uses a multi-TA IBE scheme $I' = (\mathsf{CommonSetup}, \mathsf{TASetup}, \mathsf{KeyDer}, \mathsf{Enc}', \mathsf{Dec}')$ as formalized in [34]. We recall from [34] that $\mathsf{CommonSetup}$, on input the security parameter, outputs the system's parameters par and a set of labels of the TAs in the system, and that $\mathsf{TASetup}$, on input par, outputs a master public key mpk and a master secret key msk. This algorithm is randomized and executed independently for each TA in the system. The remaining algorithms are as per a normal IBE scheme. For this primitive we consider the notion of TA anonymity, as defined in [34], which formally models the inability of the adversary to distinguish two ciphertexts corresponding to the same message and identity, but created using different TA master public keys. An example of a TA-anonymous IBE scheme is the multi-TA version of Gentry's IBE [23] developed in [35].

Now, let $I' = (\mathsf{CommonSetup}, \mathsf{TASetup}, \mathsf{KeyDer}, \mathsf{Enc}', \mathsf{Dec}')$ be a weakly robust (in the sense of a definition of robustness deferred to the full version of the paper), multi-TA IBE scheme and let $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ be a signature scheme. We will use I' and Σ to generically instantiate a BE scheme in the following way.

`BE.Setup`(λ, n)**:** Run $\mathsf{CommonSetup}$ on input of $\lambda \in \mathbb{N}$ to obtain the system's parameters par. Run $\mathsf{TASetup}(par)$ n times to obtain n distinct master key pairs $\{mpk_i, msk_i\}_{i \in U}$. Return the par, Σ and n public keys $\{mpk_i\}_{i \in U}$.

$\mathtt{BE.Key\text{-}Gen}(par, \lambda, i)$**:** Return msk_i, the secret key corresponding to the public key mpk_i of user i.

$\mathtt{BE.Enc}(par, M, S)$**:** Run $\mathcal{G}$ to obtain a one-time signature key pair $(\mathsf{SK}, \mathsf{VK})$. For each $i \in S$ run $\mathsf{Enc}'(mpk_i, M, \mathsf{VK})$ to obtain ciphertext C_i. The ANOBE ciphertext is $C = \big(\mathsf{VK}, C_{\tau(1)}, \ldots, C_{\tau(\ell)}, \sigma\big)$, where $\sigma = \mathcal{S}\big(\mathsf{SK}, C_{\tau(1)}, \ldots, C_{\tau(\ell)}\big)$ and $\tau : \{1, \ldots, \ell\} \to \{1, \ldots, \ell\}$ is a random permutation.

$\mathtt{BE.Dec}(par, msk_i, C)$**:** Parse C as $\big(\mathsf{VK}, C_1, \ldots, C_\ell, \sigma\big)$. If $\mathcal{V}\big(\mathsf{VK}, C_1, \ldots, C_\ell, \sigma\big) = 0$, return $\perp$. Otherwise, compute $sk_{i_{\mathsf{VK}}} = \mathsf{KeyDer}(mpk_i, msk_i, \mathsf{VK})$ and repeat the following steps for $j = 1$ to ℓ.

1. Compute $M' = \mathsf{Dec}'(mpk_i, sk_{i_{\mathsf{VK}}}, C_j)$. If $M' \neq \perp$ and can moreover be parsed as $M' = M||\mathsf{VK}$ for some M of appropriate length, return M.
2. If $j = \ell$ output $\perp$.

The correctness of the BE scheme follows directly from the correctness and the weak robustness of the IBE scheme I' used to construct it.

If instantiated with the multi-TA version of Gentry's IBE scheme [23,35] (which can be made weakly robust simply by applying the transform in [2]), this construction yields very short constant size private keys (just one element in $\mathbb{Z}_p^*$) and ciphertexts consisting of roughly $3 \cdot |S|$ group elements ($|S|$ in $\mathbb{G}$ and $2 \cdot |S|$ in $\mathbb{G}_T$) plus a signature and a verification key. Encryption and decryption have both cost linear in the size of S.

Theorem 3. *Let I' be a TA-anonymous, sID-IND-CPA secure IBE scheme and let Σ be a strongly unforgeable one-time signature. Then, the above BE scheme is adaptively ANO-IND-CCA secure.*

We give some intuition for the proof. We observe that, in [35], the authors apply a modified version of the Canetti-Halevi-Katz (CHK) transform [13] using the same primitives as our generic construction to obtain a key-private IND-CCA PKE scheme. We introduce further modifications to build a BE scheme achieving ANO-IND-CCA security. The idea is that, within this transform, we encrypt m for the *same* identity VK under the $|S|$ different public keys. We then sign all ciphertexts and append the verification key VK (note that this signature binds all these ciphertexts together). Upon decryption, a user verifies the signature against VK and, if valid, proceeds to derive the decryption key for identity VK by running the IBE key-extraction algorithm on input his private key. By similar arguments to those in [13] and [35], and by applying techniques analogous to those proving adaptive security in Theorem 2, we can show that adaptive ANO-IND-CCA security is achieved.

5 Efficient Decryption in the Standard Model

The generic constructions for ANOBE presented in Section 3.2 and 4 both suffer from linear time decryption. This arises from the fact that users do not know which ciphertext component is intended for them, and hence will have to perform an average of $|S|/2$ decryptions before recovering the message. Clearly this

procedure is quite cumbersome. We now present a technique which achieves *constant* time decryption in the standard model. We make use of a new primitive, called tag-based *anonymous hint systems*, for which we provide a definition, the relevant security models and a concrete instantiation.

5.1 Tag-Based Anonymous Hint Systems

A tag-based anonymous *hint* system is a tag-based encryption scheme [27] allowing to generate weak forms of encryption under a tag t and a public key pk. The result of the process consists of a *value* U and a *hint* H. The pair (U, H) should be pseudo-random (in particular, hints generated under two distinct public keys should be indistinguishable) when only the public key pk is available. Also, the private key sk makes it possible to check whether a given hint H is valid w.r.t. a tag t. A value-hint pair can be seen as an extractable commitment to a public key. Formally, such a system is defined in terms of the following algorithms.

$\mathtt{Keygen}(\mathtt{cp})$: takes as input a set of common public parameters $\mathtt{cp}$ and outputs a key pair (sk, pk). We assume that $\mathtt{cp}$ specifies a randomness space $\mathcal{R}^h$ and a space $\mathcal{T}^h$ of acceptable tags for the scheme.

$\mathtt{Hint}(\mathtt{cp}, t, pk, r)$**:** is a deterministic algorithm taking as input common public parameters $\mathtt{cp}$, a public key pk, a tag t and random coins $r \in_R \mathcal{R}^h$. It outputs pair (U, H) consisting of a value U and a hint H. It is required that U only depends on the random coins r and not on pk.

$\mathtt{Invert}(\mathtt{cp}, sk, t, U)$**:** is a deterministic "inversion" algorithm taking as input a value U, a tag t and a private key sk. It outputs either a hint H or $\perp$ if U is not in the appropriate domain.

Correctness requires that, for any pair $(sk, pk) \leftarrow \mathtt{Keygen}(\lambda)$ and any possible random coins r, if $(U, H) \leftarrow \mathtt{Hint}(t, pk, r)$, then $\mathtt{Invert}(\mathtt{cp}, sk, t, U) = H$.

Although hint systems bear similarities with tag-KEMs, as formalized by Abe *et al.* [3], the two primitives are different and incomparable. In the tag-KEM syntax, the symmetric "session key" is chosen first and it does not depend on the tag. In hint schemes, the syntax requires to choose a pair (U, H), where U does not depend on pk but the session key H can depend on both pk and the tag (this is what happens in the construction we give). The security definitions are also different since, in Definition 4 hereafter, there is no inversion oracle (that would return H given U and t) but only a verification oracle that determines if (U, H, t) form a valid triple with respect to public keys pk_0 and pk_1.

In certain aspects, hint schemes are reminiscent of extractable hash proof systems [38] but there are several differences. In [38], in addition to the value that we call U, the random coins allowing to compute U are used to compute a witness S such that (U, S) satisfies some relation. From U, the element S is also computable using the private key and the value that we call H (which is termed "hash value" in [38]). At the same time, S should be infeasible to compute without the private key or the random coins used to sample U. Hint schemes are different in that they rather require the hardness of computing H

from U without the private key. In addition, tag-based hints require that it be hard to decide if a pair (U, H) is valid for a certain tag $t^\star$ (*i.e.*, to decide if $H = \mathtt{Invert}(\mathsf{cp}, sk, t^\star, U)$) even with access to a decision oracle for tags $t \neq t^\star$.

Definition 4. *A tag-based hint system* $(\mathtt{Keygen}, \mathtt{Hint}, \mathtt{Invert})$ *is* anonymous *if no PPT adversary has non-negligible advantage in the following game:*

1. *On input of common public parameters* cp*, the adversary* $\mathcal{A}$ *chooses a tag* $t^\star$ *and sends it to the challenger.*
2. *The challenger generates two key pairs* $(sk_0, pk_0) \leftarrow \mathtt{Keygen}(\lambda)$*,* $(sk_1, pk_1) \leftarrow \mathtt{Keygen}(\lambda)$ *and gives* pk_0, pk_1 *to* $\mathcal{A}$*.*
3. *On polynomially-many occasions,* $\mathcal{A}$ *adaptively invokes a verification oracle on value-hint-tag triples* (U, H, t) *such that* $t \neq t^\star$*. The challenger replies by returning bits* $(d_0, d_1) \in \{0,1\}^2$ *where* $d_0 = 1$ *if and only if* $H = \mathtt{Invert}(\mathsf{cp}, sk_0, t, U)$ *and* $d_1 = 1$ *if and only if* $H = \mathtt{Invert}(\mathsf{cp}, sk_1, t, U)$*.*
4. *When* $\mathcal{A}$ *decides to enter the challenge phase, the challenger flips a binary coin* $b \xleftarrow{\$} \{0,1\}$ *and chooses other random coins* $r^\star \xleftarrow{\$} \mathcal{R}^h$*. It outputs* $(U^\star, H^\star) = \mathtt{Hint}(\mathsf{cp}, t^\star, pk_b, r^\star)$*.*
5. $\mathcal{A}$ *makes further queries but is not allowed to make queries involving the target tag* $t^\star$*.*
6. $\mathcal{A}$ *outputs a bit* $b' \in \{0,1\}$ *and wins if* $b' = b$*.*

As usual, $\mathcal{A}$*'s advantage is the distance* $\mathbf{Adv}^{\text{anon-hint}}(\mathcal{A}) = |\Pr[b' = b] - 1/2|$.

Definition 5. *A tag-based hint system* $(\mathtt{Keygen}, \mathtt{Hint}, \mathtt{Invert})$ *is strongly* robust *if no PPT adversary* $\mathcal{A}$ *has non-negligible advantage in the following game, where* $\mathcal{A}$*'s advantage is its probability of success.*

1. *The challenger chooses public parameters* cp *and generates pairs* $(sk_0, pk_0) \leftarrow \mathtt{Keygen}(\lambda)$*,* $(sk_1, pk_1) \leftarrow \mathtt{Keygen}(\lambda)$*. It gives* cp *and* pk_0, pk_1 *to* $\mathcal{A}$*.*
2. $\mathcal{A}$ *invokes a verification oracle on arbitrary value-hint-tag triples* (U, H, t)*. The challenger replies by returning bits* $(d_0, d_1) \in \{0,1\}^2$ *where* $d_0 = 1$ *if and only if* $H = \mathtt{Invert}(\mathsf{cp}, sk_0, t, U)$ *and* $d_1 = 1$ *if and only if* $H = \mathtt{Invert}(\mathsf{cp}, sk_1, t, U)$*.*
3. $\mathcal{A}$ *outputs a triple* $(U^\star, H^\star, t^\star)$ *and wins if* $H^\star = \mathtt{Invert}(\mathsf{cp}, sk_0, t^\star, U^\star) = 1$ *and* $H^\star = \mathtt{Invert}(\mathsf{cp}, sk_1, t^\star, U^\star) = 1$*.*

Analogously to the PKE case [2], weak robustness is defined for tag-based hint schemes by letting the adversary simply make a challenge request in step 3. The challenger then chooses a tag $t^\star$ as well as random coins $r^\star$, generates a value-hint pair $(U^\star, H^\star) = \mathtt{Hint}(\mathsf{cp}, t^\star, pk_0, r^\star)$ and $\mathcal{A}$ wins if $H^\star = \mathtt{Invert}(\mathsf{cp}, sk_1, t^\star, U^\star) = 1$. Weak robustness will be sufficient for our purposes but the scheme hereafter is also strongly robust assuming that the discrete logarithm assumption holds in $\mathbb{G}$.

To show that this newly defined primitive is indeed feasible, we give an example of an anonymous hint system based on the DDH assumption and the CCA-secure public key encryption scheme described in [14].

Let the common public parameters $\mathsf{cp} = \{\mathbb{G}, p, g\}$ consist of a group $\mathbb{G}$ of prime order $p > 2^\lambda$ with a generator $g \in_R \mathbb{G}$. We assume that tags are elements of $\mathcal{T}^h = \mathbb{Z}_p^*$ and that the randomness space is $\mathcal{R}^h = \mathbb{Z}_p^*$.

$\mathtt{Keygen}(\mathsf{cp})$**:** chooses random $x_1, x_2, y_1, y_2 \xleftarrow{\$} \mathbb{Z}_p^*$ and computes $X_i = g^{x_i}$ and $Y_i = g^{y_i}$ for each $i \in \{1, 2\}$. The public key is $pk = \big(X_1, X_2, Y_1, Y_2\big)$ and the private key is $sk = (x_1, x_2, y_1, y_2)$.

$\mathtt{Hint}(\mathsf{cp}, t, pk, r)$**:** given $pk = (\mathbb{G}, p, g, X_1, X_2, Y_1, Y_2)$, return $\perp$ if $r \notin \mathcal{R}^h = \mathbb{Z}_p^*$. Otherwise, compute (U, H) as

$$U = g^r, \qquad H = (V, W) = \big((X_1^t X_2)^r,\ (Y_1^t Y_2)^r\big).$$

$\mathtt{Invert}(\mathsf{cp}, sk, t, U)$**:** return $\perp$ if $U \notin \mathbb{G}$. Otherwise, parse the private key sk as $(x_1, x_2, y_1, y_2) \in (\mathbb{Z}_p^*)^4$ and output $H = (V, W) = (U^{t \cdot x_1 + x_2}, U^{t \cdot y_1 + y_2})$

In the full version of the paper, we prove that the scheme provides anonymity in the sense of Definition 4 under the DDH assumption and strong robustness (in the sense of Definition 5) under the discrete logarithm assumption.

5.2 ANOBE with Efficient Decryption

Let $\pi^{\mathrm{hint}} = (\mathtt{Keygen}, \mathtt{Hint}, \mathtt{Invert})$ be an anonymous hint system with its set of common public parameters cp. Let $\pi^{\mathrm{pke}} = (\mathsf{Gen}, \mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ be a PKE scheme and $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ be a signature scheme.

$\mathtt{BE.Setup}(\lambda, n)$**:** Obtain $(par) \leftarrow \mathsf{Gen}(\lambda)$ and, for each $i \in \{1, \ldots, n\}$, and generate encryption key pairs $(\tilde{sk}_i, \tilde{pk}_i) \leftarrow \pi^{\mathrm{pke}}.\mathsf{Keygen}(par)$ as well as hint key pairs $(sk_i^h, pk_i^h) \leftarrow \pi^{\mathrm{hint}}.\mathtt{Keygen}(\mathsf{cp})$. The master private key consists of BE-MSK $= \{\tilde{sk}_i, sk_i^h\}_{i=1}^n$ and the master public key is

$$\text{BE-MPK} = \Big(\mathsf{cp}, par,\ \{(\tilde{pk}_i, pk_i^h)\}_{i=1}^n,\ \Sigma\Big).$$

$\mathtt{BE.Key\text{-}Gen}(\text{BE-MPK}, \text{BE-MSK}, i)$**:** parse BE-MSK as $\{\tilde{sk}_i, sk_i^h\}_{i=1}^n$ and output $sk_i = (\tilde{sk}_i, sk_i^h)$.

$\mathtt{BE.Enc}(\text{BE-MPK}, M, S)$**:** given a receiver set $S = \{i_1, \ldots, i_\ell\} \subseteq \{1, \ldots, n\}$ of size $\ell = |S|$ and a message M, generate a one-time signature key pair $(\mathsf{SK}, \mathsf{VK}) \leftarrow \mathcal{G}(\lambda)$. Then, choose random coins $r \xleftarrow{\$} \mathcal{R}^h$ for the hint scheme and compute $(U, H_j) = \pi^{\mathrm{hint}}.\mathtt{Hint}(\mathsf{cp}, \mathsf{VK}, pk_{i_j}^h, r)$ for $j = 1$ to ℓ (recall that the first output U of $\mathtt{Hint}$ does not depend on the public key). For $j = 1$ to ℓ, compute $C_j = \pi^{\mathrm{pke}}.\mathsf{Encrypt}(par, \tilde{pk}_{i_j}, M||\mathsf{VK})$. Choose a random permutation $\tau : \{1, \ldots, \ell\} \to \{1, \ldots, \ell\}$ and set the final ciphertext as

$$C = \big(\mathsf{VK}, U, (H_{\tau(1)}, C_{\tau(1)}), \ldots, (H_{\tau(\ell)}, C_{\tau(\ell)}), \sigma\big),$$

where $\sigma = \mathcal{S}\big(\mathsf{SK}, U, (H_{\tau(1)}, C_{\tau(1)}), \ldots, (H_{\tau(\ell)}, C_{\tau(\ell)})\big)$.

$\mathtt{BE.Dec}(\text{BE-MPK}, sk_i, C)$**:** on input of $C = \big(\mathsf{VK}, U, (H_1, C_1), \ldots, (H_\ell, C_\ell), \sigma\big)$ and $sk_i = (\tilde{sk}_i, sk_i^h)$, return $\perp$ if $\mathcal{V}\big(\mathsf{VK}, U, (H_1, C_1), \ldots, (H_\ell, C_\ell), \sigma\big) = 0$ or if U is not in the appropriate space defined by π^{hint}. Otherwise, compute $H = \pi^{\mathrm{hint}}.\mathtt{Invert}(\mathsf{cp}, sk_i^h, \mathsf{VK}, U)$. If $H \neq H_j$ for all $j \in \{1, \ldots, \ell\}$, return $\perp$. Otherwise, let j be the smallest index such that $H = H_j$ and compute $M' = \pi^{\mathrm{pke}}.\mathsf{Decrypt}(\tilde{sk}_i, C_j)$. If M' can be parsed as $M' = M||\mathsf{VK}$ for some M of appropriate length, return M. Otherwise, output $\perp$.

The correctness of this scheme follows directly from the correctness and weak robustness of its component schemes π^{hint} and π^{pke}.

The following security result is proved in the full version of the paper.

Theorem 4. *The above construction is adaptively ANO-IND-CCA secure if (i) π^{hint} is anonymous; (ii) π^{pke} is AI-CCA secure and weakly robust under chosen-ciphertext attacks; (iii) Σ is a strongly unforgeable one-time signature.*

In [5] a technique to speed up decryption was presented. The scheme of [5] can be seen as using a hint scheme where tags are empty strings and pairs (U, H_j) consist of $U = g^r$ and $H_j = H(X_{i_j}^r)$, where H is a random oracle and $X_{i_j} \in \mathbb{G}$ is the public key of the hint scheme. In the present context, it is tempting to believe that simple hints of the form $X_{i_j}^r$ suffice to achieve efficient decryption in the standard model. Indeed, one step of the proof consists of a DDH-based transition from one hybrid game to another and, during that specific transition, the simulator $\mathcal{B}$ could simply handle all decryption queries using the private keys $\{\widetilde{sk}_i\}_{i=1}^n$ in the underlying encryption scheme since it knows them all. For reasons that will become apparent in the proof of a key lemma for Theorem 4 below, this does not suffice. The reason is that, the adversary can issue decryption queries where $(g, U = g^r, X_{i_j}, H_{i_j} = X_{i_j}^{r'})$ does *not* form a Diffie-Hellman tuple. In this case, the answer of the simulator would differ from that of the real decryption procedure in the chosen-ciphertext scenario: more precisely, the simulation could accept a ciphertext that would be rejected by a real decryption.

In [5], Barth, Boneh and Waters addressed this problem using a random oracle and the Gap Diffie-Hellman assumption [33]: each hint was of the form $H_j = H(X_{i_j}^r)$, where H is the random oracle. By invoking the DDH-oracle at each random oracle query, the simulator was able to figure out which ciphertext components had to be decrypted so as to perfectly emulate the real decryption algorithm. Here, we address this issue in the standard model using the tag-based anonymous hint primitive.

It is convenient to instantiate the above construction by combining our DDH-based hint scheme with an encryption scheme based on the same assumption such as the Cramer-Shoup cryptosystem. Interestingly both schemes can be instantiated using the same DDH-hard cyclic group. Considering efficiency, it is moreover possible to recycle the group element g^r of the hint system and simultaneously use it as part of a Cramer-Shoup ciphertext. In the security proof, everything goes through with these optimizations.

6 Shortening Ciphertexts with Randomness Re-use

This section considers *randomness re-use* [7,4], which is a powerful tool providing computational and bandwidth savings, as a technique to optimize ANOBE schemes. In [7], Bellare *et al.* introduce a property, called *reproducibility*, providing a condition under which randomness re-use is secure. We define the notion of *key-less reproducibility*, which is better suited for the anonymity setting.

Definition 6. *Let* $\pi^{\mathrm{pke}} = (\mathsf{Gen}, \mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *be a PKE scheme. Let* $\mathcal{M}$ *and* $\mathcal{R}$ *be the message and randomness space of* π^{pke}*. Let* R *be an algorithm that takes as input the public parameters, a ciphertext, another random message and a key pair* (sk, pk)*, and outputs a ciphertext. Consider the experiment:*

$\mathbf{Exp}^{\mathsf{KLR}}_{\pi^{\mathrm{pke}},R}(\lambda)$
$(par) \xleftarrow{\$} \mathsf{Gen}(\lambda)$
$(pk, sk) \xleftarrow{\$} \mathsf{Keygen}(par)$
$m \xleftarrow{\$} \mathcal{M}$; $r \xleftarrow{\$} \mathcal{R}$
$c = \mathsf{Encrypt}(pk, m; r)$
$(pk', sk') \xleftarrow{\$} \mathsf{Keygen}(par)$
$m' \xleftarrow{\$} \mathcal{M}$
return 1 **if** $\mathsf{Encrypt}(par, pk', m'; r) = R(par, c, m', pk', sk')$ **and** 0 **otherwise.**

π^{pke} *is* key-less reproducible *if, for any* λ*, there is a PPT algorithm* R *such that the above experiment outputs 1 with probability 1.*

We note that this definition differs from the one in [7] since the algorithm R does not take pk (the public key under which c was created) as an input. Indeed, this is a crucial difference which allows extending the notion of reproducibility to the context where anonymity is required. We now reconsider the generic construction for ANOBE presented in Section 3.2.

Let $\pi^{\mathrm{pke}} = (\mathsf{Gen}, \mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ be a key-less reproducible PKE scheme and let $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ be a one-time signature. We call $\mathrm{ANOBE}_{rr}^{\pi^{\mathrm{pke}},\Sigma}$ the scheme constructed from Σ and π^{pke} as follows.

`BE.Setup`, `BE.Key-Gen`, `BE.Dec` are as in Section 3.2.

`BE.Enc`(BE-MPK, M, S)**:** to encrypt M for a receiver set $S = \{i_1, \ldots, i_\ell\} \subseteq \{1, \ldots, n\}$ of size $\ell = |S|$, generate a signature key pair $(\mathsf{SK}, \mathsf{VK}) \leftarrow \mathcal{G}(\lambda)$. Choose $r \xleftarrow{\$} \mathcal{R}$, where $\mathcal{R}$ is the randomness space of π^{pke}_{par}. Then, for each $j = 1$ to ℓ, compute $C_j = \mathsf{Encrypt}(par, pk_{i_j}, M||\mathsf{VK}; r)$. The final BE ciphertext consists of $C = \big(\mathsf{VK}, C_{\tau(1)}, \ldots, C_{\tau(\ell)}, \sigma\big)$, where $\sigma = \mathcal{S}\big(\mathsf{SK}, C_{\tau(1)}, \ldots, C_{\tau(\ell)}\big)$ and $\tau : \{1, \ldots, \ell\} \to \{1, \ldots, \ell\}$ is a random permutation.

Theorem 5. *Let* $\pi^{\mathrm{pke}} = (\mathsf{Gen}, \mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *be an AI-CCA secure, weakly robust and key-less reproducible PKE scheme. Let* Σ *be a strongly unforgeable one-time signature scheme. Then,* $\mathrm{ANOBE}_{rr}^{\pi^{\mathrm{pke}},\Sigma}$ *is adaptively ANO-IND-CCA secure.*

The proof for Theorem 5 (which is given in the full paper) is analogous to that of Theorem 2, the only difference being the use of algorithm R in the simulation.

We have shown that the key-less reproducibility of a PKE scheme guarantees that randomness can be re-used securely. We can exploit this property to compress the ANOBE ciphertexts and, depending on the concrete instantiation, significantly increase the efficiency of the scheme. More precisely, given

an $\text{ANOBE}_{rr}^{\pi^{\text{pke}},\Sigma}$ ciphertext $C = (\mathsf{VK}, C_{\tau(1)}, \ldots, C_{\tau(\ell)}, \sigma)$, let ccc denote the *common ciphertext components* that may arise in $C_{\tau(1)}, \ldots, C_{\tau(\ell)}$ from sharing randomness across PKE components, *i.e.*,

$$C_{\tau(1)} = (\mathsf{ccc}, \tilde{c}_{\tau(1)}), \ldots, C_{\tau(\ell)} = (\mathsf{ccc}, \tilde{c}_{\tau(\ell)}).$$

The compressed ANOBE ciphertext will be $\tilde{C} = (\mathsf{VK}, \mathsf{ccc}, \tilde{c}_{\tau(1)}, \ldots, \tilde{c}_{\tau(\ell)}, \sigma)$. Upon receipt, the user simply reconstitutes the original ciphertext C and runs `BE.Dec` as usual. We explore instantiations of this idea in the full version.

7 Conclusions and Open Problems

In the context of broadcast encryption the main focus of research to date has been on reducing ciphertext size. Achieving this has entailed sacrificing *all* anonymity properties. Yet we have argued that anonymity is a *fundamental property* to strive for in broadcast encryption. With the aim of highlighting the importance of this overlooked feature, we have formally defined the notion of anonymous broadcast encryption (ANOBE) and given several constructions for this primitive. We have also shown how these constructions can be improved via anonymous hint systems (to optimize decryption performance) and randomness re-use (to reduce the ciphertext size and the computational costs of encryption).

Much work still needs to be done in this area, from improving the efficiency of ANOBE schemes to considering all the additional properties that can be found in standard BE, such as traitor tracing, revocation, dynamism of users joining the system, and realising them in the anonymous setting. There is still a gap between the sizes of ciphertexts in state-of-the-art BE schemes and our ANOBE schemes. This gap is hidden in the constants in an asymptotic evaluation of ciphertext size (when the true size of ciphertexts is measured) but is nevertheless significant in practice. A major challenge, then, is to further reduce the size of ciphertexts in ANOBE, whilst maintaining its full anonymity properties.

Acknowledgements. The work in this paper was supported in part by the European Commission through the ICT programme under contract ICT-2007-216676 ECRYPT II. The work in this paper was sponsored in part by the US Army Research Laboratory and the UK Ministry of Defense and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the US Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. The first author acknowledges the Belgian Fund for Scientific Research (F.R.S.- F.N.R.S.) for his "Collaborateur scientifique" fellowship. The second author was supported by an EPSRC Leadership Fellowship, EP/H005455/1.

References

1. Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., Shi, H.: Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. J. Cryptology 21(3), 350–391 (2008)
2. Abdalla, M., Bellare, M., Neven, G.: Robust Encryption. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 480–497. Springer, Heidelberg (2010)
3. Abe, M., Gennaro, R., Kurosawa, K., Shoup, V.: Tag-KEM/DEM: A New Framework for Hybrid Encryption and A New Analysis of Kurosawa-Desmedt KEM. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 128–146. Springer, Heidelberg (2005)
4. Barbosa, M., Farshim, P.: Randomness Reuse: Extensions and Improvements. In: Galbraith, S.D. (ed.) Cryptography and Coding 2007. LNCS, vol. 4887, pp. 257–276. Springer, Heidelberg (2007)
5. Barth, A., Boneh, D., Waters, B.: Privacy in Encrypted Content Distribution Using Private Broadcast Encryption. In: Di Crescenzo, G., Rubin, A. (eds.) FC 2006. LNCS, vol. 4107, pp. 52–64. Springer, Heidelberg (2006)
6. Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D.: Key-Privacy in Public-Key Encryption. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 566–582. Springer, Heidelberg (2001)
7. Bellare, M., Boldyreva, A., Kurosawa, K., Staddon, J.: Multirecipient encryption schemes: How to save on bandwidth and computation without sacrificing security. IEEE Trans. on Information Theory 53(11), 3927–3943 (2007)
8. Bellare, M., Boldyreva, A., Staddon, J.: Randomness Re-use in Multi-recipient Encryption Schemeas. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 85–99. Springer, Heidelberg (2003)
9. Boneh, D., Gentry, C., Waters, B.: Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005)
10. Boneh, D., Hamburg, M.: Generalized Identity Based and Broadcast Encryption Schemes. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 455–470. Springer, Heidelberg (2008)
11. Boneh, D., Sahai, A., Waters, B.: Fully Collusion Resistant Traitor Tracing with Short Ciphertexts and Private Keys. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 573–592. Springer, Heidelberg (2006)
12. Boyen, X., Waters, B.: Anonymous Hierarchical Identity-Based Encryption (Without Random Oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006)
13. Canetti, R., Halevi, S., Katz, J.: Chosen-Ciphertext Security from Identity-Based Encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
14. Cash, D., Kiltz, E., Shoup, V.: The Twin Diffie-Hellman Problem and Applications. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 127–145. Springer, Heidelberg (2008)
15. Chaum, D.: Security without identification: Transaction systems to make Big Brother obsolete. Commun. ACM 1985 28(10), 1030–1044 (1985)
16. Chaum, D., van Heyst, E.: Group Signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991)
17. Delerablée, C.: Identity-Based Broadcast Encryption with Constant Size Ciphertexts and Private Keys. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 200–215. Springer, Heidelberg (2007)

18. Delerablée, C., Paillier, P., Pointcheval, D.: Fully Collusion Secure Dynamic Broadcast Encryption with Constant-Size Ciphertexts or Decryption Keys. In: Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T. (eds.) Pairing 2007. LNCS, vol. 4575, pp. 39–59. Springer, Heidelberg (2007)
19. Dodis, Y., Fazio, N.: Public Key Broadcast Encryption for Stateless Receivers. In: Feigenbaum, J. (ed.) DRM 2002. LNCS, vol. 2696, pp. 61–80. Springer, Heidelberg (2003)
20. Dodis, Y., Katz, J.: Chosen-Ciphertext Security of Multiple Encryption. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 188–209. Springer, Heidelberg (2005)
21. Fazio, N., Perera, I.M.: Outsider-Anonymous Broadcast Encryption with Sublinear Ciphertexts. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 225–242. Springer, Heidelberg (2012)
22. Fiat, A., Naor, M.: Broadcast Encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994)
23. Gentry, C.: Practical Identity-Based Encryption Without Random Oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
24. Gentry, C., Waters, B.: Adaptive Security in Broadcast Encryption Systems (with Short Ciphertexts). In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 171–188. Springer, Heidelberg (2009)
25. Groth, J.: Efficient Maximal Privacy in Boardroom Voting and Anonymous Broadcast. In: Juels, A. (ed.) FC 2004. LNCS, vol. 3110, pp. 90–104. Springer, Heidelberg (2004)
26. Katz, J., Sahai, A., Waters, B.: Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008)
27. Kiltz, E.: Chosen-Ciphertext Security from Tag-Based Encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006)
28. Kurosawa, K.: Multi-recipient Public-Key Encryption with Shortened Ciphertext. In: Naccache, D., Paillier, P. (eds.) PKC 2002. LNCS, vol. 2274, pp. 48–63. Springer, Heidelberg (2002)
29. Kurosawa, K., Desmedt, Y.: A New Paradigm of Hybrid Encryption Scheme. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 426–442. Springer, Heidelberg (2004)
30. Libert, B., Paterson, K.G., Quaglia, E.A.: Anonymous Broadcast Encryption: Adaptive Security and Efficient Constructions in the Standard Model. Cryptology ePrint Archive: Report 2011/476
31. Mohassel, P.: A Closer Look at Anonymity and Robustness in Encryption Schemes. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 501–518. Springer, Heidelberg (2010)
32. Naor, D., Naor, M., Lotspiech, J.: Revocation and Tracing Schemes for Stateless Receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001)
33. Okamoto, T., Pointcheval, D.: The Gap-Problems: A New Class of Problems for the Security of Cryptographic Schemes. In: Kim, K.-C. (ed.) PKC 2001. LNCS, vol. 1992, pp. 104–118. Springer, Heidelberg (2001)
34. Paterson, K.G., Srinivasan, S.: Security and Anonymity of Identity-Based Encryption with Multiple Trusted Authorities. In: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5209, pp. 354–375. Springer, Heidelberg (2008)

35. Paterson, K.G., Srinivasan, S.: Building Key-Private Public-Key Encryption Schemes. In: Boyd, C., González Nieto, J. (eds.) ACISP 2009. LNCS, vol. 5594, pp. 276–292. Springer, Heidelberg (2009)
36. Phan, D.-H., Pointcheval, D., Strefler, M.: Security Notions for Broadcast Encryption. In: Lopez, J., Tsudik, G. (eds.) ACNS 2011. LNCS, vol. 6715, pp. 377–394. Springer, Heidelberg (2011)
37. Shoup, V.: A proposal for an ISO standard for public key encryption (version 2.1) (2001) (manuscript)
38. Wee, H.: Efficient Chosen-Ciphertext Security via Extractable Hash Proofs. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 314–332. Springer, Heidelberg (2010)
39. Yao, D., Fazio, N., Dodis, Y., Lysyanskaya, A.: ID-based encryption for complex hierarchies with applications to forward security and broadcast encryption. In: ACM-CCS 2004, pp. 354–363. ACM (2004)

Outsider-Anonymous Broadcast Encryption with Sublinear Ciphertexts

Nelly Fazio[1,2] and Irippuge Milinda Perera[2]

[1] The City College of CUNY
fazio@cs.ccny.cuny.edu
[2] The Graduate Center of CUNY
{nfazio,iperera}@gc.cuny.edu

Abstract. In the standard setting of broadcast encryption, information about the receivers is transmitted as part of the ciphertext. In several broadcast scenarios, however, the identities of the users authorized to access the content are often as sensitive as the content itself. In this paper, we propose the first broadcast encryption scheme with sublinear ciphertexts to attain meaningful guarantees of receiver anonymity. We formalize the notion of *outsider-anonymous broadcast encryption* (oABE), and describe generic constructions in the standard model that achieve outsider-anonymity under adaptive corruptions in the chosen-plaintext and chosen-ciphertext settings. We also describe two constructions with enhanced decryption, one under the gap Diffie-Hellman assumption, in the random oracle model, and the other under the decisional Diffie-Hellman assumption, in the standard model.

Keywords: Recipient Privacy, Broadcast Encryption, Anonymous IBE, Subset Cover Framework.

1 Introduction

Conventional encryption provides the means for secret transmission of data in point-to-point communication. The setting of broadcast encryption [1, 2], instead, consists of a *sender*, an insecure unidirectional *broadcast channel*, and a universe of *receivers*. When the sender wants to transmit some digital content, it specifies the set of authorized receivers and creates an encrypted version of the content. A secure broadcast encryption scheme enables legitimate receivers to recover the original content, while ensuring that excluded users just obtain meaningless data, even in the face of collusions.

The intrinsic access control capabilities of broadcast encryption schemes make them a useful tool for many natural applications, spanning from protecting copyrighted content distributed as stored media [3], to managing digital subscriptions to satellite TV, to controlling access in encrypted file systems [4]. Thanks to its versatility, broadcast encryption has received a lot of attention from the crypto research community in recent years (see *e.g.*, [5–14]). The quest, however, has been for ever more efficient solutions in terms of broadcast communication, key storage and encryption/decryption running time. Little attention, instead,

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 225–242, 2012.

has been devoted to the exploration of refined security models that accurately account for the requirements inherent in multi-recipient communication. More specifically, the focus has been on providing assurance for sender-oriented properties, while overlooking the security and privacy concerns of the receivers.

One problem with the above (informal) definition of broadcast encryption is the implicit requirement that, whenever the digital content is encrypted and sent in broadcast, information about the set of authorized receivers is necessary to decrypt it correctly. Therefore, the set of authorized receivers is transmitted as part of the ciphertext. This in particular implies that an eavesdropper, even if unable to recover the message, can still easily discover the identities of the actual receivers of the content. A way to address the privacy implications that result from specifying explicitly the set of authorized receivers in the broadcast is to use ephemeral IDs and to keep secret the table that associates such IDs with the actual receivers. This simple solution, however, would at best result in a pseudonym system, in which it is still possible to link pseudonyms across transmissions and determine whether the same entity is an authorized receiver for two different broadcasts.

Anonymous Broadcast Encryption. An interesting variant of the broadcast encryption setting was proposed by Barth *et al.* in [15]. Therein, the authors introduce the notion of *private* broadcast encryption scheme, explicitly aiming to protect the identities of the receivers. As a proof-of-concept, they also suggest both generic and number-theoretic public-key constructions that do not leak any information about the list of authorized receivers, and are secure in the standard model and in the random oracle model, respectively. The proposed schemes, however, have communication complexity linear in the number of recipients. In [16], Libert *et al.* recently suggested proof techniques to argue the security of (a variant of) the number-theoretic construction of [15] without reliance on random oracles, thus attaining anonymous broadcast encryption with efficient decryption in the standard model. Still, ciphertexts in the resulting construction have length linear in the number of recipients.

Krzywiecki *et al.* presented a private public-key broadcast encryption scheme with communication complexity proportional to the number of revoked users [17]. The security analysis of the proposed solution is rather informal, however, so the security guarantees are at best heuristic.

In [18], Yu *et al.* presented the first *secret-key* multicast scheme with membership anonymity and communication complexity independent of the number of receivers. The proposed scheme not only hides the *identities* of the receives, but also *the number* of users allowed to receive the content. A shortcoming is that only a single user can be revoked for each broadcast.

A promising research line toward practical receiver-anonymous broadcast encryption has recently been started by Jarecki and Liu [19]. The authors propose the first construction of an efficient unlinkable secret handshake scheme, which is an authenticated key exchange protocol providing *affiliation/policy hiding* (*i.e.*, the transmission hides the affiliation and the identities of all parties) and *unlinkability* (*i.e.*, it is impossible to link any two instances of the secret handshake

protocol). The proposed construction can be seen as a *stateful* version of a public-key broadcast encryption scheme, with the additional property of protecting the receivers' identities. Statefulness, however, implies that the key used to encrypt the broadcasts changes for each transmission, and receivers need to keep track of the changes to be able to recover the content.

An interesting trait of the of construction of [19] is that it trades some degree of anonymity for better efficiency: while the receiver's identities are hidden from outsiders, the scheme still allows authorized users to learn information about other members of the receiver set.

Our Contributions. In this paper we propose the first broadcast encryption scheme with sublinear ciphertexts to achieve meaningful guarantees of receiver anonymity. In particular, we formalize the notion of *outsider-anonymous broadcast encryption* (oABE), and describe a generic construction based on any anonymous identity-based encryption scheme (AIBE). Compared with the work of [19], our construction has the advantage of being *stateless*, and with constant public key size.

Additionally, by adapting the techniques of [15], we also obtain an efficient construction with enhanced decryption, where for a given oABE ciphertext, the decryption algorithm executes a single AIBE decryption operation. As outlined in Table 1, by relaxing the anonymity guarantees, our constructions achieve sublinear ciphertexts size and constant public key size.

Table 1. Comparison of the main efficiency parameters of our oABE scheme with [15] and [16]. Our construction trades full anonymity (achieved by [15, 16]) for sublinear ciphertexts and constant public key size.

	Scheme	PK Length	SK Length	CT Length	Decryption Attempts
Regular	BBW06 [15]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	$\mathcal{O}(N-r)$	$\mathcal{O}(N-r)$
	LPQ11 [16]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	$\mathcal{O}(N-r)$	$\mathcal{O}(N-r)$
	Ours (oABE)	$\mathcal{O}(1)$	$\mathcal{O}(\log N)$	$\mathcal{O}\left(r \log\left(\frac{N}{r}\right)\right)$	$\mathcal{O}\left(r \log\left(\frac{N}{r}\right) \log N\right)$
Enhanced	BBW06 [15]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	$\mathcal{O}(N-r)$	1
	LPQ11 [16]	$\mathcal{O}(N)$	$\mathcal{O}(1)$	$\mathcal{O}(N-r)$	1
	Ours (oABE)	$\mathcal{O}(N)$	$\mathcal{O}(\log N)$	$\mathcal{O}\left(r \log\left(\frac{N}{r}\right)\right)$	1

Organization. Section 2 provides a brief review of the Subset Cover Framework [6] and of Anonymous Identity-Based Encryption [20, 21]. The setting of outsider-anonymous broadcast encryption is introduced in Sect. 3. In Sect. 4 we first present generic constructions in the standard model that achieve outsider-anonymity under adaptive corruptions in the chosen-plaintext (Sect. 4.1) and chosen-ciphertext (Sect. 4.2) settings. Next, we describe a CCA-secure construction with enhanced decryption (Sect. 4.3) under the gap Diffie-Hellman assumption in the random oracle model, and outline how to extend it to the standard

model, using the twin-DH-based techniques of [16]. Finally, we outline an optimization for the symmetric-key setting to accommodate storage-sensitive systems and attain constant key storage at the Center, while maintaining efficient decryption and logarithmic storage at the receivers (Sect. 4.4).

2 Background

2.1 The Subset Cover Framework

The *Subset Cover Framework* proposed by Naor et al. [6] is an environment for defining and analyzing the security of revocation schemes in the symmetric key setting, where only the Center can broadcast. The main idea of this framework is to define a collection $\mathcal{S}$ of subsets of the universe of users $\mathcal{U} = \{1, \ldots, N\}$ in the system, and assign each subset $S_j \in \mathcal{S}$ a long-lived key, which is also provided to the users belonging to S_j. When broadcasting a message m, first the Center determines the set of revoked users $\mathcal{R}$, then it finds a set of disjoint subsets $\mathcal{C}$ from the collection $\mathcal{S}$ that "covers" the set $\mathcal{U} \backslash \mathcal{R}$ of receivers, and finally it encrypts the short-lived session key used to encrypt m under all the long-lived keys associated with each subset in $\mathcal{C}$.

In [6], the authors also provide two instantiations of revocation schemes in the Subset Cover framework namely, the Complete Subtree (CS) method and the Subset Difference (SD) method. In the CS method, the key assignment is information-theoretic but the ciphertext is $\mathcal{O}\left(r \log\left(\frac{N}{r}\right)\right)$ long, whereas in the SD method, the ciphertext length is $\mathcal{O}(2r - 1)$ but the key assignment is computational, where r is the number of revoked users. Although the ciphertext length of the CS method is asymptotically bigger than that of the SD method, we are still interested in the CS method due to its information-theoretic key assignment nature, which seems to be crucial for efficiently preserving the anonymity of the receivers.

Complete Subtree (CS) Method. In the Complete Subtree (CS) method as introduced in [6], the N users in the system are represented as the leaves of a full binary tree $\mathcal{T}$. Since this requires N to be a power of 2, dummy users are added to the system in case N is not a power of 2. The collection $\mathcal{S}$ contains all possible complete subtrees of $\mathcal{T}$. More precisely, $\mathcal{S}$ contains a subtree for every node $v_j \in \mathcal{T}$. Since there are $2N - 1$ nodes in $\mathcal{T}$, $|\mathcal{S}| = 2N - 1$.

As for key assignment, every subtree in $\mathcal{S}$ is assigned a long-lived symmetric key which is also made available to the users (leaves) of the given subtree. Since any user u_i, for $1 \leq i \leq N$, is a member of all the subtrees rooted at each node v_j, for $1 \leq j \leq \log N + 1$, in the path from the root of $\mathcal{T}$ down to u_i, the length of the user secret key is $\mathcal{O}(\log N)$.

The ciphertext length in the CS method is $\mathcal{O}\left(r \log\left(\frac{N}{r}\right)\right)$ due to the fact that a logarithmic number of subtrees is required to exclude each of the r revoked users (see [6] for further details).

Extension of the CS Method to the Public Key Setting. As mentioned earlier, the original CS method applies in the symmetric key setting. Thus, only

the Center can broadcast since only it knows all the long-lived keys associated with each subtree in $\mathcal{S}$. In [8], Dodis and Fazio extended the original CS method to the public key setting by using a two step process.

The first step is a unique assignment of hierarchical identifiers (HID) to the nodes in the full binary tree $\mathcal{T}$ as follows. First, assign the root of $\mathcal{T}$ a special ID, which we refer to as `Root`. Then, assign each edge of $\mathcal{T}$ with ID 0 or 1 depending on whether the edge connects its parent node to the left or right child. Now, HID_j of any node $v_j \in \mathcal{T}$ can be computed by concatenating all the edge IDs starting from the root of $\mathcal{T}$ down to v_j and then pre-pending the root ID at the front. Since any prefix of HID_j of v_j represents the valid HID of a parent node of v_j, for the simplicity of notation, we denote $\mathsf{HID}_{i|j}$ the prefix of the hierarchical identifier HID_i of length j.

The second step is to use Identity-Based Encryption (IBE), further explained in Sect. 2.2, to encrypt the short-lived session key during broadcast, essentially porting the original CS method to the public key setting. This allows any user to broadcast a message since the tree structure of the users $\mathcal{T}$ and the HIDs of the roots of the subtrees of $\mathcal{T}$ are publicly known. In this setting, the Center acts as the trusted authority to provide each user with the $(\log N + 1)$ IBE secret keys of the HIDs of the roots of the subtrees of $\mathcal{T}$ that the user belongs to.

2.2 Anonymous Identity-Based Encryption (AIBE)

Identity-Based Encryption (IBE), originally proposed by Shamir in [22], is a public key encryption scheme in which the user public key is an arbitrary bit-string and the user secret key is generated by a trusted authority known as the *Private Key Generator* (PKG) using its master key. The first implementation of this scheme was given in [23] (further implementations can be found in [24–26] to name a few).

An IBE scheme is called anonymous, formally called Anonymous Identity-Based Encryption (AIBE), if an adversary cannot distinguish the public key under which a ciphertext is generated. This notion of anonymity was first introduced in [20]. Subsequent implementations can be found in [27] and [21]. Given below is the formal definition of an AIBE scheme. We refer the reader to [20] for further details including the formal definition of security.

Definition 1. *An anonymous identity-based encryption* (AIBE) *scheme, associated with a message space $\mathcal{MSP}$, and a ciphertext space $\mathcal{CSP}$, is a tuple of probabilistic polynomial algorithms* (Init, Ext, Enc, Dec) *such that:*

$(\mathrm{PK}, \mathrm{MSK}) \leftarrow \mathsf{Init}(1^\lambda)$**:** *The initialization algorithm* Init *takes as input the security parameter 1^λ, and outputs the public key PK and the master secret key MSK of the system.*

$sk_i \leftarrow \mathsf{Ext}(\mathrm{PK}, \mathrm{MSK}, \mathsf{ID})$**:** *The key extraction algorithm* Ext *takes as input the public parameters PK, the master secret key MSK, and an identifier $\mathsf{ID}_i \in \{0,1\}^*$. It outputs the secret key sk_i capable of decrypting ciphertexts intended for the holder of the given identifier ID_i.*

$c \leftarrow \mathsf{Enc}(\mathrm{PK}, \mathsf{ID}, m)$**:** *The encryption algorithm* Enc *algorithm takes as input the public parameters PK, an identifier* $\mathsf{ID}_i \in \{0,1\}^*$*, and a message* $m \in \mathcal{MSP}$*. It then outputs a ciphertext* $c \in \mathcal{CSP}$*.*

$m/\perp := \mathsf{Dec}(\mathrm{PK}, sk_i, c)$**:** *Given a secret key* sk_i *and a ciphertext* $c \in \mathcal{CSP}$*, the decryption algorithm* Dec *either outputs a message* $m \in \mathcal{MSP}$ *or the failure symbol* $\perp$*. We assume that* Dec *is deterministic.*

Correctness. For every $\mathsf{ID}_i \in \{0,1\}^*$ and every $m \in \mathcal{MSP}$, if sk_i is the secret key output by $\mathsf{Ext}(\mathrm{PK}, \mathrm{MSK}, \mathsf{ID})$, then $\mathsf{Dec}(\mathrm{PK}, sk_i, \mathsf{Enc}(\mathrm{PK}, \mathsf{ID}, m)) = m$.

Weakly Robust AIBE. The *Robust Encryption*, formalized by Abdalla et al. [28], requires that it is hard to produce a ciphertext that is valid for two different users. In [28], the authors define two types of robustness, strong and weak. Informally, an AIBE scheme is called *weakly robust*, if any adversary has negligible advantage in producing two identities $\mathsf{ID}_0, \mathsf{ID}_1$ and a message m such that the encryption of m under ID_0 can be decrypted with the private key associated with ID_1 leading to a non-$\perp$ result. In [28], the authors also provide a transformation algorithm which makes possible to obtain a weakly robust AIBE scheme from a regular AIBE one.

3 Outsider-Anonymous Broadcast Encryption (oABE)

3.1 The Setting

Definition 2. *An outsider-anonymous broadcast encryption* (oABE) *scheme, associated with a universe of users* $U = \{1, \ldots, N\}$*, a message space* $\mathcal{MSP}$*, and a ciphertext space* $\mathcal{CSP}$*, is a tuple of probabilistic polynomial algorithms* (Setup, KeyGen, Encrypt, Decrypt) *such that:*

$(\mathrm{PK}, \mathrm{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, N)$**:** *The* Setup *algorithm takes as input the security parameter* 1^λ *and the number of users in the system* N*. It outputs the public key PK and the master secret key MSK of the system.*

$sk_i \leftarrow \mathsf{KeyGen}(\mathrm{PK}, \mathrm{MSK}, i)$**:** *The key generation algorithm* KeyGen *takes as input the public parameters PK, the master secret key MSK, and a user* $i \in U$*. It outputs the secret key* sk_i *of the user* i*.*

$c \leftarrow \mathsf{Encrypt}(\mathrm{PK}, S, m)$**:** *The* Encrypt *algorithm takes as input the public parameters PK, the set of receivers* $S \subseteq U$*, and a message* $m \in \mathcal{MSP}$*. It then outputs a ciphertext* $c \in \mathcal{CSP}$*.*

$m/\perp := \mathsf{Decrypt}(\mathrm{PK}, sk_i, c)$**:** *Given a secret key* sk_i *and a ciphertext* $c \in \mathcal{CSP}$*, the* Decrypt *algorithm either outputs a message* $m \in \mathcal{MSP}$ *or the failure symbol* $\perp$*. We assume that* Decrypt *is deterministic.*

Correctness. For every $S \subseteq U$, every $i \in S$, and every $m \in \mathcal{MSP}$, if sk_i is the secret key output by $\mathsf{KeyGen}(\mathrm{PK}, \mathrm{MSK}, i)$ then $\mathsf{Decrypt}(\mathrm{PK}, sk_i, \mathsf{Encrypt}(\mathrm{PK}, S, m)) = m$.

Notice that the decryption algorithm in the above definition does not require the set of recipients S as an input. We stress that this is crucial for providing any level of anonymity in a broadcast encryption scheme.

3.2 The Security Model

Degrees of Anonymity. The degree of recipient-set anonymity captured in our security model, which we call *outsider-anonymity*, lies between the complete lack of protection that characterizes traditional broadcast encryption schemes as introduced in [2, 14], and the full anonymity provided in schemes such as [15, 16]. In an oABE scheme, when the adversary receives a ciphertext of which she is not a legal recipient, she will be unable to learn anything about the identities of the legal recipients (let alone the contents of the ciphertext). Still, for those ciphertexts for which the adversary is in the authorized set of recipients, she might also learn the identities of some the other legal recipients. This seems a natural relaxation, since often the *contents* of the communication already reveals something about the recipient set. At the same time, our new intermediate definition of security might allow the construction of more efficient anonymous broadcast encryption schemes; for example, in Sect. 4 we describe the first broadcast encryption scheme with sub-linear ciphertexts that attains some meaningful recipient-set anonymity guarantees.

CCA Security. We now present the security requirements for a broadcast encryption scheme to be *outsider-anonymous* against chosen-ciphertext attacks (CCA). First we define the CCA of an oABE scheme as a game, which we term oABE-IND-CCA, played between a probabilistic polynomial time (PPT) adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. The security requirement is that $\mathcal{A}$'s advantage of winning the oABE-IND-CCA game is negligible. The high-level idea of this game is for any two sets of recipients $S_0, S_1 \in U$, $\mathcal{A}$ cannot distinguish between a ciphertext intended for the recipient set S_0 and a ciphertext intended for the recipient set S_1 given the fact that the $\mathcal{A}$ does not possess the secret key of any user in $S_0 \cup S_1$. We require the two sets S_0, S_1 be the same size in order to avoid trivial attacks. The formal definitions follow.

Definition 3. *The* oABE-IND-CCA *game defined for an* oABE *scheme* $\Pi =$ (Setup,KeyGen,Encrypt, Decrypt)*, a* PPT *adversary* $\mathcal{A}$*, and a challenger* $\mathcal{C}$ *is as follows:*

Setup: $\mathcal{C}$ *runs* $(\mathrm{PK}, \mathrm{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, N)$ *and gives* $\mathcal{A}$ *the resulting public key PK, keeping the master secret key MSK to itself.* $\mathcal{C}$ *also initializes the set of revoked users* Rev *to be empty.*

Phase 1: $\mathcal{A}$ *adaptively issues queries* $q_1, \ldots, q_m$ *where each* q_i *is one of the following:*

- *Secret key query* i*:* $\mathcal{A}$ *requests the secret key of the user* $i \in U$*.* $\mathcal{C}$ *runs* $sk_i \leftarrow \mathsf{KeyGen}(\mathrm{PK}, \mathrm{MSK}, i)$ *to generate the secret key* sk_i *of the user* i*, adds* i *to* Rev*, and sends* sk_i *to* $\mathcal{A}$*.*
- *Decryption query* (i, c)*:* $\mathcal{A}$ *issues a decryption query where* $i \in U$ *and* $c \in \mathcal{CSP}$*. First,* $\mathcal{C}$ *runs* $sk_i \leftarrow \mathsf{KeyGen}(\mathrm{PK}, \mathrm{MSK}, i)$ *to generate the secret key* sk_i *of the user* i*. Then, it runs* $\mathsf{Decrypt}(\mathrm{PK}, sk_i, c)$ *and gives the output to* $\mathcal{A}$*.*

Challenge: *$\mathcal{A}$ gives $\mathcal{C}$ two equal length messages $m_0, m_1 \in \mathcal{MSP}$ and two equal length sets of user identities $S_0, S_1 \subseteq U$ with the restriction that* $\mathsf{Rev} \cap (S_0 \cup S_1) = \emptyset$. *$\mathcal{C}$ picks a random bit $b \in \{0,1\}$, runs $c^* \leftarrow \mathsf{Encrypt}(\mathrm{PK}, S_b, m_b)$, and sends c^* to $\mathcal{A}$.*

Phase 2: *$\mathcal{A}$ adaptively issues additional queries $q_{m+1}, \ldots, q_n$ where each q_i is one of the following:*

- *Secret key query i such that $i \notin S_0 \cup S_1$.*
- *Decryption query (i, c) such that, if $i \in S_0 \cup S_1$, then $c \neq c^*$.*

In both cases, $\mathcal{C}$ responds as in Phase 1.

Guess: *The $\mathcal{A}$ output a guess $b' \in \{0,1\}$ and wins if $b' = b$.*

We refer to such an adversary $\mathcal{A}$ as an oABE-IND-CCA *adversary. The advantage of $\mathcal{A}$ winning the above game is defined as,*

$$\mathsf{Adv}^{\mathsf{oABE\text{-}IND\text{-}CCA}}_{\mathcal{A},\Pi} = \left|\Pr\left[b' = b\right] - \tfrac{1}{2}\right|$$

The probability is over the random bits used by the adversary $\mathcal{A}$ and the challenger $\mathcal{C}$.

Definition 4. *An* oABE *scheme* $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *is* $(t, q_{sk}, q_d, \epsilon)$*-secure if for any t-time* oABE-IND-CCA *adversary $\mathcal{A}$ making at most q_{sk} chosen secret key queries and at most q_d chosen decryption queries, we have that* $\mathsf{Adv}^{\mathsf{oABE\text{-}IND\text{-}CCA}}_{\mathcal{A},\Pi} \leq \epsilon$. *As a shorthand, we say that Π is* $(t, q_{sk}, q_d, \epsilon)$-oABE-IND-CCA *secure.*

CPA Security. The chosen plaintext attack (CPA) of an oABE scheme is defined similar to the oABE-IND-CCA game with the restriction that the adversary is not allowed to issue any decryption queries during *Phase 1* and *Phase 2*. The adversary is still allowed to issue secret key queries. The CPA security game is termed oABE-IND-CPA.

Definition 5. *An* oABE *scheme* $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *is* (t, q_{sk}, ϵ)-oABE-IND-CPA *secure if Π is* $(t, q_{sk}, 0, \epsilon)$-oABE-IND-CCA *secure.*

Remark 1. Our definition of security of an outsider-anonymous broadcast encryption scheme can be easily transformed to a definition of security of a fully anonymous broadcast encryption scheme by changing the restriction in the challenge phase, which is currently $\mathsf{Rev} \cap (S_0 \cup S_1) = \emptyset$, to $\mathsf{Rev} \cap (S_0 \triangle S_1) = \emptyset$.[1]

4 Our Constructions

We now present a CPA secure construction and two CCA secure constructions of outsider-anonymous broadcast encryption (oABE) schemes. In a nutshell, the key point of our constructions is to combine an anonymized version of the public-key extension by Dodis and Fazio [8] of the CS method by Naor et al. [6] with a

[1] For any two sets S_0, S_1, their symmetric difference is denoted by $S_0 \triangle S_1$.

fully secure weakly robust AIBE scheme such as [21]. Notice that our approach can be seen as a *framework* for achieving an oABE scheme by using any weakly robust AIBE scheme as an underlying primitive.

The ciphertext length in all constructions is $\mathcal{O}\left(r \log\left(\frac{N}{r}\right)\right)$ times the ciphertext length of the underlying AIBE scheme, and the user secret key length is $\mathcal{O}(\log N)$ times the user secret key length of the underlying AIBE scheme, where r is the number of revoked users and N is the total number of users in the system.

We provide two generic public-key constructions: a CPA secure construction in Sect. 4.1 and a CCA secure construction in Sect. 4.2. The limitation with both of these constructions is that on average, the Decrypt algorithm attempts $\mathcal{O}\left(r \log\left(\frac{N}{r}\right) \log N\right)$ decryption operations of the underlying AIBE scheme. In Sect. 4.3, we present an optimized CCA secure construction in which for a given oABE ciphertext, the Decrypt algorithm executes a single AIBE decryption operation.

4.1 A Generic CPA Public-Key Construction

Given a weakly robust AIBE scheme $\Pi' = (\mathsf{Init}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$, we construct an oABE scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ as follows. Let $\mathcal{T}$ denote the binary tree of N users in the system with respect to the CS method. For simplicity, we assume below that $N = 2^n$.

$\mathsf{Setup}(1^\lambda, N)$**:** Obtain $(\mathrm{PK}', \mathrm{MSK}') \leftarrow \mathsf{Init}(1^\lambda)$. Output the PK and MSK as follows,

$$\mathrm{PK} = (\mathrm{PK}', N) \qquad \mathrm{MSK} = \mathrm{MSK}'$$

$\mathsf{KeyGen}(\mathrm{PK}, \mathrm{MSK}, i)$**:** Let $\mathsf{HID}_i = (\mathtt{Root}, \mathsf{ID}_1, \ldots, \mathsf{ID}_n)$ be the hierarchical identifier associated with the user i in the binary tree $\mathcal{T}$. For $j = 1$ to $n+1$, compute $sk_{i,j} \leftarrow \mathsf{Ext}(\mathrm{PK}', \mathrm{MSK}', \mathsf{HID}_{i|j})$. Output the secret key sk_i of the user i as follows,

$$sk_i = (sk_{i,1}, \ldots, sk_{i,n+1})$$

$\mathsf{Encrypt}(\mathrm{PK}, S, m)$**:** Let Cover be the family of subtrees covering the set of receivers S according to the CS method. For each subtree T_j in Cover, let HID_j be the hierarchical identifier associated with the root of T_j. Let $l = |\mathsf{Cover}|$, $r = N - |S|$ and $L = \left\lfloor r \log\left(\frac{N}{r}\right) \right\rfloor$.

For $1 \le j \le l$, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathsf{HID}_j, m)$. Choose $\widetilde{m} \xleftarrow{\$} \mathcal{MSP}$.
For $l+1 \le j \le L$, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathtt{dummy}, \widetilde{m})$, where dummy is a special identifier used to obtain padding ciphertext components. Output the ciphertext c as follows,

$$c = \left(c_{\pi(1)}, \ldots, c_{\pi(L)}\right)$$

where $\pi : \{1, \ldots, L\} \to \{1, \ldots, L\}$ is a random permutation.[2]

[2] For the simplicity of exposition, our construction encrypts the actual message m. The ciphertext length could be further reduced by using a hybrid encryption where m is encrypted using a symmetric key encryption algorithm with a symmetric key k, and k is then encrypted using the oABE scheme.

$\mathsf{Decrypt}(\mathrm{PK}, sk_i, c)$**:** Parse the secret key sk_i as the tuple $(sk_{i,1}, \ldots, sk_{i,n+1})$ and the ciphertext c as the tuple $(c_1, \ldots, c_L)$. For $k = 1$ to $n+1$,

1. For $j = 1$ to L,
   (a) Compute $m \leftarrow \mathsf{Dec}(\mathrm{PK}', sk_{i,k}, c_j)$.
   (b) If $m \neq \perp$, return m. Otherwise, continue to next j.
2. If $k = n+1$, return $\perp$. Otherwise, continue to next k.

The correctness of this CPA secure generic public-key construction follows from the correctness of the underlying AIBE scheme. In Theorem 1 (whose proof is provided in Appendix A.1), we establish the security of the above generic public-key construction based on the security of the underlying AIBE scheme.

Theorem 1. *If* $\Pi' = (\mathsf{Init}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ *is* (t, q_{sk}, ϵ)-AIBE-IND-CPA *secure, then the above construction is* $\left(t, q_{sk}, 2\,\epsilon\, r \log\left(\frac{N}{r}\right)\right)$-oABE-IND-CPA *secure.*

Parameters. When the above construction is instantiated with Gentry's Fully Secure IBE scheme in the CPA setting [21], we obtain the following parameter lengths. MSK is just one element in $\mathbb{Z}_p$ and the integer N. PK is only 3 group elements in $\mathbb{G}$. The user secret key consists of $(\log N + 1)$ elements in $\mathbb{Z}_p$ and $(\log N + 1)$ elements in $\mathbb{G}$. The ciphertext consists of $\left\lfloor r \log\left(\frac{N}{r}\right)\right\rfloor$ elements in $\mathbb{G}$ and $2\left\lfloor r \log\left(\frac{N}{r}\right)\right\rfloor$ elements in $\mathbb{G}_{\mathrm{T}}$. Also notice that the Enc algorithm in Gentry's AIBE scheme does not require any pairing computations since they can be precomputed.

4.2 A Generic CCA Public-Key Construction

Given a weakly robust AIBE scheme $\Pi' = (\mathsf{Init}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ and a strongly existentially unforgeable one-time signature scheme $\Sigma = (\mathsf{Sig\text{-}Gen}, \mathsf{Sign}, \mathsf{Vrfy})$, we construct an oABE scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ as follows. Let $\mathcal{T}$ denote the binary tree of N users in the system with respect to the CS method. For simplicity, we assume below that $N = 2^n$.

$\mathsf{Setup}(1^\lambda, N)$**:** Obtain $(\mathrm{PK}', \mathrm{MSK}') \leftarrow \mathsf{Init}(1^\lambda)$. Output the PK and MSK as follows,

$$\mathrm{PK} = (\mathrm{PK}', N) \qquad \mathrm{MSK} = \mathrm{MSK}'$$

$\mathsf{KeyGen}(\mathrm{PK}, \mathrm{MSK}, i)$**:** Let $\mathsf{HID}_i = (\mathtt{Root}, \mathsf{ID}_1, \ldots, \mathsf{ID}_n)$ be the hierarchical identifier associated with the user i in the binary tree $\mathcal{T}$. For $j = 1$ to $n+1$, compute $sk_{i,j} \leftarrow \mathsf{Ext}(\mathrm{PK}', \mathrm{MSK}', \mathsf{HID}_{i|j})$. Output the secret key sk_i of the user i as follows,

$$sk_i = (sk_{i,1}, \ldots, sk_{i,n+1})$$

$\mathsf{Encrypt}(\mathrm{PK}, S, m)$**:** Generate $(\mathrm{VK}, \mathrm{SK}) \leftarrow \mathsf{Sig\text{-}Gen}(1^\lambda)$. Let Cover be the family of subtrees covering the set of receivers S according to the CS method. For each subtree T_j in Cover, let HID_j be the hierarchical identifier associated with the root of T_j.
Let $l = |\mathsf{Cover}|$, $r = N - |S|$ and $L = \left\lfloor r \log\left(\frac{N}{r}\right)\right\rfloor$.

For $1 \leq j \leq l$, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathsf{HID}_j, \mathrm{VK}||m)$. Let $\widetilde{m}$ be a random string of the same length as $\mathrm{VK}||m$. For $l+1 \leq j \leq L$, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathtt{dummy}, \widetilde{m})$, where $\mathtt{dummy}$ is a special identifier used to obtain padding ciphertext components. Compute the ciphertext c as follows,

$$c = \left(c_{\pi(1)}, \ldots, c_{\pi(L)}\right)$$

where $\pi : \{1, \ldots, L\} \rightarrow \{1, \ldots, L\}$ is a random permutation.
Generate $\sigma \leftarrow \mathsf{Sign}_{\mathrm{SK}}(\mathrm{VK}||c)$, and output $C = \sigma||c$.

$\mathsf{Decrypt}(\mathrm{PK}, sk_i, C)$**:** Parse the secret key sk_i as the tuple $(sk_{i,1}, \ldots, sk_{i,n+1})$ and the ciphertext C as the tuple $\sigma||\,(c_1, \ldots, c_L)$. For $k = 1$ to $n+1$

1. For $j = 1$ to L
   (a) Compute $m' \leftarrow \mathsf{Dec}(\mathrm{PK}', sk_{i,k}, c_j)$.
   (b) If $m' \neq \perp$, parse $m' = \mathrm{VK}||m$, and return m if $\mathsf{Vrfy}(\mathrm{VK}, \sigma, \mathrm{VK}||c)$. Otherwise, continue to next j.
2. If $k = n+1$, return $\perp$. Otherwise, continue to next k.

The correctness of this CCA secure generic public-key construction follows from the correctness of the underlying Σ and AIBE schemes. Next, in Theorem 2 (whose proof is provided in the full version of this paper [29]), we establish the security of the above generic public-key construction based on the security of the underlying Σ and AIBE schemes.

Theorem 2. *If* $\Sigma = (\mathsf{Sig\text{-}Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ *is* (t, ϵ_1)*-strongly existentially unforgeable and* $\Pi' = (\mathsf{Init}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ *is* $(t, q_{sk}, q_d, \epsilon_2)$-AIBE-IND-CCA *secure, then the above construction is* $(t,\ q_{sk},\ q_d,\ 2(\epsilon_1 + \epsilon_2)\, r \log\left(\frac{N}{r}\right))$-oABE-IND-CCA *secure.*

Parameters. The parameter lengths of the above construction when instantiated with Gentry's Fully Secure IBE scheme in the CCA setting [21] are as follows. MSK is one element in $\mathbb{Z}_p$ and the integer N. PK consists of 5 group elements in $\mathbb{G}$ and the definition of a hash function H from a family of universal one-way hash functions. The user secret key consists of $3(\log N + 1)$ elements in $\mathbb{Z}_p$ and $3(\log N + 1)$ elements in $\mathbb{G}$. The ciphertext consists of $\left\lfloor r \log\left(\frac{N}{r}\right)\right\rfloor$ elements in $\mathbb{G}$ and $3\left\lfloor r \log\left(\frac{N}{r}\right)\right\rfloor$ elements in $\mathbb{G}_{\mathrm{T}}$. Similar to Gentry's CPA secure AIBE construction, the Enc algorithm in the CCA secure construction does not require any pairing computations since they can be pre-computed.

4.3 An Enhanced CCA Public-Key Construction

The main limitation of our generic public-key constructions is the running time of the decryption algorithm. As described in the opening paragraphs of Sect. 4, decryption amounts to performing $\mathcal{O}\left(r \log\left(\frac{N}{r}\right) \log N\right)$ AIBE decryption attempts on average. The root cause behind this limitation is the decryption process's inability to identify the correct AIBE ciphertext component efficiently. In this section, we describe an enhancement of our generic public-key construction under the gap Diffie-Hellman assumption, in the random oracle model. The main

idea of this enhancement is to adapt the techniques of [15] to the structure of our ciphertexts and attach a unique tag to each AIBE ciphertext component of a given oABE ciphertext. With this optimization, the Decrypt algorithm is able to identify the correct AIBE ciphertext component via a linear search through the whole oABE ciphertext components, at which point a single AIBE decryption operation suffices to recover the original plaintext. This yields an asymptotic decryption time of $\mathcal{O}\left(r \log\left(\frac{N}{r}\right) \log N\right)$, but in fact this is in a sense an overestimate, since the cost of searching for the correct ciphertext component is much less than carrying out multiple decryption attempts.

Given a weakly robust AIBE scheme $\Pi' = (\mathsf{Init}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ and a strongly existentially unforgeable one-time signature scheme $\Sigma = (\mathsf{Sig\text{-}Gen}, \mathsf{Sign}, \mathsf{Vrfy})$, we construct an optimized oABE scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ as follows. Let $\mathcal{T}$ denote the binary tree of N users in the system with respect to the CS method. For simplicity, we assume below that $N = 2^n$. Let $\mathbb{G} = \langle g \rangle$ be a group with prime order $q > 2^\lambda$ in which CDH is hard and DDH is easy, where g is a group generator. Let $H' : \mathbb{G} \to \{0,1\}^\lambda$ be a cryptographic hash function that will be modeled as a random oracle in the security analysis.

$\mathsf{Setup}(1^\lambda, N)$**:** Obtain $(\mathrm{PK}', \mathrm{MSK}') \leftarrow \mathsf{Setup}'(1^\lambda)$. For each node (with the hierarchical identifier HID) in $\mathcal{T}$, draw $a_{\mathsf{HID}} \xleftarrow{\$} \mathbb{Z}_q^*$, and compute $y_{\mathsf{HID}} = g^{a_{\mathsf{HID}}}$. Output the PK and MSK as follows,

$$\mathrm{PK} = \left(\mathrm{PK}', N, \mathbb{G}, g, \{y_{\mathsf{HID}}\}_{\mathsf{HID} \in \mathcal{T}}\right) \qquad \mathrm{MSK} = \left(\mathrm{MSK}', \{a_{\mathsf{HID}}\}_{\mathsf{HID} \in \mathcal{T}}\right)$$

$\mathsf{KeyGen}(\mathrm{PK}, \mathrm{MSK}, i)$**:** Let $\mathsf{HID}_i = (\mathtt{Root}, \mathsf{ID}_1, \ldots, \mathsf{ID}_n)$ be the hierarchical identifier associated with the user i in the binary tree $\mathcal{T}$. For $j = 1$ to $n+1$, set $\overline{sk}_{i,j} = a_{\mathsf{HID}_{i|j}}$, and compute $sk_{i,j} \leftarrow \mathsf{Init}(\mathrm{PK}', \mathrm{MSK}', \mathsf{HID}_{i|j})$. Output the secret key sk_i of the user i as follows,

$$sk_i = \left(\left(\overline{sk}_{i,1}, sk_{i,1}\right), \ldots, \left(\overline{sk}_{i,n+1}, sk_{i,n+1}\right)\right)$$

$\mathsf{Encrypt}(\mathrm{PK}, S, m)$**:** Generate $(\mathrm{VK}, \mathrm{SK}) \leftarrow \mathsf{Sig\text{-}Gen}(1^\lambda)$. Let Cover be the family of subtrees covering the set of receivers S according to the CS method. For each subtree T_j in Cover, let HID_j be the hierarchical identifier associated with the root of T_j.

Let $l = |\mathsf{Cover}|$, $r = N - |S|$ and $L = \left\lfloor r \log\left(\frac{N}{r}\right) \right\rfloor$. Draw $s \xleftarrow{\$} \mathbb{Z}_q^*$, and compute $\overline{c}_0 = g^s$.

For $1 \le j \le l$, compute $\overline{c}_j = H'(y_{\mathsf{HID}_j}^s)$, $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathsf{HID}_j, \mathrm{VK} || y_{\mathsf{HID}_j}^s || m)$. Let $\widetilde{m}$ be a random string of the same length as $\mathrm{VK}||\overline{c}_0||m$. For $l+1 \le j \le L$, set $s_j \xleftarrow{\$} \mathbb{Z}_q^*$, and compute $\overline{c}_j = H'(g^{s_j})$, $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathtt{dummy}, \widetilde{m})$, where dummy is a special identifier used to obtain padding ciphertext components. Compute the ciphertext c as follows,

$$c = \left(\overline{c}_0, \left(\overline{c}_{\pi(1)}, c_{\pi(1)}\right), \ldots, \left(\overline{c}_{\pi(L)}, c_{\pi(L)}\right)\right)$$

where $\pi : \{1, \ldots, L\} \to \{1, \ldots, L\}$ is a random permutation. Generate $\sigma \leftarrow \mathsf{Sign}(\mathrm{SK}, \mathrm{VK}||c)$, and output $C = \sigma||c$.

$\mathsf{Decrypt}(\mathrm{PK}, sk_i, c)$**:** Parse the secret key sk_i as the tuple $((\overline{sk}_{i,1},\ sk_{i,1}), \ldots, (\overline{sk}_{i,n+1},\ sk_{i,n+1}))$ and the ciphertext C as the tuple $(\sigma || \ \overline{c}_0, (\overline{c}_1,\ c_1), \ldots, (\overline{c}_L,\ c_L))$.

1. For $k = 1$ to $n + 1$
   (a) Compute $y_k = H'(\overline{c}_0^{\ \overline{sk}_{i,k}})$
2. Check whether $\exists\ k \in [1, n + 1]$, $\exists\ j \in [1, L]$ such that $y_k = \overline{c}_j$
   (a) If suitable k, j exist, compute $m' \leftarrow \mathsf{Dec}(\mathrm{PK}', sk_{i,k}, c_j)$. Parse m' as $\mathrm{VK}||x||m$ and return m if $x = \overline{c}_0^{\ \overline{sk}_{i,k}}$ and $\mathsf{Vrfy}(\mathrm{VK}, \sigma, \mathrm{VK}||c)$.
   (b) Otherwise, return $\perp$.

Notice that the check in Step 2. can be performed in expected time $\mathcal{O}(n + L) = \mathcal{O}(L)$, *e.g.*, using a hash table $\mathrm{H_T}$ to compute the intersection between $\{y_k\}_{k\in[1,n+1]}$ and $\{\overline{c}_j\}_{j\in[1,L]}$ as follows:

a. Initialize $\mathrm{H_T}$ to be empty.
b. For $k = 1$ to $n + 1$
   – Insert (y_k, k) in $\mathrm{H_T}$.
c. For $j \in 1$ to L
   – Look up an entry of the form $(\overline{c}_j, k)$ in $\mathrm{H_T}$. If found, return k.

Theorem 3. *If* $\Sigma = (\mathsf{Sig\text{-}Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ *is* (t, ϵ_1)*-strongly existentially unforgeable,* $\Pi' = (\mathsf{Init}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ *is* $(t, q_{sk}, q_d, \epsilon_2)$-$\mathsf{AIBE\text{-}IND\text{-}CCA}$ *secure, and CDH is* (t, ϵ_3)*-hard in* $\mathbb{G}$ *and DDH is efficiently computable in* $\mathbb{G}$*, then the above construction is* $(t,\ q_{sk},\ q_d,\ 2(\epsilon_1 + \epsilon_2 + \epsilon_3)\ r \log\left(\frac{N}{r}\right))$-$\mathsf{oABE\text{-}IND\text{-}CCA}$ *secure, in the random oracle model.*

Proof. The proof follows the same structure of the proof for Theorem 2. We defer the details to the full version of this paper [29].

Remark 2. Using the twin Diffie-Hellman methodology [30] via techniques similar to [16], it is possible to modify the enhanced CCA construction of Sect. 4.3 to get an outsider-anonymous broadcast encryption scheme that is adaptively CCA secure, in the standard model, under the decisional Diffie-Hellman assumption. We defer the details to the full version of the paper [29].

4.4 An Enhanced CCA Symmetric-Key Construction

The enhanced CCA public key construction achieves a major performance gain in the Decrypt algorithm compared to the generic CCA construction, but it also changes the length of the public key from $\mathcal{O}(1)$ to $\mathcal{O}(N)$. This increase in public key length may not be a concern for many practical constructions, since the public key can be stored as a static data file on a server on the Internet and also in users' computers. Still, for the symmetric-key setting it is possible to accommodate storage-sensitive systems and attain constant key storage at the Center, while maintaining efficient decryption and logarithmic storage at the receivers.

In particular, recall from Sect. 2.1 that in the symmetric-key setting, only the Center can broadcast messages to the receivers. Thus, the $\mathcal{O}(N)$ information from which the tags for efficient decryption are created does not need to be

published. Therefore, this information can be compressed into $\mathcal{O}(1)$ key storage using a standard trick based on any length-tripling pseudo-random number generator G (*cf. e.g.*, the SD method of Naor *et al.* [6]). In other words, the random exponents associated with the subtrees of $\mathcal{T}$ (*cf.* Sect. 4.3) are now pseudorandomly generated from a single seed, by repeated invocations of G on the left or right third of the result of the previous iteration, based on the path to the root of the subtree at hand. Finally, upon reaching the subtree root, the middle third of the pseudorandom output is used to generate the required exponent.

5 Conclusions and Future Work

In this work, we introduced the notion of outsider-anonymity in the broadcast encryption setting and showed that it enables efficient constructions of broadcast encryption schemes with sublinear communication complexity and meaningful anonymity guarantees. It remains an interesting open problem to construct receiver-anonymous broadcast encryption schemes that at once afford full anonymity to the receivers and attain performance levels comparable to those of standard broadcast encryption systems.

Acknowledgment. Nelly Fazio's research was sponsored in part by NSF award #1117675 and by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. This project was also partially supported by PSC-CUNY Awards 63356-00 41 and 64578-00 42, jointly funded by The Professional Staff Congress and The City University of New York.

References

1. Berkovits, S.: How to Broadcast a Secret. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 535–541. Springer, Heidelberg (1991)
2. Fiat, A., Naor, M.: Broadcast Encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994)
3. AACS: Advanced access content system, http://www.aacsla.com/
4. Goh, E.J., Shacham, H., Modadugu, N., Boneh, D.: Sirius: Securing remote untrusted storage. In: ISOC Network and Distributed Systems Security Symposium—NDSS 2003, pp. 131–145 (2003)
5. Garay, J.A., Staddon, J., Wool, A.: Long-Lived Broadcast Encryption. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 333–352. Springer, Heidelberg (2000)
6. Naor, D., Naor, M., Lotspiech, J.: Revocation and Tracing Schemes for Stateless Receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001)

7. Halevy, D., Shamir, A.: The LSD Broadcast Encryption Scheme. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 47–60. Springer, Heidelberg (2002)
8. Dodis, Y., Fazio, N.: Public Key Broadcast Encryption for Stateless Receivers. In: Feigenbaum, J. (ed.) DRM 2002. LNCS, vol. 2696, pp. 61–80. Springer, Heidelberg (2002)
9. Dodis, Y., Fazio, N.: Public Key Trace and Revoke Scheme Secure against Adaptive Chosen Ciphertext Attack. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 100–115. Springer, Heidelberg (2003)
10. Dodis, Y., Fazio, N., Kiayias, A., Yung, M.: Scalable public-key tracing and revoking. In: Principles of Distributed Computing—PODC 2003, pp. 190–199 (2003); Invited to the PODC 2003 Special Issue of Journal of Distributed Computing
11. Yao, D., Fazio, N., Dodis, Y., Lysyanskaya, A.: ID-based encryption for complex hierarchies with applications to forward security and broadcast encryption. In: ACM Computer and Communications Security—CCS 2004, pp. 354–363 (2004)
12. Boneh, D., Gentry, C., Waters, B.: Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005)
13. Boneh, D., Waters, B.: A fully collusion resistant broadcast, trace, and revoke system. In: ACM Computer and Communications Security—CCS 2006, pp. 211–220 (2006)
14. Gentry, C., Waters, B.: Adaptive Security in Broadcast Encryption Systems (with Short Ciphertexts). In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 171–188. Springer, Heidelberg (2009)
15. Barth, A., Boneh, D., Waters, B.: Privacy in Encrypted Content Distribution Using Private Broadcast Encryption. In: Di Crescenzo, G., Rubin, A. (eds.) FC 2006. LNCS, vol. 4107, pp. 52–64. Springer, Heidelberg (2006)
16. Libert, B., Paterson, K.G., Quaglia, E.A.: Anonymous broadcast encryption. Cryptology ePrint Archive, Report 2011/476 (2011)
17. Krzywiecki, Ł., Kubiak, P., Kutyłowski, M.: A Revocation Scheme Preserving Privacy. In: Lipmaa, H., Yung, M., Lin, D. (eds.) Inscrypt 2006. LNCS, vol. 4318, pp. 130–143. Springer, Heidelberg (2006)
18. Yu, S., Ren, K., Lou, W.: Attribute-based on-demand multicast group setup with receiver anonymity. In: Security and Privacy in Communication Networks—SecureComm 2008, pp. 18:1–18:6 (2008)
19. Jarecki, S., Liu, X.: Unlinkable Secret Handshakes and Key-Private Group Key Management Schemes. In: Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 270–287. Springer, Heidelberg (2007)
20. Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., Shi, H.: Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 205–222. Springer, Heidelberg (2005)
21. Gentry, C.: Practical Identity-Based Encryption Without Random Oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
22. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
23. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
24. Boneh, D., Boyen, X.: Secure Identity Based Encryption Without Random Oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443–459. Springer, Heidelberg (2004)

25. Boneh, D., Gentry, C., Hamburg, M.: Space-efficient identity based encryption without pairings. In: IEEE Symposium on Foundations of Computer Science—FOCS 2007, pp. 647–657 (2007)
26. Waters, B.: Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)
27. Boyen, X., Waters, B.: Anonymous Hierarchical Identity-Based Encryption (Without Random Oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006)
28. Abdalla, M., Bellare, M., Neven, G.: Robust Encryption. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 480–497. Springer, Heidelberg (2010)
29. Fazio, N., Perera, I.M.: Outsider-Anonymous Broadcast Encryption with Sublinear Ciphertexts. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 225–242. Springer, Heidelberg (2012)
30. Cash, D., Kiltz, E., Shoup, V.: The Twin Diffie-Hellman Problem and Applications. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 127–145. Springer, Heidelberg (2008)

A Security Proofs

For ease of reference, we report below some notation that will be used in the proofs presented in this section.

NOTATION. $U = \{1, \ldots, N\}$ is the universe of users. $\mathcal{T}$ denotes the binary tree of N users in the system with respect to the CS method. Let r be the number of revoked users and $L = \left\lfloor r \log\left(\frac{N}{r}\right)\right\rfloor$. For $b \in \{0,1\}$, let S_b be the set of authorized receivers chosen by the adversary in the challenge phase ($|S_0| = |S_1|$). Cover_b is the family of subtrees covering the set S_b according to the CS method. Let $l_b = |\mathsf{Cover}_b|$. For each subtree T_j^b in Cover_b, let HID_j^b be the hierarchical identifier associated with the root of T_j^b where $1 \leq j \leq l_b$.

A.1 Proof of Theorem 1

Theorem 1. If $\Pi' = (\mathsf{Init}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ is (t, q_{sk}, ϵ)-AIBE-IND-CPA secure, then the above construction is $\left(t, q_{sk}, 2\,\epsilon\, r \log\left(\frac{N}{r}\right)\right)$-oABE-IND-CPA secure.

Proof. We organize our proof as a sequence of games, $\mathrm{Game}_0^0, \ldots, \mathrm{Game}_{l_0}^0 \equiv \mathrm{Game}_{l_1}^1, \ldots, \mathrm{Game}_0^1$, between the adversary $\mathcal{A}$ and the challenger $\mathcal{C}$. In the first game (Game_0^0), $\mathcal{A}$ receives an encryption of m_0 for S_0 and in the last game (Game_0^1), $\mathcal{A}$ receives an encryption of m_1 for S_1.

Game_0^0: corresponds to the game given in Definition 5 when the challenge bit b is fixed to 0. The interaction between $\mathcal{A}$ and $\mathcal{C}$ during *Setup*, *Phase 1*, and *Phase 2* follow exactly as specified in the construction Π given in Sect. 4.1. During *Challenge*, $\mathcal{A}$ gives $\mathcal{C}$ two equal length messages $m_0, m_1 \in \mathcal{MSP}$ and two equal length sets of user identities $S_0, S_1 \subseteq U$ with the restriction that $\mathsf{Rev} \cap (S_0 \cup S_1) = \emptyset$, where Rev is the set of users that $\mathcal{A}$ corrupted during *Phase 1*. $\mathcal{C}$ computes the challenge ciphertext c^*, which will subsequently be sent to $\mathcal{A}$, as follows,

1. For $j = 1$ to l_0, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathsf{HID}_j^0, m_0)$.
2. Choose $\widetilde{m} \xleftarrow{\$} \mathcal{MSP}$.
3. For $j = l_0 + 1$ to L, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathtt{dummy}, \widetilde{m})$.
4. Set $c^* = \left(c_{\pi(1)}, \ldots, c_{\pi(L)}\right)$, where $\pi : \{1, \ldots, L\} \to \{1, \ldots, L\}$ is a random permutation.

Eventually, $\mathcal{A}$ outputs a bit b' and wins if $b' = 0$.

$\mathbf{Game}_h^0$ $(1 \le h \le l_0)$: is similar to Game_{h-1}^0, but $\mathcal{C}$ computes the challenge ciphertext c^* as follows,

1. For $j = 1$ to $l_0 - h$, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathsf{HID}_j^0, m_0)$.
2. Choose $\widetilde{m} \xleftarrow{\$} \mathcal{MSP}$.
3. For $j = l_0 - h + 1$ to L, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathtt{dummy}, \widetilde{m})$.
4. Set $c^* = \left(c_{\pi(1)}, \ldots, c_{\pi(L)}\right)$, where $\pi : \{1, \ldots, L\} \to \{1, \ldots, L\}$ is a random permutation.

At the end, $\mathcal{A}$ outputs a bit b' and wins if $b' = 0$.

$\mathbf{Game}_{l_1}^1$: is identical to $\mathrm{Game}_{l_0}^0$

$\mathbf{Game}_k^1$ $(0 \le k < l_1)$: is similar to Game_{k+1}^1, but the challenge ciphertext c^* is now computed by $\mathcal{C}$ as,

1. For $j = 1$ to $l_1 - k$, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathsf{HID}_j^1, m_1)$.
2. Choose $\widetilde{m} \xleftarrow{\$} \mathcal{MSP}$.
3. For $j = l_1 - k + 1$ to L, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathtt{dummy}, \widetilde{m})$.
4. Set $c^* = \left(c_{\pi(1)}, \ldots, c_{\pi(L)}\right)$, where $\pi : \{1, \ldots, L\} \to \{1, \ldots, L\}$ is a random permutation.

Finally, $\mathcal{A}$ outputs a bit b' and wins if $b' = 0$.

For $0 \le i \le l_0$ and $0 \le j \le l_1$, let $\mathsf{Adv}_{\mathcal{A},\Pi}^{0,i}$ and $\mathsf{Adv}_{\mathcal{A},\Pi}^{1,j}$ denote $\mathcal{A}$'s advantage of winning Game_i^0 and Game_j^1 respectively. In Lemma 1, we show that if the underlying AIBE scheme is (t, q_{sk}, ϵ)-AIBE-IND-CPA secure, then $\mathcal{A}$'s advantage of distinguishing Game_{h-1}^0 from Game_h^0 is at most ϵ. Similarly, Lemma 2 states that under similar conditions $\mathcal{A}$'s advantage of distinguishing Game_{k+1}^1 from Game_k^1 is at most ϵ. Therefore, we have,

$$\begin{aligned}\left|\mathsf{Adv}_{\mathcal{A},\Pi}^{0,0} - \mathsf{Adv}_{\mathcal{A},\Pi}^{1,0}\right| &\le \epsilon\,(l_0 + l_1)\\ &\le 2\,\epsilon\,L\\ &\le 2\,\epsilon\,r \log\left(\frac{N}{r}\right).\end{aligned}$$

Lemma 1. *For $1 \le h \le l_0$, if the underlying* AIBE *scheme Π' is (t, q_{sk}, ϵ)-*AIBE-IND-CPA *secure, then $\mathcal{A}$'s adv. of distinguishing Game_{h-1}^0 from Game_h^0 is at most ϵ:*

$$\left|\mathsf{Adv}_{\mathcal{A},\Pi}^{0,h-1} - \mathsf{Adv}_{\mathcal{A},\Pi}^{0,h}\right| \le \epsilon.$$

Proof. We build a PPT adversary $\mathcal{B}$ that runs the AIBE-IND-CPA game with its challenger $\mathcal{C}'$ as follows. First, $\mathcal{B}$ receives the public key PK' of the AIBE scheme from $\mathcal{C}'$. Next, $\mathcal{B}$ internally executes the oABE-IND-CPA game with $\mathcal{A}$ in order to gain advantage in the AIBE-IND-CPA game. The specifics of the interaction between $\mathcal{C}'$, $\mathcal{B}$, and $\mathcal{A}$ are given below.

Setup: $\mathcal{B}$ forwards PK' to $\mathcal{A}$. $\mathcal{B}$ also initializes the set of revoked users Rev to be empty.

Phase 1: When $\mathcal{A}$ invokes a secret key query for user i, first, $\mathcal{B}$ computes HID_i, which is the hierarchical identifier associated with the user i in the binary tree $\mathcal{T}$. Next, for $j = 1$ to $n+1$, $\mathcal{B}$ obtains the secret key $sk_{i,j}$ of the identity $\mathsf{HID}_{i|j}$ from its challenger $\mathcal{C}'$. After adding i to Rev, $\mathcal{B}$ sends to $\mathcal{A}$ the secret key of the user i as $sk_i = (sk_{i,1}, \ldots, sk_{i,n+1})$.

Challenge: $\mathcal{B}$ receives from $\mathcal{A}$ two equal length messages $m_0, m_1 \in \mathcal{MSP}$ and two equal length sets of user identities $S_0, S_1 \subseteq U$ with the restriction that $\mathsf{Rev} \cap (S_0 \cup S_1) = \emptyset$. $\mathcal{B}$ draws $\widetilde{m} \xleftarrow{\$} \mathcal{MSP}$ and computes the components of its challenge query as follows,

$$id'_0 = \mathsf{HID}^0_{l_0-h+1}, \quad id'_1 = \mathtt{dummy} \qquad m'_0 = m_0, \quad m'_1 = \widetilde{m}$$

Observe that the condition $\mathsf{Rev} \cap (S_0 \cup S_1) = \emptyset$, together with the key assignment strategy of the CS method guarantees that the identity id'_0 hadn't been queried to $\mathcal{B}$'s extraction oracle, and thus this is a valid challenge query to $\mathcal{C}'$.

$\mathcal{B}$ sends the two identities id'_0, id'_1 and the two messages m'_0, m'_1 as the challenge query to $\mathcal{C}'$. $\mathcal{C}'$ picks a random bit $b \in \{0,1\}$ and sends $c^{*\prime} \leftarrow \mathsf{Enc}(\mathrm{PK}', id'_b, m'_b)$ to $\mathcal{B}$.

Finally, $\mathcal{B}$ computes the challenge ciphertext c^*, which is eventually sent to $\mathcal{A}$, as follows,

1. For $j = 1$ to $l_0 - h$, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathsf{HID}^0_j, m_0)$.
2. Set $c_{l_0-h+1} = c^{*\prime}$.
3. For $j = l_0 - h + 2$ to L, compute $c_j \leftarrow \mathsf{Enc}(\mathrm{PK}', \mathtt{dummy}, \widetilde{m})$.
4. Set $c^* = \big(c_{\pi(1)}, \ldots, c_{\pi(L)}\big)$, where $\pi : \{1, \ldots, L\} \to \{1, \ldots, L\}$ is a random permutation.

Phase 2: This phase is handled similarly to *Phase 1* with the usual restriction that $\mathcal{A}$ does *not* invoke a secret key query i such that $i \in S_0 \cup S_1$.

Guess: $\mathcal{A}$ outputs a guess b' and $\mathcal{B}$ passes this bit as its guess for b to $\mathcal{C}'$.

Observe that, by construction, it holds that if $\mathcal{C}'$ chooses $b = 0$, then $\mathcal{B}$ is playing Game^0_{h-1}, whereas if $b = 1$, then $\mathcal{B}$ is playing Game^0_h. Therefore, $\mathcal{B}$'s $\mathsf{AIBE\text{-}IND\text{-}CPA}$ advantage is equivalent to $\mathcal{A}$'s advantage in distinguishing Game^0_{h-1} from Game^0_h. More formally,

$$\left|\mathsf{Adv}^{0,h-1}_{\mathcal{A},\Pi} - \mathsf{Adv}^{0,h}_{\mathcal{A},\Pi}\right| = \mathsf{Adv}^{\mathsf{AIBE\text{-}IND\text{-}CPA}}_{\mathcal{B},\Pi'} \leq \epsilon.$$

Lemma 2. *For $0 \leq k < l_1$, if the underlying* AIBE *scheme Π' is (t, q_{sk}, ϵ)-*$\mathsf{AIBE\text{-}IND\text{-}CPA}$ *secure, then $\mathcal{A}$'s adv. of distinguishing Game^1_{k+1} from Game^1_k is at most ϵ. More precisely,*

$$\left|\mathsf{Adv}^{1,k+1}_{\mathcal{A},\Pi} - \mathsf{Adv}^{1,k}_{\mathcal{A},\Pi}\right| \leq \epsilon.$$

Proof. The argument is analogous to the proof of Lemma 1, and is therefore omitted.

Verifiable Predicate Encryption and Applications to CCA Security and Anonymous Predicate Authentication

Shota Yamada[1,⋆], Nuttapong Attrapadung[2], Bagus Santoso[3], Jacob C.N. Schuldt[2,⋆⋆], Goichiro Hanaoka[2], and Noboru Kunihiro[1]

[1] The University of Tokyo
{yamada@it.,kunihiro@}k.u-tokyo.ac.jp
[2] National Institute of Advanced Industrial Science and Technology (AIST)
{n.attrapadung,jacob.schuldt,hanaoka-goichiro}@aist.go.jp
[3] Institute for Infocomm Research
santosob@i2r.a-star.edu.sg

Abstract. In this paper, we focus on *verifiability* of predicate encryption. A verifiable predicate encryption scheme guarantees that all legitimate receivers of a ciphertext will obtain the same message upon decryption. While verifiability of predicate encryption might be a desirable property by itself, we furthermore show that this property enables interesting applications.

Specifically, we provide two applications of verifiable predicate encryption. Firstly, we show that for a large class of verifiable predicate encryption schemes, it is always possible to convert a chosen-plaintext secure scheme into a chosen-ciphertext secure one. Secondly, we show that a verifiable predicate encryption scheme allows the construction of a *deniable predicate authentication scheme*. This primitive enables a user to authenticate a message to a verifier using a private key satisfying a specified relation while at the same time allowing the user to deny ever having interacted with the verifier. This scheme furthermore guarantees the anonymity of the user in the sense that the verifier will learn nothing about the user's private key except that it satisfies the specified relation.

Lastly, we show that many currently known predicate encryption schemes already provide verifiability, and furthermore demonstrate that many predicate encryption schemes which do not provide verifiability, can be easily converted into schemes providing verifiability.

Our results not only highlight that verifiability is a very useful property of predicate encryption, but also show that efficient and practical schemes with this property can be obtained relatively easily.

1 Introduction

In many practical data transmission systems, we often encounter situations in which a sender would like to securely transmit data to a set of users satisfying

⋆ The first author is supported by JSPS Research Fellowships for Young Scientists.
⋆⋆ The fourth author is supported by JSPS Research Fellowships for Young Scientists.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 243–261, 2012.

certain criteria. To address this, several frameworks and concrete instantiations providing *encryption with multiple receivers* have been proposed in the literature. Examples of such frameworks include broadcast encryption (BE) [8,35], spatial encryption [9,2], and ciphertext-policy/key-policy attribute-based encryption (CP/KP-ABE) [17,7,36,22,21,28]. All of the above mentioned schemes can be seen as special cases of *predicate encryption* (PE) which is a new emerging paradigm for public key encryption that allows a fine-grained access control mechanism to be specified for encrypted data. More specifically, in an PE scheme for relation R, a ciphertext will be associated with a ciphertext attribute Y while a private key corresponds to key attribute X, and the decryption can be done only if the relation $R(X, Y)$ is satisfied. In this paper, we consider a wide class of relations which covers the above mentioned special cases.

Previous works on PE have mainly focused on security properties regarding privacy, namely *message privacy* (also referred to as payload hiding) and *ciphertext attribute hiding* (also referred to as anonymity). The former captures the property that a ciphertext with attribute X reveals no information about the encrypted messages if one does not possess a key with attribute Y such that $R(X, Y)$ is satisfied. The latter captures the property that for anyone in the possession of a private key with attribute X', a ciphertext reveals no information about the ciphertext attribute Y other than what is implied by $R(X', Y)$.

In this paper, we focus on *verifiability* of PE and the applications of PE schemes providing this property. If an PE scheme provides verifiability, it is guaranteed that all legitimate receivers of a ciphertext will obtain the same message upon decryption i.e. for a ciphertext with attribute Y, the decryption using two different private keys corresponding to attributes X and X' where both $R(X, Y)$ and $R(X', Y)$ are satisfied, will always yield the same message. Verifiability in itself is arguably a useful property and might even be required for some applications. For example, in pay-per-view systems, receivers might demand to be able to confirm that decryption results among all paying receivers are identical, especially in the case the decryption result is different from the expected. This property is guaranteed if an PE scheme with verifiability is used to broadcast data to the receivers.

However, besides guaranteeing consistency of the decryption results among all legitimate receivers, verifiability will furthermore enable interesting applications of PE schemes providing this property. In this paper, we show two specific applications of verifiable predicate encryption (VPE). More specifically, firstly we show that it is always possible to convert an arbitrary chosen-plaintext secure (CPA-secure) VPE with an arbitrary flavor into chosen-ciphertext secure (CCA-secure) one with the same flavor. For example, if it is possible to show that a CPA-secure spatial encryption scheme provides verifiability (this is, for example, the case for the spatial encryption scheme presented in [9]), we immediately obtain a CCA-secure spatial encryption scheme. One might think that this can easily be achieved by applying the Canetti-Halevi-Katz [12,10] technique. However, it is unclear whether this technique can be adapted to PE in general, and specifically, for concrete special cases of PE such as inner product

encryption and broadcast encryption, the Canetti-Halevi-Katz technique cannot be applied in a straight forward manner. The method applied in our conversion is closer related to the Naor-Yung technique [26]. We also remark that the techniques presented in this paper can be seen as a non-trivial generalization of the technique presented in [37].

Secondly, we show that a VPE scheme allows the construction of an anonymous deniable predicate authentication (ADPA) scheme. This primitive enables a user to prove to a verifier that he is the owner of a private key corresponding to a specific set of attributes while at the same time being able to deny ever having interacted with the verifier. More specifically, for a ciphertext attribute Y, possibly chosen at the time of authentication, a user can authenticate a message to a verifier using a private key with attribute X such that $R(X, Y)$ is satisfied. The deniability property furthermore guarantees that the verifier's view of the communication can be produced a posteriori without the knowledge of the private key corresponding to X. Hence, the transcript of the interaction cannot be used as evidence of the user authenticating the message to the verifier, and the user will be able to deny ever having done so. ADPA will furthermore guarantee anonymity of the user in the sense that the verifier will not be able to determine the attribute X of the private key of the user, but will only be able to verify that $R(X, Y)$ is satisfied. In other words, the verifier will be able to confirm that the user belongs to the set of users to which the key authority issued a key with property X such that $R(X, Y)$ is satisfied. This anonymity property is guaranteed to hold even if verifier collude with the authority issuing the private keys of the users. While not being directly comparable, this is reminiscent of the properties provided by anonymous credentials which allows a user to demonstrate knowledge of a credential issued by an authority, but without revealing his identity.

Lastly, we show that many concrete PE schemes already provide verifiability, and furthermore demonstrate that many PE schemes which do not provide verifiability, can be easily converted into schemes providing verifiability. Our conversion techniques are applicable to a wide range of (non-verifiable) PE schemes. As examples, we briefly discuss how Waters BE scheme [35], Attrapadung-Libert inner product encryption (IPE) scheme and spatial encryption scheme [1], and Okamoto-Takashima KP-ABE schemes [28] can be transformed into schemes providing verifiability by introducing only simple modifications. This shows that efficient and practical VPE scheme can be constructed, which, due to the results presented in this paper, implies that efficient and CCA-secure variants of these schemes can be obtained as well.

Related Works on PE. In its simplest form, PE corresponds to id-based encryption [5,4,34]. Sahai and Waters [33] proposed the first ABE system with much more expressive relations called *Fuzzy IBE*. It was subsequently generalized to support general access policies by [17,7,29,22]. These results are proved secure in a weak model called selective security. The first fully-secure ABE systems were given by Lewko et al. [21] and Okamoto and Takashima [28], following the general dual-system encryption methodology introduced in [35,24].

When efficiency is the main consideration, the first system with constant-size ciphertexts and with reasonably expressive policies was proposed by Boneh and Hamburg [9], where a system called *spatial encryption* was presented. A fully-secure scheme for spatial relations was then proposed by Attrapadung and Libert [2], where its extension to support inner-product, of which many applications such as CNF/DNF formulae policy expressions as described in [20], was also given. All the aforementioned systems so far do not concern the security property regarding the privacy of ciphertext attributes. The first *attribute-hiding* predicate encryption, or equivalently known as *functional encryption* was suggested by Boneh and Waters [11] and generalized by Katz, Sahai, and Waters [20] to support inner product relations. These attribute-hiding systems were recently made fully secure in [21,28].

Related Works on Deniable Authentication. The formal treatment of *deniability* for public key authentication was initiated by Dwork, Naor and Sahai in their paper on concurrent zero-knowledge[13], followed by a series of papers [25,30,31]. In [13], Dwork et al. propose a deniable authentication protocol based on a CCA-secure encryption scheme. Naor [25] later extended the work by Dwork et al., and introduced the concept of *deniable ring authentication* by combining the approach of Dwork et al. and the paradigm of *ring signatures* proposed by Rivest et al. [32]. In a deniable ring authentication, a member of a ring can authenticate a message in a deniable way to a receiver. Another approach not relying on CCA-secure encryption scheme was proposed by Raimondo and Gennaro [31]. They successfully eliminate the need for CCA-secure encryption by using another primitive, i.e., multi-trapdoor commitments [14]. It should be noted that all these works are in the plain model. Meanwhile, Pass in [30] investigates the possibility of constructing deniable zero-knowledge protocols in the non-plain models, i.e., the common reference string model and random oracle model. Pass shows an impossibility result regarding the construction of non-trivial deniable zero-knowledge protocols in the common reference string model, and a positive result, in the random oracle model, regarding the construction of efficient deniable zero-knowledge arguments of knowledge which preserve both the zero-knowledge property and the proof of knowledge property under concurrent executions.

Notations. $a \xleftarrow{\$} A$ denotes the action of picking a from uniform random distribution over A. $negl(\lambda)$ denotes negligible function in λ. $A \overset{c}{\approx} B$ denotes A and B are computationally indistinguishable. $[A(x)]$ for randomized algorithm A and its input x denotes a set $\{y | \Pr[A(x) = y] \neq 0\}$.

2 Definition of Verifiable Predicate Encryption

In this section we introduce the definition and security notion for PE, and furthermore introduce verifiability. Note that our definition of verifiability is similar but slightly different from the definitions given in [37]. More specifically, the definition given in [37] explicitly requires a **Verify** algorithm, whereas our definition

defines verifiability as a property of the decryption algorithm. Thus, our definition of the verifiability is more similar to that of [18]. We also note that our definition of verifiability is orthogonal to the notion defined in [6].

2.1 Definition of Predicate Encryption

Here, we define the notion of predicate encryption.

Syntax. Let $R = \{R_n : A_n \times B_n \to \{0,1\} \mid n \in \mathbb{N}\}$ be a relation family where A_n and B_n denote "key attribute" and "ciphertext attribute" spaces. A predicate encryption (PE) scheme for R consists of the following algorithms:

Setup$(\lambda, n) \to (PK, MSK)$: The setup algorithm takes as input a security parameter λ and a dimension n of the relation R and outputs a public key PK and a master secret key MSK.

KeyGen$(MSK, PK, X) \to SK_X$: The key generation algorithm takes as input the master secret key MSK, the public key PK, and a key attribute $X \in A_n$. It outputs a private key SK_X. We assume X is included in SK_X implicitly.

Encrypt$(PK, \mathsf{M}, Y) \to CT$: The encryption algorithm takes as input a public key PK, the message M, and a ciphertext attribute $Y \in B_n$. It will output a ciphertext CT.

Decrypt$(PK, CT, Y, SK_X) \to \mathsf{M}$ or $\perp$: We assume that the decryption algorithm is deterministic. The decryption algorithm takes as input the public parameters PK, a ciphertext CT, ciphertext attribute $Y \in B_n$ and a private key SK_X. It outputs the message M or $\perp$ which represents that the ciphertext is not in a valid form. We require that the decryption algorithm outputs $\perp$ if $R(X, Y) = 0$.

We require correctness of decryption: that is, for all λ, all n, all $(PK, MSK) \in$ [**Setup**(λ, n)], all $X \in A_n, Y \in B_n$ such that $R(X, Y) = 1$, all $CT \in$ [**Encrypt** (PK, M, Y)] and all $SK_X \in$ [**KeyGen**(MSK, PK, X)], **Decrypt**$(PK, CT, Y,$ $SK_X) = \mathsf{M}$ holds.

Security. We now define the security notion indistinguishability under chosen ciphertext attack (CCA-security) for an PE scheme Π. This security notion is defined by the following game between a challenger and attacker $\mathcal{A}$.

At first, the challenger runs the setup algorithm and gives PK to $\mathcal{A}$. Then $\mathcal{A}$ may adaptively make key-extraction queries and decryption queries. We denote this phase **Phase1**. In this phase, if $\mathcal{A}$ submits X to the challenger, the challenger returns $SK_X \leftarrow$ **KeyGen**(MSK, PK, X) if X has not been submitted before. Otherwise, the challenger returns the previously extracted SK_X. If $\mathcal{A}$ submits (CT, Y, X) to the challenger in a decryption query, the challenger extracts the private key for X by $SK_X \leftarrow$ **KeyGen**(MSK, PK, X) if this has not been previously extracted and returns the output of **Decrypt**(PK, CT, Y, SK_X) to $\mathcal{A}$. At some point, $\mathcal{A}$ outputs two equal length messages M_0 and M_1 and challenge ciphertext attribute $Y^\star \in B_n$. $Y^\star$ cannot satisfy $R(X, Y^\star) = 1$ for any attribute sets X such that $\mathcal{A}$ already queried private key for X. Then the challenger flips a random coin $\beta \in \{0,1\}$, runs **Encrypt**$(PK, \mathsf{M}_\beta, Y^\star) \to CT^\star$ and gives challenge

ciphertext $CT^\star$ to $\mathcal{A}$. In **Phase2**, $\mathcal{A}$ may adaptively make the same queries as in **Phase1** with following added restriction: $\mathcal{A}$ cannot make a key-extraction query for X such that $R(X, Y^\star) = 1$, and $\mathcal{A}$ cannot submit (CT, Y, X) such that $R(X, Y^\star) = 1$ and $(CT, Y) = (CT^\star, Y^\star)$. At last, $\mathcal{A}$ outputs a guess β' for β. We say that $\mathcal{A}$ succeeds if $\beta' = \beta$ and denote the probability of this event by $\Pr^{PE}_{\mathcal{A},\Pi}$. The advantage of an attacker $\mathcal{A}$ is defined as $Adv^{PE}_{\mathcal{A},\Pi} = \Pr^{PE}_{\mathcal{A},\Pi} - \frac{1}{2}$.

Definition 1. *We say that an PE scheme Π is* CCA-secure *(payload hiding)*[1] *if for all PPT $\mathcal{A}$, $Adv^{PE}_{\mathcal{A},\Pi}$ is negligible. We also say that an PE scheme Π is* CPA-secure *if for all PPT $\mathcal{A}$ who does not make any decryption queries, $Adv^{PE}_{\mathcal{A},\Pi}$ is negligible.*

We say that the PE scheme is selectively CCA/CPA-secure if we add an Initial stage **Init** before the setup where the adversary submits the target ciphertext attribute $Y^\star \in B_n$.

Typical Relations. An PE scheme captures the functionality of a large number of existing types of encryption schemes. In the following, we briefly illustrate how the most popular schemes can be obtained from an PE scheme by choosing the relation appropriately.

Broadcast Encryption. Broadcast encryption allows a sender to encrypt a message for any subset S of n users. To achieve this functionality, we set n to be the number of user, $A_n = \{1, 2, \ldots n\}$, $B_n = 2^{\{1,2,\ldots n\}}$. We define $R_n(j, S) = 1$ if and only if $j \in S$ for $j \in A_n$, $S \in B_n$.

Inner Product Encryption (for Non-Zero Relation). Inner product encryption (resp. for non-zero relation) allows a sender to encrypt a message for a vector $\boldsymbol{Y}$ so that a user with a secret key for a vector $\boldsymbol{X}$, can decrypt it if and only if $\boldsymbol{X} \cdot \boldsymbol{Y} = 0$ (resp. $\boldsymbol{X} \cdot \boldsymbol{Y} \neq 0$). To achieve this functionality, we set n to be dimension of the vectors, $A_n = \mathbb{Z}_N^n$, and $B_n = \mathbb{Z}_N^n$ where N is some integer determined by the scheme. We define $R_n(\boldsymbol{X}, \boldsymbol{Y}) = 1$ if and only $\boldsymbol{X} \cdot \boldsymbol{Y} = 0$ (resp. $\boldsymbol{X} \cdot \boldsymbol{Y} \neq 0$) for $\boldsymbol{X} \in A_n$, $\boldsymbol{Y} \in B_n$.

Spatial Encryption. Spatial encryption allows a sender to encrypt a message for some vector $\boldsymbol{V}$ so that a user with secret key for a space V such that $\boldsymbol{Y} \in V$ can decrypt it. To achieve this functionality, we set n to be the dimension of the vector, $A_n = \{\mathsf{Aff}(M, \boldsymbol{a}) | M \in \mathbb{Z}_N^{n \times l}, 0 \leq l \leq n, \boldsymbol{a} \in \mathbb{Z}_N^n\}$, $B_n = \mathbb{Z}_N^n$ where $\mathsf{Aff}(M, \boldsymbol{a}) = \{M\boldsymbol{x}^\top + \boldsymbol{a}^\top | \boldsymbol{x} \in \mathbb{Z}_N^l\}$ which is subspace of $\mathbb{Z}_N^n$. We define $R_n(V, \boldsymbol{Y}) = 1$ if and only if $\boldsymbol{Y}^\top \in V$ for $V \in A_n$, $\boldsymbol{Y} \in B_n$.

Key (Ciphertext) Policy Attribute based Encryption. Key (resp. ciphertext) policy attribute based encryption allows a sender to encrypt a message for some set of attribute S (resp. access structure $\mathbb{A}$) so that a user with secret key for an access structure $\mathbb{A}$ (resp. set of attribute S) such that $S \in \mathbb{A}$ can decrypt it. To achieve this functionality, we set n to be the size of attribute universe U. A_n is the collection of access structures over U (resp. $A_n = 2^U$). B_n is set as $B_n = 2^U$ (resp. access structure over U). Here, access

[1] In this paper, we work only on payload-hiding security and not attribute-hiding which is considered for many other predicate encryption schemes such as [20].

structure over U can be described by linear secret sharing (LSSS) matrix whose size is bounded by some polynomial. We define $R_n(\mathbb{A}, S) = 1$ if and only if $\mathbb{A} \in A_n$ accepts $S \in B_n$.

2.2 Definition of Verifiability

In this subsection, we define verifiability of an PE scheme. Intuitively, verifiability guarantees that the decryption of any ciphertext is the same regardless of which user decrypt it, as long as this user is authorized to decrypt.

Definition 2. (VERIFIABLITY) *An PE scheme Π is said to have* verifiability *if for all λ, n, $(PK, MSK) \in [\mathbf{Setup}(\lambda, n)]$, $X, X' \in A_n$, $Y \in B_n$ the following holds.*

If $SK_X \in [\mathbf{KeyGen}(MSK, PK, X)]$, $SK_{X'} \in [\mathbf{KeyGen}(MSK, PK, X')]$, and $R(X, Y) = R(X', Y)$, then for all $CT \in \{0,1\}^*$, $\mathbf{Decrypt}(PK, CT, Y, SK_X) = \mathbf{Decrypt}(PK, CT, Y, SK_{X'})$ holds.

We remark that verifiability is not implied by correctness, since the definition of correctness is only concerned about correctly generated ciphertext whereas the definition of verifiability needs is about any ciphertext (including invalid one).

We also define public verifiability which is stronger notion than verifiability. That is, we can convert any PE scheme with public verifiability into PE scheme with verifiability very easily as we explain later. The reason why we introduce the notion of public verifiability is that in many case, we can check whether an PE scheme have public verifiability or not very easily.

Definition 3. (PUBLIC VERIFIABLITY) *An PE scheme Π is said to have* public verifiability *if there exists a polynomial-time algorithm* **Verify** *which takes as input the public key PK, a possible ciphertext $CT \in \{0,1\}^*$, a ciphertext attribute $Y \in B_n$ and outputs* 0 *or* 1*. We require that for all λ, n, $(PK, MSK) \in [\mathbf{Setup}(\lambda, n)]$, $Y \in B_n$, $CT \in \{0,1\}^*$,*

$$\mathbf{Verify}(PK, CT, Y) = 1 \Leftrightarrow \exists \mathsf{M} \text{ such that } CT \in [\mathbf{Encrypt}(PK, \mathsf{M}, Y)].$$

An PE scheme with public verifiability can be modified to be verifiable by changing decryption algorithm slightly. That is, modified decryption algorithm $\mathbf{Decrypt}'(PK, CT, Y, SK_X)$ first checks whether $\mathbf{Verify}(PK, CT, Y) = 1$ holds and outputs $\mathbf{Decrypt}(PK, CT, Y, SK_X)$ if it holds. Otherwise it outputs $\perp$.

3 CCA-Secure VPE from CPA-secure VPE

In this section, we show that VPE for a large class of relations can be transformed to be CCA-secure VPE with the same relation. Our requirement for this transformation is very weak, and many important relations defined for PE schemes satisfy this requirement. Our conversion works for wide class of PE such as spatial encryption, IPE, BE, KP/CP-ABE and can be seen as a nontrivial generalization of the conversion proposed by [37] which only works for ABE. We also remark that our conversion works for BE and IPE for which the Canetti-Halevi-Katz [12] transform cannot be applied in a straightforward manner.

3.1 Definitions

We define the notions of "OR-compatibility" and "equality test" for a relation. Intuitively, a relation is said to have OR-compatibility if for two attributes, the relation is able to capture the presence of one *or* the other, whereas a relation is said to support equality test over a domain D if it can be used to emulate an equality test for elements in D. The formal definitions are as follows:

Definition 4. (OR-compatibility) *Consider a relation family* $R = \{R_n : A_n \times B_n \to \{0,1\} \mid n \in \mathbb{N}\}$. *We say that* R *is OR-compatible if for all* $n, m \in \mathbb{N}$ *there are maps* $\mathsf{OR} : B_n \times B_m \to B_{n+m}$ *and* $s : A_n \to A_{n+m}$ *and* $t : A_m \to A_{n+m}$ *such that for all* $X_1 \in A_n, X_2 \in A_m$ *and* $Y_1 \in B_n, Y_2 \in B_m$ *it holds that*

$$R_{n+m}(s(X_1), \mathsf{OR}(Y_1, Y_2)) = R_n(X_1, Y_1), R_{n+m}(t(X_2), \mathsf{OR}(Y_1, Y_2)) = R_m(X_2, Y_2).$$

Definition 5. (Equality test) *Consider a relation family* $R = \{R_n : A_n \times B_n \to \{0,1\} \mid n \in \mathbb{N}\}$. *Consider a set* D. *We say that* R *can perform equality test over* D *by using dimension* d *if there are maps* $u : D \to A_d$ *and* $v : D \to B_d$ *such that for all* $z, z' \in D$ *we have* $R_d(u(z), v(z)) = 1$ *and* $R_d(u(z), v(z')) = 0$ *if* $z \neq z'$.

3.2 Generic Conversion

Let $\Pi = (\mathbf{Setup}, \mathbf{KeyGen}, \mathbf{Encrypt}, \mathbf{Decrypt})$ be a CPA-secure PE for relation R and let $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ be a strongly unforgeable one-time signature scheme. Here, $\mathcal{G}$, $\mathcal{S}$, and $\mathcal{V}$ are the key generation, sign, and verify algorithms of the scheme, respectively. Assume that Π has verifiability, OR-compatibility (as per definition 4), and can perform equality test (as per definition 5) over the verification key space of Σ. We can construct a CCA-secure VPE scheme $\Pi' = (\mathbf{Setup}', \mathbf{KeyGen}', \mathbf{Encrypt}', \mathbf{Decrypt}')$ also for relation R as follows.

Setup$'(\lambda, n)$**.** Output **Setup**$(\lambda, n+d) \to (PK, MSK)$.

KeyGen$'(MSK, PK, X)$**.** Output **KeyGen**$(MSK, PK, s(X)) \to SK_{s(X)}$. Hence $SK'_X = SK_{s(X)}$.

Encrypt$'(PK, \mathsf{M}, Y)$ First create a one-time signature key pair by running $\mathcal{G}(\lambda) \to (vk, sk)$. Then run **Encrypt**$(PK, \mathsf{M}, \mathsf{OR}(Y, v(vk))) \to CT$ and $\mathcal{S}(sk, (CT, Y)) \to \sigma$. Lastly, output $CT' = (vk, CT, \sigma)$.

Decrypt$'(PK, CT', Y, SK'_X)$ Parses the ciphertext CT' as (vk, CT, σ). If $\mathcal{V}(vk, (CT, Y), \sigma) = 0$, output $\perp$. Output **Decrypt**$(PK, CT, \mathsf{OR}(Y, v(vk)), SK'_X)$ otherwise.

Correctness. Decryption can be done using $SK_{s(X)}$ if $R_n(X, Y) = 1$ since $R_{n+d}(s(X), \mathsf{OR}(Y, v(vk))) = R_n(X, Y) = 1$. Thus, correctness of Π' follows from correctness of Π.

Verifiability. The verifiability of Π' follows directly from the verifiability of Π.

Selective security. Our conversion can be also applied to selectively (CPA-)secure PE schemes, and in such cases, resulting CCA-secure schemes are only selectively (CCA-)secure as well.

Theorem 1. *If Π is CPA-secure PE for relation R, then Π' is CCA-secure PE for relation R.*

Proof. Assume we are given PPT adversary $\mathcal{A}$ which breaks CCA-security of the scheme Π' for relation R_n with advantage ϵ. Then we construct another adversary $\mathcal{B}$ which breaks CPA-security of the scheme Π for relation R_{n+d} with advantage negligibly close to ϵ using $\mathcal{A}$. Define adversary $\mathcal{B}$ as follows:

Setup. The challenger runs $\mathbf{Setup}(\lambda, n+d) \rightarrow (PK, MSK)$. Then $\mathcal{B}$ is given PK and gives it to $\mathcal{A}$. $\mathcal{B}$ also runs $\mathcal{G}(\lambda) \rightarrow (vk^\star, sk^\star)$.

Phase1. $\mathcal{A}$ may adaptively make queries of the following types:

– ***Key-extraction query.*** When $\mathcal{A}$ submits X, then $\mathcal{B}$ submits $s(X)$ to challenger. $\mathcal{B}$ is given private key $SK_{s(X)}$ and gives it to $\mathcal{A}$.

– ***Decryption query.*** When $\mathcal{A}$ submits (CT', Y, X) such that $CT' = (vk, CT, \sigma)$, $\mathcal{B}$ respond to $\mathcal{A}$ as follows. First, $\mathcal{B}$ checks whether $R(X, Y) = 0$ or not. If so, $\mathcal{B}$ outputs $\perp$. Otherwise $\mathcal{B}$ checks whether $\mathcal{V}(vk, (CT, Y), \sigma) = 1$ holds. If it does not hold, then $\mathcal{B}$ returns $\perp$. If it holds and $vk^\star = vk$, then $\mathcal{B}$ aborts. Otherwise $\mathcal{B}$ submits $t(u(vk))$ to the challenger and is given $SK_{t(u(vk))}$. Then $\mathcal{B}$ returns output of $\mathbf{Decrypt}(PK, CT, \mathsf{OR}(Y, v(vk)), SK_{t(u(vk))})$ to $\mathcal{A}$.

Challenge. $\mathcal{A}$ declares two equal length messages M_0 and M_1 and an challenge attribute $Y^\star$. Then $\mathcal{B}$ declares the same messages M_0, M_1 and $\mathsf{OR}(Y^\star, v(vk^\star))$ for the challenger. The challenger flips a random coin $\beta \in \{0, 1\}$, runs $\mathbf{Encrypt}(PK, \mathsf{M}_\beta, \mathsf{OR}(Y^\star, v(vk^\star))) \rightarrow CT^\star$ and gives $CT^\star$ to $\mathcal{B}$. Then $\mathcal{B}$ runs $\mathcal{S}(sk^\star, (Y^\star, CT^\star)) \rightarrow \sigma^\star$, and gives $CT^{\star\prime} = (vk^\star, CT^\star, \sigma^\star)$ to $\mathcal{A}$.

Phase2. $\mathcal{B}$ responds to $\mathcal{A}$'s queries as the same as in **Phase1**.

Guess. Finally, $\mathcal{A}$ outputs a guess β' for β. Then $\mathcal{B}$ outputs β' as its guess.

First we check that the key extraction query of $\mathcal{A}$ is legal. $\mathcal{B}$ can submit $s(X)$ to the challenger, since $R_{n+d}(s(X), \mathsf{OR}(Y^\star, v(vk^\star))) = R_n(X, Y^\star) = 0$. $\mathcal{B}$ can also submits $t(u(vk))$ to the challenger since $R_{n+d}(t(u(vk)), \mathsf{OR}(Y^\star, v(vk^\star))) = R_d(u(vk), v(vk^\star)) = 0$ if $vk \neq vk^\star$. Next, we see that in the simulation of decryption oracle, $\mathbf{Decrypt}(PK, CT, \mathsf{OR}(Y, v(vk)), SK_{t(u(vk))}) = \mathbf{Decrypt}(PK, CT, \mathsf{OR}(Y, v(vk)), SK_{s(X)})$ by the verifiability since $R_{n+d}(t(u(vk)), \mathsf{OR}(Y, v(vk))) = R_{n+d}(s(X), \mathsf{OR}(Y, v(vk)) = 1$ if $R(X, Y) = 1$. Thus the simulation is perfect if $\mathcal{B}$ does not abort.

Let **Win** denote the event that $\mathcal{A}$ correctly guess β, **Abort** denote the event that $\mathcal{B}$ aborts. If **Abort** does not occur, $\mathcal{B}$'s simulation is perfect. So, $\mathcal{B}$'s advantage for guessing β is estimated as $\Pr[\mathcal{B} \text{ correctly guesses } \beta] - \frac{1}{2} = \Pr[\mathbf{Win}|\overline{\mathbf{Abort}}] \Pr[\overline{\mathbf{Abort}}] - \frac{1}{2} \geq \Pr[\mathbf{Win}] - \Pr[\mathbf{Abort}] - \frac{1}{2} \geq \epsilon - \Pr[\mathbf{Abort}]$. Since $\Pr[\mathbf{Abort}] = negl(\lambda)$ due to the unforgeability of the one-time-signature, the proof is completed.

3.3 Qualifying Relations

In this subsection, we show that important relations defined for PE schemes in the literature satisfy OR-compatibility and can perform equality test by

describing their corresponding maps $\mathsf{OR} : B_n \times B_m \to B_{n+m}, s : A_n \to A_{n+m}, t : A_m \to A_{n+m}, u : D \to A_d, v : D \to B_d$.

Inner Product Relation. Equality test can be performed with $d = 2$.

$$\mathsf{OR}(\boldsymbol{Y}_1, \boldsymbol{Y}_2) = \boldsymbol{Y}_1 || \boldsymbol{Y}_2, \quad s(\boldsymbol{X}) = \boldsymbol{X} || \boldsymbol{0}, \quad t(\boldsymbol{X}) = \boldsymbol{0} || \boldsymbol{X},$$
$$u(z) = (z, 1), \quad v(z) = (-1, z)$$

Spatial Relation. Equality test can be performed with $d = 1$.

$$\mathsf{OR}(\boldsymbol{Y}_1, \boldsymbol{Y}_2) = \boldsymbol{Y}_1 || \boldsymbol{Y}_2, \quad s(\mathsf{Aff}(M, \boldsymbol{a})) = \mathsf{Aff}(\begin{bmatrix} M & 0 \\ 0 & I_m \end{bmatrix}, \boldsymbol{a} || \boldsymbol{0}),$$
$$t(\mathsf{Aff}(M, \boldsymbol{a})) = \mathsf{Aff}(\begin{bmatrix} I_n & 0 \\ 0 & M \end{bmatrix}, \boldsymbol{0} || \boldsymbol{a}), \quad u(z) = \mathsf{Aff}(0, (z)), \quad v(z) = (z)$$

Here, I_m and I_n are unit matrices of size m and n respectively.

We can also capture the case of CP/KP-ABE by a technique in [37]. We need to generalize the definition of equality test to instantiate BE and non-zero IPE in our framework. See the full version for the details.

4 Anonymous Deniable Predicate Authentication

In this section, we introduce the notion of ADPA. Intuitively, ADPA is a generalization of deniable authentication in which the prover holds a private key corresponding to an attribute, and the verifier will learn nothing about this attribute, except that it satisfies a relation with the verifier attribute. Firstly, we define functionality and security, and then show how a ADPA scheme can be constructed from a CCA-secure VPE scheme.

4.1 Definition of Anonymous Deniable Predicate Authentication

Syntax. Let relation $R = \{R_n : A_n \times B_n \to \{0, 1\} \mid n \in \mathbb{N}\}$ be a collection of boolean functions, where $n \in \mathbb{N}$ denotes a "scheme description", A_n and B_n denote the "prover attribute" and "verifier attribute" spaces. An *anonymous deniable predicate authentication* (ADPA) for a relation R is defined by a tuple of four algorithms $\Lambda_R^{\mathrm{DPA}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{P}, \mathsf{V})$. The setup algorithm Setup takes as inputs a security parameter λ, a scheme description $n \in \mathbb{N}$, and outputs a public key PK and a master secret key MSK. And the key generation algorithm KeyGen takes as inputs the master secret key MSK, the public key PK, a prover attribute $X \in A_n$, and outputs a private key SK_X. The interactive Turing machines prover P and verifier V perform the interactive protocol (P, V) with common inputs PK, $Y \in B_n$, message M, where P also takes input the private key for X, SK_X. At the end of protocol (P, V) , V outputs a bit to indicate whether V accepts P as a valid prover or not. With $\langle \mathsf{P}(x_{\mathsf{P}}), \mathsf{V}(x_{\mathsf{V}}) \rangle (y)$, we denote the output of verifier V at the end of execution of interactive protocol (P, V), where P and V take x_{P} and x_{V} as private inputs respectively and y denotes the

common input. For the basic requirement *completeness*, Λ_R^{DPA} needs to satisfy that for all λ, $n \in \mathbb{N}$, $X \in A_n$, $Y \in B_n$ such that $R_n(X,Y) = 1$, M, the following holds.

$$(PK, MSK) \leftarrow \mathsf{Setup}(\lambda, n), SK_X \leftarrow \mathsf{KeyGen}(MSK, PK, X) \ : \\ \langle \mathsf{P}(SK_X), \mathsf{V} \rangle (PK, Y, \mathsf{M}) = 1.$$

For security, Λ_R^{DPA} is also required to satisfy the following notions.

Concurrent Soundness. First, we define the adversary $\mathcal{A}$ as a *man-in-the-middle* attacker such that $\mathcal{A}$ is interacting with provers $\mathsf{P}_1, \ldots, \mathsf{P}_{m_L}$ in m_L "left sessions" as verifier, and at the same time interacting with an honest verifier V in a "right session" as prover, in any arbitrary interleaving, where m_L is polynomial in security parameter λ. The adversary $\mathcal{A}$ is given access to two additional oracles: (1) prover instantiator oracle $\mathcal{P}$, and (2) key generator oracle $\mathcal{K}$.

When $\mathcal{A}$ submits to prover instantiator oracle $\mathcal{P}$ a message M, a verifier attribute Y, and a prover attribute X, $\mathcal{P}$ will initiate a new prover P' with inputs $(PK, Y, \mathsf{M}, SK_X)$, where SK_X is a valid secret key corresponding to the key attribute X. The adversary $\mathcal{A}$ is allowed to send a prover attribute X and then retrieve the corresponding secret key SK_X from the key generator oracle $\mathcal{K}$ with the restriction that $R(X, Y^\star) \neq 1$ holds, where $Y^\star$ is the verifier attribute used as common input in the right session. The following notion guarantees that such adversary $\mathcal{A}$ will not be able to make the honest verifier V to accept it as a valid prover in right session.

Definition 6 (Concurrent Soundness). *Let $\Lambda_R^{\mathrm{DPA}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{P}, \mathsf{V})$ be an ADPA for relation $R = \{R_n : A_n \times B_n \to \{0,1\} \mid n \in \mathbb{N}\}$. We say that Λ_R^{DPA} satisfies concurrent soundness if for all sufficiently large λ, for any $n \in \mathbb{N}$, for any efficient algorithm $\mathcal{A}$, the following holds.*

$$\Pr\left[\mathsf{Setup}(\lambda, n) \to (PK, MSK) \ : \ \begin{matrix} \mathcal{A}^{\mathcal{P},\mathcal{K}}(PK) \to (\mathsf{M}^\star, Y^\star, state) \\ \langle \mathcal{A}^{\mathcal{P},\mathcal{K}}(state), \mathsf{V} \rangle (PK, Y^\star, \mathsf{M}^\star) = 1 \end{matrix}\right] = negl(\lambda),$$

where

- *the key generator oracle $\mathcal{K}$, on input a prover attribute $X_i \in A_n$ such that $R_n(X_i, Y^\star) \neq 1$ holds, returns $SK_{X_i} \leftarrow \mathsf{KeyGen}(PK, MSK, X_i)$,*
- *the prover instantiator oracle $\mathcal{P}$, on input a tuple $(\mathsf{M}_i, Y_i \in B_n, X_i \in A_n)$ such that $(\mathsf{M}_i, Y_i) \neq (\mathsf{M}^\star, Y^\star)$ holds, allows $\mathcal{A}$ access to a prover $\mathsf{P}_i \in \{\mathsf{P}_1, \ldots, \mathsf{P}_{m_L}\}$ which has been initiated with inputs $(PK, Y_i, \mathsf{M}_i, SK_{X_i})$, where $SK_{X_i} \leftarrow \mathsf{KeyGen}(PK, MSK, X_i)$,*
- *$\mathcal{A}$ interacts as a verifier with provers $\mathsf{P}_1, \ldots, \mathsf{P}_{m_L}$ generated by $\mathcal{P}$ concurrently, and for each instantiated prover $\mathsf{P}_1, \ldots, \mathsf{P}_{m_L}$, and $\mathcal{A}$ interacts in the protocol $(\mathsf{P}_i, \mathcal{A})$ with common inputs PK, $Y_i \in B_n$, M, where P_i also takes input SK_{X_i}.*

We remark that we can consider a weaker version of the above security notion in which the adversary is required to output $Y^\star$ at the beginning of the game. We denote this security selective concurrent soundness. Next, we define the special security notions we require an ADPA to satisfy.

Anonymity (Source Hiding). Here we describe the security notion which guarantees that no one is able to know which key attribute is associated to the prover P in the interactive protocol (P, V), even when one is allowed to act as a *cheating* verifier in the interactive protocol and is given access to the master secret key generated by the setup algorithm.

Let $R = \{R_n : A_n \times B_n \to \{0,1\} \mid n \in \mathbb{N}\}$ be a relation and $\Lambda_R^{\mathrm{DPA}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{P}, \mathsf{V})$ be an ADPA for R. Let us consider an adversary $\mathcal{A}$ which engages in the following game.

$\mathsf{Game}_{\mathcal{A}}^{\mathrm{anom}}(\lambda, n)$:
$\mathsf{Setup} \to (PK, MSK)$, $\mathcal{A}(PK, MSK) \to (X_0^\star, X_1^\star)$
$\mathsf{KeyGen}(MSK, PK, X_0^\star) \to SK_{X_0^\star}$, $\mathsf{KeyGen}(MSK, PK, X_1^\star) \to SK_{X_1^\star}$
$\mathcal{A}(PK, MSK) \to (Y^\star, \mathsf{M}^\star, state)$ s.t. $R_n(X_0^\star, Y^\star) = R_n(X_1^\star, Y^\star)$
$b \xleftarrow{\$} \{0,1\}$, $\mathcal{A}^{\mathsf{P}(SK_{X_b^\star})}(state, SK_{X_0^\star}, SK_{X_1^\star}, MSK, PK, Y^\star, \mathsf{M}^\star) \to \hat{b}$
If $b = \hat{b}$ return 1, otherwise return 0.

The notation $\mathcal{A}^{\mathsf{P}(SK_{X_b^\star})}(state, SK_{X_0^\star}, SK_{X_1^\star}, MSK, PK, Y^\star, \mathsf{M}^\star)$ in $\mathsf{Game}_{\mathcal{A}}^{\mathrm{anom}}(\lambda, n)$ denotes that $\mathcal{A}$ interacts as verifier with P in the interactive protocol $(\mathsf{P}, \mathcal{A})$ with common inputs $(PK, Y^\star, \mathsf{M}^\star)$ where P also takes input secret key $SK_{X_b^\star}$ and $\mathcal{A}$ also takes inputs $(state, SK_{X_0^\star}, SK_{X_1^\star}, MSK)$. The following notion guarantees that there no such adversary $\mathcal{A}$ will be able to correctly guess whether P uses $SK_{X_0^\star}$ or $SK_{X_1^\star}$ as its secret key.

Definition 7 (Anonymity (Source Hiding)). *Let $\Lambda_R^{\mathrm{DPA}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{P}, \mathsf{V})$ be an ADPA for relation $R = \{R_n : A_n \times B_n \to \{0,1\} \mid n \in \mathbb{N}\}$. We say that Λ_R^{DPA} satisfies anonymity (source hiding) if for all sufficiently large λ, for any $n \in \mathbb{N}$, for any machine $\mathcal{A}$ with unbounded power, $\left|\Pr\left[\mathsf{Game}_{\mathcal{A}}^{\mathrm{anom}}(\lambda, n) = 1\right] - \frac{1}{2}\right| = negl(\lambda)$ holds.*

Deniability. Here we describe the security notion which guarantees that the communication transcript which is produced from an interaction between prover and verifier in an ADPA cannot be used as a proof that an interaction between prover and verifier has taken place. More precisely, the security notion says that for any verifier (including dishonest verifiers), there exists a simulator which can poses as a valid prover even without knowledge about the secret key. Also, with $\mathsf{View}(\langle P, V \rangle)$, we denote the view which is obtained at the end of interaction between P and V, where the view is the communication transcripts concatenated by random coins used by V.

Definition 8 (Deniability). *Let $R = \{R_n : A_n \times B_n \to \{0,1\} \mid n \in \mathbb{N}\}$ be a relation and $\Lambda_R^{\mathrm{DPA}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{P}, \mathsf{V})$ be an ADPA for relation R. Let us also define the following two probability distributions for a fixed λ, $n \in \mathbb{N}$, M, and $X \in A_n$, $Y \in B_n$ such that $R_n(X, Y) = 1$.*

$$Real(\lambda, n, X, Y, \mathsf{M}) = \begin{bmatrix} \mathsf{Setup}(\lambda, n) \to (PK, MSK), \\ \mathsf{KeyGen}(PK, MSK, X) \to SK_X, \\ \mathsf{View}\left(\langle \mathsf{P}(SK_X), \mathcal{A}(MSK, X)\rangle(PK, Y, \mathsf{M})\right) \end{bmatrix},$$

$$Sim(\lambda, n, X, Y, \mathsf{M}) = \begin{bmatrix} \mathsf{Setup}(\lambda, n) \to (PK, MSK), \\ \mathsf{KeyGen}(PK, MSK, X) \to SK_X, \\ \mathsf{View}\left(\langle \mathsf{Sim}, \mathcal{A}(MSK, X)\rangle(PK, Y, \mathsf{M})\right) \end{bmatrix},$$

where $\mathcal{A}$ *and* Sim *are both efficient algorithms.* Λ_R^{DPA} *is said to be deniable if for all sufficiently large* λ*, for any* $n \in \mathbb{N}$*,* M*, and for all* $X \in A_n$*,* $Y \in B_n$ *such that* $R_n(X, Y) = 1$*, the following holds.*

$$\forall \mathcal{A}\ \exists \mathsf{Sim} : Real(\lambda, n, X, Y, \mathsf{M}) \overset{c}{\approx} Sim(\lambda, n, X, Y, \mathsf{M}).$$

4.2 Construction from CCA-Secure VPE

We can construct an ADPA $\Lambda_R^{\mathrm{DPA}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{P}, \mathsf{V})$ from a CCA secure VPE $\Pi = (\mathbf{Setup}, \mathbf{KeyGen}, \mathbf{Encrypt}, \mathbf{Decrypt})$ and a perfectly binding and computationally hiding commitment scheme $COM = (\mathsf{com}, \mathsf{open})$. Here, com and open are commit and open algorithms of the scheme, respectively. Setup and KeyGen are exactly the same as **Setup** and **KeyGen**. We describe the interactive protocol (P, V) as follows. Note that P and V perform (P, V) with common input the public key PK, a verifier attribute $Y \in B_n$, and a message M, while P also takes as input the secret key SK_X corresponding to a key attribute $X \in A_n$ such that $R_n(X, Y) = 1$ holds.

Step1 $(\mathsf{P} \Leftarrow \mathsf{V})$**:** V chooses randomly $r \leftarrow \{0,1\}^\lambda$ and then computes $CT \leftarrow \mathbf{Encrypt}(PK, \mathsf{M}||r, Y)$. Then V sends CT to P.

Step2 $(\mathsf{P} \Rightarrow \mathsf{V})$**:** P computes $y \leftarrow \mathbf{Decrypt}(PK, CT, Y, SK_X)$. If $y = \perp$ or $y = \mathsf{M}'||r'$ such that $\mathsf{M}' \neq \mathsf{M}$, P chooses random pairs $(r_{i0}, r_{i1}) \in \{0,1\}^\lambda \times \{0,1\}^\lambda$ for $i = 1, \ldots, \lambda$. Otherwise, P chooses $(r_{i0}, r_{i1}) \in \{0,1\}^\lambda \times \{0,1\}^\lambda$ such that $r_{i0} \oplus r_{i1} = r'$ holds for all $i = 1, \ldots \lambda$. Then P sends $\{(C_{i0}, C_{i1}) = (\mathsf{com}(\sigma_{i0}, r_{i0}), \mathsf{com}(\sigma_{i1}, r_{i1}))\}_{i=1,\ldots,\lambda}$ to V, where σ_{i0} and σ_{i1} are randomnesses used to calculate the commitments C_{i0} and C_{i1} respectively.

Step3 $(\mathsf{P} \Leftarrow \mathsf{V})$**:** V sends λ random bits $b_1, b_2, \ldots b_\lambda$ to P.

Step4 $(\mathsf{P} \Rightarrow \mathsf{V})$**:** P sends $\{r_{ib_i} = \mathsf{open}(\sigma_{ib_i}, C_{ib_i}), \sigma_{ib_i}\}_{i=1,2,\ldots,\lambda}$ to V.

Step5 $(\mathsf{P} \Leftarrow \mathsf{V})$**:** V opens CT by revealing r and ρ to P, where ρ is randomness used to create CT.

Step6 $(\mathsf{P} \Rightarrow \mathsf{V})$**:** P sends $\{r_{i\bar{b}_i} = \mathsf{open}(\sigma_{i\bar{b}_i}, C_{i\bar{b}_i}), \sigma_{i\bar{b}_i}\}_{i=1,2,\ldots,\lambda}$ to V.
V outputs 1 if for all $1 \leq i \leq \lambda$, $r_{i0} \oplus r_{i1} = r$, and outputs 0 otherwise.

We remark that our conversion can also be applied to selectively CCA-secure PE schemes, and in this case, the resulting ADPA schemes satisfies only selective concurrent soundness.

4.3 Security Analysis

Theorem 2. *If Π is CCA-secure VPE and COM is perfectly binding and computationally hiding commitment, then ADPA Λ_R^{DPA} constructed as above satisfies concurrent soundness, deniability, and anonymity. Especially, source hiding is satisfied for any adversary (even unbounded).*

The theorem can be proved following a very similar strategy to that of [15,25], and we will therefore only sketch how the proof is obtained in the following.

Lemma 1. *(Concurrent Soundness.) If Π is CCA-secure PE and COM is perfectly secure binding commitment, then Λ_R^{DPA} satisfies concurrent soundness.*

Similar to the case of [25], it is easy to see that the above construction satisfies concurrent soundness. We can construct an IND-CCA adversary $\mathcal{B}$ of the underlying predicate encryption using adversary $\mathcal{A}$ who violates soundness of above scheme. Note that $\mathcal{B}$ can easily simulate the prover instantiator oracle $\mathcal{P}$ perfectly using the given decryption oracle and also simulate the key generator oracle $\mathcal{K}$ perfectly via key extraction queries. The key point of the proof is that $\mathcal{B}$ is allowed to rewind $\mathcal{A}$ and to let $\mathcal{A}$ answer two different sequences of $\{b_i\}_{i=1,\ldots,\lambda}$ in **Step3**, so that $\mathcal{B}$ is able to compute $r = r_{i0} \oplus r_{i1}$ for some $i \in [1, \lambda]$. In the challenge phase, $\mathcal{B}$ can select two messages M_0, M_1 such that $\mathsf{M}_0 = \mathsf{M}||\widehat{r}$, $\mathsf{M}_1 = \mathsf{M}||\widetilde{r}$. and then forward the received challenge ciphertext $CT^\star$ to $\mathcal{A}$ in **Step1**. Since $\mathcal{B}$ can obtain r from $\mathcal{A}$ through the rewinding described above, where $\mathsf{M}||r$ is the result of the decryption of $CT^\star$, $\mathcal{B}$ can check whether $\widehat{r} = r$ or $\widetilde{r} = r$, and thereby easily determine whether $CT^\star$ is the encryption of M_0 or M_1.

Lemma 2. *(Deniability.) If Π satisfies correctness and COM is computationally hiding commitment, then Λ_R^{DPA} satisfies deniability.*

One can prove that the above instantiation is deniable using the same techniques as shown in [15,25]. Intuitively, the procedure to construct the simulator Sim is to firstly run the interaction with the verifier until **Step5** where the verifier has to reveal the randomness r it used in **Step1** to create the CT, and then rewind the verifier until the end of **Step1**. In the second run after the rewind, we can easily simulate a prover until the last step **Step6**, since the randomness r should have been obtained in the first run. (Here, we resort to correctness of Π.) The most crucial point here is how to safely perform **Step2** in the first run (before the rewind). The trick is that although we do not know the randomness r yet, we can send commitments of random messages to $\mathcal{A}$ in **Step2**, as the computationally hiding property of the underlying commitment COM prevents $\mathcal{A}$ from detecting that the commitments sent by Sim are actually commitments to random messages.

Lemma 3. *(Anonimity) If Π is VPE, then Λ_R^{DPA} satisfies anonymity.*

Anonymity of the scheme follows immediately from verifiability. Notice that the difference between an interaction with a prover which uses $SK_{X_0^\star}$ and an interaction with a prover which uses $SK_{X_1^\star}$ will only possibly occur at **Step2**, when

the prover decrypts the ciphertext CT sent by the verifier at **Step1**. Thanks to the verifiable property of the underlying verifiable predicate encryption scheme, the result of the decryption is always the same, both in the case of $SK_{X_0^\star}$ and $SK_{X_1^\star}$, as long as $R_n(X_0^\star, Y) = R_n(X_1^\star, Y)$. Note that security level of anonymity achieved by our scheme is stronger than that of [25]. We achieve anonymity even against an adversary with unbounded computational power, whereas [25] only achieves anonymity against a computational bounded adversary.

5 Instantiations

To be able to apply our framework for constructing CCA-secure PE schemes or ADPA schemes, we require that the underlying PE schemes are verifiable. We note that many selectively-secure PE schemes [8,9,17,29,22,3,7,16,36] have public verifiability. That is, we can construct an **Verify** algorithm (as per definition 3). Hence, these can be used directly in our framework. On the other hand, this is not the case for the PE schemes with full security [35,1,2,21,28]. This is because all existing fully secure PE schemes make use of the dual system encryption methodology [35]. The security of these schemes rely on the indistinguishability between normal ciphertexts and semi-functional ciphertexts where a semi-functional ciphertext is special kind of incorrectly generated ciphertext. To achieve public verifiability, we should be able to distinguish between a semi-functional (i.e. incorrectly generated) ciphertext and a normal (i.e. correctly generated) ciphertext efficiently, but this conflicts with the security of the scheme. However, even though we cannot achieve public verifiability for these schemes, it is possible to achieve our definition of (non-public) verifiability. Recall that our definition of verifiability does not require that we can check whether ciphertext is correctly generated or not, but only requires that we can check whether the decryption of a ciphertext under a different secret key will be the same or not.

In the following, we first discuss how we add verifiability to the schemes in [35,1,2,21,28], then focus on the schemes which we obtain by applying our framework to the above mentioned verifiable PE schemes.

Table 1. Overview of existing PE schemes. In the table. "PubVer" represents that the scheme has public verifiability. "Veri" represents that the schemet can be modified to be verifiable.

Schemes		Type	Verif.	Security	Assumption
Boneh et al.	[8, Sec. 3]	BE	PubVer	selective	D-l-BDHE
Boneh et al.	[8, Sec. 5]	BE	PubVer	selective	D-l-BDHE
Waters	[35, Sec. 5]	BE	Veri	full	DLIN and DBDH
Boneh et al.	[9]	Spatial	PubVer	selective	BDDHE
Attrapadung et al.	[2, Sec. B]	Spatial	Veri	full	3assumptions
Attrapadung et al.	[1]	IPE	Veri	full	DLIN and DBDH

Modifying Existing Schemes to be Verifiable. Here, we explain how we modify the schemes in AL10 spatial encryption scheme [2], OT10 KP-ABE scheme [28], Waters09 BE scheme [35], AL10 IPE scheme [1] to be verifiable. Our first approach is to modify the original scheme so that its decryption algorithm first checks the validity of a ciphertext to a certain extent. We cannot check the validity of the ciphertext perfectly because of the above reason, but for the AL10 spatial encryption scheme and the OT10 KP-ABE scheme, this partial validity check is enough to prove verifiability. (We remark that in the modification, we also make some parts of master secret key public. The anonymity of OT10 scheme is lost by this modification.) For the Waters09 BE and AL10 IPE schemes, the above strategy does not seem to be enough. We then further modify these schemes so that a user has some additional keys for the same attributes, but which uses different randomness. Then, in the decryption algorithm, the user checks whether the decryption of a ciphertext using different keys are the same or not. If it is different, then it indicates that the ciphertext is invalid. With this modification, we can prove verifiability of these schemes. For the description of the schemes and proofs of security and verifiability, see the full version of this paper.

CCA secure PE schemes. Since our conversion works for PE schemes which are not captured by the CPA-ABE to CCA-ABE conversion proposed by [37], we obtain a number of new CCA-secure PE schemes. Especially, our conversion works for BE, IPE, and spatial encryption scheme. In Tabel. 1, we list some candidate scheme in this category which we can use as underlying schemes in our framework. Hence, we can obtain an adaptively secure CCA-secure BE scheme by applying our conversion to the Waters09 BE scheme [35]. Furthermore, we also obtain a new selectively and adaptively secure CCA spatial encryption scheme by applying our conversion to the Boneh-Hamburg [9] and AL10 [2] spatial encryption schemes, respectively. Finally, we also obtain a new adaptively secure CCA IPE scheme by applying our conversion to [1]. We also note that it is easy to modify the Katz-Sahai-Waters [20] scheme to have verifiability. But the anonymity of the scheme is lost by this modification. Furthermore, it seems possible to transform the schemes [27] and [21] into verifiable variants. We note that our conversion also works for the ABE schemes [35,1,2,21,28], since our conversion capture the case of ABE as well. We also note that a special case of our conversion is considered in [18] in a context of BE. But they do not consider how to apply the conversion to the Waters09 BE scheme.

ADPA schemes. Our CCA-secure VPE scheme to ADPA scheme conversion works for all the schemes we obtained above. Hence, we can obtain a deniable ring authentication system with (adaptive) concurrent soundness and constant size ciphertexts by applying our conversion to the CCA-secure BE obtained above. As far as we know, this is the first time a scheme with these properties have been proposed. Furthermore, we can obtain an ADPA for a spatial relation and with selective and adaptive concurrent soundness by applying our conversion to the CCA-secure spatial encryption scheme obtained above. We can also obtain an ADPA for an inner product relation with (adaptive) concurrent soundness by

applying our conversion to adaptively secure CCA IPE scheme obtained above. All of these schemes are new types of deniable authentication schemes. We can also see that if we use a CCA-secure ABE as the underlyingscheme (obtained by applying the transformation in [37] to [17,29,22,3,7,16,36]), then we obtain an ADPA for an attribute based relation.

References

1. Attrapadung, N., Libert, B.: Functional Encryption for Inner Product: Achieving Constant-Size Ciphertexts with Adaptive Security or Support for Negation. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 384–402. Springer, Heidelberg (2010)
2. Attrapadung, N., Libert, B.: Functional Encryption for Public-Attribute Inner Products: Achieving Constant-Size Ciphertexts with Adaptive Security or Support for Negation. Journal of Mathematical Cryptology 5(2), 115–158 (2011); This is full version of [1]
3. Attrapadung, N., Libert, B., de Panafieu, E.: Expressive Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 90–108. Springer, Heidelberg (2011)
4. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
5. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
6. Barbosa, M., Farshim, P.: Delegatable Homomorphic Encryption with Applications to Secure Outsourcing of Computation. In: Dunkelman, O. (ed.) CT-RSA 2012. LNCS, vol. 7178, pp. 296–312. Springer, Heidelberg (2012)
7. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-Policy Attribute-Based Encryption. In: IEEE Symposium on Security and Privacy (S&P), pp. 321–334 (2007)
8. Boneh, D., Gentry, C., Waters, B.: Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005)
9. Boneh, D., Hamburg, M.: Generalized Identity Based and Broadcast Encryption Schemes. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 455–470. Springer, Heidelberg (2008)
10. Boneh, D., Katz, J.: Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity-Based Encryption. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 87–103. Springer, Heidelberg (2005)
11. Boneh, D., Waters, B.: Conjunctive, Subset, and Range Queries on Encrypted Data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 535–554. Springer, Heidelberg (2007)
12. Canetti, R., Halevi, S., Katz, J.: Chosen-Ciphertext Security from Identity-Based Encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
13. Dwork, C., Naor, M., Sahai, A.: Concurrent Zero-Knowledge. Journal of the ACM (JACM) 51(6), 851–898 (2004); Preliminary version In STOC 1998
14. Gennaro, R.: Multi-trapdoor Commitments and Their Applications to Proofs of Knowledge Secure Under Concurrent Man-in-the-Middle Attacks. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 220–236. Springer, Heidelberg (2004)

15. Goldreich, O., Kahan, A.: How to Construct Constant-Round Zero-Knowledge Proof Systems for NP. Journal of Cryptology 9, 167–189 (1996)
16. Goyal, V., Jain, A., Pandey, O., Sahai, A.: Bounded Ciphertext Policy Attribute Based Encryption. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 579–591. Springer, Heidelberg (2008)
17. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM CCS 2006, pp. 89–98 (2006)
18. Hanaoka, G., Kurosawa, K.: Efficient Chosen Ciphertext Secure Public Key Encryption under the Computational Diffie-Hellman Assumption. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 308–325. Springer, Heidelberg (2008), Full version is available at http://eprint.iacr.org/2008/211
19. Katz, J.: Efficient and Non-malleable Proofs of Plaintext Knowledge and Applications (Extended Abstract). In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 211–228. Springer, Heidelberg (2003)
20. Katz, J., Sahai, A., Waters, B.: Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008)
21. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010)
22. Lewko, A., Sahai, A., Waters, B.: Revocation Systems with Very Small Private Keys. In: IEEE Symposium on Security and Privacy (S&P), pp. 273–285 (2010)
23. Lewko, A., Waters, B.: Efficient Pseudorandom Functions from the Decisional Linear Assumption and Weaker Variants. In: ACM-CCS 2009, pp. 112–120 (2009)
24. Lewko, A., Waters, B.: New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (2010)
25. Naor, M.: Deniable Ring Authentication. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 481–498. Springer, Heidelberg (2002)
26. Naor, M., Yung, M.: Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In: STOC 1990, pp. 427–437 (1990)
27. Okamoto, T., Takashima, K.: Hierarchical Predicate Encryption for Inner-Products. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 214–231. Springer, Heidelberg (2009)
28. Okamoto, T., Takashima, K.: Fully Secure Functional Encryption with General Relations from the Decisional Linear Assumption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer, Heidelberg (2010), Full version is available at http://eprint.iacr.org/2010/563
29. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: ACM CCS 2007, pp. 195–203 (2007)
30. Pass, R.: On Deniability in the Common Reference String and Random Oracle Model. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 316–337. Springer, Heidelberg (2003)
31. Raimondo, M.D., Gennaro, R.: New approaches for deniable authentication. In: ACM-CCS 2005, pp. 112–121 (2005)
32. Rivest, R., Shamir, A., Tauman, Y.: How to Leak a Secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer, Heidelberg (2001)
33. Sahai, A., Waters, B.: Fuzzy Identity-Based Encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)

34. Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
35. Waters, B.: Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009), Full version is available at http://eprint.iacr.org/2009/385
36. Waters, B.: Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011)
37. Yamada, S., Attrapadung, N., Hanaoka, G., Kunihiro, N.: Generic Constructions for Chosen-Ciphertext Secure Attribute Based Encryption. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 71–89. Springer, Heidelberg (2011)

Public Key Encryption against Related Key Attacks

Hoeteck Wee*

George Washington University
hoeteck@gwu.edu

Abstract. In this work, we present efficient public-key encryption schemes resilient against linear related key attacks (RKA) under standard assumptions and in the standard model. Specifically, we obtain encryption schemes based on hardness of factoring, BDDH and LWE that remain secure even against an adversary that may query the decryption oracle on linear shifts of the actual secret key. Moreover, the ciphertext overhead is only an additive constant number of group elements.

1 Introduction

The traditional model for security assumes that the internal states of the honest parties are completely hidden from the adversary. We often also extend the same assumption to cryptographic hardware devices such as a RSA SecurID token; here, we assume the internal states to be both completely hidden and protected from the adversary. However, recent timing, 'cold-boot' and virtual-machine attacks demonstrated that physical side-channels can leak partial information about internal states of program executions [32, 25, 40]. Similarly, given physical access to a hardware device, we can use fault injection techniques to tamper with and induce modifications to the internal state of the device [10, 8]. When an adversary tampers with the key stored in a cryptographic hardware device and subsequently observes the outcome of the cryptographic primitive under this modified key, we have a related-key attack (RKA) [21, 7]. The key here may be a signing key of a certificate authority or SSL server or a decryption key for an encryption scheme.

1.1 RKA Security

In this work, we study public-key encryption schemes secure against related-key attacks (RKA).

Modeling RKA Security. We follow the definition of RKA security for public-key encryption given by Bellare et. al [7]. The attack is on the secret key, so we are considering a chosen-ciphertext related-key attack (CC-RKA). The decryption oracle refuses to act only when the ciphertext it is given matches the

* Supported by NSF CAREER Award CNS-0953626.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 262–279, 2012.

challenge ciphertext *and* the derived key equals the real one. We will also consider weak CC-RKA security, where the decryption oracle refuses to act whenever the ciphertext it is given matches the challenge ciphertext. Note that both notions imply IND-CCA security [39, 19], which correspond to the special case where the related-key attack uses the identity function.

We view the system as having the following components: algorithms (code), public parameters, public/secret key pairs. Of these, only the public and secret keys are subject to RKAs. The public parameters are system-wide, meaning fixed beforehand and independent of users. In an implementation, these parameters could be hardwired into the algorithm code and stored on tamper-proof hardware, or distributed via some public channel where tampering is infeasible or could be easily detected. In our constructions, the decryption algorithms do not use the public key and therefore we will only consider attacks on secret keys. We note that our model is the same as that considered in prior works [4, 7], though it is by no means the only possible model.

1.2 Our Results

We present the first public-key encryption schemes resilient against linear related key attacks (RKA) under standard assumptions and in the standard model. Specifically, we obtain encryption schemes based on hardness of factoring and BDDH that remain secure even against an adversary that may query the decryption oracle on linear shifts of the actual secret key. In addition, we present schemes based on DDH and LWE that achieve the weaker notion of RKA security where the adversary is not allowed to query the decryption oracle on the challenge ciphertext.

Moreover, in all these schemes, the ciphertext overhead is only an additive constant number of group elements. Our factoring-based scheme is also the first RKA-secure primitive based on standard number-theoretic assumptions related to factoring, as well as the first from search assumptions not related to lattices. (The latter is somewhat surprising in lieu of the negative results in [22], showing that certain natural classes of constructions based on search assumptions cannot achieve RKA-pseudorandomness).

Warm-Up. The starting point of our constructions are CCA-secure encryption schemes in which the decryption of a ciphertext C using a secret key $\phi(\text{SK})$ – where ϕ denotes a linear shift – equals the decryption of some other (efficiently computable) ciphertext C' using the original secret key SK. We refer to this property as key homomorphism. Roughly speaking, this enables us to reduce the CC-RKA-security of the scheme to its CCA-security. The same high-level strategy of exploiting homomorphism was also used in [4, 3] to achieve RKA security for pseudorandom functions and private-key encryption respectively.

The above strategy breaks down whenever the ciphertext C' equals challenge ciphertext in the CCA-security game. We address this problem with the following modifications:

- We work with a tag-based notion of CCA-security [34, 30], where we derive the tag using a strong one-time signature scheme and add a signature to the ciphertext. In addition, we require that the two ciphertexts above C and C' (where C' is derived from C via key homomorphism) share the same tag. We may then consider two cases: if C shares the same tag as the challenge ciphertext, then the one-time signature scheme tells us that C must equal the challenge ciphertext. On the other hand, if C has a different tag from the challenge ciphertext, then so does C' and we can decrypt C' using the decryption oracle in the CCA-security game. This suffices for weak CC-RKA security, where the RKA decryption oracle refuses to act whenever the ciphertext it is given matches the challenge ciphertext.
- In order to achieve "full fledged" CC-RKA security, we need to handle the case where the ciphertext C equals the challenge ciphertext but $\phi(\mathsf{SK}) \neq \mathsf{SK}$. Here, we simply stipulate that the challenge ciphertext is an invalid ciphertext under any key $\mathsf{SK}' \neq \mathsf{SK}$; we refer to this property as finger-printing (c.f. [4, 7]). In other words, a random valid ciphertext (by itself, even without the public key) uniquely determines a consistent secret key.

At this point, it suffices to describe how we instantiate the underlying building blocks, namely a tag-based CCA-secure encryption scheme that achieves both finger-printing and key-homomorphism, as well as an efficient strong one-time signature scheme.

Achieving Finger-Printing. As it turns out, the Cramer-Shoup CCA-secure constructions [15, 16] do not satisfy the finger-printing; this is in some sense inherent since the smoothness requirement in hash proof systems essentially stipulate the secret key has some residual entropy given only its evaluation on a NO instance of the underlying subset membership problem (but not the public key). Instead, we turn to constructions of CCA-secure public-key encryption based on the "all-but-one extraction" paradigm, starting with [9], and further developed in [12, 11, 30, 38, 26, 1, 31, 42, 35]. In these constructions, the secret key is often only a single group element, which makes achieving finger-printing much simpler. While the Cramer-Shoup framework inherently relies on decisional assumptions e.g., the Decisional Diffie-Hellman (DDH) assumption or the quadratic residuosity assumption, the "all-but-one extraction" paradigm admits instantiations from search assumptions, such as factoring. Search assumptions encompass a larger class of intractable problems than decisional assumptions.

Achieving Key Homomorphism. This leads us to our final technical hurdle, namely that CCA-secure public-key encryption schemes based on search assumptions may not be key-homomorphic. Take for instance the Hofheinz-Kiltz factoring-based CCA-secure scheme [26]; it is not key-homomorphic because the underlying Blum-Blum-Shub PRG is not homomorphic. As it turns out, the "all-but-one extraction" paradigm allows us to overcome this hurdle too – informally, the trapdoor decryption algorithm allows us to recover the *seed* of the PRG (for CCA security, it suffices to recover the *output* of the PRG). For this reason, we present our schemes via the framework of adaptive trapdoor relations [42, 31],

which seems particularly suited for our analysis, as it abstracts the "all-but-one" aspect for achieving CCA-security, allowing us to directly focus on the new challenges posed by CC-RKA-security. For the concrete instantiations of CC-RKA-secure encryption, we look at known instantiations of adaptive trapdoor relations given in [42, 35]; we show that the ones based on hardness of factoring and BDDH satisfy key homomorphism and finger-printing, and that the ones based on DDH and LWE satisfy key homorphism.

One-Time Signatures. As a result of independent interest, we present a new strong one-time signature scheme based on hardness of factoring, which is inspired by Groth's one-time signature based on hardness of discrete log [24]. In Appendix B, we also sketch a generic construction of strong one-time signatures starting from any Σ-protocol. In the application to CCA-security and our CC-RKA-secure schemes, we want to design one-time signature schemes where the total cost of key generation and signing is small. In our factoring-based scheme, the signing algorithm does not require knowing the factorization of the modulus and we may therefore use a modulus from the public parameter instead of generating RSA modulus from scratch (which requires a linear number of exponentiations).

1.3 Discussion

There is a general transformation for achieving security against linear related key attacks via algebraic manipulation detection (AMD) codes [18, 20] – in the case of encryption, this requires modifying the key generation algorithm of a CCA-secure encryption scheme, so that the stored secret key is the encoded version of the original secret key, using such a code (thereby increasing the secret key size). The encoding has the property that with high probability any linear shift of a valid codeword can be detected (and in those cases the new decryption algorithm would simply reject). Our constructions achieve several advantages over this generic approach: first, the key generation algorithm coincides with existing CCA-secure encryption schemes. This offers compatibility with existing public key set-ups. Second, we avoid the blow-up in key sizes. Finally, the existing constructions of AMD codes only work over finite fields, which are not applicable to the constructions based on hardness of factoring.

Perspective. In practice it is not clear that security against linear relations would actually be useful for specific applications. As such, we regard our results largely as proof of concept, demonstrating that we can indeed achieve RKA-security for a non-trivial class of functions while paying only a small overhead in efficiency and without changing existing public-key set-ups.

Additional Related Work. The works of Lucks, Goldenberg and Liskov, and Bellare, Cash and Miller [33, 22, 7] gave constructions of RKA-secure primitives from RKA-secure building blocks, but provided no new constructions of the latter and hence of the former. Also, a number of works gave RKA-secure

schemes in the standard model, notably symmetric encryption [2, 3], signatures [23] (based on q-ary assumption) in addition to PRFs [4]; these schemes all rely lattices and Diffie-Hellman type assumptions, none of these are based on number-theoretic assumptions. There are also feasibility results on RKA-secure public-key encryption based on non-standard assumptions, e.g. [28] as well as results on tamper-resilient UC-secure computation [14]. We also point out here that encryption schemes secure against linear related-key attacks have also found applications in garbled circuits used in secure computation [3, 29].

Organization. We present our main construction in Section 3. We present the instantiations from various classes of assumptions in Sections 5 through 6.

2 Preliminaries

Strong One-Time Signatures. For a stateful adversary $\mathcal{A}$, we define the advantage function $\mathsf{Adv.OTS}^{\mathcal{A}}(\lambda)$ to be:

$$\Pr\left[\begin{array}{l|l} \mathsf{Verify}(\mathrm{VKSIG}, M', \sigma') = 1 & (\mathrm{VKSIG}, \mathrm{SKSIG}) \leftarrow \mathsf{SignKeyGen}(1^\lambda); \\ \text{and } (M', \sigma') \neq (M, \sigma) & M \leftarrow \mathcal{A}(\mathrm{VKSIG}); \\ & \sigma \leftarrow \mathsf{Sign}(\mathrm{SKSIG}, M); \\ & (M', \sigma') \leftarrow \mathcal{A}(\sigma) \end{array}\right]$$

A signature scheme is a *strong one-time signature* if for all PPT adversaries $\mathcal{A}$, the advantage $\mathsf{Adv.OTS}^{\mathcal{A}}(\lambda)$ is a negligible function in λ.

Adaptive Trapdoor Relations. Informally, trapdoor functions are a family of functions $\{\mathsf{F}_{\mathrm{FID}}\}$ that are easy to sample, compute and invert with trapdoor, but hard to invert without the trapdoor (we always assume that the functions are injective). In the tag-based setting, the function takes an additional input, namely the tag; also, the trapdoor is independent of the tag. A family of *adaptive trapdoor functions* [31] is one that remains one-way even if the adversary is given access to an inversion oracle, except the adversary cannot query the oracle on the same tag as that in the challenge. In a trapdoor relation, instead of requiring that $\mathsf{F}_{\mathrm{FID}}$ be efficiently computable, we only require that we can efficiently sample from the distribution $(s, \mathsf{F}_{\mathrm{FID}}(\mathrm{TAG}, s))$ for a random s given $\mathrm{FID}, \mathrm{TAG}$.

More precisely, a family of (tag-based) *adaptive trapdoor relations* [42] is given by a family of injective functions $\{\mathsf{F}_{\mathrm{FID}}\}$ that satisfies the following properties:

(TRAPDOOR GENERATION.) There is an efficient randomized algorithm TDG that outputs a random $(\mathrm{FID}, \mathrm{TID})$.

(PUBLIC SAMPLING.) There is an efficient randomized algorithm PSamp that on input $(\mathrm{FID}, \mathrm{TAG})$, outputs $(s, \mathsf{F}_{\mathrm{FID}}(\mathrm{TAG}, s))$ for a random s.

(TRAPDOOR INVERSION.) There is an efficient algorithm TdInv such that for all $(\mathrm{FID}, \mathrm{TID}) \leftarrow \mathsf{TDG}$ and for all TAG, y, computes $\mathsf{TdInv}(\mathrm{TID}, \mathrm{TAG}, y) = \mathsf{F}^{-1}_{\mathrm{FID}}(\mathrm{TAG}, y)$.

(ADAPTIVE ONE-WAYNESS.) For all efficient stateful adversaries $\mathcal{A}$, the following quantity is negligible in λ:

$$\Pr\left[s = s' : \begin{array}{l} \text{TAG}^* \leftarrow \mathcal{A}(1^\lambda); \\ (\text{FID}, \text{TID}) \leftarrow_{\text{R}} \mathsf{TDG}(1^\lambda); \\ (s, y) \leftarrow_{\text{R}} \mathsf{PSamp}(\text{FID}, \text{TAG}^*); \\ s' \leftarrow \mathcal{A}^{\mathsf{F}_{\text{FID}}^{-1}(\cdot,\cdot)}(\text{FID}, y) \end{array}\right]$$

where $\mathcal{A}$ is allowed to query $\mathsf{F}_{\text{FID}}^{-1}(\cdot,\cdot)$ on any tag different from TAG*.

It is convenient to work with the following stronger notion of *adaptive pseudorandomness* [37], where the adversary has to distinguish $\mathsf{G}(s)$ from random given y and an inversion oracle, for some pseudorandom generator G associated with the family $\{\mathsf{F}_{\text{FID}}\}$. There is indeed a generic way to obtain adaptive pseudorandomness from adaptive one-wayness via the Goldreich-Levin hard-core bit (since the proof relativizes with respect to the inversion oracle). However, for the concrete instantiations we consider here, there are more efficient ways to derive multiple hard-core bits.

(ADAPTIVE PSEUDORANDOMNESS.) For all efficient stateful adversaries $\mathcal{A}$, the following quantity is negligible in λ:

$$\Pr\left[b = b' : \begin{array}{l} \text{TAG}^* \leftarrow \mathcal{A}(1^\lambda); \\ (\text{FID}, \text{TID}) \leftarrow_{\text{R}} \mathsf{TDG}(1^\lambda); \\ (s, y) \leftarrow_{\text{R}} \mathsf{PSamp}(\text{FID}, \text{TAG}^*); \\ K_0 := \mathsf{G}(s); K_1 \leftarrow_{\text{R}} \{0,1\}^\lambda; \\ b \leftarrow_{\text{R}} \{0,1\}; \\ b' \leftarrow \mathcal{A}^{\mathsf{F}_{\text{FID}}^{-1}(\cdot,\cdot)}(\text{FID}, y, K_b) \end{array}\right] - \frac{1}{2}$$

where $\mathcal{A}$ is allowed to query $\mathsf{F}_{\text{FID}}^{-1}(\cdot,\cdot)$ on any tag different from TAG*.

2.1 RKA Security

Related-Key Derivation Functions. Following [5], a class of Φ of related-key deriving functions (RKDFs) is a finite set of functions, all with the same domain and range that could possibly depend on the public parameter PP. The class of functions should also admit an efficient membership test, and its functions should be efficiently computable. For our concrete instantiations, we consider the class Φ^+ of linear shifts.

CC-RKA Security. We follow the definition of related-key attack (RKA) security from [7, 4]. For a stateful adversary $\mathcal{A}$, we define the advantage function $\mathsf{Adv.RKA.PKE}^{\mathcal{A},\Phi}(\lambda)$ to be:

$$\Pr\left[b = b' : \begin{array}{l} \text{PP} \leftarrow \mathsf{Setup}(1^\lambda);\ (\text{PK}, \text{SK}) \leftarrow \mathsf{Gen}(\text{PP}); \\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathsf{RKA.Dec}(\text{SK},\cdot,\cdot)}(\text{PP}, \text{PK}),\ |m_0| = |m_1|; \\ b \leftarrow_{\text{R}} \{0,1\}; \\ C^* \leftarrow \mathsf{Enc}(\text{PK}, m_b); \\ b' \leftarrow \mathcal{A}^{\mathsf{RKA.Dec}(\text{SK},\cdot,\cdot)}(C^*) \end{array}\right] - \frac{1}{2}$$

where $\mathsf{RKA.Dec}(\text{SK}, \cdot, \cdot)$ is an oracle that on input (ϕ, C): returns $\mathsf{Dec}(\phi(\text{SK}), C)$. We restrict the adversary $\mathcal{A}$ to only make queries (ϕ, C) such that $\phi \in \Phi$ and $(\phi(\text{SK}), C) \neq (\text{SK}, C^*)$. An encryption scheme is said to be *Φ-CC-RKA secure* if for all PPT $\mathcal{A}$, the advantage $\mathsf{Adv.RKA.PKE}^{\mathcal{A},\Phi}(\lambda)$ is a negligible function in λ.

Weaker CC-RKA Security. We also consider weak CC-RKA security, where in the security experiment, we further restrict the adversary $\mathcal{A}$ to only make queries (ϕ, C) such that $\phi \in \Phi$ and $C \neq C^*$ where C^* is the challenge ciphertext. Previously, we also allow queries (ϕ, C^*) as long as $\phi(\text{SK}) \neq \text{SK}$.

3 Realization from Adaptive Trapdoor Relations

In this section, we present our constructions of RKA-secure encryption via adaptive trapdoor relations. We begin by introducing two additional notions for adaptive trapdoor relations.

***Φ*-Key Homomorphism.** We say that $\{\mathsf{F}_{\text{FID}}\}$ is Φ-key homomorphic if there is a PPT algorithm T such that for all $\phi \in \Phi$ and all $\text{TID}, \text{TAG}, y$:

$$\mathsf{TdInv}(\phi(\text{TID}), \text{TAG}, y) = \mathsf{TdInv}(\text{TID}, \text{TAG}, T(\text{PP}, \phi, \text{TAG}, y))$$

In fact, a weaker formulation that asserts an oracle PPT algorithm T that outputs $\mathsf{TdInv}(\phi(\text{TID}), \text{TAG}, y)$ given oracle access to $\mathsf{TdInv}(\text{TID}, \text{TAG}, \cdot)$ suffices for our proofs. This latter formulation is more similar to the formulation of key-malleability in [4, Section 3.1] for achieving RKA-security for pseudorandom functions. A similar notion also appears in [3] for symmetric-key encryption.

***Φ*-Fingerprinting.** Informally, Φ-fingerprinting stipulates that any attempt to maul TID invalidates a random output of $\mathsf{F}_{\text{FID}}(\cdot)$. More formally, for a stateful adversary $\mathcal{A}$, we define the advantage function $\mathsf{Adv.FP}^{\mathcal{A},\Phi}(\lambda)$ to be:

$$\Pr\left[\begin{array}{l} \mathsf{TdInv}(\phi(\text{TID}), \text{TAG}^*, y) \neq \perp \\ \text{and } \phi \in \Phi \text{ and } \phi(\text{TID}) \neq \text{TID} \end{array} : \begin{array}{l} \text{TAG}^* \leftarrow \mathcal{A}(\text{PP}); \\ (\text{FID}, \text{TID}) \leftarrow \mathsf{TDG}(\text{PP}); \\ (s, y) \leftarrow_{\text{R}} \mathsf{PSamp}(\text{FID}, \text{TAG}^*); \\ \phi \leftarrow \mathcal{A}(\text{PP}, \text{FID}, \text{TID}, y); \end{array}\right]$$

A trapdoor relation admits a *Φ-fingerprint* if for all PPT adversaries $\mathcal{A}$, the advantage $\mathsf{Adv.FP}^{\mathcal{A},\Phi}(\lambda)$ is a negligible function in λ. We stress that in the above experiment, the adversary receives TID, which it can use to compute s from y.

3.1 Our Construction

We present our construction in Fig 1, which is the same as the construction of CCA-secure encryption schemes from adaptive trapdoor relations via strong one-time signatures, as given in [31, 42].

RKA PKE

$\mathsf{Gen}(\text{PP})$: Run $\mathsf{TDG}(\text{PP}) \to (\text{FID}, \text{TID})$. Output $(\text{PK}, \text{SK}) := (\text{FID}, \text{TID})$.

$\mathsf{Enc}(\text{PK}, m)$: On input PK and a message m:

1. Run $\mathsf{SignKeyGen}(\text{PP}) \to (\text{VKSIG}, \text{SKSIG})$;
2. Run $\mathsf{PSamp}(\text{PK}, \text{VKSIG}) \to (s, y)$;
3. Compute $\psi := \mathsf{G}(s) \oplus m$;
4. Run $\mathsf{Sign}(\text{SKSIG}, y\|\psi) \to \sigma$;

Output as ciphertext $\quad \text{VKSIG}\|\sigma\|y\|\psi$

$\mathsf{Dec}(\text{SK}, C)$: On input SK and a ciphertext $C = \text{VKSIG}\|\sigma\|y\|\psi$,

1. Output $\perp$ if $\mathsf{Verify}(\text{VKSIG}, y\|\psi, \sigma) = \texttt{reject}$.
2. Compute $s := \mathsf{TdInv}(\text{TID}, \text{VKSIG}, y)$. Output $\perp$ if $s = \perp$.

Otherwise, output $\quad \mathsf{G}(s) \oplus \psi$

Fig. 1. CC-RKA security from adaptive trapdoor relations

Theorem 1. *Suppose the following hold:*

1. $\{\mathsf{F}_{\text{FID}}\}$ *is a family of adaptive trapdoor relations;*
2. $\{\mathsf{F}_{\text{FID}}\}$ *is Φ-key homomorphic;*
3. $\{\mathsf{F}_{\text{FID}}\}$ *admits a Φ-fingerprinting;*
4. $(\mathsf{SignKeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *is a strong one-time signature scheme.*

Then, $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *as given in Fig 1 is a Φ-CC-RKA secure public-key encryption scheme. Moreover, if all of the conditions hold apart from Φ-fingerprinting, then* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *as given in the above construction is a Φ-weak-CC-RKA secure public-key encryption scheme.*

We observe that correctness of the encryption scheme follows readily from the correctness of trapdoor inversion. Φ-CC-RKA security follows from the next technical claim. After the proof, we explain how to deduce Φ-weak-CC-RKA security without relying on Φ-fingerprinting.

Lemma 1. *Let $\mathcal{A}$ be an adversary against the Φ-CC-RKA security of the above encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *that makes at most Q oracle queries. Then, we can construct an adversary $\mathcal{B}_0$ against the strong one-time security of* $(\mathsf{SignKeyGen}, \mathsf{Sign}, \mathsf{Verify})$*, an adversary $\mathcal{B}_1$ against Φ-fingerprinting of* $\{\mathsf{F}_{\text{FID}}\}$*,*

and an adversary $\mathcal{B}_2$ against adaptive pseudorandomness of $\{\mathsf{F}_{\mathrm{FID}}\}$ and G such that:

$$\mathsf{Adv.RKA.PKE}^{\mathcal{A},\Phi}(\lambda) \leq \mathsf{Adv.OTS}^{\mathcal{B}_0}(\lambda) + \mathsf{Adv.FP}^{\mathcal{B}_1,\Phi}(\lambda) + \mathsf{Adv.Adaptive.PRG}^{\mathcal{B}_2}(\lambda)$$

The running times of $\mathcal{B}_0$ and $\mathcal{B}_1$ are that of $\mathcal{A}$ plus an additional polynomial overhead that grows linearly with Q. The running time of $\mathcal{B}_2$ is similar to that of $\mathcal{A}$, and $\mathcal{B}_2$ makes at most Q oracle queries.

Proof. In the following, we write $C^* = \textsc{vksig}^* \| \sigma^* \| y^* \| \psi^*$ to denote the ciphertext in the Φ-CC-RKA experiment. We proceed via a sequence of games. We start with Game 0 as in the Φ-CC-RKA experiment and end up with a game where the view of $\mathcal{A}$ is statistically independent of the challenge bit b. The sequence of games is analogous to those for obtaining CCA security from all-but-one extractable hash proofs and adaptive trapdoor functions [42, 31]; the main difference lies in handling the RKA queries in the first two games.

GAME 1: ELIMINATING TAG REUSE. We replace the decapsulation mechanism $\mathsf{RKA.Dec}$ with $\mathsf{RKA.Dec}'$ that outputs $\perp$ on ciphertexts $\textsc{vksig} \| \sigma \| y \| \psi$ such that $\textsc{vksig} = \textsc{vksig}^*$ but otherwise proceeds like $\mathsf{RKA.Dec}$. We show that Games 0 and 1 are computationally indistinguishable, by arguing that $\mathsf{RKA.Dec}$ and $\mathsf{RKA.Dec}'$ essentially agree on all inputs $\textsc{vksig} \| \sigma \| y \| \psi$. We consider four cases depending on the input:

- Case 1: $\textsc{vksig} \neq \textsc{vksig}^*$. Here, $\mathsf{RKA.Dec}$ and $\mathsf{RKA.Dec}'$ agree by definition of $\mathsf{RKA.Dec}'$.
- Case 2: $\textsc{vksig} = \textsc{vksig}^*, (\sigma, y \| \psi) = (\sigma^*, y^* \| \psi^*)$ and $\phi(\textsc{sk}) = \textsc{sk}$. Such queries are ruled out by definition of the Φ-CC-RKA security game.
- Case 3: $\textsc{vksig} = \textsc{vksig}^*, (\sigma, y \| \psi) \neq (\sigma^*, y^* \| \psi^*)$. Here, by the security of the signature scheme, we have:

$$\Pr[\mathsf{Verify}(\textsc{vksig}, y \| \psi, \sigma) = 1] \leq \mathsf{Adv.OTS}(\lambda)$$

  Therefore, $\mathsf{RKA.Dec}$ outputs $\perp$ except with negligible probability.
- Case 4: $\textsc{vksig} = \textsc{vksig}^*, (\sigma, y \| \psi) = (\sigma^*, y^* \| \psi^*)$ and $\phi(\textsc{sk}) \neq \textsc{sk}$. Here, by the Φ-fingerprinting property, we have:

$$\Pr[\mathsf{TdInv}(\phi(\textsc{sk}), \textsc{vksig}^*, y) \neq \perp] \leq \mathsf{Adv.FP}(\lambda)$$

  (Here, we use the fact that the adversary in the Φ-fingerprinting experiment is given $\textsc{tid}$, which is needed to simulate the decryption oracle.) Therefore, $\mathsf{RKA.Dec}$ outputs $\perp$ except with negligible probability.

GAME 2: DECRYPTING USING $\mathsf{F}_{\mathrm{FID}}^{-1}(\cdot,\cdot)$. Next, we simulate oracle access to $\mathsf{RKA.Dec}'$ using oracle access to $\mathsf{F}_{\mathrm{FID}}^{-1}(\cdot,\cdot)$ as follows: on input $(\phi, \textsc{vksig} \| \sigma \| y \| \psi)$,

1. If $\textsc{vksig} = \textsc{vksig}^*$ or $\mathsf{Verify}(\textsc{vksig}, y \| \psi, \sigma) = 0$, output $\perp$.
2. Compute $s' := \mathsf{F}_{\mathrm{FID}}^{-1}(\textsc{vksig}, T(\textsc{pp}, \phi, \textsc{vksig}, y))$. Output $\perp$ if $s' = \perp$.
3. Otherwise, output $\psi := \mathsf{G}(s') \oplus \psi$.

Note that we only query $\mathsf{F}_{\text{FID}}^{-1}(\cdot,\cdot)$ on tags different from VKSIG^*. Observe that

$$\begin{aligned} s' &= \mathsf{F}_{\text{FID}}^{-1}(\text{VKSIG}, T(\text{PP}, \phi, \text{VKSIG}, y)) \\ &= \mathsf{TdInv}(\text{TID}, \text{VKSIG}, T(\text{PP}, \phi, \text{VKSIG}, y)) && \text{using trapdoor inversion} \\ &= \mathsf{TdInv}(\phi(\text{TID}), \text{VKSIG}, y) && \text{using } \Phi\text{-key homomorphism} \end{aligned}$$

Correctness of the simulation follows readily, and thus Games 1 and 2 are identically distributed.

GAME 3: REPLACING $\mathsf{G}(\cdot)$ WITH RANDOM. In the computation of $\mathsf{Enc}(\text{PK}, m_b)$ in the $\mathsf{Adv.RKA.PKE}$ experiment, we replace $\psi^* := \mathsf{G}(s^*) \oplus m_b$ with $\psi^* := K \oplus m_b$ where $K \leftarrow_{\text{R}} \{0,1\}^\lambda$. Then, Games 2 and 3 and computationally indistinguishable by adaptive pseudorandomness using VKSIG^* as the selective tag.

We conclude by observing that in Game 3, the distribution of ϕ^* is statistically independent of the challenge bit b. Hence, the probability that $b' = b$ is exactly $1/2$. □

Observe that in the above proof, we only used Φ-fingerprinting in the analysis of Game 1 Case 4. For Φ-weak-CC-RKA security, the queries for this case are ruled out by definition and therefore we do not need Φ-fingerprinting.

4 Instantiations from Hardness of Factoring

Fix a Blum integer $N = PQ$ for λ-bit primes $P, Q \equiv 3 \pmod 4$ such that $P = 2p + 1$ and $Q = 2q + 1$ for primes p, q. Let $\mathbb{J}_N$ denote the subgroup of $\mathbb{Z}_N^*$ with Jacobi symbol $+1$, and let $\mathbb{QR}_N$ denote the subgroup of quadratic residues. Observe that $|\mathbb{J}_N| = 2pq = 2|\mathbb{QR}_N|$. Following [27], we work over the cyclic group of signed quadratic residues, given by the quotient group $\mathbb{QR}_N^+ := \mathbb{J}_N / \pm 1$. $\mathbb{QR}_N^+$ is a cyclic group of order pq and is efficiently recognizable (by verifying that the Jacobi symbol is $+1$). Here, we use a hash function $H : \{0,1\}^* \to \{0,1\}^\lambda$, though we will treat the output of H as a number in $\mathbb{Z}_{2^\lambda}$.

4.1 Strong One-Time Signature

For main construction in Section 3, we require efficient strong one-time signature schemes, where the total computational complexity for key generation and signing is small. In addition, we want short verification key and signatures. Previous factoring-based one-time signatures [41, 36] require generating an RSA modulus during key generation, which is computationally expensive. We provide a new construction that uses a public modulus. For the one-time signature, we can work with any Blum integer $N = PQ$, that is, we do not require that P, Q be safe primes.

$\mathsf{SignKeyGen}(\text{PP}), \text{PP} = (N)$:	$\mathsf{Sign}(\text{SKSIG}, m)$:	$\mathsf{Verify}(\text{VKSIG}, m, (e, w))$
$\text{SKSIG} := (s_0, s_1, x) \leftarrow_{\text{R}} \mathbb{QR}_N^+$ $(u_0, u_1, c) := (s_0^{2^\lambda}, s_1^{2^\lambda}, x^{2^\lambda})$ $\text{VKSIG} := (u_0, u_1, c)$ return $(\text{VKSIG}, \text{SKSIG})$	$e \leftarrow_{\text{R}} \mathbb{Z}_{2^\lambda}$ $w := x \cdot s_0^e \cdot s_1^{\mathsf{H}(m)+e \bmod 2^\lambda}$ return $(e, w) \in \mathbb{Z}_{2^\lambda} \times \mathbb{QR}_N^+$	check $w^{2^\lambda} = c \cdot u_0^e \cdot u_1^{\mathsf{H}(m)+e \bmod 2^\lambda}$

Fig. 2. Factoring-based strong one-time signature

Theorem 2. *Suppose factoring Blum integers is hard on average and H is collision resistant. Then, the protocol* $(\mathsf{SignKeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *described above is a strong one-time signature scheme for signing messages $m \in \{0,1\}^*$ with perfect correctness.*

Proof. Correctness is straight-forward. To establish security, we first describe two simulators $\mathsf{Sim}_0, \mathsf{Sim}_1$ that given (u_0, s_1) and (s_0, u_1) respectively, simulates the verification key and the signature on a single message.

$\mathsf{Sim}_0(N, u_0, s_1)$: Pick $\tilde{w} \leftarrow_{\text{R}} \mathbb{QR}_N^+, e \leftarrow_{\text{R}} \mathbb{Z}_{2^\lambda}$. Output

$$\text{VKSIG} := (u_0, u_1, \tilde{w}^{2^\lambda} \cdot u_0^{-e})$$

When asked to sign a message $m \in \{0,1\}^*$, output

$$\left(e,\ \tilde{w} \cdot s_1^{H(m)+e \bmod 2^\lambda}\right)$$

$\mathsf{Sim}_1(N, s_0, u_1)$: Pick $\tilde{w} \leftarrow_{\text{R}} \mathbb{QR}_N^+, \tilde{e} \leftarrow_{\text{R}} \mathbb{Z}_{2^\lambda}$. Output

$$\text{VKSIG} := (u_0, u_1, \tilde{w}^{2^\lambda} \cdot u_1^{-\tilde{e}})$$

When asked to sign a message $m \in \{0,1\}^*$, output

$$\left(\tilde{e} - H(m) \bmod 2^\lambda,\ \tilde{w} \cdot s_0^{\tilde{e}-H(m) \bmod 2^\lambda}\right)$$

It is straight-forward to check that the outputs of both Sim_0 and Sim_1 are identically distributed to the output of a honestly generated VKSIG and an honestly generated signature on a single message. Now, we consider several cases for a forgery (e', w') on m':

- $m' = m$, same $e' = e$: then, $w' = w$.
- $e \neq e'$: in Sim_0, the forgery will allow us to compute the 2^λ'th root of $u_0^{e-e'}$ where $|e - e'| < 2^\lambda$, i.e.:

$$(u_0^{e-e'})^{2^{-\lambda}} = \frac{w}{w'} \cdot \frac{s_1^{\mathsf{H}(m')+e'}}{s_1^{\mathsf{H}(m)+e}}$$

  Using Shamir's GCD in the exponent algorithm, this value along with u_0 allows us to recover a square root of u_0.

- $e = e'$, $\mathsf{H}(m) \neq \mathsf{H}(m')$: in Sim_1, extract a square root of u_1, analogous to the previous case.
- $e = e'$, $\mathsf{H}(m) = \mathsf{H}(m')$, but $m' \neq m$: contradict collision resistance of H.

That is, we can show that if an adversary outputs a forgery with probability ϵ, then we can compute a square root of a random challenge u with probability roughly $\epsilon/2$ as follows: we pick $b \leftarrow_{\mathrm{R}} \{0,1\}$, run Sim_b with u as u_b and choosing a random s_{1-b}. □

$\mathsf{TDG}(\mathrm{PP}), \mathrm{PP} = (N, g)$:	$\mathsf{PSamp}(\mathrm{FID}, \mathrm{TAG}; r)$:	$\mathsf{TdInv}(\mathrm{TID}, \mathrm{TAG}, u \Vert \tau)$:
$\mathrm{TID} \leftarrow_{\mathrm{R}} [(N-1)/4]$	$(s, u) := (g^{2^\ell r}, g^{2^{\lambda+\ell} r})$	check $u, \tau \in \mathbb{QR}_N^+$
$\mathrm{FID} := g^{2^{\lambda+\ell} \cdot \mathrm{TID}}$	$\tau := (\mathrm{FID} \cdot g^{\mathrm{TAG}})^r$	check $\tau^{2^{\lambda+\ell}} = u^{\mathrm{TAG}+2^{\lambda+\ell} \cdot \mathrm{TID}}$
return $(\mathrm{FID}, \mathrm{TID})$	return $(s, u \Vert \tau)$	find $a, b, c \in \mathbb{Z}$: $2^c = a\mathrm{TAG} + b2^{\lambda+\ell}$
$\mathsf{G}(s) := \mathsf{BBS}(s)$		return $(\tau^a \cdot u^{b - a \cdot \mathrm{TID}})^{2^{\ell-c}}$

Fig. 3. An adaptive trapdoor relation based on factoring [42, 26]

4.2 Adaptive Trapdoor Relations

The class Φ^+. The functions $\phi_\Delta : [N/4] \to \mathbb{Z}$ in this class are indexed by $\Delta \in [-N/4, N/4]$, where $\phi_\Delta(\mathrm{TID}) := \mathrm{TID} + \Delta$.

Φ^+-key homomorphism. Observe that for all $\mathrm{TID}, \Delta \in \mathbb{Z}$, all TAG and all $u, \tau \in \mathbb{QR}_N^+$:

$$\mathsf{TdInv}(\mathrm{TID} + \Delta, \mathrm{TAG}, u \Vert \tau) = \mathsf{TdInv}(\mathrm{TID}, \mathrm{TAG}, u \Vert (\tau \cdot u^{-\Delta}))$$

The above equality follows from the fact that TdInv returns $s = u^{2^{-\lambda}}$ in both sides of the equation when the following condition holds

$$\tau^{2^{\lambda+\ell}} = u^{\mathrm{TAG}+2^{\lambda+\ell} \cdot (\mathrm{TID}+\Delta)} \iff (\tau \cdot u^{-\Delta})^{2^{\lambda+\ell}} = u^{\mathrm{TAG}+2^{\lambda+\ell} \cdot \mathrm{TID}}$$

and $\perp$ otherwise.

Φ^+-fingerprinting. We establish a stronger statement, namely Φ-fingerprinting for any class Φ of efficiently computable functions $\phi : [(N-1)/4] \to \{-N, \ldots, N\}$. Fix an adversary $\mathcal{A}$. Let $y = u \Vert \tau$ denote the challenge in the security experiment. Furthermore, suppose $\mathcal{A}$ outputs ϕ such that $\phi(\mathrm{TID}) \neq \mathrm{TID}$ and $\mathsf{TdInv}(\phi(\mathrm{TID}), \mathrm{TAG}^*, y) \neq \perp$. This means:

$$\tau^{2^{\lambda+\ell}} = u^{\mathrm{TAG}^*+2^{\lambda+\ell} \cdot \mathrm{TID}} = u^{\mathrm{TAG}^*+2^{\lambda+\ell} \cdot \phi(\mathrm{TID})}$$

and thus

$$u^{\mathrm{TID}} = u^{\phi(\mathrm{TID})}$$

With probability $1 - O(\sqrt{N})$, both g and u are generators of $\mathbb{QR}_N^+$. This means $\mathrm{TID} = \phi(\mathrm{TID}) \pmod{\phi(N)/4}$. This would allow us to factor N.

5 Instantiations from Diffie-Hellman Assumptions

5.1 Strong One-Time Signature from Hardness of Discrete Log

For completeness, we present here Groth's one-time signature scheme [24, Section 5.4]; we modified the underlying algebra in order to clarify the similarity to our factoring-based scheme. Here, we use a hash function $\mathsf{H} : \{0,1\}^* \to \mathbb{Z}_q$. The scheme is secure if computing discrete log is hard on average and H is collision resistant.

$\mathsf{SignKeyGen}(\mathrm{PP}), \mathrm{PP}$	$=\mathsf{Sign}(\mathrm{SKSIG}, m)$:	$\mathsf{Verify}(\mathrm{VKSIG}, m, (e, w))$
$(\mathbb{G}, q, g)$:	$e \leftarrow_{\mathrm{R}} \mathbb{Z}_q$	check $g^w = c \cdot u_0^e \cdot u_1^{\mathsf{H}(m)+e}$
$\mathrm{SKSIG} := (s_0, s_1, x) \leftarrow_{\mathrm{R}} \mathbb{Z}_q^3$	$w := x + e \cdot s_0 + (\mathsf{H}(m) + e) \cdot s_1$	
$(u_0, u_1, c) := (g^{s_0}, g^{s_1}, g^x)$	return $(e, w) \in \mathbb{Z}_q \times \mathbb{Z}_q$	
$\mathrm{VKSIG} := (u_0, u_1, c)$		
return $(\mathrm{VKSIG}, \mathrm{SKSIG})$		

Fig. 4. Discrete-log-based strong one-time signature [24]

5.2 Instantiations from BDDH

The class Φ^+. The functions $\phi_\Delta : Z_q \to \mathbb{Z}_q$ in this class are indexed by $\Delta \in \mathbb{Z}_q$, where $\phi_\Delta(\mathrm{TID}) := \mathrm{TID} + \Delta$.

Φ^+-key homomorphism. Observe that for all $\mathrm{TID}, \Delta \in \mathbb{Z}_q$, all TAG and all $u, \tau \in \mathbb{G}$:

$$\mathsf{TdInv}(\mathrm{TID} + \Delta, \mathrm{TAG}, u \| \tau) = \mathsf{TdInv}(\mathrm{TID}, \mathrm{TAG}, u \| (\tau \cdot u^{-\Delta}))$$

The above equality follows from the fact that on both sides of the equation, TdInv computes s such that

$$s^{\mathrm{TAG}} = \tau \cdot u^{-(\mathrm{TID}+\Delta)} = (\tau \cdot u^{-\Delta}) \cdot u^{-\mathrm{TID}}$$

Φ^+-fingerprinting. We establish a stronger statement, namely Φ-fingerprinting for any class Φ of functions $\phi : \mathbb{Z}_q \to \mathbb{Z}_q$. Fix an adversary $\mathcal{A}$. Let $y = u \| \tau$ denote the challenge in the security experiment. Furthermore, suppose $\mathcal{A}$ outputs ϕ such that $\mathsf{TdInv}(\phi(\mathrm{TID}), \mathrm{TAG}^*, y) \neq \perp$. This means:

$$(\tau \cdot u^{-\mathrm{TID}})^{\mathrm{TAG}^* - 1} = (\tau \cdot u^{-\phi(\mathrm{TID})})^{\mathrm{TAG}^* - 1}$$

$\mathsf{TDG}(\mathrm{PP}), \mathrm{PP}$	$=\mathsf{PSamp}(\mathrm{FID}, \mathrm{TAG}; r)$:	$\mathsf{TdInv}(\mathrm{TID}, \mathrm{TAG}, u \| \tau)$:
$(\mathbb{G}, q, g, g^\alpha, g^\gamma)$:	$(s, u) := ((g^\alpha)^r, g^r)$	compute $s := (\tau \cdot u^{-\mathrm{TID}})^{\mathrm{TAG}^{-1}}$
$\mathrm{TID} \leftarrow_{\mathrm{R}} \mathbb{Z}_q$; $\mathrm{FID} := g^{\mathrm{TID}}$	$\tau := (\mathrm{FID} \cdot (g^\alpha)^{\mathrm{TAG}})^r$	if $e(g, s) = e(g^\alpha, u)$:
return $(\mathrm{FID}, \mathrm{TID})$	return $(s, u \| \tau)$	return s, else $\perp$
$\mathsf{G}(s) := e(s, g^\gamma)$		

Fig. 5. An adaptive trapdoor relation based on BDDH [42, 9]

and thus

$$u^{\text{TID}} = u^{\phi(\text{TID})}$$

Hence, $\text{TID} = \phi(\text{TID})$.

$\mathsf{TDG}(\text{PP}), \text{PP} = (\mathbb{G}, q, g)$:	$\mathsf{PSamp}(\text{FID}, \text{TAG}; r)$:	$\mathsf{TdInv}(\text{TID}, \text{TAG}, u\|\tau_0\|\tau_1)$:
$\text{TID} := (\alpha, \beta, \gamma_0, \gamma_1) \leftarrow_{\text{R}} \mathbb{Z}_q^4$ $\text{FID} := (g^\alpha, g^\beta, g^{\gamma_0}, g^{\gamma_1})$ return (FID, TID) $\mathsf{G}(s) := s$	$(s, u) := ((g^\alpha)^r, g^r)$ $\tau_0 := (g^{\gamma_0} \cdot (g^\alpha)^{\text{TAG}})^r$ $\tau_1 := (g^{\gamma_1} \cdot (g^\beta)^{\text{TAG}})^r$ return $(s, u\|\tau_0\|\tau_1))$	compute $s_0 := (\tau_0 \cdot u^{-\gamma_0})^{\text{TAG}^{-1}}$ compute $s_1 := (\tau_1 \cdot u^{-\gamma_1})^{\text{TAG}^{-1}}$ if $s_0 = u^\alpha \wedge s_1 = u^\beta$: return s_0, else $\perp$

Fig. 6. An adaptive trapdoor relation based on DDH [42, 13]

5.3 Weakly CC-RKA-Secure Schemes from DDH

The class Φ^+. The functions $\phi_\Delta : \mathbb{Z}_q^4 \to \mathbb{Z}_q^4$ in this class are indexed by $\Delta \in \mathbb{Z}_q^4$, where $\phi_\Delta(\text{TID}) := \text{TID} + \Delta$.

Φ^+-key homomorphism. Observe that for all $\text{TID}, \Delta \in \mathbb{Z}_q^4$, all TAG and all $u, \tau_0, \tau_1 \in \mathbb{G}$:

$$\mathsf{TdInv}(\text{TID} + \Delta, \text{TAG}, u\|\tau_0\|\tau_1) = \mathsf{TdInv}(\text{TID}, \text{TAG}, u\|(\tau_0 \cdot u^{-\Delta})\|(\tau_1 \cdot u^{-\Delta}))$$

6 Instantiations from LWE

We rely on a construction from [35, 1]. Here, $\mathbf{G}$ is a public matrix with special structure for which the bounded-distance decoding problem is easy.

The class Φ^+. The functions $\phi_\Delta : \mathbb{Z}_q^{\overline{m} \times w} \to \mathbb{Z}_q^{\overline{m} \times w}$ in this class are indexed by $\Delta \in \mathbb{Z}_q^{\overline{m} \times w}$, where $\phi_\Delta(\text{TID}) := \text{TID} + \Delta$.

$\mathsf{TDG}(\text{PP}), \text{PP} = (\mathbf{G}, \overline{\mathbf{A}}) \in \mathbb{Z}_q^{n \times (w + \overline{m})}$:	$\mathsf{PSamp}(\text{FID}, \text{TAG})$:	$\mathsf{TdInv}(\text{TID}, \text{TAG}, \mathbf{u}\|\mathbf{v})$:
$\text{TID} := \mathbf{R} \leftarrow_{\text{R}} \mathcal{D}_q^{\overline{m} \times w}$ $\text{FID} := \mathbf{A}' := \overline{\mathbf{A}}\mathbf{R}$ return (FID, TID)	$\mathbf{u} := \overline{\mathbf{A}}^\top \mathbf{s} + \mathbf{e},\ \mathbf{s} \leftarrow_{\text{R}} \mathbb{Z}_q^n$ $\mathbf{A}_{\text{TAG}} := \mathbf{A}' + \text{TAG} \cdot \mathbf{G}$ $\mathbf{v} := \mathbf{A}_{\text{TAG}}^\top \mathbf{s} + \mathbf{e}'$ return $(\mathbf{s}, \mathbf{u}\|\mathbf{v})$	compute $\mathbf{v}' = \mathbf{v} - \mathbf{R}^\top \mathbf{u}$ solve $\mathbf{s}$ s.t. $\mathbf{v}' \approx \text{TAG} \cdot \mathbf{G}^\top \mathbf{s}$ if $\|\mathbf{v}' - \text{TAG} \cdot \mathbf{G}^\top \mathbf{s}\|, \|\mathbf{u} - \overline{\mathbf{A}}^\top \mathbf{s}\|$ are small: return $\mathbf{s}$ else $\perp$

Fig. 7. An adaptive trapdoor relation based on LWE [35]

Φ^+-key homomorphism. Observe that for all $\mathbf{R}, \Delta \in \mathbb{Z}_q^{\overline{m} \times w}$, all TAG and all $\mathbf{u} \| \mathbf{v} \in \mathbb{Z}_q^{\overline{m}+w}$:

$$\mathsf{TdInv}(\mathbf{R} + \Delta, \text{TAG}, \mathbf{u} \| \mathbf{v}) = \mathsf{TdInv}(\mathbf{R}, \text{TAG}, \mathbf{u} \| (\mathbf{v} - \Delta^\top \mathbf{u}))$$

The above equality just follows from the fact that on both sides of the equation, TdInv computes

$$\mathbf{v}' = \mathbf{v} - (\mathbf{R} + \Delta)^\top \mathbf{u} = (\mathbf{v} - \Delta^\top \mathbf{u}) - \mathbf{R}^\top \mathbf{u}$$

Acknowledgments. I would like to thank David Cash, Dennis Hofheinz, Payman Mohassel and Daniel Wichs for helpful discussions and the anonymous referees for detailed and helpful feedback.

References

[1] Agrawal, S., Boneh, D., Boyen, X.: Efficient Lattice (H)IBE in the Standard Model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010)

[2] Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 595–618. Springer, Heidelberg (2009)

[3] Applebaum, B., Ishai, Y., Kushilevitz, E.: Semantic security under related-key attacks and applications. In: ICS, pp. 45–55 (2011)

[4] Bellare, M., Cash, D.: Pseudorandom Functions and Permutations Provably Secure against Related-Key Attacks. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 666–684. Springer, Heidelberg (2010)

[5] Bellare, M., Kohno, T.: A Theoretical Treatment of Related-key Attacks: RKA-PRPs, RKA-PRFs, and Applications. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 491–506. Springer, Heidelberg (2003)

[6] Bellare, M., Shoup, S.: Two-Tier Signatures, Strongly Unforgeable Signatures, and Fiat-Shamir Without Random Oracles. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 201–216. Springer, Heidelberg (2007)

[7] Bellare, M., Cash, D., Miller, R.: Cryptography Secure against Related-Key Attacks and Tampering. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 486–503. Springer, Heidelberg (2011); Also Cryptology ePrint Archive, Report 2011/252

[8] Biham, E., Shamir, A.: Differential Fault Analysis of Secret Key Cryptosystems. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 513–525. Springer, Heidelberg (1997)

[9] Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)

[10] Boneh, D., DeMillo, R.A., Lipton, R.J.: On the Importance of Checking Cryptographic Protocols for Faults. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 37–51. Springer, Heidelberg (1997)

[11] Boyen, X., Mei, Q., Waters, B.: Direct chosen ciphertext security from identity-based techniques. In: ACM CCS, pp. 320–329 (2005)

[12] Canetti, R., Halevi, S., Katz, J.: Chosen-Ciphertext Security from Identity-Based Encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
[13] Cash, D., Kiltz, E., Shoup, V.: The Twin Diffie-Hellman problem and applications. J. Cryptology 22(4), 470–504 (2009)
[14] Choi, S.G., Kiayias, A., Malkin, T.: BiTR: Built-in Tamper Resilience. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 740–758. Springer, Heidelberg (2011)
[15] Cramer, R., Shoup, V.: A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
[16] Cramer, R., Shoup, V.: Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002); Also, Cryptology ePrint Archive, Report 2001/085
[17] Cramer, R., Damgård, I.B., Schoenmakers, B.: Proof of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 174–187. Springer, Heidelberg (1994)
[18] Cramer, R., Dodis, Y., Fehr, S., Padró, C., Wichs, D.: Detection of Algebraic Manipulation with Applications to Robust Secret Sharing and Fuzzy Extractors. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 471–488. Springer, Heidelberg (2008)
[19] Dolev, D., Dwork, C., Naor, M.: Nonmalleable cryptography. SIAM J. Comput. 30(2), 391–437 (2000)
[20] Dziembowski, S., Pietrzak, K., Wichs, D.: Non-malleable codes. In: ICS, pp. 434–452 (2010)
[21] Gennaro, R., Lysyanskaya, A., Malkin, T., Micali, S., Rabin, T.: Algorithmic Tamper-Proof (ATP) Security: Theoretical Foundations for Security against Hardware Tampering. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 258–277. Springer, Heidelberg (2004)
[22] Goldenberg, D., Liskov, M.: On Related-Secret Pseudorandomness. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 255–272. Springer, Heidelberg (2010)
[23] Goyal, V., O'Neill, A., Rao, V.: Correlated-Input Secure Hash Functions. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 182–200. Springer, Heidelberg (2011)
[24] Groth, J.: Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 444–459. Springer, Heidelberg (2006)
[25] Halderman, J.A., Schoen, S.D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J.A., Feldman, A.J., Appelbaum, J., Felten, E.W.: Lest we remember: cold-boot attacks on encryption keys. Commun. ACM 52(5), 91–98 (2009)
[26] Hofheinz, D., Kiltz, E.: Practical Chosen Ciphertext Secure Encryption from Factoring. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 313–332. Springer, Heidelberg (2009)
[27] Hofheinz, D., Kiltz, E.: The Group of Signed Quadratic Residues and Applications. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 637–653. Springer, Heidelberg (2009)
[28] Kalai, Y.T., Kanukurthi, B., Sahai, A.: Cryptography with Tamperable and Leaky Memory. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 373–390. Springer, Heidelberg (2011)

[29] Katz, J., Malka, L.: Constant-Round Private Function Evaluation with Linear Complexity. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 556–571. Springer, Heidelberg (2011); Also Cryptology ePrint Archive, Report 2010/528
[30] Kiltz, E.: Chosen-Ciphertext Security from Tag-Based Encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006)
[31] Kiltz, E., Mohassel, P., O'Neill, A.: Adaptive Trapdoor Functions and Chosen-Ciphertext Security. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 673–692. Springer, Heidelberg (2010)
[32] Kocher, P.C.: Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996)
[33] Lucks, S.: Ciphers Secure against Related-Key Attacks. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 359–370. Springer, Heidelberg (2004)
[34] MacKenzie, P.D., Reiter, M.K., Yang, K.: Alternatives to Non-malleability: Definitions, Constructions, and Applications. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 171–190. Springer, Heidelberg (2004)
[35] Micciancio, D., Peikert, C.: Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012); Also, Cryptology ePrint Archive, Report 2011/501
[36] Mohassel, P.: One-Time Signatures and Chameleon Hash Functions. In: Biryukov, A., Gong, G., Stinson, D.R. (eds.) SAC 2010. LNCS, vol. 6544, pp. 302–319. Springer, Heidelberg (2011)
[37] Pandey, O., Pass, R., Vaikuntanathan, V.: Adaptive One-Way Functions and Applications. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 57–74. Springer, Heidelberg (2008)
[38] Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: STOC, pp. 187–196 (2008)
[39] Rackoff, C., Simon, D.R.: Non-interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)
[40] Ristenpart, T., Tromer, E., Shacham, H., Savage, S.: Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In: ACM Conference on Computer and Communications Security, pp. 199–212 (2009)
[41] Shamir, A., Tauman, Y.: Improved Online/Offline Signature Schemes. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 355–367. Springer, Heidelberg (2001)
[42] Wee, H.: Efficient Chosen-Ciphertext Security via Extractable Hash Proofs. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 314–332. Springer, Heidelberg (2010)

A Related-key Attacks on Cramer-Shoup

We point out two simple linear RKAs on the Cramer-Shoup CCA-secure encryption scheme [15] based on DDH, since these attacks highlight some of the main technical difficulities in achieving RKA security. We stress that this does not undermine the Cramer-Shoup scheme in any way, since the scheme was not designed to resist RKAs. The scheme is as follows:

$\mathsf{Gen}(\mathrm{PP})$, $\mathrm{PP} = (\mathbb{G}, q, g_1, g_2)$:	$\mathsf{Enc}(\mathrm{PK}, m; r)$:	$\mathsf{Dec}(\mathrm{SK}, u\|v\|w\|e)$:
$\mathrm{SK} := (x, y, a, b, a', b') \leftarrow_{\mathrm{R}} \mathbb{Z}_q^6$	$(u, v, w) := (g_1^r, g_2^r, h^r \cdot m)$	$t := \mathsf{TCR}(u\|v\|w)$
$(h, c, d) := (g_1^x g_2^y, g_1^a g_2^b, g_1^{a'} g_2^{b'})$	$t := \mathsf{TCR}(u\|v\|w)$	if $u^{a+t\cdot a'} \cdot v^{b+t\cdot b'} = e$:
$\mathrm{PK} := (h, c, d)$	$e := (cd^t)^r$	return $w/(u^x v^y)$ else $\perp$
return $(\mathrm{PK}, \mathrm{SK})$	return $u\|v\|w\|e$	

The following attacks work for any $\Delta \in \mathbb{Z}_q$ and suppose we are given a valid encryption (u, v, w, e) of some unknown message m:

- if we change a in the secret key to $a+\Delta$, observe that $(u, v, w, e \cdot u^{\Delta})$ decrypts to m under the modified secret key.
- if we change x in the secret key to $x + \Delta$, observe that (u, v, w, e) decrypts to $m \cdot u^{-\Delta}$ under the modified secret key.

In both cases, we can easily recover the message m given the output of the decryption algorithm on the modified secret key.

B Strong One-Time Signatures from Σ Protocols

We sketch here a generic construction of one-time signatures for Σ protocols. We start with a Σ-protocol Π for any one-way relation. Applying the CDS-transform [17], we may derive another Σ-protocol that given a pair of instances (u_0, u_1), proves knowledge for one of the two witnesses. Now consider the following signature scheme: the verification key is (u_0, u_1, c_0, c_1) and a signature on a message M is a triplet (e, a_0, a_1) such that (c_0, e, a_0) and $(c_1, M \oplus e, a_1)$ are accepting transcripts for Π for the instances u_0 and u_1 respectively.

We show that this scheme is one-time unforgeable; moreover, if Π has unique responses, then the scheme is one-time strongly unforgeable. The proof of security is very simple: we generate (u_0, u_1) along with the witness for u_b, where $b \in \{0, 1\}$ is chosen at random. Using the witness, we can simulate the signature oracle for a single message. Given a forgery, we can extract a witness to one of u_0, u_1, which with probability $1/2$, is different from the one we already know.

Constructions of one-time signatures from Σ-protocols were also given in [36, 6]. However, the transformation given here as well as our factoring-based instantiation appear to be novel.

Functional Encryption for Threshold Functions (or Fuzzy IBE) from Lattices

Shweta Agrawal [1], Xavier Boyen [2], Vinod Vaikuntanathan [3], Panagiotis Voulgaris [4], and Hoeteck Wee [5]

[1] UCLA
[2] PARC
[3] University of Toronto
[4] Google Inc.
[5] George Washington University

Abstract. Cryptosystems based on the hardness of lattice problems have recently acquired much importance due to their average-case to worst-case equivalence, their conjectured resistance to quantum cryptanalysis, their ease of implementation and increasing practicality, and, lately, their promising potential as a platform for constructing advanced functionalities.

In this work, we construct "Fuzzy" Identity Based Encryption from the hardness of the Learning With Errors (LWE) problem. We note that for our parameters, the underlying lattice problems (such as gapSVP or SIVP) are assumed to be hard to approximate within supexponential factors for adversaries running in subexponential time. We give CPA and CCA secure variants of our construction, for small and large universes of attributes. All our constructions are secure against selective-identity attacks in the standard model. Our construction is made possible by observing certain special properties that secret sharing schemes need to satisfy in order to be useful for Fuzzy IBE. We also discuss some obstacles towards realizing lattice-based attribute-based encryption (ABE).

1 Introduction

Lattices have recently emerged as a powerful mathematical platform on which to build a rich variety of cryptographic primitives. Starting from the work of Ajtai [5], lattices have been used to construct one-way functions and collision-resistant hash functions [5,29], signatures [14], public-key encryption [7,35,36], identity-based encryption schemes [24,17,1,2], trapdoor functions [24] and even fully homomorphic encryption [22,23,16,15]. Lattice-based cryptography is attractive not only as a fallback in case factoring and discrete-log turn out to be easy (which they are on quantum computers), but it is also an end in its own right — lattice-based systems resist quantum and sub-exponential attacks, and they are efficient, admit highly parallel implementations and are potentially quite practical.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 280–297, 2012.

At the same time, encryption schemes have grown more and more sophisticated, and able to support complex access policies. Specifically, the idea of *functional encryption* has emerged as a new paradigm for encryption. In functional encryption in its broad sense, a secret key allows its holder to unlock data (or some piece or function of the data) based on policies and logic, rather than by merely addressing the recipient(s). The usefulness of such a primitive is evident — access to encrypted data moves beyond mere enumeration to potentially arbitrary functions.

Since its introduction with Fuzzy Identity-Based Encryption by Sahai and Waters [37], several systems have emerged that move beyond the traditional "designated recipient(s)" paradigm of encryption. In this line of work, the key (or, in some variants, the ciphertext) is associated with a predicate, say f, while the ciphertext (or the key) is associated with an attribute vector, say x. Decryption succeeds if and only if $f(x) = 1$. Specifically, *attribute-based encryption* [25,32,10,18,27,28] refers to the case where the predicate is a Boolean formula to which the attributes provide binary inputs. Fuzzy IBE is a special case where f is a k-out-of-ℓ threshold function. In *predicate encryption* [26,27], the predicate f is to be evaluated without leaking anything about the attributes other than the binary output of $f(x)$, i.e., achieving *attribute hiding* along with the standard *payload hiding*; known constructions are currently limited to inner-product predicates between embedded constants and attributes living in some field, though.

Notably, all known instantiations of Functional Encryption are based on bilinear maps on elliptic curves — and most are based on the IBE framework by Boneh and Boyen [11]. Non-pairing constructions have remained elusive, even though factoring-based IBE has been known since 2001 [19,13] and lattice-based IBE since 2008 [24]. This is even more notable in the lattice world, where we now have an array of sophisticated (hierarchical) IBE schemes [24,3,17,1,2], but the construction of more expressive functional encryption schemes has been lagging far behind.

Our Contributions. We take the first step in this direction by constructing a fuzzy identity-based encryption (fuzzy IBE) scheme based on lattices. A fuzzy IBE scheme is exactly like an identity-based encryption scheme except that (considering identities as bit-vectors in $\{0,1\}^n$) a ciphertext encrypted under an identity $\mathsf{id}_{\mathsf{enc}}$ can be decrypted using the secret key corresponding to any identity $\mathsf{id}_{\mathsf{dec}}$ that is "close enough" to $\mathsf{id}_{\mathsf{enc}}$. Examples arise when using one's biometric information as the identity, but also in general access control systems that permit access as long as the user satisfies a certain number of conditions.

Our construction is secure in the selective security model under the learning with errors (LWE) assumption and thus, by the results of [36,34], secure under the worst-case hardness of "short vector problems" on arbitrary lattices. We then extend our construction to handle large universes, and to resist chosen ciphertext (CCA) attacks. Finally, we point out some difficulties involved in extending our approach to functional encryption systems.

This work constitutes one of the first examples of lattice-based schemes that generalize the basic "(H)IBE" functionality.

Concurrent Work. A concurrent work of Agrawal, Freeman and Vaikuntanathan [4] gave a construction of inner product predicate encryption from lattices. Combined with a generic transformation given by Katz, Sahai and Waters [26, Section 5.5], this yields a lattice-based fuzzy IBE for "exact thresholds" where decryption succeeds whenever $\mathsf{id}_{\mathsf{dec}}$ and $\mathsf{id}_{\mathsf{enc}}$ differ in *exactly* k positions; we address the setting where the identities differ in *at most* k positions.

1.1 Overview of our Construction

Our construction borrows ideas from the pairing-based fuzzy IBE scheme of Sahai and Waters [37] and the lattice identity-based encryption scheme of [3,17], together with an interesting observation about the Shamir secret-sharing scheme and the Lagrange interpolation formula.

First, consider the setting where the identities are ℓ-bit strings. This corresponds to the setting where there are ℓ attributes, and each attribute can take two values (either 0 or 1). Decryption using $\mathsf{SK}_{\mathsf{id}}$ succeeds on a ciphertext encrypted under identity id' if the bitwise difference of id and id' has Hamming weight at most k. We then show how to extend it to the case where the universe of attributes is (exponentially) large in a rather generic way.

Previous Lattice-Based IBE. We begin by recalling the IBE schemes of [3,17], which we view as fuzzy IBE schemes where $k = \ell$. The public parameters consist of 2ℓ matrices $(\mathbf{A}_{1,0}, \mathbf{A}_{1,1}, \ldots, \mathbf{A}_{\ell,0}, \mathbf{A}_{\ell,1}) \in \mathbb{Z}_q^{n \times m}$ (where n is the security parameter, q is a small prime, and $m \approx n \log q$ is a parameter of the system) and a vector $\mathbf{u} \in \mathbb{Z}_q^n$. The master secret key then consists of the trapdoors $\mathbf{T}_{i,b}$ corresponding to each matrix $\mathbf{A}_{i,b}$.

We view the secret key derivation in the IBE scheme as a two-step procedure that proceeds as follows: on input an identity id:

1. First, *secret-share* the vector $\mathbf{u}$ into ℓ vectors $\mathbf{u}_1, \ldots, \mathbf{u}_\ell$ which are uniformly random in $\mathbb{Z}_q^n$ subject to the condition that $\sum_{i=1}^{\ell} \mathbf{u}_i = \mathbf{u}$.
2. The secret key $\mathsf{SK}_{\mathsf{id}}$ is then a vector $(\mathbf{e}_1, \ldots, \mathbf{e}_\ell) \in (\mathbb{Z}^m)^\ell$, where

$$\mathsf{SK}_{\mathsf{id}} \doteq (\mathbf{e}_1, \ldots, \mathbf{e}_\ell) \quad \text{and} \quad \mathbf{A}_{i,\mathsf{id}_i}\mathbf{e}_i = \mathbf{u}_i$$

   The secret key $\mathbf{e}_i$ is computed using the trapdoor $\mathbf{T}_{i,\mathsf{id}_i}$ using the Gaussian sampling algorithm of [24].

This is a different, yet completely equivalent, way to view the secret key derivation in the IBE schemes of [3,17].

To encrypt for an identity id in these schemes, one chooses a vector $\mathbf{s} \in \mathbb{Z}_q^n$ and "small error terms" $\mathbf{x}_1, \ldots, \mathbf{x}_\ell \in \mathbb{Z}^m$ and $x' \in \mathbb{Z}$, and outputs

$$\mathsf{CT}_{\mathsf{id}} \doteq \mathsf{IBE.Enc}(\mathsf{id}, b \in \{0,1\}) \doteq (\mathbf{A}_{1,\mathsf{id}_1}^T\mathbf{s} + \mathbf{x}_1, \ldots, \mathbf{A}_{\ell,\mathsf{id}_\ell}^T\mathbf{s} + \mathbf{x}_\ell, \mathbf{u}^T\mathbf{s} + x' + b\lfloor q/2 \rfloor)$$

The key observation in decryption is that if $\mathsf{id} = \mathsf{id}'$, then "pairing" each component of $\mathsf{CT}_{\mathsf{id}'}$ and $\mathsf{SK}_{\mathsf{id}}$ gives us a number that is approximately $\mathbf{u}_i^T\mathbf{s}$. Namely,

$$\mathbf{e}_i^T(\mathbf{A}_{i,\mathsf{id}_i}^T\mathbf{s} + \mathbf{x}_i) = (\mathbf{A}_{i,\mathsf{id}_i}\mathbf{e}_i)^T\mathbf{s} + \mathbf{e}_i^T\mathbf{x}_i = \mathbf{u}_i^T\mathbf{s} + \mathbf{e}_i^T\mathbf{x}_i \approx \mathbf{u}_i^T\mathbf{s} \qquad (1)$$

By linearity, we can then add up these terms and obtain (approximately) $\mathbf{u}^T\mathbf{s}$. The "approximation" we get here is not terrible, since the error terms $\mathbf{e}_i^T\mathbf{x}_i$ are small, and we add up only ℓ of them. Thus, the magnitude of the error remains much smaller than $q/2$, which is sufficient for decryption.

Our Approach. A natural thought to extend this methodology to fuzzy IBE is to use Shamir's k-out-of-ℓ secret-sharing scheme in the first step of the key derivation procedure. Since reconstructing the secret in Shamir's scheme involves computing a linear combination of the shares, we can hope to do decryption as before. As it turns out, the resulting scheme is in fact *neither correct nor secure.* For simplicity, we focus on the issue of correctness in this section.

Recall that correctness of the previous lattice-based IBE schemes lies in bounding the decryption "error terms" $\mathbf{e}_i^T\mathbf{x}_i$. More concretely, the analysis bounds the "cummulative error term"

$$x' - \sum_{i=1}^{k} \mathbf{e}_i^T\mathbf{x}_i$$

by $q/4$. Upon instantiating the previous schemes with Shamir's secret-sharing scheme, we need to bound a new cummulative error term, which is given by:

$$x' - \sum_{i \in S} L_i \mathbf{e}_i^T\mathbf{x}_i$$

Here, L_i are the fractional Lagrangian coefficients used in reconstructing the secret, interpreted as elements in $\mathbb{Z}_q$ and S identifies the subset of shares used in reconstruction. Indeed, while we can bound both the numerator and denominator in L_i as a fraction of integers, once interpreted as an element in $\mathbb{Z}_q$, the value L_i may be arbitrarily large.

The key idea in our construction is to "clear the denominators". Let $D := (\ell!)^2$ be a sufficiently large constant, so that $DL_i \in \mathbb{Z}$ for all i. Then, we multiply D into the noise vector, that is, the ciphertext is now generated as follows:

$$\mathsf{CT}_{\mathsf{id}} \doteq \mathsf{IBE.Enc}(\mathsf{id}, b \in \{0,1\}) \doteq (\mathbf{A}_{1,\mathsf{id}_1}^T\mathbf{s} + D\mathbf{x}_1, \ldots, \mathbf{A}_{\ell,\mathsf{id}_\ell}^T\mathbf{s} + D\mathbf{x}_\ell, \mathbf{u}^T\mathbf{s} + Dx' + b\lfloor q/2 \rfloor)$$

For correctness, it now suffices to bound the expression:

$$Dx - \sum_{i \in S} DL_i \mathbf{e}_i^T\mathbf{x}_i$$

by $q/4$. Now, further observe that each DL_i is an integer bounded by D^2, so it suffices to pick the noise vectors so that they are bounded by $q/4D\ell$ with overwhelming probability.

Thus, for appropriate parameter settings, we get a fuzzy IBE scheme based on the classical hardness of computing a sub-exponential approximation to "short vector problems" on arbitrary lattices.

Additional Related Work. The idea of using Shamir's secret-sharing scheme in lattice-based cryptography appears in the work of Bendlin and Damgård [9] on threshold cryptosystems. The security of their scheme, as with ours, relies on the hardness of computing sub-exponential approximation for lattice problems. In more detail, their scheme uses a pseudorandom secret-sharing from [20] in order to share a value in some interval, for which they do not have to address the issue of bounding the size of Lagrangian coefficients. Our idea of "clearing the denominator" is inspired by the work on factoring-based threshold cryptography (e.g. [39]), where the technique is used to handle a different technical issue: evaluating fractional Lagrangian coefficients over an "unknown" modulus $\phi(N)$, where N is a public RSA modulus.

2 Preliminaries

Notation: We use uppercase boldface alphabet for matrices, as in $\mathbf{A}$, lowercase boldface characters for vectors, as in $\mathbf{e}$, and lowercase regular characters for scalars, as in v. We say that a function $f : \mathbb{R}^+ \to \mathbb{R}^+$ is negligible if for all $d > d_0$ we have $f(\lambda) < 1/\lambda^d$ for sufficiently large λ. We write $f(\lambda) < \mathrm{negl}(\lambda)$. For any ordered set $S = \{\mathbf{s}_1, \ldots, \mathbf{s}_k\} \in \mathbb{R}^m$ of linearly independent vectors, we define $\|\widetilde{\mathbf{S}}\| = \max_j \|\tilde{\mathbf{s}_j}\|$, where $\widetilde{\mathbf{S}} = \{\tilde{\mathbf{s}_1}, \ldots, \tilde{\mathbf{s}_k}\}$ refers to the Gram-Schmidt orthogonalization of $\mathbf{S}$, and $\|\cdot\|$ refers to the euclidean norm. We let $\sigma_{\mathrm{TG}} := O(\sqrt{n \log q})$ denote the maximum (w.h.p.) Gram-Schmidt norm of a basis produced by $\mathsf{TrapGen}(q, n)$.

2.1 Definition: Fuzzy IBE

A Fuzzy Identity Based Encryption scheme consists of the following four algorithms:

Fuzzy.Setup$(\lambda, \ell) \to (\mathsf{PP}, \mathsf{MK})$**:** This algorithm takes as input the security parameter λ and the maximum length of identities ℓ. It outputs the public parameters PP and a master key MK.

Fuzzy.Extract$(\mathsf{MK}, \mathsf{PP}, \mathsf{id}, k) \to \mathsf{SK}_{\mathsf{id}}$**:** This algorithm takes as input the master key MK, the public parameters PP, an identity id and the threshold $k \leq \ell$. It outputs a decryption key $\mathsf{SK}_{\mathsf{id}}$.

Fuzzy.Enc$(\mathsf{PP}, b, \mathsf{id}') \to \mathsf{CT}_{\mathsf{id}'}$**:** This algorithm takes as input: a message bit b, an identity id', and the public parameters PP. It outputs the ciphertext $\mathsf{CT}_{\mathsf{id}'}$.

Fuzzy.Dec$(\mathsf{PP}, \mathsf{CT}_{\mathsf{id}'}, \mathsf{SK}_{\mathsf{id}}) \to b$**:** This algorithm takes as input the ciphertext $\mathsf{CT}_{\mathsf{id}'}$, the decryption key $\mathsf{SK}_{\mathsf{id}}$ and the public parameters PP. It outputs the message b if $|\mathsf{id} \cap \mathsf{id}'| \geq k$.

2.2 Security Model for Fuzzy IBE

We follow the Selective-ID model of security for Fuzzy Identity Based Encryption as given by Sahai and Waters [37, Section 2.1]. The security game is very

similar to the standard Selective-ID model for Identity-Based Encryption with the exception that the adversary is only allowed to query for secret keys for identities which have less than k overlap with the target identity id^*.

Target: The adversary declares the challenge identity, id^*, that he wishes to be challenged upon.
Setup: The challenger runs the Setup algorithm of Fuzzy-IBE and gives the public parameters to the adversary.
Phase 1: The adversary is allowed to issue queries for private keys for identities id_j of its choice, as long as $|\mathsf{id}_j \cap \mathsf{id}^*| < k; \forall j$
Challenge: The adversary submits a message to encrypt. The challenger encrypts the message with the challenge identity id^* and then flips a random coin r. If $r = 1$, the ciphertext is given to the adversary, otherwise a random element of the ciphertext space is returned.
Phase 2: Phase 1 is repeated.
Guess: The adversary outputs a guess r' of r. The advantage of an adversary A in this game is defined as $|\Pr[r' = r] - \frac{1}{2}|$

A Fuzzy Identity Based Encryption scheme is secure in the Selective-Set model of security if all polynomial time adversaries have at most a negligible advantage in the Selective-Set game.

The adaptive version of the above game is identical except it does not have the target step, hence the adversary is allowed to choose an attack identity adversarially.

3 Preliminaries: Lattices

Throughout the paper, we let the parameters $q = q(\lambda), m = m(\lambda), n = n(\lambda)$ are polynomial functions of the security parameter λ.

3.1 Random Integer Lattices

Definition 1. *Let* $\mathbf{B} = \left[\mathbf{b}_1 \,\middle|\, \ldots \,\middle|\, \mathbf{b}_m \right] \in \mathbb{R}^{m \times m}$ *be an* $m \times m$ *matrix whose columns are linearly independent vectors* $\mathbf{b}_1, \ldots, \mathbf{b}_m \in \mathbb{R}^m$*. The* m*-dimensional full-rank lattice* Λ *generated by* $\mathbf{B}$ *is the infinite periodic set,*

$$\Lambda = \mathcal{L}(\mathbf{B}) = \left\{ \mathbf{y} \in \mathbb{R}^m \quad s.t. \quad \exists \mathbf{s} = (s_1, \ldots, s_m) \in \mathbb{Z}^m \ , \quad \mathbf{y} = \mathbf{B}\,\mathbf{s} = \sum_{i=1}^{m} s_i\, \mathbf{b}_i \right\}$$

Here, we are interested in integer lattices, i.e, infinite periodic subsets of $\mathbb{Z}^m$, that are invariant under translation by multiples of some integer q in each of the coordinates.

Definition 2. *For* q *prime and* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *and* $\mathbf{u} \in \mathbb{Z}_q^n$*, define:*

$$\Lambda_q^{\perp}(\mathbf{A}) = \left\{ \mathbf{e} \in \mathbb{Z}^m \quad s.t. \quad \mathbf{A}\,\mathbf{e} = 0 \pmod q \right\}$$
$$\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \left\{ \mathbf{e} \in \mathbb{Z}^m \quad s.t. \quad \mathbf{A}\,\mathbf{e} = \mathbf{u} \pmod q \right\}$$

3.2 Trapdoors for Lattices: The Algorithm TrapGen

Ajtai [6] showed how to sample an essentially uniform matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ with an associated full-rank set $\mathbf{T_A} \subset \Lambda^\perp(\mathbf{A})$ of *low-norm vectors*. We will use an improved version of Ajtai's basis sampling algorithm due to Alwen and Peikert [8]:

Proposition 1 ([8]).
Let $n = n(\lambda), q = q(\lambda), m = m(\lambda)$ be positive integers with $q \geq 2$ and $m \geq 5n \log q$. There exists a probabilistic polynomial-time algorithm TrapGen *that outputs a pair $\mathbf{A} \in \mathbb{Z}_q^{n \times m}, \mathbf{T_A} \in \mathbb{Z}_q^{m \times m}$ such that $\mathbf{A}$ is statistically close to uniform and $\mathbf{T_A}$ is a basis for $\Lambda^\perp(\mathbf{A})$ with length $L = \|\widetilde{\mathbf{T_A}}\| \leq m \cdot \omega(\sqrt{\log m})$ with all but $n^{-\omega(1)}$ probability.*

3.3 Discrete Gaussians

Definition 3. *Let $m \in \mathbb{Z}_{>0}$ be a positive integer and $\Lambda \subset \mathbb{R}^m$ an m-dimensional lattice. For any vector $\mathbf{c} \in \mathbb{R}^m$ and any positive parameter $\sigma \in \mathbb{R}_{>0}$, we define:*

- $\rho_{\sigma,c}(\mathbf{x}) = \exp\left(-\pi \frac{\|\mathbf{x}-\mathbf{c}\|^2}{\sigma^2}\right)$ *: a Gaussian-shaped function on $\mathbb{R}^m$ with center $\mathbf{c}$ and parameter σ,*
- $\rho_{\sigma,\mathbf{c}}(\Lambda) = \sum_{\mathbf{x} \in \Lambda} \rho_{\sigma,\mathbf{c}}(\mathbf{x})$ *: the (always converging) discrete integral of $\rho_{\sigma,\mathbf{c}}$ over the lattice Λ,*
- $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}}$ *: the discrete Gaussian distribution over Λ with center $\mathbf{c}$ and parameter σ,*

$$\forall \mathbf{y} \in \Lambda \quad , \quad \mathcal{D}_{\Lambda,\sigma,\mathbf{c}}(\mathbf{y}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{y})}{\rho_{\sigma,\mathbf{c}}(\Lambda)}$$

For notational convenience, $\rho_{\sigma,0}$ and $\mathcal{D}_{\Lambda,\sigma,0}$ are abbreviated as ρ_σ and $\mathcal{D}_{\Lambda,\sigma}$.

Sampling Discrete Gaussians over Lattices. Gentry, Peikert and Vaikuntanathan [24] construct the following algorithm for sampling from the discrete Gaussian $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}}$, given a basis $\mathbf{B}$ for the m-dimensional lattice Λ with $\sigma \geq \|\widetilde{\mathbf{B}}\| \cdot \omega(\sqrt{\log m})$:

SampleGaussian$(\Lambda, \mathbf{B}, \sigma, \mathbf{c})$ **[24]:** On input lattice Λ, a basis $\mathbf{B}$ for Λ, a positive Gaussian parameter σ, and a center vector $\mathbf{c} \in \mathbb{R}^m$, it outputs a fresh random vector $\mathbf{x} \in \Lambda$ drawn from a distribution statistically close to $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}}$.

3.4 Preimage Sampling

We will need the following algorithm from [24]. Let $q \geq 2$, $m \geq 2n \log q$.

Algorithm SamplePre$(\mathbf{A}, \mathbf{T_A}, \mathbf{u}, \sigma)$*:* On input a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ with 'short' trapdoor basis $\mathbf{T_A}$ for $\Lambda_q^\perp(\mathbf{A})$, a target image $\mathbf{u} \in \mathbb{Z}_q^n$ and a Gaussian parameter $\sigma \geq \|\widetilde{\mathbf{T_A}}\| \cdot \omega(\sqrt{\log m})$, outputs a sample $\mathbf{e} \in \mathbb{Z}^m$ from a distribution that is within negligible statistical distance of $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\sigma}$.

3.5 Sampling from an "Encryption" Matrix

We will also need the following algorithm defined in [17,1]:

Algorithm $\mathsf{SampleLeft}(\mathbf{A}, \mathbf{M}_1, \mathbf{T_A}, \mathbf{u}, \sigma)$*:*

Inputs:

$$\begin{array}{l} \text{a rank } n \text{ matrix } \mathbf{A} \text{ in } \mathbb{Z}_q^{n\times m} \text{ and a matrix } \mathbf{M}_1 \text{ in } \mathbb{Z}_q^{n\times m_1}, \\ \text{a "short" basis } \mathbf{T_A} \text{ of } \Lambda_q^{\perp}(\mathbf{A}) \text{ and a vector } \mathbf{u} \in \mathbb{Z}_q^n, \\ \text{a gaussian parameter } \sigma > \|\widetilde{\mathbf{T_A}}\| \cdot \omega(\sqrt{\log(m+m_1)}). \end{array} \tag{2}$$

Output: Let $\mathbf{F}_1 := (\mathbf{A} \mid \mathbf{M}_1)$. The algorithm outputs a vector $\mathbf{e} \in \mathbb{Z}^{m+m_1}$ sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{F}_1),\sigma}$. In particular, $\mathbf{e} \in \Lambda_q^{\mathbf{u}}(\mathbf{F}_1)$.

3.6 Hardness Assumption

The LWE (learning with errors) problem was first defined by [36], and has since been extensively studied and used. We use the decisional version of the LWE problem.

Definition 4. *Consider a prime q, a positive integer n, and a distribution χ over $\mathbb{Z}_q$, all public. An $(\mathbb{Z}_q, n, \chi)$-LWE problem instance consists of access to an unspecified challenge oracle $\mathcal{O}$, being, either, a noisy pseudo-random sampler $\mathcal{O}_s$ carrying some constant random secret key $\mathbf{s} \in \mathbb{Z}_q^n$, or, a truly random sampler $\mathcal{O}_\$$, whose behaviors are respectively as follows:*

- $\mathcal{O}_s$**:** *outputs noisy pseudo-random samples of the form $(\mathbf{w}_i, v_i) = \left(\mathbf{w}_i,\ \mathbf{w}_i^T\,\mathbf{s} + x_i\right) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where, $\mathbf{s} \in \mathbb{Z}_q^n$ is a uniformly distributed persistent secret key that is invariant across invocations, $x_i \in \mathbb{Z}_q$ is a freshly generated ephemeral additive noise component with distribution χ, and $\mathbf{w}_i \in \mathbb{Z}_q^n$ is a fresh uniformly distributed vector revealed as part of the output.*
- $\mathcal{O}_\$$**:** *outputs truly random samples $\left(\mathbf{w}_i, v_i\right) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, drawn independently uniformly at random in the entire domain $\mathbb{Z}_q^n \times \mathbb{Z}_q$.*

The $(\mathbb{Z}_q, n, \chi)$-LWE problem statement, or LWE for short, allows an unspecified number of queries to be made to the challenge oracle $\mathcal{O}$, with no stated prior bound. We say that an algorithm $\mathcal{A}$ decides the $(\mathbb{Z}_q, n, \chi)$-LWE problem if $\left|\Pr[\mathcal{A}^{\mathcal{O}_s} = 1] - \Pr[\mathcal{A}^{\mathcal{O}_\$} = 1]\right|$ is non-negligible for a random $s \in \mathbb{Z}_q^n$.

It has been shown in [36] that there is a $poly(n, q)$-time reduction from Search LWE$(\mathbb{Z}_q, n, \chi)$ to Decision LWE$(\mathbb{Z}_q, n, \chi)$.

The confidence in the hardness of the LWE problem stems in part from a result of Regev [36] which shows that the for certain noise distributions χ, the LWE problem is as hard as the worst-case SIVP and GapSVP under a quantum reduction (see also [33]). A classical reduction with related parameters was later obtained by Peikert [34].

Proposition 2 ([36]). *Consider a real parameter $\alpha = \alpha(n) \in (0,1)$ and a prime $q = q(n) > 2\sqrt{n}/\alpha$. Denote by $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ the group of reals $[0,1)$ with addition modulo 1. Denote by Ψ_α the distribution over $\mathbb{T}$ of a normal variable with mean 0 and standard deviation $\alpha/\sqrt{2\,\pi}$ then reduced modulo 1. Denote by $\lfloor x \rceil = \lfloor x + \frac{1}{2} \rfloor$ the nearest integer to the real $x \in \mathbb{R}$. Denote by $\bar{\Psi}_\alpha$ the discrete distribution over $\mathbb{Z}_q$ of the random variable $\lfloor q\,X \rceil \mod q$ where the random variable $X \in \mathbb{T}$ has distribution Ψ_α.*

Then, if there exists an efficient, possibly quantum, algorithm for deciding the $(\mathbb{Z}_q, n, \bar{\Psi}_\alpha)$-LWE problem, there exists a quantum $q \cdot \mathsf{poly}(n)$-time algorithm for approximating the SIVP and GapSVP problems, to within $\tilde{O}(n/\alpha)$ factors in the ℓ_2 norm, in the worst case.

Since the best known algorithms for 2^k-approximations of gapSVP and SIVP run in time $2^{\tilde{O}(n/k)}$) [21,38,31], it follows from the above that the LWE problem with the noise ratio $\alpha = 2^{-n^\epsilon}$ is likely hard for some constant $\epsilon < 1$.

Two Lemmas to Bound Norms. The following lemma about the distribution $\overline{\Psi}_\alpha$ will be needed to show that decryption works correctly. The proof is implicit in [24, Lemma 8.2].

Lemma 1. *Let $\mathbf{e}$ be some vector in $\mathbb{Z}^m$ and let $\mathbf{y} \xleftarrow{R} \overline{\Psi}_\alpha^m$, where $\overline{\Psi}_\alpha$ is as defined in Proposition 2. Then the quantity $|\mathbf{e}^\top \mathbf{y}|$ treated as an integer in $[0, q-1]$ satisfies*

$$|\mathbf{e}^\top \mathbf{y}| \leq \|\mathbf{e}\|\, q\alpha\, \omega(\sqrt{\log m}\,) + \|\mathbf{e}\|\sqrt{m}/2$$

with all but negligible probability in m.

Micciancio and Regev showed that the norm of vectors sampled from discrete Gaussians is small with high probability.

Lemma 2 ([30]). *For any lattice Λ of integer dimension m, any lattice point $\mathbf{c}$, and any two reals $\epsilon \in (0,1)$ and $\sigma \geq \omega(\sqrt{\log m})$,*

$$\Pr\left\{ \mathbf{x} \sim \mathcal{D}_{\Lambda,\sigma,\mathbf{c}} \;:\; \|\mathbf{x} - \mathbf{c}\| > \sqrt{m}\,\sigma \right\} \;\leq\; \frac{1+\epsilon}{1-\epsilon}\, 2^{-m}$$

4 The Fuzzy IBE Scheme

We refer the reader to Section 1.1 for an overview of our construction, and proceed directly to the details. Let $\lambda \in \mathbb{Z}^+$ be a security parameter. Let $q = q(\lambda)$ be a prime, $n = n(\lambda)$ and $m = m(\lambda)$ two positive integers, and $\sigma = \sigma(\lambda)$ and $\alpha = \alpha(\lambda)$ two positive Gaussian parameters. We assume that $\mathsf{id} \in \{0,1\}^\ell$ for some $\ell \in \mathbb{N}$.

4.1 Construction

Fuzzy.Setup$(1^\lambda, 1^\ell)$**:** On input a security parameter λ, and identity size ℓ, do:

1. Use algorithm $\mathsf{TrapGen}(1^\lambda)$ (from Proposition 1) to select 2ℓ uniformly random $n \times m$-matrices $\mathbf{A}_{i,b} \in \mathbb{Z}_q^{n \times m}$ (for all $i \in [\ell], b \in \{0,1\}$) together with a full-rank set of vectors $\mathbf{T}_{i,b} \subseteq \Lambda_q^\perp(\mathbf{A}_{i,b})$ such that $\|\widetilde{\mathbf{T}_{i,b}}\| \leq m \cdot \omega(\sqrt{\log m})$.
2. Select a uniformly random vector $\mathbf{u} \in \mathbb{Z}_q^n$.
3. Output the public parameters and master key,

$$\mathsf{PP} = \left(\{\mathbf{A}_{i,b}\}_{i\in[\ell],b\in\{0,1\}}, \mathbf{u} \right) \quad ; \quad \mathsf{MK} = \left(\{\mathbf{T}_{i,b}\}_{i\in[\ell],b\in\{0,1\}} \right)$$

Fuzzy.Extract$(\mathsf{PP}, \mathsf{MK}, \mathsf{id}, k)$**:** On input public parameters PP, a master key MK, an identity $\mathsf{id} \in \{0,1\}^\ell$ and threshold $k \leq \ell$, do:

1. Construct ℓ shares of $\mathbf{u} = (u_1, ..., u_n) \in \mathbb{Z}_q^n$ using a Shamir secret-sharing scheme applied to each co-ordinate of $\mathbf{u}$ independently. Namely, for each $j \in [n]$, choose a uniformly random polynomial $p_j \in \mathbb{Z}_q[x]$ of degree $k-1$ such that $p_j(0) = u_j$.
   Construct the j^{th} share vector

$$\hat{\mathbf{u}}_j = (\hat{u}_{j,1}, \ldots, \hat{u}_{j,n}) \stackrel{\text{def}}{=} (p_1(j), p_2(j), \ldots, p_n(j)) \in \mathbb{Z}_q^n$$

   Looking ahead (to decryption), note that for all $J \subset [\ell]$ such that $|J| \geq k$, we can compute fractional Lagrangian coefficients L_j such that $\mathbf{u} = \sum_{j\in J} L_j \cdot \hat{\mathbf{u}}_j \pmod q$. That is, we interpret L_j as a fraction of integers, which we can also evaluate $\pmod q$.
2. Using trapdoor MK and the algorithm $\mathsf{SamplePre}$ from Section 3.3, find $\mathbf{e}_j \in \mathbb{Z}^m$ such that $\mathbf{A}_{j,\mathsf{id}_j} \cdot \mathbf{e}_j = \hat{\mathbf{u}}_j$, for $j \in [\ell]$.
3. Output the secret key for id as $(\mathsf{id}, \{\mathbf{e}_1, \ldots, \mathbf{e}_\ell\})$.

Fuzzy.Enc$(\mathsf{PP}, \mathsf{id}, b)$**:** On input public parameters PP, an identity id, and a message $b \in \{0,1\}$, do:

1. Let $D \stackrel{\text{def}}{=} (\ell!)^2$.
2. Choose a uniformly random $\mathbf{s} \stackrel{R}{\leftarrow} \mathbb{Z}_q^n$.
3. Choose a noise term $x \leftarrow \chi_{\{\alpha,q\}}$ and $\mathbf{x}_i \leftarrow {\chi_{\{\alpha,q\}}}^m$,
4. Set $c_0 \leftarrow \mathbf{u}^\top \mathbf{s} + Dx + b\lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q$.
5. Set $\mathbf{c}_i \leftarrow {\mathbf{A}_{i,\mathsf{id}_i}}^\top \mathbf{s} + D\mathbf{x}_i \in \mathbb{Z}_q^m$ for all $i \in [\ell]$.
6. Output the ciphertext $\mathsf{CT}_{\mathsf{id}} := (c_0, \{\mathbf{c}_i\}_{i\in[\ell]}, \mathsf{id})$.

Fuzzy.Dec$(\mathsf{PP}, \mathsf{SK}_{\mathsf{id}}, \mathsf{CT}_{\mathsf{id}'})$**:** On input parameters PP, a private key $\mathsf{SK}_{\mathsf{id}}$, and a ciphertext $\mathsf{CT}_{\mathsf{id}'}$:

1. Let $J \subset [\ell]$ denote the set of matching bits in id and id'. If $|J| < k$, output $\perp$. Otherwise, we can compute fractional Lagrangian coefficients L_j so that

$$\sum_{j\in J} L_j \mathbf{A}_j \mathbf{e}_j = \mathbf{u} \pmod q$$

2. Compute $r \leftarrow c_0 - \sum_{j\in J} L_j \cdot \mathbf{e}_j^\top \mathbf{c}_j \pmod q$. View it as the integer $r \in [-\lfloor \frac{q}{2} \rfloor, \lfloor \frac{q}{2} \rfloor) \subset \mathbb{Z}$.
3. If $|r| < \frac{q}{4}$, output 0, else output 1.

Correctness. To establish correctness for decryption, we only need to consider the case $|J| \geq k$. Let L_j be the fractional Lagrangian coefficients as described above. Then,

$$\begin{aligned} r &= c_0 - \sum_{j \in J} L_j \, \mathbf{e}_j^\top \, \mathbf{c}_j \pmod q \qquad (3) \\ &= \mathbf{u}^\top \mathbf{s} + Dx + b \left\lfloor \frac{q}{2} \right\rfloor - \sum_{j \in J} L_j \, \mathbf{e}_j^\top (\mathbf{A}_j{}^\top \mathbf{s} + D \cdot \mathbf{x}_j) \pmod q \\ &= b \left\lfloor \frac{q}{2} \right\rfloor + \underbrace{\left(\mathbf{u}^\top \mathbf{s} - \sum_{j \in J} (L_j \, \mathbf{A}_j \mathbf{e}_j)^\top \mathbf{s} \right)}_{= 0 \pmod q} + \underbrace{\left(Dx - \sum_{j \in J} D L_j \mathbf{e}_j^\top \mathbf{x}_j \right)}_{\approx 0} \pmod q \quad \approx b \left\lfloor \frac{q}{2} \right\rfloor \end{aligned}$$

It suffices to set the parameters so that with overwhelming probability,

$$|Dx - \sum_{j \in J} D L_j \mathbf{e}_j^\top \mathbf{x}_j| \leq D|x| + \sum_{j \in J} D^2 |\mathbf{e}_j^\top \mathbf{x}_j| < q/4 \qquad (4)$$

For the first inequality, we use the following lemma on Lagrangian coefficients which states that the numbers DL_j are integers bounded above by $D^2 \leq (\ell!)^4$.

Lemma 3. *Let $D = (\ell!)^2$. Given $k \leq \ell$ numbers $I_1, \ldots, I_k \in [1 \ldots \ell]$, define the Lagrangian coefficients*

$$L_j = \prod_{i \neq j} \frac{-I_i}{(I_j - I_i)}$$

Then, for every $1 \leq j \leq k$, DL_j is an integer, and $|DL_j| \leq D^2 \leq (\ell!)^4$.

Proof. To see this, note that the denominator of the j^{th} Lagrange coefficient L_j is of the form

$$d_j = \prod_{i \neq j} (I_j - I_i)$$

The numbers $|I_j - I_i|$ lie in the interval $[-(\ell - 1), \ldots, (\ell - 1)]$, and they can repeat at most twice (namely, for every number $n \in [\ell]$, there are at most two i, i' such that $|I_j - I_i| = |I_j - I_{i'}|$).

Since each of the factors $I_j - I_i$ can appear at most twice in absolute value, $(\ell!)^2$ divides d_j. Thus, DL_j is an integer. Also,

$$|DL_j| \leq D \cdot |\prod_{j \neq i} (-I_i)| \leq (\ell!)^3$$

4.2 Proof of Security

We show that the Fuzzy IBE construction provides ciphertext privacy under a selective identity attack as in Definition 2.2. Recall that ciphertext privacy means that the challenge ciphertext is indistinguishable from a random element in the ciphertext space. More precisely, we have the following theorem:

Theorem 1. *If there exists a PPT adversary $\mathcal{A}$ with advantage $\epsilon > 0$ against the selective security game for the Fuzzy IBE scheme of Section 4.1, then there exists a PPT algorithm $\mathcal{B}$ that decides the LWE problem with advantage $\epsilon/(\ell+1)$.*

Proof. Recall from Definition 4 that an LWE problem instance is provided as a sampling oracle $\mathcal{O}$ which can be either truly random $\mathcal{O}_\$$ or noisy pseudo-random $\mathcal{O}_s$ for some secret key $s \in \mathbb{Z}_q^n$. The simulator $\mathcal{B}$ uses the adversary $\mathcal{A}$ to distinguish between the two, and proceeds as follows:

Instance. $\mathcal{B}$ requests from $\mathcal{O}$ and receives $(\ell m+1)$ LWE samples that we denote as:

$$\begin{aligned} (\mathbf{w}_1, v_1) &\in \mathbb{Z}_q^n \times \mathbb{Z}_q \\ \{(\mathbf{w}_1^1, v_1^1), (\mathbf{w}_1^2, v_1^2), \ldots, (\mathbf{w}_1^m, v_1^m)\} &\in \left(\mathbb{Z}_q^n \times \mathbb{Z}_q\right)^m \\ &\ldots \quad \ldots \\ \{(\mathbf{w}_\ell^1, v_\ell^1), (\mathbf{w}_\ell^2, v_\ell^2), \ldots, (\mathbf{w}_\ell^m, v_\ell^m)\} &\in \left(\mathbb{Z}_q^n \times \mathbb{Z}_q\right)^m \end{aligned}$$

Targeting. $\mathcal{A}$ announces to $\mathcal{B}$ the identity it intends to attack, namely id^*.

Setup. $\mathcal{B}$ constructs the system's public parameters PP as follows:

1. The ℓ matrices $\mathbf{A}_{i,\mathsf{id}_i^*}$, $i \in [\ell]$ are chosen from the LWE challenge $\{(\mathbf{w}_i^1), (\mathbf{w}_i^2), \ldots, (\mathbf{w}_i^m)\}_{i\in[\ell]}$. The ℓ matrices $\mathbf{A}_{i,\overline{\mathsf{id}_i^*}}$, $i \in [\ell]$ are chosen using $\mathsf{TrapGen}$ with a trapdoor $\mathbf{T}_{i,\overline{\mathsf{id}_i^*}}$.
2. The vector $\mathbf{u}$ is constructed from the LWE challenge, $\mathbf{u} = \mathbf{w}_1$.

The public parameters are returned to the adversary.

Queries. $\mathcal{B}$ answers each private-key extraction query for identity id as follows:

1. Let $\mathsf{id} \cap \mathsf{id}^* := I \subset [\ell]$ and let $|I| = t < k$. Then, note that $\mathcal{B}$ has trapdoors for the matrices corresponding to the set $\bar{I}$, where $|\bar{I}| = \ell - t$. W.l.o.g., we assume that the first t bits of id are equal to id^*.
2. Represent the shares of $\mathbf{u}$ symbolically as $\hat{\mathbf{u}}_i = \mathbf{u} + \mathbf{a}_1 i + \mathbf{a}_2 i^2 + \ldots + \mathbf{a}_{k-1} i^{k-1}$ where $\mathbf{a}_1, \ldots, \mathbf{a}_{k-1}$ are vector variables of length n each.
3. For i s.t. $\mathsf{id}_i^* = \mathsf{id}_i$, pick $\mathbf{e}_i$ randomly using algorithm $\mathsf{SampleGaussian}$. Set $\hat{\mathbf{u}}_i := \mathbf{A}_{i,\mathsf{id}_i}\mathbf{e}_i$; $i \in [t]$.
4. Since $t \leq k-1$, and there are $k-1$ variables $\mathbf{a}_1 \ldots\ldots \mathbf{a}_{k-1}$, by choosing $k-1-t$ shares $\hat{\mathbf{u}}_{t+1}, \ldots, \hat{\mathbf{u}}_{k-1}$ randomly, the values for $\mathbf{a}_1 \ldots\ldots \mathbf{a}_{k-1}$ are determined. This determines all ℓ shares $\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_\ell$.
5. To find $\mathbf{e}_j$ s.t. $\mathbf{A}_{j,\mathsf{id}_j}\mathbf{e}_j = \hat{\mathbf{u}}_j$ for $j = t+1, \ldots \ell$, invoke
$$\mathsf{SamplePre}(\mathbf{A}_{j,\mathsf{id}_j}, \mathbf{T}_{j,\mathsf{id}_j}, \hat{\mathbf{u}}_j, \sigma)$$
6. Return $(\mathbf{e}_1, \ldots, \mathbf{e}_\ell)$.

Note that the distribution of the public parameters and keys in the real scheme is statistically indistinguishable from that in the simulation.

Challenge. $\mathcal{A}$ outputs a message bit $b^* \in \{0,1\}$. $\mathcal{B}$ responds with a challenge ciphertext for id^*:

1. Let $c_0 = Dv_1 + b\lfloor q/2 \rfloor$.
2. Let $\mathbf{c}_i = (Dv_i^1, Dv_i^2 \ldots\ldots Dv_i^m)$ for $i \in [\ell]$.

Guess. The adversary $\mathcal{A}$ outputs a guess b'. The simulator $\mathcal{B}$ uses that guess to determine an answer on the LWE oracle: Output "`genuine`" if $b' = b^*$, else output "`random`".

4.3 Parameters

We set the parameters to ensure that the decoding works with high probability, and that the security reductions are meaningful. Our security parameter is λ, and given (an upper bound on) ℓ, the size of the universe, the rest of the parameters are set under the following constraints:

1. For the lattice trapdoor generation algorithm of Alwen and Peikert [8], we need $m \geq 5n \log q$.
   Given this constraint on m, the $\mathsf{TrapGen}$ algorithm outputs a basis of (Gram-Schmidt) length at most $m \cdot \sqrt{\log m}$. Using the $\mathsf{SamplePre}$ algorithm, the secret key vectors $\mathbf{e}_j$ are drawn from a discrete Gaussian with standard deviation $\sigma \geq m \cdot \log m$ (using the $\mathsf{SamplePre}$ algorithm), and thus, by Proposition 2, have length at most $\sigma\sqrt{m} \leq m^{1.5} \cdot \log m$ with all but exponentially small probability.
2. We set the noise distribution $\chi = \overline{\Psi}_\alpha^m$, where $\alpha \geq 2\sqrt{m}/q$ in order to apply Regev's reduction (see Lemma 2). A vector $\mathbf{x}$ sampled from this distribution has length $O(\alpha q\sqrt{m}) \leq 2m$ with all but exponentially small probability.
3. For the correctness to hold, we need to satisfy equation 4. Since $D = (\ell!)^2$, and letting $\alpha = 1\sqrt{m}/q$, we have

$$D|x| + \sum_{j \in J} D^2 |\mathbf{e}_j^\top \mathbf{x}_j| \leq D \cdot \alpha q\sqrt{m} + \ell \cdot D^2 \cdot (\alpha q\sqrt{m} \cdot m^{1.5} \log m \cdot \sqrt{m})$$
$$\leq 4 \cdot m^3 \log m \cdot \ell(\ell!)^4 \leq m^3 \log m \cdot 2^{5\ell}$$

   where we used the fact that $(\ell!)^4 \leq (\ell)^{4\ell} \leq 2^{5\ell}$. Setting $q \geq m^3 \log m \cdot 2^{5\ell}$ ensures correctness.

As for concrete parameters settings under these constraints, we set:

- The lattice dimension $n = \lambda$ and $\ell = n^\epsilon$ for some constant $\epsilon \in (0, 1)$.
- The modulus q to be a prime in the interval $[n^6 2^{5\ell}, 2 \cdot n^6 2^{5\ell}]$.
- $m = n^{1.5} \geq 5n \log q$, satisfying (1) above.

Putting together the last two bullets, we see that $q \geq m^3 \log m \cdot 2^{5\ell}$, satisfying (3) above.

- The noise parameter $\alpha = 2\sqrt{m}/q = 1/(2^{5n^\epsilon} \cdot \mathsf{poly}(n))$.

Combining this with the worst-case to average-case connection (Proposition 2), we get security under the hardness of $2^{O(n^\epsilon)}$-approximating gapSVP or SIVP on n-dimensional lattices using algorithms that run in time $q \cdot \mathsf{poly}(n) = 2^{O(n^\epsilon)}$. With our state of knowledge on lattice algorithms and algorithms for LWE, security holds for $\epsilon < 1/2$.

We describe a construction for identities that live in a large universe in Appendix A and connections to attribute based encryption in Appendix B.

5 Conclusion

We constructed a Fuzzy Identity-Based Encryption scheme, selectively secure in the standard model, from the hardness of the Learning With Errors problem. Ours is among the first realization of attribute-based encryption from lattices, and among the first and only "*post-quantum*, *beyond-IBE*" cryptosystems known to date. Extending the system by showing full security, improving the parameters of the underlying LWE assumption, or transforming it to support more expressive attributes, are important open problems.

Acknowledgments. The first author wishes to thank a DARPA/ONR PROCEED award, and NSF grants 1118096, 1065276, 0916574 and 0830803 for research support. The second author gratefully acknowledges support from European Union FP7 project grant HiPerLatCryp at the University of Liège, where part of this work was done. The third author gratefully acknowledges support from an NSERC Discovery Grant and from DARPA under Agreement number FA8750-11-2-0225. The last author's work was partly done while at Queens College CUNY. He was supported by NSF CAREER Award CNS-0953626, and the US Army Research laboratory and the UK Ministry of Defense under agreement number W911NF-06-3-0001. The authors would like to warmly thank Microsoft Research Redmond for its hospitality during various stages of this research. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defense, or the UK Government.

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient Lattice (H)IBE in the Standard Model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010)
2. Agrawal, S., Boneh, D., Boyen, X.: Lattice Basis Delegation in Fixed Dimension and Shorter-Ciphertext Hierarchical IBE. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 98–115. Springer, Heidelberg (2010)
3. Agrawal, S., Boyen, X.: Identity-based encryption from lattices in the standard model (July 2009) (manuscript), http://www.cs.stanford.edu/~xb/ab09/
4. Agrawal, S., Freeman, D.M., Vaikuntanathan, V.: Functional Encryption for Inner Product Predicates from Learning with Errors. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 21–40. Springer, Heidelberg (2011)
5. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: STOC 1996: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing, pp. 99–108. ACM, New York (1996)
6. Ajtai, M.: Generating Hard Instances of the Short Basis Problem. In: Wiedermann, J., Van Emde Boas, P., Nielsen, M. (eds.) ICALP 1999. LNCS, vol. 1644, pp. 1–9. Springer, Heidelberg (1999)

7. Ajtai, M., Dwork, C.: A public-key cryptosystem with worst-case/average-case equivalence. In: STOC, pp. 284–293 (1997)
8. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. In: STACS, pp. 75–86 (2009)
9. Bendlin, R., Damgård, I.: Threshold Decryption and Zero-Knowledge Proofs for Lattice-Based Cryptosystems. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 201–218. Springer, Heidelberg (2010)
10. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: SP 2007: Proceedings of the 2007 IEEE Symposium on Security and Privacy, pp. 321–334. IEEE Computer Society, Washington, DC (2007)
11. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
12. Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM Journal on Computing 36, 1301–1328 (2007)
13. Boneh, D., Gentry, C., Hamburg, M.: Space-efficient identity based encryption without pairings. In: Proceedings of FOCS 2007, pp. 647–657 (2007)
14. Boyen, X.: Lattice Mixing and Vanishing Trapdoors: A Framework for Fully Secure Short Signatures and More. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 499–517. Springer, Heidelberg (2010)
15. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE (2011) (in submission)
16. Brakerski, Z., Vaikuntanathan, V.: Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 505–524. Springer, Heidelberg (2011)
17. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai Trees, or How to Delegate a Lattice Basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010), http://eprint.iacr.org/
18. Cheung, L., Newport, C.C.: Provably secure ciphertext policy ABE. In: ACM Conference on Computer and Communications Security, pp. 456–465 (2007)
19. Cocks, C.: An Identity Based Encryption Scheme Based on Quadratic Residues. In: Honary, B. (ed.) Cryptography and Coding 2001. LNCS, vol. 2260, pp. 360–363. Springer, Heidelberg (2001)
20. Cramer, R., Damgård, I., Ishai, Y.: Share Conversion, Pseudorandom Secret-Sharing and Applications to Secure Computation. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 342–362. Springer, Heidelberg (2005)
21. Gama, N., Nguyen, P.Q.: Finding short lattice vectors within Mordell's inequality. In: STOC 2008 – Proc. 40th ACM Symposium on the Theory of Computing. ACM (2008)
22. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC, pp. 169–178 (2009)
23. Gentry, C.: Toward Basing Fully Homomorphic Encryption on Worst-Case Hardness. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 116–137. Springer, Heidelberg (2010)
24. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., Dwork, C. (eds.) STOC, pp. 197–206. ACM (2008)
25. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: CCS 2006: Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 89–98. ACM, New York (2006)

26. Katz, J., Sahai, A., Waters, B.: Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008)
27. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010)
28. Lewko, A., Waters, B.: Unbounded HIBE and Attribute-Based Encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 547–567. Springer, Heidelberg (2011)
29. Micciancio, D.: Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. In: FOCS, pp. 356–365 (2002)
30. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. In: FOCS 2004: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, pp. 372–381. IEEE Computer Society, Washington, DC (2004)
31. Micciancio, D., Voulgaris, P.: A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. In: Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, pp. 351–358. ACM, New York (2010)
32. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: CCS 2007: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 195–203. ACM, New York (2007)
33. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem. In: STOC 2009 (2009)
34. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: STOC 2009: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, pp. 333–342. ACM (2009)
35. Regev, O.: New lattice-based cryptographic constructions. J. ACM 51(6), 899–942 (2004)
36. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC 2005: Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing, pp. 84–93. ACM, New York (2005)
37. Sahai, A., Waters, B.: Fuzzy Identity-Based Encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
38. Schnorr, C.-P.: A hierarchy of polynomial time lattice basis reduction algorithms. Theor. Comput. Sci. 53, 201–224 (1987)
39. Shoup, V.: Practical Threshold Signatures. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 207–220. Springer, Heidelberg (2000)

A Extensions

CCA security. Both our small-universe and the large-universe schemes can be lifted from CPA to CCA security using standard methods [12]. Here we describe the extension for our small universe construction; details for the large universe construction follow directly.

Specifically, we make use of a one-time strongly unforgeable signature scheme S_0 to augment the underlying FuzzyIBE scheme. The $\mathsf{Fuzzy.Setup}$ and $\mathsf{Fuzzy.Extract}$ algorithms remain unchanged.

During $\mathsf{Fuzzy.Enc}$, the encryptor runs $\mathsf{S}_0.\mathsf{KeyGen}$ to obtain a public-secret key pair, which we denote by $(\mathsf{VK}, \mathsf{SK})$. We assume that VK is represented as a binary string. Then, the encryptor picks the identity id he wants to encrypt to, and sets $\mathsf{id}' = (\mathsf{id}|\mathsf{VK})$. Let $\mathsf{CT}_{\mathsf{id}'} \leftarrow \mathsf{Fuzzy.Enc}(\mathsf{PP}, b, \mathsf{id}')$. Next, the encryptor sets $\sigma \leftarrow \mathsf{S}_0.\mathsf{Sign}(\mathsf{CT}_{\mathsf{id}'}, \mathsf{SK})$ and returns the tuple $(\sigma, \mathsf{VK}, \mathsf{CT}_{\mathsf{id}'})$.

During $\mathsf{Fuzzy.Dec}$, the decryptor first checks that $\mathsf{S}_0.\mathsf{Verify}(\mathsf{CT}_{\mathsf{id}'}, \sigma, \mathsf{VK}) = \top$, and rejects if not. Next, she uses her secret key $\mathsf{SK}_{\mathsf{id}_1}$ to derive a secret key $\mathsf{SK}_{\mathsf{id}''}$ for the "delegated" identity $\mathsf{id}'' \leftarrow (\mathsf{id}_1|\mathsf{VK})$. Such delegation can be done using the standard technique from [17]. Note that if the Hamming weight $|\mathsf{id} - \mathsf{id}_1| \leq k$, then $|\mathsf{id}' - \mathsf{id}''| \leq k$, and conversely. Hence, if the decryptor is authorized to decrypt in the underlying scheme, she can use her extended key $\mathsf{SK}_{\mathsf{id}''}$ to decrypt in the augmented scheme, and only then. The details are deferred to the full paper.

Construction for Identities in a Large Universe. The construction outlined above can only support identities that are binary vectors of length ℓ. We desire to have the identities live in a larger space so that they capture more expressive attributes.

At a high level, we shall combine our small-universe Fuzzy IBE with a compatible standard-model IBE, such as [3,17,1], to construct a Fuzzy IBE that can support large-universe identities. In the scheme outlined here, we use the efficient IBE from Agrawal, Boneh, and Boyen [1] to provide large-universe entities. Our identities are now ℓ-vectors of attributes in $\mathbb{Z}_q^n$, while our parameters are linear in ℓ (ℓ depends on n however; see Section 4.3). We defer the detailed construction to the full version.

B Connections to Attribute Based Encryption

A natural question that arises from this work is whether the construction can be generalized to Attribute-Based Encryption (ABE) for more expressive access structures. Specifically, we could ask that the secret key for a user be associated with a set of her attributes (e.g., "PhD Student at University X", "Ran in Boston marathon") represented by some vector $\boldsymbol{x}$, and the ciphertext be created with respect to an access policy, represented by a (polynomial-size) Boolean circuit C, so that decryption works if and only if $C(\boldsymbol{x}) = 1$. (Conversely, we could instead bind the policy C to a user and the attributes $\boldsymbol{x}$ to a ciphertext.) In the world of bilinear maps, many constructions are known [25,32,10,18,27,28], the most general being for access policies that can be described using *Boolean formulas*.

The difficulty of generalizing our construction to handle arbitrary Boolean formulas is quite subtle. To see this, recall that Fuzzy IBE is a particular type of ABE where the policy is restricted to a single k-out-of-n threshold gate. Since any monotone Boolean formula has an associated linear secret sharing scheme (LSSS), we might imagine generalizing the Fuzzy IBE construction as follows:

1. During ABE.Setup, sample ℓ matrices $\mathbf{A}_1, \ldots, \mathbf{A}_\ell$ with trapdoors.
2. During ABE.Extract, given a formula f, represent it as a LSSS matrix $\mathbf{M}$, share $\mathbf{u}$ according to $\mathbf{M}$ to obtain $\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_\ell$ (instead of using Shamir secret sharing). Compute $\mathbf{e}_i, i \in [\ell]$ such that $\mathbf{A}_i \mathbf{e}_i = \hat{\mathbf{u}}_i \bmod q$ and release $\mathbf{e}_1, \ldots \mathbf{e}_\ell$.
3. During ABE.Enc: Say γ is a binary vector representing attributes. Then let $\mathbf{c}_i = \mathbf{A}_i^\top \mathbf{s} + \mathbf{x}$ for i *s.t.* $\gamma_i = 1$. Let $c_0 = \mathbf{u}^\top \mathbf{s} + y + b\lceil \frac{q}{2} \rceil$ as before ($\mathbf{x}, y$ is Gaussian noise and b is the bit being encrypted).
4. During ABE.Dec, if attributes γ satisfy f, we can find low norm coefficients ρ_i so that $\rho_i \hat{\mathbf{u}}_i = \mathbf{u}$ and decrypt by computing $c_0 - \sum_i \rho_i \mathbf{e}_i^\top \mathbf{c}_i$ as before.

The problem with this scheme is that the shares $\hat{\mathbf{u}}_i, \hat{\mathbf{u}}_j$ may be correlated; for, e.g. it is possible to get $\mathbf{u}_1 = \mathbf{u}_2$ for queries such as $(x_1 \vee x_2) \wedge x_3$ and $(x_1 \vee x_2) \wedge x_5$, etc. Then, their preimages $\mathbf{e}_1$ and $\mathbf{e}_2$ can be combined to form a short vector in the null-space of $[\mathbf{A}_1 | \mathbf{A}_2]$. Over several such queries, the attacker can then construct a full basis for $\Lambda^\perp([\mathbf{A}_1 | \mathbf{A}_2])$, that can be used to break the challenge ciphertext for a target attribute vector such as $1100 \ldots 00$.

This problem does not arise in our Fuzzy IBE approach since we enforce the policy using secret sharing based on Reed Solomon (RS) codes. RS codes have the property that given k shares, either the shares are sufficient to reconstruct the vector $\mathbf{u}$, or they look jointly uniformly random. This property is crucial in the Fuzzy IBE simulation, and is not satisfied by the ABE generalization outlined above. Thus, we suspect that new techniques will be required to construct Attribute-Based Encryption from lattices.

Variants of Waters' Dual System Primitives Using Asymmetric Pairings
(Extended Abstract)

Somindu C. Ramanna[1], Sanjit Chatterjee[2], and Palash Sarkar[1]

[1] Applied Statistics Unit
Indian Statistical Institute
203, B.T. Road, Kolkata
India 700108
{somindu_r,palash}@isical.ac.in
[2] Department of Computer Science and Automation
Indian Institute of Science
Bangalore, India 560012
sanjit@csa.iisc.ernet.in

Abstract. Waters, in 2009, introduced an important technique, called dual system encryption, to construct identity-based encryption (IBE) and related schemes. The resulting IBE scheme was described in the setting of symmetric pairing. A key feature of the construction is the presence of random tags in the ciphertext and decryption key. Later work by Lewko and Waters removed the tags and proceeding through composite-order pairings led to a more efficient dual system IBE scheme using asymmetric pairings whose security is based on non-standard but static assumptions. In this work, we have systematically simplified Waters 2009 IBE scheme in the setting of asymmetric pairing. The simplifications retain tags used in the original description. This leads to several variants, the first one of which is based on standard assumptions and in comparison to Waters' original scheme reduces ciphertexts and keys by two elements each. Going through several stages of simplifications, we finally obtain a simple scheme whose security can be based on two standard assumptions and a natural and minimal extension of the decision Diffie-Hellman problem for asymmetric pairing groups. The scheme itself is also minimal in the sense that apart from the tags, both encryption and key generation use exactly one randomiser each. This final scheme is more efficient than both the previous dual system IBE scheme in the asymmetric setting due to Lewko and Waters and the more recent dual system IBE scheme due to Lewko. We extend the IBE scheme to hierarchical IBE (HIBE) and broadcast encryption (BE) schemes. Both primitives are secure in their respective full models and have better efficiencies compared to previously known schemes offering the same level and type of security.

Keywords: identity-based encryption, dual system encryption, asymmetric pairing.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 298–315, 2012.

1 Introduction

Constructions of identity-based encryption schemes constitute one of the most challenging problems of public-key cryptography. The notion of IBE was proposed in [14] and solved in [3,7]. This lead to a great deal of research on the topic. The solution in [3], though simple and elegant, had several features which were not satisfactory from a theoretical point of view.

In this work, we will be interested in IBE schemes built from bilinear pairings. Till date, most pairing based cryptographic schemes have been based on a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, where $\mathbb{G}$ is a prime-order group of elliptic curve points over a finite field and $\mathbb{G}_T$ is a subgroup of a finite field. Such maps arise from Weil and Tate pairings and there is an extensive literature on efficient implementation of such maps. Since the two components of the domain of e are same, such an e is called a symmetric pairing. Another kind of pairings, where the order of $\mathbb{G}$ is composite has been proposed [4]. Such pairings are called composite-order pairings and provide additional flexibility in designing schemes. The trade-off, however, is that computing the pairing itself becomes significantly slower and also the representation of the group elements becomes substantially longer.

Symmetric pairings (over prime order groups), are neither the most general nor the most efficient of possible pairings over elliptic curves. A general bilinear map is of the form $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, where $\mathbb{G}_1$ is a prime-order group of points of an elliptic curve over a finite field $\mathbb{F}$ and $\mathbb{G}_2$ is a group (of the same prime-order) of points of the same curve over an extension of $\mathbb{F}$. Such maps are called asymmetric pairings. Studies [16,8,5] have indicated that compared to symmetric pairings, asymmetric pairings are much faster and more compact to implement.

An important work on pairing based IBE is [17] which builds upon earlier work in [1,2] to provide an efficient IBE scheme with full security without random oracles. Variants have been reported [6,12] which result in IBE schemes which are efficient and have practical sized parameters. Though important, a drawback of the scheme in [17] is that the size of the public parameters grows linearly with the security parameter.

In a major innovation, Waters [18] introduced a new technique – called dual system encryption – for construction of IBE schemes and related primitives. The scheme presented in [18] has the feature that the size of the public parameters is constant while retaining full security. Dual system encryption is by itself an interesting notion and worthy of further investigation. The goal of a better understanding of dual system encryption would be to obtain IBE schemes with improved efficiency compared to the one proposed in [18].

An immediate follow-up work [11] took the route of composite-order pairings. Such pairing groups have 'more structure' which can possibly help in getting a clearer understanding of the technique. (Waters remarks in [18] that his scheme was first obtained for composite order groups.) The approach taken by [11] is to look at a realization of the IBE scheme of [1] in the setting of composite order groups so as to obtain adaptive-id security. They also gave a conversion of their composite-order IBE scheme to an IBE scheme using prime-order asymmetric

pairing. In a very recent work [10], the framework of dual system encryption has been thoroughly investigated and an IBE scheme using prime-order pairing has been presented. We note that the conversion from composite-order to prime-order pairings in [11] and considering prime-order groups in [10] are motivated by efficiency considerations.

Waters IBE scheme in [18] is based on symmetric pairings. The security of the scheme is based on the hardness of the decision linear (DLin) and the decision bilinear Diffie-Hellman (DBDH) assumptions. It is of interest to convert this to asymmetric pairings. For one thing, this will enable faster and smaller implementations which will arise from the advantages of asymmetric pairings over their symmetric variants. There is, however, another reason. Use of asymmetric pairings brings forward the possibility of reducing the number of group elements in ciphertexts and keys. In fact, Waters [18] himself mentions: "using the SXDH assumption we might hope to shave off three group elements from both ciphertexts and private keys". The rationale for this comment is that for asymmetric pairings with no known efficiently computable isomorphisms between the groups $\mathbb{G}_1$ and $\mathbb{G}_2$, the decision Diffie-Hellman (DDH) assumption holds for both $\mathbb{G}_1$ and $\mathbb{G}_2$. This is the symmetric external Diffie-Hellman (SXDH) assumption. For symmetric pairings the DDH assumption does not hold in $\mathbb{G}$. Using the SXDH assumption will potentially lead to a simpler scheme requiring a lesser number of group elements.

Following up on the above mentioned remark by Waters, we have systematically investigated the various possibilities for using asymmetric pairings. To start the study, we performed a straightforward conversion to the setting of asymmetric pairings. The scheme in [18] is quite complex. Several scalars are used in the public parameters, encryption and key generation. These have definite and inter-connected roles in the security proof. Our first task was to pin down the relationships between these scalars and separate them out. This enabled us to work with one group of scalars with minimal changes to other groups.

With a good understanding of the roles of the scalars, we are able to apply simplifications in a stage-wise manner. The first simplification gives an IBE scheme (Scheme 1) which shrinks ciphertexts and keys by two elements each and whose security can be based on DDH1 (DDH assumption in $\mathbb{G}_1$), DLin and DBDH assumptions. We argue that the DDH2 assumption cannot be directly used. So, the afore-mentioned suggestion by Waters cannot be fulfilled. On the other hand, we show that using a natural and minimal extension of the DDH2 assumption, a significantly more efficient scheme (Scheme 6) can be obtained.

Waters original scheme [18] used random tags in the ciphertext and the decryption key. Simplification of this scheme by both Lewko-Waters [11] and Lewko [10] yielded IBE schemes which did not use such tags. In contrast, all our simplifications retain the tags used in the original description [18]. Even then, we are able to obtain significant simplifications and efficiency improvements. This suggests that for the purpose of simplification as an IBE it is not important to do away with the tags. Removing them has other positive consequences such as obtaining a constant size ciphertext hierarchical IBE [11].

Scheme 6 has the interesting feature that, apart from the tags, exactly one randomiser each is used for encryption and key generation which is minimal in case of ciphertext. However, it is not known whether the key generation could be made deterministic within the dual system framework. To show that our simplification retains the flexibility of the original technique by Waters, we obtain an analogue of the HIBE scheme and prove it secure in the full security model. This HIBE scheme inherits all the security properties from [18], but, provides improved efficiency. From this HIBE scheme we construct an adaptively secure BE scheme which is more efficient than all the previously known BE schemes with adaptive security. We provide only the construction of the BE scheme here; the full version of this paper [13] contains the security proof. The construction and proof for the HIBE scheme will appear in the full version [13].

A comparison of the features of various IBE schemes based on the dual system technique is shown in Tables 1 and 2. The columns $\#\mathcal{PP}$, $\#\mathcal{MSK}$, #cpr, #key provide the number of group elements in the public parameters, the master secret key, ciphertexts and decryption keys. The public parameters and ciphertexts consist of elements of $\mathbb{G}_1$ while the master secret key and decryption keys consist of elements of $\mathbb{G}_2$. Encryption efficiency counts the number of scalar multiplications in $\mathbb{G}_1$ while decryption efficiency counts the number of pairings that are required. Key generation (a less frequent activity) efficiency is given by the number of scalar multiplications in $\mathbb{G}_2$. Currently, Scheme 6 is the most efficient among all the known dual system IBE schemes.

Table 1. Comparison of dual system IBE schemes secure under standard assumptions. Waters-09 and Lewko-11 use symmetric pairings while Scheme 6 uses asymmetric pairings.

scheme	$\#\mathcal{PP}$	$\#\mathcal{MSK}$	#cpr	#key	enc eff	dec eff	key gen	assump
Waters-09 [18]	13	5	9	8	14	9	12	DLin, DBDH
Lewko-11 [10]	24	30	6	6	24	6	6	DLin
Scheme 1	9	8	7	6	10	6	9	DDH1, DLin, DBDH

Table 2. Comparison of dual system IBE schemes secure under non-standard but static assumptions. Both the schemes use asymmetric pairings. DDH1 is a weaker assumption than LW1 and DDH2v is a weaker assumption than LW2.

scheme	$\#\mathcal{PP}$	$\#\mathcal{MSK}$	#cpr	#key	enc eff	dec eff	key gen	assump
LW [11]	9	6	6	6	9	6	10	LW1, LW2, DBDH
Scheme 6	6	7	4	4	7	3	6	DDH1, DDH2v, DBDH

The figures in the table indicate that Scheme 6 is more efficient than Lewko-Waters scheme. In particular, decryption in Scheme 6 is about twice as fast as that of LW scheme (note that both constructions are based on Type-3 pairings). Since Scheme 1 and Scheme 6 use Type-3 pairings which offer much better performance compared to symmetric pairings, the gain in speed over Waters'

scheme and Lewko's scheme cannot be quantified just in terms of the number of operations performed. It also depends on the performance gain of asymmetric pairings over their symmetric variants for the chosen security level.

2 Prerequisites

We follow standard definitions and corresponding full security models of IBE, HIBE and BE schemes. Here we briefly describe asymmetric pairings and related assumptions. For more details on these the reader is referred to [16,8,5].

2.1 Bilinear Maps

Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be cyclic groups of prime order p. $\mathbb{G}_1$ and $\mathbb{G}_2$ are written additively while $\mathbb{G}_T$ is written multiplicatively. A cryptographic bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ has the following properties.

1. **Bilinearity:** For elements $A_1, B_1 \in \mathbb{G}_1$ and $A_2, B_2 \in \mathbb{G}_2$, $e(A_1 + B_1, A_2) = e(A_1, A_2)e(B_1, A_2)$ and $e(A_1, A_2 + B_2) = e(A_1, A_2)e(A_1, B_2)$.
2. **Non-degeneracy:** If $e(P_1, P_2) = 1_T$, the identity of $\mathbb{G}_T$, then either P_1 is the identity of $\mathbb{G}_1$ or P_2 is the identity of $\mathbb{G}_2$.
3. **Efficiency:** The map e is efficiently computable.

A bilinear map is called symmetric or a Type-1 bilinear map if $\mathbb{G}_1 = \mathbb{G}_2$; otherwise it is asymmetric. Asymmetric bilinear maps are further classified into Type-2 and Type-3 bilinear maps. In the Type-2 setting, there is an efficiently computable isomorphism either from $\mathbb{G}_1$ to $\mathbb{G}_2$ or from $\mathbb{G}_2$ to $\mathbb{G}_1$ whereas in the Type-3 setting no such isomorphisms are known. Previous works [16,8,5] have established that the Type-3 setting is the most efficient from an implementation point of view.

We introduce some notation: Given generators P_1 of $\mathbb{G}_1$ and P_2 of $\mathbb{G}_2$ and elements $R_1 \in \mathbb{G}_1$ and $R_2 \in \mathbb{G}_2$, the notation $R_1 \sim R_2$ indicates that R_1 has the same discrete logarithm to base P_1 as that of R_2 to base P_2. For a set $\mathbb{X}$, let $x \in_{\mathrm{R}} \mathbb{X}$ denote that x is a uniform random element of $\mathbb{X}$.

In the following, we will assume the availability of a Type-3 bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ where $\mathbb{G}_1 = \langle P_1 \rangle$, $\mathbb{G}_2 = \langle P_2 \rangle$ and both $\mathbb{G}_1$ and $\mathbb{G}_2$ are groups of the same prime order p. Being of prime order, any non-identity element of $\mathbb{G}_1$ is a generator of the group and the same holds for $\mathbb{G}_2$.

2.2 Hardness Assumption

We introduce a new hardness assumption for Type-3 pairings. Here we provide a discussion of this. The other standard hardness assumptions required in this work are DDH in $\mathbb{G}_1$, DLin and DBDH assumptions.

Let P_1 and P_2 be random generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. The DDH problem in $\mathbb{G}_1$ is to decide, given $(P_1, x_1P_1, x_2P_1, P_2, Z_1)$, whether $Z_1 = x_1x_2P_1$

or Z_1 is a random element of $\mathbb{G}_1$. Here $x_1, x_2 \in_R \mathbb{Z}_p$. Similarly one can define the DDH assumption in $\mathbb{G}_2$. In this case, an instance will have the form $(P_1, P_2, x_1P_2, x_2P_2, Z_2)$ and the task is to determine whether $Z_2 = x_1x_2P_2$ or whether Z_2 is a random element of $\mathbb{G}_2$. For convenience we will denote the DDH problem in $\mathbb{G}_1$ as DDH1 and that in $\mathbb{G}_2$ as DDH2. The symmetric external Diffie-Hellman (SXDH) assumption is that both DDH1 and DDH2 problems are hard. Note that for a symmetric pairing (i.e., for $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G} = \langle P \rangle$), DDH is easy to solve by comparing $e(P, Z_2)$ with $e(x_1P, x_2P)$.

We will use DDH1 in our proofs. But DDH2 is not directly applicable to our proofs. An instance of DDH2 has a single element P_1 of $\mathbb{G}_1$. For our proofs, we will require some information about x_1P_1 to be carried as part of the instance. If the instance is directly augmented by x_1P_1, then the problem becomes easy, since one can compute the pairing $e(x_1P_1, x_2P_2)$ and compare to $e(P_1, Z_2)$. Suppose that instead of x_1P_1 we include the elements zP_1 and zx_1P_1 where z is chosen randomly from $\mathbb{Z}_p$. This pair of elements carries some information about x_1P_1, but, not the element itself. An instance will now be $(P_1, zP_1, zx_1P_1, P_2, x_1P_2, x_2P_2, Z_2)$. It, however, is easy to check whether Z_2 equals $x_1x_2P_2$ by checking whether $e(zx_1P_1, x_2P_2)$ equals $e(zP_1, Z_2)$. This suggests that the information about zP_1 itself needs to be blinded by another randomiser. So, instead of having zP_1 directly, the elements dP_1, dzP_1 and dP_2 are included where d is a random element of $\mathbb{Z}_p$. The information about x_1P_1 is carried by the elements dP_1, dzP_1, zx_1P_1 and dP_2. Augmenting an instance of DDH2 with these elements embeds information about x_1P_1 but, does not seem to provide any way to use this information to determine whether Z_2 is real or random. The entire thing can now be formulated as an assumption in the following manner.

Assumption DDH2v. Let P_1, P_2 be random generators of $\mathbb{G}_1$, $\mathbb{G}_2$ respectively and let x_1, x_2, d, z be random elements of $\mathbb{Z}_p$. The DDH2v problem is to decide, given $(P_1, dP_1, dzP_1, zx_1P_1, P_2, dP_2, x_1P_2, x_2P_2, Z_2)$, whether $Z_2 = x_1x_2P_2$ or Z_2 is a random element of $\mathbb{G}_2$.

This corresponds to a two-level blinding of x_1P_1. We have seen that providing x_1P_1 directly or using a single-level blinding makes the problem easy. So, a two-level blinding is the minimum that one has to use to get to an assumption about hardness.

The assumption DDH2v (the "v" stands for variant) is no harder than DDH2. This is because an instance of DDH2v contains an embedded instance of DDH2 and an algorithm to solve DDH2 can be invoked on this embedded instance to solve the instance of DDH2v. On the other hand, there is no clear way of using an algorithm to solve DDH2v to solve DDH2. Intuitively, this is due to the fact that an instance of DDH2v contains some information about x_1P_1 whereas an instance of DDH2 does not contain any such information.

In our reduction, we will use the assumption DDH2v. Since assumption DDH2v does not appear earlier in the literature, it is a non-standard assumption. Having said this, we would also like to remark that DDH2v arises naturally as a minimal assumption when one tries to augment an instance of DDH2 with some

information about $x_1 P_1$ while maintaining the hardness of the problem. A proof of security of this assumption in the generic group model is provided in the full version [13]. We feel that assumption DDH2v will have applications elsewhere for schemes based on asymmetric pairings.

3 Framework for Conversion

Our goal is to transform Waters-2009 IBE scheme to the asymmetric setting so that we can reduce the number of components both in the ciphertext and the key. To that end, we first perform a straightforward conversion of Waters IBE from the setting of symmetric pairing to the setting of asymmetric pairing. (See [18] for the original description of Waters 2009 scheme.)

Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a Type 3 bilinear map and let P_1 and P_2 be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. After the conversion, either the ciphertext or the key will consist of elements of $\mathbb{G}_1$; the other will consist of elements from $\mathbb{G}_2$. Elements of $\mathbb{G}_1$ have shorter representation compared to those of $\mathbb{G}_2$. For encryption, we want the ciphertext to be short and hence we choose its elements to be from $\mathbb{G}_1$. The public parameters will consist of elements of $\mathbb{G}_1$ whereas the master secret key will consist of elements of $\mathbb{G}_2$. We note that if the final goal were to construct a signature scheme, then one would perform a conversion where the secret key consists of elements of $\mathbb{G}_1$.

A straightforward conversion will have the same structure as the one described in [18]. We use the convention in this and later schemes that the subscript 1 will denote elements of $\mathbb{G}_1$ while the subscript 2 will denote elements of $\mathbb{G}_2$. Further, messages are elements of $\mathbb{G}_T$ and identities are elements of $\mathbb{Z}_p$.

To generate the public parameters $\mathcal{PP}$, first choose α, b, a_1, a_2 at random from $\mathbb{Z}_p$ and consider the following. Let v, v' and v'' be random elements of $\mathbb{Z}_p$ and define $V_2 = vP_2$, $V_2' = v'P_2$ and $V_2'' = v''P_2$. Let $\tau = v + a_1 v'$ and $\tau' = v + a_2 v''$. Set $T_1 = \tau P_1$ and $T_1' = \tau' P_1$. The $\mathcal{PP}$ will have elements $Q_1, U_1, W_1 \in \mathbb{G}_1$ and correspondingly the master secret key will have elements $Q_2, U_2, W_2 \in \mathbb{G}_2$ with $Q_2 \sim Q_1$, $U_2 \sim U_1$ and $W_2 \sim W_1$. The structure of the $\mathcal{PP}$ and the $\mathcal{MSK}$ are as follows.

$$\begin{aligned}\mathcal{PP} \quad &: (P_1, bP_1, a_1P_1, a_2P_1, ba_1P_1, ba_2P_1, T_1, T_1', bT_1, bT_1',\\ &\quad Q_1, W_1, U_1, e(P_1, P_2)^{ba_1\alpha}).\\ \mathcal{MSK}&: (P_2, \alpha P_2, a_1\alpha P_2, V_2, V_2', V_2'', Q_2, W_2, U_2).\end{aligned}$$

Encrypt$(M, \mathsf{id}, \mathcal{PP})$: Randomisers $s_1, s_2, t, \mathsf{ctag}$ are chosen from $\mathbb{Z}_p$ and define $s = s_1 + s_2$. The ciphertext is $(C_0, C_1, \ldots, C_7, E_1, E_2, \mathsf{ctag})$ where the various elements are defined as follows.

$$\begin{aligned}&C_0 = M \cdot e(P_1, P_2)^{ba_1\alpha s_2}\\ &C_1 = bsP_1,\ C_2 = ba_1s_1P_1,\ C_3 = a_1s_1P_1,\ C_4 = ba_2s_2P_1,\\ &C_5 = a_2s_2P_1, C_6 = s_1T_1 + s_2T_1',\ C_7 = s_1bT_1 + s_2bT_1' - tW_1\\ &E_1 = t(\mathsf{id}Q_1 + \mathsf{ctag}W_1 + U_1),\ E_2 = tP_1\end{aligned}$$

KeyGen$(\mathsf{id}, \mathcal{MSK}, \mathcal{PP})$: Randomisers $r_1, r_2, z_1, z_2, \mathsf{ktag}$ are chosen from $\mathbb{Z}_p$ and define $r = r_1 + r_2$. The key $\mathcal{SK}_{\mathsf{id}}$ is $(K_1, \ldots, K_7, \mathsf{ktag})$ where the various elements are defined as follows.

$$K_1 = a_1\alpha P_2 + rV_2,\ K_2 = -\alpha P_2 + rV_2' + z_1P_2,\ K_3 = -z_1bP_2$$
$$K_4 = rV_2'' + z_2P_2,\ K_5 = -z_2bP_2,\ K_6 = r_2bP_2,\ K_7 = r_1P_2$$
$$D = r_1(\mathsf{id}Q_2 + \mathsf{ktag}W_2 + U_2).$$

The decryption algorithm (as described by Waters) requires 9 pairings and succeeds only if ctag in the ciphertext is not equal to ktag of the decryption key, an event which occurs with overwhelming probability (see [18] for the details).

Waters defines algorithms to generate semi-functional ciphertexts and keys. These cannot be computed without knowledge of the secret components and are only used in the security reduction. They are defined such that one should be able to decrypt a semi-functional ciphertext with a normal key and a normal ciphertext with a semi-functional key; but decryption of a semi-functional ciphertext with a semi-functional key should fail.

Semi-functional Ciphertext: Let $C_0', \ldots, C_7', E_1', E_2', \mathsf{ctag}$ be the ciphertext elements normally generated by the **Encrypt** algorithm for message M and identity id. Choose $\mu \in \mathbb{Z}_p$ at random. Let $V_1' = v'P_1$ and $V_1'' = v''P_1$ so that $V_1' \sim V_2'$ and $V_1'' \sim V_2''$. The semi-functional ciphertext generation algorithm will modify the normal ciphertext as: $C_0 = C_0'$, $C_1 = C_1'$, $C_2 = C_2'$, $C_3 = C_3'$, $E_1 = E_1'$, $E_2 = E_2'$ and

$$C_4 = C_4' + ba_2\mu P_1,\ C_5 = C_5' + a_2\mu P_1,\ C_6 = C_6' - a_2\mu V_1'',\ C_7 = C_7' - ba_2\mu V_1''.$$

Semi-functional Key: Let $K_1', \ldots, K_7', D', \mathsf{ktag}$ be secret key components normally generated by the **KeyGen** algorithm for the identity id. Choose at random $\gamma \in \mathbb{Z}_p$. The semi-functional key generation algorithm will modify the normal key as: $K_3 = K_3'$, $K_5 = K_5'$, $K_6 = K_6'$, $K_7 = K_7'$, $D = D'$ and

$$K_1 = K_1' - a_1a_2\gamma P_2,\ K_2 = K_2' + a_2\gamma P_2,\ K_4 = K_4' + a_1\gamma P_2.$$

It is easy to see that one can decrypt a semi-functional ciphertext with a normal key and a normal ciphertext with a semi-functional key. However, decryption of a semi-functional ciphertext with a semi-functional key will fail because the masking factor $e(P_1, P_2)^{ba_1\alpha s_2}$ will be blinded by the factor $e(P_1, P_2)^{ba_1a_2\mu\gamma}$.

Security Proof. The security argument for the scheme proceeds through $q + 3$ games where q is the number of key extraction queries made by the adversary. These games are

$$\mathrm{Game}_{real}, \mathrm{Game}_0, \ldots, \mathrm{Game}_q, \mathrm{Game}_{final}.$$

The transition between these games can be seen as three different reductions.

First Reduction: The transition from Game_{real} to Game_0 is made by replacing the challenge ciphertext by a semi-functional ciphertext. It is argued that detecting this change should be hard.

Second Reduction: There is a sequence of q changes from Game_{k-1} to Game_k (for $k = 1, \ldots, q$). The k-th change is as follows. For the queries numbered 1 to $k-1$, the adversary is given a semi-functional key; for queries numbered $k+1$ to q, the adversary is given a normal key. For the k-th query, the adversary is given a response such that deciding whether the response is normal or semi-functional is hard. The challenge ciphertext is semi-functional as in the first reduction.

Third Reduction: This tackles the transition from Game_q to Game_{final}. At this point, all responses to key extraction queries are semi-functional and so is the challenge ciphertext. In the last transition, the challenge ciphertext is changed such that deciding whether it is the encryption of a message or whether it is statistically independent of the challenge messages is hard.

The first and second reductions are based on the hardness of the DLin problem whereas the third reduction is based on the hardness of the DBDH problem. In the proof, the second reduction is the most complex step. The subtle point is that the simulator should not be able to generate a semi-functional ciphertext for the k-th identity which will allow it to easily determine whether the key for this identity is semi-functional or not. This is ensured by using algebraic techniques from [1] to create ktag using a pair-wise independent function so that the simulator is able to create a semi-functional ciphertext for id_k only with $\mathsf{ctag} = \mathsf{ktag}$, in which case decryption fails unconditionally and hence the simulator gains no information.

3.1 An Analysis

Our conversion to asymmetric pairing and subsequent simplifications are based on an analysis of the various scalars used in the scheme and their respective roles in the proof. Based on the scheme itself and a study of the three reductions used by Waters, we make the following observations.

1. $\mathcal{PP}$ uses the scalars a_1, a_2 and b, while $\mathcal{MSK}$ uses the scalars α and a_1.
2. Key generation uses scalar randomisers r_1, r_2 and z_1, z_2. The scalar r is set to $r_1 + r_2$. We will call this the *split of* r.
3. Ciphertext generation uses the scalar randomisers s_1, s_2 and t. The scalar s is set to $s_1 + s_2$. We will call this the *split of* s.
4. The first two reductions in Waters proof are based on the DLin assumption. The first reduction uses the split of s whereas the second reduction uses the split of r.

For conversion to asymmetric pairing, the following points are to be noted. These have been inferred from a careful analysis of the security proof in [18].

1. The scalar α needs to be retained.
2. The tags are chosen randomly and they play a crucial role in the security argument. We do not consider the question of removing tags in this paper. If the tags are removed, then it will be necessary to introduce copies of the

identity-hash (as done in [11]) to obtain the functionality of tags in the semi-functional components. This leads to an increase in the number of elements in the ciphertext and key.
3. There are three basic possibilities for simplification: remove the split of s; remove the split of r; remove z_1, z_2.
4. Getting rid of a_1 and a_2 and using a single a will eliminate the requirement of the split of s. This also means that the separate z_1 and z_2 are not required and instead a single z can be used.
5. Removing the split of r does not have a direct influence on the other scalars.
6. Removing the split of r and also z_1, z_2 means that the scalar b is no longer required.
7. In all but one of our schemes, the scalar t is kept either as part of the ciphertext or as part of the key. In the final scheme, we show that the scalar t can also be removed. For this scheme, there is a single randomiser s for the ciphertext and a single randomiser r for the key, excluding the tags.
8. If the first reduction is to be based on DLin, then the split of s and a_1,a_2 must be retained. If the split is removed, then we can base the first reduction on DDH1.
9. If the split of r is retained, then the second reduction has to be based on DLin. If it is removed, we can no longer base the second reduction on DLin. However, it can neither be based on DDH2 for the following reason. An instance of DDH2 will provide P_1 and some elements of $\mathbb{G}_2$. Apart from P_1 no other element of $\mathbb{G}_1$ is provided. The $\mathcal{PP}$ consists of elements of $\mathbb{G}_1$ which have to be related to the instance in some way. Just having P_1 does not provide any way to construct the $\mathcal{PP}$ in the second reduction. So, removing the split of r implies that the second reduction can be based on neither DLin nor DDH2. The assumption DDH2v introduced in Section 2 provides the necessary mechanism for carrying the proof through.

Based on the above points, we explore the different natural ways in which Waters 2009 IBE scheme can be converted to asymmetric pairing. These are discussed below.

Scheme 1: Remove the split of s. This eliminates the requirement of having separate a_1, a_2 and z_1, z_2. Reductions of ciphertext and key are by two elements each. Removing the split of s allows the first reduction to be based on DDH1. Since the split of r is retained, the second reduction is still based on DLin.

Scheme 2: Retain the split of s; this means that separate a_1 and a_2 are required. Remove the split of r and also remove z_1 and z_2; this means that b can be removed. Leads to reductions of ciphertext and key by 3 elements each. The first reduction of the proof can be based on DLin, but, the second reduction cannot be based on either DLin or DDH2.

Scheme 3: Remove the split of s; retain the split of r but, remove z. Reductions of ciphertext and key are by 3 elements each. In the proof, the first reduction can be based on DLin. The second reduction cannot be based on DDH2.

Neither can it be based on DLin. This requires a more involved reasoning which we provide in the full version [13].

Scheme 4: Remove the splits of both r and s, but, retain z. Ciphertext and key are reduced by 3 elements each. In the proof, the first reduction can be based on DDH1, but, the second reduction cannot be based on either DLin or DDH2.

Scheme 5: Remove the splits of both r and s and also remove z. Ciphertext and keys are reduced by 4 elements each. As in the previous case, the first reduction of the proof can be based on DDH1, but, the second reduction cannot be based on either DLin or DDH2.

Scheme 6: In Schemes 1 to 5, the randomiser t is present in the ciphertext. In Scheme 6, the splits of both r and s are removed; z is removed and the role of t is played by s. This leads to a scheme where there is exactly one randomiser for encryption and exactly one randomiser for key generation. Compared to Waters' IBE [18], ciphertext size is reduced by 5 elements and the key size by 4 elements. The first reduction of the proof can be based on DDH1, while the second reduction is based on assumption DDH2v.

In Table 3, we provide the use of scalars in the various schemes. This illustrates the manner in which the simplification has been obtained.

Table 3. Usage of scalars in various schemes. Note that all the schemes use ktag for key generation and ctag for encryption.

scheme	$\mathcal{PP}$	$\mathcal{MSK}$	key gen	enc
Waters-09 [18]	α, a_1, a_2, b	α, a_1	$r_1, r_2, (r = r_1 + r_2), z_1, z_2$	$s_1, s_2, (s = s_1 + s_2), t$
Scheme 1	α, a, b	α, b	$r_1, r_2, (r = r_1 + r_2), z$	s, t
Scheme 2	α, a_1, a_2	α	r	$s_1, s_2, (s = s_1 + s_2), t$
Scheme 3	α, a, b	α, b	$r_1, r_2, (r = r_1 + r_2)$	s, t
Scheme 4	α, a, b	α, b	r, z	s, t
Scheme 5	α, a	α	r	s, t
Scheme 6	α, a	α	r	s

4 Constructions

In this section, we provide the description of Scheme 6. In the full version [13], the description of Scheme 1 along with its security proof are provided. For Schemes 2 to 5, only the descriptions are provided in the full version. These schemes primarily serve the purpose of showing the stepping stones in moving from Scheme 1 to Scheme 6. In Section 5, we present a security proof for Scheme 6.

4.1 Scheme 6

Descriptions of $\mathcal{PP}$, $\mathcal{MSK}$, ciphertext generation, key generation and decryption are provided.

Parameters $P_1, P_2, Q_1, W_1, U_1, Q_2, W_2, U_2, \alpha$ are chosen as described in Section 3. Let a, v, v' be random elements of $\mathbb{Z}_p$. Set $V_2 = vP_2$, $V_2' = v'P_2$ and $\tau = v + av'$ so that $\tau P_2 = V_2 + aV_2'$.

$\mathcal{PP}$: $(P_1, aP_1, \tau P_1, Q_1, W_1, U_1, e(P_1, P_2)^\alpha)$.
$\mathcal{MSK}$: $(P_2, \alpha P_2, V_2, V_2', Q_2, W_2, U_2)$.

Encrypt$(M, \mathsf{id}, \mathcal{PP})$: Choose random s, ctag from $\mathbb{Z}_p$; ciphertext $\mathcal{C}$ is given by $(C_0, C_1, C_2, C_3, E, \mathsf{ctag})$ where the elements are defined as follows.

$C_0 = M \cdot e(P_1, P_2)^{\alpha s}$,
$C_1 = sP_1,\ C_2 = asP_1,\ C_3 = -\tau s P_1 + sW_1,\ E = s(\mathsf{id}Q_1 + \mathsf{ctag}W_1 + U_1)$.

KeyGen$(\mathsf{id}, \mathcal{MSK}, \mathcal{PP})$: Choose random r, ktag from $\mathbb{Z}_p$; the secret key $\mathcal{SK}_{\mathsf{id}}$ is $(K_1, K_2, K_3, D, \mathsf{ktag})$ where the elements are defined as follows.

$K_1 = \alpha P_2 + rV_2,\ K_2 = rV_2',\ K_3 = rP_2,\ D = r(\mathsf{id}Q_2 + \mathsf{ktag}W_2 + U_2)$.

Decrypt $(\mathcal{C}, \mathsf{id}, \mathcal{SK}_{\mathsf{id}}, \mathcal{PP})$: As before, decryption succeeds only when $\mathsf{ctag} \neq \mathsf{ktag}$. Define $\vartheta = (\mathsf{ctag} - \mathsf{ktag})^{-1}$. Decryption is done by unmasking the message as follows.

$$M = \frac{C_0}{e(C_1, K_1 + \vartheta D)e(C_2, K_2)e(C_3 - \vartheta E, K_3)}$$

The correctness of decryption is shown by the following calculations. We break up the denominator into two parts - A_1 and A_2 such that the product $A_1 A_2$ gives the masking factor.

$$\begin{aligned}
A_1 &= e(C_1, \vartheta D)e(-\delta E, K_3) \\
&= e(C_1, D)^\vartheta e(-E, K_3)^\vartheta \\
&= e(sP_1, r(\mathsf{id}Q_2 + \mathsf{ktag}W_2 + U_2))^\vartheta e(-s(\mathsf{id}Q_1 + \mathsf{ctag}W_1 + U_1), rP_2)^\vartheta \\
&= e(-(\mathsf{id}Q_1 + \mathsf{ktag}W_1 + U_1), P_2)^{-rs\vartheta} e(\mathsf{id}Q_1 + \mathsf{ctag}W_1 + U_1, P_2)^{-rs\vartheta} \\
&= e(\vartheta(\mathsf{ctag} - \mathsf{ktag})W_1, P_2)^{rs} \\
&= e(W_1, P_2)^{-rs} \\
A_2 &= e(C_1, K_1)e(C_2, K_2)e(C_3, K_3) \\
&= e(sP_1, \alpha P_2 + rV_2)e(asP_1, rV_2')e(\tau s P_1 + sW_1, rP_2) \\
&= e(P_1, P_2)^{\alpha s} e(P_1, V_2 + aV_2' - \tau P_2)^{rs} e(W_1, P_2)^{rs} \\
&= e(P_1, P_2)^{\alpha s} e(W_1, P_2)^{rs}
\end{aligned}$$

Extension to HIBE: Waters extends the IBE scheme in [18] in a natural way to a HIBE scheme. In the full version of this paper [13], we show that our simplification of Waters scheme retains the original flexibility and describe a HIBE which extends Scheme 6. This HIBE scheme is secure under the DDH1, DDH2v and the DBDH assumptions and provides lesser and smaller parameters and better efficiencies of key generation, delegation, encryption and decryption compared to the HIBE in [18]. The security proof for the HIBE follows the Shi-Waters model [15] and is given in [13].

Conversion to Signature Scheme: There is a "dual" of Scheme 6 where the ciphertext elements are in $\mathbb{G}_2$ and decryption keys consist of elements of $\mathbb{G}_1$. Using Naor's observation, this dual of Scheme 6 can be converted to a secure signature scheme. The signatures will be composed of elements of $\mathbb{G}_1$ and will be smaller than the signatures obtained by the conversion of Waters' 2009 scheme to a signature scheme. In a similar manner, one can convert the dual of our HIBE to obtain a HIBS scheme where signatures consist of elements of $\mathbb{G}_1$.

4.2 Broadcast Encryption

The full version of Waters paper described a public key broadcast encryption (BE) scheme based on the dual system IBE in [18]. In this section, we describe a BE scheme based on Scheme 6 in the Type-3 pairing setting. The security proof is given in the full version [13] and is based on the hardness of the DDH1, DDH2v and the DBDH problems. The new BE scheme provides adaptive security and is more efficient than previously known BE schemes providing adaptive security [9,18]. In what follows, n denotes the total number of users and $\{1, \ldots, n\}$, the set of users.

Setup(n): Generators $P_1 \in_R \mathbb{G}_1$ and $P_2 \in_R \mathbb{G}_2$ are chosen. Also choose random elements $Q_{1,1}, \ldots Q_{1,n}, W_1 \in \mathbb{G}_1$ and $Q_{2,1}, \ldots, Q_{2,n}, W_2 \in \mathbb{G}_2$ such that $Q_{2,i} \sim Q_{1,i}$ for $1 \leq i \leq n$, $W_2 \sim W_1$. Let α, a, v, v' be random elements of $\mathbb{Z}_p$. Set $V_2 = vP_2$, $V_2' = v'P_2$ and $\tau = v + av'$ so that $\tau P_2 = V_2 + aV_2'$. The public key $\mathcal{PK}$ and secret key $\mathcal{SK}$ are given by

$$\begin{aligned} \mathcal{PK} &: (P_1, aP_1, \tau P_1, Q_{1,1}, \ldots, Q_{1,n}, W_1, e(P_1, P_2)^\alpha). \\ \mathcal{SK} &: (P_2, \alpha P_2, V_2, V_2', Q_{2,1}, \ldots, Q_{2,n}, W_2). \end{aligned}$$

Encrypt$(\mathcal{PK}, S \subseteq \{1, \ldots, n\}, M)$: Choose random s from $\mathbb{Z}_p$; ciphertext $\mathcal{C}$ for the subset S of users is (C_0, C_1, C_2, C_3, E) where the elements are defined as follows.

$$\begin{aligned} &C_0 = M \cdot e(P_1, P_2)^{\alpha s}, \\ &C_1 = sP_1,\ C_2 = asP_1,\ C_3 = -\tau s P_1 + sW_1,\ E = s(\textstyle\sum_{i \in S} Q_{1,i}). \end{aligned}$$

KeyGen$(\mathcal{SK}, j \in \{1, \ldots, n\})$: Let r be chosen at random from $\mathbb{Z}_p$; secret key for user j is $\mathcal{SK}_j = (K_1, K_2, K_3, D, \forall_{i \neq j} D_i)$ where the elements are defined as follows.

$$\begin{aligned} &K_1 = \alpha P_2 + rV_2,\ K_2 = rV_2',\ K_3 = rP_2 \\ &D = r(Q_{2,j} + W_2),\ D_i = rQ_{2,i} \text{ for } i \neq j. \end{aligned}$$

Decrypt $(\mathcal{C}, S, \mathcal{SK}_j)$: Decryption works only if $j \in S$. Unmask the message as

$$M = \frac{C_0}{e(C_1, K_1 - D - \sum_{\substack{i \in S \\ i \neq j}} D_i) e(C_2, K_2) e(C_3 + E, K_3)}.$$

5 Security Proof for Scheme 6

First we define the semi-functional key and ciphertext for Scheme 6. As mentioned earlier, these are used only in the security reductions and are not part of the scheme itself.

Semi-functional Ciphertext: Let $(C'_0, C'_1, C'_2, C'_3, E', \mathsf{ctag})$ be a normal ciphertext. Choose a random μ from $\mathbb{Z}_p$. The semi-functional ciphertext is given by $(C_0, C_1, C_2, C_3, E, \mathsf{ctag})$ where $C_0 = C'_0$, $C_1 = C'_1$, $C_2 = C'_2 + \mu P_1$, $C_3 = C'_3 - \mu V'_1$ and $E = E'$.

Semi-functional Key: Let $(K'_1, K'_2, K'_3, D, \mathsf{ktag})$ be a normal key. Choose a random γ from $\mathbb{Z}_p$. The semi-functional key is $(K_1, K_2, K_3, D, \mathsf{ktag})$ where $K_1 = K'_1 - a\gamma P_2$, $K_2 = K'_2 + \gamma P_2$, $K_3 = K'_3$ and $D = D'$.

Let Game_{real}, Game_k (for $0 \leq k \leq q$) and Game_{final} be defined as in Section 3. Let Let X_{real}, X_k and X_{final} denote the events that the adversary wins in Game_{real}, Game_k and Game_{final} for $0 \leq k \leq q$ respectively.

Lemma 1. *If there exists an adversary $\mathcal{A}$ such that $\mathsf{Adv}^{\mathcal{A}}_{\text{Game}_{real}} - \mathsf{Adv}^{\mathcal{A}}_{\text{Game}_0} = \varepsilon$, then we can build an algorithm $\mathcal{B}$ having advantage ε in solving the* DDH1 *problem.*

Proof. The algorithm $\mathcal{B}$ receives $(P_1, sP_1, aP_1, P_2, Z_1)$ as an instance of DDH1. We describe how it will simulate each phase in the security game.

Setup: $\mathcal{B}$ chooses random elements $\alpha, y_v, y'_v, y_q, y_w, y_u$ from $\mathbb{Z}_p$ and sets the parameters as follows: $P_1 = P_1$, $aP_1 = aP_1$, $Q_1 = y_q P_1$, $W_1 = y_w P_1$, $U_1 = y_u P_1$, $P_2 = P_2$, $V_2 = y_v P_2$, $V'_2 = y'_v P_2$, $Q_2 = y_q P_2$, $W_2 = y_w P_2$, $U_2 = y_u P_2$. The element τP_1 is computed as $y_v P_1 + y'_v(aP_1)$ implicitly setting $\tau = y_v + ay'_v$. The simulator computes the remaining parameters using α and gives the following public parameters to $\mathcal{A}$: $\mathcal{PP} = (P_1, P_2, aP_1, \tau P_1, Q_1, W_1, U_1, e(P_1, P_2)^\alpha)$.

Phase 1: $\mathcal{A}$ makes a number of key extract queries. $\mathcal{B}$ knows the master secret and using that it returns a normal key for every key extract query made by $\mathcal{A}$.

Challenge: $\mathcal{B}$ receives the target identity id^* and two messages M_0 and M_1 from $\mathcal{A}$. It chooses $\beta \in \{0, 1\}$ at random. To encrypt M_β, $\mathcal{B}$ chooses ctag^* at random from $\mathbb{Z}_p$ and computes the ciphertext elements as follows: $C_0 = M_\beta \cdot e(sP_1, P_2)^\alpha$, $C_1 = sP_1$, $C_2 = Z_1$, $C_3 = -y_v(sP_1) - y'_v Z_1 + y_w(sP_1)$ and $E = (\mathsf{id}^* y_q + \mathsf{ctag}^* y_w + y_u)(sP_1)$. $\mathcal{B}$ returns $\mathcal{C}^* = (C_0, C_1, C_2, C_3, E, \mathsf{ctag}^*)$ to $\mathcal{A}$.

If $Z_1 = asP_1$ then the challenge ciphertext is normal; otherwise if Z_1 is a random element of $\mathbb{G}_1$ i.e., $Z_1 = (as + c)P_1$ then the ciphertext is semi-functional with $\mu = c$. Note that, to check whether $\mathcal{C}^*$ is semi-functional or not, $\mathcal{B}$ itself could try to decrypt it with a semi-functional key for id^*. However since aP_2 is not known to $\mathcal{B}$, it cannot create such a key.

Phase 2: As in first phase, $\mathcal{B}$ returns a normal key for every query.

Guess: The adversary returns its guess β' to $\mathcal{B}$.

If $\mathcal{C}^*$ is normal then $\mathcal{B}$ simulates Game_{real} and if it is semi-functional $\mathcal{B}$ simulates Game_0. Therefore if $\mathcal{A}$ is able to distinguish between Game_{real} and Game_0,

then the $\mathcal{B}$ can solve the DDH1 problem with advantage

$$\mathsf{Adv}^{\mathcal{B}}_{\mathrm{DDH1}} = |\Pr[X_{real}] - \Pr[X_0]| = \mathsf{Adv}^{\mathcal{A}}_{\mathrm{Game}_{real}} - \mathsf{Adv}^{\mathcal{A}}_{\mathrm{Game}_0} = \varepsilon.$$

□

Lemma 2. *If there exists an adversary $\mathcal{A}$ such that $\mathsf{Adv}^{\mathcal{A}}_{\mathrm{Game}_{k-1}} - \mathsf{Adv}^{\mathcal{A}}_{\mathrm{Game}_k} = \varepsilon$, then we can build an algorithm $\mathcal{B}$ having advantage ε in breaking the assumption* DDH2v.

Proof. Let $(P_1, dP_1, dzP_1, zx_1P_1, P_2, dP_2, x_1P_2, x_2P_2, Z_2)$ denote the instance of DDH2v that $\mathcal{B}$ receives.

Setup: $\mathcal{B}$ chooses random elements $a, \alpha, \lambda, \nu, y'_v, y_q, y_u, y_w \in_R \mathbb{Z}_p$ and sets the parameters as follows. $P_1 = P_1, P_2 = P_2, Q_2 = -\lambda(dP_2) + y_qP_2$, $U_2 = -\nu(dP_2) + y_uP_2$, $W_2 = dP_2 + y_wP_2$, $V_2 = -a(x_1P_2)$ and $V'_2 = x_1P_2 + y'_vP_2$ setting $\tau = ay'_v$ using which one can compute $\tau P_1 = ay'_vP_1$. The public parameters Q_1, W_1, U_1 can be computed since $\mathcal{B}$ has dP_1. The remaining parameters required to provide $\mathcal{PP}$ to $\mathcal{A}$ are computed using a, α and other elements of the problem instance.

Phases 1 and 2: The key extraction queries for identities $\mathsf{id}_1, \ldots, \mathsf{id}_q$ are answered in the following way. For $i < k$, a semi-functional key is returned and for $i > k$ a normal key is returned. Note that normal and semi-functional keys can be generated since $\mathcal{B}$ has the $\mathcal{MSK}$ and knows a. For $i = k$, a normal key K'_1, K'_2, K'_3, D' is generated using randomiser $r' \in_R \mathbb{Z}_p$, $\mathsf{ktag} = \lambda\mathsf{id}_k + \nu$ and then modified as: $K_1 = K'_1 - aZ_2$, $K_2 = K'_2 + Z_2 + y'_v(x_2P_2)$, $K_3 = K'_3 + x_2P_2$ and $D = D + (y_q\mathsf{id} + y_w\mathsf{ktag} + y_u)(x_2P_2)$, thus implicitly setting $r = r' + x_2$. Since dx_2P_2 is not known to $\mathcal{B}$ it can create D only when $\mathsf{ktag} = \lambda\mathsf{id}_k + \nu$. If $Z_2 = x_1x_2P_2$ then the key for id_k will be normal and otherwise it will be semi-functional with $\gamma = c$ where $Z_2 = (x_1x_2 + c)P_2$. Note that a semi-functional ciphertext for id_k with any value of ctag except for $\neq \lambda\mathsf{id}_k + \nu$ cannot be generated without the knowledge of dx_1zP_1 which is neither available from the assumption nor can be computed by $\mathcal{B}$. This rules out the obvious way of checking whether the key for id_k is semi-functional or not.

Challenge: $\mathcal{B}$ receives two messages M_0, M_1 and a challenge identity id^* during the challenge phase. It chooses $\beta \in_R \{0, 1\}$ and sets $\mathsf{ctag}^* = \lambda\mathsf{id}^* + \nu$. Since λ and ν are chosen independently and uniformly at random, the function $\lambda X + \nu$ is a pairwise independent function for a variable X over $\mathbb{Z}_p$. This causes the tag values of the challenge ciphertext and the k-th key to appear properly distributed from the adversary's view. $\mathcal{B}$ computes the ciphertext elements as: $C_0 = e(zx_1P_1, P_2)^{\alpha}$, $C_1 = zx_1P_1$, $C_2 = a(zx_1P_1) + dzP_1$, $C_3 = (y_w - ay'_v)(zx_1P_1) - y'_v(dzP_1)$ and $E = (y_q\mathsf{id} + \mathsf{ctag}^*y_w + y_u)(zx_1P_1)$, setting $s = zx_1$ and $\mu = dz$. It is easy to check that C_3 is well-formed.

Now $\mathcal{A}$ will be able to distinguish between Game_{k-1} and Game_k if it can decide whether $\mathcal{SK}_{\mathsf{id}_k}$ is normal or semi-functional. In this case $\mathcal{B}$ can break the assumption DDH2v with advantage

$$\mathsf{Adv}^{\mathcal{B}}_{\mathrm{DDH2v}} = |\Pr[X_{k-1}] - \Pr[X_k]| = \mathsf{Adv}^{\mathcal{A}}_{\mathrm{Game}_{k-1}} - \mathsf{Adv}^{\mathcal{A}}_{\mathrm{Game}_k} = \varepsilon.$$

□

Lemma 3. *If there exists an adversary $\mathcal{A}$ such that* $\mathsf{Adv}^{\mathcal{A}}_{\mathrm{Game}_q} - \mathsf{Adv}^{\mathcal{A}}_{final} = \varepsilon$, *then we can build an algorithm $\mathcal{B}$ having advantage ε in breaking the* DBDH *assumption.*

Proof. $\mathcal{B}$ receives $(P_1, xP_1, aP_1, sP_1, P_2, xP_2, aP_2, sP_2, Z)$ as an instance of the DBDH problem.
Setup: With y_v, y'_v, y_q, y_w, y_u chosen at random from $\mathbb{Z}_p$, $\mathcal{B}$ sets the parameters as: $P_1 = P_1$, $P_2 = P_2$, $aP_1 = aP_1$, $V_2 = y_vP_2$, $V'_2 = y'_vP_2$, $\tau P_1 = y_vP_1 + y'_v(aP_1)$, $Q_1 = y_qP_1$, $W_1 = y_wP_1$, $U_1 = y_uP_1$, $e(P_1, P_2)^{\alpha} = e(xP_1, aP_2)$, thus implicitly setting $a = a$, $\alpha = xa$ and $\tau = y_v + ay'_v$. The remaining parameters can be computed easily. $\mathcal{B}$ returns $\mathcal{PP}$ to $\mathcal{A}$.
Phases 1 and 2: When $\mathcal{A}$ asks for the secret key for the i'th identity id_i, $\mathcal{B}$ chooses at random $r, \mathsf{ktag}, \gamma' \in \mathbb{Z}_p$ implicitly setting $\gamma' = x - \gamma$. It then computes a semi-functional key for id_i as follows.

$$K_1 = \gamma'(aP_2) + rV_2 = xaP_2 - a\gamma P_2 + rV_2 = \alpha P_2 + rV_2 - a\gamma P_2$$

$$K_2 = rV'_2 - \gamma' P_2 + xP_2 = rV'_2 - xP_2 + \gamma P_2 + xP_2 = rV'_2 + \gamma P_2$$

$$K_3 = rP_2, D = r(\mathsf{id}_i Q_2 + \mathsf{ktag} W_2 + U_2).$$

Here $\mathcal{B}$ knows γ' but not γ. Also, observe that $\mathcal{B}$ does not know α and hence cannot create a normal key.
Challenge: $\mathcal{B}$ receives the challenge identity id^* and two messages M_0 and M_1 from $\mathcal{A}$. It chooses $\beta \in \{0, 1\}$ and $\mathsf{ctag}^*, \mu' \in \mathbb{Z}_p$ at random and generates a semi-functional challenge ciphertext as follows. Here $\mathcal{B}$ implicitly sets $\mu' = \mu + as$ and it does not know μ.

$$C_0 = M_\beta \cdot Z$$

$$C_1 = sP_1, C_2 = \mu' P_1 = asP_1 + \mu P_1$$

$$C_3 = -y_v(sP_1) - \mu' y'_v P_1 + y_w(sP_1) = -\tau sP_1 - \mu V'_1 + sW_1$$

$$E = (y_q \mathsf{id}^* + y_w \mathsf{ctag}^* + y_u)(sP_1)$$

The challenge ciphertext $\mathcal{C}^* = (C_0, C_1, C_2, C_3, E, \mathsf{ctag}^*)$ is returned to $\mathcal{A}$. If $Z = e(P_1, P_2)^{xas}$ then $\mathcal{C}^*$ will be a semi-functional encryption of M_β; if Z is a random element of $\mathbb{G}_T$ then $\mathcal{C}^*$ will be a semi-functional encryption of a random message. If $\mathcal{A}$ can identify whether the game simulated was Game_q or Game_{final}, then $\mathcal{B}$ will be able to decide whether $Z = e(P_1, P_2)^{xas}$ or not and hence break the DBDH assumption with advantage

$$\mathsf{Adv}^{\mathcal{B}}_{\mathrm{DBDH}} = |\Pr[X_q] - \Pr[X_{final}]| = \mathsf{Adv}^{\mathcal{A}}_{\mathrm{Game}_q} - \mathsf{Adv}^{\mathcal{A}}_{\mathrm{Game}_{final}} = \varepsilon.$$

□

Theorem 1. *If the* DDH1, DDH2v *and* DBDH *assumptions hold, then no polynomial time adversary $\mathcal{A}$ making at most q key extraction queries can break the security of Scheme 6.*

Proof. Using lemmas 1, 2 and 3, we have for any polynomial time attacker $\mathcal{A}$,

$$\begin{aligned}\mathsf{Adv}^{\mathcal{A}}_{\text{Scheme 6}} &\leq |\Pr[X_{real}] - \Pr[X_0]| + \sum_{k=1}^{q} (|\Pr[X_{k-1}] - \Pr[X_k]|) \\ &\quad + |\Pr[X_q] - \Pr[X_{final}]| \\ &= \varepsilon_{\text{DDH1}} + q\varepsilon_{\text{DDH2v}} + \varepsilon_{\text{DBDH}}\end{aligned}$$

which is negligible in the security parameter κ. □

6 Conclusion

We have converted Waters dual system IBE scheme from the setting of symmetric pairings to that of asymmetric pairings. This has been done in a systematic manner going through several stages of simplifications. We have described in detail an IBE scheme, Scheme 6, which is quite simple and minimal in the sense that both encryption and key generation use one randomiser each. The security of Scheme 6 is based on two standard assumptions and a natural and minimal extension of the DDH assumption for $\mathbb{G}_2$. On the other hand, security of Scheme 1 is based on standard assumptions and reduces the sizes of ciphertexts and keys by 2 elements each from the original scheme of Waters.

Acknowledgement. We would like to thank the anonymous reviewers of PKC 2012 for helpful comments and suggestions.

References

1. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
2. Boneh, D., Boyen, X.: Secure Identity Based Encryption Without Random Oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443–459. Springer, Heidelberg (2004)
3. Boneh, D., Franklin, M.K.: Identity-Based Encryption from the Weil Pairing. SIAM J. Comput. 32(3), 586–615 (2003); Earlier version appeared in the proceedings of CRYPTO 2001
4. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF Formulas on Ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)
5. Chatterjee, S., Menezes, A.: On cryptographic protocols employing asymmetric pairings – the role of ψ revisited. Discrete Applied Mathematics 159(13), 1311–1322 (2011)
6. Chatterjee, S., Sarkar, P.: Trading Time for Space: Towards an Efficient IBE Scheme with Short(er) Public Parameters in the Standard Model. In: Won, D.H., Kim, S. (eds.) ICISC 2005. LNCS, vol. 3935, pp. 424–440. Springer, Heidelberg (2006)

7. Cocks, C.: An Identity Based Encryption Scheme Based on Quadratic Residues. In: Honary, B. (ed.) Cryptography and Coding 2001. LNCS, vol. 2260, pp. 360–363. Springer, Heidelberg (2001)
8. Galbraith, S.D., Paterson, K.G., Smart, N.P.: Pairings for cryptographers. Discrete Applied Mathematics 156(16), 3113–3121 (2008)
9. Gentry, C., Waters, B.: Adaptive Security in Broadcast Encryption Systems (with Short Ciphertexts). In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 171–188. Springer, Heidelberg (2009)
10. Lewko, A.: Tools for simulating features of composite order bilinear groups in the prime order setting. Cryptology ePrint Archive, Report 2011/490 (2011), http://eprint.iacr.org/
11. Lewko, A., Waters, B.: New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (2010)
12. Naccache, D.: Secure and practical identity-based encryption. IET Information Security 1(2), 59–64 (2007)
13. Ramanna, S.C., Chatterjee, S., Sarkar, P.: Variants of Waters' Dual System Primitives Using Asymmetric Pairings. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 298–315. Springer, Heidelberg (2012), http://eprint.iacr.org/
14. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
15. Shi, E., Waters, B.: Delegating Capabilities in Predicate Encryption Systems. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 560–578. Springer, Heidelberg (2008)
16. Smart, N.P., Vercauteren, F.: On computable isomorphisms in efficient asymmetric pairing-based systems. Discrete Applied Mathematics 155(4), 538–547 (2007)
17. Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
18. Waters, B.: Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)

From Selective to Full Security: Semi-generic Transformations in the Standard Model

Michel Abdalla[1], Dario Fiore[2,⋆], and Vadim Lyubashevsky[1]

[1] Département d'Informatique, École normale supérieure, France.
{Michel.Abdalla,Vadim.Lyubashevsky}@ens.fr
[2] Department of Computer Science, New York University, USA.
fiore@cs.nyu.edu

Abstract. In this paper, we propose an efficient, standard model, semi-generic transformation of selective-secure (Hierarchical) Identity-Based Encryption schemes into fully secure ones. The main step is a procedure that uses admissible hash functions (whose existence is implied by collision-resistant hash functions) to convert any selective-secure *wild-carded* identity-based encryption (WIBE) scheme into a fully secure (H)IBE scheme. Since building a selective-secure WIBE, especially with a selective-secure HIBE already in hand, is usually much less involved than directly building a fully secure HIBE, this transform already significantly simplifies the latter task. This black-box transformation easily extends to schemes secure in the Continual Memory Leakage (CML) model of Brakerski et al. (FOCS 2010), which allows us obtain a new fully secure IBE in that model. We furthermore show that if a selective-secure HIBE scheme satisfies a particular security notion, then it can be generically transformed into a selective-secure WIBE. We demonstrate that several current schemes already fit this new definition, while some others that do not obviously satisfy it can still be easily modified into a selective-secure WIBE.

1 Introduction

The concept of identity-based encryption (IBE) is a generalization of the standard notion of public-key encryption in which the sender can encrypt messages to a user based only on the identity of the latter and a set of user-independent public parameters. In these systems, there exists a trusted authority, called private key generator, that is responsible for generating decryption keys for all identities in the system. Since being introduced by Shamir in 1984 [28], IBE has received a lot of attention due to the fact that one no longer needs to maintain a separate public key for each user. Despite being an attractive concept, it was only in 2001 that the first practical IBE construction was proposed based on elliptic curve pairings [11]. Later that year, Cocks proposed an alternative IBE construction based on the quadratic residuosity problem [19].

⋆ Work done while at ENS.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 316–333, 2012.

The now-standard definition of security of IBE schemes, first suggested by Boneh and Franklin [11], is indistinguishability under adaptive chosen-identity attacks (we refer to it as *full security*). In this security model, the adversary is allowed to obtain secret keys for adaptively chosen identities before deciding the identity upon which it wishes to be challenged. By allowing these queries, this notion implicitly captures resistance against collusion attacks as different users should be unable to combine their keys in an attempt to decrypt ciphertexts intended to another user.

In 2002, Horwitz and Lynn introduced the notion of hierarchical identity-based encryption (HIBE), which allows intermediate nodes to act as private key generators. They also provided a two-level HIBE construction based on the Boneh-Franklin IBE scheme, but their scheme could provide full collusion resistance only in the upper level. The first HIBE scheme to provide full collusion resistance in all levels is due to Gentry and Silverberg [22]. Like the Horwitz-Lynn HIBE scheme, the Gentry-Silverberg HIBE scheme was also based on the Boneh-Franklin IBE scheme and proven secure in the random-oracle model [6].

The first HIBE to be proven secure in the standard model is due to Canetti, Halevi, and Katz [16], but in a weaker security model, called the *selective-identity* model. Unlike the security definitions used in previous constructions of (H)IBE schemes, the selective-identity model requires the adversary to commit to the challenge identity before obtaining the public parameters of the scheme. Despite providing weaker security guarantees, Canetti, Halevi, and Katz showed that the selective-identity model is sufficient for building forward-secure encryption schemes, which was the main motivation of their paper.

Although the selective-identity model has been considered in many works, and is interesting in its own right (e.g., it implies forward-secure public key encryption), if we focus solely on the (H)IBE application, then the selective notion is clearly unrealistic because it does not model the real capabilities of an adversary attacking a (H)IBE scheme. So while the design of selective-identity secure schemes seems to be an easier task, the quest for fully secure solutions is always considered the main goal for (H)IBE construction.

It is therefore a very interesting problem to investigate whether there are ways to efficiently convert a selective secure scheme into a fully secure one. In the random oracle model, this question has been resolved by Boneh, Boyen and Goh [9], who provided a very efficient black-box transformation. In the standard model, however, no such conversion is known[1], and all fully-secure (H)IBE schemes (e.g., [8], [30], [18]) had to be constructed and proved secure essentially from scratch.

Our Results. In this paper, we explore the relationship between selective-identity and fully secure (H)IBE schemes in the standard model.

[1] It was shown by Boneh and Boyen in [7] that any selective secure IBE scheme is already fully secure, but the concrete security degrades by a factor $1/|\mathcal{ID}|$, where $\mathcal{ID}$ is the scheme's identity space. Since $\mathcal{ID}$ is usually of exponential size, this conversion is too expensive in terms of efficiency to be considered practical.

From Selective-Secure WIBE to Fully-Secure HIBE. Our first main contribution is a generic construction of *fully-secure* HIBE schemes from *selective-pattern-secure* wildcarded identity-based encryption (WIBE) schemes. The notion of a WIBE, introduced by Abdalla *et al.* [1], is very similar to the notion of a HIBE except that the sender can encrypt messages not only to a specific identity, but to a whole range of receivers whose identities match a certain pattern defined through a sequence of fixed strings and a special wildcard symbol ($*$). The security notion, called selective-pattern security, requires the adversary to commit ahead of time to the pattern P^* that he intends to attack. He can then ask for the secret keys of any identity not matching P^*, and for the challenge ciphertext on any pattern P matching P^*. This notion of security is slightly more general than that given in [1]. Yet, as noted in Remark 1 at the end of Section 2, it is satisfied by all known WIBE constructions.

Our transformation from *any* selective-pattern-secure WIBE to a fully-secure HIBE is generic and relies on the notion of admissible hash functions (whose existence is implied by collision-resistant hash functions) introduced by Boneh and Boyen in [8]. Since building selective-pattern-secure WIBE schemes seems to be much easier than directly building a fully secure HIBE scheme, this transformation already significantly simplifies the latter task. In fact, it is worth noticing that the selective-pattern security of all currently-known instantiations of WIBE schemes (see [1]) follows from the selective-identity security of their respective underlying HIBE schemes.

One direct consequence of our construction is that several existing fully secure (H)IBE schemes can be seen as a particular case of our transformation. For instance, the fully secure IBE scheme of Boneh and Boyen in [8] turns out to be a particular case of our generic construction when instantiated with the selective-pattern-secure Boneh-Boyen WIBE scheme given in [1]. Likewise, the fully secure HIBE by Cash, Hofheinz, Kiltz, and Peikert [18] can be seen as the result of our generic transformation when applied to our new WIBE scheme in Section 5. Another consequence of our transformation is that one can obtain new constructions of fully secure HIBE schemes by applying our methodology to existing selective-pattern-secure WIBE schemes, such as the Boneh-Boyen-Goh WIBE in [1]. Interestingly, the result obtained from this instantiation closely resembles the Waters (H)IBE scheme [30].

An important point about our transformation from WIBE to (H)IBE is that it also works in the Continual Memory Leakage (CML) model [15,20]. In particular, we show how to modify the IBE scheme in [15] into a WIBE scheme and prove it selective-pattern-secure in the CML model under the same assumption. Then, by applying our transformation to this newly-constructed WIBE, we obtain a (CML) fully-secure version of the IBE in [15]. For lack of space we fully describe these extensions in the full version of our work.

The Role of WIBE in our Transformation. Somewhat surprisingly, our transformation seems to imply that the WIBE notion is of central importance when going from selective to full security in (H)IBE. To see why, one has to take a look at our proof strategy and at the notion of Admissible hash functions

(AHF). AHFs are a tool which allows to partition the identity space into two subsets, $\mathtt{B}$ and $\mathtt{R}$ (both of which are of exponential size) so that in the security proof the identities of secret key queries fall in $\mathtt{B}$ while the challenge identity falls in $\mathtt{R}$. In particular, by carefully selecting the AHFs parameters (as described in [8], for instance) one can make sure that the above (good) event occurs with non-negligible probability. In our proof from selective-secure WIBE to fully-secure HIBE, the simulator first uses AHFs to partition the identity space into $\mathtt{B}$ and $\mathtt{R}$. Next, it declares to the WIBE challenger a challenge pattern which corresponds to $\mathtt{R}$, by expressing $\mathtt{R}$ in the form of a pattern. By the property of AHFs, if the good event occurs (for all key derivation queries and the challenge identity chosen by the adversary), then the simulator can easily forward all queries to the WIBE challenger. In particular, it is guaranteed that the challenge identity falls in $\mathtt{R}$. When that happens, the simulator can output the challenge identity chosen by the adversary as its own challenge.

We remark that the proof strategy described above does not work if one starts from a selective-secure HIBE instead of a WIBE. Unlike the selective-WIBE simulator, the simulator against the selective security of a HIBE should commit to the challenge identity ID^* at the very beginning. And even if the simulator chooses the AHFs parameters so that all secret key queries fall in $\mathtt{B}$ and the challenge identity falls in $\mathtt{R}$, it still needs to guess ID^* in $\mathtt{R}$ at the very beginning. But the probability that the challenge identity chosen by the adversary matches such ID^* is $1/|\mathtt{R}|$, which is negligible (recall that both $\mathtt{B}$ and $\mathtt{R}$ are of exponential size).

Selective WIBE from selective HIBE. The second contribution of this paper is to identify conditions under which we can generically transform a selective-identity-secure HIBE scheme into a selective-pattern-secure WIBE scheme. Towards this goal, we introduce a new notion of security for HIBE schemes, called *security under correlated randomness*, which allows us to transform a given HIBE into a WIBE by simply re-encrypting the same message to a particular set of identities by reusing the same randomness. Informally speaking, in order for a HIBE scheme to be secure under correlated randomness, it must satisfy the following two properties. First, when given an encryption of the same message under the same randomness for two identity vectors $ID_0 = (ID_{0,1}, \ldots, ID_{0,j}, \ldots, ID_{0,\lambda})$ and $ID_1 = (ID_{1,1}, \ldots, ID_{1,j}, \ldots, ID_{1,\lambda})$ differing in exactly one position (say j), one can easily generate a ciphertext for any identity vector matching the pattern $ID = (ID_{1,1}, \ldots, *, \ldots, ID_{1,\lambda})$. Secondly, when given these two ciphertexts, the adversary should not be able to generate an encryption of the same message under the same randomness for any identity vector that does not match the pattern. In Section 4 we show that selective-correlated-randomness-secure HIBE schemes can be converted to selective-pattern-secure WIBEs. Moreover, in the full version, we show that several existing HIBE schemes already satisfy this slightly stronger notion of security, e.g., [7,9,30], and in particular we show that their security under correlated randomness black-box reduces to their selective-identity security.

Hence, if we combine our first generic transformation from selective-pattern-secure WIBE to fully-secure (H)IBE, together with our second result described above, we obtain a compiler that allows us to construct a fully secure (H)IBE starting from a selective-secure (H)IBE. In particular, the resulting transformation works in the standard model and is semi-generic because the second part assumes a specific property of the underlying scheme (i.e., security under correlated randomness). Nevertheless, by reducing the task of building fully secure HIBE schemes to that of building a selective-pattern-secure WIBE scheme, we believe that our result makes the former task significantly easier to achieve.

New WIBE Schemes. One final contribution of this paper are two constructions of selective-pattern-secure WIBE schemes. The first one, whose description is given in the full version of this paper, is obtained by modifying the IBE in [15]. It is based on pairings and is secure under the Decision Linear assumption in the CML model. Such modification essentially follows the correlated-randomness paradigm. Since for some technical reasons (related to the specific scheme) the selective-pattern security of this WIBE cannot be black-box reduced to the selective-identity security of the related IBE (like we do for other pairing-based WIBEs), we give a direct proof under the Decision Linear assumption. However, we notice that such proof closely follows the one in [15]. The second WIBE is based on lattices and its security follows from the selective-identity secure HIBE construction from [18]. Even though the Cash-Hofheinz-Kiltz-Peikert HIBE scheme does not meet the notion of security under correlated randomness introduced in Section 4 (because the scheme is not secure when the same randomness is reused for encryption), we show in Section 5 that one can easily modify it to obtain a selective-pattern-secure WIBE scheme. Similarly to the case of pairing-based WIBE schemes, the selective-pattern security of the new WIBE can be reduced directly to the selective-identity security of the original Cash-Hofheinz-Kiltz-Peikert HIBE scheme. However, in this case, it turns out to be even simpler to prove the selective-pattern security of our scheme directly from the decisional Learning With Errors Problem (LWE) [27,26].

Discussion. In this paper, we concentrate on building HIBE schemes that are adaptive-identity-secure against chosen-plaintext attacks. As shown by Boneh, Canetti, Halevi, and Katz [17,13,10], such schemes can easily be made chosen-ciphertext-secure with the help of one-time signature schemes or message authentication codes. Similarly to the (H)IBE schemes by Boneh and Boyen [8], by Waters [30], and by Cash, Hofheinz, Kiltz, and Peikert [18], the schemes obtained via our transformation are only provably secure when the maximum hierarchy's depth L is some fixed constant due to the loss of a factor which is exponential in L. While for lattice-based HIBE schemes [18,3,4], this seems to be the state of the art, the same is not true for pairing-based HIBE schemes. More precisely, there have been several proposals in recent years (e.g., [21,29,25,24]), which are fully secure even when the HIBE scheme has polynomially many levels. Most of these schemes use a new proof methodology, known as dual system encryption [29].

Organization. The paper is organized as follows. In Section 2, we start by recalling some standard definitions and notations used throughout the paper. Next, in Section 3, we present our first main contribution, which is a generic construction which can transform any selective-pattern-secure WIBE into a fully secure HIBE scheme. Then, in Section 4, we introduce the notion of security under correlated randomness for HIBE schemes and show how such schemes can be used to build selective-pattern-secure WIBEs. In Section 5, we show a selective-pattern-secure WIBE scheme that is obtained by transforming the Cash-Hofheinz-Kiltz-Peikert HIBE. Finally, in Section 6, we summarize some future directions left open by our work.

2 Basic Definitions

(Hierarchical) Identity Based Encryption. A *hierarchical identity-based encryption* scheme (HIBE) is defined by a tuple of algorithms $\mathcal{HIBE} = (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$, a message space $\mathcal{M}$, and an identity space $\mathcal{ID}$. The algorithm Setup is run by a trusted authority to generate a pair of keys (mpk, msk) such that mpk is made public, whereas msk is kept private. The users are hierarchically organized in a tree of depth L whose root is the trusted authority. The identity of a user at level $1 \leq \ell \leq L$ is represented by a vector $\overrightarrow{ID} = (ID_1, \ldots, ID_\ell) \in \mathcal{ID}^\ell$. A user at level ℓ with identity $\overrightarrow{ID} = (ID_1, \ldots, ID_\ell)$ can use the key derivation algorithm $\mathsf{KeyDer}(sk_{\overrightarrow{ID}}, \overrightarrow{ID}')$ to generate a secret key for any of its children $\overrightarrow{ID}' = (ID_1, \ldots, ID_\ell, ID_{\ell+1})$ at level $\ell + 1$. Since this process can be iterated, every user can generate keys for all its descendants. Then, every user holding the master public key mpk, can encrypt a message $m \in \mathcal{M}$ for the identity $\overrightarrow{ID}$ by running $C \xleftarrow{\$} \mathsf{Enc}(mpk, \overrightarrow{ID}, m)$. Finally, the ciphertext C can be decrypted by running the deterministic decryption algorithm, $m \leftarrow \mathsf{Dec}(sk_{\overrightarrow{ID}'}, C)$. For correctness, it is required that for all honestly generated master keys $(mpk, msk) \xleftarrow{\$} \mathsf{Setup}$, for all messages $m \in \mathcal{M}$, all identities $\overrightarrow{ID} \in \mathcal{ID}^\ell$ and all $\overrightarrow{ID}'$ ancestors of $\overrightarrow{ID}$, $m \leftarrow \mathsf{Dec}(\mathsf{KeyDer}(msk, \overrightarrow{ID}'), \mathsf{Enc}(mpk, \overrightarrow{ID}, m))$ holds with overwhelming probability. An IBE is defined as an HIBE with a hierarchy of depth 1.

The security of a HIBE scheme is captured by the standard notion of indistinguishability under chosen-plaintext attacks. Informally, this is captured by the following game. The adversary $\mathcal{A}$ receives as input the master public key and it can ask for the secret key of any identities of its choice. Then it chooses a challenge identity $\overrightarrow{ID}^*$ and two messages m_0 and m_1, and it is given the encryption of m_β under $\overrightarrow{ID}^*$ for a random β. The goal of the adversary is to guess β under the restriction that $\mathcal{A}$ never asks for the secret key of $\overrightarrow{ID}^*$.

In the context of hierarchical identity-based encryption a lot of works in the literature also considered a weaker notion of security, called *selective-identity* indistinguishability under chosen-plaintext attacks (IND-sHID-CPA). The main difference with the standard IND-HID-CPA notion is that here the adversary

is required to commit ahead of time to the challenge identity $\overrightarrow{ID}^*$. The rest of the game is the same as IND-HID-CPA. Sometimes, in order to have a clear distinction with the standard notion of IND-HID-CPA, the latter is called "full security".

Identity Based Encryption with Wildcards. The notion of *Identity-Based Encryption with Wildcards* was introduced by Abdalla *et al.* in [1] as a generalization of the HIBE's notion. A WIBE scheme is defined by a tuple of algorithms $\mathcal{WIBE} = (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$ that works exactly as a HIBE, except that here the encryption algorithm takes as input a value $P \in (\mathcal{ID} \cup *)^\ell$ (for $1 \leq \ell \leq L$), i.e., the pattern, instead of an identity vector. Such pattern may contain a special "don't care" symbol $*$, the wildcard, at some levels. An identity $\overrightarrow{ID} = (ID_1, \ldots, ID_\ell) \in \mathcal{ID}^\ell$ is said to *match* a pattern $P \in (\mathcal{ID} \cup *)^{\ell'}$, denoted as $\overrightarrow{ID} \in_* P$, if and only if $\ell \leq \ell'$ and $\forall i = 1, \ldots, \ell$: $ID_i = P_i$ or $P_i = *$. Note that under this definition, any ancestor of a matching identity is also a matching identity. This makes sense for the notion of WIBE, as any ancestor can derive the secret key of a matching descendant identity anyway. For any pattern $P \in (\mathcal{ID} \cup *)^\ell$, we denote with $\mathrm{W}(P)$ the set of indices $j \in [\ell]$ such that $P_j = *$. For correctness, it is required that for all honestly generated master keys $(mpk, msk) \xleftarrow{\$} \mathsf{Setup}$, for all messages $m \in \mathcal{M}$, all patterns $P \in (\mathcal{ID} \cup *)^{\ell'}$ and all identities $\overrightarrow{ID} \in \mathcal{ID}^\ell$ such that $\overrightarrow{ID} \in_* P$, $m \leftarrow \mathsf{Dec}(\mathsf{KeyDer}(msk, \overrightarrow{ID}), \mathsf{Enc}(mpk, P, m))$ holds with all but negligible probability.

Similarly to HIBEs, WIBE schemes allow for similar notions of security under chosen-plaintext attacks. In particular, in our work we consider only the notion of selective security that we call IND-sWID-CPA. Roughly speaking, it is similar to the IND-sHID-CPA notion for HIBE, except that here the adversary has to commit to a pattern P^* (instead of an identity $\overrightarrow{ID}^*$) at the beginning of the game. Next, when he has to choose the challenge pattern, he can provide any P that matches P^*, i.e., such that either P is an identity matching P^*, or P is a sub-pattern of P^*.

Remark 1. We notice that our notion of selective-security for WIBE schemes is slightly more general than the one that was originally proposed in [1]. The main difference is that in the original work of Abdalla *et al.* the notion is purely selective, meaning that the adversary declares the challenge pattern P^* at the beginning of the game, and later it receives an encryption of either m_0 or m_1 under P^*. Instead, our notion allows for more flexibility. Indeed, the adversary still declares P^* at the beginning of the game, but later it may ask the challenge ciphertext on a pattern P, possibly different from P^*, but such that P matches P^*. We stress that this property is not artificial for at least two reasons. First, it is more general than the previous one. Second, it is satisfied by all known WIBE schemes, and in particular we will show that it is satisfied by those schemes obtained through our transformation, from selective-secure HIBE to selective WIBE, that we describe in Section 4.

3 Fully-Secure HIBE from Selective-Secure WIBE

In this section we concentrate on the first part of our main result. We show how to construct a fully-secure HIBE scheme starting from any WIBE scheme that is secure only in a selective sense. Our transformation is black-box and makes use of admissible hash functions, a notion introduced by Boneh and Boyen in [8] that we recall below.

Admissible Hash Functions. Admissible hash functions were first introduced by Boneh and Boyen in [8] as a tool for proving the full security of their identity-based encryption scheme in the standard model. Such functions turn out to be particularly suitable for this purpose as they provide a way to implement the so-called "partitioning technique", a proof methodology that allows to secretly partition the identity space into two sets, the *blue* set and the *red* set, both of exponential size, so that there is a non-negligible probability that the adversary's secret key queries fall in the blue set and the challenge identity falls in the red set. This property has been shown useful to prove the full security of some identity-based encryption schemes (e.g., [8,30,18]). In particular, it fits those cases when, in the reduction, one can program the simulator so that it can answer secret key queries for all the blue identities, whereas it is prepared to generate a challenge ciphertext only for red identities.

In our work we employ admissible hash functions for a similar purpose, i.e., constructing a fully-secure HIBE from a selective-secure WIBE, and in particular we adopt a definition of admissible hash functions which follows the one used by Cash *et al.* in [18]. The formal definition follows.

Let $k \in \mathbb{N}$ be the security parameter, w and λ be two values that are at most polynomial in k, and Σ be an alphabet of size s. Let $\mathcal{H} = \{H : \{0,1\}^w \to \Sigma^\lambda\}$ be a family of functions. For $H \in \mathcal{H}$, $K \in (\Sigma \cup \{*\})^\lambda$ and any $x \in \{0,1\}^w$ we define the following function which colors strings in $\{0,1\}^w$ as follows:

$$F_{K,H}(x) = \begin{cases} \mathtt{R} & \text{if } \forall i \in \{1,\ldots,\lambda\} : H(x)_i = K_i \text{ or } K_i = * \\ \mathtt{B} & \text{if } \exists i \in \{1,\ldots,\lambda\} : H(x)_i \neq K_i \end{cases}$$

For any $\mu \in \{0,\ldots,\lambda\}$, we denote with $\mathcal{K}^{(\lambda,\mu)}$ the uniform distribution over $(\Sigma \cup \{*\})^\lambda$ such that exactly μ components are not $*$. Moreover, for every $H \in \mathcal{H}$, $K \in \mathcal{K}^{(\lambda,\mu)}$, and every vector $\boldsymbol{x} \in (\{0,1\}^w)^{Q+1}$ we define the function

$$\gamma(\boldsymbol{x}) = \Pr[F_{K,H}(x_0) = \mathtt{R} \wedge F_{K,H}(x_1) = \mathtt{B} \wedge F_{K,H}(x_2) = \mathtt{B} \wedge \cdots \wedge F_{K,H}(x_Q) = \mathtt{B}].$$

Definition 2. [Admissible Hash Functions] $\mathcal{H} = \{H : \{0,1\}^w \to \Sigma^\lambda\}$ is a family of (Q, δ_{min})-*admissible hash functions* if for every polynomial $Q = Q(k)$, there exists an efficiently computable function $\mu = \mu(k)$, efficiently recognizable sets $bad_H \subseteq (\{0,1\}^w)^*$ and an inverse of a polynomial $\delta_{min} = 1/\delta(k,Q)$ such that the following properties holds:

1. For every PPT algorithm $\mathcal{A}$ that, on input $H \in \mathcal{H}$, outputs $\boldsymbol{x} \in (\{0,1\}^w)^{Q+1}$, there exists a negligible function $\epsilon(k)$ such that:

$$\mathbf{Adv}^{adm}_{\mathcal{H}}(\mathcal{A}) = \Pr[\boldsymbol{x} \in bad_H : H \leftarrow \mathcal{H}, \boldsymbol{x} \leftarrow \mathcal{A}(H)] \leq \epsilon(k)$$

2. For every $H \in \mathcal{H}$, $K \xleftarrow{\$} \mathcal{K}^{(\lambda,\mu)}$, and every vector $\boldsymbol{x} \in (\{0,1\}^w)^{Q+1} \setminus bad_H$ such that $x_0 \notin \{x_1, \ldots, x_Q\}$ we have: $\gamma(\boldsymbol{x}) \geq \delta_{min}$.

Our Transformation. Let $\mathcal{WIBE}$ be a WIBE scheme with identity space $\mathcal{ID} = \Sigma$ of size s and depth $\leq \lambda \cdot L$, and $\mathcal{H} = \{H : \{0,1\}^w \to \Sigma^\lambda\}$ be a family of functions. Then we construct the following HIBE scheme that has identity space $\mathcal{ID}' = \{0,1\}^w$ and depth at most L:

$\mathcal{HIBE}.\mathsf{Setup}$**:** run $(mpk', msk') \xleftarrow{\$} \mathcal{WIBE}.\mathsf{Setup}$ and select $H_1, \ldots, H_L \xleftarrow{\$} \mathcal{H}$. Output $mpk = (mpk', H_1, \ldots, H_L)$ and $msk = msk'$.

$\mathcal{HIBE}.\mathsf{KeyDer}(msk, \overrightarrow{ID})$**:** let $\overrightarrow{ID} = (ID_1, \ldots, ID_\ell)$ and define $\boldsymbol{I} = (H_1(ID_1), \ldots, H_\ell(ID_\ell)) \in \Sigma^{\lambda \cdot \ell}$. Output $sk_{\overrightarrow{ID}} = \mathcal{WIBE}.\mathsf{KeyDer}(msk, \boldsymbol{I})$.

$\mathcal{HIBE}.\mathsf{Enc}(mpk, \overrightarrow{ID}, \mathfrak{m})$**:** let $\overrightarrow{ID} = (ID_1, \ldots, ID_\ell)$ and define $\boldsymbol{I} = (H_1(ID_1), \ldots, H_\ell(ID_\ell)) \in \Sigma^{\lambda \cdot \ell}$. Output $C = \mathcal{WIBE}.\mathsf{Enc}(mpk, \boldsymbol{I}, m)$.

$\mathcal{HIBE}.\mathsf{Dec}(sk_{\overrightarrow{ID}}, C)$**:** return $m = \mathcal{WIBE}.\mathsf{Dec}(sk_{\overrightarrow{ID}}, C)$.

Our scheme is very simple. Essentially, the HIBE algorithm uses the algorithms of the WIBE scheme in a black-box way, where each identity component ID_i is first hashed using a function $H_i \in \mathcal{H}$. Boneh and Boyen show how to construct admissible hash functions based on collision-resistance and error-correction, and propose some concrete parameters for their instantiation (which satisfy our definition). In particular, for convenience of their construction, they consider functions that map to strings in an alphabet Σ of size $s = 2$. Here we notice that if the given WIBE has an alphabet Σ' of size $s' > 2$, then one can simply choose two values $x_1, x_2 \in \Sigma'$, set $\Sigma = \{x_1, x_2\}$, and then consider the same WIBE restricted to these two identities.

The security of our scheme follows from the following theorem, whose proof is deferred to the full version.

Theorem 3. *If $\mathcal{H} = \{H : \{0,1\}^w \to \Sigma^\lambda\}$ is a family of (Q, δ_{min})-admissible hash functions, and $\mathcal{WIBE}$ is* IND-sWID-CPA*-secure, then the scheme $\mathcal{HIBE}$ given in Section 3 is* IND-HID-CPA*-secure, where the maximum hierarchy's depth L is some fixed constant.*

Intuitively speaking, the proof of Theorem 3 proceeds by showing an algorithm $\mathcal{B}$ that plays game IND-sWID-CPA against the scheme $\mathcal{WIBE}$ and simulates the game IND-HID-CPA to an adversary $\mathcal{A}$ against $\mathcal{HIBE}$. $\mathcal{B}$ first generates the parameters for the admissible hash functions, which define the partitions $\mathtt{B}$ and $\mathtt{R}$, and then it declares the set $\mathtt{R}$ as the challenge pattern (notice that by definition of $K \in \mathcal{K}^{(\lambda,\mu)}$, $\mathtt{R}$ can be described using a pattern). In this way, all secret key queries made by $\mathcal{A}$ for identities in $\mathtt{B}$ can be forwarded by $\mathcal{B}$ to its own challenger, and the same can be done if the challenge identity chosen by $\mathcal{A}$ falls in $\mathtt{R}$. In particular, by the properties of admissible hash functions, the event that the identities of secret key queries fall in $\mathtt{B}$ and the challenge identity falls in $\mathtt{R}$ occurs with non-negligible probability. However, things are not that simple, as there may be unlucky events in which $\mathcal{B}$ is unable to simulate the right game to $\mathcal{A}$ and thus it needs to abort. As it already occurred in other

works [30,18], these events may not be independent of the adversary's view, and one solution is to force the simulator to run an expensive artificial abort step. Our proof of Theorem 3 proceeds in this way, requiring $\mathcal{B}$ to (eventually) artificially abort at the end of the simulation. Alternatively, one can extend the techniques introduced by Bellare and Ristenpart in [5] to obtain a proof of Theorem 3 which avoids the need of artificial aborts. However, this requires a slightly different definition of admissible hash functions.

Remark 4. Even though our transformation requires a WIBE scheme with $\lambda \cdot L$ levels to get a HIBE with L levels, we observe that the HIBE key derivation algorithm will use the WIBE key derivation at most L times. The point is that while L is supposed to be a constant, λ can be instead non-constant, as it is the case for known constructions of admissible hash functions, whose output length depends on the number of secret key queries made by the adversary. This might have been a problem for those WIBE schemes that do not support key derivation (delegation) for a polynomial number of levels, such as the new lattice-based scheme described in the full version of this paper.

Extensions. Our transformation easily allows for two extensions. First, it can be used to build an IBE by using a WIBE without the delegation property. Second, we show that it works also in the Continual Memory Leakage model of [15,20]. We provide a complete description of these extensions in the full version of our work.

4 Selective WIBE Schemes from Selective HIBE

In this section we investigate methodologies that allow to build a selective-pattern secure WIBE scheme starting from a HIBE which is selective-identity secure. In particular, we identify conditions under which this transformation works, and then, in the full version we will show that such conditions are satisfied by many known schemes, e.g., [7,9,30]. Then, by combining this result, i.e., a transformation from selective-identity secure HIBE to selective-pattern secure WIBE, with the result of Section 3, i.e., a conversion from selective-pattern secure WIBE to fully-secure HIBE, we obtain a methodology which allows to turn a selective-secure HIBE into a fully-secure one.

Security under Correlated Randomness. Towards this goal, our first contribution is a notion of security for HIBE schemes, called *security under correlated randomness*.The main idea can be described as follows. Assume that one is given encryptions of the same message with the same randomness but for different identities $\overrightarrow{ID}^0, \ldots, \overrightarrow{ID}^n$. Then there should be an efficient algorithm that allows to efficiently generate a new ciphertext encrypting the same message but intended to another identity $\overrightarrow{ID}' \in \mathcal{ID}' \subseteq \mathcal{ID}$. The first technical point is to delineate which is this subspace $\mathcal{ID}'$ of the identity space. So, our first contribution is to show that $\mathcal{ID}'$ follows from the differences between the identities

$\overrightarrow{ID}^0, \ldots, \overrightarrow{ID}^n$. More technically, we will show that starting from any set of identities $\overrightarrow{ID}^0, \ldots, \overrightarrow{ID}^n$ one can define a matrix Δ whose column i contains the vector which is computed as the difference between $\overrightarrow{ID}^0$ and $\overrightarrow{ID}^i$ (i.e., $\Delta^{(i)} = \overrightarrow{ID}^0 - \overrightarrow{ID}^i$). Then the identity subspace $I\mathcal{D}'$ fixed by $\overrightarrow{ID}^0, \ldots, \overrightarrow{ID}^n$ is the set of all identities that can be obtained by making affine operations over $\overrightarrow{ID}^0$ and Δ. (i.e., $\overrightarrow{ID}^0$ plus vectors obtained from integer linear combinations of vectors in Δ). Given this property, encrypting a message with the same randomness for $\overrightarrow{ID}^0, \ldots, \overrightarrow{ID}^n$ is equivalent to encrypting for the entire $I\mathcal{D}'$, that we call $\mathsf{Span}(\overrightarrow{ID}^0, \ldots, \overrightarrow{ID}^n)$. As one may guess, this is already a first step towards building a WIBE, in which the set of recipients of an encryption is actually a subspace of $I\mathcal{D}$ described by the pattern P.

Given the intuitive notion of Span described above, we define below the property for HIBE schemes that we call *Ciphertext Conversion*.

Property 1 (Ciphertext Conversion). *A HIBE scheme satisfies Ciphertext Conversion if there exists an algorithm* $\mathsf{Convert}$ *that, on input* $n+1$ *ciphertexts* $(C_0, \ldots, C_n)$ *encrypting the same message with the same randomness* r*, under identities* $(\overrightarrow{ID}^0, \ldots, \overrightarrow{ID}^n)$ *respectively, can generate a new ciphertext (encrypting the same message) intended to any* $\overrightarrow{ID} \in \mathsf{Span}(\overrightarrow{ID}^0, \ldots, \overrightarrow{ID}^n)$.

For any HIBE satisfying Property 1, the notion of selective security under correlated randomness (IND-sCR-CPA) is defined by a game which is the same as the IND-sID-CPA one except that: at the beginning the adversary chooses $n+1$ identities $\overrightarrow{ID}^0, \ldots, \overrightarrow{ID}^n$; it receives n+1 challenge ciphertexts generated using the same randomness under identities $\overrightarrow{ID}^0, \ldots, \overrightarrow{ID}^n$ respectively; it cannot ask for the secret keys of identities in $\mathsf{Span}(\overrightarrow{ID}^0, \ldots, \overrightarrow{ID}^n)$. The IND-sCR-CPA notion is parametrized by a distribution $\mathcal{R}$ on the identities $\overrightarrow{ID}^0, \ldots, \overrightarrow{ID}^n$ that can be chosen by the adversary.

We defer the interested reader to the full version of our work for more formal and precise definitions.

From HIBE Selective-Secure under Correlated Randomness to Selective-Secure WIBE. Now that we have defined the notion of selective security under correlated randomness (IND-sCR-CPA), we can show how to build a selective-pattern secure WIBE from an IND-sCR-CPA-secure HIBE. Towards this goal, let us first introduce some notation and basic definitions.

Let $I\mathcal{D} = \mathbb{Z}_q^\lambda$ be the identity space, for some $q \geq 2$ and $\lambda \geq 1$. For any pattern $P \in (I\mathcal{D} \cup \{*\})^\ell$ we define the function $(\overrightarrow{ID}^0, \ldots, \overrightarrow{ID}^n) \leftarrow F(P)$ as follows. Let $\{j_1, \ldots, j_{n'}\} = \mathrm{W}(P) \subseteq \{1, \ldots, \ell\}$ be the set of levels in which P contains $*$. Let $n = n' \cdot \lambda$, $(\overrightarrow{ID}^0, \ldots, \overrightarrow{ID}^n)$ is defined as:

$$ID_i^0 = \begin{cases} P_i & if \;\; P_i \neq * \\ 0^\lambda & if \;\; P_i = * \end{cases}$$

$$ID_{i,m}^{k+l-1} = \begin{cases} -1 & if \;\; i = j_k \wedge m = l \\ ID_{i,m}^0 & \text{otherwise} \end{cases} : \begin{array}{l} 1 \leq k \leq n',\ 1 \leq l \leq \lambda \\ 1 \leq i \leq \ell,\ 1 \leq m \leq \lambda \end{array}$$

Moreover, we let $B = \left[B^{(1)}||\cdots||B^{(\ell\lambda)}\right] \in \{0,1\}^{\ell\lambda\times\ell\lambda}$ be the canonical basis of $\mathbb{R}^{\ell\lambda}$.

The function $F(P)$ allows to specify a set of identities $(\overrightarrow{ID}^0,\ldots,\overrightarrow{ID}^n)$ such that the induced subspace $\mathsf{Span}(\overrightarrow{ID}^0,\ldots,\overrightarrow{ID}^n)$ is exactly the same subspace described by the pattern P. Intuitively, this can be seen by looking at the way the identities are defined. $\overrightarrow{ID}^0$ is equal to P on all the positions different from $*$ and 0 elsewhere. Instead each identity $\overrightarrow{ID}^i$ is such that its difference with $\overrightarrow{ID}^0$ leads to a 1 in the *single* position where they differ and 0 elsewhere. Basically, this means that the matrix Δ obtained from $F(P)$ contains a subset of vectors in B. In this way, adding linear combinations of these vectors to $\overrightarrow{ID}^0$ allows to reach identities $\overrightarrow{ID}$ such that $ID_i = P_i$ where $P_i \neq *$, while ID_i can take any value in $\mathcal{ID}$ in those positions i where $P_i = *$. Notice that the number n of such linearly independent vectors strictly depends on the number of $*$ in P. We formally show this property of $F(\cdot)$ by proving the following claim (the proof appears in the full version of our paper):

Claim 5. *For any $P \in (\mathcal{ID}\cup\{*\})^\ell$ and any $\overrightarrow{ID} \in \mathcal{ID}^\ell$ it holds $\overrightarrow{ID} \in \mathsf{Span}(F(P))$ iff $\overrightarrow{ID} \in_* P$.*

Our WIBE Scheme. Let $\mathcal{HIBE} = (\mathsf{Setup}', \mathsf{KeyDer}', \mathsf{Enc}', \mathsf{Dec}', \mathsf{Convert})$ be a HIBE scheme with identity space $\mathcal{ID} = \mathbb{Z}_q^\lambda$ (for $q \geq 2$ and $\lambda \geq 1$), and equipped with an efficient algorithm $\mathsf{Convert}$ satisfying Property 1. Then we construct the scheme $\mathcal{WIBE} = (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$ as follows.

Setup: Return the output of Setup'.

$\mathsf{KeyDer}(sk_{\overrightarrow{ID}'}, \overrightarrow{ID})$: Run $sk_{\overrightarrow{ID}} \xleftarrow{\$} \mathsf{KeyDer}'(sk_{\overrightarrow{ID}'}, \overrightarrow{ID})$ and output $sk_{\overrightarrow{ID}}$.

$\mathsf{Enc}(mpk, P, m)$: Let $(\overrightarrow{ID}^0,\ldots,\overrightarrow{ID}^n) \leftarrow F(P)$. For all $i = 0$ to n, compute $C_i \xleftarrow{\$} \mathsf{Enc}'(mpk, \overrightarrow{ID}^k, m; r)$, where r is taken at random from the randomness space of $\mathcal{HIBE}.\mathsf{Enc}$. Finally, output $C = (C_0,\ldots,C_n)$.

$\mathsf{Dec}(sk_{\overrightarrow{ID}}, C, P)$: If $\overrightarrow{ID} \notin_* P$, then output $\perp$. Otherwise, compute $(\overrightarrow{ID}^0,\ldots,\overrightarrow{ID}^n) \leftarrow F(P)$, run $C' \leftarrow \mathsf{Convert}(mpk, C_0, \overrightarrow{ID}^0, \ldots, C_n, \overrightarrow{ID}^n, \overrightarrow{ID})$ and then output $m \leftarrow \mathsf{Dec}'(sk_{\overrightarrow{ID}}, C')$.

Remark 6. We notice that our transformation assumes a HIBE scheme that works with the identities returned by our function $F(\cdot)$. This function is defined so that it assigns to the identities values P_i, 0 or -1. However, it may be the case that 0 and/or 1 are not considered valid values in some specific identity space (e.g., assume $\mathcal{ID} = \mathbb{Z}_q \setminus \{0\}$). This issue can be overcome by observing that everything still works if one takes any two different (and valid) values of the identity space, instead of 0 and 1. All we want is that when we compute the matrix Δ, if two identity components are equal, then their difference becomes 0, otherwise they lead to some value c (not necessarily 1). To see that everything works even with any constant c, observe that it is possible to consider our operations over Δ/c.

Now, we state the security of our scheme via the following theorem, whose proof can be found in the full version.

Theorem 7. *If $\mathcal{HIBE}$ satisfies Property 1 and is* IND-sCR-CPA*-secure w.r.t. $\mathcal{R} = \mathcal{ID}^{\ell\times(n+1)}$, then the scheme $\mathcal{WIBE}$ described above is correct and* IND-sWID-CPA *secure.*

A Sufficient Distribution for Building a WIBE. In the previous section, we showed that an HIBE scheme satisfying Property 1 and the notion of selective-security under correlated randomness can be transformed into a WIBE. In particular, Theorem 7 considers the most general definition where the distribution $\mathcal{R}$ is arbitrary, i.e., $\mathcal{R} = \mathcal{ID}^{\ell\times(n+1)}$. However, we observe that in order for the transformation to work, it is sufficient to consider a more restricted distribution that we call $\mathcal{R}_{WIBE}$.

Let $B = \left[B^{(1)}||\cdots||B^{(\ell\lambda)}\right] \in \{0,1\}^{\ell\lambda\times\ell\lambda}$ be the canonical basis. defined in the previous section. We define the distribution

$$\mathcal{R}_{WIBE} = \{(\overrightarrow{ID}^0,\ldots,\overrightarrow{ID}^n) : \overrightarrow{ID}^0 \in \mathbb{Z}_q^{\lambda\ell},\ \overrightarrow{ID}^i = \overrightarrow{ID}^0 + k_i \cdot B^{(j_i)},\ 1 \le i \le n,$$

$$j_i \in \{1,\ldots,\lambda\ell\}, \boldsymbol{k} \in \mathbb{Z}^n\}$$

It is interesting to observe that for any pattern P the identities obtained from $F(P)$ follow the distribution $\mathcal{R}_{WIBE}$. We show the following claim whose proof appears in the full version.

Claim 8. *For any pattern $P \in (\mathcal{ID} \cup \{*\})^\ell$ we have $F(P) \in \mathcal{R}_{WIBE}$.*

Hence, we can combine the results of Theorem 7 and Claim 8 to obtain the following Corollary.

Corollary 9. *If $\mathcal{HIBE}$ satisfies Property 1 and is secure under the* IND-sCR-CPA *notion w.r.t. $\mathcal{R}_{WIBE}$, then the scheme $\mathcal{WIBE}$ described above is correct and* IND-sWID-CPA*-secure.*

5 Lattice-Based WIBE

In this section, we give a construction of a lattice-based selectively-secure WIBE, based on the hardness of the LWE Problem [27], that very closely resembles the selectively-secure HIBE construction from [18]. In fact, the master/secret key generation and delegation procedures are exactly the same for the HIBE and the WIBE. The only difference lies in the encryption and decryption procedures; yet even there, the distinction is fairly minor. For the benefit of those readers familiar with the HIBE of [18], we present the constructions of the WIBE along with the construction of the HIBE and also try to use the same notational conventions.

Algorithms Used in Constructing the HIBE and WIBE. We now briefly describe the algorithms that were used in [18] to construct the HIBE, which we will be using in this section for constructing the WIBE.

1. $\texttt{GenBasis}(1^n, 1^m, q)$: This algorithm generates a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ (where $m = \Omega(n \log q)$) and a basis $\mathbf{S} \in \mathbb{Z}^{m \times m}$ of $\Lambda^{\perp}(\mathbf{A})$ such that the distribution of $\mathbf{A}$ is negligibly close to uniform over $\mathbb{Z}_q^{n \times m}$ and $\|\tilde{\mathbf{S}}\| = O(\sqrt{n \log q})$.
2. $\texttt{ExtBasis}(\mathbf{S}, \mathbf{A}' = \mathbf{A}||\bar{\mathbf{A}})$: This algorithm takes as input a matrix $\mathbf{A}' = \mathbf{A}||\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times (m+\bar{m})}$ and a matrix $\mathbf{S} \in \mathbb{Z}^{m \times m}$, which is basis of $\Lambda^{\perp}(\mathbf{A})$, and outputs a matrix $\mathbf{S}' \in \mathbb{Z}^{(m+\bar{m}) \times (m+\bar{m})}$ that is a basis of $\Lambda^{\perp}(\mathbf{A}')$ such that $\|\tilde{\mathbf{S}}\| = \|\widetilde{\mathbf{S}'}\|$.
3. $\texttt{SampleD}(\mathbf{S}, \mathbf{A}, \mathbf{y}, s)$: This algorithm takes as input a basis $\mathbf{S} \in \mathbb{Z}^{m \times m}$ of the lattice $\Lambda^{\perp}(\mathbf{A})$, a vector $\mathbf{y} \in \mathbb{Z}_q^n$, and a real number $s \geq \|\tilde{\mathbf{S}}\| \cdot \omega(\sqrt{\log n})$ and outputs a vector $\mathbf{z} \sim D_{\Lambda_{\mathbf{y}}^{\perp}(\mathbf{A}), s}$.
4. $\texttt{RandBasis}(\mathbf{S}, s)$: This algorithm takes as input an $m \times m$ lattice basis $\mathbf{S}$ and a real number $s \geq \|\tilde{\mathbf{S}}\| \cdot \omega(\sqrt{\log n})$, and outputs a basis $\mathbf{S}'$ of the same lattice such that $\|\mathbf{S}'\| \leq s\sqrt{m}$. Furthermore, if $\mathbf{S_0}, \mathbf{S_1}$ are bases of the same lattice and $s > \max\{\|\widetilde{\mathbf{S_0}}\|, \|\widetilde{\mathbf{S_1}}\|\}$, then the distributions of $\texttt{RandBasis}(\mathbf{S_0}, s)$ and $\texttt{RandBasis}(\mathbf{S_1}, s)$ are statistically close.

The Lattice-Based WIBE Scheme. We now describe the master key generation, key derivation, encryption and decryption algorithms of our WIBE scheme. For any distribution χ over $\mathbb{Z}$, and any vector $\mathbf{x} \in \mathbb{Z}_q^n$ let $\texttt{Noisy}_{\chi}(\mathbf{x})$ be the distribution obtained by first creating a vector $\mathbf{y} \in \mathbb{Z}^n$ each of whose coordinates is independently sampled according to χ, and then outputting $\mathbf{x} + \mathbf{y} \bmod q$.

Master Key Generation. We assume that the identities are of the form $\{0,1\}^t$, for all $t \leq L$. The generation of the master public and secret keys is done exactly in the same fashion in the HIBE and in the WIBE. The WIBE is parametrized by the integers n, m, q where n is the security parameter, m is an integer of size $\Omega(n \log q)$ and q is some prime whose size is related to the number of allowable key derivations, and is discussed in Section 5. We first run the $\texttt{GenBasis}(1^n, 1^m, q)$ procedure to obtain a matrix $\mathbf{A}_0 \in \mathbb{Z}_q^{n \times m}$ and a basis $\mathbf{S}_0 \in \mathbb{Z}^{m \times m}$ of $\Lambda^{\perp}(\mathbf{A})$. Then for all $(i, j) \in \{0,1\} \times \{1, \ldots, L\}$, we generate a uniformly random matrix $\mathbf{A}_j^{(i)} \in \mathbb{Z}_q^{n \times m}$, and choose a uniformly-random $\mathbf{y} \in \mathbb{Z}_q^n$. The master public key is

$$\left[\mathbf{A}_0, \mathbf{A}_1^{(0)}, \mathbf{A}_1^{(1)}, \ldots, \mathbf{A}_L^{(0)}, \mathbf{A}_L^{(1)}, \mathbf{y}\right],$$

and the master secret key is $\mathbf{S}_0$.

Key Derivation. The key derivation procedure is again performed exactly the same for the HIBE and the WIBE. The public key of identity $id = (id_1, \ldots, id_t)$ is $(\mathbf{A}_{id}, \mathbf{y})$, where $\mathbf{A}_{id} = \mathbf{A}_0 \| \mathbf{A}_1^{(id_1)} \| \ldots \| \mathbf{A}_t^{(id_t)}$. The secret key of user id is $(\mathbf{S}_{id}, \mathbf{x}_{id})$ where $\mathbf{S}_{id}$ is a "short" basis of the lattice $\Lambda^{\perp}(\mathbf{A}_{id})$ and $\mathbf{x}_{id}$ is a "short" vector satisfying $\mathbf{A}_{id}^T \mathbf{x}_{id} = \mathbf{y}$. The matrix $\mathbf{S}_{id}$ will be used for delegation, while the vector $\mathbf{x}_{id}$ will be used for decryption.

If a user with $id = (id_1, \ldots, id_t)$ would like to generate a secret key for a user $id' = (id_1, \ldots, id_t, id_{t+1}, \ldots, id_{t'})$ whose public key is $(\mathbf{A}_{id'} = \mathbf{A}_{id} || \bar{\mathbf{A}}, \mathbf{y})$, where

$\bar{\mathbf{A}} = \mathbf{A}_{t+1}^{(id_{t+1})} \| \ldots \| \mathbf{A}_{t'}^{(id_{t'})}$, he computes the following:

$$\mathbf{S}_{id'} \leftarrow \texttt{RandBasis}(\texttt{ExtBasis}(\mathbf{S}_{id}, \mathbf{A}_{id'}), s)$$

$$\mathbf{x}_{id'} \leftarrow \texttt{SampleD}(\texttt{ExtBasis}(\mathbf{S}_{id}, \mathbf{A}_{id'}), \mathbf{A}_{id'}, \mathbf{y}, s)$$

where $s \geq \|\widetilde{\mathbf{S}_{id}}\| \cdot \omega(\sqrt{\log n})$. We point out that with every key derivation, the value of $\|\widetilde{\mathbf{S}_{id}}\|$ increases by a factor of $\tilde{O}(\sqrt{n})$. When the norm of the secret key gets too large, decryption becomes impossible, and so, just like in [18], it is important to adjust the ratio of the size of the secret key $\mathbf{S}_0$ and the prime q based on how many levels of delegations one wishes to have. With each level of delegation increasing the norm of the user id by a factor of $\tilde{O}(\sqrt{n})$, the ratio between $\|\widetilde{\mathbf{S}_0}\|$ and q should be on the order of $\sqrt{n}^d$, where d is the maximum allowable levels in the hierarchy. Since the $\texttt{LWE}_{n,q,\chi}$ problem becomes easier as q gets larger (and the distribution χ stays the same), there is a trade-off between security and the maximum number of delegation levels. We direct the reader to [18] for the precise parameters.

Encryption and Decryption. In the HIBE, encryption of a message $\kappa \in \{0,1\}$ is performed to identity $id = (id_1, \ldots, id_t)$ by picking a random $\mathbf{r} \in \mathbb{Z}_q^n$ and outputting the pair $(\mathbf{u}_{id}, v) \in \mathbb{Z}_q^{m(t+1)+1}$, where

$$(\mathbf{u}_{id}, v) = \left(\texttt{Noisy}_\chi\left(\mathbf{A}_{id}^T \mathbf{r}\right), \texttt{Noisy}_\chi\left(\mathbf{y}^T \mathbf{r} + \kappa \cdot \lfloor q/2 \rfloor\right)\right)$$

where

$$\mathbf{A}_{id} = \mathbf{A}_0 \| \mathbf{A}_1^{(id_1)} \| \ldots \| \mathbf{A}_t^{(id_t)} \tag{1}$$

and χ is some "narrow" distribution such that the $\texttt{LWE}_{n,q,\chi}$ problem is hard.

The decryption of the HIBE ciphertext by the identity $id = (id_1, \ldots, id_t)$ is performed as follows: for a ciphertext $(\mathbf{u}_{id}, v)$ and secret key $\mathbf{x}_{id}$, the algorithm computes $v - \mathbf{x}_{id}^T \mathbf{u}_{id} \bmod q$ and outputs 0 if the result is closer to 0 than to $q/2$, and outputs 1 otherwise.

In our WIBE, encryption is defined in essentially the same way as in the HIBE. To encrypt to a pattern $pat = (pat_1, \ldots, pat_t) \in \{0, 1, *\}^t$, we pick a random $\mathbf{r} \in \mathbb{Z}_q^n$, define

$$\mathbf{A}_{pat} = \mathbf{A}_0 \| \mathbf{A}_1^{(pat_1)} \| \ldots \| \mathbf{A}_t^{(pat_t)} \tag{2}$$

where $\mathbf{A}_i^* := \mathbf{A}_i^{(0)} \| \mathbf{A}_i^{(1)}$, and output the pair $(\mathbf{u}_{pat}, v) \in \mathbb{Z}_q^{m(t+t_*+1)+1}$ (where t_* is the number of $*$ in the pattern pat),

$$(\mathbf{u}_{pat}, v) = \left(\texttt{Noisy}_\chi(\mathbf{A}_{pat}^T \mathbf{r}), \texttt{Noisy}_\chi\left(\mathbf{y}^T \mathbf{r} + \kappa \cdot \lfloor q/2 \rfloor\right)\right).$$

Notice that instead of the matrix $\mathbf{A}_{pat}$ being $n \times mt$ as in the HIBE, it can be as large as $n \times 2mt$ because every position pat_i that contains the wildcard $*$ results in the concatenation of both $\mathbf{A}_i^{(0)}$ and $\mathbf{A}_i^{(1)}$ into the matrix $\mathbf{A}_{pat}$. Therefore the ciphertext of the WIBE could be twice as large as the HIBE ciphertext.

The decryption procedure of the WIBE is also very similar to that of the HIBE. For every $id = (id_1, \ldots, id_t) \in \{0,1\}^t$, the matrix $\mathbf{A}_{pat}$ contains the

matrix $\mathbf{A}_{id}$, where $\mathbf{A}_{id}$ is defined as in (1). Therefore, since we know $\mathbf{u}_{pat} = \mathtt{Noisy}_{\chi}(\mathbf{A}_{pat}^T\mathbf{r})$, we can retrieve from it $\mathbf{u}_{id} = \mathtt{Noisy}_{\chi}(\mathbf{A}_{id}^T\mathbf{r})$. And now, using the secret key $\mathbf{x}_{id}$, the user can decrypt the ciphertext $(\mathbf{u}_{id}, v)$ the same way as in the HIBE scheme by computing $v - \mathbf{x}_{id}^T\mathbf{u}_{id} \bmod q$ and outputting 0 if the result is closer to 0 than to $q/2$, and 1 otherwise.

Security. The security proof of our scheme, which can be found in the full version of this paper, is a simple adaptation of the HIBE security proof in [18].

Theorem 10. *Given an adversary $\mathcal{A}$ who breaks the WIBE with parameters n, m, q allowing d key derivations, there exists an algorithm $\mathcal{S}$ that solves the $\mathtt{LWE}_{n,q,\chi}$ problem where $q > \sigma \cdot n^{d/2} \cdot poly(n)$ where σ is the standard deviation of the distribution χ and $poly(n)$ is some fixed polynomial function in n.*

6 Future Directions

First, in its most general form (i.e., without restrictions on $\mathcal{R}$), our notion of security under correlated randomness gives a generic methodology for encrypting messages to sets S of recipients that are defined by $\mathsf{Span}(\overrightarrow{ID}^0, \ldots, \overrightarrow{ID}^n)$. In this sense, a WIBE can be seen as a special case of this notion in which the recipients' sets always have a fixed form specified by the pattern P, i.e., $S = \mathsf{Span}(F(P))$. However, one may think of a more general notion in which these sets can have a more "irregular" form that we can express using a set of identities $(\overrightarrow{ID}^0, \ldots, \overrightarrow{ID}^n)$ and its Span.

Since we were mostly interested in building WIBE schemes in this work, we considered security under correlated randomness w.r.t. the distribution $\mathcal{R}_{WIBE}$. However, as a future direction, it would be interesting to explore whether there exist HIBE schemes that are IND-sCR-CPA-secure according to the most generic notion, i.e., without any restriction on $\mathcal{R}$. Perhaps more interestingly, the resulting primitive could be seen as the dual version of the notion of Spatial Encryption proposed by Boneh and Hamburg in [12]. In Spatial Encryption, ciphertexts are associated to points in $\mathbb{Z}_p^\ell$, while secret keys correspond to affine subspaces of $\mathbb{Z}_p^\ell$. In this setting, a ciphertext for $x \in \mathbb{Z}_p^\ell$ can be decrypted by any secret key for $W \subseteq \mathbb{Z}_p^\ell$ as long as $x \in W$. In contrast, our new notion would consider ciphertexts that are associated to affine subspaces of $\mathcal{ID}^\ell$.

As a second direction, it would be interesting to investigate whether our techniques can be applied to other cryptographic primitives. Indeed, the problem of selective vs. full security has already been considered in the context of other cryptographic notions, such as attribute-based encryption or verifiable random functions (VRFs). In the particular case of VRFs, finding a fully secure scheme has been a long standing open problem until the very recent works by Hohenberger and Waters [23] and by Boneh *et al.* [14]. In fact, both of these works can be seen as obtaining a fully secure VRF from a selective secure one. While the work of Boneh *et al.* explicitly builds a selective-secure VRF and then turns it into a fully secure one, the work of Hohenberger and Waters can be interpreted as a fully secure version of the selective-secure VRF scheme of Abdalla *et al.* [2].

Acknowledgments. This work was supported in part by the European Research Council, in part by the European Commission through the ICT Program under Contract ICT-2007-216676 ECRYPT II, and in part by the French ANR-10-SEGI-15 PRINCE Project.

References

1. Abdalla, M., Catalano, D., Dent, A.W., Malone-Lee, J., Neven, G., Smart, N.P.: Identity-Based Encryption Gone Wild. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006, Part II. LNCS, vol. 4052, pp. 300–311. Springer, Heidelberg (2006)
2. Abdalla, M., Catalano, D., Fiore, D.: Verifiable Random Functions from Identity-Based Key Encapsulation. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 554–571. Springer, Heidelberg (2009)
3. Agrawal, S., Boneh, D., Boyen, X.: Efficient Lattice (H)IBE in the Standard Model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010)
4. Agrawal, S., Boneh, D., Boyen, X.: Lattice Basis Delegation in Fixed Dimension and Shorter-Ciphertext Hierarchical IBE. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 98–115. Springer, Heidelberg (2010)
5. Bellare, M., Ristenpart, T.: Simulation without the Artificial Abort: Simplified Proof and Improved Concrete Security for Waters' IBE Scheme. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 407–424. Springer, Heidelberg (2009)
6. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Ashby, V. (ed.) ACM CCS 1993, pp. 62–73. ACM Press (November 1993)
7. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
8. Boneh, D., Boyen, X.: Secure Identity Based Encryption Without Random Oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443–459. Springer, Heidelberg (2004)
9. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical Identity Based Encryption with Constant Size Ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
10. Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM Journal on Computing 36(5), 1301–1328 (2007)
11. Boneh, D., Franklin, M.K.: Identity based encryption from the Weil pairing. SIAM Journal on Computing 32(3), 586–615 (2003)
12. Boneh, D., Hamburg, M.: Generalized Identity Based and Broadcast Encryption Schemes. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 455–470. Springer, Heidelberg (2008)
13. Boneh, D., Katz, J.: Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity-Based Encryption. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 87–103. Springer, Heidelberg (2005)
14. Boneh, D., Montgomery, H.W., Raghunathan, A.: Algebraic pseudorandom functions with improved efficiency from the augmented cascade. In: Al-Shaer, E., Keromytis, A.D., Shmatikov, V. (eds.) ACM CCS 2010, pp. 131–140. ACM Press (October 2010)

15. Brakerski, Z., Kalai, Y.T., Katz, J., Vaikuntanathan, V.: Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In: 51st FOCS, pp. 501–510. IEEE Computer Society Press (2010)
16. Canetti, R., Halevi, S., Katz, J.: A Forward-Secure Public-Key Encryption Scheme. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 255–271. Springer, Heidelberg (2003)
17. Canetti, R., Halevi, S., Katz, J.: Chosen-Ciphertext Security from Identity-Based Encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
18. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai Trees, or How to Delegate a Lattice Basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010)
19. Cocks, C.: An Identity Based Encryption Scheme Based on Quadratic Residues. In: Honary, B. (ed.) Cryptography and Coding 2001. LNCS, vol. 2260, pp. 360–363. Springer, Heidelberg (2001)
20. Dodis, Y., Haralambiev, K., López-Alt, A., Wichs, D.: Cryptography against continuous memory attacks. In: 51st FOCS, pp. 511–520. IEEE Computer Society Press (2010)
21. Gentry, C., Halevi, S.: Hierarchical Identity Based Encryption with Polynomially Many Levels. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 437–456. Springer, Heidelberg (2009)
22. Gentry, C., Silverberg, A.: Hierarchical ID-Based Cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002)
23. Hohenberger, S., Waters, B.: Constructing Verifiable Random Functions with Large Input Spaces. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 656–672. Springer, Heidelberg (2010)
24. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010)
25. Lewko, A., Waters, B.: New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (2010)
26. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Mitzenmacher, M. (ed.) 41st ACM STOC, pp. 333–342. ACM Press (May/June 2009)
27. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC, pp. 84–93. ACM Press (May 2005)
28. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
29. Waters, B.: Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)
30. Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)

Circular and KDM Security for Identity-Based Encryption

Jacob Alperin-Sheriff and Chris Peikert*

Georgia Institute of Technology

Abstract. We initiate the study of security for key-dependent messages (KDM), sometimes also known as "circular" or "clique" security, in the setting of identity-based encryption (IBE). Circular/KDM security requires that ciphertexts preserve secrecy even when they encrypt messages that may depend on the secret keys, and arises in natural usage scenarios for IBE.

We construct an IBE system that is circular secure for affine functions of users' secret keys, based on the learning with errors (LWE) problem (and hence on worst-case lattice problems). The scheme is secure in the standard model, under a natural extension of a selective-identity attack. Our three main technical contributions are (1) showing the circular/KDM-security of a "dual"-style LWE public-key cryptosystem, (2) proving the hardness of a version of the "extended LWE" problem due to O'Neill, Peikert and Waters (CRYPTO'11), and (3) building an IBE scheme around the dual-style system using a novel lattice-based "all-but-d" trapdoor function.

1 Introduction

Traditional notions of secure encryption, starting with semantic (or IND-CPA) security [22], assume that the plaintext messages do not depend on the secret decryption key (except perhaps indirectly, via the public key or other ciphertexts). In many settings, this may fail to be the case. One obvious scenario is, of course, careless or improper key management: for example, some disk encryption systems end up encrypting the symmetric secret key itself (or a derivative) and storing it on disk. However, there are also situations in which key-dependent messages are used as an integral part of an cryptosystem. For example, in their anonymous credential system, Camenisch and Lysyanskaya [13] use a cycle of key-dependent messages to discourage users from delegating their secret keys. More recently, Gentry's "bootstrapping" technique for obtaining a fully homomorphic cryptosystem [19] encrypts a secret key under the corresponding public key in order to support unbounded homomorphism; the cryptosystem therefore

* This material is based upon work supported by the National Science Foundation under CAREER Award CCF-1054495 and by the Alfred P. Sloan Foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the Sloan Foundation.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 334–352, 2012.

needs to be "circular secure." More generally, a system that potentially reveals encryptions of any party's secret key under any user's public key needs to be "clique" or "key-dependent message" (KDM) secure. This notion allows for proving formal symbolic soundness of cryptosystems having complexity-based security proofs [1].

Since Boneh *et al.*'s breakthrough work [9] giving a KDM-secure encryption scheme, in the standard model, from the Decision Diffie-Hellman assumption, a large number of results (mostly positive) have been obtained regarding circular- and KDM-secure encryption [23, 5, 6, 10, 4, 26, 11, 12]. However, all these works have dealt only with the symmetric or public-key settings; in particular, the question of circular/KDM security for *identity-based* cryptography has not yet been considered. Recall that in identity-based encryption [35], any string can serve as a public key, and the corresponding secret keys are generated and administered by a trusted Private Key Generator (PKG).

Circular Security for IBE. In this work we define and construct a circular/KDM-secure identity-based encryption (IBE) scheme. KDM security is well-motivated by some natural usage scenarios for IBE, as we now explain.

Recall that identity-based encryption gives a natural and lightweight solution to revocation, via expiring keys. The lifetime of the cryptosystem is divided into time periods, or "epochs." An identity string consists of a user's "true" identity (e.g., name) concatenated with an epoch; when encrypting, one uses the identity for the current epoch. To support revocation, the PKG gives out a user's secret key only for the current epoch, and only if the user is still authorized to be part of the system. Therefore, a user's privileges can be revoked by simply refusing to give out his secret key in future epochs; in particular, this revocation is transparent to the encrypter.

The above framework makes it necessary for users to periodically get new secret keys from the PKG, confidentially. The most lightweight method, which eliminates the need for the user to prove his identity every time, is simply for the PKG to encrypt the new secret key under the user's identity for the previous epoch. This can be proved secure, assuming the underlying IBE is CPA-secure, *as long as there are no cycles of encrypted keys*. However, if a user deletes or loses an old secret key and wants to decrypt a ciphertext from the corresponding epoch, it would be natural for the authority to provide the old secret key encrypted under the user's identity for the current epoch. But because the current secret key has also been encrypted (perhaps via a chain of encryptions) under the old identity, this may be unsafe unless the IBE is KDM-secure.

1.1 Our Contributions

As already mentioned, in this work we define a form of circular/KDM security for identity-based encryption, and give a standard-model construction based on the learning with errors (LWE) problem, hence on worst-case lattice problems via the reductions of [34, 32].

As in prior positive results on circular security [9, 5, 10], our definition allows the adversary to obtain encrypted "key cliques" for affine functions of the secret keys. More precisely, for any tuple of identities $(id_1, \ldots, id_d)$, the attacker may adaptively query encryptions of $f(sk_{id_i})$ under any of the identities id_j, where f is any affine function over the message space, and each sk_{id_i} is a secret key for identity id_i. Our attack model is in the style of a "selective identity" attack, wherein the adversary must declare the target identities $id_1, \ldots, id_d$ (but not the functions f) before seeing the public parameters of the scheme. While this is not the strongest security notion we might hope for, it appears to at least capture the main security requirements of the scenarios described above, because encrypted key cycles are only ever published for the same "real-world" identity at different time epochs. Therefore, just as in a standard selective-identity attack for IBE, the adversary is actually limited to attacking just a single real-world identity, on a set of d epochs (which could, for example, include all valid epochs). We also point out that by a routine hybrid argument, security also holds when attacking a *disjoint* collection of identity cliques (that are named before seeing the public parameters).

Our IBE construction is built from two components, both of which involve some novel techniques. First, we give an LWE-based *public-key* cryptosystem that is clique secure (even for an *unbounded* number of users) for affine functions, and is suitable for embedding into an IBE like the one of [20]. Second, we construct a lattice-based "all-but-d" trapdoor function that serves as the main foundation of the IBE. We elaborate on these two contributions next.

Clique-Secure Public-Key Cryptosystem. We first recall that Applebaum *et al.* [5] showed that a variant of Regev's so-called "primal" LWE cryptosystem [34] is clique secure for affine functions. Unfortunately, this primal-type system does not seem suitable as the foundation for identity-based encryption using the tools of [20]. The first reason is that the proof of clique security from [5] needs the users' public keys to be completely independent, rather than incorporating a shared random string (e.g., the public parameters in an IBE system). The second reason is a bit more technical, and is already described in [20]: in primal-style systems, the user-specific public keys are exponentially sparse pseudorandom values (with unique secret keys), and it is difficult to design an appropriate mapping from identities to valid public keys that actually admit usable underlying secret keys.

Therefore, we first need to obtain clique security for a so-called "dual"-type cryptosystem (using the terminology from [20]), in which *every* syntactically valid public key has a functional underlying secret key (actually, many such secret keys) that can be extracted by the PKG. It turns out that achieving this goal is quite a bit more technically challenging than it was for the "primal"-style schemes. This is primarily because the KDM-secure scheme from [5] (like the earlier one from [9]) has the nice property that given the public key alone, one can efficiently generate *statistically well-distributed* encryptions of the secret key (without knowing the corresponding encryption randomness). This immediately implies circular security for "self-loops," and clique security follows from a couple of other related techniques.

Unfortunately, this nice statistical property on ciphertexts does not seem attainable for dual-style LWE encryption, because now valid ciphertexts are exponentially sparse and hard to generate without knowing the underlying encryption randomness. In addition, because the adversary may obtain an *unbounded* number of key-dependent ciphertexts, we also cannot rely on any statistical entropy of the secret key conditioned on the public key, as is common in the security proofs of most dual-style cryptosystems.

We resolve the above issues by relying on computational assumptions twice in our security proof, first when changing the way that challenge ciphertexts are produced (i.e., by using knowledge of the secret key), and then again when changing the form of the public key. Notably, unlike prior works (e.g., [17, 31]) in which ciphertexts in intermediate games are created by "encrypting with an (information theoretically revealed) secret key," we are able to avoid the use of super-polynomially large noise to "overwhelm" the slight statistical difference between the two ways of generating ciphertexts. This lets us prove security under fully polynomial lattice/LWE assumptions, i.e., those involving a polynomially bounded modulus q and inverse error rate for the LWE problem, and therefore polynomial approximation factors for worst-case lattice problems. We do so by proving the hardness of a version of the *extended*-LWE problem, as defined and left open by the recent work of [31]. We believe that this result should be useful in several other contexts as well.

All-but-d trapdoor functions. We use the clique-secure cryptosystem described above as the foundation for a clique-secure IBE. To make the cryptosystem identity-based, as in [20] we need to embed a "strong" trapdoor into the public parameters so that the PKG can extract a secret key for any identity. Here we use the ideas behind the tag-based algebraic construction of [2], and follow the somewhat simpler and more efficient realization recently due to [28]. We remark that these trapdoor constructions are well-suited to security definitions in which the adversary attacks a *single* tag, because the trapdoor can be "punctured" (i.e., made useless for extracting secret keys, and useful for embedding an LWE challenge) at exactly one predetermined tag. Unfortunately, this does not appear to be sufficient for our purposes, because in the clique security game, the adversary is attacking d identities at once and can obtain challenge ciphertexts under all of them.

To resolve the insufficiency of a single puncture, we extend the trapdoor constructions of [2, 28] so that it is possible to puncture the trapdoor at up to d arbitrary, prespecified tags. To accomplish this, we show how to statistically hide in the public key a degree-d polynomial $f(\cdot)$ over a certain ring $\mathcal{R}$, so that $f(id_i) = 0$ for all the targeted tags (identities) id_i, while $f(id)$ is a unit in $\mathcal{R}$ (i.e., is invertible) for all other identities. The d components of the public key can be combined so as to homomorphically evaluate f on any desired tag. The underlying trapdoor is punctured exactly on tags id where $f(id) = 0$, and is effective for inversion whenever $f(id)$ is a unit in $\mathcal{R}$. Our construction is analogous to the one of [15] in the setting of prime-order groups with bilinear pairings (where arithmetic "in the exponent" happens in a field), and the all-but-d lossy

trapdoor functions of [24]. However, since lattice-based constructions do not work with fields or rings like $\mathbb{Z}_N$, we instead use techniques from the literature on secret sharing over groups and modules, e.g., [16, 18].

We remark that, for technical reasons relating to the number of "hints" for which we can prove the hardness of the extended-LWE problem, we have not been able to prove the KDM-security of our IBE under fully polynomial assumptions (as we were able to do for our basic public-key cryptosystem). We instead rely on the conjectured hardness of LWE for a slightly super-polynomial modulus q and inverse error rate $1/\alpha$, which translates via known reductions [34, 32] to the conjectured hardness of worst-case lattice problems for slightly super-polynomial approximation factors, against slightly super-polynomial-time algorithms. Known lattice algorithms are very far from disproving such conjectures.

2 Preliminaries

We denote the real numbers by $\mathbb{R}$ and the integers by $\mathbb{Z}$. For a positive integer d, we use $[d]$ to denote the set $\{1, \ldots, d\}$. We denote vectors over $\mathbb{R}$ and $\mathbb{Z}$ with lower-case bold letters (e.g. $\mathbf{x}$), and matrices by upper-case bold letters (e.g. $\mathbf{A}$). We say that a function is *negligible*, written $\mathrm{negl}(n)$, if it vanishes faster than the inverse of any polynomial in n. The *statistical distance* between two distributions X, Y over a finite or countable set D is $\Delta(X, Y) = \frac{1}{2}\sum_{w \in D} |X(w) - Y(w)|$. Statistical distance is a metric, and in particular obeys the triangle inequality. Let $\{X_n\}$ and $\{Y_n\}$ be ensembles of random variables indexed by the security parameter n. We say that X and Y are *statistically close* if $\Delta(X_n, Y_n) = \mathrm{negl}(n)$. For a matrix $\mathbf{X} \in \mathbb{R}^{n \times k}$, the *largest singular value* (also known as the *spectral norm*) of $\mathbf{X}$ is defined as $s_1(\mathbf{X}) = \max_{\|\mathbf{u}\|=1} \|\mathbf{X}\mathbf{u}\|$.

2.1 Lattices and Gaussians

A (full-rank) m-dimensional *integer lattice* Λ is an additive subgroup of $\mathbb{Z}^m$ with finite index. This work is concerned with the family of integer lattices whose cryptographic importance was first demonstrated by Ajtai [3]. For integers $n \geq 1$, modulus $q \geq 2$, an m-dimensional lattice from this family is specified by an "arity check" matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$:

$$\Lambda^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{0} \in \mathbb{Z}_q^n\} \subseteq \mathbb{Z}^m.$$

For any $\mathbf{y}$ in the subgroup of $\mathbb{Z}_q^n$ generated by the columns of $\mathbf{A}$, we also define the coset

$$\Lambda_{\mathbf{y}}^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{y} \bmod q\} = \Lambda^{\perp}(\mathbf{A}) + \bar{\mathbf{x}},$$

where $\bar{\mathbf{x}} \in \mathbb{Z}^m$ is an arbitrary solution to $\mathbf{A}\bar{\mathbf{x}} = \mathbf{y}$.

We briefly recall Gaussian distributions over lattices (for more details see [29, 20]). For $s > 0$ and dimension $m \geq 1$, the Gaussian function $\rho_s : \mathbb{R}^m \to (0, 1]$ is defined as $\rho_s(\mathbf{x}) = \exp(-\pi\|\mathbf{x}\|^2/s^2)$. For a coset $\Lambda + \mathbf{c}$ of a lattice Λ, the *discrete*

Gaussian distribution $D_{\Lambda+\mathbf{c},s}$ (centered at zero) assigns probability proportional to $\rho_s(\mathbf{x})$ to each vector in the coset, and probability zero elsewhere.

We will need a few standard concepts and facts about discrete Gaussians over lattices. First, for $\epsilon > 0$ the *smoothing parameter* [29] $\eta_\epsilon(\Lambda)$ of an n-dimensional lattice is a positive real value. We will not need its precise definition, which depends on the notion of the *dual* lattice, but only recall the few relevant facts that we need; for details, see, e.g., [29, 20, 28].

Lemma 1. *Let $m \geq Cn \lg q$ for some constant $C > 1$.*

1. *For any $\omega(\sqrt{\log n})$ function, we have $\eta_\epsilon(\mathbb{Z}^n) \leq \omega(\sqrt{\log n})$ for some negligible $\epsilon(n) = \mathrm{negl}(n)$.*
2. *With all but $\mathrm{negl}(n)$ probability over the uniformly random choice of $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the following holds: For $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,r}$ where $r = \omega(\sqrt{\log n})$, the distribution of $\mathbf{y} = \mathbf{Ae} \bmod q$ is within $\mathrm{negl}(n)$ statistical distance of uniform, and the conditional distribution of $\mathbf{e}$ given $\mathbf{y}$ is $D_{\Lambda_\mathbf{y}^\perp(\mathbf{A}),r}$.*
3. *For any m-dimensional lattice Λ, any $\mathbf{c} \in \mathbb{Z}^m$, and any $r \geq \eta_\epsilon(\Lambda)$ where $\epsilon(n) = \mathrm{negl}(n)$, we have $\|D_{\Lambda+\mathbf{c},r}\| \leq r\sqrt{m}$ with all but $\mathrm{negl}(n)$ probability. In addition, for $\Lambda = \mathbb{Z}$ we have $|D_{\mathbb{Z},r}| \leq r \cdot \omega(\sqrt{\log n})$ except with $\mathrm{negl}(n)$ probability.*
4. *For any $r > 0$, and for $\mathbf{R} \leftarrow D_{\mathbb{Z},r}^{n\times k}$, we have $s_1(\mathbf{R}) \leq r \cdot O(\sqrt{n} + \sqrt{k})$ except with $\mathrm{negl}(n)$ probability.*

Lemma 2. *For any real number $r = \omega(\sqrt{\log n})$ and $c \in \mathbb{Z}$, the statistical distance between $D_{\mathbb{Z},r}$ and $c + D_{\mathbb{Z},r}$ is $O(|c|/r)$.*

2.2 Trapdoors for Lattices

We recall the efficient trapdoor construction and associated sampling algorithm of Micciancio and Peikert [28]. This construction uses a universal public "gadget" matrix $\mathbf{G} \in \mathbb{Z}_q^{n\times w}$ for which there is an efficient discrete Gaussian sampling algorithm for any parameter $r \geq \omega(\sqrt{\log n}) \geq \eta_\epsilon(\Lambda^\perp(\mathbf{G}))$ (for some $\epsilon(n) = \mathrm{negl}(n)$), i.e., an algorithm that, given any $\mathbf{y} \in \mathbb{Z}_q^n$ and r, outputs a sample from $D_{\Lambda_\mathbf{y}^\perp(\mathbf{G}),r}$. For concreteness, as in [28] we take $\mathbf{G} = \mathbf{I}_n \otimes [1, 2, 4, \ldots, 2^{k-1}] \in \mathbb{Z}_q^{n\times nk}$ for $k = \lceil \lg q \rceil$.

Following [28], we say that an integer matrix $\mathbf{R} \in \mathbb{Z}^{(m-n)\times w}$ is a "strong" trapdoor with tag H for $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ if $\mathbf{A}\left[\begin{smallmatrix}\mathbf{R}\\ \mathbf{I}\end{smallmatrix}\right] = H(\mathbf{G})$ for some efficiently computable and invertible linear transformation H over $\mathbb{Z}_q^n$, which is applied column-wise to $\mathbf{G}$. Equivalently, in place of $H(\mathbf{G})$ we may write $\mathbf{H} \cdot \mathbf{G}$ for some invertible matrix $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$, but in our constructions it will be more natural to work with invertible linear transformations, without explicitly referring to the matrices that represent them.

Lemma 3 ([28, Theorem 5.1]). *Let $\mathbf{R}$ be a strong trapdoor for $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$. There is an efficient randomized algorithm that, given $\mathbf{R}$, any $\mathbf{u} \in \mathbb{Z}_q^n$, and any $r \geq s_1(\mathbf{R}) \cdot \omega(\sqrt{\log n}) \geq \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$ (for some $\epsilon(n) = \mathrm{negl}(n)$), samples from a distribution within $\mathrm{negl}(n)$ distance of $D_{\Lambda_\mathbf{u}^\perp(\mathbf{A}),r}$.*

2.3 Learning With Errors

The *learning with errors* (LWE) problem is parameterized by a dimension $n \geq 1$, an integer modulus $q \geq 2$ and an error distribution χ over $\mathbb{Z}$ (or its induced distribution over $\mathbb{Z}_q$). For a vector $\mathbf{s} \in \mathbb{Z}_q^n$, the distribution $A_{\mathbf{s},\chi}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is sampled by choosing $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random and outputting $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + x)$, where $x \leftarrow \chi$.

The search version of LWE is to recover an arbitrary $\mathbf{s}$ given oracle access to $A_{\mathbf{s},\chi}$. The decision version of LWE is to distinguish, with non-negligible advantage, between samples from $A_{\mathbf{s},\chi}$ for uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$ and uniformly random samples from $\mathbb{Z}_q^n \times \mathbb{Z}_q$. There are search-to-decision reductions for LWE for a variety of moduli q and parameter conditions ([34, 32, 5, 27, 28]). Of particular importance to us are the reductions from [5, 28] for $q = p^e$, where p is prime, $e \geq 1$ is an integer, and $\Pr_{\mathbf{x} \leftarrow \chi}[|\mathbf{x}| \geq p/2] = \mathrm{negl}(n)$. The reductions runs in time polynomial in n, p, and e.

For error distribution $\chi = D_{\mathbb{Z},\alpha q}$, where $\alpha q \geq 2\sqrt{n}$, the search version of LWE is at least as hard as *quantumly* approximating certain worst-case problems on n-dimensional lattices to within $\tilde{O}(n/a)$ factors [34]; for certain parameters, a classical reduction is known for a subset of these lattice problems [32]. Note that the original hardness result for search-LWE was for a continuous Gaussian error distribution, but this can be converted to a discrete Gaussian disribution with a suitable randomized rounding method [33].

We will need the transformation of Applebaum *et al.* [5] from the standard decision-LWE problem (where $\mathbf{s}$ is uniform) to one where the secret $\mathbf{s}$ is chosen from the error distribution χ.

Lemma 4 ([5, Lemma 2]). *Let $q = p^e$ be a prime power. There is a deterministic polynomial-time transformation that, for arbitrary $\mathbf{s} \in \mathbb{Z}_q^n$ and error distribution χ, maps $A_{\mathbf{s},\chi}$ to $A_{\bar{\mathbf{x}},\chi}$ where $\bar{\mathbf{x}} \leftarrow \chi^n$, and maps $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$ to itself. The transformation also produces an invertible square matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times n}$ and $\bar{\mathbf{b}} \in \mathbb{Z}_q^n$ that, when mapping $A_{\mathbf{s},\chi}$ to $A_{\bar{\mathbf{x}},\chi}$, satisfy $\bar{\mathbf{x}} = -\bar{\mathbf{A}}^t \mathbf{s} + \bar{\mathbf{b}}$.*

2.4 Key-Dependent Message Security

In defining key-dependent message security for public-key encryption and for identity-based encryption, we adapt the original definitions of Black *et al.* [7]. As in their definitions, the adversary plays a game with a challenger, and is able to make encryption queries for functions from a certain family $\mathcal{F}$ of the users' secret keys. (Technically, $\mathcal{F}$ is a family of sets of functions parameterized by the security parameter n and the number of users d.)

To simplify our security proofs, in our definition the adversary specifies two functions $(f_0, f_1) \in \mathcal{F}$ with each query, and must distinguish between encryptions of f_0 and encryptions of f_1. If $f(k_1, \ldots, k_d) = 0$ is contained in $\mathcal{F}$ (which should be the case if we want KDM security to imply standard semantic security), then this definition is at least as strong as (and is in fact equivalent to) the original.

To define KDM-security for identity-based encryption, we extend the standard definition of selective security for IBE from [14, 8]. Note that we add a parameter

d to the Setup algorithm denoting the maximum number of users in a clique (i.e., a set of users such that the secret key for any user in the clique may be safely encrypted under the identity for any user in the clique). An adversary plays a game with a challenger that answer encryption queries for functions of the secret keys for identities from a list $\mathcal{I}$, encrypted under identities from $\mathcal{I}$. For selective security, $\mathcal{I}$ must be specified before the adversary sees the public key and remains static throughout the game. In addition to (key-dependent) encryption queries, the adversary is also allowed to make extraction queries for any identity $id \notin \mathcal{I}$.

For an identity-based encryption scheme $(\mathsf{Setup}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$, the security game between an adversary and a challenger is parameterized by some $\beta \in \{0,1\}$ and proceeds as follows.

1. $\mathcal{A}(1^n, d)$ outputs a list of (distinct) target identities $\mathcal{I} = (id_1, id_2, \ldots id_\ell)$ for some $\ell \leq d$.
2. The challenger runs $(mpk, msk) \leftarrow \mathsf{Setup}(1^n, d)$. The adversary is given mpk. The challenger then extracts secret keys for each of the target identities, running $sk_i \leftarrow \mathsf{Ext}_{msk}(id_i)$ for each $i \in [\ell]$.
3. $\mathcal{A}$ then can make extraction and encryption queries, in the order of its choice.
   Extraction Queries: $\mathcal{A}$ can query $\mathsf{Ext}_{msk}(\cdot)$ for any identity $id \notin \mathcal{I}$
   Encryption Queries: $\mathcal{A}$ can make encryption queries of the form (f_0, f_1, i), where $f_0, f_1 \in \mathcal{F}$ and $1 \leq i \leq \ell$. The challenger computes $m \leftarrow f_\beta(sk_1, \ldots, sk_\ell)$ and $c \leftarrow \mathsf{Enc}(id_i, m)$, and returns c to $\mathcal{A}$.

We say that the scheme is selective-identity KDM-CPA secure with respect to $\mathcal{F}$ if the games for $\beta = 0, 1$ are computationally indistinguishable.

We define KDM-CPA security for a public-key scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ in a similar manner. Starting at phase two above (since there are no identities to target), the challenger now runs Gen d times, and gives $pk_1, \ldots, pk_d$ to the adversary. In phase three, the adversary can only make encryption queries (since there are no identities to extract), and requests encryptions under public keys instead of identities. Everything else is exactly the same.

3 Hardness of Extended LWE

In this section we describe the *extended*-LWE problem (as originally defined in [31]), and give a reduction to it from the standard LWE problem (with polynomially bounded parameters), thus establishing its hardness under a mild assumption.

3.1 Background and the Problem

O'Neill, Peikert and Waters [31] introduced the extended-LWE problem as a simplifying tool for certain security proofs in which LWE is used in a "hash proof-like" fashion, and additional information about the secret key is revealed to the attacker. In prior works, dealing with such situations often involved adding some "overwhelming" (super-polynomial) extra noise in order to disguise a small

but noticeable statistical difference between, e.g., creating a ciphertext honestly according to an encryption algorithm, and creating one by combining the secret key with a challenge LWE instance. Unfortunately, the use of such overwhelming noise requires an underlying LWE problem with super-polynomial modulus q and inverse error rate $1/\alpha$, which corresponds to a substantially stronger assumption than is needed in the security proofs for many other cryptosystems.

Here we recall the formal definition of the extended-LWE problem. In addition to the usual n, q, and χ parameters for LWE, we also have a number $m = \text{poly}(n)$ of LWE samples, an efficiently sampleable "hint" distribution τ over $\mathbb{Z}^m$ (often, a discrete Gaussian $D^m_{\mathbb{Z},r}$ for some $r \geq 1$) and another Gaussian parameter $\beta > 0$. The problem is to distinguish, with non-negligible advantage, between the two experiments described next; the extended-LWE assumption is that this distinguishing problem is hard. In the ExptLWE experiment, the challenger chooses $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$, a secret $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ and error vector $\mathbf{x} \leftarrow \chi^m$ defining $\mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{x}^t$, along with a "hint" vector $\mathbf{z} \leftarrow \tau$ and error term $\tilde{x} \leftarrow D_{\mathbb{Z},\beta q}$, and outputs

$$(\mathbf{A}, \mathbf{b}, \mathbf{z}, b' = \langle \mathbf{x}, \mathbf{z} \rangle + \tilde{x}).$$

The first two components are just m LWE samples, while the latter two components may be seen as a hint about the error vector $\mathbf{x} \in \mathbb{Z}^m$ in the form of a (noisy) inner product with a vector $\mathbf{z} \in \mathbb{Z}^m$, which is not reduced modulo anything. The ExptUnif experiment is the same, except that $\mathbf{b}$ is defined to be uniformly random and independent of everything else.

Notice that because $\mathbf{A}$ and $\mathbf{z}$ are public, one can "amortize" the extended-LWE problem by including any $\text{poly}(n)$ number of vectors $\mathbf{b}_i^t = \mathbf{s}_i^t\mathbf{A} + \mathbf{x}_i^t$ and hints $b'_i = \langle \mathbf{x}_i, \mathbf{z} \rangle$, for independent $\mathbf{s}_i, \mathbf{x}_i$ (and the same $\mathbf{A}, \mathbf{z}$). By a routine hybrid argument, the two forms of the problem are equivalent, up to a $\text{poly}(n)$ factor in the distinguishing advantage. We use the amortized form of the problem in our security proof in Section 4.

As observed in [31] (and implicitly in prior works like [21, 17]), there is a straightforward reduction from LWE with $\chi = D_{\mathbb{Z},\alpha q}$ to extended-LWE where τ is any m-fold product distribution with variance r^2, if the ratio $\beta/(r \cdot \alpha)$ is superpolynomial in n. In fact, in this case we can securely give out an *unbounded* polynomial number of hints $\mathbf{z}_i, b'_i = \langle \mathbf{x}, \mathbf{z}_i \rangle + \tilde{x}_i$ about the error $\mathbf{x}$. This is simply because by Lemma 2, the terms $\tilde{x} \leftarrow D_{\mathbb{Z},\beta q}$ statistically hide the inner product $\langle \mathbf{x}, \mathbf{z} \rangle$, since the latter has magnitude $\approx r\|\mathbf{x}\| \leq r\alpha q\sqrt{m} = \beta q \cdot \text{negl}(n)$. As a result, the reduction can just simulate the hints $(\mathbf{z}, \langle \mathbf{x}, \mathbf{z} \rangle + \tilde{x})$ on its own. The disadvantage of this approach is that in order to be useful, the modulus q and inverse error rate $1/\alpha$ typically must be super-polynomially large in n, which corresponds to assuming the worst-case hardness of various lattice problems for super-polynomial approximation factors and running times.

We also point out that in the above setting, if the ratio $\beta q/r$ is polynomial in n and a sufficiently large $h = \text{poly}(n)$ number of hints are given out, then extended-LWE is *easy* to solve. To see this, view the h hints as $(\mathbf{Z} \in \mathbb{Z}^{m \times h}, \mathbf{y}^t := \mathbf{x}^t\mathbf{Z} + \tilde{\mathbf{x}}^t)$. With overwhelming probability, the singular values of $\mathbf{Z}$ will all be $r \cdot \Omega(\sqrt{h} - C\sqrt{n+m})$ for some universal constant $C > 0$ (see, e.g., [36, Theorem

5.39]). Thus, for sufficiently large $h = \text{poly}(n)$, with overwhelming probability the singular values of the right-inverse $\mathbf{Z}^+ \in \mathbb{R}^{h \times m}$ of $\mathbf{Z}$ will all be small enough so that $\lfloor \tilde{\mathbf{x}}^t \cdot \mathbf{Z}^+ \rceil = \mathbf{0}$. As a result, we can compute $\lfloor \mathbf{y}^t \mathbf{Z}^+ \rceil = \mathbf{x}^t$, which trivially allows for solving the extended-LWE problem.

In the full version, we contrast our results for extended-LWE with syntactically similar (but qualitatively different) results, such as the Goldreich-Levin theorem and those of [21, 17].

3.2 Reduction from LWE

Here we give a tight reduction from standard LWE to extended-LWE, which holds for the same parameters $n, q, \chi, m \geq n + \omega(\log n)$ in the two problems, and in which *no noise* is added to the hint $\langle \mathbf{z}, \mathbf{x} \rangle$ (i.e., $\beta = 0$). Our reduction imposes one requirement on the parameters: for $\mathbf{x} \leftarrow \chi^m$ and $\mathbf{z} \leftarrow \tau$, we need it to be the case that $|\langle \mathbf{x}, \mathbf{z} \rangle| < p$ with overwhelming probability, where p is the smallest prime divisor of the modulus q. For example, if $\chi = D_{\mathbb{Z}, \alpha q}$ and $\tau = D^m_{\mathbb{Z}, r}$, by standard tail inequalities it suffices to have $\alpha q \cdot r\sqrt{m+n} \cdot \omega(\sqrt{\log n}) < p$. In other words, the LWE inverse error rate is $1/\alpha > (q/p) \cdot r\sqrt{m+n}$, which is only polynomial in n when q, r, m are.

Theorem 1. *There exists a probabilistic polynomial-time oracle machine (a simulator) $\mathcal{S}$ such that for any adversary $\mathcal{A}$,*

$$\mathbf{Adv}_{\mathsf{LWE}}(\mathcal{S}^{\mathcal{A}}) \geq \tfrac{1}{2p-1} \cdot \mathbf{Adv}_{\mathsf{ELWE}}(\mathcal{A}) - \text{negl}(n),$$

where the parameters of the LWE *and extended-*LWE *problems satisfy the condition specified above.*

Proof. For the proof it is convenient to use the equivalent "knapsack" form of LWE, which is: given $\mathbf{H} \leftarrow \mathbb{Z}_q^{(m-n) \times m}$ and $\mathbf{c} \in \mathbb{Z}_q^{m-n}$, where $\mathbf{c}$ is either $\mathbf{c} = \mathbf{H}\mathbf{x}$ for $\mathbf{x} \leftarrow \chi^m$, or is uniformly random and independent of $\mathbf{H}$, determine (with non-negl(n) advantage) which is the case. The extended form of the problem also reveals a hint $(\mathbf{z}, \langle \mathbf{x}, \mathbf{z} \rangle + \tilde{x})$, analogously to extended-LWE. The equivalence between LWE and its knapsack form for $m \geq n + \omega(\log n)$, which also applies to their extended versions, has been noticed in several works; a proof appears in [27, Lemmas 4.8 and 4.9].

We construct the reduction $\mathcal{S}$ as follows. It receives an LWE instance (in knapsack form) $\mathbf{H} \in \mathbb{Z}_q^{(m-n) \times m}, \mathbf{c} \in \mathbb{Z}_q^{m-n}$. It samples $\mathbf{z} \leftarrow \tau$, $\mathbf{x}' \leftarrow \chi^m$, and $\mathbf{v} \leftarrow \mathbb{Z}_q^{m-n}$, then lets

$$\mathbf{H}' := \mathbf{H} - \mathbf{v}\mathbf{z}^t \in \mathbb{Z}_q^{(m-n) \times m}, \quad \mathbf{c}' = \mathbf{c} - \mathbf{v} \cdot \langle \mathbf{z}, \mathbf{x}' \rangle \in \mathbb{Z}_q^{m-n}.$$

It sends $(\mathbf{H}', \mathbf{b}', \mathbf{z}, \langle \mathbf{x}', \mathbf{z} \rangle)$ to $\mathcal{A}$ (an adversary for extended-LWE in knapsack form), and outputs what $\mathcal{A}$ outputs.

We now analyze the behavior of $\mathcal{S}$. First consider the case where $\mathbf{H}, \mathbf{c}$ are uniform and independent. Then it is clear that $\mathbf{H}', \mathbf{c}'$ are as well, and both $\mathbf{x}'$

and $\mathbf{z}$ are also chosen exactly as in ExptUnif, so $\mathcal{S}$ perfectly simulates ExptUnif to $\mathcal{A}$.

Now, consider the case where $\mathbf{H}, \mathbf{c}$ are drawn from the knapsack distribution, with $\mathbf{c} = \mathbf{H}\mathbf{x}$ for $\mathbf{x} \leftarrow \chi^m$. In this case, we have that $\mathbf{H}'$ is uniformly random (solely over the choice of $\mathbf{H}$), and

$$\mathbf{c}' = \mathbf{H}\mathbf{x} - \mathbf{v} \cdot \langle \mathbf{z}, \mathbf{x}' \rangle = \mathbf{H}'\mathbf{x} + \mathbf{v} \cdot \langle \mathbf{z}, \mathbf{x} - \mathbf{x}' \rangle.$$

So in the event that $\langle \mathbf{x}', \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{z} \rangle$, we have $\mathbf{c}' = \mathbf{H}'\mathbf{x}$ and so $\mathcal{S}$ perfectly simulates ExptLWE to $\mathcal{A}$. Whereas if $\langle \mathbf{z}, \mathbf{x} - \mathbf{x}' \rangle$ is a unit modulo q, then for any fixed choice of $\mathbf{H}'$, $\mathbf{z}$, $\mathbf{x}$, and $\mathbf{x}'$ (subject to this condition), we have that $\mathbf{c}'$ is uniformly random over the choice of $\mathbf{v}$ alone. Finally, since $\mathbf{x}$ and $\mathbf{x}'$ are identically distributed, it follows that $\mathcal{S}$ perfectly simulates ExptUnif to $\mathcal{A}$.

It remains to analyze the probabilities that $\langle \mathbf{z}, \mathbf{x} - \mathbf{x}' \rangle$ is zero or a unit (modulo q), respectively. First, by assumption $|\langle \mathbf{z}, \mathbf{x} - \mathbf{x}' \rangle| < p$ with overwhelming probability, so exactly one of the two cases holds; moreover, we have $\langle \mathbf{x}, \mathbf{z} \rangle = \langle \mathbf{x}', \mathbf{z} \rangle$ with probability at least $\frac{1}{2p-1} - \mathrm{negl}(n)$ because $\mathbf{x}$ and $\mathbf{x}'$ are independent. The theorem then follows from a routine calculation.

Normal Form. In our cryptosystems, we need to assume the hardness of extended-LWE in "normal form" (as in [30, 5]), where the secret $\mathbf{s} \leftarrow \chi^n$ is drawn from the *error* distribution, the matrix $\mathbf{A}$ and vector $\mathbf{b}^t$ have $m - n$ columns, and the hint is of the form $\mathbf{z} \leftarrow \tau$, $b' = \langle (\mathbf{s}, \mathbf{x}), \mathbf{z} \rangle \in \mathbb{Z}$. Suppose m is sufficiently large so that a uniformly random matrix from $\mathbb{Z}_q^{n \times m}$ contains an invertible n-by-n submatrix with overwhelming probability. Then the reduction from [30, 5] applies to extended-LWE in this form, with the slight modification that LWE samples in the first phase are never "thrown away" but are instead recycled to the second phase.

4 KDM-CPA Secure Public-Key Scheme

Here we present a "dual"-style LWE cryptosystem that is KDM-CPA secure for affine functions of the secret keys. In fact, by setting the parameters appropriately, the construction and security proof also encompass (a slight variant of) the cryptosystem from [25], which has somewhat smaller keys and ciphertexts than "primal" or "dual" systems. In Section 6 we build a KDM-CPA secure IBE around this system.

The cryptosystem involves a few parameters: a modulus $q = p^2$ for a prime p where the message space is $\mathbb{Z}_p$; integer dimensions n, m relating to the underlying LWE problems; and a Gaussian parameter r for key generation and encryption. To make embedding this scheme into our IBE more natural, Gen includes an additional parameter d, which will be used to specify the size of identity cliques in the IBE scheme, and outputs public keys $\mathbf{A}$ that are md columns wide. In the public-key scheme alone, the value d is unrelated to the number of public keys that the adversary can obtain in an attack (which is unbounded), and we would just fix $d = 1$.

- $\mathsf{Gen}(1^n, d)$: choose $\mathbf{A} \in \mathbb{Z}_q^{n \times md}$, $\mathbf{z}_0 \leftarrow D_{\mathbb{Z},r}^n$, $\mathbf{z}_1 \leftarrow D_{\mathbb{Z},r}^{md}$, and let $\mathbf{y} = \mathbf{z}_0 - \mathbf{A}\mathbf{z}_1 = [\mathbf{I}_n \mid -\mathbf{A}]\mathbf{z} \in \mathbb{Z}_q^n$ where $\mathbf{z} = (\mathbf{z}_0, \mathbf{z}_1) \in \mathbb{Z}^{n+md}$. The public key is $(\mathbf{A}, \mathbf{y})$ and the secret key is $\mathbf{z}_1$.
  (Notice that, unlike the dual-style encryption of [20], but like the scheme of [25], the public key component $\mathbf{y}$ is a *perturbed* value of $-\mathbf{A}\mathbf{z}_1$. This will be important in the proof of KDM security.)
- $\mathsf{Enc}(\mathbf{A}, \mathbf{y}, \mu)$: to encrypt a message $\mu \in \mathbb{Z}_p$, choose $\mathbf{x}_0 \leftarrow D_{\mathbb{Z},r}^n$, $\mathbf{x}_1 \leftarrow D_{\mathbb{Z},r}^{md}$ and $x' \leftarrow D_{\mathbb{Z},r}$. Output the ciphertext $\mathbf{c}^t = \mathbf{x}_0^t[\mathbf{A} \mid \mathbf{y}] + [\mathbf{x}_1^t \mid x'] + [\mathbf{0} \mid p \cdot \mu]$.
- $\mathsf{Dec}(\mathbf{z}_1, \mathbf{c})$: Compute $\mu' = \mathbf{c}^t \left[\begin{smallmatrix} \mathbf{z}_1 \\ 1 \end{smallmatrix}\right] \in \mathbb{Z}_q$. Output the $\mu \in \{0, \ldots, p-1\} = \mathbb{Z}_p$ such that μ' is closest to $(p\mu) \bmod q$.

For the public-key system alone, it suffices to take $m \geq n + \omega(\log n)$ by our use of the extended-LWE assumption and its proof of hardness as in Section 3. When embedding the system into an IBE scheme, however, we will use $m = \Theta(n \log q)$ because we need the public parameters to be statistically close to uniform over the choice of the master secret key. The error parameter r must be small enough (relative to q/p) so that decryption is correct with overwhelming probability, but large enough to satisfy the reductions to LWE from worst-case lattice problems [34, 32]; for the latter purpose, $r \geq 2\sqrt{n}$ suffices. (Note that even if part of the security proof relies on LWE in dimension $> n$, this problem is no easier than LWE in dimension n, and so we can still securely use $r = 2\sqrt{n}$ with the larger dimension.)

Here we give some example bounds. Let $r = 2\sqrt{n}$, let

$$p = r^2\sqrt{n + md} \cdot \omega(\sqrt{\log n}) = n\sqrt{n + md} \cdot \omega(\sqrt{\log n}),$$

and let $q = p^2$. Then decryption is correct except with probability $\mathrm{negl}(n)$: let $(\mathbf{A}, \mathbf{y}, \mathbf{z}) \leftarrow \mathsf{Gen}(1^n, d)$. For a ciphertext $\mathbf{c} \leftarrow \mathsf{Enc}(\mathbf{A}, \mathbf{y}, \mu)$, we have

$$\mathbf{c}^t \left[\begin{smallmatrix} \mathbf{z}_1 \\ 1 \end{smallmatrix}\right] = \mathbf{x}_0^t\mathbf{A}\mathbf{z}_1 + \langle \mathbf{x}_1, \mathbf{z}_1 \rangle + \langle \mathbf{x}_0, \mathbf{y} \rangle + x' + p \cdot \mu = \langle \mathbf{x}_0, \mathbf{z}_0 \rangle + \langle \mathbf{x}_1, \mathbf{z}_1 \rangle + x' + p \cdot \mu \bmod q,$$

so decryption is correct whenever $|\langle \mathbf{x}_0, \mathbf{z}_0 \rangle + \langle \mathbf{x}_1, \mathbf{z}_1 \rangle + x'| < p/2$. By known tail bounds on discrete Gaussians, this bound holds except with probability $\mathrm{negl}(n)$ (over the choice of all the random variables), as required.

A proof of the following appears in the full version.

Theorem 2. *The above cryptosystem is KDM-CPA secure with respect to the set of affine functions over $\mathbb{Z}_p$, under the extended-*LWE *assumption for parameters described above.*

5 All-But-*d* Trapdoor Functions

Here we develop a technique for constructing "all-but-d" (tag-based) trapdoor functions, which, informally, are trapdoor functions for which the trapdoor enables efficient inversion for all but (up to) d tags, which are specified at the time of key generation. This is the main tool we use for embedding our KDM-CPA-secure public-key cryptosystem into an identity-based encryption scheme.

Our construction is a generalization (to higher-degree polynomials) of the main technique from [2]. For simplicity and somewhat better efficiency, we follow the construction of [28], specifically the use of a fixed, public "gadget" matrix $\mathbf{G}$ as described in Section 2.2.

5.1 Algebraic Background

Let $n \geq 1$, $q \geq 2$, and $d = \text{poly}(n)$ be integers. Let $\mathcal{R}$ denote any commutative ring (with efficiently computable operations, including inversion of multiplicative units) such that the additive group $\mathbb{G} = \mathbb{Z}_q^n$ is an $\mathcal{R}$-module, and such that there are at least $d+1$ known elements $U = \{u_0 = 0, u_1, u_2, \ldots\} \subseteq \mathcal{R}$ where $u_i - u_j$ is invertible in $\mathcal{R}$ (i.e., a unit) for every $i \neq j$. In particular, we have an (efficiently computable) scalar multiplication operation $\mathcal{R} \times \mathbb{G} \to \mathbb{G}$. Note that multiplication by $u \in \mathcal{R}$ is an invertible linear transformation on $\mathbb{G}$ exactly when u is invertible (i.e., a unit). We extend scalar multiplication in the natural way to vectors and matrices, i.e., $R^{a \times b} \times \mathbb{G}^{b \times c} \to \mathbb{G}^{a \times c}$. To avoid confusion with vectors and matrices over $\mathbb{Z}_q$, we use $\boldsymbol{u}$ notation for vectors over $\mathcal{R}$, and V notation for matrices over $\mathcal{R}$.

To construct a suitable ring, we use ideas from the literature on secret sharing over groups and modules, e.g., [16, 18]. We use an extension ring $\mathcal{R} = \mathbb{Z}_q[x]/(F(x))$ for any monic, degree-n, irreducible $F(x) = F_0 + F_1 x + \cdots + F_{n-1}x^{n-1} + x^n \in \mathbb{Z}_q[x]$. Scalar multiplication $\mathcal{R} \times \mathbb{G} \to \mathbb{G}$ is defined by identifying each $\mathbf{a} = (a_0, \ldots, a_{n-1})^t \in \mathbb{G}$ with the polynomial $a(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} \in \mathcal{R}$, multiplying in $\mathcal{R}$, then mapping back to $\mathbb{G}$. In other words, scalar multiplication is defined by the linear transformation $x \cdot (a_0, \ldots, a_{n-1})^t = (0, a_0, \ldots, a_{n-2})^t - a_{n-1}(F_0, F_1, \ldots, F_{n-1})^t$. It is easy to check that with this scalar product, $\mathbb{G}$ is an $\mathcal{R}$-module. In addition, by the Chinese remainder theorem, $r \in \mathcal{R}$ is a unit if and only if it is nonzero (as a polynomial residue) modulo every prime integer divisor p of q. (This is because $\mathbb{Z}_p[x]/(F(x))$ is a field by construction.) Letting p be the smallest such divisor of q, we can define the universe $U = \{u_0 = 0, u_1, u_2, \ldots\} \subseteq \mathcal{R}$ to consist of all the polynomial residues having coefficients in $\{0, \ldots, p-1\}$. Then $|U| = p^n \geq 2^n$ and $u_i - u_j$ is a unit for all $i \neq j$, as desired.

5.2 Basic Construction

As in [28], we fix a universal public "gadget" matrix $\mathbf{G} \in \mathbb{Z}_q^{n \times w}$ for which there is an efficient Gaussian preimage sampling algorithm for parameter $s \geq \omega(\sqrt{\log n})$, i.e., an algorithm that given any $\mathbf{u} \in \mathbb{Z}_q^n$ outputs a sample from $D_{\Lambda_{\mathbf{u}}^{\perp}(\mathbf{G}),s}$. E.g., we can let $\mathbf{G} = \mathbf{I}_n \otimes (1, 2, 4, \ldots, 2^{k-1}) \in \mathbb{Z}_q^{n \times nk}$ for $k = \lceil \lg q \rceil$.

As input, the trapdoor generator takes:

- an integer $d \geq 1$ and a monic degree-d polynomial $f(z) = c_0 + c_1 z + \cdots + z^d \in \mathcal{R}[z]$,
- a (usually uniformly random) matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{(nd) \times \bar{m}}$ for some $\bar{m} \geq 1$, which is made up of stacked submatrices $\bar{\mathbf{A}}_i \in \mathbb{Z}_q^{n \times \bar{m}}$ for $i = 0, \ldots, d-1$.

– a "short" secret $\mathbf{R} \in \mathbb{Z}^{\bar{m} \times w}$ chosen at random from an appropriate distribution (typically, a discrete Gaussian) to serve as a trapdoor.

As output it produces a matrix $\mathbf{A} \in \mathbb{Z}_q^{(nd) \times (\bar{m}+w)}$ (which is statistically close to uniform, when the parameters and input $\bar{\mathbf{A}}$ are appropriately chosen). In addition, for each tag $u \in U$ there is an efficiently computable (from $\mathbf{A}$) matrix $\mathbf{A}_u \in \mathbb{Z}_q^{n \times (\bar{m}+w)}$ for which $\mathbf{R}$ *may* be a trapdoor, depending on the value of $f(u) \in \mathcal{R}$.

We write the coefficients of $f(z)$ as a column vector $\boldsymbol{c} = (c_0, c_1, \ldots, c_{d-1})^t \in \mathcal{R}^d$, and define

$$\mathbf{A}'_f := \left[\bar{\mathbf{A}} \; \boldsymbol{c} \otimes \mathbf{G}\right] = \begin{bmatrix} \bar{\mathbf{A}}_0 & c_0 \cdot \mathbf{G} \\ \vdots & \vdots \\ \bar{\mathbf{A}}_{d-1} & c_{d-1} \cdot \mathbf{G} \end{bmatrix} \in \mathbb{Z}_q^{(nd) \times (\bar{m}+w)}.$$

To hide the polynomial f, we output the public key

$$\mathbf{A} := \mathbf{A}'_f \cdot \begin{bmatrix} \mathbf{I} & -\mathbf{R} \\ & \mathbf{I} \end{bmatrix} = \left[\bar{\mathbf{A}} \; (\boldsymbol{c} \otimes \mathbf{G}) - \bar{\mathbf{A}}\mathbf{R}\right].$$

Note that as long as the distribution of $[\bar{\mathbf{A}} \mid -\bar{\mathbf{A}}\mathbf{R}]$ is statistically close to uniform, then so is $\mathbf{A}$ for any f.

The tag space for the trapdoor function is the set $U \subset \mathcal{R}$. For any tag $u \in U$, define the row vector $\boldsymbol{u}^t := (u^0, u^1, \cdots, u^{d-1}) \in \mathcal{R}^d$ (where $0^0 = 1$) and the derived matrix for tag u to be

$$\mathbf{A}_u := \boldsymbol{u}^t \cdot \mathbf{A} + \left[\mathbf{0} \; u^d \cdot \mathbf{G}\right] = \left[\boldsymbol{u}^t \cdot \bar{\mathbf{A}} \; f(u) \cdot \mathbf{G}\right] \cdot \begin{bmatrix} \mathbf{I} & -\mathbf{R} \\ & \mathbf{I} \end{bmatrix}.$$

By the condition in Lemma 3, $\mathbf{R}$ is a (strong) trapdoor for $\mathbf{A}_u$ exactly when $f(u) \in \mathcal{R}$ is a unit, because $\mathbf{A}_u \cdot \left[\begin{smallmatrix} \mathbf{R} \\ \mathbf{I} \end{smallmatrix}\right] = f(u) \cdot \mathbf{G}$ and $f(u)$ represents an invertible linear transformation when it is a unit.

5.3 Puncturing

In our cryptosystems and security proofs we will need to generate (using the above procedure) an all-but-d trapdoor function that is "punctured" at up to d tags. More precisely, we are given as input:

– a set of distinct tags $P = \{u_1, \ldots, u_\ell\} \subseteq U$ for some $\ell \leq d$,
– uniformly random matrices $\mathbf{A}_i^* \in \mathbb{Z}_q^{n \times \bar{m}}$ for $i \in [\ell]$ (which often come from an SIS or LWE challenge),
– a "short" secret $\mathbf{R} \in \mathbb{Z}^{\bar{m} \times w}$ chosen at random from an appropriate distribution (typically, a discrete Gaussian) to serve as a trapdoor,
– optionally, some uniformly random auxiliary matrices $\mathbf{Y}_i^* \in \mathbb{Z}_q^{n \times k}$ for $i \in [\ell]$ and some $k \geq 0$.

As output we produce a public key $\mathbf{A} \in \mathbb{Z}_q^{(nd)\times\bar{m}}$ and auxiliary matrix $\mathbf{Y} \in \mathbb{Z}_q^{(nd)\times k}$ so that:

1. Each $\mathbf{A}_{u_i}$ matches the challenge matrix $\mathbf{A}_i^*$, and $\mathbf{R}$ is only a "weak" trapdoor for $\mathbf{A}_{u_i}$. More precisely,
$$\mathbf{A}_{u_i} = \left[\mathbf{A}_i^* \ \mathbf{0}\right] \cdot \begin{bmatrix} \mathbf{I} & -\mathbf{R} \\ & \mathbf{I} \end{bmatrix}.$$
2. $\mathbf{R}$ is a (strong) trapdoor for $\mathbf{A}_u$ for any *nonzero* $u \in U \setminus P$, i.e., $f(u)$ is a unit.
3. The auxiliary matrix $\mathbf{Y}_{u_i} := \boldsymbol{u_i}^t \cdot \mathbf{Y}$ equals the auxiliary input $\mathbf{Y}_i^*$ for each $u_i \in P$.

We satisfy these criteria by invoking the above trapdoor generator with the following inputs f and $\bar{\mathbf{A}}$:

1. We define the monic degree-d polynomial
$$f(z) = z^{d-\ell} \cdot \prod_{i\in[\ell]} (z - u_i) \in \mathcal{R}[z]$$
and expand to compute its coefficients $c_i \in \mathcal{R}$. Note that $f(u_i) = 0$ for every $u_i \in P$, and $f(u)$ is a unit for any nonzero $u \in U \setminus P$ because $0 \in U$ and $u_i - u_j$ is a unit for every distinct $u_i, u_j \in U$.
2. We define $\bar{\mathbf{A}}$ using interpolation: let $\mathbf{A}^* \in \mathbb{Z}_q^{(n\ell)\times\bar{m}}$ denote the stack of challenge matrices $\mathbf{A}_i^*$, and let $V \in \mathcal{R}^{\ell\times d}$ be the Vandermonde matrix whose rows are the vectors $\boldsymbol{u_i}^t$ defined above. We then let $\bar{\mathbf{A}} \in \mathbb{Z}_q^{(nd)\times\bar{m}}$ be a uniformly random solution to $V \cdot \bar{\mathbf{A}} = \mathbf{A}^*$.
Such a solution exists, and is efficiently computable and uniformly random (over the uniformly random choice of $\mathbf{A}^*$ and the random solution chosen). To see this, extend V to an invertible $d \times d$ Vandermonde matrix over $\mathcal{R}$ having unit determinant $\prod_{i<j}(u_j - u_i) \in \mathcal{R}^*$, by adding $d - \ell$ additional rows $\boldsymbol{u_j}^t$ for arbitrary distinct $u_j \in U \setminus P$. Likewise, extend $\mathbf{A}^*$ to have dimension $(nd) \times \bar{m}$ by adding uniformly random rows. Then for any fixed choice of the (extended) matrix V, the (extended) matrix $\mathbf{A}^*$ and solution $\bar{\mathbf{A}}$ are in bijective correspondence, and so the latter is uniformly random because the former is.
3. We also define the auxiliary matrix $\mathbf{Y}$ similarly using interpolation, as a uniformly random solution to $V \cdot \mathbf{Y} = \mathbf{Y}^*$.

6 Circular-Secure IBE

Our IBE scheme is a generalization of the efficient IBE scheme of Agrawal *et al.* [2]. Other than some minor changes in the parameters, the main difference is the use of the all-but-d trapdoor construction, which allows us to "puncture"

the master public key at up to d identities in the security proof. The scheme has parameters modulus q, message space $\mathbb{Z}_p$ for some $p < q$, dimension m, and Gaussian parameters r and γ. Most of the parameters match those in the public-key encryption scheme of Section 4, with the additional constraint that r must be large enough that we can run the preimage sampling algorithm (Lemma 3) in Ext. Due to space considerations, the conditions on the parameters are described in the full version.

The identity space for the scheme is $U \setminus \{0\} \subset \mathcal{R}$, where U, $\mathcal{R}$ are constructed as in Section 5.

- $\mathsf{Setup}(1^n, d)$: On input security parameter 1^n and secret key clique size d:
  1. Sample $\mathbf{R} \leftarrow D^{md \times w}_{\mathbb{Z},\omega(\sqrt{\log n})}$, and for $i = 0, \ldots, d-1$, choose uniformly random $\mathbf{A}_i \leftarrow \mathbb{Z}_q^{n \times md}$, $\mathbf{y}_i \leftarrow \mathbb{Z}_q^n$ and let $\tilde{\mathbf{A}}_i = -\mathbf{A}_i\mathbf{R} \in \mathbb{Z}_q^{n \times w}$. (Note that this is simply calling the all-but-d trapdoor construction from Section 5 with an empty set of punctured tags.) Let $\mathbf{A}^t := \left[\mathbf{A}_0^t \cdots \mathbf{A}_{d-1}^t\right]$, $\tilde{\mathbf{A}}^t := \left[\tilde{\mathbf{A}}_0^t \cdots \tilde{\mathbf{A}}_{d-1}^t\right]$, $\mathbf{y}^t := \left[\mathbf{y}_0^t \cdots \mathbf{y}_{d-1}^t\right]$. Note that $\tilde{\mathbf{A}} = -\mathbf{A}\mathbf{R}$.
  2. The public key is $mpk = (\mathbf{A}, \tilde{\mathbf{A}}, \mathbf{y})$. The master secret key is $msk = (\mathbf{R})$.
- $\mathsf{Ext}(mpk, msk, u)$ On input mpk, msk and $u \in U \setminus \{0\} \subseteq \mathcal{R}$:
  1. Let $\boldsymbol{u}^t := (u^0, u^1, \ldots, u^{d-1})$, $\bar{\mathbf{A}}_u = \boldsymbol{u}^t \cdot \mathbf{A}$, $\mathbf{y}_u = \boldsymbol{u}^t \cdot \mathbf{y}$ and $\mathbf{A}_u = [\bar{\mathbf{A}}_u \mid u^d\mathbf{G} - \bar{\mathbf{A}}_u\mathbf{R}]$, as in Section 5.
  2. Sample $\mathbf{z}_0 \leftarrow D^n_{\mathbb{Z},r}$, $\mathbf{z}_1 \leftarrow D_{\Lambda^\perp_{\mathbf{z}_0 - \mathbf{y}_u}(\mathbf{A}_u),r}$ using the preimage sampling algorithm (Lemma 3), so that $\mathbf{y}_u = \mathbf{z}_0 - \mathbf{A}_u\mathbf{z}_1$ (as in the public-key cryptosystem from Section 4). Output $sk_u := \mathbf{z}_1$.
     Note that the above is possible because $u^d \in \mathcal{R}$ is a unit, and by our choice of r below, because $s_1(\mathbf{R}) = O(\sqrt{md}+\sqrt{w})\cdot\omega(\sqrt{\log n}) = O(\sqrt{md})\cdot\omega(\sqrt{\log n})$ with all but $\mathrm{negl}(n)$ probability by Lemma 1.
- $\mathsf{Enc}(mpk, u, \mu)$: On input master public key, identity $u \in U \setminus \{0\}$, and message $\mu \in \mathbb{Z}_p$ do:
  1. Let $\boldsymbol{u}^t := (u^0, u^1, \ldots, u^{d-1})$, $\mathbf{A}_u = [\boldsymbol{u}^t \cdot \mathbf{A} \mid u^d\mathbf{G} + \boldsymbol{u}^t \cdot \tilde{\mathbf{A}}] \in \mathbb{Z}_q^{n \times md+w}$, and $\mathbf{y}_u = \boldsymbol{u}^t \cdot \mathbf{y}$.
  2. Choose $\mathbf{s} \leftarrow D^n_{\mathbb{Z},r}$, $\mathbf{x}_0 \leftarrow D^{md}_{\mathbb{Z},r}, \mathbf{x}_1 \leftarrow D^w_{\mathbb{Z},\gamma}$, $x_2 \leftarrow D_{\mathbb{Z},r}$. Let $\mathbf{x}^t = [\mathbf{x}_0^t \mid \mathbf{x}_1^t]$.
  3. Output the ciphertext $\mathbf{c}^t = \mathbf{s}^t[\mathbf{A}_u \mid \mathbf{y}_u] + [\mathbf{x}^t \mid x_2] + [\mathbf{0} \mid p \cdot \mu]$.
- $\mathsf{Dec}(mpk, sk_u = \mathbf{z}_1, \mathbf{c})$: output the $\mu \in \mathbb{Z}_p$ such that $\mathbf{c}^t \left[\begin{smallmatrix}\mathbf{z}_1\\ 1\end{smallmatrix}\right]$ is closest to $p \cdot \mu$ modulo q.

Theorem 3. *For the above parameters, the above IBE scheme is selective identity KDM-CPA secure with respect to the set of affine functions over $\mathbb{Z}_p$, under the $\mathsf{LWE}_{q,\chi}$ assumption for $\chi = D_{\mathbb{Z},r}$, and the KDM-CPA security of the system from Section 4.*

Proof (Sketch). Here we give an overview of the proof strategy, deferring the formal proof to the full version. Game 0 is the actual attack game. In Game 1, we use the all-but-d trapdoor construction from Section 5 to generate the master public key, "puncturing" it at the targeted identities. Finally, in Game 2, we play the KDM-CPA security game against a challenger running the public-key encryption scheme from Section 4 and use the outputs of the challenger

to simulate Game 1. This requires some care because the IBE secret keys and ciphertexts have larger dimension by an additive term of w (the width of $\mathbf{G}$). To address this, we fill in the missing dimensions of the secret keys by choosing them ourselves, and use knowledge of the master secret key to fill in the missing dimensions of the ciphertexts (here is where we use the fact that noise with parameter γ "overwhelms" noise with parameter r).

Acknowledgments. We thanks Oded Regev for helpful comments, and for pointing out a subtle error in a prior version of our reduction from Section 3.

References

[1] Adão, P., Bana, G., Herzog, J., Scedrov, A.: Soundness of Formal Encryption in the Presence of Key-Cycles. In: de Capitani di Vimercati, S., Syverson, P.F., Gollmann, D. (eds.) ESORICS 2005. LNCS, vol. 3679, pp. 374–396. Springer, Heidelberg (2005)

[2] Agrawal, S., Boneh, D., Boyen, X.: Efficient Lattice (H)IBE in the Standard Model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010)

[3] Ajtai, M.: Generating hard instances of lattice problems. Quaderni di Matematica 13, 1–32 (2004); Preliminary version in STOC 1996

[4] Applebaum, B.: Key-Dependent Message Security: Generic Amplification and Completeness. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 527–546. Springer, Heidelberg (2011)

[5] Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 595–618. Springer, Heidelberg (2009)

[6] Barak, B., Haitner, I., Hofheinz, D., Ishai, Y.: Bounded Key-Dependent Message Security. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 423–444. Springer, Heidelberg (2010)

[7] Black, J., Rogaway, P., Shrimpton, T.: Encryption-Scheme Security in the Presence of Key-Dependent Messages. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 62–75. Springer, Heidelberg (2003)

[8] Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM J. Comput. 36(5), 1301–1328 (2007)

[9] Boneh, D., Halevi, S., Hamburg, M., Ostrovsky, R.: Circular-Secure Encryption from Decision Diffie-Hellman. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 108–125. Springer, Heidelberg (2008)

[10] Brakerski, Z., Goldwasser, S.: Circular and Leakage Resilient Public-Key Encryption Under Subgroup Indistinguishability - (or: Quadratic Residuosity Strikes Back). In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 1–20. Springer, Heidelberg (2010)

[11] Brakerski, Z., Goldwasser, S., Kalai, Y.T.: Black-Box Circular-Secure Encryption beyond Affine Functions. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 201–218. Springer, Heidelberg (2011)

[12] Brakerski, Z., Vaikuntanathan, V.: Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 505–524. Springer, Heidelberg (2011)

[13] Camenisch, J., Lysyanskaya, A.: An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer, Heidelberg (2001)
[14] Canetti, R., Halevi, S., Katz, J.: A Forward-Secure Public-key Encryption Scheme. J. Cryptology 20(3), 265–294 (2007); Preliminary version in EUROCRYPT 2003
[15] Chatterjee, S., Sarkar, P.: Generalization of the Selective-ID Security Model for HIBE Protocols. In: Yung, M., et al. (eds.) PKC 2006. LNCS, vol. 3958, pp. 241–256. Springer, Heidelberg (2006)
[16] Desmedt, Y., Frankel, Y.: Perfect homomorphic zero-knowledge threshold schemes over any finite abelian group. SIAM J. Discrete Math. 7(4), 667–679 (1994)
[17] Dodis, Y., Goldwasser, S., Tauman Kalai, Y., Peikert, C., Vaikuntanathan, V.: Public-Key Encryption Schemes with Auxiliary Inputs. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 361–381. Springer, Heidelberg (2010)
[18] Fehr, S.: Span programs over rings and how to share a secret from a module. Master's thesis, ETH Zurich, Institute for Theoretical Computer Science (1998)
[19] Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC, pp. 169–178 (2009)
[20] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC, pp. 197–206 (2008)
[21] Goldwasser, S., Kalai, Y.T., Peikert, C., Vaikuntanathan, V.: Robustness of the learning with errors assumption. In: ICS, pp. 230–240 (2010)
[22] Goldwasser, S., Micali, S.: Probabilistic encryption. J. Comput. Syst. Sci. 28(2), 270–299 (1982); Preliminary version in STOC 1982
[23] Haitner, I., Holenstein, T.: On the (Im)Possibility of Key Dependent Encryption. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 202–219. Springer, Heidelberg (2009)
[24] Hemenway, B., Libert, B., Ostrovsky, R., Vergnaud, D.: Lossy Encryption: Constructions from General Assumptions and Efficient Selective Opening Chosen Ciphertext Security. In: Lee, D.H. (ed.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 70–88. Springer, Heidelberg (2011)
[25] Lindner, R., Peikert, C.: Better Key Sizes (and Attacks) for LWE-Based Encryption. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 319–339. Springer, Heidelberg (2011)
[26] Malkin, T., Teranishi, I., Yung, M.: Efficient Circuit-Size Independent Public Key Encryption with KDM Security. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 507–526. Springer, Heidelberg (2011)
[27] Micciancio, D., Mol, P.: Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 465–484. Springer, Heidelberg (2011)
[28] Micciancio, D., Peikert, C.: Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012)
[29] Micciancio, D., Regev, O.: Worst-case to average-case reductions based on Gaussian measures. SIAM J. Comput. 37(1), 267–302 (2004); Preliminary version in FOCS 2004
[30] Micciancio, D., Regev, O.: Lattice-based cryptography. In: Post Quantum Cryptography, pp. 147–191. Springer (February 2009)
[31] O'Neill, A., Peikert, C., Waters, B.: Bi-Deniable Public-Key Encryption. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 525–542. Springer, Heidelberg (2011)

[32] Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem. In: STOC, pp. 333–342 (2009)
[33] Peikert, C.: An Efficient and Parallel Gaussian Sampler for Lattices. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 80–97. Springer, Heidelberg (2010)
[34] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM 56(6), 1–40 (2005); Preliminary version in STOC 2005
[35] Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
[36] Vershynin, R.: Introduction to the non-asymptotic analysis of random matrices (January 2011), http://www-personal.umich.edu/~romanv/papers/non-asymptotic-rmt-plain.pdf (last accessed February 4, 2011)

NTRUCCA: How to Strengthen NTRUEncrypt to Chosen-Ciphertext Security in the Standard Model

Ron Steinfeld[1,⋆], San Ling[2], Josef Pieprzyk[3], Christophe Tartary[4], and Huaxiong Wang[2]

[1] Clayton School of Information Technology
Monash University, Clayton VIC 3800, Australia
ron.steinfeld@monash.edu
[2] Div. of Mathematical Sciences,
School of Physical and Mathematical Sciences,
Nanyang Technological University, Singapore, 637371
{lingsan,hxwang}@ntu.edu.sg
[3] Centre for Advanced Computing - Algorithms and Cryptography,
Dept. of Computing,
Macquarie University, Sydney, NSW 2109, Australia
josef.pieprzyk@mq.edu.au
[4] Institute for Theoretical Computer Science,
Tsinghua University, People's Republic of China
ctartary@mail.tsinghua.edu.cn

Abstract. NTRUEncrypt is a fast and practical lattice-based public-key encryption scheme, which has been standardized by IEEE, but until recently, its security analysis relied only on heuristic arguments. Recently, Stehlé and Steinfeld showed that a slight variant (that we call pNE) could be proven to be secure under chosen-plaintext attack (IND-CPA), assuming the hardness of worst-case problems in ideal lattices. We present a variant of pNE called NTRUCCA, that is IND-CCA2 secure in the standard model assuming the hardness of worst-case problems in ideal lattices, and only incurs a *constant factor* overhead in ciphertext and key length over the pNE scheme. To our knowledge, our result gives the first IND-CCA2 secure variant of NTRUEncrypt in the standard model, based on standard cryptographic assumptions.

As an intermediate step, we present a construction for an All-But-One (ABO) lossy trapdoor function from pNE, which may be of independent interest. Our scheme uses the lossy trapdoor function framework of Peikert and Waters, which we generalize to the case of $(k-1)$-of-k-correlated input distributions.

Keywords: Chosen-Ciphertext Security, Lossy Trapdoor Function, Lattice-based cryptography, NTRU, ideal lattice, provable security.

⋆ This work was done while the first author was with Macquarie University.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 353–371, 2012.

1 Introduction

Background. It is now widely recognized that most practical applications of public-key cryptosystems require more than the basic passive security against chosen-plaintext eavesdropping attacks (known as IND-CPA security). The de facto standard requirement that suffices for the majority of applications is security against chosen-ciphertext attacks, known as IND-CCA2 security[28].

With the recent development of *lattice*-based cryptography, a public-key cryptosystem with public-key length $O(n^2 \log^2 n)$ and ciphertext length $O(n \log^2 n)$ (for security parameter n) was given by Regev [29], having IND-CPA security provably based on the Learning With Errors (LWE) problem, which in turn was shown to be as hard as the quantum worst-case hardness of standard lattice problems. A 'dual' variant system with similar complexity was later proposed in [11]. The large quadratic factor n^2 in the public-key length is due to the unstructured matrices used in the LWE problem. By moving to a structured matrix (first proposed for lattice-based hash functions in [20,17,25]), it was shown independently and concurrently in [33] and [19] how one could construct variants of Regev's cryptosystem based on the *Ring-LWE* problem (a variant of LWE over rings of cyclotomic number fields) with IND-CPA security provably based on the quantum worst-case hardness of lattice problems over the class of structured lattices called *ideal lattices*. The corresponding structured matrices allow the public-key length to be reduced to $O(n \log n)$ (as well as the encryption and decryption complexity, by using FFT techniques).

While the above systems are supported by theoretically sound proofs of security, the most practical and efficient lattice-based cryptosystem to date has been the `NTRU` encryption scheme, proposed in 1996 [7]. The scheme, now known as `NTRUEncrypt`, has been suggested as one of the most practical public-key encryption scheme with conjectured 'post-quantum' security (see, e.g., [27]). Its practicality is also evidenced by its industrial standardization by the IEEE [15]. However, until recently, the security of `NTRUEncrypt` has only been analyzed heuristically. But recently, Stehlé and Steinfeld [34] showed that a slight variant of `NTRUEncrypt`(that we call `pNE`) can be shown to achieve IND-CPA security based on worst-case lattice problems over ideal lattices. Unfortunately, the `pNE` scheme (like the original `NTRUEncrypt` scheme) is trivially insecure against chosen-ciphertext attacks, due to its homomorphic properties.

Our Results. The practicality and standardization of the `NTRUEncrypt` scheme on the one hand, together with the recent result of [34] on the passive (IND-CPA) security of a slight variant of `NTRUEncrypt`, raise the natural question of whether `NTRUEncrypt` can be adapted efficiently to achieve IND-CCA2 security in the standard model, while preserving the strong security guarantees of [34] based on the worst-case hardness of lattice problems in ideal lattices. In this paper, we answer this question affirmatively, in the asymptotic sense. We present a variant of `NTRUEncrypt` called `NTRUCCA`, that is IND-CCA2 secure in the standard model assuming the worst-case quantum hardness of problems in ideal lattices, and only incurs a *constant factor* overhead in ciphertext and key length over the `pNE` variant shown to be IND-CPA in [34]. Namely, our scheme still enjoys a key

and ciphertext length and encryption/decryption computation costs quasi-linear in the security parameter, given the best known attacks. To our knowledge, our scheme is the first efficient variant of NTRUEncrypt achieving IND-CCA2 security based on standard cryptographic assumptions. We emphasize that our aim is here is to show the asymptotic feasibility of obtaining an efficient IND-CCA2 NTRUEncrypt variant from standard cryptographic assumptions, and we leave it to future work to reduce the constant factor overhead incurred by our construction, as well as the overhead incurred by the pNE scheme of [34] over the original NTRUEncrypt scheme.

As an intermediate step, we present a construction for an All-But-One (ABO) lossy trapdoor function from pNE, which may be of independent interest. The public key of our ABO function consists of just one NTRU public-key and one NTRU ciphertext, while our function output is a single NTRU ciphertext. As part of our ABO construction, using the results of [32] on a variant of the NTRUSign signature scheme, we also present a variant of pNE, preserving its security reduction, but with full randomness recovery during decryption (i.e. the randomness used in encryption is recovered during decryption along with the message, whereas in the pNE scheme from [34], only the message is recovered in decryption). Our NTRUCCA scheme is built from our ABO lossy trapdoor function by using a generalization of the generic Peikert-Waters construction of IND-CCA2 encryption from ABO lossy trapdoor functions. This generalization, which may be of independent interest, is required since our pNE-based ABO does not have a sufficient lossiness to be used within the generic IND-CCA2 construction of Peikert and Waters [26]. Our generalized construction uses $(k-1)$-of-k-correlated input distributions (used also in [22]) to weaken the lossiness requirement from the ABO sufficiently to allow us to use it.

Related Work. The first construction of a cryptosystem with IND-CCA2 security provably based on worst-case lattice problems (in the standard model) was given by Peikert and Waters [26]. Their general framework, which also forms the basis for our result, was a construction of IND-CCA2 secure encryption from a primitive called a *lossy* ABO trapdoor function family, along with a one-time signature scheme. They then showed how to construct a lossy ABO family from the LWE problem (and hence from worst-case lattice problems). The resulting IND-CCA2 scheme, however, has quadratic complexity $\Omega(n^2)$ in the security parameter n due to the use of the LWE problem in the underlying ABO, rather than the structured Ring-LWE problem. While the ABO construction of [26] could be applied in the Ring-LWE setting to obtain a quasi-linear complexity in n (like the complexity of our NTRUEncrypt-based ABO in this paper), the lossiness of the construction is based on non-square Ring-LWE matrices (having at least two ring elements), and is not directly applicable to our NTRUEncrypt setting in which the Ring-LWE matrix is square and consists of a single ring element. Instead, we show how to use a 'masking' based approach to provide lossiness in our NTRUEncrypt-based ABO (see Sec. 3 for more details).

Rosen and Segev [30] gave another general construction for an IND-CCA2 secure scheme inspired by Peikert and Waters [26], but starting from a weaker

primitive called a correlation-secure trapdoor function family (which can be constructed from a lossy trapdoor function family). Subsequently, Peikert [24] showed how to construct a correlation-secure trapdoor function family from the LWE problem, and used it within the Rosen-Segev scheme, to obtain another lattice-based IND-CCA2 secure scheme. Unfortunately, the latter scheme suffers from long public-key and ciphertext length of $\Omega(n^2)$ in the security parameter n, even if applied in the Ring-LWE setting.

More constructions of IND-CCA2 secure lattice-based encryption schemes can be obtained by using the lattice-based selective-ID secure IBE schemes of [1,2] within the generic construction of [5], and a one-time signature or commitment scheme. Until very recently, it was unknown how to instantiate the most efficient scheme from [1] based on Ring-LWE with a poly-time reduction from worst-case problems in ideal lattices, but this has now been resolved by Langlois and Stehlé [16], who show the hardness of decision Ring-LWE for any modulus q. A similar and more efficient (in terms of constant factors) system follows by adapting the recent techniques of [21] to the Ring-LWE setting. Thus several candidates now exist, besides our NTRU-based scheme, for efficient IND-CCA2 encryption based on Ring-LWE. We leave it to future work to optimize and compare the concrete performance of all these schemes.

The 'masking' approach we use for constructing our `NTRUEncrypt` based ABO is similar to that used in constructions of lossy functions in [9] based on classical number-theoretic assumptions; our construction shows how to extend this approach to the `NTRUEncrypt` setting. Our use of a $(k-1)$-of-k correlated input distribution in our IND-CCA2 scheme is similar to a technique used by Mol and Yilek [22] to improve the Rosen-Segev [30] construction. Our generalized Peikert-Waters construction offers efficiency gains by a factor linear in the security parameter, when one starts from an ABO lossy function losing a constant fraction of its input entropy (such as our `NTRUEncrypt`-based ABO function).

Note that this paper focuses exclusively on the *standard* model. If one is willing to use hash functions modeled as random oracles [3], then one can obtain efficient IND-CCA2 secure variants of `NTRUEncrypt` by generic transformations from IND-CPA secure cryptosystems [10], or by using more optimized variants tailored for `NTRUEncrypt` [23,14,31]. However, in practice, when the random oracle is instantiated with a public cryptographic hash function, one does not obtain any security guarantees for the resulting scheme from standard cryptographic assumptions.

Due to space limitations, we have omitted some proofs in this version of the paper. They can be found in the full version, on the authors' web page.

2 Preliminaries

2.1 Notation

We assume throughout this paper that n is a power of 2, and q is a prime such that $x^n + 1$ splits into n linear factors modulo q (i.e. $2n$ divides $q - 1$), and we denote by R and R_q the rings $\mathbb{Z}[x]/(x^n+1)$ and $\mathbb{Z}_q[x]/(x^n+1)$, respectively, and

by K the field $\mathbb{Q}[x]/(x^n+1)$. The set of invertible elements of R_q is denoted by $R_q^\times$. We use the asymptotic notations $O(\cdot), \widetilde{O}(\cdot), o(\cdot), \omega(\cdot), \Omega(\cdot), \widetilde{\Omega}(\cdot), \Theta(\cdot)$. We let $U(D)$ denote the uniform distribution over domain D.

2.2 Lattice Background

A lattice is a set of the form $L = \sum_{i \leq n} \mathbb{Z}\boldsymbol{b}_i$, where the $\boldsymbol{b}_i$'s are linearly independent vectors in $\mathbb{R}^n$. The integer n is called the *lattice dimension*, and the $\boldsymbol{b}_i$'s are called a *basis* of L. The *minimum* $\lambda_1(L)$ is the Euclidean norm of any shortest non-zero vector of L. A lattice L is called *ideal* if it consists of the set of coefficient vectors of the elements in an ideal of the ring R. The γ-Ideal-SVP(*IdSVP*) problem is, given a basis for an ideal lattice L, to compute a non-zero vector in L whose norm is at most $\gamma\lambda_1(L)$.

For a lattice L and a deviation parameter $\sigma > 0$, we denote by $D_{L,\sigma}$ the discrete Gaussian probability distribution on L, defined by $D_{L,\sigma}(\boldsymbol{x}) = \rho_\sigma(\boldsymbol{x})/\rho(L)$, where $\rho_\sigma(\boldsymbol{x}) = \exp(-\pi\|\boldsymbol{x}\|^2/\sigma^2)$. We denote by χ_α a certain discrete 'Gaussian-like' distribution (denoted $\bar{\Gamma}_\alpha$ in [34]) on ring R, which is used in [19] as the error distribution for the Ring-LWE problem in order to allow a security reduction from the γ-Ideal-SVP problem. The precise definition of this distribution is quite technical (we refer to [34] and [19] for more details). For the purposes of this paper, it suffices to know that χ_α can be sampled efficiently (in expected time $\widetilde{O}(n)$) and samples from it have small norm. Here we need a stronger version of this Lemma that applies for all $r \in R_p$, rather than just for one fixed r.

Lemma 1 (Adapted from [32]). *For y sampled from χ_α, we have:*

$$\Pr\left[\exists r \in R_p \text{ such that } \|yr\|_\infty \geq p \cdot \omega(n\sqrt{\log n}) \cdot \alpha q\right] \leq n^{-\omega(1)}$$

and

$$\Pr\left[\exists r \in R_p \text{ such that } \|yr\|_\infty \geq p \cdot n^{1.5} \cdot \alpha q\right] \leq 2^{-\Omega(n)}.$$

For $s \in R_q$, let A_{s,χ_α} denote the distribution on $R_q \times R_q$, where a sample from A_{s,χ_α} consists of a pair (a, y) with a independently and uniformly distributed in $R_q^\times$ and $y = a \cdot s + e$ with e independently sampled from χ_α. The *Ring-LWE problem* R-LWE$_{\alpha,q}$ (denoted by R-LWE$^\times_{\text{HNF}}$ in [34]) is the following: Let $s \in R_q$ be sampled from χ_α. Given an oracle $\mathcal{O}$ that produces samples in $R_q \times R_q$, distinguish whether $\mathcal{O}$ outputs samples from the distribution A_{s,χ_α} or from the uniform distribution on $R_q^\times \times R_q$.

Theorem 1 (Adapted from [19]). *Assume that $\alpha q = \omega(n\sqrt{\log n})$ (resp. $\Omega(n^{1.5})$) with $\alpha \in (0,1)$ and $q = \mathcal{P}oly(n)$. There exists a randomized polynomial-time (resp. subexponential) quantum reduction from γ-*Ideal-SVP *to* R-LWE$_{q,\alpha}$, with $\gamma = \omega(n^{1.5}\log n)/\alpha$ (resp. $\Omega(n^{2.5})/\alpha$).*

We recall the scheme pNE, the provably secure variant of NTRUEncrypt, with parameters n, q, p, α, σ [34]. pNE differs from the original NTRUEncrypt [13] in several minor aspects: the choice of ring $R = \mathbb{Z}[x]/(x^n+1)$ (versus $R = \mathbb{Z}[x]/(x^n-1)$),

Key generation.
- Sample f from $D_{\mathbb{Z}^n,\sigma}$; let $f = p \cdot f' + 1$; if $(f \bmod q) \notin R_q^\times$, resample.
- Sample g from $D_{\mathbb{Z}^n,\sigma}$; if $(g \bmod q) \notin R_q^\times$, resample.
- Return secret key $sk = f \in R_q^\times$ and public key $pk = h = g/f \in R_q^\times$.

Encryption. Given message $M \in R_p$, set $s, e \hookleftarrow \chi_\alpha$ and return ciphertext $C = p \cdot (hs + e) + M \in R_q$.

Decryption. Given ciphertext C and secret key f, compute $C' = f \cdot C \in R_q$ and return message $M = C' \bmod p$.

Fig. 1. The encryption scheme $\mathtt{pNE}(n, q, p, \sigma, \alpha)$

the choice of q prime (versus q a power of 2), the choice of distributions for f, g as restricted discrete Gaussians (versus sparse binary polynomials), and the extra error term pe in encryption $C = phs + pe + M$ (versus $C = phs + M$).

We will need a variant of pNE with message space B a large subset of $R_p = \mathbb{Z}_p[x]/(x^n + 1)$ such that $b_1 - b_2$ is invertible in R_p for *all* $b_1 \neq b_2$ in B. If $x^n+1 = \prod_{i=1}^r f_i \bmod p$ denotes the factorization of x^n+1 into irreducibles f_i over $\mathbb{Z}_p$, then by the Chinese Remainder Theorem, a polynomial $b \in \mathbb{Z}_p[x]/(x^n + 1)$ is invertible in R_p if and only if it is coprime to f_i over $\mathbb{Z}_p$ for all $i = 1, \ldots, r$. The following lemma shows how to choose p such that $r = 2$ and f_1, f_2 are both irreducibles of degree $n/2$. This allows us to take $B = \{b \in R_p : \deg(b) < n/2\}$.

Lemma 2 ([4]). *If $n = 2^k$ with $k \geq 2$ and p is a prime with $p \equiv 3 \bmod 8$, then $x^n + 1 = f_1 f_2 \bmod p$ where each f_i is irreducible in $\mathbb{Z}_p[x]$ and can be written $f_i = x^{n/2} + t_i x^{n/4} - 1$ with $t_i \in \mathbb{Z}_p$.*

Our generalized Peikert-Waters construction of IND-CCA2 encryption from lossy trapdoor functions uses the following Generalized Leftover Hash lemma.

Lemma 3 ([8]). *Suppose that random variable X on $\{0,1\}^n$ has min-entropy ℓ_x and random variable Y (that may depend on X) has at most 2^{ℓ_y} possible values. Let $\mathcal{H}$ be a family of universal hash functions from $\{0,1\}^n$ to $\{0,1\}^\ell$ with $\ell_x - (\ell_y + \ell) \geq 2 \log 1/\epsilon$ for some $\epsilon > 0$. Then the statistical distance between $(h, h(X), Y)$ (for h chosen uniformly from $\mathcal{H}$) and (h, r, Y) (for h chosen uniformly from $\mathcal{H}$ and r chosen uniformly and independently from $\{0,1\}^\ell$) is at most ϵ.*

2.3 ABO Lossy Trapdoor Functions

We recall the definition of ABO Lossy Trapdoor Functions [26].

Definition 1. *An ABO Lossy Trapdoor Function Family $\mathcal{F} = (\mathsf{KG}_\mathcal{F}, \mathsf{F}, \mathsf{F}^{-1})$ is a collection of three polynomial time algorithms:*

- ***Key Generation algorithm** $\mathsf{KG}_\mathcal{F}$: On input 1^n (for a security parameter $n \in \mathbb{N}$), and a lossy* branch $b^* \in B$ *(B denotes the branch space), the probabilistic algorithm $\mathsf{KG}_\mathcal{F}$ outputs a public/secret key pair (pk, sk).*

- ***Evaluation algorithm*** F*: On input public key* pk*,* $x \in X$ *(*X *denotes the function input space) and branch* $b \in B$*, the deterministic algorithm* F *returns an output* $y = \mathsf{F}(pk, b, x) \in Y$ *(where* Y *denotes the output space).*
- ***Inversion algorithm*** F^{-1}*: On input* $y \in Y$*,* $b \in B$ *and secret key* sk*, the deterministic algorithm* F^{-1} *returns* $x = \mathsf{F}^{-1}(sk, b, y) \in X \cup \{\perp\}$ *(where* $\perp$ *indicates an inversion failure).*

These algorithms satisfy the following properties, for some parameters $\delta \in (0, 1)$ *(failure probability) and* $\rho \in (0, 1)$ *(lossiness leakage rate):*

- δ***-Inversion Correctness:*** *For any* $b^* \in B$*, except with negligible probability* $\leq \delta$ *over the key pair* (sk, pk) *output by* $\mathsf{KG}_{\mathcal{F}}(n, b^*)$*, we have* $\mathsf{F}^{-1}(sk, b, \mathsf{F}(pk, b, x)) = x$ *for all* $x \in X$ *and* $b \in B \setminus \{b^*\}$*.*
- ρ***-Lossiness (with failure probability*** δ***):*** *For any* $b^* \in B$*, except with negligible probability* $\leq \delta$ *over the key pair* (sk, pk) *output by* $\mathsf{KG}_{\mathcal{F}}(n, b^*)$*, the size of the image set* $\{y \in Y : \exists x \in X \text{ with } y = \mathsf{F}(pk, b^*, x)\}$ *is at most* $|X|^{\rho}$*.*
- (T, ϵ) ***Lossy Branch Hiding:*** *The advantage of any* T*-time (for* $T = \mathcal{P}oly(n)$*) attacker* $\mathcal{A}$ *in distinguishing between the following two experiments* $\mathbf{Exp}(0)$ *and* $\mathbf{Exp}(1)$ *is a negligible function* ϵ *of the security parameter* n*. For* $i \in \{0, 1\}$*, the experiment* $\mathbf{Exp}(i)$ *is defined as follows. On input* 1^n*,* $\mathcal{A}$ *outputs a pair of branches* $b_0^*, b_1^* \in B$*. Then* $\mathsf{KG}_{\mathcal{F}}$ *is run on input* $(1^n, b_i^*)$*, returning a key pair* (pk, sk)*, and* $\mathcal{A}$ *is given* pk*.*

Remark 1. In our definition of ρ-lossiness, ρ is an upper bound on the *leakage rate* of the lossy branch, i.e. the fraction of the input min-entropy that is leaked by the output.

3 An ABO Lossy Trapdoor Function from pNE

3.1 Modifying pNE for Full Randomness Recovery in Decryption

The decryption algorithm for the provable NTRUEncrypt variant pNE from [34] only recovers the encrypted message M but not the randomness (s, e) used to encrypt M. For constructing the ABO trapdoor function that is used in our NTRUCCA scheme, we need an additional randomness recovery algorithm that can also recover the randomness (s, e). In this section, we show how to modify the scheme pNE to achieve this, while preserving its security reduction. It turns out that most of the tools we need in this section have been worked out in [32] for the purpose of analyzing the NTRUSign signature scheme, and we only need to slightly tweak them for our application.

Our main observation for constructing a randomness recovery algorithm for pNE is that, after M is recovered by the decryption algorithm and $C' = p^{-1} \cdot (C - M) = h \cdot s + e$ is computed, we have:

$$\begin{bmatrix} C' \\ 0 \end{bmatrix} = \begin{bmatrix} h \\ -1 \end{bmatrix} \cdot s + \begin{bmatrix} e \\ s \end{bmatrix}.$$

The vector $\boldsymbol{c} = [C', 0]^T \in R_q^2$ is in the form of an (Ring) LWE instance $\boldsymbol{c} = A \cdot \boldsymbol{s} + \boldsymbol{e}$ over the ring R_q, where $A = [h, -1]^T \in R_q^{2\times 1}$ and $\boldsymbol{e} = [e, s]^T \in R^2$ is 'small'. Thus, given a *full trapdoor matrix* $T \in R^{2\times 2}$ for the matrix A over R, (i.e. the entries of T have 'small' coefficients, $T \cdot A = 0 \bmod q$ and T has full rank over the field $K = \mathbb{Q}[x]/(x^n + 1)$), the randomness $\boldsymbol{e}$ can be recovered by standard techniques for LWE inversion [11,24,33], namely one can compute $T \cdot \boldsymbol{c} \bmod q = T \cdot \boldsymbol{e} \bmod q = T\boldsymbol{e}$, where the last equality holds over K, since $\|T \cdot \boldsymbol{e}\|_\infty < q/2$ when $\|T\|$ and $\boldsymbol{e}$ are sufficiently small. Since T has full rank over K, T^{-1} exists over K, and $\boldsymbol{e}$ can be recovered from $\boldsymbol{e} = T^{-1} \cdot (T \cdot \boldsymbol{c} \bmod q)$. Note that since the secret key polynomials f, g satisfy $f \cdot h - g = 0 \bmod q$, the vector $[f, g]^T$ can serve as the first row of the trapdoor matrix T. In designing their signature scheme, the `NTRUSign` authors [12] give a heuristic algorithm to compute another small pair $(F, G) \in R^2$ such that $F \cdot h - G \bmod q$, which is linearly independent of $[f, g]$ over K. A variant of this algorithm, that we call TrapKG, is presented and analyzed rigorously in [32]. In [32], the algorithm TrapKG is applied for obtaining a provably secure variant of `NTRUSign`. Here, we apply it to obtain a provably secure variant of `pNE` with full randomness recovery. For our application, one does not need to store the full trapdoor matrix T. Indeed, from the above description of the decryption process, it is clear that one need only store (f, F) and a low precision approximation $\widetilde{T}$ to T^{-1} The algorithm TrapKG is shown in Fig. 2. To

Inputs: $n, q, p \in \mathbb{Z}$, $\sigma, \eta \in \mathbb{R}$.
Output: A key pair (sk, pk).

1. Sample f' from $D_{\mathbb{Z}^n,\sigma}$; if $(f \bmod q) \notin R_q^\times$ or $(f \bmod p) \notin R_p^\times$, resample.
2. Sample g from $D_{\mathbb{Z}^n,\sigma}$; if $(g \bmod q) \notin R_q^\times$, resample.
3. If $\|f\| > \sqrt{n} \cdot \sigma$ or $\|g\| > \sqrt{n} \cdot \sigma$, restart.
4. If ideal $\langle f, g \rangle \neq R$, restart.
5. Compute $F_1, G_1 \in R$ such that $fG_1 - gF_1 = 1$; $F_q := qF_1$, $G_q := qG_1$.
6. Use Babai's nearest plane algorithm to approximate (F_q, G_q) by an integer linear combination of $(f, g), (xf, xg), \ldots, (x^{n-1}f, x^{n-1}g)$.

Let $(F, G) \in R^2$ be the output with $(F, G) = (F_q, G_q) - k(f, g)$ and $k \in R$.

7. If $\|(F, G)\| > n\sigma$, restart.
8. Compute $T = \begin{bmatrix} f & g \\ F & G \end{bmatrix}$.
9. Compute $\widetilde{T} \in K^{2\times 2}$, an approximation to T^{-1} (over K) with precision η. (i.e. the entries of matrix $\widetilde{T} - T^{-1}$ have infinity norm at most η).
10. Return secret key $sk = (f, F, \widetilde{T})$, $pk = h \stackrel{\text{def}}{=} g/f \in R_q^\times$.

Fig. 2. Full Trapdoor Key Generation Algorithm TrapKG (adapted from [32])

obtain a high efficiency for our `NTRUCCA` scheme, we will choose $p = n^{\theta(1)}$, versus the choice $p = O(1)$ used in `pNE`. To obtain a tighter security reduction with this choice, we dropped the restriction $f = 1 \bmod p$ used in `pNE`. Instead, we sample f from a Gaussian (as in the `NTRUSign` variant of [32]), but here we must reject and resample f if it is not invertible mod q or mod p.

Lemma 4 (Adapted from Lemma 4.1 in [32]). *Let $n \geq 8$, $q \geq 5$ and $p = 3 \bmod 8$. Let $\sigma \geq \sqrt{n \ln(2n(1+1/\delta))/\pi} \cdot q^{1/2}$, for an arbitrary $\delta \in (0, 1/2)$. Let $a \in R$ and $p \in R_q^\times$. Then $\Pr_{f \hookleftarrow D_{\mathbb{Z}^n,\sigma}}[f \notin R_q^\times \cap R_p^\times] \leq n(1/q + 2\delta) + 2 \cdot (1/q^{n/2} + 2\delta)$.*

The algorithm TrapKG in Fig. 2 differs from the NTRUSign key generation algorithm analyzed in [32] only in the extra rejection step for f if $f \notin R_p$. Using the above Lemma 4 (in place of Lemma 4.1 of [32]) to evaluate the rejection probability in the proof of Lemma 4.4 of [32] gives the following performance result for this algorithm.

Theorem 2 (Adapted from [32], Th. 4.2). *Suppose $q \geq 256n$ and p is a prime with $p = 3 \bmod 8$. Let $\varepsilon \in (0, 1/2)$ and $\sigma \geq \max(2n\sqrt{\ln(8nq)} \cdot q^{\frac{1}{2}+2\varepsilon}, \omega(n^{1.5}\log^5 n))$. Then the algorithm of Fig. 2 terminates in expected polynomial time, and $T \cdot [h, -1]^T = 0 \bmod q$. Furthermore, we have $\|f\|, \|g\| \leq \sqrt{n}\sigma$ and $\|F\|, \|G\| \leq n\sigma$. Finally, if n is sufficiently large, the distribution of the returned h is rejected with probability $c < 1$ for some absolute constant c from a distribution whose statistical distance from $U(R_q^\times)$ is $\leq 2^{3n} q^{-\lfloor \varepsilon n \rfloor}$.*

Our pNE variant with randomness recovery, called pNErr, is shown in Fig. 3. The decryption algorithm for pNErr requires an additional multiplication by $f_p^{-1} \bmod p$ during decryption (versus pNE) since in pNErr we have dropped the restriction $f = 1 \bmod p$.

Key generation. Given input parameters (n, q, p, σ, η), run algorithm TrapKG of Fig. 2 on input (n, q, p, σ, η) and return $sk = (f, F, \widetilde{T})$, $pk = h \stackrel{\text{def}}{=} g/f \in R_q^\times$.
Encryption. Given message $M \in \mathcal{P}$, set $s, e \hookleftarrow \chi_\alpha$ and return ciphertext $C = p \cdot (hs + e) + M \in R_q$.
Decryption. Given ciphertext C and secret key $(f, F, \widetilde{T})$, compute $C' = f \cdot C \in R_q$ and return message $M = f_p^{-1} C' \bmod p$, where f_p^{-1} denotes the multiplicative inverse of f in R_p.
Randomness Recovery. Given ciphertext C, message M and secret key $(f, F, \widetilde{T})$, compute $C' = p^{-1} \cdot (C - M) \in R_q$, $t_e = fC' \in R_q$ and $t_s = FC' \in R_q$, and $[e, s]^T = \lceil \widetilde{T} \cdot [t_e, t_s]^T \rfloor \in R^2$, where $\lceil \cdot \rfloor$ denotes rounding coordinate-wise to the nearest integers. Return (s, e).

Fig. 3. The encryption scheme $\texttt{pNErr}(n, q, p, \sigma, \alpha, \eta)$

Conditions on the scheme parameters that guarantee correctness of decryption and randomness recovery are summarized in the following Lemma. Note that we gain a factor $\|p\|$ over the bounds in [32] due to dropping the condition $f = 1 \bmod p$.

Lemma 5. *If $\omega(\sqrt{n}\log n)\alpha p\sigma < 1$, the decryption algorithm of* pNErr *recovers M with probability $1 - n^{-\omega(1)}$ over the choice of s, e, f, g. If the conditions*

$\omega(n \log n)\alpha\sigma < 1$ *and* $\eta < \frac{1}{mnq}$ *hold, then the randomness recovery algorithm of* pNErr *recovers* (s, e) *with probability* $1 - n^{-\omega(1)}$ *over the choice of* s, e, f, g.

As in [32], the security of the scheme follows from the invertibility of p in R_q, and the hardness of the decisional Ring LWE problem in R_q with h uniform in $R_q^\times$. Here we also have to deal with the additional fact that h is sampled from a distribution that is rejected with constant probability from an almost uniform distribution on $R_q^\times$ (by Theorem 2).

Lemma 6. *Suppose* $q \geq 256n$ *and* $p \in R_q^\times$ *is a prime with* $p = 3 \bmod 8$. *Let* $\varepsilon \in (0, 1/2)$ *and* $\sigma \geq \max(2n\sqrt{\ln(8nq)} \cdot q^{\frac{1}{2}+2\varepsilon}, \omega(n^{1.5} \log^5 n))$. *If there exists an IND-CPA attack against* ***pNErr*** *that runs in time* T *and has success probability* $1/2+\delta$, *then there exists an algorithm solving* R-LWE$_{\alpha,q}$ *that runs in time* $T' = O(n \cdot \bar{\delta}^{-2} \cdot T)$ *and has advantage* $\delta' \geq \bar{\delta}^2/4 - 2^{-\Omega(n)}$, *where* $\bar{\delta} = (1 - c) \cdot \delta - q^{-\Omega(n)}$ *and* $c < 1$ *is the rejection constant from Theorem 2.*

3.2 Our ABO Lossy Trapdoor Function

Outline. We now use the pNErr scheme to construct an ABO Lossy Trapdoor function. Our construction uses as a starting point the paradigm underlying the constructions presented in [26]. In this paradigm, one starts with an encryption scheme E that is homomorphic with respect to addition and multiplication by known messages, i.e. given a ciphertext $c = E(b)$ for message b, and two messages b_1 and b_2, then $c' = b_1 \cdot E(b) + b_2$ is a ciphertext for the message $b' = b_1 \cdot b + b_2$. Given such an encryption scheme E, for a desired lossy branch b^*, the ABO key generation algorithm computes ciphertext $pk = E(b^*)$ as the public key (with the decryption key as the trapdoor), and on input a message x and branch b, the function evaluation algorithm computes $F(pk, b, x) = x \cdot (pk - b) = x \cdot E(b^* - b) = E(x \cdot (b^* - b))$. Thus, when evaluating F on the lossy branch ($b = b^*$), we just have $F(pk, b, x) = E(0)$, a ciphertext of a zero message independent of x, and we may hope that $F(pk, b^*, x)$ indeed loses at least some information on x, whereas for $b \neq b^*$, we have $F(pk, b, x) = E(x \cdot (b^* - b))$, which allows recovery of x if the mapping $x \mapsto (b^* - b) \cdot x$ is injective. Unfortunately, this idea does not immediately work for pNErr. On the positive side, the pNErr scheme has the desired homomorphic properties. Namely, given a ciphertext $c = h \cdot s + pe + M \in R_q$ for a message $M \in R_p$ and two messages $M_1, M_2 \in R_p$, we have that $M_1 \cdot c + M_2 = h \cdot (M_1 s) + p(M_1 e) + (M_1 M + M_2)$ is a valid ciphertext for $M_1 M + M_2 \bmod p$, assuming that $M_1 s$ and $M_1 e$ are chosen small enough compared to q. The problem is that the resulting function evaluated on a lossy branch i.e. $y = F(pk, b^*, x) = x \cdot (pk - b) = x \cdot (hs + pe)$, is not lossy, indeed it is injective with high probability. This is because $pk - b$ may be invertible in R_q, and even if it is not, one can recover x with high probability from $x \cdot s$ and $x \cdot e$, where the latter two can be recovered from $y = x \cdot (hs + pe) = h \cdot (xs) + p(xe)$ and h using the randomness recovery algorithm of pNErr.

Our solution to the lossiness problem of the above construction uses the observation that pNErr is in fact additively homomorphic with respect to addition

of two *ciphertexts*, not just with respect to addition of a known message to a ciphertext, i.e. given ciphertexts $E(b_1)$ and $E(b_2)$ for messages b_1, b_2 respectively, $E(b_1) + E(b_2)$ is a ciphertext for the message $b_1 + b_2$. This means that we can modify the function evaluation algorithm to add an encryption of the zero message without hurting message recovery for injective branches, i.e. we can use the function evaluation $y = F(pk, b, (x, \bar{s}, \bar{e})) = x \cdot (pk - b) + (h\bar{s} + p\bar{e}) = h(xs + \bar{s}) + p(xe + \bar{e}) + x(b^* - b)$, where $h\bar{s} + p\bar{e}$ is a random ciphertext for the zero message. Note that y is still an encryption of $x(b^* - b)$ as before, allowing recovery of x by decryption for injective branches. But the additional randomness of $\bar{s}, \bar{e}$ masks the x-dependent terms xe and xs in y for evaluation of F on the branch $b = b^*$, making this branch lossy, as required, assuming the masking terms $\bar{s}, \bar{e}$ are sufficiently large. Of course, since F must be a *deterministic* algorithm, the masking terms $\bar{s}, \bar{e}$ now become part of the function input (along with x), and must be recoverable by the ABO's inversion algorithm F^{-1} for injective branches $b \neq b^*$. For the latter, note that once x is recovered (by the decryption algorithm), then we can recover the added ciphertext of zero, namely $y - x(pk - b) = h\bar{s} + p\bar{e}$ and use the randomness recovery algorithm of pNErr to obtain $\bar{s}, \bar{e}$.

Construction. Our ABO construction $\mathcal{F}_{\texttt{NTRU}}$ is shown in Fig. 4. We give conditions for ABO inversion correctness in Lemma 7. Unlike Lemma 5 for pNErr, which is only valid probabilistically over the randomness of the encryption algorithm, our definition of ABO inversion correctness requires that, except for a set of keys of negligible probability, inversion succeeds for *all* valid outputs of F. This is used in the CCA security proof, to prevent attacks that choose outputs that make the inversion fail in one game but not the other.

Lemma 7 (Inversion Correctness). *If* $\alpha q > \sqrt{n}$, $\eta < \frac{1}{mnq}$, *and* $q > \max(p^2 \cdot \omega(n^2\sqrt{\log n}) \cdot \alpha q \cdot \sigma + 2p\bar{p} \cdot n \cdot \sigma + p^2 \cdot n^2 \cdot \sigma, \bar{p} \cdot n^{1.5} \cdot \sigma)$ *(resp.* $q > \max(2p^2 \cdot n^{2.5} \cdot \alpha q \cdot \sigma + 2p\bar{p} \cdot n \cdot \sigma + p^2 \cdot n^2 \cdot \sigma, \bar{p} \cdot n^{1.5} \cdot \sigma)$*), then* $\mathcal{F}_{\texttt{NTRU}}$ *satisfies* $n^{-\omega(1)}$*-Inversion Correctness (resp.* $2^{-\Omega(n)}$*-Inversion Correctness).*

Proof. Any output $y = \mathsf{F}((h, c), b, (x, \bar{s}, \bar{e}))$ of F has the form of a pNErr ciphertext $y = p \cdot (hs' + e') + (b^* - b)x$ for message $(b^* - b) \cdot x$, with $s' = sx + \bar{s}$ and $e' = ex + \bar{e}$ being the ciphertext randomness. By the choice of p and Lemma 2, $(b^* - b)_p^{-1}$ exists. A sufficient condition for successful recovery of x is that $\|C'\|_\infty < q/2$, where $C' = p(gs' + fe') + f(b^* - b)x$. The Cauchy-Schwarz inequality gives $\|gs'\|_\infty \leq \|g\| \cdot \|s'\|$. From Theorem 2, we have $\|g\| \leq \sqrt{n}\sigma$, while Lemma 1 says that $\|sx\| \leq p \cdot \omega(n^{1.5}\sqrt{\log n}) \cdot \alpha q$ (resp. $\|sx\| \leq p \cdot n^2 \cdot \alpha q$) for every $x \in R_p$, except with probability $\leq n^{-\omega(1)}$ (resp. $\leq 2^{-\Omega(n)}$) over the choice of s during key generation. Since $\|\bar{s}\| \leq \sqrt{n}\bar{p}$, it follows that $\|pgs'\|_\infty \leq p^2 \cdot \omega(n^2\sqrt{\log n}) \cdot \alpha q \cdot \sigma + p\bar{p} \cdot n \cdot \sigma$ (resp. $\|pgs'\|_\infty \leq p^2 \cdot n^{2.5} \cdot \alpha q \cdot \sigma + p\bar{p} \cdot n \cdot \sigma$). The same argument gives the same bound on $\|pfe'\|_\infty$. Finally, applying Cauchy-Schwarz again, we have $\|f(b^* - b)x\|_\infty \leq \sqrt{n} \cdot \|f\| \cdot \|b^* - b\| \cdot \|x\| \leq p^2 \cdot n^2 \cdot \sigma$. This implies $\|C'\|_\infty < q/2$ by the assumed lower bound on q.

The inversion algorithm succeeds to recover $(\bar{s}, \bar{e})$ if $\|T \cdot [\bar{e}, \bar{s}]^T\|_\infty = \|[f\bar{e} + g\bar{s}, F\bar{e} + G\bar{s}]^T\|_\infty < q/2$ and $\eta < \frac{1}{mnq}$. Using the bounds $\|f\|, \|g\|, \|F\|, \|G\| \leq n\sigma$

- **Key generation $\mathsf{KG}_{\mathcal{F}_{\mathtt{NTRU}}}$:** Given as input 1^n, primes q, p, integer $\bar{p}$ and reals α, σ, η and $b^* \in B$ (where $B = \{b \in R_p : \deg(b) < n/2\}$ denotes the branch space), run the key generation algorithm of `pNErr` on input $(1^n, q, p, \sigma, \alpha, \rho)$ to obtain a public key $h = gf^{-1} \in R_q^\times$ and a secret key $(f, F, \widetilde{T})$ for `pNErr`. Return $pk = (h, c = p{\cdot}(hs{+}e){+}b^* \in R_q)$, where $s, e \hookleftarrow \chi_\alpha$ and $sk = (f, F, \widetilde{T})$.
- **Evaluation algorithm F:** Given as input public key $pk = (h, c) \in R_q^2$, branch $b \in B$ and function input $(x, \bar{s}, \bar{e}) \in X$ (where $X = R_p \times R_{\bar{p}}^2$ denotes the input space), return $y = \mathsf{F}((h, c), b, (x, \bar{s}, \bar{e})) = (c{-}b){\cdot}x{+}p{\cdot}(h\bar{s}{+}\bar{e}) \in R_q$.
- **Inversion algorithm F^{-1}:** Given as input $y \in R_q$, $b \in B$ and secret key $sk = (f, F, \widehat{T})$:
  - Use the decryption algorithm of `pNErr` to decrypt ciphertext y with secret key f to recover message $x \in R_p$ (i.e. compute $y' = f \cdot y \in R_q$ and $x = (b^* - b)_p^{-1} \cdot f_p^{-1} \cdot y' \bmod p$, where $(b^* - b)_p^{-1}$ and f_p^{-1} denote multiplicative inverses of f and $b^* - b$, respectively, in R_p).
  - Compute $y'' = y - (c - b) \cdot x \in R_q$ and use the randomness recovery algorithm of `pNErr` to recover randomness $(\bar{s}, \bar{e})$ from ciphertext y'' with message 0 and secret key sk (i.e. compute $t_e = fp^{-1}y'' \in R_q$ and $t_s = Fp^{-1}y'' \in R_q$, $[\bar{e}, \bar{s}]^T = \lceil \widetilde{T} \cdot [t_e, t_s]^T \rfloor \in R^2$, where $\lceil{\cdot}\rfloor$ denotes rounding coordinate-wise to the nearest integers).
  - Return $(x, \bar{s}, \bar{e})$.

Fig. 4. The ABO Lossy Trapdoor Function Family $\mathcal{F}_{\mathtt{NTRU}}(n, q, p, \bar{p}, \sigma, \alpha, \eta)$

from Theorem 2, and $\|\bar{e}\|, \|\bar{s}\| \le \bar{p}\sqrt{n}$, the Cauchy-Schwarz inequality gives $\|f\bar{e} + g\bar{s}\|_\infty, \|F\bar{e} + G\bar{s}\|_\infty \le \bar{p}n^{1.5}\sigma < q/2$, by the assumed condition on q, as required. □

We now analyze the lossiness of $\mathcal{F}_{\mathtt{NTRU}}$.

Lemma 8 (Lossiness). *If $\bar{p} > p \cdot \omega(n\sqrt{\log n}) \cdot \alpha q$ (resp. $\bar{p} > 2p \cdot n^{1.5} \cdot \alpha q + 1$), then $\mathcal{F}_{\mathtt{NTRU}}$ satisfies ρ-Lossiness with failure probability $n^{-\omega(1)}$ (resp. $2^{-\Omega(n)}$), with $\rho \le \frac{\log(4\bar{p}^2)}{\log(p\bar{p}^2)}$.*

Proof. For evaluation on the lossy branch b^*, the function output is $h \cdot (xs + \bar{s}) + p(xe + \bar{e})$. Hence the number of possible outputs N is upper bounded by $(2B{+}1)^{2n}$, where B is an upper bound on $\|xs{+}\bar{s}\|_\infty$ and $\|xe{+}\bar{e}\|_\infty$. By Lemma 1, we have $\|xs\|_\infty \le p \cdot \omega(n\sqrt{\log n}) \cdot \alpha q$ (resp. $\|xs\|_\infty \le p \cdot n^{1.5} \cdot \alpha q$) for all $x \in R_p$ except with probability $\le n^{-\omega(1)}$ (resp. $2^{-\Omega(n)}$) over the choice of $s \hookleftarrow \chi_\alpha$ in key generation, and $\|\bar{s}\|_\infty \le \bar{p}/2$. The same bounds also hold for $\|xe\|_\infty$ and $\|\bar{e}\|_\infty$, respectively. Using the condition on $\bar{p}$, we have $2B + 1 \le \bar{p}$, and since $|X| = (p\bar{p}^2)^n$, we get the stated bound on ρ. □

Note that the bound on the leakage rate ρ of $\mathcal{F}_{\mathtt{NTRU}}$ in Lemma 8 is (since $\log \bar{p} > \log p + O(\log n)$) greater than $1 - \frac{\log p}{3 \log p + O(\log n)} > 2/3$.

The lossy branch hiding property follows directly from the IND-CPA security of the underlying pNErr encryption scheme, which in turn is as hard as the Ring-LWE problem, by Lemma 6.

Lemma 9 (Lossy Branch Hiding). *If there exists an attack against the lossy branch hiding of $\mathcal{F}_{\text{NTRU}}$ that runs in time T and has distinguishing advantage ϵ, then there exists an IND-CPA attack against pNErr with run time T and success probability at least $1/2 + \epsilon/2$.*

4 The NTRUCCA Scheme

4.1 Generalized Peikert-Waters Construction

Outline. The Peikert-Waters construction [26] of IND-CCA2 encryption from ABO lossy trapdoor functions uses a pair of ABO lossy trapdoor functions F_1 and F_2[1]. The ciphertext contains $F_1(b, x)$ and $F_2(b, x)$ for a random x that is hashed to obtain a key with which to mask the message. The security proof relies on the assumption that for the lossy branch $b = b^*$, the pair $(F_1(b^*, x), F_2(b^*, x))$ does not leak all the information on x. If both F_1 and F_2 have leakage rate ρ on their lossy branch b^*, then the leakage rate of the pair $(F_1(b^*, x), F_2(b^*, x))$ is at most 2ρ, so to ensure that not all the information on x is leaked, we must have $\rho < 1/2$. Unfortunately, the leakage rate of our ABO $\mathcal{F}_{\text{NTRU}}$ is greater than $2/3$, so $\mathcal{F}_{\text{NTRU}}$ cannot be directly used in this construction.

Instead, we show that the Peikert-Waters construction generalizes to use ciphertexts containing $k \geq 2$ ABO evaluations $F^{(k)}(x_1, \ldots, x_k) \stackrel{\text{def}}{=} (F_1(b, x_1), \ldots, F_k(b, x_k))$, where $F_1, \ldots, F_k$ denote k ABO functions, and the evaluation points $(x_1, \ldots, x_k)$ sampled from a $(k-1)$-of-k Subset Reconstructible Distribution ($\mathcal{SRD}_{k-1,k}$), in which any subset of $k-1$ of the x_i's suffices to uniquely reconstruct all x_i's (the Peikert-Waters construction corresponds to the case $k = 2$). The advantage of using the $\mathcal{SRD}_{k-1,k}$ distribution for $k > 2$, as first observed by Mol and Yilek [22], is that the min-entropy of the $\mathcal{SRD}_{k-1,k}$ distribution when sampled with a Reed-Solomon code is $(k-1)\log|X|$ versus the $\leq k\rho \log|X|$ leaked min-entropy, implying that the leakage rate of $F^{(k)}$ on the lossy branch $b = b^*$ with input distribution $\mathcal{SRD}_{k-1,k}$ is $\rho^{(k)} \leq \frac{k}{k-1} \cdot \rho$. Hence by using a sufficiently large k, one can make $\rho^{(k)}$ exceed ρ by an arbitrarily small amount. In particular, starting with $\rho \approx 2/3$ as in our ABO, a constant $k \geq 4$ suffices for our scheme, so the ciphertext length only incurs a constant factor overhead over the length of a single ABO output (which corresponds to a single NTRU ciphertext).

We remark that Mol and Yilek applied the k-product one-way function $F^{(k)}$ to the IND-CCA2 encryption scheme of Rosen and Segev [30], that requires $F^{(k)}$ to be one-way under the $SRD_{k-1,k}$ distribution. The advantage of our generalized Peikert-Waters scheme over Rosen-Segev when the underlying functions

[1] Actually only F_2 needs to be an ABO lossy trapdoor function, whereas F_1 can be just a plain lossy trapdoor function.

F_i are lossy, is that in our scheme the only lower bound constraint on k comes from the requirement that $F^{(k)}$ is lossy (which for our ABO $\mathcal{F}_{\mathtt{NTRU}}$, can be satisfied with a constant $k = O(1)$), whereas in the Rosen-Segev scheme, k is also lower bounded by the security parameter (because in Rosen-Segev, k is lower bounded by the public key length of a one-time signature scheme, or at least the length of a collision-resistant hash of the public key). Thus, starting from ρ-lossy ABO functions F_i, our generalized Peikert-Waters scheme yields shorter ciphertexts than Rosen-Segev by a factor $\Omega((1-\rho)\cdot n)$, where n denotes the security parameter.

Construction. Figure 5 shows our generalized Peikert-Waters scheme GPW_k, parameterized by an integer k. We use an ABO lossy trapdoor function family $\mathcal{F} = (\mathsf{KG}_{\mathcal{F}}, \mathsf{F}, \mathsf{F}^{-1})$ with function input space X and branch space B, which is ρ-lossy. As in the Peikert-Waters scheme, we also use a strongly unforgeable one-time signature scheme $\mathsf{OTS} = (\mathsf{OTS.KG}, \mathsf{OTS.Sign}, \mathsf{OTS.Ver})$ with public key space P. We assume for convenience that $P \cup \{b_0\} \subseteq B$, for some branch $b_0 \notin P$ (if $|P| > |B|$, we can hash a key in P into $P' \subset B$ using a collision-resistant hash function). We also use a family $\mathcal{H}$ of universal hash functions from X^k to $\{0,1\}^\ell$. We assume that we have efficient algorithms $\mathsf{Samp}_{k-1,k}$ and $\mathsf{Rec}_{k-1,k}$ for, respectively, sampling from the distribution $\mathcal{SRD}_{k-1,k}$ over X^k, and reconstructing x_j from $\{x_i\}_{i\neq j}$ for any $(x_1,\ldots,x_k)$ output by $\mathsf{Samp}_{k-1,k}$ and any $j \in [k]$, and that the min-entropy of $\mathcal{SRD}_{k-1,k}$ is $\mu \geq (k-1)\log X$ (as mentioned above, the latter assumption can be satisfied using Shamir's secret sharing scheme [22]).

Key generation. Given input parameters 1^n and k, run algorithm $\mathsf{KG}_{\mathcal{F}}$ k times on input $(1^n, b_0)$ to get k independent key pairs (pk_i, sk_i) $(i \in [k])$ for ABO lossy trapdoor function family $\mathcal{F}$, all having lossy branch b_0. Sample a hash function $h \hookleftarrow \mathcal{H}$. Return key pair (pk, sk) with secret key $sk = (sk_1,\ldots,sk_{k-1})$ and public key $pk = (pk_1,\ldots,pk_k,h)$.
Encryption. Given public key $pk = (pk_1,\ldots,pk_k,h)$ and message $M \in \{0,1\}^\ell$, run $\mathsf{OTS.KG}$ to generate a one-time signature key pair (sk_S, pk_S). Sample $(x_1,\ldots,x_k) = \mathsf{Samp}_{k-1,k}$ and for $i \in [k]$, compute $y_i = \mathsf{F}(pk_i, pk_S, x_i)$ (i.e. use branch pk_S for all k evaluations). Compute $C = M \oplus h(x_1,\ldots,x_k)$, and $\sigma = \mathsf{Sign}(sk_S, (y_1,\ldots,y_k,C))$. Return ciphertext $c = (pk_S, y_1,\ldots,y_k, C, \sigma)$.
Decryption. Given ciphertext $c = (pk_S, y_1,\ldots,y_k,C,\sigma)$ and secret key $sk = (sk_1,\ldots,sk_{k-1})$, check that $\mathsf{OTS.Ver}(pk_S,(y_1,\ldots,y_k,C),\sigma) = Acc$. If not, return $\perp$. Compute $x_i = \mathsf{F}^{-1}(sk_i, pk_S, y_i)$ for $i \in [k-1]$. Compute $x_k = \mathsf{Rec}_{k-1,k}(x_1,\ldots,x_{k-1})$. If $x_i \in X$ and $\mathsf{F}(pk_i, pk_S, x_i) = y_i$ for all $i \in [k]$ then return $M = C \oplus h(x_1,\ldots,x_k)$. Else, return $\perp$.

Fig. 5. The generalized Peikert-Waters encryption scheme GPW_k

The security of the scheme is summarized by Theorem 3, a quantitative generalization of Theorem 4.2 in [26] (the latter is the special case $k = 2$).

Theorem 3. *Suppose there exists an* IND-CCA2 *attack* $\mathcal{A}$ *against the* GPW_k *encryption scheme of Fig. 5, that runs in time* T *and has success probability* $1/2 + \varepsilon$, $\mathcal{F}$ *satisfies* δ*-correctness and* ρ*-lossiness, the min-entropy* $\mu \geq (k-1) \cdot \log|X|$, *and* $k \geq \frac{1}{1-\rho} \cdot \left(1 + \frac{2n+\ell}{\log|X|}\right)$. *Let* $\varepsilon' = \varepsilon - 2k\delta - 2^{-n}$. *Then, at least one of the following attacks exist:*

- *An attack* $\mathcal{A}_s$ *against the strong existential unforgeability of* OTS *with run-time* $T_s = T$ *and success probability* $\varepsilon_s \geq \frac{\varepsilon'}{k+1}$.
- *An attack* $\mathcal{A}_h$ *against the lossy branch hiding property of* $\mathcal{F}$, *with run-time* $T_h = T$ *and distinguishing advantage* $\varepsilon_h \geq \frac{\varepsilon'}{k+1}$.

A Simpler IND-CCA2 KEM. For encrypting long messages efficiently, one typically uses a hybrid IND-CCA2 encryption scheme, combining an IND-CCA2 *Key Encapsulation Mechanism* (KEM) with an efficient IND-CCA2 symmetric encryption scheme [6]. The encryption algorithm of a KEM takes as input the public key and a security parameter, and returns a uniformly random key K in the key space $\{0,1\}^\ell$ and ciphertext c for K. The above construction can be simplified in the KEM setting, replacing the one-time signature scheme in the above scheme by a collision-resistant hash function family $\mathcal{G}$ mapping X^k to $B_\mathcal{G} \subseteq B$, i.e. the branch pk_S encryption is replaced by $b = g(x_1, \ldots, x_k)$ where $g \in \mathcal{G}$ is the hash function in the public key. The decryption algorithm checks that $b = g(x_1, \ldots, x_k)$ (here X and B denote the input and branch space, respectively, of the ABO lossy trapdoor function family). The security result is only slightly modified to account for the extra leakage by b on $(x_1, \ldots, x_k)$. We call the resulting scheme GPWKEM_k (see full paper for a detailed definition).

Theorem 4. *Suppose there exists an* IND-CCA2 *attack* $\mathcal{A}$ *against the* GPWKEM_k *KEM that runs in time* T *and has success probability* $1/2+\varepsilon$, $\mathcal{F}$ *satisfies* δ*-correctness and* ρ*-lossiness,* $\mu \geq (k-1) \cdot \log|X|$, *and* $k \geq \frac{1}{1-\rho} \cdot \left(1 + \frac{2n+\ell+\log|B_\mathcal{G}|}{\log|X|}\right)$. *Let* $\varepsilon' = \varepsilon - 2k\delta - 2^{-n}$. *Then, at least one of the following attacks exist:*

- *An attack* $\mathcal{A}_c$ *against the collision-resistance of hash family* $\mathcal{G}$ *with run-time* $T_c = T$ *and success probability* $\varepsilon_c \geq \frac{\varepsilon'}{k+1}$.
- *An attack* $\mathcal{A}_h$ *against the lossy branch hiding property of* $\mathcal{F}$, *with run-time* $T_h = T$ *and distinguishing advantage* $\varepsilon_h \geq \frac{\varepsilon'}{k+1}$.

4.2 Instantiation and Choice of Parameters

Our NTRUCCA scheme is defined as the GPW_k scheme with the following instantiation choices, in terms of n, the security parameter. We let $\varepsilon, \varepsilon_p > 0$ denote positive constants (independent of n) that one may adjust to trade-off the scheme's concrete performance. The constant ε controls the uniformity of the NTRU key h (its statistical distance from uniform over R_q is at most $2^{3n} q^{-\varepsilon \cdot n}$, by Theorem 2). The constant ε_p controls the size of the ABO branch space B (its size is $|B| = p^{n/2}$). The procedure we use for choosing parameters is as follows. We choose $\alpha q = \theta(n^{1.5})$ to satisfy worst-case reduction condition against $2^{o(n)}$-time

attacks, by Theorem 1. Next, setting $p = n^{\varepsilon_p}$, we choose $\bar{p} = p \cdot \omega(n^{1.5} \log n\alpha q)$, the condition in lossiness Lemma 8. Then, we plug the condition on σ from Lemmas 9 and 6 in the condition on q from Lemma 7. This determines our choice of q and σ and η, and then we can determine from αq and q the value of α^{-1} and hence the resulting γ-Ideal-SVP approximation factor.

- **ABO Trapdoor Function Family $\mathcal{F}$:** We use $\mathcal{F}_{\mathtt{NTRU}}(n, q, p, \bar{p}, \sigma, \alpha, \eta)$ from Sec. 3.2 with the following parameters:
  - $q = \widetilde{\Theta}\left(n^{\frac{\max(5.5+\varepsilon_p, 5+2\varepsilon_p)}{1/2-2\varepsilon}}\right)$, $p = n^{\varepsilon_p}$, $\bar{p} = \widetilde{\Theta}\left(n^{3+\varepsilon_p}\right)$.
  - $\sigma = \widetilde{\Theta}\left(n^{1+\max(5.5+\varepsilon_p, 5+2\varepsilon_p)\cdot\frac{1/2+2\varepsilon}{1/2-2\varepsilon}}\right)$.
  - $\alpha^{-1} = \widetilde{\Theta}\left(n^{\frac{\max(5.5+\varepsilon_p, 5+2\varepsilon_p)}{1/2-2\varepsilon}-1.5}\right)$.
  - $\eta^{-1} = \widetilde{\Theta}(nq)$.

  Note that this choice of parameters implies:
  - $\mathcal{F}_{\mathtt{NTRU}}$ leakage rate, $\rho \leq 1 - \frac{1-\frac{2}{\log p}}{1+2\frac{\log \bar{p}}{\log p}} \leq 1 - \frac{1}{3+6\varepsilon_p^{-1}+o(1)}$ (By Lemma 8).
  - $\mathcal{F}_{\mathtt{NTRU}}$ input entropy, $\log |X| = n\cdot(\log p + 2\log \bar{p}) = (3\varepsilon_p + 6 + o(1))\cdot n \log n$.
  - $k = \left\lceil 3 + \varepsilon_p^{-1} \cdot \left(6 + \frac{2+\ell/n}{\log n}\right) + o(1) \right\rceil$. ($k = 4$ is possible with $\ell = \theta(n \log n)$).
  - Worst-Case IdSVP Approximation Factor, $\gamma = O(n^{2.5}\alpha^{-1})$.
- **One-Time Signature Scheme OTS:** We use the One-Time Signature scheme of [18]. It operates on vectors of dimension $m_{ots} \geq 2$ over the ring $R_{q_{ots}} = \mathbb{Z}_{q_{ots}}[x]/(x^n_{ots}+1)$, with a public key of length $(m_{ots}+2)\cdot n_{ots} \log q_{ots}$ and a signature of length $\leq m_{ots} \cdot n_{ots} \log q_{ots}$. We instantiate it with:
  - $m_{ots} = 2$.
  - $q_{ots} = \Theta(n^5_{ots} \log^{5+\varepsilon'} n_{ots})$.
  - Worst-case IdSVP Approximation Factor, $\gamma_{ots} = O(n^4_{ots} \log^3 n_{ots})$.
  - $n_{ots} \leq \frac{n \log p}{8 \log q_{ots}} = \Theta(n)$. (Note this implies that the verification key length is $\leq B$).
- **Universal Hash Family $\mathcal{H}$:** We use a random linear mapping from $GF(2^\ell)^{k'}$ to $GF(2^\ell)$, where:
  - $k' = \frac{\log |X|}{\ell} = O(1)$. (This means that the key length of $\mathcal{H}$ is $O(n \log n)$ and evaluating it costs $\widetilde{O}(k'\ell) = \widetilde{O}(n)$ time).
- $\mathsf{Samp}_{k-1,k}$ **and** $\mathsf{Rec}_{k-1,k}$: We use three Reed-Solomon codes (one over $GF(p^n)$ and two over $GF(\bar{p}^n)$) to implement $\mathsf{Samp}_{k-1,k}$ for encoding $x \in R_p$ and $\bar{s}, \bar{e} \in R_{\bar{p}}$, and we use Lagrange interpolation to implement $\mathsf{Rec}_{k-1,k}$. Both can be done in time $\widetilde{O}(n)$.

Overall, we obtain our main asymptotic result.

Corollary 1. *If there exists an attack against the* IND-CCA2 *security of* `NTRUCCA` *with run-time $T = 2^{o(n)}$ and success probability $2^{-o(n)}$, then there exists a quantum algorithm with run-time $2^{o(n)}$ against the γ-IdSVP problem with $\gamma = \widetilde{\Theta}\left(n^{1+\frac{\max(5.5+\varepsilon_p, 5+2\varepsilon_p)}{1/2-2\varepsilon}}\right)$. The scheme has key and ciphertext size of $O(n \log n)$ and encryption and decryption computation time of $\widetilde{O}(n)$.*

Note that with the current state of the art, the best quantum attack against $\mathcal{P}oly(n)$-IdSVP takes time $2^{\Omega(n)}$, so with this assumption, the above results says that for any constant $0 < \varepsilon < 1/2$, and $\varepsilon_p > 0$, the time required to break the IND-CCA2 security of NTRUCCA is $2^{\Omega(n)}$.

5 Conclusions

We constructed the first asymptotically efficient IND-CCA2 secure variant of the NTRUEncrypt encryption scheme, with a provable security from worst-case problems in ideal lattices. Although the efficiency overhead of our scheme over the IND-CPA scheme of [34]) amounts to only a constant factor, this factor could in practice be quite significant. An interesting direction for future work is to construct provably secure variants of NTRUEncrypt which have a smaller constant overhead factor close to 1 (as well as reducing the constant overhead of [34] over the original heuristic NTRUEncrypt scheme).

Acknowledgements. We thank Damien Stehlé for helpful discussions. S. Ling, H. Wang and C. Tartary gratefully acknowledge the hospitality of the Dept. of Computing, Macquarie University, during their research visits. The research of R. Steinfeld and J. Pieprzyk was supported by an Australian Research Fellowship (ARF) from the Australian Research Council (ARC), and ARC Discovery Grants DP0987734 and DP110100628. Research of S. Ling and H. Wang was supported in part by the National Research Foundation of Singapore under Research Grant NRF-CRP2-2007-03. C. Tartary's research was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61033001, 61073174. C. Tartary also acknowledges support from the Danish National Research Foundation and the National Natural Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the Theory of Interactive Computation (CTIC) within which part of this work was performed. C. Tartary's work was also financed by the International Young Scientists program of the Natural Science Foundation of China (61050110147 and 61150110344).

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient Lattice (H)IBE in the Standard Model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010)
2. Agrawal, S., Boneh, D., Boyen, X.: Lattice Basis Delegation in Fixed Dimension and Shorter-Ciphertext Hierarchical IBE. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 98–115. Springer, Heidelberg (2010)
3. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Proc. of the 1st CCS, pp. 62–73. ACM Press (1993)
4. Blake, I.F., Gao, S., Mullin, R.C.: Explicit factorization of $x^{2^k} + 1$ over f_p with prime $p \equiv 3 \mod 4$. App. Alg. in Eng., Comm. and Comp. 4, 89–94 (1992)
5. Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM J. Comput. 36(5), 1301–1328 (2007)

6. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM J. Comput. 33, 167–226 (2003)
7. NTRU Cryptosystems. Technical reports (2002), http://www.ntru.com
8. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. SIAM J. Comput. 38(1), 97–139 (2008)
9. Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More Constructions of Lossy and Correlation-Secure Trapdoor Functions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 279–295. Springer, Heidelberg (2010)
10. Fujisaki, E., Okamoto, T.: How to Enhance the Security of Public-Key Encryption at Minimum Cost. In: Imai, H., Zheng, Y. (eds.) PKC 1999. LNCS, vol. 1560, pp. 53–68. Springer, Heidelberg (1999)
11. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC 2008, pp. 197–206. ACM (2008)
12. Hoffstein, J., Howgrave-Graham, N., Pipher, J., Silverman, J.H., Whyte, W.: NTRUSIGN: Digital Signatures Using the NTRU Lattice. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 122–140. Springer, Heidelberg (2003)
13. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A Ring-Based Public Key Cryptosystem. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 267–288. Springer, Heidelberg (1998)
14. Howgrave-Graham, N., Silverman, J.H., Singer, A., Whyte, W.: NAEP: Provable security in the presence of decryption failures. Technical report, Cryptology ePrint Archive (2003), http://eprint.iacr.org/2003/172
15. IEEE P1363. Standard specifications for public-key cryptography, http://grouper.ieee.org/groups/1363/
16. Langlois, A., Stehlé, D.: Hardness of decision (r)lwe for any modulus. Cryptology ePrint Archive, Report 2012/091 (2012), http://eprint.iacr.org/2012/091
17. Lyubashevsky, V., Micciancio, D.: Generalized Compact Knapsacks Are Collision Resistant. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006, Part II. LNCS, vol. 4052, pp. 144–155. Springer, Heidelberg (2006)
18. Lyubashevsky, V., Micciancio, D.: Asymptotically Efficient Lattice-Based Digital Signatures. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 37–54. Springer, Heidelberg (2008)
19. Lyubashevsky, V., Peikert, C., Regev, O.: On Ideal Lattices and Learning with Errors over Rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (2010)
20. Micciancio, D.: Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. Comput. Complexity 16(4), 365–411 (2007)
21. Micciancio, D., Peikert, C.: Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. Cryptology ePrint Archive, Report 2011/501 (2011), http://eprint.iacr.org/2011/501; In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012)
22. Mol, P., Yilek, S.: Chosen-Ciphertext Security from Slightly Lossy Trapdoor Functions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 296–311. Springer, Heidelberg (2010)
23. Nguyên, P.Q., Pointcheval, D.: Analysis and Improvements of NTRU Encryption Paddings. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 210–225. Springer, Heidelberg (2002)

24. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem. In: STOC 2009, pp. 333–342. ACM (2009)
25. Peikert, C., Rosen, A.: Efficient Collision-Resistant Hashing from Worst-Case Assumptions on Cyclic Lattices. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 145–166. Springer, Heidelberg (2006)
26. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: STOC 2008, pp. 187–196 (2008)
27. Perlner, R.A., Cooper, D.A.: Quantum resistant public key cryptography: a survey. In: IDtrust, pp. 85–93. ACM (2009)
28. Rackoff, C., Simon, D.R.: Non-interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)
29. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM 56(6) (2009)
30. Rosen, A., Segev, G.: Chosen-Ciphertext Security via Correlated Products. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 419–436. Springer, Heidelberg (2009)
31. Stam, M.: A Key Encapsulation Mechanism for NTRU. In: IMA Int. Conf., pp. 410–427 (2005)
32. Stehlé, D., Steinfeld, R.: Making NTRU as Secure as Worst-Case Problems Over Ideal Lattices. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 27–47. Springer, Heidelberg (2011), http://web.science.mq.edu.au/~rons
33. Stehlé, D., Steinfeld, R., Tanaka, K., Xagawa, K.: Efficient Public Key Encryption Based on Ideal Lattices. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 617–635. Springer, Heidelberg (2009)
34. Stehlé, D., Steinfeld, R.: Making NTRU as Secure as Worst-Case Problems Over Ideal Lattices. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 27–47. Springer, Heidelberg (2011)

Generating Provable Primes Efficiently on Embedded Devices

Christophe Clavier[1], Benoit Feix[1,2], Loïc Thierry[2,⋆], and Pascal Paillier[3]

[1] XLIM, University of Limoges,
christophe.clavier@unilim.fr
[2] INSIDE Secure
bfeix@insidefr.com, thierry.loic@hotmail.fr
[3] CryptoExperts
pascal.paillier@cryptoexperts.com

Abstract. This paper introduces new techniques to generate provable prime numbers efficiently on embedded devices such as smartcards, based on variants of Pocklington's and the Brillhart-Lehmer-Selfridge-Tuckerman-Wagstaff theorems. We introduce two new generators that, combined with cryptoprocessor-specific optimizations, open the way to efficient and tamper-resistant on-board generation of provable primes. We also report practical results from our implementations. Both our theoretical and experimental results show that constructive methods can generate provable primes essentially as efficiently as state-of-the-art generators for probable primes based on Fermat and Miller-Rabin pseudo-tests. We evaluate the output entropy of our two generators and provide techniques to ensure a high level of resistance against physical attacks. This paper intends to provide practitioners with the first practical solutions for fast and secure generation of provable primes in embedded security devices.

Keywords: Prime Numbers, Pocklington's theorem, Public Key Cryptography, Embedded Software, Modular Exponentiation, Cryptographic Accelerators, Primality Proving.

1 Introduction

Large prime numbers are a basic ingredient of keys in several standardized primitives such as RSA [21], Digital Signature Algorithm (DSA) [12] or Diffie-Hellman key exchange (DH) [10]. This paper precisely addresses the generation of provable prime numbers in embedded, crypto-enabled devices.

When it comes to RSA key generation, two approaches coexist: key pairs may be generated off-board (i.e. out of the device) in a secure environment such as a certified Hardware Security Module (HSM) running in a personalization center, and loaded into devices afterwards. Key pairs may also be generated on-board,

⋆ Part of this work was carried out when the author was doing his Master's thesis at Inside Secure.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 372–389, 2012.

that is, by the device itself. In this case the private key cannot be compromised as it is never transmitted to the outside world. This capability also allows the device to generate new keys later on, when deployed in the field. However it implies that the device must be able to generate large primes very efficiently and in a side-channel-secure manner.

Surprisingly enough, in spite of a quite abundant literature on primality testing and on the validation of provable primes, research works that specifically suggest generators for embedded devices are pretty inexistant. Commonly found prime number generators rely on primality (pseudo-)tests to provide a high level of confidence that the output number is prime. It is widely known that this confidence level can be increased arbitrarily by applying sufficiently many iterations of the Miller-Rabin test [12].

Technical requirements for the generation of prime numbers well-suited for RSA, DSA and ECDSA are described in industry standards such as FIPS 186-3 [12]. To ensure compliance, generating a 1024-bit DSA prime number requires as many as 40 Miller-Rabin iterations, which can be reduced to 3 when performing an additional Lucas test. However carrying out a Lucas test is more costly on an embedded device than a single modular exponentiation, and thus leads to a performance loss. This paper investigates another approach, namely the application of constructive techniques to achieve truly provable primality.

In this paper, we introduce two efficient methods for generating provable primes and present fast implementations of these methods on a popular smartcard cryptoprocessor. Our methods rely on Pocklington's theorem and an extended result due to Brillhart, Lehmer and Selfridge. We establish bounds on the entropy of the output distribution of each method and provide evidence that both of them are secure and can be used for cryptographic purposes. Performance measurements are given that demonstrate the efficiency of our algorithms and how they compare with probable prime generation. We also suggest a number of countermeasures against state-of-the-art side-channel and fault-based analysis to ensure security in an untrusted environment.

Roadmap. Section 2 recalls the usual methods for primality testing, where we distinguish between probabilistic and true tests. Generation algorithms for provable primes are discussed in Section 3, where we introduce our two efficient constructive methods. The security of these methods in terms of output entropy is discussed in Section 4. Practical results are reported in Section 5 together with performance comparisons for smartcard implementations of our probable prime and provable prime generators. Section 6 addresses threats arising from side-channel attacks and shows how to adapt our algorithms to resist these. We conclude in Section 7.

2 Prime Number Generation Based on Primality Testing

In the broadest possible sense, a primality test $\top$ is a procedure that outputs a guess $\top(n) \in \{\mathsf{true}, \mathsf{false}\}$ as to whether a positive integer n is prime or composite. It can be a pseudo-primality test (also called compositeness test), in which case

the guess can be a false positive with some probability, or a true primality test that never fails and provides a proof for primality when positively answered. Once one is given some primality test $\top$, it is natural to derive Algorithm 2.1 which provides a generic method for generating prime numbers.

Alg. 2.1. Generic Prime Number Generation

Input: a primality test $\top$, a constraining property $\mathcal{P}$
Output: a prime integer n

1. generate a random candidate n verifying property $\mathcal{P}$
2. **while** $\top(n) =$ **false do**
3. update n while preserving property $\mathcal{P}$
4. **return** n

Following the naming of Brandt and Damgård [18], we refer to the list of tested candidates as the *search sequence*. In the generic prime number generator, each candidate along the search sequence is required to verify some property $\mathcal{P}$. The purpose of this requirement is to reduce the average number of calls to $\top$, which is assumed to be the most time-consuming subroutine of the algorithm, by avoiding candidates known to be composite.

Without this requirement – or equivalently, when $\mathcal{P}$ is satisfied for any n – the average number of calls to $\top$ when generating an ℓ-bit prime is close to $\ln(2^\ell)$. An obvious improvement is to let $\mathcal{P}$ be the property that n is odd and proceed to updating a candidate by adding 2 to it. In that case the average number of calls to $\top$ drops to $\ln(2^\ell)/2$. A straightforward generalization of this idea is to take for $\mathcal{P}$ the property that n is relatively prime with the t smallest primes $p_1, \ldots, p_t$. The first candidate in the search sequence thus requires the generation of an invertible element modulo $\Pi = \prod_{i=1}^{t} p_i$, which can be done either with trial divisions by each of $p_1, \ldots, p_t$, using Chinese remaindering (*e.g.* Garner [13] or Gauss algorithms), or using a technique due to [16] based on Carmichael's theorem. Several methods can be applied to update n while preserving $\gcd(n, \Pi) = 1$; Π can simply be added to n, or one can keep track of an array of indicators $\omega_i = n \bmod p_i$ for $i = 1, \ldots, t$ and modular-add 2 to all of those until none is equal to zero. Alternately, an efficient method for preserving $\gcd(n, \Pi) = 1$ for maximally large Π is found in Joye et al. [15,16]. Overall, the techniques described in [15,16] provide the most efficient approach on a cryptoprocessor as they generate an invertible element modulo Π faster than the classical trial division method. Irrespective of the chosen methods to implement the different subroutines of Algorithm 2.1, the average number of calls to $\top$ is close to

$$N(\ell, \Pi) = \ln(2^\ell) \cdot \frac{\phi(\Pi)}{\Pi}$$

where ϕ is Euler's function. The optimal choice therefore consists in taking the largest possible prime product $\Pi = p_1 \cdot \cdots \cdot p_t$. While $N(\ell, \Pi)$ obviously further decreases with larger t, the relative gain rapidly decreases as well as Π becomes larger.

2.1 Pseudo-primality Tests

Pseudo-primality tests may erroneously view a composite number as being prime. Among these, Fermat and Miller-Rabin tests are the most commonly used in embedded applications as they are particularly fast and easy to implement. The random-base Miller-Rabin test has an error probability $\varepsilon < 1/4$. By iterating this test h times with different random bases this probability is (often quite loosely) upper bounded by $1/4^h$. Practitioners choose the number h of iterations depending on the bitsize of the tested number, the cryptosystem intended to make use of the generated prime, and the specific security requirements imposed by industry standards. Referring to FIPS 186-3, a 1024-bit prime to be used as a DSA parameter requires 40 Miller-Rabin tests (or 3 Miller-Rabin tests followed by a Lucas test). For a 2048-bit RSA key, each 1024-bit prime must pass 4 Miller-Rabin tests, and although applying the Lucas test is not required, it is highly recommended. The random-base Fermat test has approximately the same efficiency as the random-base Miller-Rabin test while its error probability is higher. However, it is more simple to implement and leads to optimally efficient pseudo-testing when using a base fixed to 2: modular multiplications by 2 can then be replaced with modular additions in the modular exponentiation $2^{n-1} \bmod n$. Fermat testing is usually performed first with $a = 2$, and only when n passes the Fermat test, does it undergo several Miller-Rabin rounds with random bases before being considered to be prime. This leads to the efficient prime number generator referred to as Algorithm 2.2, where $\mathsf{F}_a(n)$ and $\mathsf{MR}_a(n)$ respectively denote Fermat and Miller-Rabin tests with base a.

Alg. 2.2. Efficient Generation of Probable Primes

Input: a bitsize ℓ, $\Pi = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot p_t$, a confidence parameter h
Output: an ℓ-bit probable prime n

1. generate a random ℓ-bit integer n with $\gcd(n, \Pi) = 1$ and go to 3
2. update n such that $\gcd(n, \Pi) = 1$
3. if $\mathsf{F}_2(n) =$ **false** then go to 2
4. **for** $i = 1$ **to** h **do**
5. pick a base a at random from $[2, n-2]$
6. if $\mathsf{MR}_a(n) =$ **false** then go to 2
7. **return** n

Neglecting the probability that the output prime is a Fermat or a strong pseudoprime, and denoting respectively by T_i, T_u, T_{F_2} and T_{MR_a} the execution times of the routines for generating the first candidate, updating the current candidate and performing Fermat and Miller-Rabin tests, the average total execution time to generate a probable l-bit prime amounts to

$$T_{\text{probable}}(\ell) = T_i(\ell) - T_u(\ell) + N(\ell, \Pi) \cdot (T_u(\ell) + T_{\mathsf{F}_2}(\ell)) + h \cdot T_{\mathsf{MR}_a}(\ell)\ . \quad (1)$$

This generation method is among the most popular ones in use in the embedded security industry at the present time. Section 5 reports practical performance figures for a typical smartcard implementation of this generator.

2.2 True Primality Tests

Prime number generators make use of pseudo-primality tests because of their efficiency. However, to fully eliminate the error probability ε, one has to rely on true primality testing a.k.a. primality proving. The asymptotically fastest true primality test is the AKS method [1], which is the only known algorithm that runs in polynomial time. However, the preferred general-purpose method for testing large numbers is currently the Elliptic Curve Primality Proving test [4] which was used to ascertain the primality of the largest general number, a prime with more than 20′000 decimal digits. Unfortunately the AKS and ECPP methods are way too complex to be of any interest for embedded implementations, where algorithms are preferably based on simple arithmetic operations such as modular exponentiations.

A possible step in this direction relates to a deterministic variant of the Miller-Rabin criterion. Following a result from Ankeny [3], Bach [5] proved under the Extended Riemann Hypothesis (ERH) that any composite number n has a strong witness[1] upper bounded by $2\ln^2 n$. Thus, verifying that n passes Miller-Rabin testing for all bases smaller than $2\ln^2 n$ would actually prove that n is prime. The drawback of this approach is the fairly large amount of bases to consider before making sure that n is prime. Proving the primality of a 512-bit number would require more than 250′000 Miller-Rabin rounds. A secondary drawback is that the primality proof only holds under ERH.

Instead of relying on the existence of a small witness, it may be better to rely on the existence of a small set containing at least one witness. Given an upper bound x on candidates, a *reliable set* of witnesses is a set $\mathcal{W}$ such that every odd composite integer $n \leq x$ has a witness in $\mathcal{W}$. An interesting result from Alford et al. [2] unconditionally proves the existence of a reliable set containing at most $(6/5)\ln x$ integers smaller than x. This result does not rely on any conjecture and proves that n is prime with much fewer Miller-Rabin rounds (only 426 rounds for 512-bit numbers). Unfortunately the constructive method put forward by the authors for identifying such a reliable set does not seem to be computationally practical.

3 Constructive Generation of Provable Primes

As previously discussed, there does not seem to be any practical true primality test that would suit our context. Rather than testing the true primality of candidates along a search sequence, we revisit Maurer's approach [18] wherein provable primes are generated in a *constructive* manner using Pocklington's criterion:

Theorem 1 (Pocklington's theorem). *Let $n > 3$ be an odd integer, and let $n = rF + 1$ where the factorization of F is known as $F = \prod_{j=1}^{s} q_j^{e_j}$. If there exists an integer a such that*

[1] A *strong witness* for a composite number n is an integer a such that n does not pass the Miller-Rabin test with base a, thereby proving its compositeness.

(i) $a^{n-1} \equiv 1 \pmod{n}$ *and*
(ii) $\gcd(a^{(n-1)/q_j} - 1, n) = 1$ *for each* $j = 1 \ldots s$,

then every prime divisor p *of* n *is congruent to* 1 *modulo* F. *In particular, if* $F > \sqrt{n} - 1$ *then* n *is prime.*

As opposed to Fermat and Miller-Rabin's theorems, Pocklington's theorem isolates sufficient conditions for true primality. Unfortunately it cannot be used to test any given integer since the factorization of $n - 1$ must be partially known. Based on Pocklington's theorem, Maurer [18] suggested a constructive method for generating provable primes. The main idea there is to construct a prime n such that $n - 1$ is divisible by one or more smaller primes. A recursive use of the criterion then allows to generate larger primes at each round starting from small integers whose primality proof is trivial.

Theorem 2. *Let* p *be an odd prime, and* r *an integer such that* $r < p$. *Let* $n = 2rp + 1$.

(i) If there exists an integer a *with* $2 \leq a < n$ *such that* $a^{n-1} \equiv 1 \pmod{n}$ *and* $\gcd(a^{2r} - 1, n) = 1$ *then* n *is prime.*
(ii) If n *is prime, the probability that a random value* a *satisfies* $a^{n-1} \equiv 1 \pmod{n}$ *and* $\gcd(a^{2r} - 1, n) = 1$ *is* $1 - 1/p$.

A generation algorithm can be derived from Theorem 2 (i) by iteratively producing provable primes twice larger at each iteration. Maurer proposed an iterative (and recursive) provable generation method based on this approach [19]. This iterative method requires precomputing and storing the intermediate bitsize of all provable primes from the highest to the lowest. In Maurer's algorithm, the number of iterations is variable and depends on a parameter r which is computed in order to provide the best output entropy. The main drawback of this implementation is that it is not efficient enough and therefore not suited to embedded implementations.

3.1 The Square Root Method

We now show how to generate provable primes more efficiently using Theorem 2 with fixed bitsizes for intermediate primes. We generate a provable prime by doubling at each iteration the size of the current prime p to derive the new prime $n = 2rp + 1$. While the entropy of this approach – estimated later in the paper – is not as optimized as in Maurer's algorithm, this offers a more suitable and efficient algorithm in embedded environments.

The intermediate prime sizes can be seen as equivalent to those in Maurer's algorithm when fixing $r = 0.5$. An iterative and recursive method relying on this idea – doubling each time the size of primes – was also proposed by Shawe-Taylor in [22] before Maurer's publication and is recommended by the NIST [12] to generate provable primes for public key schemes. The first algorithm we propose can therefore be seen as an adaptation of the Shawe-Taylor method, which also

relies on Pocklington's theorem. As opposed to Shawe-Taylor, our algorithm is not recursive but directly generates the primes iteratively from the smallest to the largest and many additional optimizations are put forward to improve efficiency.

Initialization. Before making use of Pocklington's theorem, one starts the generation with a first prime with initial bitsize l_0. In his algorithm, Maurer suggests generating the first prime (which is 20-bit long in the best case) using Erathostene's sieve. Our approach here is different and applies the Miller-Rabin criterion to generate initial primes up to 2^{32}. Indeed, Pomerance et al. [20] and Jaeschke [14] have proven that any number lesser[2] than 2^{32} is proven prime if it successfully passes the Miller-Rabin test with the three bases 2, 7 and 61. Making use of this trick, we obtain the algorithm $\mathsf{InitGenPrime}(\ell_0)$ (given in Appendix A). We define the bitsize of the initial prime as

$$\ell_0 = \min_{k>0} \left\{ \left\lceil \frac{\ell_n - 1}{2^k} \right\rceil + 1 \text{ such that } \left\lceil \frac{\ell_n - 1}{2^{k-1}} \right\rceil + 1 > 32 \right\} .$$

As indicated previously, we make use of $\mathsf{InitGenPrime}(\ell_0)$ to generate the initial prime p for any given size ℓ_0 lesser than 32. To illustrate the different steps of our method, Table 1 gives for different bitsizes ℓ_n, the initial prime size ℓ_0, the number k of iterations of Pocklington's theorem, and the intermediate prime sizes ℓ_i at each iteration.

Table 1. Intermediate bitsizes (ℓ_0 and ℓ_i) and number k of iterations

ℓ_n	k	ℓ_0	ℓ_1	ℓ_2	ℓ_3	ℓ_4	ℓ_5	ℓ_6	ℓ_7
512	5	17	33	65	129	257	512	-	-
768	5	25	49	97	193	385	768	-	-
1024	6	17	33	65	129	257	513	1024	-
2048	7	17	33	65	129	257	513	1025	2048

In order to reduce the number of Fermat tests throughout the generation, we apply the same idea as in the generation of probable primes: we get rid of candidates n which are not coprime to a product Π of the smallest primes. We thus obtain the provable prime generator presented as Algorithm 3.1.

Selection and Update of r and n. A first solution for finding a suitable r at Step 10 of Algorithm 3.1 consists in randomly selecting a first value $r \in [I+1, 2I]$, setting $n = 2rp + 1$, and then incrementing r by 1 and n by $2p$ until the modular residues $(\omega_i = n \bmod p_i)_{i=1,\ldots,t}$ are all non zero. Each ω_i is then incremented by $2p \bmod p_i$. An efficient trick consists in obtaining the values $2p \bmod p_i$ by doubling modulo p_i the residues ω_i of the previous iteration since the previous value of n corresponds to the new value of p in the current iteration.

[2] More precisely, the exact bound is $4'759'123'141$.

Alg. 3.1. Efficient-Square-Root-Generation(ℓ_n)

Input: a bitsize ℓ_n, $\Pi = 3 \cdot 5 \cdot \ldots \cdot p_t$
Output: an ℓ_n-bit provable prime n

1. $\ell \leftarrow \ell_n$
2. **while** $\ell > 31$ **do**
3. $\quad \ell \leftarrow \ell/2$
4. $\ell \leftarrow \ell + 1$
5. $n \leftarrow \mathsf{GenInitPrime}(\ell)$ [compute the initial small prime]
6. **while** $\ell < \ell_n$ **do**
7. $\quad p \leftarrow n$
8. $\quad \ell \leftarrow \min(2\ell - 1, \ell_n)$
9. $\quad I \leftarrow \lfloor \frac{2^{\ell-1}}{2p} \rfloor$
10. $\quad$ Select r at random from $[I + 1, 2I]$ such that $n \leftarrow 2rp + 1$ is coprime to Π and go to 12
11. $\quad$ Update r in $[I + 1, 2I]$ such that $n \leftarrow 2rp + 1$ is coprime to Π
12. $\quad$ **if** $\ell < 129$ **then**
13. $\quad\quad$ pick an integer a at random from $[2, n - 2]$
14. $\quad$ **else**
15. $\quad\quad a \leftarrow 2$
16. $\quad$ if $a^{n-1} \bmod n \neq 1$ then go to 11
17. $\quad$ if $\gcd(a^{2r} - 1, n) \neq 1$ then go to 11
18. **return** n

At Step 11, the same incremental update of r and n is applied for generating the next candidate coprime to Π.

A second solution consists in generating n simultaneously compliant with Pocklington's property (an even multiple of p plus one) and coprime to Π. This is done by first selecting r as $(x - (2p)^{-1} \bmod \Pi)$ where x is randomly selected from $\mathbb{Z}^{\star}_{\Pi}$ using the technique of [15] based on Carmichael's function. Then r is added to a random multiple of Π so that it lies in $[I + 1, 2I]$, and the first candidate n is computed as $2rp + 1$. Doing so, n is constructively coprime to Π. At Step 11, the next candidate is computed in the same vein from the updated value $x \leftarrow p_{t+1} \cdot x \bmod \Pi$.

Fixing $a = 2$ in Fermat Testing. From Theorem 2 (ii), we know that the probability that a random value a rejects a prime n at Step 16 or 17 is $1/p$. Assuming that the fraction of rejected primes does not vary much from one value of a to another, choosing a constant value a has a negligible impact on the distribution of the generated primes when the bitsize ℓ is sufficiently large. For instance when generating a 128-bit prime number $n = 2rp + 1$ from a 65-bit provable prime p, less than $1/2^{64}$ of the primes would never be reached. We accept this negligible loss of entropy and use $a = 2$ for the Fermat test when $\ell > 128$. This leads to faster exponentiations for steps 16 and 17 where modular multiplications by the base can be replaced with modular additions.

Estimated Performance. Denoting respectively by T_{init}, T_I, T_u, T_{F_a} and T_g the execution times taken by the initialization, computing I, updating the candidate n, the Fermat test with base a and the gcd computation, the total average execution time of Algorithm 3.1 amounts to

$$T_{\text{provable}}(\ell_n) = T_{\text{init}}(\ell_0) + \sum_{i=1}^{k} \left(T_I(\ell_i) + N(\ell_i, \Pi) \cdot (T_u(\ell_i) + T_{\mathsf{F}_a}(\ell_i)) + T_g(\ell_i)\right) . \quad (2)$$

We report experimental results from our smartcard implementation of this prime number generator in Section 5. Note that the value $N(\ell_i, \Pi)$ equals the average number of primality tests in the generation of probable primes for ℓ_i-bit integers coprime to Π. Also, as expected, we observed in our simulations that only one gcd is computed per ℓ_i-bit prime so that its execution time is almost negligible compared to the overall execution time.

3.2 The Cube Root Method

Our second method relies on (what we refer to) as the Cube Root Theorem put forward by Brillhart, Lehmer and Selfridge in 1970. More details on this result can be found in [6].

Theorem 3 (Brillhart-Lehmer-Selfridge-Tuckerman-Wagstaff [6]). *Let $n > 3$ be an odd integer, let $n = rF + 1$ where F is completely factored and $gcd(F, r) = 1$. Suppose there exists an integer a such that*

(i) $a^{n-1} \equiv 1 \pmod{n}$,
(ii) $\gcd(a^{(n-1)/q} - 1, n) = 1$ *for each prime factor q of F.*

Let $r = uF + s$, $1 \leq s < F$, and suppose $n < 2F^3 + 2F$, $F > 2$. If u is odd, or if u is even and $s^2 - 4u$ is not a perfect square, then n is prime.

As a corollary of Theorem 3, we derive the following result:

Theorem 4 (Cube Root Theorem). *Let p be an odd prime, $n = 2rp+1$ with r an integer such that $r < p^2 + 1$. If there exists an integer a with $2 \leq a \leq n$ such that*

(i) $a^{n-1} \equiv 1 \pmod{n}$ *and* $\gcd(a^{2r} - 1, n) = 1$,
(ii) $r = up + s$, $1 \leq s < p$ *for odd u,*

then n is prime.

Theorem 4 makes it possible to put together a prime number generator that iteratively produces provable primes three times larger at each iteration (instead of twice larger in the Square Root method). In order to speed-up the whole generation, we only consider cases where the quotient u is odd. This reduces the output entropy by one bit but has no significant impact on the security of cryptosystems such as RSA and DSA. To generate a provable prime of ℓ_n bits,

our algorithm starts with the generation of an initial prime p of ℓ_0 bits, where ℓ_0 is established as follows:

$\ell_0 \leftarrow \ell_n$
while $(\ell_0 > 31)$ $\quad \ell_0 \leftarrow \lfloor \ell_0/3 \rfloor + 1$

The generation of this ℓ_0-bit initial prime is performed as previously using the Miller-Rabin criterion and algorithm $\mathsf{InitGenPrime}(\ell_0)$ of Appendix A. The sizes ℓ_i of intermediate primes are displayed on Table 2.

Table 2. Intermediate sizes (ℓ_0 and ℓ_i) and number k of iterations

ℓ_n	k	ℓ_0	ℓ_1	ℓ_2	ℓ_3	ℓ_4
512	3	20	59	176	512	-
768	3	29	86	257	768	-
1024	4	14	41	122	365	1024
2048	4	26	77	230	689	2048

We then obtain the Cube Root prime number generator described in Algorithm 3.2.

Alg. 3.2. Efficient-Cube-Root-Generation(ℓ_n)

Input: a bitsize ℓ_n, $\Pi = 3 \cdot 5 \cdot \ldots \cdot p_t$
Output: an ℓ_n-bit provable prime n

1. $\ell \leftarrow \ell_n$
2. **while** $\ell > 31$ **do**
3. $\quad \ell \leftarrow \lfloor \ell/3 \rfloor$
4. $\ell \leftarrow \ell + 1$
5. $n \leftarrow \mathsf{GenInitPrime}(\ell)$ [compute the initial small prime]
6. **while** $\ell < \ell_n$ **do**
7. $\quad p \leftarrow n$
8. $\quad \ell \leftarrow \min(3\ell - 1, \ell_n)$
9. $\quad I \leftarrow \lfloor \frac{2^{\ell-1}}{2p} \rfloor$
10. $\quad$ Select r at random from $[I+1, 2I]$ such that $r = up + s$, $1 \le s < p$ for odd u and $n \leftarrow 2rp + 1$ is coprime to Π and go to 12
11. $\quad$ Update r in $[I+1, 2I]$ such that $r = up+s$, $1 \le s < p$ for odd u and $n \leftarrow 2rp+1$ is coprime to Π
12. $\quad$ **if** $\ell < 129$ **then**
13. $\quad\quad$ select a at random from $[2, n-2]$
14. $\quad$ **else**
15. $\quad\quad a \leftarrow 2$
16. $\quad$ if $a^{n-1} \bmod n \neq 1$ then go to 11
17. $\quad$ if $\gcd(a^{2r} - 1, n) \neq 1$ then go to 11
18. **return** n

Initial Selection and Update of r and n. A first solution for selecting a suitable r at Step 10 of Algorithm 3.2 is similar to the one used in the Square Root algorithm 3.1. An additional step is necessary that consists in computing u and s in $r = up + s$ in order to avoid candidates for which u is even.

Our second and most efficient solution for Step 10 consists in generating n in a constructive manner so that n is simultaneously compliant with Pocklington's requirement (an even multiple of p plus one), is coprime to Π and such that the quotient $u = \lfloor r/p \rfloor$ is forced to be odd. To this end, we keep track of an invertible element $x \in \mathbb{Z}_{\Pi}^{\star}$ which will serve as the residue of n modulo the prime product Π, and set $r = x - 1/(2p) \bmod \Pi$ to ensure that $n = 2xp \bmod \Pi$ is invertible modulo Π, so that the first two requirements are fulfilled. Now note that letting $r = up + s$, u is odd if and only if r and s have opposite parities. Therefore, if s is set to a fixed odd value throughout the search sequence, it is enough to ensure that r is even to force the parity of u to one. We now describe our method in more detail. Focusing on the search sequence associated with the i-th iteration, our generator proceeds as follows:

1. Fetch precomputed values $\Pi \leftarrow \Pi[i]$ and $\Lambda \leftarrow \Lambda[i]$ from code data. $\Pi \approx 2^{\ell_{i-1}-2}$ is a product of small odd primes (thereby excluding 2 from the factorization of Π), and Λ is the Carmichael function of Π.
2. Use [15] to generate a random invertible element $x \in \mathbb{Z}_{\Pi}^{\star}$, namely:
   (a) Randomly select x modulo Π
   (b) Compute $t = x^{\Lambda} \bmod \Pi$
   (c) If $t \neq 1$
      i. Randomly select z modulo Π
      ii. Update $x = x + z(1 - t) \bmod \Pi$
      iii. Goto 2b
3. Compute $1/(2p) = (2p)^{\Lambda-1} \bmod \Pi$ and derive $1/p \bmod \Pi$
4. Randomly select an odd value s modulo p
5. Use Chinese remaindering to compute $r \in [0, 2p\Pi]$ such that $r = x - 1/(2p) \bmod \Pi$, $r = s \bmod p$ and $r = 0 \bmod 2$. More precisely:
   (a) Compute $r_{\Pi p} = (((x - 1/(2p) - s)/p) \bmod \Pi) \cdot p + s$
   (b) Compute $r = (r_{\Pi p} \bmod 2) \cdot \Pi \cdot p + r_{\Pi p}$
   (c) Add appropriate multiple of $2p\Pi$ to r to get $r \in [I + 1, 2I]$

This concludes the initialization of the i-th loop *i.e.* the random selection of r at Step 10 at the i-th iteration. Updating r consists in just refreshing x as $x = 2x \bmod \Pi$ and performing a new round of Chinese remaindering as per Step 5 above. It is worthwhile noticing optimizations here: since p and s are fixed throughout the search sequence, the generator can just compute $1/p \bmod \Pi$ and $(-1/(2p) - s) \bmod \Pi$ once and for all and store these values. Step 5 then amounts to a couple of multiplications and additions. Also, modular exponentiations modulo Π are particularly efficient since Λ is small due to the particular form – extreme smoothness – of Π.

4 Estimating the Output Entropy

The rule for deriving at each iteration an ℓ_i-bit provable prime from an ℓ_{i-1}-bit other provable prime ($n \leftarrow 2rp + 1$) intrinsically generates primes p_i such that $p_i - 1$ is a multiple of a *half-size* prime p_{i-1}. This particular structure is not representative of the majority of prime integers, and obviously does not allow to generate them all. This section establishes the entropy of the output distribution of primes generated by Algorithms 3.1 and 3.2[3] and compare the output entropy with that obtained by a perfect generator that outputs uniformly random primes of a given bitsize ℓ_n.

Let us denote by R_{ℓ_i} the number of ℓ_i-bit primes that are attainable by the Square Root method at the end of iteration i. Note that any one of them can be uniquely derived from the sequence $(r_1, \ldots, r_i)$ of the values taken by r at each iteration. Since r is drawn at random, this suggests the heuristic approximation that the distribution of generated primes is uniform and that its entropy is equal to $H_{\ell_i} = \log_2(R_{\ell_i})$. According to Gauss's theorem, the number $\pi(x)$ of primes lesser than x is well approximated by $\frac{x}{\ln(x)}$ for large x. The number of exactly ℓ-bit primes can thus be estimated by

$$S_\ell = \frac{2^\ell}{\ln(2^\ell)} - \frac{2^{\ell-1}}{\ln(2^{\ell-1})} \, .$$

In an initial step, the algorithm randomly generates an ℓ_0-bit prime p_0, so that $R_{\ell_0} = S_{\ell_0}$. For $x \in \left[2^{\ell_{i-1}-1}, 2^{\ell_{i-1}}\right]$, consider an interval of width dx centered on x. Every p_{i-1} in this interval can generate $I = \lfloor \frac{2^{\ell_i - 1}}{2 \cdot p_{i-1}} \rfloor \simeq \frac{2^{\ell_i - 2}}{x}$ candidates among which $\frac{2^{\ell_i - 2}}{x \cdot \ln(2^{\ell_i})}$ are prime numbers[4]. The total number of primes – that can or cannot be reached by the generator – in the considered interval is $\frac{dx}{\ln(x)}$, but only a fraction

$$\frac{R_{\ell_{i-1}} \cdot \ln(2^{\ell_{i-1}})}{2^{\ell_{i-1}-1}}$$

of these can be generated at iteration $(i-1)$, so that the number of primes p_{i-1} to consider in the interval is

$$\frac{R_{\ell_{i-1}} \cdot \ln(2^{\ell_{i-1}}) \cdot dx}{2^{\ell_{i-1}-1} \cdot \ln(x)} \, .$$

Integrating over $\left[2^{\ell_{i-1}-1}, 2^{\ell_{i-1}}\right]$ the number of primes that each p_{i-1} can generate, we obtain

[3] Note that for efficiency purposes Algorithm 3.2 only selects r values for which $u = \lfloor \frac{r}{p} \rfloor$ is odd. In the sequel we first derive the entropy of our method when ignoring this trick. We subsequently address the effect of this feature later on.

[4] This derives from a commonly accepted approximation that the Chebotarëv density theorem also stands for large intervals. This theorem actually implies that for any coprime integers a and d, the proportion of primes less than x belonging to the arithmetic progression $\{a + nd\}_n$ tends to $\frac{1}{\phi(d)}$ when x tends to infinity.

$$\begin{aligned}\frac{R_{\ell_i}}{R_{\ell_{i-1}}} &\simeq \int_{2^{\ell_{i-1}-1}}^{2^{\ell_{i-1}}} \frac{\ln(2^{\ell_{i-1}}) \cdot 2^{\ell_i-2}}{2^{\ell_{i-1}-1} \cdot \ln(2^{\ell_i})} \cdot \frac{dx}{x \ln(x)} \\ &\simeq \frac{\ell_{i-1} \cdot 2^{\ell_i-2}}{\ell_i \cdot 2^{\ell_{i-1}-1}} \cdot \int_{2^{\ell_{i-1}-1}}^{2^{\ell_{i-1}}} \frac{dx}{x \ln x} \\ &\simeq \frac{\ell_{i-1}}{\ell_i} \cdot 2^{\ell_i-\ell_{i-1}-1} \cdot \big(\ln(\ell_{i-1}) - \ln(\ell_{i-1}-1)\big) \\ &\simeq \frac{\ell_{i-1}}{\ell_i} \cdot \frac{2^{\ell_i-\ell_{i-1}-1}}{\ell_{i-1}-1}\end{aligned}$$

whence

$$R_{\ell_n} = S_{\ell_0} \cdot \frac{\ell_0}{\ell_n} \cdot \frac{2^{\ell_n-\ell_0-k}}{\prod_{i=1}^{k}(\ell_{i-1}-1)} \tag{3}$$

where examples cases for k, ℓ_0 and ℓ_i are given in Tables 1 and 2.

As mentioned above, Equation (3) does not take into account that only half of the values for r are selected as prime candidates in Algorithm 3.2. Assuming that even and odd values of u are evenly distributed for r ranging from $I+1$ to $2I$, the effect of ignoring half of potential candidates is that every prime p_{i-1} in the neighborhood of x can generate only $\frac{2^{\ell_i-3}}{x \cdot \ln(2^{\ell_i})}$ primes. This results in the following expression for the number of l_n-bit primes generated by Algorithm 3.2 when only odd u values are selected:

$$R_{\ell_n} = S_{\ell_0} \cdot \frac{\ell_0}{\ell_n} \cdot \frac{2^{\ell_n-\ell_0-2k}}{\prod_{i=1}^{k}(\ell_{i-1}-1)} \,. \tag{4}$$

The estimated entropies H_{ℓ_n} provided by Algorithms 3.1 and 3.2 are given in Table 3 for different output bitsizes ℓ_n together with the entropy $H^*_{\ell_n}$ of a perfectly uniform distribution.

Table 3. Entropy loss w.r.t. ideal prime generation

ℓ_n	512	768	1024	1536	2048
$H^*_{\ell_n}$	503	758	1014	1525	2037
H_{ℓ_n} (Alg. 3.1, Eq. (3))	467	720	968	1476	1980
H_{ℓ_n} (Alg. 3.2, Eq. (4))	479	733	981	1490	2000

The entropy loss of the proposed prime generation ranges from 36 bits for 512-bit primes to 57 bits for 2048-bit primes for the Square Root method, and only from 24 to 37 bits for the Cube Root method. While somewhat larger than the entropy loss of about 4 bits found in Maurer's method, it is noticeable that it is small enough so that exhaustive search remains infeasible for currently secure bitsizes. We believe that the security of RSA and DSA cryptosystems is not (or only marginally) affected by using either Algorithm 3.1 or 3.2 for generating provable primes.

5 Implementation Results and Practical Aspects

5.1 On-board Generation of Probable Primes

Our implementation relies on an AT90SC chip supplied by Inside Secure embedding the Ad-X cryptoprocessor and the 8-bit AVR core both running at 30 MHz. The chip manufacturer provides a cryptographic toolbox for cryptography developers with all basic operations over large integers: modular multiplication, modular exponentiation, GCD, inversion, division, and so forth. The associated documentation provides estimated performances (cycle count) for these operations. Using this information we know the exact cycle count for any step of the generation algorithm. The exact average timings of our prime number generators can then be deduced on this component using Equation 1. Using the development kit from IAR running on a chip emulator loaded with the toolbox, the performance of our implementation of the generator for probable primes was experimentally confirmed to coincide perfectly with Equation 1.

The Fermat test with base 2 runs in 11 ms for a 512-bit integer n while the Miller-Rabin test with a random base is computed in 18 ms. We chose $t = 54$, so that Π is the product of small primes ranging from 2 to 251 and we choose $h = 3$ (the number of Miller-Rabin rounds).

On average, our generator outputs 512-bit probable primes in 580 ms ($N(512, \Pi) = 35.6$), 768-bit probable primes in 2′130 ms ($N(768, \Pi) = 53.4$) and 1024-bit probable primes in 5′780 ms ($N(1024, \Pi) = 71.2$).

5.2 Generating Provable Primes

Similarly, we deduced from Equation 2 the execution timings for our generator of provable primes on the same smartcard platform. We made use of the base-2 Fermat test when ℓ is greater than 128 bits, and took the same value for Π as in the case of probable primes. We have also implemented Algorithm 3.1 on the target chip. As a result, using the Square Root method to generate provable primes of respectively 512, 768 and 1024 bits requires on average 810, 2′580 and 5′940 ms. The Cube Root method decreases these figures to 760, 2′240 and 5′700 ms respectively.

5.3 Comparing Generators for Probable and Provable Primes

Given the expressions of $T_{\text{Prob}}(\ell)$ and $T_{\text{Provable}}(\ell)$, a rough guesstimate is that about the same number of modular exponentiations should be required to generate probable and provable primes of the same size, assuming trial divisions and identical values for Π. This is because the extra workload needed to generate the sequence of intermediate primes in the provable case remains fairly small compared to the resources needed to generate the full-length ℓ_n-bit provable prime. Moreover, this extra workload is somewhat compensated by the absence of final Miller-Rabin rounds or the Lucas test. All in all, we observe that the generation of a provable prime is slightly less efficient that the one of a probable

prime when only a few Miller-Rabin rounds are required. However, the Cube Root algorithm becomes the fastest option when either a significant amount of Miller-Rabin iterations or a Lucas test is needed.

Figure 1 provides performance measurements for the various generation methods discussed in the paper.

Bitlength ℓ_n	h	512	768	1024	1536	2048	Lucas test
Algorithm 2.2	3	**640**	**2130**	**5780**	**25700**	**74400**	yes
Algorithm 2.2	40	1170	3700	9290	36800	98900	no
Algorithm 3.1	-	**810**	**2580**	**5940**	**26500**	**75600**	provable
Algorithm 3.2	-	**760**	**2240**	**5700**	**24400**	**73550**	provable

Fig. 1. Time (in milliseconds) measurements for various prime number generators

We find that a Lucas test, as defined in FIPS 186-3, is roughly equivalent to 3.5 Miller-Rabin rounds and is therefore rather efficient on the AT90SC – comparatively to higher ratios found on other architectures. Overall, our experimental validation shows that the Cube Root method is essentially as efficient as the state-of-the-art generation algorithms for probable primes.

6 Achieving Leakage-Resistant Prime Number Generation

This section addresses side-channel attacks and ways to protect prime number generation from information leakage. Recent research works [11,8] have highlighted that prime number generation may be subject to power analysis. It is therefore necessary to ensure resistance against side-channels, especially when the device is operated in an untrusted environment. We give in this section a few guidelines for designing a protected implementation.

Assets to be protected are the output prime number as well as the secret elements used throughout its generation, more precisely the random values r and the sequence of intermediate primes reached by each iteration. It is therefore necessary to ensure that the implementation does not leak these values either during their generation or while they are being manipulated by the generation algorithm.

A first information leakage can occur during the generation of the first ℓ_0-bit prime. Since this is done using the Miller-Rabin criterion, the Miller-Rabin test itself has to be protected against side-channel attacks. A typical protection mechanism consists in performing an atomic modular exponentiation in the sense of [7] but since the base we use here is small, there is a risk that the exponent $n-1$ leaks at each multiplication as explained in [9]. The exponentiation may therefore be computed using a Square and Multiply-Always exponentiation which is a regular algorithm. A second operation to protect is the computation of I. This step involves the manipulation of p which must be kept secret. We therefore suggest to implement a secure division algorithm as described in [17].

Finke et al.presented in [11] an attack that specifically targets the computation of the next prime candidate (coprime to Π) at Step 2. of Algorithm 2.2. The attack is particularly applicable when a trial update operation is done with increments of 2 or Π. This attack does not seem applicable on Step 9 (performed with trial updates) of Algorithm 3.1 since the value used for next value of n is $n + 2p$ and p is unknown to the attacker. We recommend to implement the constructive method which is not sensitive to this attack and resists physical observation if the computation of p is done with the same exponentiation as the one used when applying the Miller-Rabin criterion.

We also note that the exponentiation $a^{n-1} \bmod n$ in Step 11 must be performed securely and that the atomic exponentiation is neither resistant nor efficient when $a = 2$. This part can be computed in a regular way using a Square and Multiply-Always exponentiation. In this case using $a = 2$ still results in negligible computational time for the multiplication and the computation remains protected against the SPA attack published in [9]. However the first squaring and multiplication operations (when the accumulator is still a power of 2 smaller than the modulus n) could leak information. It would then reveal the first bits of the exponent (about 10). It is then recommended to blind the modulus with a random value: in that case the computation would be $(2^{n-1} \bmod r_1 \cdot n) \bmod n$.

The final computations to protect from power analysis lie in Step 12. The exponentiation $2^{2r} \bmod n$ must be protected against the disclosure or r by using, as previously, the Square and Multiply-Always exponentiation technique. Also, the GCD operation $\gcd(2^{2r} - 1, n)$ could reveal the value of p if not implemented in a secure way. Our implementation of the GCD calculation has been carried out in constant time using dummy operations.

Applying these methods we obtain a side-channel protected efficient generator for provable primes. Finally, we note that fault-based attacks are not considered as a serious threat for prime number generators at the present time. This is mainly due to the inherently randomized nature of the generation algorithms.

7 Conclusion

The paper introduced two new methods to efficiently generate provable primes in embedded environments. We put forward novel algorithmic solutions and report practical results from our smartcard implementations. We have demonstrated that efficient generators exist for provable primes in constrained environments and compared the new methods with state-of-the-art generators for probable primes. We addressed side-channel analysis to ensure secure implementations of our generation methods. Overall, the paper opens the way to embedded generation of provable primes in nearly similar or better performances than current generators.

Acknowledgments. The authors would like to thank Vincent Verneuil for his valuable comments on this manuscript.

References

1. Agrawal, M., Kayal, N., Saxena, N.: PRIMES is in P. Annals of Mathematics 2, 781–793 (2002)
2. Alford, W.R., Granville, A., Pomerance, C.: On the Difficulty of Finding Reliable Witnesses. In: Huang, M.-D.A., Adleman, L.M. (eds.) ANTS 1994. LNCS, vol. 877, pp. 1–16. Springer, Heidelberg (1994)
3. Ankeny, N.C.: The least quadratic non residue. Annals of Mathematics 55, 65–72 (1952)
4. Atkin, A.O.L., Morain, F.: Elliptic Curves And Primality Proving. Mathematics of Computation 61, 29–68 (1993)
5. Bach, E.: Explicit bounds for primality testing and related problems. Mathematics of Computation 55, 355–380 (1990)
6. Brillhart, J., Lehmer, D.H., Selfridge, J.L., Tuckerman, B., Wagstaff Jr., S.S.: Factorization of $b^n \pm 1, b = 2, 3, 5, 7, 10, 11, 12$ Up to High Powers, vol. 22. American Mathematical Society (1988)
7. Chevallier-Mames, B., Ciet, M., Joye, M.: Low-Cost Solutions for Preventing Simple Side-Channel Analysis: Side-Channel Atomicity. IEEE Transactions on Computers 53(6), 760–768 (2004)
8. Clavier, C., Coron, J.-S.: On the Implementation of a Fast Prime Generation Algorithm. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 443–449. Springer, Heidelberg (2007)
9. Courrège, J.-C., Feix, B., Roussellet, M.: Simple Power Analysis on Exponentiation Revisited. In: Gollmann, D., Lanet, J.-L., Iguchi-Cartigny, J. (eds.) CARDIS 2010. LNCS, vol. 6035, pp. 65–79. Springer, Heidelberg (2010)
10. Diffie, W., Hellman, M.E.: New Directions in Cryptography. IEEE Transactions on Information Theory 22(6), 644–654 (1976)
11. Finke, T., Gebhardt, M., Schindler, W.: A New Side-Channel Attack on RSA Prime Generation. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 141–155. Springer, Heidelberg (2009)
12. FIPS PUB 186-3. Digital Signature Standard. National Institute of Standards and Technology (October 2009)
13. Garner, H.L.: The residue number system. In: Proceedings of the Western Joint Computer Conference, pp. 146–153 (1959)
14. Jaechke, G.: On strong pseudoprimes to several bases. Mathematics of Computation 61, 915–926 (1993)
15. Joye, M., Paillier, P.: Fast Generation of Prime Numbers on Portable Devices: An Update. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 160–173. Springer, Heidelberg (2006)
16. Joye, M., Paillier, P., Vaudenay, S.: Efficient Generation of Prime Numbers. In: Koç, Ç.K., Paar, C. (eds.) CHES 2000. LNCS, vol. 1965, pp. 340–354. Springer, Heidelberg (2000)
17. Joye, M., Villegas, K.: A protected division algorithm. In: Proceedings of the Fifth Smart Card Research and Advanced Application Conference, CARDIS 2002 (2002)
18. Maurer, U.M.: Fast Generation of Secure RSA-Moduli with Almost Maximal Diversity. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 636–647. Springer, Heidelberg (1990)
19. Maurer, U.M.: Fast generation of prime numbers and secure public-key cryptographic parameters. J. Cryptology 8(3), 123–155 (1995)

20. Pomerance, C., Selfridge, C., Wagstaff, J.L.: The pseudoprimes to 25.10e9. Mathematics of Computation 35, 1003–1026 (1990)
21. Rivest, R.L., Shamir, A., Adleman, L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM 21, 120–126 (1978)
22. Shawe-Taylor, J.: Generating strong primes. Electronic Letters 22(16), 875–877 (1986)

A Detailed Efficient Algorithms for Our Method

Alg. A.1. Generation of the initial prime based on Miller-Rabin testing

Input: bitsize $\ell_0 < 32$ of the initial (provable) prime, $\Pi = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot p_t$
Output: $\mathsf{GenInitPrime}(\ell_0)$: a ℓ_0-bit provable prime

1. generate a random ℓ_0-bit integer n with $\gcd(n, \Pi) = 1$ and go to 3
2. update n such that $\gcd(n, \Pi) = 1$,
3. if $\mathsf{F}_2(n) =$ **false** then go to 2
4. if $\mathsf{MR}_2(n) =$ **false** then go to 2
5. if $\mathsf{MR}_7(n) =$ **false** then go to 2
6. if $\mathsf{MR}_{61}(n) =$ **false** then go to 2
7. **return** n

Password-Based Authenticated Key Exchange

David Pointcheval

ENS, Paris, France*

Abstract. *Authenticated Key Exchange* protocols enable several parties to establish a shared cryptographically strong key over an insecure network using various authentication means, such as strong cryptographic keys or short (*i.e.*, low-entropy) common secrets. The latter example is definitely the most interesting in practice, since no additional device is required, but just a human-memorable password, for authenticating the players.

After the seminal work by Bellovin and Merritt, many settings and security notions have been defined, and many protocols have been proposed, in the two-user setting and in the group setting.

1 Introduction

Key exchange protocols are cryptographic primitives used to provide several users (two or more), communicating over a public unreliable channel, with a secure session key. This thus allows establishment of virtual secure channels over insecure networks, which is one of the main practical applications of cryptography. Bellare and Rogaway gave the first foundations in [13,14], but password-based authentication required more work: in this setting, where the authentication means is a short secret chosen from a small set of possible values (a four-digit pin, for example), the brute-force method which consists in trying all the possible values in the dictionary succeeds after a rather small number of attempts. This attack is called *on-line dictionary attack* and is unavoidable. But its damages can be limited by a policy that invalidates or blocks the use of a password if a certain number of failed attempts has occurred, unless failures are undetectable [27].

This paper presents a brief survey on Password-based Authenticated Key Exchange (PAKE) protocols, with a presentation of some security models in Section 2, and relations to practice. Section 3 deals with some practical constructions.

2 Security Models

Bellare, Pointcheval and Rogaway [12], and Boyko, MacKenzie and Patel [16] first formalized security of Password-based Authenticated Key Exchange, in two different frameworks.

* CNRS – UMR 8548 and INRIA – EPI Cascade.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 390–397, 2012.

2.1 Game-Based Security

The former model [12], the so-called *Find-then-Guess* scenario, is in the indistinguishability-based framework where an adversary should not be able to get an advantage significantly greater than q_S/N (or at most $\mathcal{O}(q_S)/N$ for some technicality reasons) in distinguishing a random session key from a real session key, if q_S is the number of active attacks and N the size of the dictionary. It has thereafter been improved to the *Real-or-Random* scenario [7]. More precisely, the adversary is given access to oracles: Execute-queries model passive attacks, Send-queries model active attacks, Corrupt-queries model corruptions with the leakage of long-term secrets, Reveal-queries model bad uses of session keys and thus the leakage of ephemeral secrets, and Test-queries model the semantic security of the session key with a real or random answer. In the Find-then-Guess scenario, only one Test-query can be asked, whereas in the Real-or-Random scenario many Test-queries can be asked with either always-real or always-random answers. The latter is clearly at least as strong as the former. But while both scenarios were known to be equivalent for encryption schemes [11], a linear loss in the number of Test-queries makes them quite different for PAKE, where the advantage should remain in $\mathcal{O}(q_s)/N$, whatever the number of Test-queries. We have then showed [7] that in this Real-or-Random scenario, Reveal-queries are not useful anymore, hence simplifying the security games.

2.2 Simulation-Based Security

The latter model [16] is in the simulation-based framework, with an ideal functionality in which the adversary is allowed to check one password per session. This models on-line dictionary attacks. Excepted this test instance password, no information is leaked about the passwords and the session keys.

2.3 Universal Composability

In both above models, one formalized the fact that, with an active attack, the adversary can basically test one password, whereas passive eavesdropping does not (computationally) leak any information. The goal is essentially to rule out *off-line dictionary attacks* in which the adversary makes some active and passive attacks, and then makes an off-line brute-force attack on the dictionary. On-line brute-force attacks, which are unavoidable, should be the only possible way to have some information about the session keys, and thus many interactions with a real player are required.

However, there were still some limitations on the password distributions and for composition with other protocols, which were overcome by Canetti, Halevi, Katz, Lindell and MacKenzie [24]. They indeed provided an ideal functionality in the Universally Composable (UC) security framework [23], see Figure 1. This functionality also models on-line dictionary attacks with a TestPwd-query that can be asked once to each user in sessions. An important property is that passwords are chosen by the environment which then hands them to the parties

The functionality $\mathcal{F}_{\mathsf{PAKE}}$ is parameterized by a security parameter k. It interacts with an adversary $\mathcal{S}$ and a set of parties $P_1,\ldots,P_n$ via the following queries:

- P_i asks for a **(NewSession, sid, P_i, P_j, pw)**: Send (NewSession, sid, P_i, P_j) to $\mathcal{S}$. If this is the first NewSession-query, or if this is the second NewSession-query and there is a record (P_j, P_i, pw'), then record (P_i, P_j, pw) and mark this record fresh.
- $\mathcal{S}$ asks for a **(TestPwd, sid, P_i, pw')**: If there is a record of the form (P_i, P_j, pw) which is fresh, then do:
  - If $pw = pw'$, mark the record compromised and reply with "correct guess";
  - If $pw \neq pw'$, mark the record interrupted and reply with "wrong guess".
- $\mathcal{S}$ asks for a **(NewKey, sid, P_i, sk)**: If there is a record of the form (P_i, P_j, pw), and this is the first NewKey-query for P_i, then:
  - If this record is compromised, or either P_i or P_j is corrupted, then output (sid, sk) to player P_i;
  - If this record is fresh, and there is a record (P_j, P_i, pw') with $pw' = pw$, and a key sk' was sent to P_j, and (P_j, P_i, pw) was fresh at the time, then output (sid, sk') to P_i;
  - In any other case, pick a new random key sk' of length k and send (sid, sk') to P_i.

  Either way, mark the record (P_i, P_j, pw) as completed.

Fig. 1. The PAKE Ideal Functionality $\mathcal{F}_{\mathsf{PAKE}}$

as inputs. This guarantees security even in the case where two honest players execute the protocol with two different passwords: the environment can emulate any distribution, mistypes of passwords and related passwords. Also note that allowing the environment to choose the passwords guarantees forward secrecy. This functionality mimics quite well some concrete requirements, but still, some leakage of information is not modeled, and could be exploited by a real-life adversary, whereas the ideal functionality does not allow it to the ideal-world adversary.

Explicit Authentication. With the above functionality, if neither party is corrupted, then they both end up with a uniformly-distributed session key, either the same key if the passwords are the same (success), or independent keys if the passwords are different (failure). Furthermore, the adversary learns nothing about the keys and the passwords, and even nothing about the status of the session (success or failure), but the users either. *Explicit authentication*, or mutual authentication modeled in [5], provides the players with a session key if and only if the passwords are the same, informing the adversary of success or not. This is an interesting additional feature, which is also more relevant in practice. In the real life, the adversary anyway learns whether the protocol succeeded or not, since in the latter case the communication stops.

Combined with the split functionality [10], it also allows to remove the TestPwd-query since the NewKey-query would reveal to the adversary whether the passwords are the same or not, by leaking the success or failure status. The split functionality allows the adversary to split a session between users Alice and

Bob into two sessions, one between Alice and the adversary trying to impersonate Bob, and a second one between Bob and the adversary trying to impersonate Alice. When the adversary plays with Alice, in case of success, this means it has guessed Alice's password, which is similar to the TestPwd-query.

Contributiveness. In the $\mathcal{F}_{\mathsf{PAKE}}$ functionality, if one party is corrupted, or if the adversary successfully guessed the player's password, the adversary is granted the right to fully determine the session key. Note that as soon as a party is corrupted, the adversary anyway learns the key, so one can think that nothing is lost by allowing it to fully determine it. But this is precisely the difference between *key agreement* and *key distribution* protocols.

In case of groups, this makes a huge difference. Hence the more recent functionality proposed by Abdalla, Catalano, Chevalier and Pointcheval [4] which provides the *contributiveness* property to Group Password-based Authenticated Key Exchange (GPAKE), see Figure 2. PAKE is a particular case of GPAKE with

The functionality $\mathcal{F}_{\mathsf{GPAKE}}$ is parameterized by a security parameter k, and the parameter t of the contributiveness. It interacts with an adversary $\mathcal{S}$ and a set of parties $P_1, \ldots, P_n$ via the following queries:

- P_i asks for a **(NewSession, sid, Pid, P_i, pw_i)**: If this is the first NewSession-query for P_i, where Pid is a set of at least two distinct identities containing P_i, record $(\mathsf{sid}, \mathsf{Pid}, P_i, pw_i)$, mark it fresh, and send $(\mathsf{sid}, \mathsf{Pid}, P_i)$ to $\mathcal{S}$. Ignore any subsequent NewSession-queries with a different Pid set. If all the players involved in Pid have submitted their NewSession-queries, then record $(\mathsf{sid}, \mathsf{Pid}, \mathsf{ready})$ and send it to $\mathcal{S}$.
- $\mathcal{S}$ asks for a **(TestPwd, sid, Pid, P_i, pw')**: If there exists a record of the form $(\mathsf{sid}, \mathsf{Pid}, P_i, pw_i)$ which is fresh:
  - If $pw_i = pw'$, mark the record compromised and reply with "correct guess";
  - If $pw_i \neq pw'$, mark the record interrupted and reply with "wrong guess".
- $\mathcal{S}$ asks for a **(NewKey, sid, Pid, sk)**: If there is a record of the form $(\mathsf{sid}, \mathsf{Pid}, \mathsf{ready})$, then, denote by n_c the number of corrupted players, and
  - If all $P_i \in \mathsf{Pid}$ have the same passwords and $n_c < t$, choose $sk' \in \{0,1\}^k$ uniformly at random and store $(\mathsf{sid}, \mathsf{Pid}, sk')$.
  - If all $P_i \in \mathsf{Pid}$ have the same passwords but $n_c \geq t$, store $(\mathsf{sid}, \mathsf{Pid}, sk)$.

  In both cases, for all $P_i \in \mathsf{Pid}$, mark the record $(\mathsf{sid}, \mathsf{Pid}, P_i, pw_i)$ completed. In any other case, store $(\mathsf{sid}, \mathsf{Pid}, \mathsf{error})$, and for all $P_i \in \mathsf{Pid}$, mark the record $(\mathsf{sid}, \mathsf{Pid}, P_i, pw_i)$ error. When the key is set, report the result (either error or completed) to $\mathcal{S}$.
- $\mathcal{S}$ asks for a **(SendKey, b, sid, Pid, P_i)**: If $P_i \in \mathsf{Pid}$ and there is a recorded tuple $(\mathsf{sid}, \mathsf{Pid}, \alpha)$ where $\alpha \in \{0,1\}^k \cup \{\mathsf{error}\}$, send $(\mathsf{sid}, \mathsf{Pid}, \alpha)$ to P_i if $b = 1$ or $(\mathsf{sid}, \mathsf{Pid}, \mathsf{error})$ if $b = 0$.
- $\mathcal{S}$ asks for a **(Corrupt, sid, Pid, P_i)**: If there is a recorded tuple $(\mathsf{sid}, \mathsf{Pid}, P_i, pw_i)$, then reveal pw_i to $\mathcal{S}$. If there also is a recorded tuple $(\mathsf{sid}, \mathsf{Pid}, sk)$, that has not yet been sent to P_i, then send $(\mathsf{sid}, \mathsf{Pid}, sk)$ to $\mathcal{S}$.

Fig. 2. The Contributory GPAKE Ideal Functionality $\mathcal{F}_{\mathsf{GPAKE}}$

groups of size 2. The latter property allows the adversary to fully determine the session key only if it has corrupted enough players, more than a threshold. This threshold can even be maximal: as soon as a player is honest, if a common key is generated, it is uniformly distributed in an unpredictable way. This means that no player has a more important role, and so there is no player to corrupt in priority for the adversary. As explained above, and as done in [5], one can even remove TestPwd-queries, allowing the adversary to split the group into several subgroups, with sub-session-IDs, where the adversary plays the role of the other users.

3 Constructions

3.1 Two-Party Password-Based Authenticated Key Exchange

Bellovin and Merritt [15] proposed the first scheme, the so-called Encrypted Key Exchange (EKE), see Figure 3 for a sketch of the protocol, where $\mathcal{E}$ is assumed to be an encryption scheme onto the group $\mathbb{G}$, sometimes modeled as an ideal cipher. A first security analysis has been provided in the indistinguishability-based framework, in the ideal-cipher model [12], followed by several proofs of variations [18, 19, 8], trying to reduce the need of ideal models but still keeping the initial efficiency of EKE. EKE has also been studied in the simulation-based framework, in the random-oracle model [16], followed by studies in the UC framework [3] with security against adaptive corruptions, but still in ideal models. Our "simple PAKE" protocols [8] are definitely the most efficient, with a random oracle only for extracting the session key, with a security analysis in the Find-then-Guess scenario, under the CDH assumption.

Katz, Ostrovsky and Yung [33] proposed the first practical scheme, but still less efficient than above schemes, in the standard model with a common reference string, followed by a generalization from Gennaro and Lindell (GL) [29, 28], using the power of smooth-projective hash functions [26], in the Find-then-Guess scenario. Many variations [24, 6, 34, 31, 35] have thereafter been proposed, to get security in the UC framework, to improve round efficiency, or to rely on new assumptions.

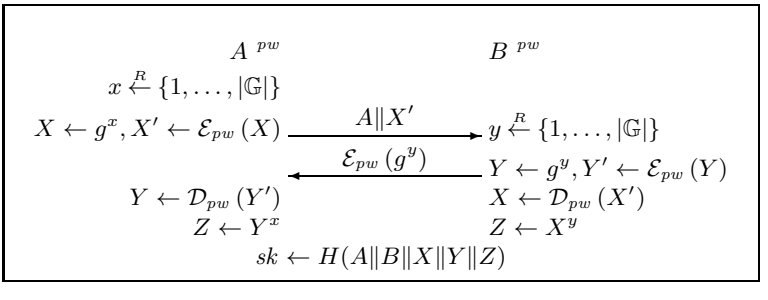


Fig. 3. Encrypted Key Exchange

Whereas the huge majority of the protocols rely on Diffie-Hellman assumptions, some efficient schemes have also been proposed on factoring-related assumptions [36, 37, 25, 30]. Besides the Secure Remote Password (SRP) protocol [39] and the Simple Password Exponential Key Exchange (SPEKE) protocol [32] that have been standardized, EKE-like and GL-like schemes are the two main streams, with security analyses in the UC framework.

3.2 Group Password-Based Authenticated Key Exchange

For groups, while the first proposals were extensions of the group Diffie-Hellman key exchange [38, 20, 17], the Burmester and Desmedt construction [21, 22] became more appropriate, because of its constant number of rounds, independently of the size of the group. Several group password-based authenticated key exchange protocols have then been proposed [2, 9, 1, 5], essentially combining a two-party PAKE with the Burmester and Desmedt methodology.

Acknowledgments. I would like to thank Céline Chevalier for her valuable comments, and all my other co-authors on this quite active and interesting area: Michel Abdalla, Mihir Bellare, Emmanuel Bresson, Dario Catalano, Olivier Chevassut, Pierre-Alain Fouque, Louis Granboulan, Thomas Pornin, Jean-Jacques Quisquater and Phil Rogaway.

References

1. Abdalla, M., Bohli, J.-M., González Vasco, M.I., Steinwandt, R.: (Password) Authenticated Key Establishment: From 2-Party to Group. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 499–514. Springer, Heidelberg (2007)
2. Abdalla, M., Bresson, E., Chevassut, O., Pointcheval, D.: Password-Based Group Key Exchange in a Constant Number of Rounds. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 427–442. Springer, Heidelberg (2006)
3. Abdalla, M., Catalano, D., Chevalier, C., Pointcheval, D.: Efficient Two-Party Password-Based Key Exchange Protocols in the UC Framework. In: Malkin, T. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 335–351. Springer, Heidelberg (2008)
4. Abdalla, M., Catalano, D., Chevalier, C., Pointcheval, D.: Password-Authenticated Group Key Agreement with Adaptive Security and Contributiveness. In: Preneel, B. (ed.) AFRICACRYPT 2009. LNCS, vol. 5580, pp. 254–271. Springer, Heidelberg (2009)
5. Abdalla, M., Chevalier, C., Granboulan, L., Pointcheval, D.: Contributory Password-Authenticated Group Key Exchange with Join Capability. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 142–160. Springer, Heidelberg (2011)
6. Abdalla, M., Chevalier, C., Pointcheval, D.: Smooth Projective Hashing for Conditionally Extractable Commitments. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 671–689. Springer, Heidelberg (2009)
7. Abdalla, M., Fouque, P.-A., Pointcheval, D.: Password-Based Authenticated Key Exchange in the Three-Party Setting. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 65–84. Springer, Heidelberg (2005)

8. Abdalla, M., Pointcheval, D.: Simple Password-Based Encrypted Key Exchange Protocols. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 191–208. Springer, Heidelberg (2005)
9. Abdalla, M., Pointcheval, D.: A Scalable Password-Based Group Key Exchange Protocol in the Standard Model. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 332–347. Springer, Heidelberg (2006)
10. Barak, B., Canetti, R., Lindell, Y., Pass, R., Rabin, T.: Secure Computation Without Authentication. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 361–377. Springer, Heidelberg (2005)
11. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption. In: 38th Annual Symposium on Foundations of Computer Science, pp. 394–403. IEEE Computer Society Press (October 1997)
12. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated Key Exchange Secure against Dictionary Attacks. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 139–155. Springer, Heidelberg (2000)
13. Bellare, M., Rogaway, P.: Entity Authentication and Key Distribution. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 232–249. Springer, Heidelberg (1994)
14. Bellare, M., Rogaway, P.: Provably secure session key distribution: The three party case. In: 27th Annual ACM Symposium on Theory of Computing, pp. 57–66. ACM Press (May/June 1995)
15. Bellovin, S.M., Merritt, M.: Encrypted key exchange: Password-based protocols secure against dictionary attacks. In: 1992 IEEE Symposium on Security and Privacy, pp. 72–84. IEEE Computer Society Press (May 1992)
16. Boyko, V., MacKenzie, P.D., Patel, S.: Provably Secure Password-Authenticated Key Exchange Using Diffie-Hellman. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 156–171. Springer, Heidelberg (2000)
17. Bresson, E., Chevassut, O., Pointcheval, D.: Group Diffie-Hellman Key Exchange Secure against Dictionary Attacks. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 497–514. Springer, Heidelberg (2002)
18. Bresson, E., Chevassut, O., Pointcheval, D.: Security proofs for an efficient password-based key exchange. In: Jajodia, S., Atluri, V., Jaeger, T. (eds.) ACM CCS 2003: 10th Conference on Computer and Communications Security, pp. 241–250. ACM Press (October 2003)
19. Bresson, E., Chevassut, O., Pointcheval, D.: New Security Results on Encrypted Key Exchange. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 145–158. Springer, Heidelberg (2004)
20. Bresson, E., Chevassut, O., Pointcheval, D., Quisquater, J.-J.: Provably authenticated group Diffie-Hellman key exchange. In: ACM CCS 2001: 8th Conference on Computer and Communications Security, pp. 255–264. ACM Press (November 2001)
21. Burmester, M., Desmedt, Y.: A Secure and Efficient Conference Key Distribution System (Extended Abstract). In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 275–286. Springer, Heidelberg (1995)
22. Burmester, M., Desmedt, Y.: A secure and scalable group key exchange system. Information Processing Letters 94(3), 137–143 (2005)
23. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: 42nd Annual Symposium on Foundations of Computer Science, pp. 136–145. IEEE Computer Society Press (October 2001)

24. Canetti, R., Halevi, S., Katz, J., Lindell, Y., MacKenzie, P.: Universally Composable Password-Based Key Exchange. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 404–421. Springer, Heidelberg (2005)
25. Catalano, D., Pointcheval, D., Pornin, T.: **IPAKE**: Isomorphisms for Password-Based Authenticated Key Exchange. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 477–493. Springer, Heidelberg (2004)
26. Cramer, R., Shoup, V.: Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)
27. Ding, Y., Horster, P.: Undetectable on-line password guessing attacks. SIGOPS Oper. Syst. Rev. 29, 77–86 (1995)
28. Gennaro, R.: Faster and Shorter Password-Authenticated Key Exchange. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 589–606. Springer, Heidelberg (2008)
29. Gennaro, R., Lindell, Y.: A Framework for Password-based Authenticated Key Exchange. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 524–543. Springer, Heidelberg (2003), http://eprint.iacr.org/2003/032.ps.gz
30. Gentry, C., Mackenzie, P.D., Ramzan, Z.: Password authenticated key exchange using hidden smooth subgroups. In: Atluri, V., Meadows, C., Juels, A. (eds.) ACM CCS 2005: 12th Conference on Computer and Communications Security, pp. 299–309. ACM Press (November 2005)
31. Groce, A., Katz, J.: A new framework for efficient password-based authenticated key exchange. In: Al-Shaer, E., Keromytis, A.D., Shmatikov, V. (eds.) ACM CCS 2010: 17th Conference on Computer and Communications Security, pp. 516–525. ACM Press (October 2010)
32. Jablon, D.P.: Strong password-only authenticated key exchange. SIGCOMM Comput. Commun. Rev. 26(5), 5–26 (1996)
33. Katz, J., Ostrovsky, R., Yung, M.: Efficient Password-Authenticated Key Exchange Using Human-Memorable Passwords. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 475–494. Springer, Heidelberg (2001)
34. Katz, J., Vaikuntanathan, V.: Smooth Projective Hashing and Password-Based Authenticated Key Exchange from Lattices. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 636–652. Springer, Heidelberg (2009)
35. Katz, J., Vaikuntanathan, V.: Round-Optimal Password-Based Authenticated Key Exchange. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 293–310. Springer, Heidelberg (2011)
36. Lucks, S.: Open Key Exchange: How to Defeat Dictionary Attacks Without Encrypting Public Keys. In: Christianson, B., Lomas, M. (eds.) Security Protocols 1997. LNCS, vol. 1361, pp. 79–90. Springer, Heidelberg (1998)
37. MacKenzie, P., Patel, S., Swaminathan, R.: Password-Authenticated Key Exchange Based on RSA. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 599–613. Springer, Heidelberg (2000)
38. Steiner, M., Tsudik, G., Waidner, M.: Diffie-Hellman key distribution extended to group communication. In: ACM CCS 1996: 3rd Conference on Computer and Communications Security, pp. 31–37. ACM Press (March 1996)
39. Wu, T.D.: The secure remote password protocol. In: ISOC Network and Distributed System Security Symposium – NDSS 1998. The Internet Society (March 1998)

Constant-Round Multi-party Private Set Union Using Reversed Laurent Series

Jae Hong Seo[1], Jung Hee Cheon[2,⋆], and Jonathan Katz[3,⋆⋆]

[1] National Institute of Information and Communications Technology
Tokyo, Japan
jaehong@nict.go.jp
[2] ISaC & Dept. of Mathematical Sciences
Seoul National University
Seoul, Korea
jhcheon@snu.ac.kr
[3] Dept. of Computer Science
University of Maryland
Maryland, USA
jkatz@cs.umd.edu

Abstract. We introduce the idea of associating a set of elements with a *rational function* represented using a *reversed Laurent series*. Using this representation, we propose private set-union protocols in the multi-party setting, assuming an honest majority. Our protocols are the first efficient protocol for private set union with constant round complexity (in both the semi-honest and malicious settings), as well as the first with statistical security (in the semi-honest setting).

1 Introduction

We focus here on constructing protocols for *privacy-preserving set operations*. In this setting, we have a set of parties $\mathcal{P}_1, \ldots, \mathcal{P}_n$ with each party $\mathcal{P}_i$ holding a set $S_i \subseteq \mathcal{U}$ of elements in some known universe $\mathcal{U}$; the parties want to compute some function of their sets such as their intersection $\bigcap_i S_i$ or union $\bigcup_i S_i$. Of course, the problem can be solved using protocols for generic secure multi-party computation [13,3], but we are interested in more efficient solutions. This problem, for various types of set operations, has received a lot of attention in both the two-party [10,4,14,7,6,17,9,15,8] and multi-party [19,11,23] settings.

In this paper, we propose a new framework for privacy-preserving set operations based on representing sets using *rational polynomial functions* and manipulating this representation using *reversed Laurent series*. (See the following section for an overview.) Although our framework can be extended to apply to a more general class of set operations, we focus here on computing *set union*

⋆ Supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2012-0001243)
⋆⋆ Supported by NSF grant #1111599.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 398–412, 2012.

Table 1. Privacy-preserving set-union protocols. DCR denotes the decisional composite-residuosity assumption [21], and DL denotes the discrete-logarithm assumption. The number of parties is n, the maximum set size is k, the number of corrupted parties is t, and τ_N, ρ_N (resp., τ_p, ρ_p) are the size and multiplication cost for a modulus N (resp. prime p) used for Paillier encryption (resp., representing domain elements). τ'_p and ρ'_p are the size and multiplication cost for a cyclic group of order p used for Pedersen commitment scheme and Gennaro-Rabin-Rabin verifiable secret sharing scheme.

Semi-honest:

Ref.	Rounds	Communication	Computation	Assumptions	Threshold
[19]	$O(n)$	$O(n^2\, k\, \tau_N)$	$O(n^3 k^2 \tau_N \rho_N)$	DCR	$t < n$
[11]	$O(n)$	$O(n^2\, k\, \tau_N)$	$O(nk^2 \tau_N \rho_N)$	DCR	$t < n$
Here	$O(1)$	$O(n^3 k^2 \tau_p)$	$O((n^4 k^2 + n^2 k^2 \tau_p)\rho_p)$	none	$t < n/2$

Malicious:

Ref.	Rounds	Communication	Computation	Assumptions	Threshold
[11]	$O(n)$	$O((n^3\, k\, + n^2\, k^2)\, \tau_N)$	$O(nk^2 \tau_N \rho_N)$	DCR	$t < n$
Here	$O(1)$	$O(n^3 k^2 \tau'_p)$	$O(n^4 k^2 \tau'_p \rho'_p)$	DL	$t < n/2$

in the multi-party setting. Set union is not trivial to compute securely, and in particular the solution in which each party publicly reveals its set is not secure since it reveals which parties hold which elements, as well as the *multiplicity* of each element in the union.

Our framework yields efficient multi-party protocols for private set union that are secure against any dishonest minority, and in particular we obtain the first efficient multi-party protocols for set union (in both the semi-honest and malicious settings) that use a *constant* number of rounds. Moreover, our protocol achieves *statistical* security in the semi-honest (aka, honest-but-curious) setting. In contrast, previous protocols [19,11] have round complexity linear in the number of parties, and achieve computational security even in the semi-honest setting. On the other hand, previous protocols tolerate any number of corrupted users. We compare our work to prior work in Table 1.

Beyond the result just stated, we believe our techniques are of independent interest as they provide what is, to the best of our knowledge, a novel approach to privacy-preserving computation on sets. We explain our approach in more detail in the following section.

1.1 Overview of Our Techniques

As in some prior work (e.g., [10,19,11,6,7]), we begin with the observation that a set S can be represented by a polynomial $f_S(x)$ over a field $\mathbb{F} \supseteq S$ such that the roots of $f_S(x)$ are exactly the elements of S; namely,

$$f_S(x) = \prod_{s \in S} (x - s).$$

In contrast to previous work, however, we then switch to viewing S as being represented by the *rational* polynomial $1/f_S(x)$. This representation is well suited for computing set union, since

$$\frac{1}{f_S(x)} + \frac{1}{f_{S'}(x)} = \frac{f_S(x) + f_{S'}(x)}{f_S(x) \cdot f_{S'}(x)} = \frac{\gcd\left(f_S(x), f_{S'}(x)\right) \cdot p(x)}{f_S(x) \cdot f_{S'}(x)}$$
$$= \frac{p(x)}{\mathrm{lcm}\left(f_S(x), f_{S'}(x)\right)}, \tag{1}$$

for some polynomial $p(x)$. That is, the denominator of (the reduced representation of) the rational polynomial $\frac{1}{f_S(x)} + \frac{1}{f_{S'}(x)}$ is a polynomial $f_{S \cup S'}(x) \stackrel{\text{def}}{=} \mathrm{lcm}(f_S(x), f_{S'}(x))$ with no repeated roots, whose roots are exactly the elements of $S \cup S'$. Because of how it is defined (in particular, the fact that it has no repeated roots), the polynomial $f_{S \cup S'}(x)$ reveals nothing beyond $S \cup S'$ and therefore provides a starting point for secure computation of the union.

The above does not yet give a secure protocol for computing the union, as we must still address several challenges. First, we need an efficient way to manipulate rational polynomials. For this, we rely on the *reversed Laurent series* representation of rational functions [25, Section 16.8]; see Section 2 for details. Second, we need to deal with the fact that the numerator in (1) might reveal information beyond the union $S \cup S'$. We thus modify the above, having the parties choose random polynomials $r(x), r'(x)$ of degree at most $|S|-1$ and $|S'|-1$, respectively, and then compute

$$\frac{r(x)}{f_S(x)} + \frac{r'(x)}{f_{S'}(x)} = \frac{f_S(x)r'(x) + f_{S'}(x)r(x)}{f_S(x) \cdot f_{S'}(x)} = \frac{\gcd\left(f_S(x), f_{S'}(x)\right) \cdot u(x)}{f_S(x) \cdot f_{S'}(x)}$$
$$= \frac{u(x)}{\mathrm{lcm}\left(f_S(x), f_{S'}(x)\right)}.$$

We prove that $u(x)$, above, is a *uniformly distributed* polynomial of degree at most $\deg(\mathrm{lcm}(f_S(x), f_{S'}(x))) - 1$. Thus, assuming $|\mathbb{F}| \gg |S|$, it holds with overwhelming probability that $u(x)$ and $f_{S \cup S'}(x)$ have no roots in common and so recovering the denominator of the above still yields the correct result. Moreover, uniformity of $u(x)$ implies that computing the above leaks no information about either party's original set.

Although we describe the two-party case above for simplicity, we can easily extend the above argument to the case $n > 2$ in which we are mostly interested. See Section 3.1 for details.

1.2 Related Work

Private set-union protocols should hide both (1) which parties hold which elements, and (2) the multiplicity of each element in the union. There are only a few multi-party protocols satisfying these two requirements. Kissner and Song [19] proposed a protocol which can be utilized for multi-party set union in the semi-honest setting. Frikken [11] proposed a privacy-preserving set-union protocol in the malicious setting. Both protocols rely on a "mix-net" approach, where $t+1$ parties mix encrypted elements (when security against t corruptions is required). This approach inherently requires round complexity $O(t)$.

Some protocols achieving relaxed privacy guarantees have been proposed. In particular, Kissner and Song [19], Sang and Shen [23], and Hong et al. [16] proposed multi-party set-union protocols that leak the multiplicity of each element in the union.

In the two-party case, other protocols are known. Brickell and Shmatikov [4] proposed two-party set-union protocols secure against *honest-but-curious* adversaries. Recently, Hazay and Nissim [15] proposed very efficient protocols for privacy-preserving set union secure against malicious adversaries; their protocol achieves (almost) linear complexity in the number of private inputs. Neither of these protocols appear to generalize easily to the multi-party case.

1.3 Outline of the Paper

In the next section, we recall the notion of the *reversed Laurent series* (RLS) representation of a rational function, and discuss efficient conversions between a rational function and its RLS representation. In Section 3, we show how to use the RLS representation of rational functions to perform set union. As applications of our technique, we give constant-round protocols for computing set union in both the semi-honest and malicious settings.

2 Reversed Laurent Series

We let $\mathbb{Z}_p$ denote the set of integers modulo p. In this paper, we always take p prime so that $\mathbb{Z}_p$ is also the finite field of size p. As usual, $\mathbb{Z}_p[x]$ denotes the set of polynomials over $\mathbb{Z}_p$. We use $[a, b]$ (with $a \leq b$ and both possibly negative) to denote the set of integers between a and b, inclusive.

2.1 Reversed Laurent Series and Rational Functions

A *reversed Laurent series* (RLS) over $\mathbb{Z}_p$ is a singly infinite, formal sum of the form

$$f(x) = \sum_{i=-\infty}^{m} a_i \, x^i \quad (a_m \neq 0),$$

for m an integer and $a_i \in \mathbb{Z}_p$. We refer to m as the *degree* of f, denoted $\deg(f)$. Given $d_1 \leq d_2 \leq m$, we define

$$f(x)_{[d_1,d_2]} = \sum_{i=d_1}^{d_2} a_i \, x^i.$$

The set of all reversed Laurent series, denoted $\mathbb{Z}_p((x^{-1}))$, forms a field with addition and multiplication defined in the natural way. Since $\mathbb{Z}_p[x]$ is a subring of $\mathbb{Z}_p((x^{-1}))$, any rational function f/g with $f, g \in \mathbb{Z}_p[x]$ and $g \neq 0$ can be expressed as a reversed Laurent series and we refer to this as the *RLS representation* of the rational function f/g. Note that the RLS representation for a given rational function is unique. That is, if $f/g = f'/g'$ then the RLS representations of f/g and f'/g' are identical.

2.2 Conversion from a Rational Function to Its RLS

Let $f, g \in \mathbb{Z}_p[x]$, and assume $\deg(f) < \deg(g) \leq \ell$. (The case $\deg(f) \geq \deg(g)$ can be reduced to this case by first performing polynomial division with remainder.) One can compute $k > \deg(g)$ high-order terms of the RLS representation of f/g using the following algorithm:

RationalToRLS(f, g, k):

(1) Compute $F(x) = f(x) \cdot x^k$.
(2) Use polynomial division to compute $Q(x)$ and $R(x)$ with $F(x) = g(x) \cdot Q(x) + R(x)$ and $\deg(R) < \deg(g)$.
(3) Output $Q(x) \cdot x^{-k}$.

Since $F/g = Q + R/g$ and $\deg(R) < \deg(g)$, we have $Q = (F/g)_{[0,k+\deg(f)-\deg(g)]}$. Since $F/g = x^k \cdot f/g$ and we assumed $\deg(f) < \deg(g)$, we see that the output consists of exactly the k high-order terms of the RLS of f/g; that is, $Q(x) \cdot x^{-k} = (f/g)_{[-k,\deg(f)-\deg(g)]} = (f/g)_{[-k,-1]}$.

The computational cost of the above algorithm is essentially just the complexity of polynomial division.

2.3 Conversion from an RLS Representation to a Rational Function

The RLS representation of a rational function f/g will, in general, have infinitely many terms. However, all "information" about f/g is contained in a finite number of high-order terms. Specifically, let $f, g \in \mathbb{Z}_p[x]$ with $\deg(f) < \deg(g) \leq \ell$ and $g \neq 0$. Then the rational function f/g is determined by the 2ℓ high-order terms of its RLS representation. Moreover, there is an efficient algorithm to recover f/g (in reduced terms) given these 2ℓ high-order terms and the bound ℓ on the degree of g. See [25, Section 17.5.1] for details.

3 Privacy-Preserving Set Union

We begin with an overview of our approach to computing set union, followed by formal descriptions of protocols in the semi-honest and malicious settings. We consider n parties, each of whom holds a set over some universe $\mathcal{U} \subset \mathbb{Z}_p$ where $p > n$ is known and $\mathcal{U}$ is a negligible fraction of $\mathbb{Z}_p$. (This can be easily obtained by padding every element in the original universe with sufficiently many 0s.) We further assume the size k_i of each party's set is known. (In fact, for simplicity here we assume that $k_i = k$ for all i. A treatment of the general case will be found in the full version.) By having parties pad out their sets to some maximum size using random elements, this can be relaxed to requiring only that $\sum_i k_i$ is known; we omit the details.

3.1 Representing Sets and Computing Their Union

Given a set $S \subseteq \mathcal{U}$ of size $|S| = d$, we define the polynomial

$$f_S(x) \stackrel{\text{def}}{=} \prod_{s \in S} (x - s).$$

Note that $\deg(f_S) = |S|$. We are actually going to work with the RLS representation of $1/f_S(x)$. The set S can be recovered from the $2|S|$ high-order terms of the RLS representation of $1/f_S(x)$: given the high-order terms, we first reconstruct $f_S(x)$ using the conversion algorithm; the entire set S can then be obtained by factoring $f_S(x)$ (which can be done in polynomial time over the finite field $\mathbb{Z}_p$).

Given sets $S_1, \ldots, S_n$ held by n parties $\mathcal{P}_1, \ldots, \mathcal{P}_n$, note that $f_{\cup_i S_i}(x) = \text{lcm}(f_{S_1}(x), \ldots, f_{S_n}(x))$. Rather than have the parties compute the least common multiple directly (which would be difficult to do securely), we have them compute it using the following high-level approach:

1. The parties collectively define random polynomials $r_1(x), \ldots, r_n(x)$ of degree (at most) $d - 1$ in such a way that no coalition of up to t parties knows anything about any of the $r_i(x)$. This is done via standard techniques using Shamir secret sharing (in the semi-honest setting) or a form of verifiable secret sharing (in the malicious setting).
2. The parties securely compute (sufficiently many terms of the RLS of) the sum $\sum_i \frac{r_i(x)}{f_{S_i}(x)}$. Note that

$$\sum_{i=1}^{n} \frac{r_i(x)}{f_{S_i}(x)} = \frac{u(x)}{\text{lcm}(f_{S_1}(x), \ldots, f_{S_n}(x))}$$

   for some polynomial $u(x)$ of degree at most $\deg(\text{lcm}(f_{S_1}(x), \ldots, f_{S_n}(x))) - 1$.
3. Each party locally computes $u'(x)$ and $L(x)$ such that $u'(x)/L(x) = \sum_i \frac{r_i(x)}{f_{S_i}(x)}$ and $\gcd(u'(x), L(x)) = 1$. Each party then factors $L(x)$ over $\mathbb{Z}_p$ and outputs the roots as $\bigcup_i S_i$.

We need to prove both correctness and privacy of the above. To do so we will rely on the following result:

Lemma 1. *Let $f_1(x), \ldots, f_n(x)$ be polynomials of degree $d_1, \ldots, d_n \geq 1$. Say $r_1(x), \ldots, r_n(x)$ are chosen uniformly and independently from the set of polynomials of degree at most $d_i - 1$, respectively, and let $u(x)$ be such that*

$$\frac{u(x)}{\text{lcm}(f_1(x), \ldots, f_n(x))} = \sum_{i=1}^{n} \frac{r_i(x)}{f_i(x)}.$$

Then $u(x)$ is uniformly distributed among polynomials having degree at most $\deg(\text{lcm}(f_1(x), \cdots, f_n(x))) - 1$.

Proof. We prove the lemma for $n = 2$; the general case follows by induction. Let $f_1(x)$ and $f_2(x)$ be polynomials of degree d_1, d_2, respectively. Say $r_1(x)$ and $r_2(x)$ are chosen uniformly and independently from the set of polynomials of degree at most $d_1 - 1$ and $d_2 - 1$, respectively, and let $u(x)$ be such that $\frac{r_1(x)}{f_1(x)} + \frac{r_2(x)}{f_2(x)} = \frac{u(x)}{\mathrm{lcm}(f_1(x), f_2(x))}$. We show that $u(x)$ is uniformly distributed among polynomials of degree at most $\deg(\mathrm{lcm}(f_1(x), f_2(x))) - 1$.

Define $f_1'(x) = f_1(x)/\gcd(f_1(x), f_2(x))$, with $f_2'(x)$ defined analogously. We have

$$\begin{aligned}\frac{r_1(x)}{f_1(x)} + \frac{r_2(x)}{f_2(x)} &= \frac{r_1(x)f_2(x) + r_2(x)f_1(x)}{f_1(x)f_2(x)} \\ &= \frac{\gcd(f_1(x), f_2(x)) \cdot u(x)}{f_1(x)f_2(x)} = \frac{u(x)}{\mathrm{lcm}(f_1(x), f_2(x))}\end{aligned}$$

where $u(x) = r_1(x)f_2'(x) + r_2(x)f_1'(x)$ has degree at most

$$d' \stackrel{\text{def}}{=} \deg(\mathrm{lcm}(f_1(x), f_2(x))) - 1.$$

Identifying a polynomial of degree at most d with a vector over $\mathbb{Z}_p$ of length $d + 1$, consider the map $M : \mathbb{Z}_p^{d_1} \times \mathbb{Z}_p^{d_2} \to \mathbb{Z}_p^{d'+1}$ defined via $M(r_1(x), r_2(x)) = r_1(x)f_2'(x) + r_2(x)f_1'(x)$. Say $M(r_1(x), r_2(x)) = M(r_1'(x), r_2'(x))$. This implies

$$(r_1(x) - r_1'(x)) \cdot f_2'(x) = (r_2'(x) - r_2(x)) \cdot f_1'(x).$$

Since $\gcd(f'(x), g'(x)) = 1$, the above holds iff there exists some $h(x) \in \mathbb{Z}_p[x]$ such that

$$\begin{aligned}r_1(x) - r_1'(x) &= h(x) \cdot f_1'(x) \\ r_2'(x) - r_2(x) &= h(x) \cdot f_2'(x).\end{aligned}$$

Note that $\deg(h) \leq \gcd(f_1(x), f_2(x)) - 1$ because of the bound on the degrees of $r_1(x)$, $r_1'(x)$, $r_2(x)$, and $r_2'(x)$. The above means that each point $M(r_1(x), r_2(x))$ in the image of M has exactly $K \stackrel{\text{def}}{=} p^{\gcd(f_1(x), f_2(x))}$ pre-images. Furthermore, since

$$\begin{aligned}|\mathbb{Z}_p^{d_1} \times \mathbb{Z}_p^{d_2}|/K &= p^{d_1+d_2}/p^{\gcd(f_1(x), f_2(x))} \\ &= p^{\mathrm{lcm}(f_1(x), f_2(x))} = p^{d'+1} = |\mathbb{Z}_p^{d'+1}|,\end{aligned}$$

we see that M is also surjective. Since M is regular and surjective, choosing $r_1(x), r_2(x)$ uniformly and independently at random yields a uniform element $u(x) = M(r_1(x), r_2(x))$ in its range. ∎

Correctness and privacy now follow easily from the lemma. Since $u(x)$ is random, and the universe $\mathcal{U}$ is a negligible fraction of $\mathbb{Z}_p$, the probability that $u(x)$ and $\mathrm{lcm}(f_{S_1}(x), \ldots, f_{S_n}(x))$ have a factor in common is negligible. Thus, $u'(x) = u(x)$ and $L(x) = \mathrm{lcm}(f_{S_1}(x), \ldots, f_{S_n}(x))$ with overwhelming probability and so correctness holds. Moreover, the view of any coalition of up to t parties can be

simulated given the result $\bigcup_i S_i$, implying privacy. This simulation is done as follows. Let $D = |\bigcup_i S_i|$. Compute $f_{\cup_i S_i}(x)$, choose a random polynomial $u(x)$ of degree at most $D-1$, and then compute (sufficiently many high-order terms of) the RLS representation of $u(x)/f_{\cup_i S_i}(x)$.

In the next sections, we fill in the missing details in the above description and give a protocol for computing set union in the semi-honest setting. We then show how to extend the protocol to the malicious setting as well.

3.2 (Verifiable) Secret Sharing of Polynomials

In our protocols, we use Shamir's secret-sharing scheme [24] in the semi-honest model, and the verifiable secret-sharing (VSS) protocol of Gennaro et al. [12], denoted GRR-VSS scheme, in the malicious model. (We assume the availability of private channels between all pairs of parties.) In either case, addition of shares can be performed locally (without interaction), and multiplication of shares can be done using a suitable multiplication sub-protocol (i.e., Simple-Mult in the semi-honest model, and Mult in the malicious model [12]).

A polynomial can be (verifiably) shared by (verifiably) sharing each of its coefficients. Addition and multiplication of polynomial shares follows from addition and multiplication of the underlying shares of the coefficients. In particular, addition of polynomial shares can be done locally. Multiplication of two shared polynomials of degrees d_1, d_2 requires $O(d_1 \cdot d_2)$ invocations of an underlying Mult protocol (plus local additions); nevertheless, because these can be parallelized, the entire process takes only a constant number of rounds.

3.3 A Protocol Secure against Honest-But-Curious Adversaries

We propose a privacy-preserving set-union protocol, denoted PPSU-HBC, for the honest-but-curious (HBC) adversary model. Every party contributes to obtaining $\cup_{i\in[1,n]} S_i$, where S_i is the private set of the i-th party; however, a semi-honest adversary corrupting less than $n/2$ parties should not obtain additional information about the set of any other party (except for its size). For simplicity here, we assume that for each set S_i has the same cardinality, denoted by k.

In Figure 1, we present the protocol. The basic idea follows the overview from Section 3.1. Each party $\mathcal{P}_i$ contributes random polynomials $r_{ij}(x)$ for $j \in \{1,\ldots,n\}$. Define $r_j(x) = \sum_{i=1}^{n} r_{ij}(x)$. The parties then (privately) compute the high-order $2nk$ terms of the RLS representation of

$$U(x) = \sum_{j\in[1,n]} \frac{r_j(x)}{f_j(x)},$$

where $f_j(x)$ is the polynomial associated with the set of party $\mathcal{P}_j$. To compute the $2nk$ higher-order terms of $U(x)$, we utilize the fact that the $2nk$ higher-order terms of

$$\sum_{j\in[1,n]} r_j(x) \cdot \left(\frac{1}{f_j}\right)_{[-(2n+1)k-1,-k]}$$

Private Input for each party $\mathcal{P}_i$ ($i \in [1, n]$): A set $S_i \subset \mathcal{U}$ of size k.
Goal: Each party obtains $\bigcup_{i\in[1,n]} S_i$.

Each party $\mathcal{P}_i$:

1. Constructs $f_i(x) = \prod_{\alpha\in S_i}(x - \alpha)$, runs RationalToRLS$(1, f_i(x), (2n + 1)k - 1) \to (\frac{1}{f_i(x)})_{[-(2n+1)k+1,-k]}$, defines $\tilde{f}_i(x) := (\frac{1}{f_i(x)})_{[-(2n+1)k+1,-k]} \cdot x^{(2n+1)k-1}$, and chooses random polynomials $r_{ij}(x)$ of degree at most $k - 1$ for $j \in [1, n]$.
2. Secret shares $\tilde{f}_i(x)$ and $r_{ij}(x)$ for $\forall j \in [1, n]$ in parallel.
3. Locally sums his shares of $r_{ij}(x)$ to obtain shares of $r_j(x) = \sum_{i\in[1,n]} r_{ij}(x)$ for $\forall j \in [1, n]$.
4. Runs (in parallel) a shared polynomial multiplication protocol to compute shares of $\tilde{f}_j(x) \cdot r_j(x)$ for $\forall j \in [1, n]$.
5. Locally sums his shares of $\tilde{f}_j(x)r_j(x)$ to obtain shares of the $2nk$ high-order terms of $U'(x) = \sum_{j\in[1,n]} \tilde{f}_j(x)R_j(x)$ (i.e., $U'(x)_{[k-1,(2n+1)k-2]}$).
6. All parties reconstruct the $2nk$ high-order terms of the RLS representation of $U(x)$, and then use these to recover two polynomials $u(x)$ and $L(x)$ such that $(\frac{u(x)}{L(x)})_{[-2nk,-1]} = U(x)_{[k-1,(2n+1)k-2]} \cdot x^{-(2n+1)k+1}$ and $\gcd(u(x), L(x)) = 1$. Then, each party extracts all roots of $L(x)$.

Fig. 1. The PPSU-HBC protocol

is equal to that of $U(x)$, where the degree of r_j is $k - 1$. Then, each party can recover (the rational function) $U(x)$ from its RLS representation using the conversion algorithm; each party can then compute the union by factoring the denominator of $U(x)$.

Privacy follows from Lemma 1, along with the fact that the $r_j(x)$ are random polynomials for any coalition of fewer than $n/2$ corrupted parties. (This security threshold comes from the threshold needed by the Simple-Mult protocol.)

Theorem 1. *The PPSU-HBC protocol presented in Figure 1 is statistically t-secure against a semi-honest adversary, for any $t < n/2$.*

Proof. Let C be a coalition of t corrupted parties controlled by the adversary $\mathcal{A}$, and let H be the set of honest parties. Given all private inputs of corrupted parties and the result $S = \cup_{i\in[1,n]} S_i$, we construct a simulator Sim as follows: It first divides $S \setminus (\cup_{i\in C} S_i)$ into sets $\hat{S}_i$ (for $i \in H$) such that the number of elements in each set is exactly k. (An element may appears in multiple sets, if necessary.) Now, $\cup_{i\in H} \hat{S}_i = S \setminus (\cup_{i\in C} S_i)$ and $|\hat{S}_i| \le k$. Then, Sim runs the PPSU-HBC protocol by treating each $\hat{S}_i$ as private input of an honest party.

We argue that the view of $\mathcal{A}$ in the simulation is identically distributed to the view of $\mathcal{A}$ in the real world. It is easy to see that this holds for steps (1)–(5) of the protocol. In step (6), the only information revealed consists of the polynomials $u(x)$ and $L(x)$. But (with all but negligible probability) $L(x)$ exactly encodes the union (i.e., $L(x) = \prod_{\alpha\in S}(x - \alpha)$), and $u(x)$ is a random polynomial of appropriate degree (using here the fact that the r_i are uniform conditioned

on the adversary's view, since they are generated by summing over random contributions from all parties). ■

Complexity Analysis: Secret sharing requires $O(n^2)$ multiplications in $\mathbb{Z}_p$, and an execution of also uses $O(n^2)$ multiplications.

The computation overheads of $\mathcal{P}_i$ in each step of PPSU-HBC protocol is as follows:

- To compute $f_i(x)$ and $\tilde{f}_i(x)$, $O(nk^2)$ multiplications are required.
- To secret-share the $((2n+1)k-1)$-degree polynomial $\tilde{f}_i(x)$ and $(k-1)$-degree polynomial $r_{ij}(x)$ for $j \in [1,n]$, $O(n^3k)$ multiplications are required.
- To compute $U'(x) = \sum_{j\in[1,n]} \tilde{f}_j(x)(\sum_{i\in[1,n]} r_{ij}(x))$ from $\tilde{f}_j(x)$ and $r_{ij}(x)$, we need $O(n^2k^2)$ multiplications and $O(n^2k^2)$ additions. Therefore, all parties should run Simple-Mult and the local addition $O(n^2k^2)$ times; hence, each party requires $O(n^4k^2)$ multiplications in all.
- To recover $U'(x)_{[k-1,(2n+1)k-2]}$, $O(n^3k)$ multiplications are required.
- To recover a rational function $\frac{G(x)}{F(x)}$ from $U'(x)_{[k-1,(2n+1)k-2]} \cdot x^{-(2n+1)k+1}$, $O(n^2k^2)$ multiplications are required. To factor a polynomial $F(x)$ with a degree of at most nk, $O((nk)^{1.5+o(1)} + (nk)^{1+o(1)} \log p)$ multiplications are required [20,26].

Therefore, the total computation cost is $O(n^4k^2 + n^2k^2 \log p)$ multiplications in $\mathbb{Z}_p$.

The communication overheads of secret-sharing and Simple-Mult are $O(n)$ integers modulus p for each party; hence, the PPSU-HBC protocol's communication cost is $O(n^3k^2)$ elements in $\mathbb{Z}_p$. Further, the round complexity is constant since, in each step, all transmissions can be performed in parallel.

3.4 A Protocol Secure against Malicious Adversaries

We can extend the protocol presented in Section 3.3 to obtain security in the presence of malicious adversaries by using verifiable secret sharing and adding zero-knowledge proofs. Intuitively, in the PPSU-HBC protocol, if we utilize GRR-VSS and Mult instead of secret sharing and Simple-Mult, respectively, no coalition of fewer-than-half corrupted parties can behave maliciously without detection. In addition, however, we require each party to prove that they honestly follow Step (1). Namely, they must prove that $\tilde{f}_j(x)$ is well-formed; that is, that it is the RLS representation of $1/f(x)$ for some f of degree k. We let $\mathsf{ZKPK}[\mathsf{Com}(f(x)), \mathsf{Com}(g(x))]$ denote a zero-knowledge proof that (committed) polynomials f, g of known degree satisfy $g(x) = (\frac{1}{f(x)})_{[-(2n+1)k+1,-k]} \cdot x^{(2n+1)k-1}$. We give the details of such a proof now.

Pedersen Commitment Scheme. To commit an element in $a \in \mathbb{Z}_p$, we use the Pedersen commitment scheme [22]. Here, a commitment of a is $\mathsf{Com}(a; r) = g^a h^r$ for random $r \in \mathbb{Z}_p$, where g, h are group elements of a cyclic group $\mathbb{G}$ of order p.

(When there will be no confusion, we write $\mathsf{Com}(\cdot)$ instead of $\mathsf{Com}(\cdot;\cdot)$.) The Pedersen commitment scheme is additively homomorphic. That is,

$$\mathsf{Com}(a;r)\cdot\mathsf{Com}(b;s) = g^a h^r g^b h^s = g^{a+b}h^{r+s} = \mathsf{Com}(a+b;r+s).$$

In addition, the Pedersen commitment scheme is perfectly hiding and computationally binding under the *discrete logarithm* assumption in $\mathbb{G}$.

Define a commitment of a polynomial $f(x) = \sum_{i\in[0,k]} a_i x^i$ to be a tuple of commitments to its coefficients. Given $\mathsf{Com}(f(x))$ and $\mathsf{Com}(g(x))$ where f, g are monic polynomials of degree k and $\deg(g(x)) = 2nk-1$, respectively, we provide a zero-knowledge proof that $g(x) = (\frac{1}{f(x)})_{[-(2n+1)k+1-,-k]} \cdot x^{(2n+1)k-1}$. (Note that the degrees of f, g can be verified if they are known to be monic by simply decommitting to their high-order coefficient.) The main observation is that the desired relation holds iff $\deg(f(x)g(x) - x^{\deg(f(x))+\deg(g(x))}) < \deg(f(x))$, using the following lemma.

Lemma 2. *If* $f(x), g(x)$ *satisfy* $\deg(f(x)g(x) - x^{\deg(f(x))+\deg(g(x))}) < \deg(f(x))$, *then* $g(x) = (\frac{1}{f(x)})_{[-\deg(f(x))-\deg(g(x)),-\deg(f(x))]} \cdot x^{\deg(f(x))+\deg(g(x))}$.

Proof. Let $d_f = \deg(f)$ and $d_g = \deg(g)$. The assumption of the lemma is that $\deg(f(x)g(x) - x^{d_f+d_g}) < d_f$. Then,

$$f(x)g(x) = x^{d_f+d_g} + \sum_{i\in[0,d_f-1]} a_i x^i,$$

and hence,

$$g(x)\cdot x^{-d_f-d_g} = \frac{1}{f(x)} + \frac{1}{f(x)}\cdot\left(\sum_{i\in[-d_f-d_g,-d_g-1]} a_i x^i\right)$$

for some $a_i \in \mathbb{Z}_p$.

Since $g(x)\cdot x^{-d_f-d_g}$ and $\frac{1}{f(x)}\cdot(\sum_{i\in[-d_f-d_g,-d_g-1]} a_i x^i)$ have no common monomials, we obtain

$$g(x)\cdot x^{-d_f-d_g} = (\frac{1}{f(x)})_{[-d_f-d_g,-d_f]}.$$

This concludes the proof. ∎

Given the above, a zero-knowledge protocol for $\mathsf{ZKPK}[\mathsf{Com}(f(x)), \mathsf{Com}(g(x))]$ can be constructed using standard techniques, following [5]. We omit the details.

By applying all above changes to PPSU-HBC, we obtain a protocol PPSU-MAL for the malicious adversary model; see Figure 2. Note that GRR-VSS and our zero-knowledge proofs use the Pedersen commitment scheme, which is binding under the *discrete logarithm assumption* so that the security of PPSU-MAL requires such a computational assumption. The following theorem proves the security of PPSU-MAL.

Common Input: A domain of private data $P \subset \mathbb{Z}_p$. Description of GRR-VSS and Mult: common reference strings for public parameters in GRR-VSS.
Private Input for each party $\mathcal{P}_i$ ($i \in [1,n]$): A set S_i of k private elements in P.
Goal: Each party obtains $\cup_{i\in[1,n]} S_i$ without learning other information.

Each party $\mathcal{P}_i$

1. Constructs $f_i(x) = \prod_{\alpha\in S_i}(x-\alpha)$, runs RationalToRLS$(1, f_i(x), (2n+1)k-1) \to (\frac{1}{f_i(x)})_{[-(2n+1)k+1,-k]}$, defines $\tilde{f}_i(x) := (\frac{1}{f_i(x)})_{[-(2n+1)k+1,-k]} \cdot x^{(2n+1)k-1}$, and chooses random polynomials $r_{ij}(x)$ of degree at most $k-1$ for $j \in [1,n]$.
2. Commits to $f_i(x)$ and $\tilde{f}_i(x)$, runs ZKPK[Com(f_i), Com$(\tilde{f}_i)$] protocol.
3. Verifiably secret shares $\tilde{f}_i(x)$ and $r_{ij}(x)$ for $\forall j \in [1,n]$ in parallel.
4. Locally sums his shares of $r_{ij}(x)$ to obtain shares of $r_j(x) = \sum_{i\in[1,n]} r_{ij}(x)$ for each $j \in [1,n]$.
5. Runs (in parallel) a shared polynomial multiplication protocol to compute shares of $\tilde{f}_j(x) \cdot r_j(x)$ for $\forall j \in [1,n]$.
6. Locally sums his shares of $\tilde{f}_j(x) r_j(x)$ to obtain shares of $2nk$ high-order terms of $U'(x) = \sum_{j\in[1,n]} \tilde{f}_j(x) r_j(x)$ (i.e., $U'(x)_{[k-1,(2n+1)k-2]}$).
7. All parties reconstruct $U'(x)_{[k-1,(2n+1)k-2]}$, and then use it to recover two polynomials $u(x)$ and $L(x)$ such that $(\frac{u(x)}{L(x)})_{[-2nk,-1]} = U'(x)_{[k-1,(2n+1)k-2]} \cdot x^{-(2n+1)k+1}$ and $\gcd(u(x), L(x)) = 1$. Then, each party extracts all roots of $L(x)$.

Fig. 2. PPSU-MAL protocol in the malicious adversary model

Theorem 2. *Assuming that the number of corrupted parties is $t < n/2$, where n is the number of all parties of the PPSU-MAL protocol in Figure 2, and that the discrete logarithm assumption holds in the underlying cyclic group, then PPSU-MAL protocol is computationally t-secure in the malicious setting.*

Proof. We prove this theorem by showing that for any arbitrarily malicious adversary $\mathcal{A}$ controlling all corrupted parties ($t < n/2$), there exists an efficient simulator $\mathcal{S}$ such that for any inputs to all parties, the view of the corrupted parties and the outputs of the honest parties in the PPSU-MAL protocol are computationally indistinguishable from the outputs of $\mathcal{S}$ and the honest parties in the ideal world interacting with a trusted third party $\mathcal{F}$ computing set union.

Now, we describe $\mathcal{S}$. Let C be a coalition of corrupted parties controlled by $\mathcal{A}$, and H be a set of honest parties.

1. $\mathcal{S}$ generates public parameters for GRR-VSS and Mult and publishes them with securely keeping the discrete logarithms between parameters as a trapdoor. Then, Sim interacts with C on behalf of H. First, it chooses random polynomials $f_i(x)$ of degree at most k, the corresponding $\tilde{f}_i(x)$, and random polynomials $r_{ij}(x)$ of degree at most $k-1$ for $i \in H$ and $j \in [1,n]$. Then, it runs Steps 2 of the PPSU-MAL protocol.
2. From the ZKPK[Com$(f_i(x))$, Com$(\tilde{f}_i(x))$] protocol for $i \in C$, Sim extracts witnesses $f_i(x)$ for $i \in C$ using the strong soundness property of the

zero-knowledge proof protocol. Then, it computes all roots of $f_i(x)$, which are the inputs of the corrupted parties, using a polynomial factoring algorithm.

3. Let C' be a set of corrupted parties who correctly pass the zero-knowledge proofs protocol in Step 2 and secret-share their inputs in Step 3. Sim participates with the inputs of C' in the ideal world. It receives the result of the ideal PPSU functionality, which is the union of the inputs of C' and H. Then, it computes $U(x) = \frac{u(x)}{L(x)}$, where $L(x)$ is the polynomial associated with the result set of the ideal PPSU functionality and $u(x)$ is a random polynomial of degree at most $\deg(L(x)) - 1$.
4. Sim rewinds $\mathcal{A}$ with the same auxiliary inputs and runs protocol through Step 6 with the same public parameters and polynomials $f_i(x)$ of H as the previous execution.
5. In Step 7, Sim contributes to recover $U(x)_{[-2nk,-1]} \cdot x^{(2n+1)k-1}$. Since Sim has a trapdoor of GRR-VSS, Sim can equivocate on the recovered secret-shared values. Further, Sim already knows shared secrets of corrupted parties so that Sim can contribute $U(x)_{[-2nk,-1]} \cdot x^{(2n+1)k-1}$ to be recovered in Step 7.
6. Sim outputs a transcript of all interactions with $\mathcal{A}$ in the last execution.

At the end of the simulation, $\mathcal{A}$ obtains the union of all inputs of C' and H. GRR-VSS, Mult and ZKPK are secure against any probabilistic polynomial-time adversary $\mathcal{A}$ under the discrete logarithm assumption. That is, any probabilistic polynomial-time adversary $\mathcal{A}$ cannot anomalously behave without detection during the protocols GRR-VSS, Mult and ZKPK when the discrete logarithm assumption holds in the underlying cyclic group. Furthermore, if $\mathcal{A}$ follows the predetermined description of PPSU-MAL protocol, then Sim's output $(\text{OUT}^{\mathcal{S}}_{\mathcal{F},\mathcal{S}(\mathsf{aux})}(1^\lambda, \boldsymbol{x}))$ and outputs of H $(\text{OUT}^{\mathsf{hon}}_{\mathcal{F},\mathcal{S}(\mathsf{aux})}(1^\lambda, \boldsymbol{x}))$ in the ideal world is identical to the view of $\mathcal{A}$ $(\text{VIEW}_{\Pi,\mathcal{A}(\mathsf{aux})}(1^\lambda, \boldsymbol{x}))$ and outputs of H $(\text{OUT}^{\mathsf{hon}}_{\Pi,\mathcal{A}(\mathsf{aux})}(1^\lambda, \boldsymbol{x}))$ in the real world, respectively, since GRR-VSS, Mult and ZKPK are perfectly simulatable when the honest parties are majority. Therefore, there exists only negligible chance in the security parameter that two distributions will be different so that

$$\left\{\text{REAL}_{\Pi,\mathcal{A}(\mathsf{aux})}(1^\lambda, \boldsymbol{x})\right\}_{\lambda\in\mathbb{N},\boldsymbol{x}\in\{0,1\}^*} \quad \text{and} \quad \left\{\text{IDEAL}_{\mathcal{F},\mathcal{S}(\mathsf{aux})}(1^\lambda, \boldsymbol{x})\right\}_{\lambda\in\mathbb{N},\boldsymbol{x}\in\{0,1\}^*}$$

are computationally indistinguishable. ■

Complexity Analysis: In GRR-VSS, the dealer requires $O(n)$ exponentiations and $O(n^2)$ multiplications in $\mathbb{G}$. The verifier requires $O(1)$ exponentiations. In the reconstruction phase of GRR-VSS, each party requires $O(n)$ exponentiations and $O(n^2)$ multiplications. In Mult, each party requires $O(n^2)$ exponentiations and $O(n^2)$ multiplications. In local addition, $O(n)$ multiplications are required for each party. The zero-knowledge proofs protocol do not significantly affect the computational and communication complexity. The overall computational overheads and communication overheads of PPSU-MAL are $O(n^4k^2)$ exponentiations in $\mathbb{G}$, and $O(n^3k^2)$ group elements $\mathbb{G}$, respectively. The round complexity of the malicious protocol is still $O(1)$.

4 Conclusion

We introduced the Reversed Laurent Series (RLS) representation of a rational function, and showed a surprising relationship between rational function arithmetics (in particular, addition and multiplication) and set union computations. On the basis of these approach, we developed constant-round private set union protocol in both the semi-honest setting and the malicious setting. Our protocol is the first constant-round multi-party private set union protocol without aids of third party.

To the best of our knowledge, this paper shows the first instantiation of using the Reversed Laurent Series for cryptographic purpose. We leave finding other cryptographic applications, either inside or outside secure computing of set operations, as an interesting open problem.

References

1. Ateniese, G., De Cristofaro, E., Tsudik, G.: (If) Size Matters: Size-Hiding Private Set Intersection. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 156–173. Springer, Heidelberg (2011)
2. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF Formulas on Ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)
3. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: 20th Annual ACM Symposium on Theory of Computing (STOC), pp. 1–10. ACM Press (1988)
4. Brickell, J., Shmatikov, V.: Privacy-Preserving Graph Algorithms in the Semi-honest Model. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 236–252. Springer, Heidelberg (2005)
5. Camenisch, J.: Proof systems for general statements about discrete logarithms. Technical Report 260, Dept. of Computer Science, ETH Zurich (March 1997)
6. Camenisch, J., Zaverucha, G.M.: Private Intersection of Certified Sets. In: Dingledine, R., Golle, P. (eds.) FC 2009. LNCS, vol. 5628, pp. 108–127. Springer, Heidelberg (2009)
7. Dachman-Soled, D., Malkin, T., Raykova, M., Yung, M.: Efficient Robust Private Set Intersection. In: Abdalla, M., Pointcheval, D., Fouque, P.-A., Vergnaud, D. (eds.) ACNS 2009. LNCS, vol. 5536, pp. 125–142. Springer, Heidelberg (2009)
8. De Cristofaro, E., Kim, J., Tsudik, G.: Linear-Complexity Private Set Intersection Protocols Secure in Malicious Model. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 213–231. Springer, Heidelberg (2010)
9. De Cristofaro, E., Tsudik, G.: Practical Private Set Intersection Protocols with Linear Complexity. In: Sion, R. (ed.) FC 2010. LNCS, vol. 6052, pp. 143–159. Springer, Heidelberg (2010)
10. Freedman, M.J., Nissim, K., Pinkas, B.: Efficient Private Matching and Set Intersection. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 1–19. Springer, Heidelberg (2004)
11. Frikken, K.B.: Privacy-Preserving Set Union. In: Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 237–252. Springer, Heidelberg (2007)

12. Gennaro, R., Rabin, M.O., Rabin, T.: Simplified VSS and fast-track multiparty computations with applications to threshold cryptography. In: 17th Annual ACM Symposium on Principles of Distributed Computing (PODC), pp. 101–111. ACM Press (1998)
13. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game, or a completeness theorem for protocols with honest majority. In: 19th Annual ACM Symposium on Theory of Computing (STOC), pp. 218–229. ACM Press (1987)
14. Hazay, C., Lindell, Y.: Efficient Protocols for Set Intersection and Pattern Matching with Security Against Malicious and Covert Adversaries. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 155–175. Springer, Heidelberg (2008)
15. Hazay, C., Nissim, K.: Efficient Set Operations in the Presence of Malicious Adversaries. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 312–331. Springer, Heidelberg (2010)
16. Hong, J., Kim, J., Kim, J., Park, K., Cheon, J.: Constant-Round Privacy Preserving Multiset Union, http://eprint.iacr.org/2011/138
17. Jarecki, S., Liu, X.: Efficient Oblivious Pseudorandom Function with Applications to Adaptive OT and Secure Computation of Set Intersection. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 577–594. Springer, Heidelberg (2009)
18. Kaltofen, E., Shoup, V.: Subquadratic-time factoring of polynomials over finite fields. Mathematics of Computation 67(223), 1179–1197 (1998)
19. Kissner, L., Song, D.: Privacy-Preserving Set Operations. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 241–257. Springer, Heidelberg (2005); See also Technical Report CMU-CS-05-133, Carnegie Mellon University
20. Kedlaya, K.S., Umans, C.: Fast modular composition in any characteristic. In: 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 146–155. IEEE computer Society (2008)
21. Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
22. Pedersen, T.P.: Non-interactive and Information-Theoretic Secure Verifiable Secret Sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992)
23. Sang, Y., Shen, H.: Efficient and secure protocols for privacy-preserving set operations. ACM Trans. Information and System Security 13(1) (2009)
24. Shamir, A.: How to share a secret. Communications of the ACM 22, 612–613 (1979)
25. Shoup, V.: A Computational Introduction to Number Theory and Algebra, 2nd edn. Cambridge University Press (2009)
26. Umans, C.: Fast polynomial factorization and modular composition in small characteristic. In: 40th Annual ACM Symposium on Theory of Computing (STOC), pp. 481–490. ACM (2008)

Policy-Enhanced Private Set Intersection: Sharing Information While Enforcing Privacy Policies*

Emil Stefanov, Elaine Shi, and Dawn Song

UC Berkeley
{emil,elaines,dawnsong}@cs.berkeley.edu

Abstract. Companies, organizations, and individuals often wish to share information to realize valuable social and economic goals. Unfortunately, privacy concerns often stand in the way of such information sharing and exchange.

This paper proposes a novel cryptographic paradigm called Policy-Enhanced Private Set Intersection (PPSI), allowing two parties to share information while enforcing the desired privacy policies. Our constructions require minimal additional overhead over traditional Private Set Intersection (PSI) protocols, and yet we can handle rich policy semantics previously not possible with traditional PSI and Authorized Private Set Intersection (APSI) protocols. Our scheme involves running a standard PSI protocol over carefully crafted encodings of elements formed as part of a challenge-response mechanism. The structure of these encodings resemble techniques used for aggregating BLS signatures in bilinear groups. We prove that our scheme is secure in the malicious model, under the CBDH assumption, the random oracle model, and the assumption that the underlying PSI protocol is secure against malicious adversaries.

1 Introduction

The need for two parties to exchange privacy-sensitive information arises in numerous application domains. Often, the two parties involved in the exchange are mutually distrustful and do not wish to reveal any additional information other than what is necessary. In particular, we consider the scenario where two parties each hold a set of elements and wish to find the intersection of their elements without revealing other elements that are not in the intersection. In such applications, it is important to ensure that *each data item being exchanged*

* This material is based upon work partially supported by the Air Force Office of Scientific Research under MURI Grant No. 22178970-4170 and No. FA9550-08-1-0352, by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-0946797, by Intel through the ISTC for Secure Computing, and by a grant from the Amazon Web Services in Education program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 413–430, 2012.

is properly authenticated or authorized by the owner(s) or creator(s) of that data item due to the following reasons:

- **Thwart dishonest behavior.** Unless some form of authentication is required, a malicious party can claim possession of fictitious data items, in an attempt to find out whether the other party possesses these data items. For example, if hospitals A and B are trying to find out their common patients, a malicious hospital A can fictitiously claim that Carol is their patient, in an attempt to find out whether Carol is a patient with hospital B.
- **Comply with privacy policies.** Sharing of privacy-sensitive information may be governed by certain privacy regulations, either made by the government or individual organizations. For example, two healthcare providers A and B may wish to exchange information about their common patients to improve service and facilitate diagnosis. However, due to privacy regulations such as the Health Insurance Portability and Accountability Act (HIPAA), they can only share a patient's record if both providers have obtained the patient's consent. The above is an example of a simple privacy policy. In other application scenarios, we may also desire the ability to support richer privacy policies. We demonstrate how to support rich privacy policies in Section 4.

In this paper, we propose Policy-Enhanced Private Set Intersection (PPSI). In PPSI, each party has a set of elements, where each element may be authorized (signed) by a different authority or authorities. PPSI allows two parties to find the intersection of their sets, while enforcing rich privacy policies. The policies specify what authorizations each party must possess for its elements. Our scheme thwarts dishonest behavior by preventing a malicious party from using unauthorized elements during the set intersection to violate the privacy of the other party.

1.1 Results and Contributions

New problem definitions One important contribution we make is the definition of a new problem, namely, Policy-Enhanced Private Set Intersection (PPSI). Existing Private Set Intersection (PSI) protocols and Authorized Private Set Intersection (APSI) protocols are not general enough and fail to adequately address the needs of above-mentioned application scenarios. To resolve this problem our PPSI scheme offers the following rich capabilities not previously possible with existing PSI and APSI protocols:

- **Multiple authorities.** PPSI supports privacy policies where each element may be authorized by a different authority or different authorities. This makes PPSI particularly useful when each data item may not be owned or created by the same entity.
- **Rich privacy policies.** Many applications desire the ability to support expressive privacy policies. Our PPSI constructions can support rich policy

Table 1. Efficiency of our construction in Section 3.3. n is the maximum number of elements per user and m is the maximum number of authorities per element. The complexities are calculated assuming that we use [15] as the underlying PSI scheme.

	Overall	**Additional overhead over PSI**
Computation	$O(nm + n \log \log n)$	at most nm pairings
Bandwidth	$O(n)$	2 group elements
# Rounds	$O(1)$	1 round

semantics during the information sharing process, including *conjunctive and disjunctive policies, asymmetric policies, policies with attributes, and bundles of elements.*

Novel, Provably Secure Constructions. We propose novel PPSI constructions that offer two main functionalities: 1) a signing functionality which allows an authority to authenticate or authorize an element for a party; 2) a set intersection protocol that allows two parties to find the intersection of their elements, while enforcing the desired privacy policy.

We prove the security of our scheme against *malicious* adversaries, assuming that the underlying PSI scheme is also secure in the malicious model. The proof also relies on the CBDH assumption and the random oracle model.

Efficiency. Our constructions are efficient in practice. Specifically, we require $O(n)$ communication bandwidth and $O(nm + n \log \log n)$ computation, where n is the maximum number of elements per party and m is the maximum number of authorities per element. Also, our protocol executes in $O(1)$ rounds.

Notably, our constructions require only minimal overhead over standard Private Set Intersection (PSI) protocols, but can support rich policies that are not possible with standard PSI. We need one additional round of communication over standard PSI, during which the parties exchange two group elements (elliptic points). In terms of computation, we incur an additional overhead of at most nm bilinear pairings over the traditional PSI protocols.

Table 1 summarizes the efficiency of our basic construction described in Section 3.3.

1.2 Technical Challenges and Highlight of Our Techniques

It turns out that the problem is non-trivial, even with relatively simple policies. A straightforward idea is to adopt an existing PSI protocol and require that each party demonstrate a zero-knowledge proof that each element encoded in a cryptographic commitment has the appropriate authorizations. However, *the complication is that when each element has a different authority or different authorities, one cannot reveal the identity of the authority when performing the zero-knowledge proofs, as the identity of the authority can leak information about the corresponding element.*

Special encodings. Our techniques may be of independent interest. Our scheme leverages a type of *special encoding* that allows us to circumvent the need for performing complicated and costly zero-knowledge proofs. Each party computes the special encodings over their elements and then runs a standard PSI protocol over these encodings.

A party's encoding for an element x is essentially a product of terms demonstrating its authorization on x and the anticipated terms demonstrating the other party's authorization on x. The terms are cleverly crafted so that a party can compute its own terms by combining its own authorizations and a challenge sent by the other party. It can also compute the anticipated terms of the other party without having the other party's authorizations.

If both parties satisfy their respective policies for an element, then both parties obtain the same encoding for that element, and this particular encoding appears in the set intersection. However, if a party does not possess the correct authorizations for an element, it is unable (computationally intractable) to compute the correct encoding for this element. As a result, this party is unable to learn whether the other party owns the element.

We point out that our encoding idea bears resemblance to techniques used for aggregating BLS signatures in bilinear groups [5].

1.3 Related Work

A Private Set Intersection (PSI) protocol [9, 10, 11, 12, 13, 14, 15, 16, 17, 18] allows two parties to find the intersection of their respective sets such that neither party can infer elements in the other party's set that are not in the intersection. However, PSI protocols allow each party to place any element in their own set. A dishonest party can therefore insert fabricated elements in its set that she suspects the other party might have. The intersection will reveal if the other party indeed has those elements in its set.

To address this issue, Authorized Private Set Intersection (APSI) and variants [6, 8, 10] ensure that each party can only use elements certified by a trusted authority in the intersection protocol. Existing APSI protocols assume that for each party, there exists a single authority responsible for certifying all of its elements. Therefore, these schemes do not support rich privacy policies coming from multiple authorities, such as the application scenarios mentioned earlier.

2 Problem Definitions

2.1 Notations and Terminology

Let $\mathcal{U}$ denote a (countable) universe of all possible elements. Let Λ denote the set of all authorities.

As mentioned earlier, two parties, P_A and P_B, wish to find the intersection of their sets in a way that complies with certain privacy policies, that is, only when both parties have the appropriate authorizations for an element should it appear in the intersection.

Table 2. Table of notations

n	Max number of elements in each party's set.
m	Max number of authorities per element.
$\mathcal{U}$	The set of all possible elements.
$x \in \mathcal{U}$	An element.
Λ	The set of all authorities
$\mathsf{auth}_i \in \Lambda$	An authority that signs elements.
P_i	A party participating in our protocol.
S_i	P_i's set.
I	The resulting intersection.
vk_i, sk_i	auth_i's public verification key and secret signing key.
σ or σ_i	A signature issued by an authority.
attr	An attribute attached to a signature.
$F(x)$	Authorities for element x (for symmetric policies).
$F(x, P_i)$	Authorities for element x and party P_i (for asymmetric policies).
g	A random generator for the bilinear group.
$R_i = g^{r_i}$	P_i's challenge for the other party.

Policy. A privacy policy defines which authority or authorities must sign an element for a given party. For ease of exposition, we will first focus on *symmetric policies*, where each element needs to be authorized by one or more authorities, and the set of authorities is determined by the element itself, but is not dependent on the parties. Two exampless of symmetric policies are: (1)Claimed friendship with Alice needs to be authorized by Alice, and (2) Claimed membership in a social group needs to be authorized by the administrator(s) of the group.

As each element's authorities are determined by the element itself, we can use a function F to describe symmetric policies. Formally, let $F : \mathcal{U} \to 2^{|\Lambda|}$ denote a publicly-known policy function that maps each element to the set of authorities that must sign it. For example, let $x \in \mathcal{U}$, if $F(x) = \{\mathsf{auth}_1, \mathsf{auth}_3\}$, this means that element x has to be signed by authorities auth_1 and auth_3. One simple policy function is the identity function, e.g., each patient's record must be authorized by the patient herself, or claimed friendship with a user must be authorized by that user herself.

We say that $x \in \mathcal{U}$ is an *authorized element* for party P, if party P has received all the necessary signatures for x, i.e., P has received a signature σ_i for every $\mathsf{auth}_i \in F(x)$.

2.2 Basic Problem Definitions

Apart from the necessary setup and key generation functionalities, a PPSI scheme should offer two main functionalities: 1) a signature scheme allowing an authority auth_i to authorize an element x for a party P; 2) a set intersection protocol that allows two parties to find the intersection of their authorized elements.

We now present formal definitions for a basic PPSI scheme supporting symmetric policies. A Policy-Enhanced Private Set Intersection (PPSI) scheme (supporting symmetric policies) should provide the following algorithms or protocols:

- $\mathsf{Setup}(\lambda)$: The Setup algorithm is run only once at system initialization to generate public parameters param. The input λ represents the security parameter.
- $\mathsf{KeyGen}(\mathsf{param})$: Each authority auth_i runs the KeyGen algorithm to generate a signing and verification key pair $(\mathsf{sk}_i, \mathsf{vk}_i)$. auth_i then announces the public verification key vk_i but keeps the private signing key sk_i to itself.
- $\mathsf{Authorize}(\mathsf{param}, \mathsf{sk}_i, x, P_j)$: The $\mathsf{Authorize}$ algorithm allows an authority auth_i to grant a party P_j a signature on a specific element x.
- $\mathsf{Intersect}(P_i, P_j, S_i, S_j)$: Let $S_i, S_j \subseteq \mathcal{U}$. $\mathsf{Intersect}$ is an interactive protocol run by any two parties P_i and P_j on input sets S_i and S_j respectively. When both parties are honest, and assuming that P_i and P_j both have the necessary signatures for elements in S_i and S_j respectively, then both parties would learn the intersection $S_i \cap S_j$ at the end of the protocol.

2.3 Security Definitions

We prove the security of our protocol against a *malicious adversary*, who may deviate arbitrarily from the specified protocol. We define security by comparing what a malicious adversary can do in the *real* protocol execution against what the adversary can do in an *ideal* world. In the ideal-world execution, both parties would submit their sets to an imaginary trusted third-party denoted as T. The trusted party T would make sure that both parties have the correct authorizations on the elements they submitted. If a party submits an element without the necessary authorizations, T simply ignores that element. T then computes the intersection of the elements satisfying the privacy policy and returns the intersection to both parties. In the real-world, we do not use T and the parties communicate directly to execute the real set intersection protocol. Roughly speaking, the security definition implies that any attack that a polynomial-time adversary can perform in the real world is also possible in the ideal world. Intuitively, this suggests that the real-world set intersection protocol is as secure as the protocol in the ideal world that relies on a trusted third-party.

We now formally define the ideal functionality. The security definition involves multiple parties a subset of which is controlled by the adversary.

Authorize. T receives an authorization request from party P_i, requesting auth_j to authorize element x. T forwards the request to auth_j, who can either accept or reject the request. If auth_j accepts the authorization request, T replies accept to P_i and remembers that T has authorized P_i on element x. Otherwise, T replies reject to P_i.

SetIntersect. T receives a request from party P_i to perform set intersection with party P_j. T forwards the request to P_j. P_i and P_j now run an ideal set intersection protocol as below (unless P_j replies abort).

- i) P_i sends a set S_i to T, and T sends $|S_i|$ to P_j; or ii) P_i sends abort.
- i) P_j sends a set S_j to T, and T sends $|S_j|$ to P_i; or ii) P_j sends abort.
- T now checks whether each element in S_i and S_j has appropriate authorizations. Let $\widehat{S}_i \subseteq S_i$ and $\widehat{S}_j \subseteq S_j$ denote maximal subsets of S_i and S_j that have appropriate authorizations. T computes the intersection $I \leftarrow \widehat{S}_i \cap \widehat{S}_j$.
- T sends I to P_i, and P_i responds ok or abort.
- T sends I to P_j, and P_j responds ok or abort.

Definition 1. *Let $E = (E_1, E_2, \ldots, E_m)$ denote a sequence of events, where each E_i is of the form $(Authorize, P_i, \mathsf{auth}_j)$ or $(SetIntersect, P_i, P_j)$. Let $\textsc{Ideal}_{\mathcal{S},E}$ denote the joint output distribution of all parties and the adversary $\mathcal{S}$ in the ideal world under event sequence E. Let $\textsc{Real}_{\mathcal{A},E}$ denote the joint output distribution of all parties and the adversary $\mathcal{A}$ in the real world under event sequence E.*

We say that a PPSI scheme is secure, if for any polynomial-time adversary $\mathcal{A}$ in the real world, there exists simulator $\mathcal{S}$ in the ideal world, such that for any sequence of events E,

$$\textsc{Ideal}_{\mathcal{S},E} \stackrel{c}{=} \textsc{Real}_{\mathcal{A},E}$$

where $\stackrel{c}{=}$ denotes computational indistinguishability.

Note that we cannot prevent an adversary from refusing to participate in the protocol or aborting in the middle of the protocol execution. As a result, our definition explicitly allows the ideal-world adversary to abort any time during the ideal-world protocol. Our definition also allows each party to use only a subset of their authorized elements as input to the protocol.

Our protocol is not size-hiding, i.e., each party can learn the size of the other party's set. Therefore, in the ideal functionality, the trusted third-party reveals to each party the size of the other party's set. In particular, when both parties honestly use their authorized elements as inputs, i.e., $\widehat{S}_A = S_A$ and $\widehat{S}_B = S_B$, each party learns the size of the other party's authorized set. However, notice that a party can potentially fuzz the size of its set by padding the input set with random dummy elements for which it does not possess appropriate authorizations. These dummy random elements will not appear in the final set intersection due to lack of authorizations; however, they can hide the number of authorized elements each party has.

3 Construction

3.1 Strawman Schemes

One strawman approach would be for the two parties to perform a regular Private Set Intersection (PSI) over the elements' signatures, thereby revealing the signed elements that they have in common. However, this requires that both parties have the exact same signature for the same element. This does not allow authorities to bind a signature to a specific party. The signature can thus be easily transferred to unauthorized parties.

It is conceivable that there are other solutions based on standard techniques for the problem than our construction. For example, we can imagine schemes based on secure multi-party computation, verifiable shuffles, and matching pairs of blinded elements. However, to the best of our knowledge, these approaches all tend to have much higher computational and bandwidth complexity than our construction which achieves $O(nm + n \log \log n)$ computational overhead and $O(n)$ bandwidth overhead – both almost linear in the number of elements n if we assume the number of authorities per element m to be a constant.

3.2 Preliminaries

Bilinear group. Our scheme utilizes a bilinear $\mathbb{G}$ group of primary order p. There exists a non-degenerate bilinear mapping $\mathsf{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ such that $\forall g_1, g_2 \in \mathbb{G}, \forall a, b \in \mathbb{Z}, \mathsf{e}(g_1^a, g_2^b) = \mathsf{e}(g_1, g_2)^{ab}$. Our scheme relies on the following computational assumption.

Computational Bilinear Diffie-Hellman (CBDH) Assumption. Let $g \in \mathbb{G}$ denote a random generator of the group. The CBDH assumption posits the computational hardness of the following problem. Given randomly chosen $g^a, g^b, g^c \in \mathbb{G}$, compute $\mathsf{e}(g, g)^{abc}$.

Private Set Intersection. A Private Set Intersection (PSI) protocol allows two parties to compute their set intersection without revealing other elements. Our protocol utilizes a standard PSI protocol (e.g., the scheme by Hazay and Nissim [15]) as a blackbox. We assume that the PSI protocol is secure in the malicious model, and refer the readers to [15] for a formal security definitions of PSI.

3.3 Main Construction

Our construction involves running a standard PSI protocol over special encodings formed as part of a challenge-response protocol. Below, we first describe our construction, including the key generation and authorization algorithm, as well as the intersection protocol. Then, in Section 3.4, we explain in detail how to construct the encodings used in the set intersection protocol, and the properties required for the encodings.

Setup. The Setup algorithm chooses a bilinear group $\mathbb{G}$ of prime order p with pairing function $\mathsf{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. It then chooses a random generator $g \in \mathbb{G}$. Next, it picks a hash function $H : \{0,1\}^* \to \mathbb{G}$ which will be modeled as a random oracle. Finally, the Setup algorithm publishes a description of the bilinear group, the generator g, as well as the hash function.

Key generation algorithm. To pick a signing and verification key pair, each authority auth_i randomly selects $\mathsf{sk}_i \in_R \mathbb{Z}_p$. The verification key is $\mathsf{vk}_i := g^{\mathsf{sk}_i}$. Each authority i publishes its public verification key vk_i, but withholds its secret signing key sk_i.

Inputs: P_A, P_B each has sets S_A and S_B, with the appropriate authorizations.
Outputs: P_A, P_B each obtains $I{:=}S_A \cap S_B$

Protocol:

1. P_A : Select random $r_A \in_R \mathbb{Z}_p$, let $R_A \leftarrow g^{r_A}$
   P_B : Select random $r_B \in_R \mathbb{Z}_p$, let $R_B \leftarrow g^{r_B}$
   $P_A \rightarrow P_B : R_A$
   $P_B \rightarrow P_A : R_B$
2. P_A : $C_A \leftarrow \{\mathsf{EncodeElem}(x, r_A, R_B, P_A, P_B) | x \in S_A\}$
   P_B : $C_B \leftarrow \{\mathsf{EncodeElem}(x, r_B, R_A, P_B, P_A) | x \in S_B\}$
3. $P_A \Leftrightarrow P_B$: Engage in a PSI protocol with input sets C_A and C_B respectively. As a result, both parties obtain the set $C'{:=}C_A \cap C_B$ of encodings.
4. P_A, P_B : Recover the intersection I from their encodings C'.

Fig. 1. Intersection protocol

Authorization algorithm. Let $H : \{0,1\}^* \rightarrow \mathbb{G}$ denote a hash function modeled as a random oracle. We assume that each party P_i has a publicly-known unique name (e.g., an assigned name or a randomly generated identifier). Without risk of ambiguity, we overload the notation P_i to denote either the party or its name.

For authority auth_i to authorize element x for party P_j, auth_i computes the following BLS signature [5] and issues it to P_j.

$$\sigma \leftarrow H(x, P_j)^{\mathsf{sk}_i}$$

In the above, the hash function is computed over the name of the element concatenated with the name of the receiving party. Notice that since the name of the party is incorporated into the signature, the signature cannot be transferred to another party.

Intersection protocol. Our protocol takes place in following four phases. The detailed construction is presented in Figure 1.

1. **Challenge Phase.** P_A sends P_B a random challenge R_A, and P_B sends P_A a random challenge R_B.
2. **Encoding Phase.** Each party computes an encoding for each element it possesses and with the appropriate authorizations. The encoding is dependent on the random challenges R_A and R_B. Figure 2 specifies the encoding function.
3. **Set Intersection Phase.** Both parties perform a standard Private Set Intersection (PSI) protocol using their respective encodings as the inputs. For the underlying PSI scheme we use the protocol described in [15].
4. **Recovery Phase.** At the end of the PSI protocol, each party learns the intersection of the encodings. Through the intersection of encodings, each party recovers the original elements in the intersection.

function EncodeElem$(x, r_{\text{self}}, R_{\text{other}}, P_{\text{self}}, P_{\text{other}})$
  $c \leftarrow 1 \in \mathbb{G}_T$
  for $\mathsf{auth}_i \in F(x)$ **do**
    Let σ_i denote P_{self}'s signature on x from auth_i.
    $c \leftarrow c \cdot \mathsf{e}(\sigma_i, R_{\text{other}}) \cdot \mathsf{e}(H(x, P_{\text{other}}), \mathsf{vk}_i)^{r_{\text{self}}}$
  end for
  return c
end function

$x \in \mathcal{U}$ is the element to encode. $r_{self} \in \mathbb{Z}_p$ is the random exponent generated by the party itself. $R_{other} \in \mathbb{G}$ is the random challenge received from the other party. P_{self} and P_{other} represent the names of the party itself and the party it is communicating with.

Fig. 2. The EncodeElem function

3.4 Encodings for Symmetric Policies

As shown in Figure 2, the encoding is computed as a product of multiple terms, where each term is the result of a bilinear pairing. Intuitively, the encodings satisfy the following properties:

- **Conformity.** If both parties have an element $x \in \mathcal{U}$ and the appropriate authorizations, their respective encodings of the element x will be the same. Therefore, the encoding for element x will appear in the intersection at the end of the PSI protocol.
- **Unforgeability.** If a party does not have appropriate authorizations for the element x, it is unable to forge the correct encoding for x. In this way, a dishonest party who does not possess authorizations for element x cannot find out whether the other party has element x.

The encoding contains two corresponding terms for each element-authority pair (x, auth_i):

- *Response to the other party's challenge.* The first term, $\mathsf{e}(\sigma_i, R_{\text{other}})$, is a response to the other party's challenge R_{other}. Intuitively, if a party does not possess an authorization from auth_i, then it will not be able to generate this part of the encoding.
- *Anticipated response from the other party to one's own challenge.* The second term, $\mathsf{e}(H(x, P_{\text{other}}), \mathsf{vk}_i)^{r_{\text{self}}}$, is the anticipated response from the other party for one's own challenge R_{self}. Note that a party is always able to compute the anticipated response for its own challenge, even without knowing the other party's signature, since a party knows the exponent r_{self} of a challenge generated by itself. Let $\sigma'_i := H(x, P_{\text{other}})^{\mathsf{sk}_i}$ denote the signature given to P_{other} from auth_i on element x. It is not hard to see that

$$\mathsf{e}(\sigma'_i, R_{\text{self}}) = \mathsf{e}(H(x, P_{\text{other}}), \mathsf{vk}_i)^{r_{\text{self}}}$$

In other words, if P_{other} has the correct signature from auth_i, its actual response to P_{self}'s challenge should match the response anticipated by P_{self}. In summary, this term enforces that the other party can only compute the encoding if it has a signature from the correct authority.

Theorem 1. *The PPSI scheme described in this section is secure against malicious adversaries, assuming 1) the underlying PSI protocol is simulatable in the malicious model; 2) the Computational Bilinear Diffie-Hellman (CBDH) assumption holds in the bilinear group $\mathbb{G}$; and 3) the hash function H is a random oracle.*

Proof overview. We now give an overview of our proof, and defer the detailed proof to Appendix 5. We first define a hybrid protocol by replacing the PSI protocol with an ideal functionality for PSI. Due to the sequential modular composition theorems by Canetti [7], it suffices to prove that the hybrid protocol securely computes the ideal functionality defined in Section 2. We then construct a simulator which is given black-box access to a hybrid-world adversary $\mathcal{A}$. We show that if the encoding scheme is unforgeable in some sense, then the joint output distribution of all parties in the ideal world is indistinguishable from the joint output distribution in the hybrid world.

The description of our protocol in Figure 1 does not hide the number of authorized elements from the other party. If this number is also considered sensitive, a party can pad its set of encodings with random dummy encodings, and use the resulting set as inputs to the PSI protocol. Effectively, this reveals to the other party the total number of authorized elements and dummy elements.

Another possible method to hide the size of one's set is to use a Size-Hiding PSI protocol in place of the PSI protocol used in our construction. Our security proofs would still hold if the Size-Hiding PSI protocol is simulatable in the malicious model. Notably, Ateniese *et al.* recently propose a Size-Hiding PSI protocol secure under the semi-honest model [3]. Therefore, it is conceivable that a Size-Hiding PSI protocol in the malicious model will become available in the near future.

4 Extensions for Richer Policies

In this section, we describe how to compute the encodings in the intersection protocol for different kinds of policies. Our main construction in Section 3.3 supports simple symmetric policies, and we now incrementally add support for asymmetric policies, attributes, bundles, and DNFs.

4.1 Asymmetric Policies

So far we have focused on symmetric policies, where the authorities associated with each element depend on the element itself. In other application scenarios, the right authority may depend on both the element and the party performing set intersection.

function $\mathsf{EncodeElem}(x, r_{\text{self}}, R_{\text{other}}, P_{\text{self}}, P_{\text{other}})$
  $c \leftarrow 1 \in \mathbb{G}_T$
  for $\mathsf{auth}_i \in F(x, P_{\text{self}})$ **do**
    Let σ_i denote auth_i's signature for P_{self} on x.
    $c \leftarrow c \cdot \mathsf{e}(\sigma_i, R_{\text{other}})$
  end for
  for $\mathsf{auth}_i \in F(x, P_{\text{other}})$ **do**
    Let σ_i denote auth_i's signature for P_{other} on x.
    $c \leftarrow c \cdot \mathsf{e}(H(x, P_{\text{other}}), \mathsf{vk}_i)^{r_{\text{self}}}$
  end for
  return c
end function

Fig. 3. The EncodeElem function for asymmetric policies

Let $\mathcal{U}$ denote a countable universe of elements, let $\mathcal{P}$ denote the set of all parties, and let Λ denote the set of authorities. We denote asymmetric policies using a publicly known policy function $F : \mathcal{U} \times \mathcal{P} \rightarrow 2^{\Lambda}$. F outputs the set of appropriate authorities given an element and a party. For example, if $F(x, P) = \{\mathsf{auth}_1, \mathsf{auth}_2\}$, this means that auth_1 and auth_2 must sign element x for party P.

Figure 3 describes how to modify the EncodeElem function to support asymmetric policies.

The idea is essentially the same as the symmetric case. If auth_i must sign element x for party P_{self}, then P_{self} computes the term $\mathsf{e}(\sigma_i, R_{\text{other}})$, which is a response to the challenge from P_{other}. If auth_j must sign element x for party P_{other}, then P_{self} computes the term $\mathsf{e}(H(x, P_{\text{other}}), \mathsf{vk}_j)^{r_{\text{self}}}$, which is the anticipated response from P_{other} to one's own challenge. The final encoding for an element is basically the product of all responses to the other party's challenge, and all anticipated responses from the other party.

4.2 Attributes

Authorities may wish to attach attributes to an element when making authorizations. For example, attributes may be used to determine the type or level of authorization given (e.g., sensitivity level of medical records). We show that our construction can be extended to support policy attributes.

Let $\mathcal{V}$ denote the set of all possible attributes. Suppose a public function $F : \mathcal{U} \times \mathcal{P} \rightarrow 2^{\Lambda \times \mathcal{V}}$ exists which outputs the necessary (authority, attribute) pairs given an element and a party. For example, if

$$F(x, P) = \{(\mathsf{auth}_1, \mathsf{attr}_1), (\mathsf{auth}_1, \mathsf{attr}_2), (\mathsf{auth}_5, \mathsf{attr}_3)\},$$

it means that for party P to be a rightful owner of element x, it is necessary that auth_1 has signed element x with attributes attr_1 and attr_2 for party P, and auth_5 has signed element x with attribute attr_3 for party P.

function $\mathsf{EncodeElem}(x, r_{\text{self}}, R_{\text{other}}, P_{\text{self}}, P_{\text{other}})$
  $c \leftarrow 1 \in \mathbb{G}_T$
  for $(\mathsf{auth}_i, \mathsf{attr}) \in F(x, P_{\text{self}})$ **do**
    Let σ_i denote auth_i's signature for P_{self} on x and attribute attr.
    $c \leftarrow c \cdot \mathsf{e}(\sigma_i, R_{\text{other}})$
  end for
  for $(\mathsf{auth}_i, \mathsf{attr}) \in F(x, P_{\text{other}})$ **do**
    Let σ_i denote auth_i's signature for P_{other} on x and attribute attr.
    $c \leftarrow c \cdot \mathsf{e}(H(x, \mathsf{attr}, P_{\text{other}}), g^{\mathsf{vk}_i})^{r_{\text{self}}}$
  end for
  return c
end function

Fig. 4. The EncodeElem function supporting attributes and asymmetric policies

To support attributes, first, the authorities need to incorporate the attribute values in the hash when computing signatures. To authorize element x with attribute attr to party P, auth_i now computes the following signature:

$$\sigma \leftarrow H(x, \mathsf{attr}, P)^{\mathsf{sk}_i}$$

Second, the EncodeElem function needs to be modified to incorporate the attributes as in Figure 4.

4.3 Bundles

A group of elements may form a bundle. The bundle should appear in the intersection if both parties have all elements in the bundle, as well as the appropriate authorizations. Otherwise, the bundle should not appear in the intersection, and neither party should learn any partial information about elements in the bundle that the other party has.

Our scheme can be easily adapted to handle bundles by combining the encoding of each element of the bundle to produce a single encoding for the entire bundle. Specifically, the bundle's encoding is the product of the encodings of its elements.

4.4 Disjunctions and DNFs

So far, we have considered conjunctive policies. More generally, policies may also contain disjunctions, or Disjunctive Normal Forms (DNFs). For example, if a hospital A may want to share Carol's record with hospital B either if the record has low sensitivity and hospital B has permissions to receive low sensitivity records from Carol, or the record is cardiology related, and hospital B has permissions to retrieve Carol's cardiology records.

As another example, imagine two online stores (e.g., Dell and Newegg) want to investigate a consumption pattern of their shared customers. Specifically, they

want to determine which customers have bought both a computer from Dell and a monitor from Newegg. Therefore, they need to perform a set intersection operation on their sales datasets. Meanwhile, to prevent each company from inserting fictitious records, each sales record must be authorized by a recognized credit company, Mastercard *or* Visa.

In general, for parties P_A and P_B to share an element x, a DNF-style policy of the following form must be satisfied:

$$\mathsf{policy} \;:=\; C_1 \vee C_2 \vee \ldots \vee C_k$$

where each $C_i (1 \leq i \leq k)$ is a conjunctive clause of the form:

$$(\mathsf{auth}_{i_1}, P_A, x, \mathsf{attr}_1) \wedge \ldots \wedge (\mathsf{auth}_{i_\ell}, P_A, x, \mathsf{attr}_\ell)$$
$$\wedge (\mathsf{auth}_{j_1}, P_B, x, \mathsf{attr}_1) \wedge \ldots \wedge (\mathsf{auth}_{j_{\ell'}}, P_B, x, \mathsf{attr}_{\ell'})$$

In the above, each tuple $(\mathsf{auth}_i, P, x, \mathsf{attr})$ means that "auth_i gave authorizations to party P on element x with attribute attr". Specifically, each conjunctive clause specifies the policies for party P_A and P_B respectively.

Our basic construction can be extended to support DNFs, with the caveat that each party reveals to the other party which conjunctive clause is satisfied for an element. The idea is quite straightforward: for each conjunctive clause, each party uses the algorithm described in Figure 4 to compute an encoding. The encoding for an element is now the union of all encodings corresponding to all conjunctive clauses. Furthermore, each party will use the union of all encodings for all elements as inputs to the PSI protocol.

5 Proofs of Security

Suppose the PSI protocol we use in the protocol is fully simulatable under the malicious model. Due to the sequential modular composition theorems by Canetti [7], we can replace the PSI module in our protocol with the ideal functionality for PSI. We refer to the resulting protocol as the hybrid protocol. We formally describe the hybrid protocol below. Although not explicitly stated, parties P_A and P_B may abort the protocol at any message boundary. The proofs for Lemma 1 and 2 are available in the full version [1] of this paper.

- P_A picks random $r_A \in \mathbb{Z}_p$, and sends to P_B the value $R_A := g^{r_A} \in \mathbb{G}$.
- P_B picks random $r_B \in \mathbb{Z}_p$, and sends to P_A the value $R_B := g^{r_B} \in \mathbb{G}$.
- P_A computes $C_A \leftarrow \{\mathsf{EncodeElem}(x, r_A, R_B, P_A, P_B) | x \in S_A\}$ and sends it to $\mathsf{T}_{\mathrm{PSI}}$. $\mathsf{T}_{\mathrm{PSI}}$ sends $|C_A|$ to P_B.
- P_B computes $C_B \leftarrow \{\mathsf{EncodeElem}(x, r_B, R_A, P_B, P_A) | x \in S_B\}$ and sends it to $\mathsf{T}_{\mathrm{PSI}}$. $\mathsf{T}_{\mathrm{PSI}}$ sends $|C_B|$ to P_A.
- $\mathsf{T}_{\mathrm{PSI}}$ computes $C' := C_A \cap C_B$, and sends C' to P_A.
- $\mathsf{T}_{\mathrm{PSI}}$ sends C' to P_B.

Due to the sequential modular composition theorems by Canetti [7], it suffices to show that the hybrid protocol is secure as stated by Lemma 1.

Definition 2. *Let E denote an event sequence. Let* $\text{IDEAL}_{\mathcal{S},E}$ *denote the joint output distribution of all parties and the adversary $\mathcal{S}$ in the ideal world, under event sequence E. Let* $\text{HYBRID}_{\mathcal{A},E}$ *denote the joint output distribution of all parties and the adversary $\mathcal{A}$ in the hybrid world, under event sequence E. We say that the hybrid protocol securely computes the ideal functionality defined in Section 2.3, if for any polynomial-time adversary $\mathcal{A}$ in the hybrid world, there exists simulator $\mathcal{S}$ in the ideal world, such that for any sequence of events E,*

$$\text{IDEAL}_{\mathcal{S},E} \stackrel{c}{=} \text{HYBRID}_{\mathcal{A},E}$$

where $\stackrel{c}{=}$ denotes computational indistinguishability.

Lemma 1 (Security of the hybrid protocol). *Assume that the CBDH assumption holds in the bilinear group $\mathbb{G}$, and the hash function H is a random oracle. Then, the hybrid protocol described earlier securely computes the ideal functionality defined in Section 2.3.*

Lemma 2 (Unforgeability of encodings). *Assume that the CBDH assumption holds in the group $\mathbb{G}$, and the hash function H is a random oracle. Let $\mathcal{A}$ denote polynomial-time adversary in the hybrid protocol, who has full control of all corrupted parties. Let P_A denote a corrupted party, and assume that P_A has not received* auth_i*'s signature on element x. Then, during a set intersection protocol between P_A and any honest party P_B, $\mathcal{A}$ is unable to compute the correct encoding* $\mathsf{EncodeElem}(x, r_A, R_B, P_A, P_B)$ *except with negligible probability. In the above, $R_B \in_R \mathbb{G}$ is chosen at random by P_B, and $r_A \in \mathbb{Z}_p$ is chosen arbitrarily by the adversary $\mathcal{A}$.*

6 Performance

In this section, we present the asymptotic complexities and experimental performance of our protocol.

6.1 Asymptotic Complexities

We first analyze the performance of our basic construction (described in Section 3.3) supporting symmetric policies. Later, in Section 6.3, we discuss the performance of the various extensions (described in Section 4).

The efficiency of our protocol depends on both the number of elements (n) and the number of authorities per element (m). We now present asymptotic bounds for the amount of computation, amount of bandwidth, and the number of communication rounds.

Computation: $O(nm+n \log\log n)$ The encoding phase performs a constant number of operations for each element-authority pair and is hence $O(nm)$. It computes a single encoding for each element resulting in $O(n)$ encodings. Those encodings are the input for the PSI phase, and by using the protocol by Hazay and Nissim [15], we can perform the PSI phase in $O(n \log\log n)$ time. The recovery phase is trivially $O(nm)$ and the challenge phase is $O(1)$. Summing up the above, the total computation is $O(nm + n \log\log n)$.

Bandwidth: $O(n)$ The communication between the two parties consists of the PSI protocol's communication and the two group elements sent during the challenge phase. Since the input size for the PSI is n, using the PSI protocol by Hazay and Nissim [15], the bandwidth overhead for the PSI phase is $O(n)$. Therefore, the combined communication bandwidth for our scheme is $O(n)$.

Rounds: $O(1)$ The PSI protocol by Hazay and Nissim [15] consists of $O(1)$ communication rounds. We add one additional round for the challenge phase.

Note that our construction introduces only a small overhead on top of PSI, namely, a single round of extra communication where 2 group elements are exchanged, and at most nm bilinear pairings. And with this small additional overhead, we provide the ability to support rich privacy policies previously not possible with existing PSI and APSI schemes.

6.2 Empirical Performance

Our protocol can be broken down into two time consuming phases: (1) encoding elements, and (2) performing standard Private Set Intersection (PSI). There is a large body of existing work on building efficient PSI protocols [9, 10, 11, 12, 13, 15, 16, 17, 18], and one can plug into our construction any existing PSI protocol that is fully simulatable under a malicious adversary model. Therefore, our experimental analysis below focuses on the additional overhead introduced by the encoding phase.

We generated 2,000 random elements with attributes and then computed the signatures for 2 parties by 5 authorities. We used different authorities for each element-authority pair, hence we have the total number of authorities $|\Lambda| =$ 10,000. We set the maximum number of authorities per element to be $m = 5$. We then varied $m = 1, \ldots, 5$ by choosing a random subsets of the corresponding authorities for each element, and computed all of the the element encodings in parallel. After repeating this experiment 20 times, we calculated the average encoding time per element and standard deviation. The results are shown in Table 3.

Our experiment was implemented in C# and was run on 64-bit Windows 7 with an Intel Core i7 3.33 GHz CPU and 12GB of RAM (although the experiment used much less memory). For all of the pairing and elliptic curve operations, we used the Pairing Based Crypto Library [19].

Table 3. The time (in ms) for encoding an element given the number of authorities for that element. These results are for the symmetric policy construction with attributes.

m	1	2	3	4	5
average	1.70	3.10	4.45	5.65	7.07
std. dev.	0.06	0.17	0.22	0.04	0.27

6.3 Performance for Rich Policies

So far, we focused on the performance of the basic construction supporting symmetric policies. The performance of our protocols supporting richer policies can be analyzed in a similar fashion.

Asymmetric Policies. The performance for asymmetric policies is essentially the same as the performance for symmetric policies. Therefore, encoding n elements each having at most m authorities per element using an asymmetric policy is at least as fast as encoding using a symmetric policy for the same n and m.

Attributes. With attribute-enriched policies, the number of bilinear pairings is the number of (element, authority, attribute) tuples for each party.

Bundle.s The cost of encoding a bundle scales linearly with the number of elements. For example, the cost of encoding a bundle of b elements is b times times the cost of encoding a single element. This is due to the fact that the elements of the bundle have to be first encoded individually. Combining them incurs a series of elliptic point multiplications, but their cost is significantly outweighed by the pairing function that is applied to each element.

DNF policies. Each DNF policy consists of multiple conjunctive clauses. The cost of encoding an element under a DNF policy is simply the sum of the cost of encoding each conjunctive clause, where the cost for encoding a conjunctive clause has been discussed earlier – depending on whether the conjunctive clause is symmetric, asymmetric, attribute-enriched, etc.

With a DNF policy consisting of k conjunctive clauses, the encoding for an element will consist of k group elements instead of one.

To summarize, suppose the maximum number of conjunctive clauses for a DNF policy is k, and the maximum number of literals for a conjunctive clause is m. Then, the communication overhead of our protocol will be $O(nk)$, and the computational overhead will be $O(nmk + nk \log\log(nk))$.

7 Conclusion

We introduced a new cryptographic paradigm for private set intersection with rich policies, allowing two parties to selectively share data while satisfying privacy policies. Our protocol ensures that only properly authorized elements which satisfy certain privacy policies appear in the set intersection. Our protocols support rich policies, including conjunctive and disjunctive policies, attribute-enriched policies, asymmetric policies, and bundles of elements. We prove that our scheme is secure under the malicious model, given the CBDH assumption, the security of the underlying PSI protocol, and assuming the random oracle model.

References

1. Policy-enhanced private set intersection: Sharing information while enforcing privacy policies. Techinical Report (2012), http://eprint.iacr.org/2011/509.pdf
2. Anton, A.I., Eart, J.B., Vail, M.W., Jain, N., Gheen, C.M., Frink, J.M.: HIPAA's effect on web site privacy policies. IEEE Security and Privacy 5 (January 2007)
3. Ateniese, G., Cristofaro, E.D., Tsudik, G.: Size-hiding private set intersection. Cryptology ePrint Archive, Report 2010/220 (2010), http://eprint.iacr.org/
4. Bertino, E., Ooi, B.C., Yang, Y., Deng, R.H.: Privacy and ownership preserving of outsourced medical data. In: ICDE (2005)
5. Boneh, D., Lynn, B., Shacham, H.: Short Signatures from the Weil Pairing. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 514–532. Springer, Heidelberg (2001)
6. Camenisch, J., Zaverucha, G.M.: Private Intersection of Certified Sets. In: Dingledine, R., Golle, P. (eds.) FC 2009. LNCS, vol. 5628, pp. 108–127. Springer, Heidelberg (2009)
7. Canetti, R.: Security and composition of multi-party cryptographic protocols. Journal of Cryptology (1998)
8. De Cristofaro, E., Jarecki, S., Kim, J., Tsudik, G.: Privacy-Preserving Policy-Based Information Transfer. In: Goldberg, I., Atallah, M.J. (eds.) PETS 2009. LNCS, vol. 5672, pp. 164–184. Springer, Heidelberg (2009)
9. De Cristofaro, E., Kim, J., Tsudik, G.: Linear-Complexity Private Set Intersection Protocols Secure in Malicious Model. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 213–231. Springer, Heidelberg (2010)
10. De Cristofaro, E., Tsudik, G.: Practical Private Set Intersection Protocols with Linear Complexity. In: Sion, R. (ed.) FC 2010. LNCS, vol. 6052, pp. 143–159. Springer, Heidelberg (2010)
11. Dachman-Soled, D., Malkin, T., Raykova, M., Yung, M.: Efficient Robust Private Set Intersection. In: Abdalla, M., Pointcheval, D., Fouque, P.-A., Vergnaud, D. (eds.) ACNS 2009. LNCS, vol. 5536, pp. 125–142. Springer, Heidelberg (2009)
12. Freedman, M.J., Ishai, Y., Pinkas, B., Reingold, O.: Keyword Search and Oblivious Pseudorandom Functions. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 303–324. Springer, Heidelberg (2005)
13. Freedman, M.J., Nissim, K., Pinkas, B.: Efficient Private Matching and Set Intersection. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 1–19. Springer, Heidelberg (2004)
14. Hazay, C., Lindell, Y.: Efficient Protocols for Set Intersection and Pattern Matching with Security Against Malicious and Covert Adversaries. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 155–175. Springer, Heidelberg (2008)
15. Hazay, C., Nissim, K.: Efficient Set Operations in the Presence of Malicious Adversaries. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 312–331. Springer, Heidelberg (2010)
16. Jarecki, S., Liu, X.: Efficient Oblivious Pseudorandom Function with Applications to Adaptive OT and Secure Computation of Set Intersection. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 577–594. Springer, Heidelberg (2009)
17. Jarecki, S., Liu, X.: Fast Secure Computation of Set Intersection. In: Garay, J.A., De Prisco, R. (eds.) SCN 2010. LNCS, vol. 6280, pp. 418–435. Springer, Heidelberg (2010)
18. Kissner, L., Song, D.: Privacy-Preserving Set Operations. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 241–257. Springer, Heidelberg (2005)
19. Lynn, B.: Pairing-based cryptography library, http://crypto.stanford.edu/pbc/

Efficiently Shuffling in Public

Udaya Parampalli[1], Kim Ramchen[1], and Vanessa Teague[1]

Department of Computer Science and Software Engineering
University of Melbourne
{udaya,ramchen,vteague}@csse.unimelb.edu.au

Abstract. We revisit *shuffling in public* [AW07a], a scheme which allows a shuffle to be precomputed. We show how to obfuscate a Paillier shuffle with $O(N \log^{3.5} N)$ exponentiations, leading to a very robust and efficient mixnet: when distributed over $O(N)$ nodes the mixnet achieves mixing in polylogarithmic time, independent of the level of privacy or verifiability required. Our construction involves the use of layered Paillier applied to permutation networks. With an appropriate network the shuffle may be confined to a particular subset of permutations, for example to rotations. While it is possible that the mixnet may produce biased output, we show that certain networks lead to an acceptable bias-efficiency tradeoff.

Keywords: Public key obfuscation, homomorphic encryption, electronic voting.

1 Introduction

A re-encryption mix permutes and re-encrypts its input [PIK94]. A series of mixes is called a *mixnet* and guarantees that input ciphertexts cannot be linked to the decrypted output unless all mixes collude. A *proof of shuffle* allows a mix to prove they have correctly processed their input. A fundamental challenge in electronic voting is the design of mixnets that can accomodate a large number of encrypted ballots in a relatively short space of time. An additional goal is *robustness*, a mixnet must be able to recover from the failure of faulty or dishonest mixes. In this paper we present efficient constructions for *shuffling in public* [AW07a], a scheme which allows a shuffle to be *precomputed* before any input is received. Evaluating the precomputed shuffle upon input is public, that is requires no secret information and can be performed even by *untrusted* parties. Evaluation is also highly *parallelisable*, thus the work required to mix votes can be distributed over an arbitrary number of workstations at election time.

1.1 Improving the Efficiency and Robustness of Mixnets

Most mixnets achieve robustness through the detection and replacement of corrupt mixes. Although a few schemes [Cha81, PIK94, GZB+02] verify that the mixnet as a whole functioned properly, these schemes provide no way to recover from errors or identify dishonest mixes. The most common method to audit a

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 431–448, 2012.

mix is to require it to output a zero knowledge proof. Cut and choose techniques [SK95] are generally applicable but inefficient, hence much work has been devoted to optimising the proof of a shuffle.

Abe [Abe99] and later Jakobsson and Juels [JJ99] presented the first practical proofs of a shuffle based upon re-encryption permutation networks. Furukawa and Sako [FS01] used a commitment to a permutation matrix and Neff [Nef01] used unique factorisation of polynomials to prove a shuffle of ElGamal ciphertexts with improved efficiency. Efficient arguments and proofs have also been devised in the case that the shuffle is restricted to a subset of permutations [RW04, dHSŠV09]. Generic techniques may be used to further optimise the above proofs, including pre-computation of re-encryption factors, fixed base and multi-exponentiation and batch proof techniques [BGR98] and PRGs for challenge generation. Wikström [Wik09] has also observed that a proof of shuffle may be split into offline and online phases. A mix provably commits to its permutation offline allowing a highly efficient commitment-consistent proof in the online phase.

Despite these enhancements there remain inherent limitations on the robustness and efficiency of mixnets which makes use of *private* techniques for online mixing. Firstly, if a mix is detected cheating then the mixnet must either be restarted or delayed until a replacement is found. Secondly, the opportunities for parallelisation are quite limited. As each mix must keep its permutation secret, it must perform its round of mixing and output a correct proof without assistance. Therefore the runtime is at least *linear* in the number of votes and mixes. Thirdly, it is commonly assumed that each mix server in a mixnet should belong to a different organisation (e.g. political party). Online private mixing depends upon a quorum of these co-operating in the short space of time before tallying begins.

In contrast, two schemes *shuffling in public* [AW07a] and *offline/online mixing* [AW07b] allow a shuffle to be precomputed. These schemes imply that no mix servers need be present at election time for mixing to take place. A major downside of offline/online mixing is that each voter requires a separate key to encrypt their vote. Additionally the scheme significantly restricts the number and size of votes. The main disadvantage of shuffling in public is its inefficiency, with generation and evaluation of the precomputed shuffle requiring $O(N^2)$ exponentiations. In this work we reduce both phases to $O(N \log^{3.5} N)$ exponentiations. Experiments indicate that our scheme is faster when $N > 1200$.

1.2 Shuffling in Public

The goal of shuffling in public is the *public-key obfuscation* of the shuffle phase of a mix-net comprising either a *decryption shuffle* or *re-encryption shuffle* functionality (program) [AW07a]. Informally, a public-key obfuscator $\mathcal{O}$ takes a program F and outputs a new program $\mathcal{O}(F)$ which outputs encryptions of F's outputs. That is $\exists \diamond \forall x \; \mathcal{O}(F) \diamond x = O(F(x))$ for some encryption function O and we say the operator $\diamond$ *evaluates* the obfuscated program on input x. A formal model is proposed in Definition 3 [AW07a] which builds upon an earlier definition by

Ostrovsky and Skeith [OS07]. Adida and Wikström present obfuscators for decryption and re-encryption shuffles in the BGN [BGN05] and Paillier [Pai99] cryptosystems respectively. They also prove that their obfuscators are semantically secure (Definition 4 [AW07a]). Given a set of parties who sample and obfuscate a shuffle before any input is received, one can construct a mixnet provided that joint decryption is verifiable.

1.3 Our Contributions

We public key obfuscate a Paillier shuffle using permutation networks. Our obfuscated shuffle comprises $O(N \log N)$ ciphertexts and requires $O(N \log^{3.5} N)$ exponentiations to generate and evaluate, rather than $O(N^2)$ ciphertexts and exponentiations in [AW07a]. Utilising a suitable network, we can restrict the space of permutations, for example we can obfuscate homomorphic rotation. We propose a distributed protocol for sampling and obfuscating a shuffle allowing the construction of a verifiable mixnet. A side effect of the use of permutation networks is that the resulting distribution over permutations may be biased. However it is possible to reduce the bias at the expense of increasing the complexity to $O(N \log^c N)$ for a constant $c > 3.5$. Moreover for some applications weaker anonymity may be acceptable.

1.4 Outline

The paper is organised as follows. In Section 2 we discuss cryptographic preliminaries. In Section 3 we review permutation networks. In Section 4 we show how to obfuscate shuffles of Damgård-Jurik ciphertexts as well as an operation to compose obfuscations. These ideas when applied to permutation re-encryption networks lead to an improved obfuscator for a Paillier shuffle. In Section 5 we provide a distributed protocol for sampling and obfuscating a shuffle via an arbitrary permutation network. In Section 6 we analyse the properties of the resulting mixnet and prove that it is secure under standard assumptions. In Section 7 we conclude and suggest future directions.

2 Preliminaries

2.1 Notation

We denote by κ the security parameter (i.e the bitlength of the RSA modulus), and say that a function $\epsilon(\kappa)$ is negligible if for each $c \in \mathbb{N}$ there exists $\kappa_0 \in \mathbb{N}$ such that for all $\kappa > \kappa_0$, $\epsilon(\kappa) < \kappa^{-c}$. We denote probabilistic polynomial time by PPT and assume all adversaries are PPT Turing machines. Let Σ_N be the symmetric group on N elements. By a "random encryption" of a message m, we will implicity mean an encryption of m where the randomisation factor is chosen uniformly and independently from the randomisation space. Suppose a

PPT Turing Machine A distinguishes distributions D_1 and D_0. We denote by $Adv(A)$ the advantage of A in distinguishing D_1 and D_0, where

$$Adv(A) = |\Pr_{t \leftarrow D_1}[A(t) = 1] - \Pr_{t \leftarrow D_0}[A(t) = 1]|$$

is a function of κ.

2.2 Homomorphic Encryption

Definition 1 (Homomorphic). *The public key of a homomorphic cryptosystem $\mathcal{CS} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ specifies a message space $(M_{pk}, +)$, a randomiser space $(R_{pk}, \cdot)$ and a ciphertext space $(C_{pk}, \times)$ all of which are abelian groups. Encryption is homomorphic*

$$\mathcal{E}_{pk}(m, r) \times \mathcal{E}_{pk}(m', r') = \mathcal{E}_{pk}(m + m', r \cdot r').$$

For any homomorphic cryptosystem we can define a scalar homomorphism generically

$$c \otimes \mathcal{E}_{pk}(m, r) = \underbrace{\mathcal{E}_{pk}(m, r) \times \ldots \times \mathcal{E}_{pk}(m, r)}_{c} = \mathcal{E}_{pk}(cm, r^c).$$

Definition 2 (Indistinguishability under Chosen Plaintext Attacks). *Let $\mathcal{CS} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a cryptosystem and $A = (A_1, A_2)$ be an adversary. Define*

Experiment $Exp^{IND-CPA-b}_{A,\mathcal{CS}}(\kappa)$:

$(pk, sk) \leftarrow \mathcal{G}(1^\kappa); (m_0, m_1, \delta) \leftarrow A_1(pk) : |m_0| = |m_1|; c \leftarrow \mathcal{E}_{pk}(m_b);$

$v \leftarrow A_2(m_0, m_1, \delta, c)$

return v

and let

$$Adv(A) = |\Pr[Exp^{IND-CPA-1}_{A,\mathcal{CS}}(\kappa) = 1] - \Pr[Exp^{IND-CPA-0}_{A,\mathcal{CS}}(\kappa) = 1]|$$

Then $\mathcal{CS}$ satisfies indistinguishability under chosen plaintext attacks (IND-CPA) if for any PPT A, $Adv(A)$ is negligible.

2.3 The Damgård-Jurik Cryptosystem

The Damgård-Jurik Cryptosystem [DJ01] is a generalisation of the Paillier Cryptosystem [Pai99] based on the isomorphism $\mathbb{Z}_{n^i} \times \mathbb{Z}^*_n \to \mathbb{Z}^*_{n^{i+1}}$. Let $\mathcal{E}_{i,n}$ be the i^{th} generalised Paillier encryption, where $i \leq s$ for some integer s.

Key Generation. Let $n = pq$ be an RSA modulus. Let $\lambda = lcm(p-1, q-1)$. Compute via the Chinese Remainder Theorem d such that $d = 1 \mod n^s$ and $d = 0 \mod \lambda$.

Encryption. Given a plaintext $m \in \mathbb{Z}_{n^i}$, choose a random $r \in \mathbb{Z}^*_n$. Let $\mathcal{E}_{i,n}(m, r) = (1+n)^m r^{n^i} \mod n^{i+1}$.

Decryption. $\mathcal{D}_{i,n}(c) = \log_{(1+n)} c^d \bmod n^{i+1}$. We have:
$c^d = (1+n)^{md} r^{n^i d} = (1+n)^{md \bmod n^i} r^{n^i d} = (1+n)^m \bmod n^{i+1}$.
One can extract m given $(1+n)^m \bmod n^{i+1}$ using the binomial-expansion based algorithm presented in Section 3 of [DJ01].

Semantic Security. Semantic security of the scheme is based upon the *Decision Composite Residuosity Assumption* (DCRA) [Pai99], which states that no PPT algorithm can distinguish the uniform distribution on $\mathbb{Z}^*_{n^2}$ from the uniform distribution on the subgroup of n^{th} residues in $\mathbb{Z}^*_{n^2}$. In fact an adversary with advantage $\epsilon_i(\kappa)$ against $\mathcal{E}_{i,n}$ implies an adversary with advantage at least $\epsilon_i(\kappa)/i$ against $\mathcal{E}_{1,n}$ as shown in [Gjø05].

2.4 Privacy of a Shuffle

Nguyen et al. in [NSNK04] formally define shuffle privacy by observing that a shuffle of ciphertexts is an "encryption" that hides the permutation. The corresponding security notion is "indistinguishability under chosen permutation attacks" (IND-CPA$_S$). A discussion is included in Appendix A.

3 Permutation Networks

A permutation network is a circuit composed of configurable switches that permutes a set of inputs. For convenience we assume that every switch accepts the same number of inputs. This includes the important special cases of rotation and shuffling - the networks we present are also *optimal* in the sense that the size and depth are minimal. We assume that the number of inputs, N, is a power of two.

Definition 3. *Suppose Ψ is a permutation network of dimension $\Delta \times W$. Then each layer consists of W independent switches where each switch imposes a fixed mapping on $N' = N/W$ inputs when its control bit is true. Let the switches in the i^{th} layer partition the set of inputs $\{1, \ldots, N\}$ into subsets $V_{i,1}, \ldots, V_{i,W}$ with corresponding mappings $\sigma_{i,1}, \ldots, \sigma_{i,W}$. Let $A^{(i)}$ be the adjacency matrix of the i^{th} layer. Then*

$$A^{(i)}_{lm} = \begin{cases} 1 \text{ if } l = m \vee \sigma_{i,j}(l) = m \text{ for some } j \in [W] \\ 0 \text{ otherwise.} \end{cases} \tag{1}$$

Let the control bits of the i^{th} layer be $b_{i,1}, \ldots, b_{i,W}$. Then the permutation imposed by that layer is $\pi_i \triangleq \sigma_{i,1}^{b_{i,1}} \ldots \sigma_{i,W}^{b_{i,W}}$. Moreover the state of the network is $\pi \overset{\Psi}{=} \pi_\Delta \ldots \pi_1$.

3.1 Rotation

We describe the *barrel shifter* network which is capable of implementing every possible rotation. Let the set of rotations be $\phi^0, \ldots, \phi^{N-1}$. Observe that a switch

on N inputs can output ϕ^0 or ϕ^{2^i} for each i in $[\log_2 N]$. Cascading these switches together, we obtain a network that has $\Delta = \log_2 N$ and $W = 1$. Clearly $\phi^j : 0 \leq j < N$ is output iff $b_{\log_2 N-1} \dots b_0 = j_2$, therefore each rotation is possible. Figure 1(a) shows an example for $N = 8$.

3.2 Shuffling

We require a *rearrangeable* permutation network, i.e. one that is capable of implementing all possible permutations of its input. Since there are $N!$ possible outputs, at least $\log_2 N! = \Omega(N \log N)$ switches are required. A number of networks meet this bound for example the Waksman network [Wak68] consists of $N \log_2 N - N + 1$ switches. For convenience we will use the slightly simpler Beneš network [Ben64] which consists of a butterfly network composed with a reflected butterfly network, where the middle layer is shared. Note that the network has $\Delta = 2 \log_2 N - 1, W = N/2$. Figure 1(b) shows an example for $N = 8$.

3.3 Biased Networks

Abe and Hoshino observed that setting each switch uniformly and independently in most permutation networks leads to a biased distribution over Σ_N [AH01]. For example, in the Beneš network, there are $2^{(N/2)(\log_2 N-1)}$ switch settings that produce the identity permutation, while other permutations result from only one switch setting. This issue cannot be avoided in Protocol 1 leading to biased output of the mixnet. However we provide some results that suggest that for certain applications the protocol may be acceptable.

Definition 4. *Suppose Ψ is a permutation network. Let $k \leq N$ be a positive integer. Let C_k and P_k be the set of ordered (resp. unordered) k-tuples whose elements are drawn without replacement from $1, \dots, N$. For $\boldsymbol{t} \in C_k$, let $C_{\boldsymbol{t}}$ and $P_{\boldsymbol{t}}$ be the distributions of $\{\Psi(t_1), \dots, \Psi(t_k)\}$ and $(\Psi(t_1), \dots, \Psi(t_k))$ respectively, when all switches are set uniformly at random. The bias over ordered (resp. unordered) k-tuples of Ψ is*

$$\epsilon_{C_k}(N) = \max_{\boldsymbol{t} \in C_k} \|C_{\boldsymbol{t}} - U(C_k)\|, \quad \epsilon_{P_k}(N) = \max_{\boldsymbol{t} \in C_k} \|P_{\boldsymbol{t}} - U(P_k)\|$$

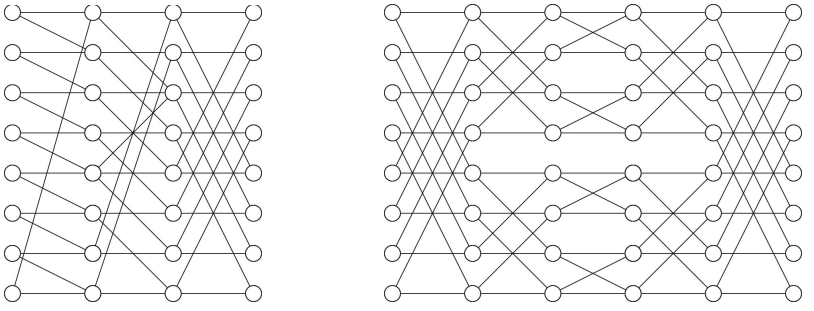
where U is the uniform distribution and $\|\cdot - \cdot\|$ denotes the statistical distance.

Proposition 1. *The bias over 1-tuples of the Beneš network is 0.*

Proof. It is well-known that the Butterfly network sends any input to each output with probability $1/N$ when set uniformly. This property is maintained when the network is composed with its reflection.

Theorem 1 (Lemma 4.2 [CKKK01]). *One can construct a permutation network of depth $O(\log^4 N)$ with bias over ordered $N/\log^2 N$-tuples in $O(1/N^2)$.*

Theorem 2 (Corollary 1.10 [CKKK01]). *There are permutation networks of depth $O(\log^2 N)$ with bias over unordered N-tuples in $O(1/N)$.*



(a) Barrel Shifter. (b) Beneš.

Fig 1. Permutation Networks

Proposition 1 guarantees the privacy of any input in the absence of information about the images of other inputs. On the other hand, the network in Theorem 1 guarantees that the images of any $k \leq N/\log^2 N$ inputs cannot be determined except with low probability. Unfortunately privacy is only guaranteed *between* ordered k-tuples, for example it is possible that the order of inputs is maintained. For applications where there is a lot of redundancy in the inputs, such as first-past-the post voting, this level of bias may be acceptable. Theorem 2 is unfortunately non-constructive, but states that efficient networks with small bias exist.

4 Obfuscation of a Paillier Shuffle

In this section we show how to obfuscate Paillier shuffles via permutation networks. A key property we use is that the Damgård-Jurik cryptosystem supports *nested* homomorphic encryption. Let $\mathcal{E}_{i,n} : \mathcal{M}_{i,n} \times \mathcal{R}_n \to \mathsf{C}_{i,n}$ denote the i^{th} generalised Paillier encryption, where $\mathcal{M}_{i,n}$ and $\mathsf{C}_{i,n}$ are the message and ciphertext spaces.[1] Similiarly define re-encryption $\mathcal{RE}_{i,n} : \mathsf{C}_{i,n} \times \mathcal{R}_n \to \mathsf{C}_{i,n}$. Then $\mathsf{C}_{i,n} \subseteq \mathcal{M}_{i+1,n}$ for all $i \geq 1$ and additionally $\mathcal{E}_{i,n}(m,r) \otimes \mathcal{E}_{j,n}(m',r') = \mathcal{E}_{j,n}(\mathcal{E}_{i,n}(m,r) \times m', \mathcal{E}_{i,n}(m,r) \otimes r')$ for all $m \in \mathsf{C}_{i,n}, m' \in \mathcal{M}_{j,n}, r, r' \in \mathcal{R}_n$. The latter property appears to have been first observed by Lipmaa [Lip05] who used it in a recursive fashion to develop an efficient 1-out-of-n computationally private information retrieval (CPIR) protocol. Adida and Wikström in [AW07a] independently noticed this fact and used it define a form of homomorphic matrix multiplication.

4.1 Matrix Notation

Let $\bar{\mathcal{E}}_{i,n}$ and $\bar{\mathcal{RE}}_{i,n}$ be encryption and re-encryption defined for matrices of inputs. Let $\circ$ denote point-wise matrix multiplication and $\otimes$ denote point-wise

[1] Note that $\mathcal{R}_n$ is actually equivalent to $\mathsf{C}_{0,n}$.

matrix exponentiation, note that like ciphertext exponentiation the exponent matrix is written on the left. Variables in lower case will generally denote vectors, while variables in upper case will denote matrices.

Definition 5 (Homomorphic Matrix Multiplication [AW07a]). *Suppose* $d = (d_l) \in \mathsf{C}_{i,n}^{1\times N}$ *and* $C = (c_{lm}) \in \mathsf{C}_{j,n}^{N\times N}$. *Then* $d \star C \triangleq \left(\prod_{l=1}^{N}(c_{lm})^{d_l}\right)_{m=1}^{N}$.

Proposition 2. *Suppose* $d \in \mathsf{C}_{i,n}^{1\times N}$ *and* $C = \bar{\mathcal{E}}_{j,n}(M, R) \in \mathsf{C}_{j,n}^{N\times N}$. *Then* $d\star C = \bar{\mathcal{E}}_{j,n}(d \times M, d \star R)$.

Proof. The proof follows easily from the homomorphic properties of $\mathcal{E}_{j,n}$.

4.2 Obfuscation of Damgård-Jurik Shuffles

The main idea behind standard Paillier obfuscation is to represent the shuffle as a *modified* permutation matrix where the ones are replaced by re-encryption factors and then encrypt it (see Definition 9 [AW07a]). A straightforward generalisation allows one to obfuscate Damgård-Jurik shuffles of arbitrary degree (Proposition 3). Moreover it turns out that homomorphic matrix multiplication actually *composes* obfuscated shuffles, although this fact was not noted in [AW07a]. The composition of two obfuscated shuffles multiplies both underlying permutations, but re-encrypts using the re-encryption factors of the first shuffle. This is formalised in Lemma 1.

Definition 6. *Suppose* $\Lambda^\pi = (\lambda_{lm}^\pi)$ *is a permutation matrix. Let* $a = (a_l) \in \mathcal{M}_{i,n}^{1\times N}$ *and* $r = (r_l) \in \mathcal{R}_n^{1\times N}$ *be vectors. Then* $P_i^\pi[a, r] \triangleq (\lambda_{lm}^\pi \mathcal{E}_{i,n}(a_l, r_l))$.

Definition 7. *The* i^{th} *generalised Paillier shuffle is a functionality* $\mathcal{PS}_i = \{\mathcal{PS}_{i,N(\kappa),\kappa}\}_{\kappa\in\mathbb{N}}$, *where* $N(\cdot)$ *is a polynomially bounded and polynomially computable function, such that for every* $\kappa \in \mathbb{N}, \mathcal{PS}_{i,N(\kappa),\kappa} = \{PS_i[\pi, r]\}_{\pi\in\Sigma_{N(\kappa)}, r\in(\{0,1\}^*)^{N(\kappa)}}$ *and for every* $(n, sk) \in \mathcal{G}(1^\kappa)$ *and* $(c_1, \ldots, c_N) \in \mathsf{C}_{i,n}^{1\times N}$ *the circuit* $PS_i[\pi, r]$ *is defined by*

$$PS_i[\pi, r](n, c_1, \ldots, c_N) = (c_1, \ldots, c_N) \times P_i^\pi[0, r].$$

Proposition 3. *Let* $PS_i[\pi, r] \in \mathcal{PS}_{i,N(\kappa),\kappa}$ *be a shuffle and let* $i < j$. *Then* $C = \bar{\mathcal{E}}_{j,n}(P_i^\pi[0, r], S) : S \in_R {\mathcal{R}_n}^{N\times N}$ *is an obfuscation of* $PS_i[\pi, r]$.

Proof. Let $d \in \mathsf{C}_{i,n}^{1\times N}$. Define $d' = d \star C$. By Proposition 2

$$d' = d \star \bar{\mathcal{E}}_{j,n}(P_i^\pi[0, r], S) = \bar{\mathcal{E}}_{j,n}(d \times P_i^\pi[0, r], d \star S) = \bar{\mathcal{E}}_{j,n}(PS_i[\pi, r](d), d \star S).$$

Lemma 1 (Composition). *Suppose* $PS_i[\mu, r] \in \mathcal{PS}_{i,N(\kappa),\kappa}$, $PS_j[\nu, r'] \in \mathcal{PS}_{j,N,\kappa}$ *are shuffles with corresponding obfuscations* $C_i^\mu \triangleq \bar{\mathcal{E}}_{i+1,n}(P_i^\mu[0, r], S)$, $C_j^\nu \triangleq \bar{\mathcal{E}}_{j+1,n}(P_j^\nu[0, r'], S')$, *where* $i < j$. *Then* (C_i^μ, C_j^ν) *is an obfuscation of* $PS_i[\nu\mu, r]$.

Proof. Let $d \in \mathsf{C}_{i,n}^{1\times N}$. Define $d' = d \star C_i^{\mu}$ and $d'' = d' \star C_j^{\nu}$. By Proposition 2

$$\begin{aligned} d'' &= \bar{\mathcal{E}}_{j+1,n}(PS_j[\nu, r'](d'), d' \star S') \\ &= \bar{\mathcal{E}}_{j+1,n}(PS_j[\nu, r'](\bar{\mathcal{E}}_{i+1,n}(PS_i[\mu, r](d), d \star S)), d' \star S') \\ &= \bar{\mathcal{E}}_{j+1,n}(\bar{\mathcal{E}}_{i+1,n}(PS_i[\mu, r](d), d \star S + r'^{(j-i)}) \times \Lambda^{\nu}, d' \star S') \\ &= \bar{\mathcal{E}}_{j+1,n}(\bar{\mathcal{E}}_{i+1,n}(PS_i[\nu\mu, r](d), \nu(d \star S + r'^{(j-i)})), d' \star S'). \end{aligned}$$

4.3 Obfuscation of Shuffle Networks

The composition lemma implies that a sequence of obfuscated shuffles of arbitrary length may be composed by multiplication, the result is an obfuscated shuffle that composes *all* permutations but inherits re-encryption factors from only the *first* shuffle in the sequence. Therefore it is possible to obfuscate the set of shuffles induced by the layers of a re-encryption permutation network [Abe99, JJ99] and compose them, provided that the i^{th} layer is lifted to accept i^{th} generalised Paillier ciphertexts (Proposition 4). We further observe that each layer can be obfuscated using only $O(N)$ ciphertexts, by decomposing the corresponding permutation into switches (Proposition 5). Combining these observations yields an efficient obfuscator of an arbitrary shuffle (Definition 8). We prove that our obfuscator is semantically secure if the network has polylogarithmic depth and the DCRA holds (Theorem 3).

Proposition 4. *Let Ψ be a re-encryption permutation network with state $\pi \overset{\Psi}{=} \pi_\Delta \dots \pi_1$. Suppose that $PS_1[\pi_1, r^{(1)}] \in \mathcal{PS}_{1,N(\kappa),\kappa}, \dots, PS_\Delta[\pi_\Delta, r^{(\Delta)}] \in \mathcal{PS}_{\Delta,N(\kappa),\kappa}$ is the sequence of shuffles corresponding to the layers of Ψ. Let $\{C_i^{\pi_i} = \bar{\mathcal{E}}_{i+1,n}(P_i^{\pi_i}[0, r^{(i)}], S_i) : S_i \in \mathcal{R}_n^{N\times N}\}$ be obfuscations. Then $(C_1^{\pi_1}, \dots, C_\Delta^{\pi_\Delta})$ is an obfuscation of $PS_1[\pi, r^{(1)}]$.*

Proof. The result follows from recursive application of Lemma 1.

Proposition 5. *Let $PS_i[\pi_i, r^{(i)}]$ and $C_i^{\pi_i}$ be defined as in Proposition 4 and let $A^{(i)}$ be the adjacency matrix of the i^{th} layer of Ψ. Then $C_i'^{\pi_i} = A^{(i)} \otimes C_i^{\pi_i}$ is also an obfuscation of $PS_i[\pi_i, r^{(i)}]$.*

Proof. Observe that by Equation (1), Definition 3, $A^{(i)} \circ \Lambda^{\pi_i} = \Lambda^{\pi_i}$. By the homomorphic properties of ciphertext exponentiation

$$\begin{aligned} A^{(i)} \otimes C_i^{\pi_i} &= A^{(i)} \otimes \bar{\mathcal{E}}_{i+1,n}(P_i^{\pi_i}[0, r^{(i)}], S_i) \\ &= \bar{\mathcal{E}}_{i+1,n}(A^{(i)} \circ P_i^{\pi_i}[0, r^{(i)}], A^{(i)} \otimes S_i) \\ &= \bar{\mathcal{E}}_{i+1,n}(P_i^{\pi_i}[0, r^{(i)}], A^{(i)} \otimes S_i). \end{aligned}$$

Note that matrix $A^{(i)}$ is zero except for the co-ordinates which correspond to input and output nodes in the i^{th} layer which are linked by switch. It follows that $C_i'^{\pi_i}$ is a matrix which consists of only $2N$ non-trivial ciphertexts.

Definition 8. *Let Ψ be a rearrangeable permutation network of depth Δ. The obfuscator $\mathcal{O}_\Psi$ for the Paillier shuffle $\mathcal{PS}_1$ takes as input the tuple $(1^\kappa, n, d, PS_1[\pi, r])$, where $(n, sk) \in \mathcal{G}(1^\kappa)$ and $PS_1[\pi, r] \in \mathcal{PS}_{1,N(\kappa),\kappa}$. It computes $\pi \stackrel{\Psi}{=} \pi_\Delta \dots \pi_1$ and generates shuffles $PS_1[\pi_1, r^{(1)}] \in \mathcal{PS}_{1,N(\kappa),\kappa}, \dots, PS_\Delta[\pi_\Delta, r^{(\Delta)}] \in \mathcal{PS}_{\Delta,N(\kappa),\kappa}$ such that $r^{(1)} = r$ and $r^{(2)}, \dots, r^{(N)} \in_R \mathcal{R}_n^{1\times N}$. It produces obfuscations $\{C_i{}^{\pi_i} = A^{(i)} \otimes \bar{\mathcal{E}}_{i+1,n}(P^{\pi_i}[0, r^{(i)}], S_i) : S_i \in \mathcal{R}_n^{N\times N}\}_{i=1}^{\Delta}$. It outputs a circuit with hardcoded $C_1{}^{\pi_1}, \dots C_\Delta{}^{\pi_\Delta}$ that, on input $d \in \mathsf{C}_{1,n}^{1\times N}$ outputs $d' = d \star C_1{}^{\pi_1} \star \dots \star C_\Delta{}^{\pi_\Delta} \in \mathsf{C}_{\Delta+1,n}$.*

Theorem 3. *The obfuscator $\mathcal{O}_\Psi$ is polynomially indistinguishable (Definition 4 [AW07a]) if Ψ has polylogarithmic depth and the DCRA holds.*

5 Distributed Sampling and Obfuscation of a Shuffle

We construct a distributed protocol for sampling and obfuscating a shuffle via an arbitrary permutation network. Suppose that mix servers $\mathcal{M}_1 - \mathcal{M}_k$ sample and obfuscate the shuffle. Denote the switch at position (i, j) in Ψ by $\chi_{i,j}$. Recall that the permutation in the i^{th} layer of a permutation network may be written as a product of the switches which are set to true, i.e. $\pi_i = \sigma_{i,1}^{b_{i,1}} \dots \sigma_{i,W}^{b_{i,W}}$. To ensure that the state of the permutation network is set uniformly at random, every mix flips the state of $\chi_{i,j}$ at random hence $b_{i,j} = 1$ with probability $1/2$. In practice $\chi_{i,j}$ is simply a permutation matrix of encrypted control bits, hiding $\sigma_{i,j}^{b_{i,j}}$. When the matrices $\chi_{i,1}, \dots, \chi_{i,W}$ are superimposed, they form the permutation matrix C_i of the shuffle π_i. Thus the obfuscated shuffle is the tuple $(C_1, \dots, C_\Delta)$.

Protocol 1 (Sampling and Obfuscation of a Shuffle).
COMMON INPUT: A Paillier public key n, integer N and permutation network Ψ of dimension $\Delta \times W$.
Mix server $\mathcal{M}_I$ proceeds as follows.

1. For $i = 1, \dots, \Delta$ do:
   (a) Generate $N \times N$ matrix C_i whose entries are all initially $\mathcal{E}_{i+1,n}(0, 0^*)$.
   (b) For $j = 1, \dots, W$ do:
      i. Generate N' double encrypted zeroes of the form $\mathcal{E}_{i+1,n}(\mathcal{E}_{i,n}(0))$ in a distributed way using Protocol 3 [AW07a]. Denote these $(c_{i,j}^{(1)}, \dots, c_{i,j}^{(N')})$.
      ii. Form the matrix:

$$\chi_{i,j}^{(0)} = \begin{pmatrix} c_{i,j}^{(1)} & \dots, & c_{i,j}^{(N')} \\ \mathcal{E}_{i+1,n}(0, 0^*) & \dots, & \mathcal{E}_{i+1,n}(0, 0^*) \end{pmatrix}.$$

      iii. For $l = 1, \dots, k$ do:

– If $l = I$, permute the rows of $\chi_{i,j}^{(l-1)}$ with $\mu_{i,j}^{(l)} \in_R \Sigma_2$ respectively and re-encrypt them with randomness $r_i^{(l)} \in_R \mathcal{R}_n^{2\times N'}$ publishing matrix

$$\chi_{i,j}^{(l)} = \bar{\mathcal{RE}}_{i+1,n}(\mu_{i,j}^{(l)}(\chi_{i,j}^{(l-1)}), r_i^{(l)}).$$

– If $l \neq I$, verify that the above equation holds.

iv. Suppose $V_{i,j} = \{l_1, \ldots, l_{N'}\}$. Update matrix C_i:

$$C_i \begin{pmatrix} l_1, l_1 & , \ldots , & l_{N'}, l_{N'} \\ l_1, \sigma_{i,j}(l_1) & , \ldots , & l_{N'}, \sigma_{i,j}(l_{N'}) \end{pmatrix} \leftarrow \chi_{i,j}^{(k)}.$$

2. Output $(C_1, \ldots, C_\Delta)$.

6 Mixnet Properties

We analyse the mixnet which result from mix servers generating an obfuscated shuffle according to Protocol 1, evaluating it upon input and requesting that a threshold number of decryption servers decrypt the output. Note that $N(\Delta+1) = O(N \log N)$ threshold decryptions are required to recover the input messages.

6.1 Privacy

We assume the existence of at least one honest mix in Protocol 1, hence the obfuscated shuffle should be identically distributed to the output of a trusted party running obfuscator $\mathcal{O}_\Psi$ (Definition 8), albeit according to a biased permutation distribution, namely that formed by setting the network uniformly at random. Therefore the security of the mixnet follows from the following theorem.

Theorem 4. *Suppose the DCRA holds. Let $(\mathcal{CS}^{pai}, S, (\mathcal{P}, \mathcal{V}))$ be the verifiable shuffle which results from a trusted party obfuscating a random Paillier shuffle according to Definition 8, evaluating it upon input and then revealing (and proving correct) each layer of intermediate decryptions. Then $(\mathcal{CS}^{pai}, S, (\mathcal{P}, \mathcal{V}))$ is IND-CPA$_S$ secure.*

We note that a weakness of the IND-CPA$_S$ model is that it does not guarantee that all information usable by an attacker remains hidden when the mixed ciphertexts are finally opened. In particular an attacker will at least know their own output and may combine this with knowledge of the bias to infer other outputs. Therefore analysis of a realisable ideal (biased) mixnet in the universally composable framework [Can01] is desirable but unfortunately beyond the scope of this paper.

Remark 1. Alternatively it is possible to construct an unbiased mixnet as follows. The I^{th} mix samples a shuffle from $\mathcal{PS}_{(I-1)\Delta+1}$ at random and obfuscates it by setting the state of the Beneš network accordingly and applying the obfuscator in Definition 8. The obfuscated shuffles are then homomorphically multiplied with the input. Note, however, this approach incurs an overhead of $k^{3.5}$ (see Section 6.2) thus is only practicable for small k.

6.2 Complexity

The expansion factor of the mix-net is $1/\Delta$. We count the effective number of multiplications modulo n for each stage of the mixing process, and compare them to [AW07a]. We assume that multiplication modulo n^s is $s^{1.5}$ times as costly as multiplication modulo n, and that exponentiation is performed by repeated squaring. This implies complexity proportional to $\Delta^{3.5}$. Note that κ_c is a parameter of Protocol 2 [AW07a] and satisfies $\kappa_c \ll \kappa$.

	Sample & Obfuscate	Prove	Evaluate	Decrypt
[AW07a] (Shuffle)	$O(N^2\kappa)$	$O((N^2 + N\kappa_c)\kappa\ )$	$O(N^2\kappa)$	$O(N \log N\kappa)$
Proposed (Shuffle & Rotate)	$O(N \log^{3.5} N\kappa)$	$O(N \log^{3.5} N\kappa_c\kappa)$	$O(N \log^{3.5} N\kappa)$	$O(N \log^{3.5} N (\log N + \kappa))$

An implementation using GMP [Gra12] suggests that our scheme is faster when $N > 1200$.

6.3 Parallelisation

Generating an obfuscated shuffle makes use of a private mixnet, therefore parallelisation is limited to that within individual mixes, of course the verification of each mix's shuffle proofs can be distributed over other mixes or the public. The evaluation of the shuffle is public, though, hence can be safely parallelised over arbitrarily many parties. The most obvious parallelisation has k processors evaluate $O(N/k)$ switches at each layer of the network, resulting in $O(\log N)$ parallel steps. Thus when $k \approx N$ it is possible to mix votes in polylogarithmic time.

7 Conclusion

We have presented a more efficient method of obfuscating a Paillier shuffle based upon re-encryption permutation networks. An interesting further direction is to investigate to what extent it is possible to distribute the sampling and obfuscation of a shuffle over a variable number of parties, without incurring a prohibitive loss in efficiency. Such a protocol could conceivably allow voters to directly contribute to the anonymisation of their votes without any assistance from third parties.

References

[Abe99] Abe, M.: Mix-Networks on Permutation Networks. In: Lam, K.-Y., Okamoto, E., Xing, C. (eds.) ASIACRYPT 1999. LNCS, vol. 1716, pp. 258–273. Springer, Heidelberg (1999)

[AH01] Abe, M., Hoshino, F.: Remarks on Mix-Network Based on Permutation Networks. In: Kim, K.-C. (ed.) PKC 2001. LNCS, vol. 1992, pp. 317–324. Springer, Heidelberg (2001)

[AW07a] Adida, B., Wikström, D.: How to Shuffle in Public. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 555–574. Springer, Heidelberg (2007)

[AW07b] Adida, B., Wikström, D.: Offline/Online mixing. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) ICALP 2007. LNCS, vol. 4596, pp. 484–495. Springer, Heidelberg (2007)

[Ben64] Beneš, V.E.: Permutation groups, complexes, and rearrangeable connecting networks. The Bell System Technical Journal 43(4), 1619–1640 (1964)

[BGN05] Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF Formulas on Ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)

[BGR98] Bellare, M., Garay, J.A., Rabin, T.: Fast Batch Verification for Modular Exponentiation and Digital Signatures. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 236–250. Springer, Heidelberg (1998)

[Can01] Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: Proceedings of the 42nd IEEE symposium on Foundations of Computer Science, FOCS 2001, pp. 136–145. IEEE Computer Society, Washington, DC (2001)

[Cha81] Chaum, D.L.: Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM 24, 84–90 (1981)

[CKKK01] Czumaj, A., Kanarek, P., Kutyłowski, M., Loryś, K.: Switching networks for generating random permutations. In: Switching Networks: Recent Advances. Network Theory and Applications, vol. 5, pp. 25–61. Kluwer Academic Publishers (2001)

[dHSŠV09] de Hoogh, S., Schoenmakers, B., Škorić, B., Villegas, J.: Verifiable Rotation of Homomorphic Encryptions. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 393–410. Springer, Heidelberg (2009)

[DJ01] Damgård, I., Jurik, M.: A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. In: Kim, K.-C. (ed.) PKC 2001. LNCS, vol. 1992, pp. 119–136. Springer, Heidelberg (2001)

[FS01] Furukawa, J., Sako, K.: An Efficient Scheme for Proving a Shuffle. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 368–387. Springer, Heidelberg (2001)

[Gjø05] Gjøsteen, K.: Homomorphic Cryptosystems Based on Subgroup Membership Problems. In: Dawson, E., Vaudenay, S. (eds.) Mycrypt 2005. LNCS, vol. 3715, pp. 314–327. Springer, Heidelberg (2005)

[Gra12] Granlund, T.: Gnu multiple precision arithmetic library (2012), http://gmplib.org/

[GZB+02] Golle, P., Zhong, S., Boneh, D., Jakobsson, M., Juels, A.: Optimistic Mixing for Exit-Polls. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 451–465. Springer, Heidelberg (2002)

[JJ99] Jakobsson, M., Juels, A.: Millimix: Mixing in small batches. Technical report, Center for Discrete Mathematics and Theoretical Computer Science, DIMACS (1999)

[Lip05] Lipmaa, H.: An Oblivious Transfer Protocol with Log-Squared Communication. In: Zhou, J., López, J., Deng, R.H., Bao, F. (eds.) ISC 2005. LNCS, vol. 3650, pp. 314–328. Springer, Heidelberg (2005)
[Nef01] Andrew Neff, C.: A verifiable secret shuffle and its application to e-voting. In: Proceedings of the 8th ACM Conference on Computer and Communications Security, CCS 2001, pp. 116–125. ACM (2001)
[NSNK04] Nguyen, L., Safavi-Naini, R., Kurosawa, K.: Verifiable Shuffles: A Formal Model and a Paillier-Based Efficient Construction with Provable Security. In: Jakobsson, M., Yung, M., Zhou, J. (eds.) ACNS 2004. LNCS, vol. 3089, pp. 61–75. Springer, Heidelberg (2004)
[OS07] Ostrovsky, R., Skeith, W.E.: Private searching on streaming data. Journal of Cryptology 20, 397–430 (2007)
[Pai99] Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
[PIK94] Park, C.-S., Itoh, K., Kurosawa, K.: Efficient Anonymous Channel and All/Nothing Election Scheme. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 248–259. Springer, Heidelberg (1994)
[RW04] Reiter, M.K., Wang, X.: Fragile mixing. In: Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004, pp. 227–235. ACM, New York (2004)
[SK95] Sako, K., Kilian, J.: Receipt-Free Mix-Type Voting Scheme: A Practical Solution to the Implementation of a Voting Booth. In: Guillou, L.C., Quisquater, J.-J. (eds.) EUROCRYPT 1995. LNCS, vol. 921, pp. 393–403. Springer, Heidelberg (1995)
[Wak68] Waksman, A.: A permutation network. J. ACM 15, 159–163 (1968)
[Wik09] Wikström, D.: A Commitment-Consistent Proof of a Shuffle. In: Boyd, C., González Nieto, J. (eds.) ACISP 2009. LNCS, vol. 5594, pp. 407–421. Springer, Heidelberg (2009)

A Shuffle Privacy

A *verifiable shuffle* is a tuple $(\mathcal{RP}, S, (\mathcal{P}, \mathcal{V}))$ where $\mathcal{RP}$ is a public-key cryptosystem with a re-encryption algorithm, S is a PPT algorithm that shuffles input ciphertexts and $(\mathcal{P}, \mathcal{V})$ is a proof system that proves the existence of re-encryption factors linking input and output ciphertexts [NSNK04]. One definition for security of a verifiable shuffle is *indistinguishability under chosen permutation attacks* (IND-CPA$_S$) and is an extension of classical IND-CPA security for cryptosystems. A related definition is *semantic privacy under chosen permutation attacks* (SP-CPA$_S$) which specifies that whatever can be computed after the shuffle execution could be computed using only prior information. Nguyen et al. [NSNK04] prove that the two notions are equivalent.

Definition 9 (Indistinguishability under Chosen Permutation Attacks). *Let* $(\mathcal{RP}, S, (\mathcal{P}, \mathcal{V}))$ *be a verifiable shuffle and* $A = (A_1, A_2)$ *be a pair of PPT algorithms. Let* $t \in \{0,1\}^{poly(\kappa)}$. *Define*

Experiment $Exp^{IND-CPA_S-b}_{A,(\mathcal{RP},S,(\mathcal{P},\mathcal{V}))}(\kappa,t)$:

$$(pk,sk) \leftarrow \mathcal{G}(1^\kappa); ((\pi_{(0)},\pi_{(1)},L_{in},L_{in}^{(p)},C_{E_{pk}}^{L_{in}^{(p)}}),\delta) \leftarrow A_1(pk,t);$$
$$L_{out} \leftarrow S(pk,L_{in},\pi_{(b)});$$
$$o_{(b)} \leftarrow (L_{out}, VIEW_{\mathcal{V}}^{\mathcal{P}}(pk,L_{in},L_{out}),L_{in},L_{in}^{(p)},C_{E_{pk}}^{L_{in}^{(p)}});$$
$$v \leftarrow A_2(\delta,o_{(b)})$$
return v

and let

$$Adv(A) = \max_{t\in\{0,1\}^{poly(\kappa)}} |\Pr[Exp^{IND-CPA_S-1}_{A,(\mathcal{RP},S,(\mathcal{P},\mathcal{V}))}(\kappa,t)=1] - \Pr[Exp^{IND-CPA_S-0}_{A,(\mathcal{RP},S,(\mathcal{P},\mathcal{V}))}(\kappa,t)=1]|$$

Then $(\mathcal{RP},S,(\mathcal{P},\mathcal{V}))$ satisfies indistinguishability under chosen plaintext attacks (IND-CPA$_S$) if for any A with polynomially bounded auxiliary input, $Adv(A)$ is negligible.

B Proofs

B.1 Proof of Theorem 3

Proof. Suppose there is an adversary A against the the obfuscator $\mathcal{O}_\Psi$ with advantage $\epsilon(\kappa)$. Let A' be an adversary in the IND-CPA experiment for $\mathcal{E}_{\Delta+1,n}$. When A outputs challenge circuits PS^0, PS^1, adversary A' generates sequences $\{P_i^0\}_{i=1}^{\Delta}$, $\{P_i^1\}_{i=1}^{\Delta}$ as would be generated by $\mathcal{O}_\Psi$. When the IND-CPA experiment returns $S^b = \{\bar{\mathcal{E}}_{\Delta+1,n}(P_i^b)\}_{i=1}^{\Delta}$, A' produces $S^b_{red} = \{A^{(i)} \otimes \bar{\mathcal{E}}_{\Delta+1,n}(P_i^b) \pmod{n^{i+1}}\}_{i=1}^{\Delta}$ and passes it to A, outputting 1 iff A does. Since S^b_{red} is identically distributed to $\mathcal{O}_\Psi(PS^b)$, the advantage of A' in the IND-CPA experiment is identical to that of A in distinguishing the obfuscated shuffles. Then by the remarks in Section 2.3, A' has advantage at least $\epsilon(\kappa)/(\Delta+1) = \epsilon(\kappa)/O(\log^c N)$ in breaking the DCRA. Since $N < 2^\kappa$, $\log^c N < \kappa^c$ thus $\epsilon(\kappa)/\kappa^c$ must be negligible if the DCRA holds. Then $\epsilon(\kappa)$ is also negligible and polynomial indistinguishability of $\mathcal{O}_\Psi$ follows.

B.2 Proof of Theorem 4

Proof. The proof is via a hybrid argument. First define $\bar{\mathcal{D}}_{i,n}$ to be the vector form of $\mathcal{D}_{i,n}$, and define $\bar{\mathcal{D}}_{j:i,n} = \bar{\mathcal{D}}_{j,n} \circ \ldots \circ \bar{\mathcal{D}}_{i,n}$ for $j > i$. Suppose $\Pi_{(b)}$ is the distribution of the challenge $o_{(b)}$ in $Exp^{IND-CPA_S-b}_{A,\ (\mathcal{CS}^{pai},S,(\mathcal{P},\mathcal{V}))}$, where $A = (A_1, A_2)$ is

an adversary. Define the following hybrid distributions:

$$\begin{aligned}
\Pi_{(b)} = \Bigg(& L_{in}^{(p)}, C_{\mathcal{E}_{1,n}}^{L_{in}^{(p)}}, L_{in}, L_{eval}, \bar{\mathcal{D}}_{\Delta+1,n}(L_{eval}), \ldots, \bar{\mathcal{D}}_{\Delta+1:2,n}(L_{eval}) \Bigg) : \\
& (L_{in}, L_{in}^{(p)}, C_{\mathcal{E}_{1,n}}^{L_{in}^{(p)}}, \pi_{(0)}, \pi_{(1)}) \leftarrow A_1(n,t), \pi_{(b)} \stackrel{\Psi}{=} \pi_{\Delta} \ldots \pi_1, \\
& C_i = A^{(i)} \otimes \bar{\mathcal{E}}_{i+1,n}(P^{\pi_i}[0, r^{(i)}], S_i) : r^{(i)} \in \mathcal{R}_n^{1\times N}, S_i \in \mathcal{R}_n^{N\times N}, \\
& L_{eval} \leftarrow L_{in} \star \prod_{i=1}^{\Delta} C_i. \\
\hat{\Pi}_{(b)} = \Bigg(& L_{in}^{(p)}, C_{\mathcal{E}_{1,n}}^{L_{in}^{(p)}}, L_{in}, L_{eval}, \bar{\mathcal{D}}_{\Delta+1,n}(L_{eval}), \ldots, \bar{\mathcal{D}}_{\Delta+1:2,n}(L_{eval}) \Bigg) : \\
& (L_{in}, L_{in}^{(p)}, C_{\mathcal{E}_{1,n}}^{L_{in}^{(p)}}, \pi_{(0)}, \pi_{(1)}) \leftarrow A_1(n,t), \pi_{(b)} \stackrel{\Psi}{=} \pi_{\Delta} \ldots \pi_1, \\
& C_i = A^{(i)} \otimes \bar{\mathcal{E}}_{i+1,n}(P^{\pi_i}[x^{(i)}, r^{(i)}], S_i) : x^{(i)} \in \mathcal{M}_{i,n}^{1\times N}, r^{(i)} \in \mathcal{R}_n^{1\times N}, S_i \in \mathcal{R}_n^{N\times N}, \\
& L_{eval} \leftarrow L_{in} \star \prod_{i=1}^{\Delta} C_i. \\
\tilde{\Pi}_{\Psi} = \Bigg(& L_{in}^{(p)}, C_{\mathcal{E}_{1,n}}^{L_{in}^{(p)}}, L_{in}, L_{eval}, y_{\Delta}, \ldots, y_1 \Bigg) : \\
& (L_{in}, L_{in}^{(p)}, C_{\mathcal{E}_{1,n}}^{L_{in}^{(p)}}, \pi_{(0)}, \pi_{(1)}) \leftarrow A_1(n,t), \\
& C_i = A^{(i)} \otimes \bar{\mathcal{E}}_{i+1,n}(\mathbf{0}, S_i) : S_i \in \mathcal{R}_n^{N\times N}, \\
& L_{eval} \leftarrow L_{in} \star \prod_{i=1}^{\Delta} C_i, y_{\Delta} \in_R \mathsf{C}_{\Delta,n}^{1\times N}, \ldots, y_1 \in_R \mathsf{C}_{1,n}^{1\times N}.
\end{aligned}$$

We are required to show that the distributions $\Pi_{(0)}$ and $\Pi_{(1)}$ are computationally indistinguishable. By transitivity it suffices to prove that $\Pi_{(b)}$ and $\tilde{\Pi}_{(b)}$ are computationally indistinguishable for each b. However this in turn follows from combining Lemmas 2 and 3.

Lemma 2. *Suppose the DCRA holds. Then the distributions $\Pi_{(b)}$ and $\hat{\Pi}_{(b)}$ are computationally indistinguishable.*

Proof. Suppose there is an adversary $A = (A_1, A_2)$ against $\Pi_{(b)}$ and $\hat{\Pi}_{(b)}$ with advantage $\epsilon(\kappa)$. Let A' be the adversary that distinguishes a ciphertext c_{Δ} that is a random encryption of 0 or a uniform message under $\mathcal{E}_{\Delta,n}$ as follows.

Adversary $A'(c_{\Delta}, n, t)$

1. Set $(L_{in}, L_{in}^{(p)}, C_{\mathcal{E}_{1,n}}^{L_{in}^{(p)}}, \pi_{(0)}, \pi_{(1)}) \leftarrow A_1(n,t)$
2. Compute $\pi_{(b)} \stackrel{\Psi}{=} \pi_{\Delta} \ldots \pi_1$.
3. For $i = 1, \ldots, \Delta$ do:
   (a) Compute $c_i \equiv c_{\Delta} \bmod n^{i+1}$.

(b) Apply Protocol 2 on (c_i, π_i) to generate modified permutation matrix $P_i^{\pi_i}[a^{(i)}, r^{(i)}]$.
(c) Generate $C_i = A^{(i)} \otimes \bar{\mathcal{E}}_{i+1,n}(P_i^{\pi_i}[a^{(i)}, r^{(i)}], S_i) : S_i \in_R \mathcal{R}_n^{N \times N}$.

4. Set $L_{eval} \leftarrow L_{in} \star \prod_{i=1}^{\Delta} C_i$.
5. For $i = 1, \ldots, \Delta$ do:
   (a) Compute $\bar{\mathcal{D}}_{\Delta+1:i+1,n}(L_{eval}) = \pi_\Delta \ldots \pi_{i+1}\ (l_{i-1} \times P_i^{\pi_i}[a^{(i)}, r^{(i)}])$.
6. Run A_2 on

$$o_{(b)} = (L_{in}^{(p)}, C_{\mathcal{E}_{1,n}}^{L_{in}^{(p)}}, L_{in}, L_{eval}, \bar{\mathcal{D}}_{\Delta+1,n}(L_{eval}), \ldots, \bar{\mathcal{D}}_{\Delta+1:2,n}(L_{eval}))$$

and output 1 iff A_2 does.

Clearly A' has advantage $\epsilon(\kappa)$ in breaking the semantic security of $\mathcal{E}_{\Delta,n}$. Since $\Delta = O(\log^c N)$ the DCRA implies $\epsilon(\kappa)$ is negligible, hence the lemma follows.

Lemma 3. *Suppose the DCRA holds. Then the distributions $\hat{\Pi}_{(b)}$ and $\tilde{\Pi}$ are computationally indistinguishable.*

Proof. Suppose there is an adversary $A = (A_1, A_2)$ against $\hat{\Pi}_{(b)}$ and $\tilde{\Pi}$ with advantage $\epsilon(\kappa)$. Let A' be the adversary that distinguishes a ciphertext $c_{\Delta+1}$ that is a random encryption of 0 or 1 under $\mathcal{E}_{\Delta+1}$ as follows.

Adversary $A'(c_{\Delta+1}, n, t)$

1. Set $(L_{in}, L_{in}^{(p)}, C_{\mathcal{E}_{1,n}}^{L_{in}^{(p)}}, \pi_{(0)}, \pi_{(1)}) \leftarrow A_1(n, t)$.
2. Compute $\pi_{(b)} \stackrel{\Psi}{=} \pi_\Delta \ldots \pi_1$.
3. For $i = 1, \ldots, \Delta$ do:
   (a) Compute $c_i \equiv c_{\Delta+1} \bmod n^{i+2}$.
   (b) Generate modified permutation matrix $P_i^{\pi_i}[x^{(i)}, r^{(i)}]$ where $x^{(i)} \in_R {\mathcal{M}_{i,n}}^{1 \times N}, r^{(i)} \in_R \mathcal{R}_n^{1 \times N}$.
   (c) Apply Protocol 3 on (c_i, π_i) to generate matrix of ciphertexts M_i.
   (d) Generate $C_i = A^{(i)} \otimes (P_i^{\pi_i}[x^{(i)}, r^{(i)}] \otimes M_i \oplus \mathcal{E}_{i,n}(\mathbf{0}, S_i)) : S_i \in_R \mathcal{R}_n^{N \times N}$.
4. Set $L_{eval} \leftarrow L_{in} \star \prod_{i=1}^{\Delta} C_i$.
5. For $i = 1, \ldots, \Delta$ do:
   (a) Compute $\bar{\mathcal{D}}_{\Delta+1:i+1,n}(L_{eval}) = \pi_\Delta \ldots \pi_{i+1}\ (l_{i-1} \times P_i^{\pi_i}[x^{(i)}, r^{(i)}])$.
6. Run A_2 on

$$o_{(b)} = (L_{in}^{(p)}, C_{\mathcal{E}_{1,n}}^{L_{in}^{(p)}}, L_{in}, L_{eval}, \bar{\mathcal{D}}_{\Delta+1,n}(L_{eval}), \ldots, \bar{\mathcal{D}}_{\Delta+1:2,n}(L_{eval}))$$

and output 1 iff A_2 does.

Clearly A' has advantage $\epsilon(\kappa)$ in breaking the semantic security of $\mathcal{E}_{\Delta+1,n}$. Since $\Delta = O(\log^c N)$ the DCRA implies $\epsilon(\kappa)$ is negligible, hence the lemma follows.

Protocol 2.

Input Ciphertext c which is a random encryption of 0 or a uniformly chosen message under $\mathcal{E}_i$ and permutation $\pi_i \in \Sigma_N$.

Output Modified permutation matrix $P_i^{\pi_i}[a, r]$ with distribution

$$P_i^{\pi_i}[0, r] : r \in_R \mathcal{R}_n^{1\times N} \text{ if } c \text{ is a random encryption of 0,}$$
$$P_i^{\pi_i}[x, r] : x \in_R {\mathcal{M}_{i,n}}^{1\times N}, r \in_R \mathcal{R}_n^{1\times N} \text{ otherwise.}$$

Procedure Use standard amplification to generate N independent copies $c_1, \ldots, c_N$ which have the same distribution as c. Replacing the ones in Λ^{π_i} with $\{c_i\}_{i=1}^N$ yields the required distribution.

Protocol 3.

Input Ciphertext c which is a random encryption of 0 or 1 under $\mathcal{E}_{i+1,n}$ and a permutation $\pi_i \in \Sigma_N$.

Output Matrix M_i with distribution

$$\bar{\mathcal{E}}_{i+1,n}(\mathbf{0}, S) : S \in_R \mathcal{R}_n^{N\times N} \text{ if } c \text{ is a random encryption of 0,}$$
$$\bar{\mathcal{E}}_{i+1,n}(\Lambda^{\pi_i}, S) : S \in_R \mathcal{R}_n^{N\times N} \text{ otherwise.}$$

Procedure Use standard amplification to generate N independent copies $c_1, \ldots, c_N$ which have the same distribution as c. Replacing the ones in Λ^{π_i} with $\{c_i\}_{i=1}^N$ and the zeroes with random encryptions of zero yields the required distribution.

Efficient Password Authenticated Key Exchange via Oblivious Transfer

Ran Canetti[1,⋆], Dana Dachman-Soled[2], Vinod Vaikuntanathan[3], and Hoeteck Wee[4,⋆⋆]

[1] Tel Aviv University & Boston University
[2] Microsoft Research New England
[3] University of Toronto
[4] George Washington University

Abstract. We present a new framework for constructing efficient password authenticated key exchange (PAKE) protocols based on oblivious transfer (OT). Using this framework, we obtain:

- an efficient and simple UC-secure PAKE protocol that is secure against adaptive corruptions *without erasures*.
- efficient and simple PAKE protocols under the Computational Diffie-Hellman (CDH) assumption and the hardness of factoring. (Previous efficient constructions rely on hash proof systems, which appears to be inherently limited to decisional assumptions.)

All of our constructions assume a common reference string (CRS) but do not rely on random oracles.

Keywords: Password Authenticated Key Exchange, UC security, adaptive security, oblivious transfer, search assumptions.

1 Introduction

Password authenticated key-exchange (PAKE) allows two parties with a shared password to mutually authenticate each other and establish a shared key, without explicitly revealing the password in the process [BM93]. PAKE is well-suited for use in web authentication (in place of having the user input her password directly), as it resists phishing and other social engineering attacks; if a user mistakenly authenticates herself to a phisher via a PAKE protocol, the protocol will fail, but the user's password remains safe. For this application, it is important that the PAKE protocol remains secure even amidst concurrent executions, as is unavoidable on the Internet.

⋆ Supported by the Check Point Institute for Information Security, Marie Curie grant PIRG03-GA-2008-230640, and ISF grant 0603805843.
⋆⋆ Supported by NSF CAREER Award CNS-0953626.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 449–466, 2012.

Prior work. The study of PAKE was initiated by Bellovin and Merritt [BM93]. Formal models for PAKE were developed several years later in [BPR00, BMP00, GL01, CHK+05], and solutions were first presented in the random oracle/ideal cipher models [BPR00, BMP00, MPS00]. Since then, there has been a large number of constructions in the standard model, without relying on random oracles. For instance, we now know how to achieve security in the "plain model" without any additional trusted set-up [GL01, NV04, BCL+05, GJO10]; these constructions typically rely on general techniques for secure computation. However, these protocols are fairly inefficient in terms of communication, computation and round complexity and seem unlikely to lead to a practical instantiation.

In this work, we focus on efficient constructions in the common reference string (CRS) model, initiated by Katz, Ostrovsky and Yung [KOY01] and revisited in [GL03, JG04, CHK+05, KMTG05, G08, ACP09, KV09, GK10, KV11]. Note that in practice, the CRS can be hard-coded into an implementation of the protocol. In addition to being computationally efficient and constant-round, these protocols remain secure even with adversarially coordinated concurrent executions. All of these works rely on the paradigm of smooth projective hashing [CS02, CS98] (either directly or indirectly). The most general and most recent is that of Groce and Katz [GK10], which building on [JG04], shows how to realize efficient PAKE with two building blocks: a CPA-secure encryption scheme supporting projective hashing, and a CCA-secure encryption scheme. This improves over previous works which require a CCA-secure scheme that supports smooth projective hashing.

The reliance on smooth projective hashing leads to two limitations on the ensuing protocols: first, all known instantiations of smooth projective hashing rely on decisional assumptions. e.g., the Decisional Diffie-Hellman (DDH) assumption or the quadratic residuosity assumption. In general, decisional assumptions are a much stronger class of assumptions than computational assumptions based on search problems, such as factoring, finding shortest vectors in lattices, or even the Computational Diffie-Hellman (CDH) problem. Indeed, there are groups, such as certain elliptic curve groups with bilinear pairing map, where the DDH assumption does not hold, but the Computational Diffie-Hellman (CDH) problem appears to be hard. As such, schemes based on search problems are generally preferred to those based on decisional assumptions. However, such schemes seem very hard to obtain.

Second, modifying the schemes based on smooth projective hashing to achieve security against adaptive corruptions (where an adversary may choose which parties to corrupt during the execution of the protocol) appears to be fairly challenging. This was first achieved in the recent work of Abdalla et al. [ACP09], under the additional assumption of secure erasures.

1.1 Our Contributions

We present the first construction of reasonably efficient PAKE protocols that bypass the "projective hashing" paradigm. Instead, we rely on oblivious transfer (OT) as the main cryptographic building block. We obtain new PAKE protocols

that achieve various combinations of the following properties: (a) conceptual simplicity, (b) efficiency, (c) security against adaptive corruptions even without erasures, and (d) reliance on relatively weak hardness assumptions.

Before we outline our result, we first mention that there are two prevailing security notions for PAKE that achieve security under concurrent executions and in particular, guarantee resilience against man-in-the-middle attacks. The first and most basic notion is that of "concurent PAKE" put forth by Bellare et al. and Boyko et al. [BPR00, BMP00]. The second and stronger notion is that of "UC secure PAKE" [CHK+05, C01], which guarantee security amidst composition with arbitrary protocols, and with arbitrary, unknown and possibly correlated password distributions.

Our results. Specifically, we show:

- Two UC-secure PAKE protocols. The first only assumes an ideal OT functionality, and is secure against adaptive corruptions without erasures. Combined with the OT protocol with Garay et al. [GWZ09], we obtain a reasonably efficient UC-secure PAKE protocol in the CRS model that is secure against adaptive corruptions without erasures. (Prior protocols that achieve adaptive security are either in the Random Oracle model [ACCP08], or require secure erasures [ACP09] or are highly inefficient [BCL+05].)
  The second protocol builds on [GK10], is a more efficient variant of the first, and relies on a CCA2-secure PKE in addition to OT. It only tolerates static corruptions. We defer the details of this construction to the full version.
- New PAKE protocols under search assumptions, notably CDH and hardness of factoring. Previous efficient instantiations rely on hash proof systems, which appears to be inherently limited to decisional assumptions. This construction requires a special variant of OT. Here we also provide some constructions of this special OT variant.

1.2 Overview of Our Constructions

We proceed to provide an overview of our constructions.

The UC Constructions. The main novelty in our UC constructions are protocols that assume ideal authenticated channels as well as ideal "OT channels" and realize the following two party functionality, which we call randomized equality ($\mathcal{F}_{\mathsf{re}}$): If the inputs provided by the parties are equal, then both parties obtain the same fresh random key. If the parties provide different inputs, then each party obtains a special symbol $\perp$.

Given such a protocol, we use a generic transformation from [BCL+05] to obtain a protocol that realizes the "split version" of $\mathcal{F}_{\mathsf{re}}$, which turns out to be equivalent to $\mathcal{F}_{\mathsf{pwKE}}$, the ideal password-based key exchange functionality. The above transformation results in an additional cost of generating a key for a signature scheme, and then signing each message. Alternatively, we may rely on a more efficient transformation described in [CCGS10], that costs only a single key exchange protocol, plus a MAC creation and verification per message (although this transformation only achieves adaptive security *with erasures*).

First Construction. Our first protocol for realizing $\mathcal{F}_{\mathsf{re}}$ is extremely simple. Assume for now that we have an ideal 1-out-of-$|\mathcal{D}|$ OT functionality. The first party acts as the OT receiver and uses as input his password. The second party acts as the OT sender and picks $|\mathcal{D}|$ random strings $r_1, \ldots, r_{|\mathcal{D}|}$ as input. The first party uses the OT output as his session key, and the second party uses the string indexed by his password. Indeed, if both parties are honest, they agree on the same random key, and if the first party is corrupted, he learns nothing about the session key unless he guesses the right password. There are two issues with the protocol as described:

- The protocol only handles dictionaries of polynomial size. To fix this, we observe that we only require that the $|\mathcal{D}|$ random strings be pairwise independent. In particular, we can replace the 1-out-of-$|\mathcal{D}|$ OT functionality with $\log |\mathcal{D}|$ copies of 1-out-of-2 OT, where the second party now picks $\log |\mathcal{D}|$ pairs of random string, and the first party outputs the XOR of the $\log |\mathcal{D}|$ OT outputs. (In the overview of the remaining constructions, we omit this optimization for simplicity.)
- The protocol does not tolerate corruptions of the second party; for instance, the second party could set all $|\mathcal{D}|$ strings to be equal thereby learning the session key. To fix this, we repeat the basic protocol one more time, with the roles of the parties reversed, and the final session key is the XOR of the two session keys. By running the basic protocol in reverse, we guarantee that the second party also learns nothing about the session key unless it guesses the right password. (This idea of running a basic protocol with reversed roles appears in the early works of Katz et al. [KOY01, GL03] too.)

Combining this construction with the adaptively secure OT given in [GWZ09], we obtain the following result:

> **Proposition 1 (informal).** *There exists a constant-round UC-secure PAKE protocol in the CRS model that is secure against an adaptive adversary without erasures and without authenticated channels. The protocol may be based on DDH or DCR and both parties exchange a constant number of group elements.*

Second Construction. To motivate our second construction, which is inspired by that of Groce and Katz [GK10, JG04], consider again our basic protocol based on an ideal 1-out-of-$|\mathcal{D}|$ OT functionality. Instead of running the basic protocol a second time in order to handle corruptions of the second party, we have the second party send an encryption of her password. The advantage over the first protocol is that the computation costs for a CCA2-secure encryption is typically lower than that of running another OT protocol. In more detail, we assume in addition a common reference string (CRS), and handle corruptions of the second party as follows:

- Both parties run the basic protocol. Let $r_1, \ldots, r_{|\mathcal{D}|}$ denote $|\mathcal{D}|$ random strings chosen by the second party, and let π denote her password. She then

parses r_π as a pair of random strings $\mathsf{skey}\|\mathsf{rand}$, sends along an encryption C of π (and her identifier) using randomness rand, under a public key for a CCA2-secure encryption scheme that is part of the CRS. The first party encrypts her password with randomness determined by the output from the basic protocol. If the ciphertext matches C, both parties output skey as the session key.

If the first party is corrupted and fails to guess the right password, then both skey and rand are truly random from her point of view, and the ciphertext C reveals no information about the second party's password via semantic security. On the other hand, if the second party is corrupted and fails to guess the right password, then C will not match the first party's password by (perfect) correctness of the underlying encryption. In the proof of security, the simulator will decrypt C to extract the password of the second party.

The Concurrently-Secure PAKE. Our concurrently-secure PAKE is essentially the same as our second UC-secure construction, except we replace the underlying UC-secure OT with an OT protocol that achieves much weaker guarantees. Roughly speaking, we relax the security guarantee for corrupted senders to an indistinguishability-based notion, and moreover, we no longer require that the OT guarantee non-malleability. The resulting construction may also be viewed as an abstraction of the Groce-Katz protocol [GK10, JG04], where we use an OT primitive in lieu of the CPA-secure encryption with projective hashing. We provide two different approaches towards realizing the underlying OT primitive.

Concurrent PAKE from Lossiness. Our first approach is based on dual-mode cryptosystems, a "lossy" primitive introduced by Peikert et al. [PVW08]. Combined with our general framework, we obtain the following result:

Proposition 2 (informal). *There exists a three-message PAKE protocol in the CRS model that relies on black-box access to a dual-mode cryptosystem and a CCA-secure encryption scheme and achieves concurent security with mutual authentication (against a static adversary).*

We stress that this construction should be viewed mainly as a feasibility result on black-box constructions of PAKE protocols in the CRS model based on general assumptions. The work of Peikert and Waters [PW08] introduced the notion of lossy trapdoor functions, and showed that they also yield CCA-secure encryption schemes. This raised the natural question of understanding connections between smooth projective hashing and "lossy" primitives. Our work demonstrates that for concurrently-secure PAKE protocol, it is indeed possible to avoid the use of smooth projective hashing and rely solely on "lossy" primitives (notably the dual mode encryption scheme in [PVW08] and lossy trapdoor functions) in a black-box way.

Concurrent PAKE from Search Assumptions. Our second approach starts with the Bellare-Micali OT protocol based on CDH. Combined with our general framework, we obtain the following result:

> **Proposition 3 (informal).** *There exists a constant-round PAKE protocol in the CRS model based on hardness of factoring or CDH (computational Diffie-Hellman assumption) that achieves concurrent security with mutual authentication (against a static adversary). Moreover, each party sends a quadratic number of group elements.*

Password Based Group Key Exchange. The second UC construction and the concurrently secure construction have the following additional attractive property: The generated session key is determined exclusively by one of the parties. Furthermore, this key can be chosen by this party in advance, before the protocol begins. This property allows for a natural extension of these PAKE protocols to efficient password based group key exchange protocols: One party exchanges a key with each one of the other parties, using the above property to ensure that all parties agree on the same key.

This approach to group key exchange is indeed different than the approach in prior works on this problem, e.g. [ABCP06, AP06], which concentrate on "contributory protocols" where all parties "contribute" to the group key. Still, it arguably provides an adequate level of security. This approach is particularly suitable to groups where there is one special party (either the group manager or the multi-caster of the data): here this party is the only one that does work that's proportional to the size of the group. The work done by all other parties is independent of the size of the group.

2 UC-Secure PAKE from Oblivious Transfer

We present a UC-secure PAKE protocol from Oblivious Transfer. An alternative construction appears in the full version.

Definitions. For simplicity and clarity, we begin by realizing single-session PAKE, and we extend all of these definitions and results to multi-sessions in the full version[1]. We present the functionality $\mathcal{F}_{\mathsf{pwKE}}$ for password-based key exchange. The description of the functionality is a modified version of the description in [GK10] (which is itself a modification of [CHK+05]). In particular, $\mathcal{F}_{\mathsf{pwKE}}$ captures PAKE protocols which achieve *explicit mutual authentication*. We refer the reader to [CHK+05, GK10] for motivating discussion regarding the particular choices made in this formulation of the functionality.

[1] Note that for single-session PAKE we may require an independent common reference string for each concurrent PAKE session; however, realizing multi-session PAKE allows us to have a single global common reference string for an unbounded number of concurrent PAKE sessions.

Functionality $\mathcal{F}_{\mathsf{pwKE}}$

The functionality $\mathcal{F}_{\mathsf{pwKE}}$ is parameterized by a security parameter λ. It interacts with an adversary $\mathcal{S}$ and a set of parties via the following queries:

Upon receiving a query $(\mathsf{NewSession}, \mathsf{sid}, I, R\pi_I)$ **from party** I**:** Record (I, R, π_I), mark this record fresh, and send a message (sid, I, R) to $\mathcal{S}$. Ignore all future messages from I with the same ssid.

Upon receiving a query $(\mathsf{sid}, \mathsf{ok})$ **from** $\mathcal{S}$**:** Send a message $(\mathsf{NewSession}, \mathsf{sid}, I, R)$ to R. Ignore all future (ok) messages.

Upon receiving a query $(\mathsf{Respond}, \mathsf{sid}, I, R, \pi_R)$ **from** R**:** Record (R, I, π) and mark this record fresh.

Upon receiving a query $(\mathsf{TestPwd}, \mathsf{sid}, P, \pi')$ **from the adversary** $\mathcal{S}$**:** If $P \in \{I, R\}$, there is a record of the form $(P, *, \pi)$ which is fresh, then do: If $\pi' = \pi$, mark the record compromised and reply to $\mathcal{S}$ with "correct guess". If $\pi \neq \pi'$, mark the record interrupted and reply to $\mathcal{S}$ with "wrong guess". fresh,

Upon receiving a query $(\mathsf{NewKey}, \mathsf{sid}, P, \mathsf{skey})$ **from** $\mathcal{S}$**, where** $|\mathsf{skey}| = \lambda$**:**

If there is a record of the form $(P, *, \pi)$ that is not marked completed, do:

- If this record is compromised, or either I or R is corrupted, then output $(\mathsf{sid}, \mathsf{skey})$ to player P.
- else, if there is a record $(*, P, \pi', \mathsf{server}, \mathsf{skey}')$ with $\pi' = \pi$, then send skey' to player P.

Fig. 1. The password-based key-exchange functionality $\mathcal{F}_{\mathsf{pwKE}}$

Constructions. The construction of both protocols proceeds in three steps. First, in Section 2.1, we define a (randomized) equality-testing functionality $\mathcal{F}_{\mathsf{re}}$ which, informally speaking, captures PAKE in the authenticated channels model. In Section 2.2, we show a protocol that securely implements $\mathcal{F}_{\mathsf{re}}$ in the OT-hybrid model, tolerating adaptive corruptions (a second protocol that implements $\mathcal{F}_{\mathsf{re}}$ is presented in the full version). These protocols assume built-in authenticated channels whereas our end goal, of course, is to implement PAKE without any authenticated channels. Thus, our second step is to transform these protocols into ones that do not assume authenticated channels, but implement a "split version" of $\mathcal{F}_{\mathsf{re}}$ (See Section 2.3 for more details) using the transformation of Barak, Canetti, Lindell, Pass and Rabin [BCL+05]. Together with the adaptively secure OT protocol of Garay, Wichs and Zhou [GWZ09], this gives us a protocol implementing the split $\mathcal{F}_{\mathsf{re}}$ functionality in the common reference string model, tolerating adaptive corruptions. Finally, we show (in Proposition 4) that the split $\mathcal{F}_{\mathsf{re}}$ functionality already captures UC-secure PAKE. We note that this three step method of constructing UC PAKE protocols was pointed out in the work of Barak et al. [BCL+05].

2.1 The Randomized Equality-Testing Functionality

We define a (randomized) equality-testing functionality $\mathcal{F}_{\mathsf{re}}$ that, roughly speaking, takes inputs from two parties and does the following:

– if the inputs are equal, sends both parties the same random session key; moreover, if either party is corrupted, the adversary is allowed to set the key.
– if the inputs are unequal, send both parties the special symbol $\perp$.

More precisely, $\mathcal{F}_{\mathsf{re}}$ captures a protocol between two players – an "initiator" I and a "responder" R. The initiator starts the protocol by sending a message to the functionality $\mathcal{F}_{\mathsf{re}}$ that includes his input π_I. The functionality then allows the adversary $\mathcal{S}$ to determine when to "wake up" the responder R into starting the protocol. Once woken up, R sends his input x_R to the functionality. If the inputs match, then the functionality assigns the same random key to both parties. Otherwise, it assigns a special symbol $\perp$ to both of them. Thus, this definition corresponds to achieving *explicit mutual authentication*. We allow the ideal-model adversary two special powers. First, we allow him to set the shared key if one of the parties is corrupted and both the parties have the same input (jumping ahead, we note that this corresponds to his ability to set the key in case he guessed one of the parties' password correctly). Furthermore, he controls the delivery of messages to the parties. This is an ability that he inevitably has in the real world.

Functionality $\mathcal{F}_{\mathsf{re}}$

The functionality $\mathcal{F}_{\mathsf{re}}$ is parameterized by a security parameter λ and a "dictionary" $\mathcal{D}$. It interacts with an *initiator* I, a *responder* R, and the adversary $\mathcal{S}$ via the following messages:

Upon receiving a query $(\mathsf{Init}, \mathsf{sid}, I, R, \pi_I), \pi_I \in \mathcal{D}$ **from party** I**:**
Record (I, π_I) and send a message (sid, I, R) to $\mathcal{S}$. Ignore all future messages from I with the same ssid.

Upon receiving a query $(\mathsf{sid}, \mathsf{ok})$ **from** $\mathcal{S}$**:**
Send a message $(\mathsf{wakeup}, \mathsf{sid}, I, R)$ to R. Ignore all future (ok) messages.

Upon receiving a query $(\mathsf{Respond}, \mathsf{sid}, I, R, \pi_R)$ **from** R**:**
– If $\pi_R = \pi_I$, then choose $\mathsf{skey} \leftarrow \{0,1\}^\lambda$ and store $\mathsf{out}_I = \mathsf{out}_R = \mathsf{skey}$.
– If $\pi_R \neq \pi_I$, then set $\mathsf{out}_I = \mathsf{out}_R = \perp$.
In both cases, ignore subsequent inputs from R.

Upon receiving a query $(\mathsf{Corrupt}, \mathsf{sid}, I, R, (P, K))$ **from** $\mathcal{S}$**, where** $P \in \{I, R\}$**:**

If $\pi_R = \pi_I$, then set $\mathsf{out}_I = \mathsf{out}_R = K$. Output the message $(\mathsf{corrupted})$ to P.

Upon receiving a query $(\mathsf{sid}, \mathsf{Out}, P), P \in \{I, R\}$ **from** $\mathcal{S}$**:**
Send $(\mathsf{output}, \mathsf{sid}, I, R, \mathsf{out}_P)$ to the player P. Ignore all subsequent (Out, P) queries for the same player P.

Fig. 2. The Randomized Equality-Testing functionality $\mathcal{F}_{\mathsf{re}}$

Connection to $\mathcal{F}_{\mathsf{pwKE}}$. Let $\mathsf{s}\mathcal{F}_{\mathsf{re}}$ be the functionality obtained by applying the "split functionality" transformation of [BCL+05] to the functionality $\mathcal{F}_{\mathsf{re}}$. We show that $\mathsf{s}\mathcal{F}_{\mathsf{re}}$ is already powerful enough to capture the password-authenticated key exchange functionality $\mathcal{F}_{\mathsf{pwKE}}$. More formally, we show the following proposition whose proof is deferred to the full version.

Proposition 4. *There is a protocol* Π_{REtoPAKE} *that securely implements the* $\mathcal{F}_{\mathsf{pwKE}}$ *functionality in the* $\mathsf{s}\mathcal{F}_{\mathsf{re}}$*-hybrid model, tolerating adaptive corruptions and without assuming authenticated channels.*

2.2 Randomized Equality Testing Protocol 1

We now describe our first randomized equality testing protocol Π_{REfromOT} in the $\mathcal{F}_{\mathsf{OT}}$-hybrid model. We show that the protocol is secure against *adaptive corruptions* in a model with built-in authenticated channels.

UC Randomized Equality Testing Protocol Π_{REfromOT} in the $\mathcal{F}_{\mathsf{OT}}$-Hybrid Model

The protocol is between two players I and R. Assume that the dictionary $\mathcal{D} \subseteq \{0,1\}^{\ell}$.

Code for Player P_b interacting with P_{1-b}, where $b \in \{0,1\}$ and $P_0, P_1 \in \{I, R\}$.

1. P_b, on input $\pi \in \mathcal{D}$ does the following. Let $\pi = \pi_1, \ldots, \pi_\ell$, where $\pi_i \in \{0,1\}$.
   - **(Run OT as the Receiver)** For every $i \in [1 \ldots \ell]$, send $(\mathsf{Receiver}, \mathsf{sid}||i, \pi_i)$ to $\mathcal{F}_{\mathsf{OT}}$.
   - **(Run OT as the Sender)** For every $i \in [1 \ldots \ell]$, choose a pair of random strings $(w^b_{i,0}, w^b_{i,1}) \in \{0,1\}^{3\lambda}$ and send the message $(\mathsf{Sender}, \mathsf{sid}||i, (w^b_{i,0}, w^b_{i,1}))$ to $\mathcal{F}_{\mathsf{OT}}$.
2. P_b waits to receive messages $(\mathsf{Output}, \mathsf{sid}||i, (w')^b_i)$ from $\mathcal{F}_{\mathsf{OT}}$ for all $i \in [1 \ldots \ell]$. It then computes $K' = \bigoplus_{i=1}^{\ell} w'_i = \mathsf{skey}'||\mathsf{test}'_0||\mathsf{test}'_1$.
3. P_b computes the value

$$K = \bigoplus_{i=1}^{\ell} w^b_{i,\pi_i} \stackrel{\Delta}{=} \mathsf{skey}||\mathsf{test}_0||\mathsf{test}_1 \quad (\text{where } \mathsf{skey}, \mathsf{test}_0, \mathsf{test}_1 \in \{0,1\}^{\lambda})$$

   and sends $(\mathsf{test}_b \oplus \mathsf{test}'_b)$ to P_{1-b}.
4. P_b waits to receive $\mathsf{test} \oplus \mathsf{test}' \in \{0,1\}^{\lambda}$ from P_{1-b}, and checks if $\mathsf{test} \oplus \mathsf{test}'$ matches $\mathsf{test}_{1-b} \oplus \mathsf{test}'_{1-b}$.
   - If the check does not pass, then output $\perp$.
   - If the check passes, output $(\mathsf{sid}, \mathsf{skey}' \oplus \mathsf{skey})$.

   In either case, terminate the session.

Fig. 3. Randomized Equality Testing Protocol Π_{REfromOT}

Theorem 1. *The protocol Π_{REfromOT} in Figure 3 securely realizes the randomized equality testing functionality $\mathcal{F}_{\mathsf{re}}$ in the $\mathcal{F}_{\mathsf{OT}}$-hybrid model, in the presence of adaptive corruptions, and assuming authenticated channels.*

Proof. Let $\mathcal{A}$ be an *adaptive* adversary interacting with a pair of parties I and R running the protocol Π_{REfromOT}. We show that for every such $\mathcal{A}$, there is an ideal-world adversary (simulator) $\mathcal{S}$ interacting with dummy parties and the ideal functionality $\mathcal{F}_{\mathsf{re}}$ such that no environment $\mathcal{Z}$ can distinguish between an interaction with $\mathcal{A}$ in the protocol Π_{REfromOT} and an interaction with $\mathcal{S}$ in the ideal world.

Description of the Simulator. The simulator $\mathcal{S}$ starts by invoking a copy of $\mathcal{A}$ and running a simulated interaction of $\mathcal{A}$ with the environment $\mathcal{Z}$ and the parties running the protocol. $\mathcal{S}$ proceeds as follows:

Simulating the Communication with $\mathcal{Z}$: Every message that $\mathcal{S}$ receives from the environment $\mathcal{Z}$ is written to $\mathcal{A}$'s input tape. In the same vein, every output value that $\mathcal{A}$ writes to its output tape is copied to $\mathcal{S}$'s own output tape (to be read later by $\mathcal{Z}$).

Simulating the Case when the Initiator I is Corrupted: $\mathcal{S}$ does the following.

- Upon receiving a message $(\mathsf{Sender}, \mathsf{sid}||i, (\omega_0, \omega_1))$ from $\mathcal{A}$ in session $\mathsf{sid}, \mathsf{ssid}$, record $w^b_{i,0} = \omega_0$ and $w^b_{i,1} = \omega_1$.
- Upon receiving a message $(\mathsf{Receiver}, \mathsf{sid}||i, \beta)$ from $\mathcal{A}$, record $\pi_i = \beta$. Choose a uniformly random string $(w')^b_{i,\pi_i} \leftarrow \{0,1\}^\lambda$ and send it to $\mathcal{A}$.
- As soon as all the bits π_i are received, let $\pi = \pi_1 \dots \pi_\ell$, and write the message $(\mathsf{Init}, \mathsf{sid}, \mathsf{ssid}, I, R, \pi)$ on the outgoing communication tape of the corrupted (ideal model) I (to be sent to the functionality $\mathcal{F}_{\mathsf{re}}$). Also send (ok) to the ideal functionality $\mathcal{F}_{\mathsf{re}}$.
- As soon as all the pairs $(w^b_{i,0}, w^b_{i,1})$ have been recorded (for all $i \in [\ell]$), compute the key

$$K' = \bigoplus_{i=1}^{\ell} (w')^b_{i,\pi_i} \triangleq \mathsf{skey}'||\mathsf{test}' \qquad \text{and} \qquad K = \bigoplus_{i=1}^{\ell} w^b_{i,\pi_i} \triangleq \mathsf{skey}||\mathsf{test}$$

  where $\mathsf{skey}, \mathsf{skey}', \mathsf{test}, \mathsf{test}' \in \{0,1\}^\lambda$. Send a message $(\mathsf{Corrupt}, \mathsf{sid}, I, R, \mathsf{skey} \oplus \mathsf{skey}')$ to the functionality $\mathcal{F}_{\mathsf{re}}$.
- Send the messages (out, I) and (out, R) to $\mathcal{F}_{\mathsf{re}}$, and receive out_I from $\mathcal{F}_{\mathsf{re}}$. (*Remark:* Note that in case the inputs of I and R match, $\mathsf{out}_I = \mathsf{skey} \oplus \mathsf{skey}'$, otherwise $\mathsf{out}_I = \perp$. Thus, given out_I, $\mathcal{S}$ can tell if the inputs of I and R are the same or not.)
- If $\mathsf{out}_I \neq \perp$, send test' to $\mathcal{A}$. Otherwise send a uniformly random string $\mathsf{test}'' \leftarrow \{0,1\}^\lambda$ to $\mathcal{A}$.

Simulating the case when the Responder R is Corrupted: Since the protocol is completely symmetric between the two parties, the simulation is identical to that for a corrupted initiator I, except that $\mathcal{S}$ runs the following pre-amble phase:

– Wait to receive a message $(\mathsf{sid}, \mathsf{ssid}, I, R)$ from the functionality $\mathcal{F}_{\mathsf{re}}$. Send $(\mathsf{sid}, \mathsf{ssid}, \mathsf{ok})$ to $\mathcal{F}_{\mathsf{re}}$ and receive a message $(\mathsf{wakeup}, \mathsf{sid}, \mathsf{ssid}, I, R)$ from $\mathcal{F}_{\mathsf{re}}$.

The simulation from this point on is identical to the simulation for a corrupted I.

Simulating the case when both or neither of the parties is Corrupted: When both parties are corrupted, the simulator simply runs $\mathcal{A}$ internally (who itself generates all the messages). When neither party is corrupted, $\mathcal{S}$ produces uniformly random strings $\mathsf{test}, \mathsf{test}' \leftarrow \{0,1\}^{\ell}$ and forwards them to $\mathcal{A}$.

Dealing with Corruptions: Upon receiving a "Corrupt P_b" message from $\mathcal{A}$, where $P_b \in \{I, R\}$, corrupt the ideal-model $\tilde{P}_b \in \{\tilde{I}, \tilde{R}\}$, and obtain its input π_b and output out_{P_b}. When party P_b is corrupted by $\mathcal{A}$, $\mathcal{S}$ must produce both an input (and output) as well as random tape and private view for party P_b in the simulation. The random tape of party P_b consists of the pairs $(w^b_{i,0}, w^b_{i,1})$ for every $i \in [1 \dots \ell]$ and the private view of party P_b consists of the strings $(w')^b_i$ for every $i \in [1 \dots \ell]$. Thus, upon corruption of party P_b $\mathcal{S}$ will return to $\mathcal{A}$ the input π_b and output out_{P_b} obtained by corrupting the ideal-model $\tilde{P}_b$ as well as the values $w^b_{i,0}, w^b_{i,1} \in \{0,1\}^{3\lambda}, (w')^b_i$ for every $i \in [1 \dots \ell]$. There are several cases to consider:

Corruption of party P_b before messages have been exchanged in Stage 3. $\mathcal{S}$ corrupts the ideal-model $\tilde{P}_b \in \{\tilde{I}, \tilde{R}\}$, and obtains its input π_{P_b}.

– If party P_{1-b} is not yet corrupted then $\mathcal{S}$ chooses $w^b_{i,0}, w^b_{i,1}, (w')^b_i$ for every $i \in [1 \dots \ell]$ uniformly at random and returns these values to $\mathcal{A}$. $\mathcal{S}$ continues the simulation for the case that party P_b is corrupted.

– If party P_{1-b} has already been corrupted then the values $w^{1-b}_{i,0}, w^{1-b}_{i,1}, (w')^{1-b}_i$ for every $i \in [1 \dots \ell]$ are already known and so $\mathcal{S}$ must ensure that the values of $w^b_{i,0}, w^b_{i,1}, (w')^b_i$ for every $i \in [1 \dots \ell]$ are consistent with these values.

Thus, $\mathcal{S}$ does the following: For every $i \in [1 \dots \ell]$, $\mathcal{S}$ sets $w^b_{i,\pi_{1-b,i}} = (w')^{1-b}_i$ and chooses $w^b_{i,1-\pi_{1-b,i}}$ uniformly at random. For every $i \in [1 \dots \ell]$, $\mathcal{S}$ sets $(w')^b_i = w^{1-b}_{i,\pi_{b,i}}$. $\mathcal{S}$ returns these values to $\mathcal{A}$ and continues the simulation for the case that both parties are corrupted.

Corruption of party P_b after messages have been exchanged in Stage 3. $\mathcal{S}$ corrupts the ideal-model $\tilde{P}_b \in \{\tilde{I}, \tilde{R}\}$, obtains its input π_b, and output of either skey or $\perp$.

– If party P_{1-b} is not yet corrupted then $\mathcal{S}$ does the following: If the output is skey then $\mathcal{S}$ chooses $w^b_{i,0}, w^b_{i,1}, (w')^b_i$ for every $i \in [1 \dots \ell]$ uniformly at random conditioned on $K \oplus K'$ being consistent with $\mathsf{test}_b \oplus \mathsf{test}'_b$, $\mathsf{test}_{1-b} \oplus \mathsf{test}'_{1-b}$ and returns these values to $\mathcal{A}$. If the output is $\perp$ $\mathcal{S}$ chooses $w^b_{i,0}, w^b_{i,1}, (w')^b_i$ for every $i \in [1 \dots \ell]$ uniformly at random conditioned on $K \oplus K'$ being consistent with $\mathsf{test}_b \oplus \mathsf{test}'_b$ and returns these values to $\mathcal{A}$. $\mathcal{S}$ continues the simulation for the case that party P_b is corrupted.

– If party P_{1-b} has already been corrupted then the values $w_{i,0}^{1-b}, w_{i,1}^{1-b}, (w')_i^{1-b}$ for every $i \in [1 \dots \ell]$ are already known and so $\mathcal{S}$ must ensure that the values of $w_{i,0}^{b}, w_{i,1}^{b}, (w')_i^{b}$ for every $i \in [1 \dots \ell]$ are consistent with these values.
Thus, if the output is skey, $\mathcal{S}$ does the following: For every $i \in [1 \dots \ell]$, $\mathcal{S}$ sets $w_{i,\pi_{1-b,i}}^{b} = (w')_i^{1-b}$ and chooses $w_{i,1-\pi_{1-b,i}}^{b}$ uniformly at random. For every $i \in [1 \dots \ell]$, $\mathcal{S}$ sets $(w')_i^{b} = w_{i,\pi_{b,i}}^{1-b}$. $\mathcal{S}$ returns these values to $\mathcal{A}$.
If the output is $\perp$ then there must be some $i^* \in [1 \dots \ell]$ such that $\pi_{b,i^*} \neq \pi_{1-b,i^*}$. Thus, $\mathcal{S}$ does the following: For every $i \in [1 \dots \ell]$, $\mathcal{S}$ sets $w_{i,\pi_{1-b,i}}^{b} = (w')_i^{1-b}$ and chooses $w_{i,1-\pi_{1-b,i}}^{b}$ uniformly at random conditioned on $K \oplus K'$ being consistent with $\mathsf{test}_b \oplus \mathsf{test}'_b$. (note that this is always possible since we can set $w_{i^*,\pi_{b,i^*}}^{b}$ to be whatever we want. For every $i \in [1 \dots \ell]$, $\mathcal{S}$ sets $(w')_i^{b} = w_{i,\pi_{b,i}}^{1-b}$. $\mathcal{S}$ returns these values to $\mathcal{A}$ and continues the simulation for the case that both parties are corrupted.

Proof of Indistinguishability. We show that $\text{IDEAL}_{\mathcal{F}_{\mathsf{re}},\mathcal{S},\mathcal{Z}} \equiv \text{REAL}_{\Pi_{\mathsf{REfromOT}},\mathcal{A},\mathcal{Z}}$. The main idea of the proof is this: Let π_I and π_R be the inputs of I and R. (In case one or both of them are corrupted, then set π_I, resp. π_R, to be the string that the simulator extracts from I, resp. R) If $\pi_I = \pi_R$, it is easy to see that the simulation is perfect. If $\pi_I \neq \pi_R$, then we claim that the key K_R that the responder R computes is uniformly random from the view of $\mathcal{A}$. This is because the adversary $\mathcal{A}$ receives $w'_{i,\pi_{I,i}}$ for all $i \in [\ell]$ and K_R is computed as

$$K_R = \bigoplus_{i=1}^{\ell} w'_{i,\pi_{R,i}}$$

Without loss of generality, say $\pi_{R,1} \neq \pi_{I,1}$. Then, $(w')_{1,\pi_{R,1}}^{b}$ is uniformly random from the view of $\mathcal{A}$. In particular, this means that K_R is uniformly random from $\mathcal{A}$'s view, and thus, the message test' that it gets is correctly distributed. Furthermore, the simulated distribution is identical to the distribution generated by executing Π_{REfromOT} except for this. Thus, it follows that $\text{IDEAL}_{\mathcal{F}_{\mathsf{re}},\mathcal{S},\mathcal{Z}} \equiv \text{REAL}_{\Pi_{\mathsf{REfromOT}},\mathcal{A},\mathcal{Z}}$.

2.3 Implementing the Split $\mathcal{F}_{\mathsf{re}}$ Functionality without Authenticated Channels

The protocol Π_{REfromOT} in Section 2.2 implements the randomized equality testing functionality $\mathcal{F}_{\mathsf{re}}$ in the *authenticated channels* model. In this section, we use the results of Barak et al. [BCL+05] together with a specific implementation of the $\mathcal{F}_{\mathsf{OT}}$ functionality from Garay, Wichs and Zhou [GWZ09] to show that the protocol can be transformed into a protocol $\mathsf{s}\Pi_{\mathsf{REfromOT}}$ that implements the "split version" of the equality-testing functionality (called $\mathsf{s}\mathcal{F}_{\mathsf{re}}$). The new protocol does not assume authenticated channels, and yet, retain security against adaptive corruptions. For completeness, we define $\mathsf{s}\mathcal{F}_{\mathsf{re}}$ in the full version, and state the result of this transformation in Theorem 2.

Theorem 2. *There is a protocol* $\mathsf{s}\Pi_{\mathsf{REfromOT}}$ *that securely implements the split functionality* $\mathsf{s}\mathcal{F}_{\mathsf{re}}$ *in the* $\mathcal{F}_{\mathsf{crs}}$*-hybrid model, tolerating adaptive corruptions without erasures and without authenticated channels. The protocol is based on either DDH or the decisional composite residuosity (DCR) assumption, runs in a constant number of rounds and exchanges a constant number of group elements per session key.*

Proof. First, we note that the multi-session version of $\mathcal{F}_{\mathsf{re}}$ can be implemented using access to the multi-session version of $\mathcal{F}_{\mathsf{OT}}$ – essentially each new session of $\mathcal{F}_{\mathsf{re}}$ utilizes new invocation of the OT protocol. Then, using the result of Garay et al., the multi-session version of $\mathcal{F}_{\mathsf{OT}}$ can be implemented in the $\mathcal{F}_{\mathsf{crs}}$-hybrid model under either the DDH or DCR assumption. Put together, we have a protocol that implements the multi-session version of $\mathcal{F}_{\mathsf{re}}$ in the $\mathcal{F}_{\mathsf{crs}}$-hybrid model. Now, a theorem of Barak et al. [BCL+05] shows that any such protocol can be converted into a protocol for the split functionality $\mathsf{s}\mathcal{F}_{\mathsf{re}}$.

3 Concurrent PAKE from OT

We present a framework for concurrent PAKE based on OT, and show how to instantiate the underlying building blocks from search assumptions.

Definitions. We begin with an overview of the security definition for concurrent PAKE given in [GK10, BPR00] (detailed definitions are presented in the full version). Informally, an adversary interacts with various instances in the following ways:

- it can initiate and interact in an instance with any honest party;
- it can ask for the session key for some completed instance;
- it can passively eavesdrop on an instance between two honest parties;

The first two modes of interaction constitute a so-called "on-line attack"; the third one does not. Informally, a secure PAKE protocol guarantees secrecy of the session keys even in the presence of an active adversary. That is, we say that an adversary succeeds if it manages to distinguish the session key for some fresh instance from random (where an instance is "fresh" if the adversary has not previous asked for its session key). We use $\mathbf{AdvPAKE}_{\mathcal{A}}(\lambda)$ to denote the success probability of an adversary $\mathcal{A}$. Now, an adversary can always succeed with probability 1 by trying all passwords in the dictionary one-by-one. Informally, a protocol is secure if this is the best an adversary can do. Formally, we say that an instance represents an *on-line attack* if the adversary participated in the instance. The number of on-line attacks is a bound on the number of passwords the adversary could have tested in an on-line fashion.

We say that a PAKE protocol is *concurrently secure with explicit mutual authentication* if for all dictionaries $\mathcal{D}_\lambda$ and for all PPT adversaries $\mathcal{A}$ making at most $Q(\lambda)$ online attacks, the quantity $\mathbf{AdvPAKE}_{\mathcal{A}}(\lambda) - Q(\lambda)/|\mathcal{D}_\lambda|$ is bounded by a negligible function.

3.1 A General Framework

We present our general framework for concurrent PAKE (a variant of the Groce-Katz protocol) in Fig 4. The ingredients are a labeled CCA-secure encryption $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, and an OT protocol $(\mathbf{S}, \mathbf{R})$ in the CRS model that is (1) computationally hiding against $\mathbf{S}^*$ and (2) straight-line extractable and statistically hiding against $\mathbf{R}^*$

Overview. Here is an overview of the construction, assuming 1-out-of-$|\mathcal{D}|$ OT for simplicity:

- Both parties U and U' run the basic protocol: U acts as the OT receiver and uses as input his password $\pi_{U,U'}$. U' acts as the OT sender and picks $|\mathcal{D}|$ random strings $r_1, \ldots, r_{|\mathcal{D}|}$ as input. U parses the OT output as $\mathsf{skey}\|\mathsf{rand}$ and U' parses $r_{\pi_{U,U'}}$ as $\mathsf{skey}'\|\mathsf{rand}'$.
- U' sends an encryption C of $\pi_{U,U'}$ using randomness rand' and as label the transcript of the basic protocol (plus the identities), under a public key for a CCA2-secure encryption scheme that is part of the CRS.
- U checks if C is computed with the same password by encrypting $\pi_{U,U'}$ with randomness rand. If the ciphertext matches C, both parties output skey as the session key.

See Figure 4 for a description of the protocol. We establish the following:

Proposition 5. *Suppose* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a labeled CCA-secure encryption scheme and* $(\mathbf{S}, \mathbf{R})$ *is an OT protocol in the CRS model that is (1) computationally hiding against* $\mathbf{S}^*$ *and (2) straight-line extractable and statistically hiding against* $\mathbf{R}^*$*. Then, the protocol in Fig 4 is a secure PAKE protocol with explicit mutual authentication.*

Proof Overview. We begin with a brief argument of security for the case where there is a single instance on the left and on the right:

- First, we want to argue that by OT security against senders, the LHS Stage 1 hides U's input π (which we extract) and so $\mathcal{A}$'s input $\tilde{\pi}$ to the RHS Stage 1 must be "independent" of π. This would imply that with probability $1 - 1/|\mathcal{D}|$, we have $\tilde{\pi} \neq \pi$ and thus U''s challenge test is statistically hidden from $\mathcal{A}$. Thus we bound the probability $\mathcal{A}$ wins on the right.
- Next, observe that if $\mathcal{A}$ plays a relaying strategy for Stages 1 on the left and the right, then it must continue to play a relaying strategy for U or U' to accept (since the transcript of Stage 1 uniquely determines an accepting transcript for the protocol). Otherwise, the labels for the CCA encryptions in Stage 2 on the left and right must differ. We may then argue that U''s encryption of π on the right does not help $\mathcal{A}$ provide a valid encryption of π on the left. Thus, we bound the probability $\mathcal{A}$ wins on the left.

The main subtlety lies in the first step: as stated, we require the underlying OT protocol to hide the receiver's input against a cheating sender that has access to an extraction trapdoor (which would require that the underlying OT protocol non-malleable). We bypass this issue via a more refined analysis. We defer the formal proof to the full version.

Concurrent PAKE

Common Reference String: The CRS for $(\mathbf{S}, \mathbf{R})$ and a public key PK for $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

Inputs: Parties U and U' participating in instances Π_U^i and $\Pi_{U'}^j$, respectively, hold joint password $\pi = \pi_{U,U'} \in \mathcal{D}$, where $\mathcal{D} \subseteq \{0,1\}^\ell$.

PAKE phase:

Stage 1. U and U' engage in ℓ executions of $(\mathbf{S}, \mathbf{R})$ in parallel. In the i'th execution of $(\mathbf{S}, \mathbf{R})$:

- U' chooses a pair of random strings $(w_i^0, w_i^1) \leftarrow_R \{0,1\}^{3\lambda}$ and runs $\mathbf{S}$ with input (w_i^0, w_i^1).
- U runs $\mathbf{R}$ with input $\pi_i \in \{0,1\}$ and receives output $w_i' := w_i^{\pi_i}$.

Stage 2. U' computes

$$\mathsf{rand}||\mathsf{test}||\mathsf{skey} := \bigoplus_{i=1}^{\ell} w_i^{\pi_i} \quad (\text{where } \mathsf{rand}, \mathsf{skey}, \mathsf{test} \in \{0,1\}^\lambda)$$

and sends $C := \mathsf{Enc}_{\mathrm{PK}}^{U||U'||\mathsf{trans}}(\pi; \mathsf{rand})$ to U where trans is the concatenation of the transcripts of all ℓ executions of $(\mathbf{S}, \mathbf{R})$.

Stage 3. U computes

$$\mathsf{rand}'||\mathsf{test}'||\mathsf{skey}' := \bigoplus_{i=1}^{\ell} w_i' \quad (\text{where } \mathsf{rand}', \mathsf{skey}', \mathsf{test}' \in \{0,1\}^\lambda)$$

and sends test' and sets its session key to skey' if $C = \mathsf{Enc}_{\mathrm{PK}}^{U||U'||\mathsf{trans}}(\pi; \mathsf{rand}')$ and aborts otherwise.

Stage 4. U' sets its session key to skey if $\mathsf{test}' = \mathsf{test}$ and aborts otherwise.

Fig. 4. Concurrent PAKE

3.2 Instantiating the Underlying OT

We present two approaches for instantiating the underlying OT in our general framework for concurrent PAKE. Recall that we require an OT protocol $(\mathbf{S}, \mathbf{R})$ in the CRS model that is (1) computationally hiding against $\mathbf{S}^*$ and (2) straight-line extractable and statistically hiding against $\mathbf{R}^*$.

Instantiations from Dual-Mode Encryption. In [PVW08], Peikert, Vaikuntanathan and Waters present a novel abstraction called "dual-mode cryptosystems" and show how to construct UC-secure OT from any dual-mode cryptosystem in the CRS model (where every pair of parties share a CRS). Moreover, in the so-called "messy mode", the ensuing OT protocol achieves statistical security against a corrupted receiver. We observe that the same protocol also achieves the security guarantees that we require. Combined with our general framework, we obtain the result stated in Proposition 2.

Instantiations from CDH and hardness of factoring. We start with a two-message bit-OT protocol in the CRS model that is (1) computationally hiding against $\mathbf{R}^*$ and (2) straight-line extractable and statistically hiding against $\mathbf{S}^*$ (note these are the "opposite" properties of what we need). Indeed, the Bellare-Micali OT protocol [BM89] based on CDH satisfies these properties. To obtain an instantiation based on hardness of factoring, we use the fact that CDH over $\mathbb{Z}_N^*$ is as hard as factoring [HK09, M88, S85]. We note that 2-message OT protocols were given by Halevi and Kalai [HK07]; however, their constructions are based on hash proof systems and thus are limited to decisional assumptions.

Next, we apply the "OT reversal" transformation of Wolf and Wullschleger [WW06] to obtain a three-message bit-OT protocol. We show that the ensuing bit-OT protocol has the properties we need, namely computationally hiding against $\mathbf{S}^*$ and straight-line extractable and statistically hiding against $\mathbf{R}^*$. Finally, we apply the bit OT to string OT transformation of Brassard, et. al [BCR86] (which is round-preserving) to obtain a string OT protocol with the properties we need. We defer details to the full version.

References

[ABCP06] Abdalla, M., Bresson, E., Chevassut, O., Pointcheval, D.: Password-Based Group Key Exchange in a Constant Number of Rounds. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 427–442. Springer, Heidelberg (2006)

[ACCP08] Abdalla, M., Catalano, D., Chevalier, C., Pointcheval, D.: Efficient Two-Party Password-Based Key Exchange Protocols in the UC Framework. In: Malkin, T. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 335–351. Springer, Heidelberg (2008)

[ACP09] Abdalla, M., Chevalier, C., Pointcheval, D.: Smooth Projective Hashing for Conditionally Extractable Commitments. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 671–689. Springer, Heidelberg (2009)

[AP06] Abdalla, M., Pointcheval, D.: A Scalable Password-Based Group Key Exchange Protocol in the Standard Model. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 332–347. Springer, Heidelberg (2006)

[BCL+05] Barak, B., Canetti, R., Lindell, Y., Pass, R., Rabin, T.: Secure Computation Without Authentication. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 361–377. Springer, Heidelberg (2005)

[BCR86] Brassard, G., Crépeau, C., Robert, J.M.: All-or-Nothing Disclosure of Secrets. In: Brassard, G. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 234–238. Springer, Heidelberg (1987)

[BM89] Bellare, M., Micali, S.: Non-interactive Oblivious Transfer and Applications. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 547–557. Springer, Heidelberg (1990)

[BM93] Bellovin, S.M., Merritt, M.: Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password file compromise. In: Ashby, V. (ed.) 1st ACM Conference on Computer and Communications Security, pp. 244–250. ACM Press (November 1993)

[BMP00] Boyko, V., MacKenzie, P., Patel, S.: Provably Secure Password-Authenticated Key Exchange Using Diffie-Hellman. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 156–171. Springer, Heidelberg (2000)

[BPR00] Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated Key Exchange Secure against Dictionary Attacks. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 139–155. Springer, Heidelberg (2000)

[C01] Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: 42nd Annual Symposium on Foundations of Computer Science (FOCS), pp. 136–145. IEEE Computer Society Press (2001)

[CCGS10] Camenisch, J., Casati, N., Gross, T., Shoup, V.: Credential Authenticated Identification and Key Exchange. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 255–276. Springer, Heidelberg (2010)

[CHK+05] Canetti, R., Halevi, S., Katz, J., Lindell, Y., MacKenzie, P.: Universally Composable Password-Based Key Exchange. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 404–421. Springer, Heidelberg (2005)

[CS98] Cramer, R., Shoup, V.: A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)

[CS02] Cramer, R., Shoup, V.: Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)

[G08] Gennaro, R.: Faster and Shorter Password-Authenticated Key Exchange. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 589–606. Springer, Heidelberg (2008)

[GJO10] Goyal, V., Jain, A., Ostrovsky, R.: Password-Authenticated Session-Key Generation on the Internet in the Plain Model. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 277–294. Springer, Heidelberg (2010)

[GK10] Groce, A., Katz, J.: A new framework for efficient password-based authenticated key exchange. In: ACM Conference on Computer and Communications Security, pp. 516–525 (2010)

[GL01] Goldreich, O., Lindell, Y.: Session-Key Generation Using Human Passwords Only. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 408–432. Springer, Heidelberg (2001), http://eprint.iacr.org/2000/057

[GL03] Gennaro, R., Lindell, Y.: A Framework for Password-Based Authenticated Key Exchange (Extended Abstract). In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 524–543. Springer, Heidelberg (2003), http://eprint.iacr.org/2003/032.ps.gz

[GWZ09] Garay, J.A., Wichs, D., Zhou, H.-S.: Somewhat Non-committing Encryption and Efficient Adaptively Secure Oblivious Transfer. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 505–523. Springer, Heidelberg (2009)

[HK07] Kalai, Y.T.: Smooth Projective Hashing and Two-Message Oblivious Transfer. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 78–95. Springer, Heidelberg (2005); Cryptology ePrint Archive, Report 2007/118 (2007)

[HK09] Hofheinz, D., Kiltz, E.: The Group of Signed Quadratic Residues and Applications. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 637–653. Springer, Heidelberg (2009)

[JG04] Jiang, S., Gong, G.: Password Based Key Exchange with Mutual Authentication. In: Handschuh, H., Hasan, M.A. (eds.) SAC 2004. LNCS, vol. 3357, pp. 267–279. Springer, Heidelberg (2004)

[KMTG05] Katz, J., MacKenzie, P., Taban, G., Gligor, V.D.: Two-Server Password-Only Authenticated Key Exchange. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 1–16. Springer, Heidelberg (2005)

[KOY01] Katz, J., Ostrovsky, R., Yung, M.: Efficient Password-Authenticated Key Exchange Using Human-Memorable Passwords. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 475–494. Springer, Heidelberg (2001)

[KV09] Katz, J., Vaikuntanathan, V.: Smooth Projective Hashing and Password-Based Authenticated Key Exchange from Lattices. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 636–652. Springer, Heidelberg (2009)

[KV11] Katz, J., Vaikuntanathan, V.: Round-Optimal Password-Based Authenticated Key Exchange. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 293–310. Springer, Heidelberg (2011)

[M88] McCurley, K.S.: A key distribution system equivalent to factoring. Journal of Cryptology 1(2), 95–105 (1988)

[MPS00] MacKenzie, P., Patel, S., Swaminathan, R.: Password-Authenticated Key Exchange Based on RSA. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 599–613. Springer, Heidelberg (2000)

[NV04] Nguyen, M.-H., Vadhan, S.P.: Simpler Session-Key Generation from Short Random Passwords. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 428–445. Springer, Heidelberg (2004)

[PVW08] Peikert, C., Vaikuntanathan, V., Waters, B.: A Framework for Efficient and Composable Oblivious Transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008)

[PW08] Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Ladner, R.E., Dwork, C. (eds.) 40th Annual ACM Symposium on Theory of Computing (STOC), pp. 187–196. ACM Press (May 2008)

[S85] Shmuely, Z.: Composite diffie-hellman public-key generating systems are hard to break. Technical Report 356, Technion (1985)

[WW06] Wolf, S., Wullschleger, J.: Oblivious Transfer Is Symmetric. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 222–232. Springer, Heidelberg (2006)

Strongly Secure Authenticated Key Exchange from Factoring, Codes, and Lattices

Atsushi Fujioka, Koutarou Suzuki, Keita Xagawa, and Kazuki Yoneyama

NTT Information Sharing Platform Laboratories
3-9-11 Midori-cho Musashino-shi Tokyo 180-8585, Japan
yoneyama.kazuki@lab.ntt.co.jp

Abstract. An unresolved problem in research on authenticated key exchange (AKE) is to construct a secure protocol against advanced attacks such as key compromise impersonation and maximal exposure attacks without relying on random oracles. HMQV, a state of the art AKE protocol, achieves both efficiency and the strong security model proposed by Krawczyk (we call it the CK^+ model), which includes resistance to advanced attacks. However, the security proof is given under the random oracle model. We propose a generic construction of AKE from a key encapsulation mechanism (KEM). The construction is based on a chosen-ciphertext secure KEM, and the resultant AKE protocol is CK^+ secure in the standard model. The protocol gives the first CK^+ secure AKE protocols based on the hardness of integer factorization problem, code-based problems, or learning problems with errors. In addition, instantiations under the Diffie-Hellman assumption or its variant can be proved to have strong security without non-standard assumptions such as πPRF and KEA1.

Keywords: authenticated key exchange, CK^+ model, key encapsulation mechanism.

1 Introduction

1.1 Background

Establishing secure channels is one of the most important areas of cryptographic research. Secure channels provide secrecy and authenticity for both communication parties. When parties can share secret information via a public communication channel, secure channels would be constructed on (symmetric key) encryptions and message authentication codes with the shared secret information called session keys. Public-key cryptography can provide various solutions: one approach uses a *key encapsulation mechanism* (KEM) and another uses *authenticated key exchange* (AKE).

In KEM, a receiver has public information, called a *public key*, and the corresponding secret information, called a *secret key*. The public key is expected to be certified with the receiver's identity through an infrastructure such as a *public key infrastructure* (PKI). A sender who wants to share information, a *session key*, with

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 467–484, 2012.

the receiver sends a *ciphertext* of the information and, the receiver decrypts the ciphertext to extract the information. KEM can be easily constructed from *public-key encryption* (PKE) under the reasonable condition that the plaintext space is sufficiently large. The desirable security notion of KEM is formulated as the *indistinguishability against chosen ciphertext attacks* (IND-CCA).

In AKE, each party has public information, called a *static public key*, and the corresponding secret information, called a *static secret key*. The static public key is also expected to be certified with a party's identity through an infrastructure such as PKI. A party who wants to share information with a party exchanges *ephemeral public keys*, generated from the corresponding *ephemeral secret keys*, and computes a *session state* from their static public keys, the corresponding static secret keys, the exchanged ephemeral public keys, and the corresponding ephemeral secret keys. Both parties then derive a *session key* from these values including the session state using a function called the *key derivation function*. Many studies have investigated the security notion of AKE [1,2,3,4,5]. The first security notion of AKE based on indistinguishability was provided by Bellare and Rogaway [1] (BR model). The BR model captures basic security requirements for AKE such as known key security and impersonation resilience. However, the BR model cannot grasp more complicated situations where a static secret key or session state of a party has been leaked. Accordingly, Canetti and Krawczyk [2] defined the first security notion of AKE capturing the leakage of static secret keys and session state and called it the *Canetti-Krawczyk (CK) model*. Though the CK model represents leakage of information other than the target session of the adversary, some advanced attacks such as key compromise impersonation (KCI), the breaking of weak perfect forward secrecy (wPFS) and maximal exposure attacks (MEX) use secret information of the target session; thus, the CK model is not resilient to such attacks. KCI means that when given a static secret key, an adversary will try to impersonate some honest party in order to fool the owner of the leaked secret key. wPFS implies that an adversary cannot recover a session key if the adversary does not modify messages of the target session and the session is executed before the static secret keys are compromised. In MEX, an adversary tries to distinguish the session key from a random value under the disclosure of any pair of secret static keys and ephemeral secret keys of the initiator and the responder in the session except for both the static and ephemeral secret keys of the initiator or the responder. Resistance to MEX requires security against any leakage situation that was not presumed. For example, an implementer of AKE may pretend to generate secret keys in an insecure host machine in order to prevent the randomness generation mechanisms in a tamper-proof module such as a smart card. Additionally, if a pseudo-random number generator implemented in a system is poor, secret keys will be known to the adversary even when the generation of ephemeral secret keys is operated in a tamper-proof module. Most AKE protocols are proved in the CK model; however, it is unclear whether such protocols satisfy resistance to advanced attacks due to the limitations of the CK model. A state of the art AKE protocol HMQV [3] satisfies all known security requirements for AKE, including

resistance to KCI, wPFS[1], and MEX, as well as provable security in the CK model. In this paper, we call this security model the CK^+ model; it is known to be one of the 'strongest' models for AKE. LaMacchia et al. [4] and Sarr et al. [5] also proposed very strong security models for AKE by re-formulating the concept of the CK^+ model; they called them the eCK model and the seCK model, respectively. These models allow an adversary to pose a query that directly reveals the ephemeral secret key of the target session. However, Cremers points out that the CK model and the eCK model are incomparable [9,10]; thus, the eCK model is not stronger than the CK model while the CK^+ model is. We will briefly show the difference between the CK^+ model and these models. Since MEX includes any non-trivial leakage situation, HMQV (and CK^+ secure protocols) achieves surprisingly strong security.

1.2 Motivating Problem

HMQV is one of the most efficient protocols and satisfies one of the strongest security models (i.e., CK^+ security). However, the security proof is given in the random oracle model (ROM) under a specific number-theoretic assumption (Diffie-Hellman (DH) assumption). Moreover, to prove resistance to MEX, the knowledge-of-exponent assumption (KEA1) [11] (a widely criticized assumption such as [12]) is also necessary. Hence, one of the open problems in research on AKE is to construct a secure scheme in the CK^+ model without relying on random oracles under standard assumptions.

Boyd et al. [13,14,15] gave a partial solution to this problem by noting that KEM and AKE are closely related and that it might be natural to construct AKE from KEM. They proposed a generic construction of AKE from KEM (BCGNP construction), and its security is proved in the CK model in the standard model (StdM). Also, the BCGNP construction is shown to satisfy resistance to KCI. However, it is unclear whether the BCGNP construction is secure when leakage of secret information occurs (i.e., resistance to MEX). In fact, the BCGNP construction fails to satisfy CK^+ security when we consider the following attack scenario: Two parties exchange ciphertexts of an IND-CCA secure KEM scheme and generate a session key from these. An adversary who obtains the ephemeral secret keys (randomness used in generating ciphertexts) of the parties can compute the session key and win the game. Though the BCGNP construction can be extended to satisfy wPFS, it is guaranteed under the DH assumption, not a general assumption. It is quite restrictive because it cannot be instantiated from the hardness of something other than the DH assumption such as an integer

[1] HMQV does not provide full perfect forward secrecy (fPFS), which is the same as wPFS except that the adversary can modify messages of the target session. Some schemes [6,7,8] have achieved fPFS. However, the schemes [6,7] are clearly vulnerable to MEX; that is, the session key is computable if an adversary obtains an ephemeral secret key of parties in the target session. The other scheme [8] is resilient to MEX, but security is proved in the random oracle model. Upgrading wPFS to fPFS is not that difficult; it can be done by simply adding MAC or a signature of ephemeral public keys. Thus, we do not discuss fPFS in this paper.

factoring problem, code-based problem, or lattice problem. Thus, we still have no AKE protocol that is secure in the 'strongest' model under just a general assumption without relying on random oracles (ROs).

1.3 Our Contribution

We fully solve the open problem by providing a generic construction of AKE from KEM. Our construction is a generalization of the BCGNP construction. The BCGNP construction uses IND-CCA KEM, a strong randomness extractor, and a pseudo-random function (PRF) as building blocks. Our construction effectively follows the design principle of the BCGNP construction. However, we first point out that the security proof of the BCGNP construction is not complete. Specifically, a requirement for KEM has not been formulated. KEM keys must have enough min-entropy in order to make outputs of the strong randomness extractor statistically indistinguishable from a uniformly random chosen element. Thus, the assumption that the KEM scheme satisfies such a property is additionally required. Fortunately, almost all IND-CCA KEM schemes satisfy that. Also, we need an IND-CPA secure KEM in addition to the BCGNP construction. Such an additional KEM can make our scheme wPFS and resilient to MEX. The resultant AKE protocol is CK^+ secure. Its security is proved under the existence of such KEMs, a strong randomness extractor, and a PRF in the StdM. The existence of an IND-CCA secure KEM has been shown from the hardness of integer factoring [16,17], a code-based problem [18,19], or a lattice problem [20,21,22,23,24,25,26]. To the best of our knowledge, our generic construction provides the first CK^+ secure AKE protocols based on the hardness of the above problems. Regarding the DH assumption or its variant, our generic construction is the first protocol that achieves CK^+ security in the StdM without non-standard assumptions (e.g., πPRF and KEA1).

We also rewrite the CK^+ model before proving the security of our generic construction in order to simplify the original model in [3]. Specifically, the original model is defined as a mix of four definitions (i.e., the CK model, wPFS, and resistance to KCI and MEX); thus, the security proof must also be separated into four theorems, which may reduce the readability. Therefore, we reformulate the CK^+ model as follows: wPFS, resistance to KCI, and resistance to MEX are integrated into the experiment of the extended model by exhaustively classifying leakage patterns. This definition is handy to prove security and rigourously captures all required properties.

We summarize our contributions as follows:

- We propose a two-pass generic CK^+ secure AKE construction from IND-CCA secure KEM and PRF in the StdM.
- We achieve the first CK^+ secure AKE protocols based on the hardness of integer factorization problem, code-based problem, and lattice-based problem in the StdM.
- We achieve the first CK^+ secure AKE protocol based on the DH assumption or its variant in the StdM without knowledge assumptions.
- We reformulate the CK^+ model to gain readability of the security proof.

The proposed generic construction can allow a hybrid instantiation; that is, the initiator and the responder can use different KEMs under different assumptions. For example, the initiator uses a factoring-based KEM while the responder uses a lattice-based KEM.

2 Security Model

In this section, we recall the CK$^+$ model that was introduced by [3]. We show a model specified to two pass protocols for simplicity. It can be trivially extended to any round protocol.

2.1 CK$^+$ vs. eCK

As indicated in Table 1, the CK$^+$ model captures all non-trivial patterns of leakage of static and ephemeral secret keys. The eCK model [4], which is a variant of the CK model [2], also captures all non-trivial patterns of leakage, as in Table 1. Since the CK$^+$ model captures all non-trivial patterns of leakage of static and ephemeral secret keys, the CK$^+$ model can theoretically be seen as a completion of the AKE security model.

In Table 1, the six cases in Definition 2 are listed, and these six cases cover wPFS, resistance to KCI, and MEX as follows: Cases 2-(a), 2-(c), and 2-(f) capture KCI, since the adversary obtains the static secret key of one party and the ephemeral secret key of the other party of the test session. Case 2-(e) captures wPFS, since the adversary obtains the static secret keys of both parties of the test session. Cases 2-(b) and 2-(d) capture MEX, since the adversary obtains the ephemeral secret keys of both parties of the test session.

The main difference between the CK$^+$ model and the eCK model is that the CK$^+$ model captures the session state reveal attack, but the eCK model does not. Thus, we adopt the CK$^+$ model, which is stronger than the eCK model from the viewpoint of the session state reveal attack, in this paper.

Notice that the timing of the static and ephemeral key reveal differs in the eCK and CK$^+$ models. In the eCK model, an adversary can issue the static and ephemeral key reveal query adaptively. In contrast, in the CK$^+$ model, an adversary can issue a corrupt query to obtain the static key, and the ephemeral key is given to the adversary when it is determined. We summarize this in Table 2.

2.2 CK$^+$ Security Model

We denote a party by U_i, and party U_i and other parties are modeled as probabilistic polynomial-time (PPT) Turing machines w.r.t. security parameter κ. For party U_i, we denote static secret (public) key by s_i (S_i) and ephemeral secret (public) key by x_i (X_i). Party U_i generates its own keys, s_i and S_i, and the static public key S_i is linked with U_i's identity in some systems like PKI.[2]

[2] Static public keys must be known to both parties in advance. They can be obtained by exchanging them before starting the protocol or by receiving them from a certificate authority. This situation is common for all PKI-based AKE schemes.

Table 1. Classification of attacks and proposed CK^+ model [3] and eCK model [4]

Cases in Def.2	ssk_A	esk_A	ssk_B	esk_B	attack type	CK^+ model [3]	eCK model [4]
2-(a)	r	ok	ok	n	KCI	✓	✓
2-(b)	ok	r	ok	n	MEX	✓	✓
2-(c)	r	ok	ok	r	KCI	✓	✓
2-(d)	ok	r	ok	r	MEX	✓	✓
2-(e)	r	ok	r	ok	wPFS	✓	✓
2-(f)	ok	r	r	ok	KCI	✓	✓

"2-(*)" means the corresponding case in Definition 2. "ssk_A" means the static secret key of owner A of test session sid^*, and "ssk_B" means the static secret key of peer B of test session sid^*. "esk_A" means the ephemeral secret key of test session sid^*, and "esk_B" means the ephemeral secret key of the matching session $\overline{\mathsf{sid}^*}$. "ok" means the secret key is not revealed, "r" means the secret key may be revealed, and "n" means no matching session exists. A ✓ means that the model captures the attack.

Table 2. Comparison of CK^+ model [3] and eCK model [4]

	CK^+ model [3]	eCK model [4]
All non-trivial key leakage	✓	✓
Session state reveal	✓	✗
Adaptive key leakage	✗	✓

A ✓/✗means that the model does/does not capture the attack.

Session. An invocation of a protocol is called a *session*. Session activation is done by an incoming message of the forms $(\Pi, \mathcal{I}, U_A, U_B)$ or $(\Pi, \mathcal{R}, U_B, U_A, X_A)$, where we equate Π with a protocol identifier, $\mathcal{I}$ and $\mathcal{R}$ with role identifiers, and U_A and U_B with user identifiers. If U_A is activated with $(\Pi, \mathcal{I}, U_A, U_B)$, then U_A is called the session *initiator*. If U_B is activated with $(\Pi, \mathcal{R}, U_B, U_A, X_A)$, then U_B is called the session *responder*. The initiator U_A outputs X_A, then may receive an incoming message of the forms $(\Pi, \mathcal{I}, U_A, U_B, X_A, X_B)$ from the responder U_B, U_A then computes the session key SK if U_A received the message. On the contrary, the responder U_B outputs X_B, and computes the session key SK.

If U_A is the initiator of a session, the session is identified by $\mathsf{sid} = (\Pi, \mathcal{I}, U_A, U_B, X_A)$ or $\mathsf{sid} = (\Pi, \mathcal{I}, U_A, U_B, X_A, X_B)$. If U_B is the responder of a session, the session is identified by $\mathsf{sid} = (\Pi, \mathcal{R}, U_B, U_A, X_A, X_B)$. We say that U_A is the *owner* of session sid, if the third coordinate of session sid is U_A. We say that U_A is the *peer* of session sid, if the fourth coordinate of session sid is U_A. We say that a session is *completed* if its owner computes the session key. The *matching session* of $(\Pi, \mathcal{I}, U_A, U_B, X_A, X_B)$ is session $(\Pi, \mathcal{R}, U_B, U_A, X_A, X_B)$ and vice versa.

Adversary. The adversary $\mathcal{A}$, which is modeled as a probabilistic polynomial-time Turing machine, controls all communications between parties including session activation by performing the following adversary query.

- Send(message): The message has one of the following forms: $(\Pi, \mathcal{I}, U_A, U_B)$, $(\Pi, \mathcal{R}, U_B, U_A, X_A)$, or $(\Pi, \mathcal{I}, U_A, U_B, X_A, X_B)$. The adversary $\mathcal{A}$ obtains the response from the party.

To capture leakage of secret information, the adversary $\mathcal{A}$ is allowed to issue the following queries.

- $\mathsf{SessionKeyReveal}(\mathsf{sid})$: The adversary $\mathcal{A}$ obtains the session key SK for the session sid if the session is completed.
- $\mathsf{SessionStateReveal}(\mathsf{sid})$: The adversary $\mathcal{A}$ obtains the session state of the owner of session sid if the session is not completed (the session key is not established yet). The session state includes all ephemeral secret keys and intermediate computation results except for immediately erased information but does not include the static secret key.
- $\mathsf{Corrupt}(U_i)$: This query allows the adversary $\mathcal{A}$ to obtain all information of the party U_i. If a party is corrupted by a $\mathsf{Corrupt}(U_i, S_i)$ query issued by the adversary $\mathcal{A}$, then we call the party U_i *dishonest*. If not, we call the party *honest*.

Freshness. For the security definition, we need the notion of freshness.

Definition 1 (Freshness). *Let* $\mathsf{sid}^* = (\Pi, \mathcal{I}, U_A, U_B, X_A, X_B)$ *or* $(\Pi, \mathcal{R}, U_A, U_B, X_B, X_A)$ *be a completed session between honest users* U_A *and* U_B*. If the matching session exists, then let* $\overline{\mathsf{sid}^*}$ *be the matching session of* sid^**. We say session* sid^* *is* fresh *if none of the following conditions hold:*

1. *The adversary* $\mathcal{A}$ *issues* $\mathsf{SessionKeyReveal}(\mathsf{sid}^*)$*, or* $\mathsf{SessionKeyReveal}(\overline{\mathsf{sid}^*})$ *if* $\overline{\mathsf{sid}^*}$ *exists,*
2. $\overline{\mathsf{sid}^*}$ *exists and the adversary* $\mathcal{A}$ *makes either of the following queries*
   - $\mathsf{SessionStateReveal}(\mathsf{sid}^*)$ *or* $\mathsf{SessionStateReveal}(\overline{\mathsf{sid}^*})$*,*
3. $\overline{\mathsf{sid}^*}$ *does not exist and the adversary* $\mathcal{A}$ *makes the following query*
   - $\mathsf{SessionStateReveal}(\mathsf{sid}^*)$*.*

Security Experiment. For the security definition, we consider the following security experiment. Initially, the adversary $\mathcal{A}$ is given a set of honest users and makes any sequence of the queries described above. During the experiment, the adversary $\mathcal{A}$ makes the following query.

- $\mathsf{Test}(\mathsf{sid}^*)$: Here, sid^* must be a fresh session. Select random bit $b \in_U \{0, 1\}$, and return the session key held by sid^* if $b = 0$, and return a random key if $b = 1$.

The experiment continues until the adversary $\mathcal{A}$ makes a guess b'. The adversary $\mathcal{A}$ *wins* the game if the test session sid^* is still fresh and if the guess of the adversary $\mathcal{A}$ is correct, i.e., $b' = b$. The advantage of the adversary $\mathcal{A}$ in the AKE experiment with the PKI-based AKE protocol Π is defined as

$$\mathrm{Adv}_{\Pi}^{\mathrm{AKE}}(\mathcal{A}) = \Pr[\mathcal{A}\ wins] - \frac{1}{2}.$$

We define the security as follows.

Definition 2 (Security). *We say that a PKI-based AKE protocol Π is* secure in the CK$^+$ model *if the following conditions hold:*

1. *If two honest parties complete matching sessions, then, except with negligible probability, they both compute the same session key.*
2. *For any PPT bounded adversary $\mathcal{A}$, $\mathrm{Adv}_{\Pi}^{\mathrm{AKE}}(\mathcal{A})$ is negligible in security parameter κ for the test session sid^*,*
   (a) *if $\overline{\mathsf{sid}^*}$ does not exist, and the static secret key of the owner of sid^* is given to $\mathcal{A}$.*
   (b) *if $\overline{\mathsf{sid}^*}$ does not exist, and the ephemeral secret key of sid^* is given to $\mathcal{A}$.*
   (c) *if $\overline{\mathsf{sid}^*}$ exists, and the static secret key of the owner of sid^* and the ephemeral secret key of $\overline{\mathsf{sid}^*}$ are given to $\mathcal{A}$.*
   (d) *if $\overline{\mathsf{sid}^*}$ exists, and the ephemeral secret key of sid^* and the ephemeral secret key of $\overline{\mathsf{sid}^*}$ are given to $\mathcal{A}$.*
   (e) *if $\overline{\mathsf{sid}^*}$ exists, and the static secret key of the owner of sid^* and the static secret key of the peer of sid^* are given to $\mathcal{A}$.*
   (f) *if $\overline{\mathsf{sid}^*}$ exists, and the ephemeral secret key of sid^* and the static secret key of the peer of sid^* are given to $\mathcal{A}$.*

Note that the items 2.a, 2.c, and 2.f correspond to resistance to KCI, item 2.e corresponds to wPFS, and items 2.b and 2.d correspond to resistance to MEX.

3 Generic AKE Construction from KEM without Random Oracles

In this section, we propose a generic construction of CK$^+$-secure AKE from KEM.

3.1 Preliminaries

Security Notions of KEM Schemes. Here, we recall the definition of IND-CCA and IND-CPA security for KEM, and min-entropy of KEM keys as follows.

Definition 3 (Model for KEM Schemes). *A KEM scheme consists of the following 3-tuple* (KeyGen, EnCap, DeCap)*:*

$(ek, dk) \leftarrow \mathsf{KeyGen}(1^\kappa, r_g)$ *: a key generation algorithm which on inputs 1^κ and $r_g \in \mathcal{RS}_G$, where κ is the security parameter and $\mathcal{RS}_G$ is a randomness space, outputs a pair of keys (ek, dk).*

$(K, CT) \leftarrow \mathsf{EnCap}_{ek}(r_e)$: *an encryption algorithm which takes as inputs encapsulation key* ek *and* $r_e \in \mathcal{RS}_E$*, outputs session key* $K \in \mathcal{KS}$ *and ciphertext* $CT \in \mathcal{CS}$*, where* $\mathcal{RS}_E$ *is a randomness space,* $\mathcal{KS}$ *is a session key space, and* $\mathcal{CS}$ *is a ciphertext space.*

$K \leftarrow \mathsf{DeCap}_{dk}(CT)$: *a decryption algorithm which takes as inputs decapsulation key* dk *and ciphertext* $CT \in \mathcal{CS}$*, and outputs session key* $K \in \mathcal{KS}$*.*

Definition 4 (IND-CCA and IND-CPA Security for KEM). *A KEM scheme is IND-CCA-secure for KEM if the following property holds for security parameter* κ*; For any PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*,* $\mathbf{Adv}^{\mathrm{ind-cca}} = |\Pr[r_g \leftarrow \mathcal{RS}_G; (ek, dk) \leftarrow \mathsf{KeyGen}(1^\kappa, r_g); (state) \leftarrow \mathcal{A}_1^{\mathcal{DO}(dk,\cdot)}(ek); b \leftarrow \{0,1\}; r_e \leftarrow \mathcal{RS}_E; (K_0^*, CT_0^*) \leftarrow \mathsf{EnCap}_{ek}(r_e); K_1^* \leftarrow \mathcal{K}; b' \leftarrow \mathcal{A}_2^{\mathcal{DO}(dk,\cdot)}(ek, (K_b^*, CT_0^*), state); b' = b] - 1/2| \leq negl$*, where* $\mathcal{DO}$ *is the decryption oracle,* $\mathcal{K}$ *is the space of session key and state is state information that* $\mathcal{A}$ *wants to preserve from* $\mathcal{A}_1$ *to* $\mathcal{A}_2$*.* $\mathcal{A}$ *cannot submit the ciphertext* $CT = CT_0^*$ *to* $\mathcal{DO}$*.*

We say a KEM scheme is IND-CPA-secure for KEM if $\mathcal{A}$ *does not access* $\mathcal{DO}$*.*

Definition 5 (Min-Entropy of KEM Key). *A KEM scheme is* k*-min-entropy KEM if for any* ek*, for distribution* $D_{\mathcal{KS}}$ *of variable* K *defined by* $(K, CT) \leftarrow \mathsf{EnCap}_{ek}(r_e)$ *and random* $r_e \in \mathcal{RS}_E$*,* $H_\infty(D_{\mathcal{KS}}) \geq k$ *holds, where* H_∞ *denotes min-entropy.*

Security Notions of Randomness Extractor and Pseudo-Random Function. Let $Ext : S \times X \rightarrow Y$ be a function with finite seed space S, finite domain X, and finite range Y.

Definition 6 (Strong Randomness Extractor). *We say that function Ext is a strong randomness extractor, if for any distribution* D_X *over* X *with* $H_\infty(D_X) \geq k$*,* $\Delta((U_S, Ext(U_S, D_X)), (U_S, U_Y)) \leq negl$ *holds, where both* U_S *in* $(U_S, Ext(U_S, D_X))$ *denotes the same random variable,* Δ *denotes statistical distance,* U_S, U_X, U_Y *denotes uniform distribution over* S, X, Y *respectively,* $|X| = n \geq k$*,* $|Y| = k$*, and* $|S| = d$*.*

Let κ be a security parameter and $\mathsf{F} = \{F_\kappa : Dom_\kappa \times \mathcal{FS}_\kappa \rightarrow Rng_\kappa\}_\kappa$ be a function family with a family of domains $\{\mathsf{Dom}_\kappa\}_\kappa$, a family of key spaces $\{\mathsf{FS}_\kappa\}_\kappa$ and a family of ranges $\{\mathsf{Rng}_\kappa\}_\kappa$.

Definition 7 (Pseudo-Random Function). *We say that function family* $\mathsf{F} = \{F_\kappa\}_\kappa$ *is the PRF family, if for any PPT distinguisher* $\mathcal{D}$*,* $\mathbf{Adv}^{\mathrm{prf}} = |\Pr[\mathcal{D}^{F_\kappa(\cdot)} \rightarrow 1] - \Pr[\mathcal{D}^{RF_\kappa(\cdot)} \rightarrow 1]| \leq negl$*, where* $RF_\kappa : Dom_\kappa \rightarrow Rng_\kappa$ *is a truly random function.*

3.2 Construction

Our construction (GC) is based on an IND-CCA secure KEM, an IND-CPA secure KEM, PRFs, and strong randomness extractors. While the requirements

for the underlying building blocks are not stronger than those for the previous generic construction [13,14], GC achieves stronger security (i.e., CK^+ security) without random oracles.

Necessity of Min-Entropy of KEM Key. In the BCGNP construction, a KEM scheme is only assumed to be IND-CCA. However, it is not enough to prove the security. Both parties derive the session key by applying decapsulated KEM keys to a strong randomness extractor before applying them to PRFs. This extractor guarantees to output a statistically indistinguishable value from a uniform randomly chosen element from the same space. It requires as input a seed and a KEM key with min-entropy κ, where κ is a security parameter. IND-CCA states that no PPT adversary can distinguish the KEM key from a random element, but this is "only" computational indistinguishability. What we need is statistical indistinguishability. Thus, we must also assume that min-entropy of the KEM key is equal or larger than κ. This property is not very strong; almost all IND-CCA secure schemes satisfy it. We will discuss later about this property of concrete KEM schemes.

Design Principle. The main ideas to achieve CK^+ security are to use the *twisted PRF* trick and *session-specific* key generation.

First, we have to consider resistance to MEX. The most awkward pattern of MEX is the disclosure of ephemeral secret keys of the initiator and the responder. If we use KEM naturally, all randomness used to generate ciphertexts is leaked as ephemeral secret keys; thus, the adversary can obtain encrypted messages without knowing secret keys. Hence, we have to avoid using ephemeral secret keys as randomness of KEM directly. A possible solution is to generate randomness from the static secret key as well as the ephemeral secret key by using a technique such as the ordinary NAXOS trick [4]. Though this trick leads to security against leakage of ephemeral secret keys, the trick must apply an RO to the concatenation of the static and ephemeral secret keys, and it uses the output as a quasi-ephemeral secret key. It is unsuitable for our purpose to construct secure protocols in the StdM. Thus, we use a trick to achieve the same properties as the NAXOS trick but without ROs. We call it the twisted PRF trick.[3] This trick uses two PRFs (F, F') with reversing keys; we choose two ephemeral keys (r, r') and compute $F_\sigma(r) \oplus F'_{r'}(\sigma)$, where σ is the static secret key. The twisted PRF trick is especially effective in the following two scenarios: leakage of both ephemeral secret keys of the initiator and the responder, and leakage of the static secret key of the initiator and the ephemeral secret key of the responder (i.e., corresponding to KCI). If (r, r') is leaked, $F_\sigma(r)$ cannot be computed without knowing σ. Similarly, if σ is leaked, $F'_{r'}(\sigma)$ cannot be computed without knowing r'. In our KEM-based generic construction, the output of the twisted PRF is used as randomness for the encapsulation algorithm.

Next, we have to consider the scenario in which static secret keys are leaked as the attack scenario in wPFS. We cannot achieve a CK^+ secure scheme by

[3] A similar trick is used in the Okamoto AKE scheme [27].

any combination of KEMs using static secret keys as decapsulation keys against leakage of both static secret keys of the initiator and the responder because an adversary can obtain all information the parties can obtain by using static secret keys. Our solution is to generate session-specific decapsulation and encapsulation keys. The initiator sends the temporary encapsulation key to the responder, the responder encapsulates a KEM key with the temporary encapsulation key, and the initiator decapsulates the ciphertext. Since this procedure does not depend on the static secret keys, the KEM key is hidden even if both static secret keys of the initiator and the responder are leaked. Note that security of KEM for temporary use only requires IND-CPA. The session-specific key generation is effective for achieving wPFS.

As the BCGNP construction [13,14], we use IND-CCA secure KEM schemes to exchange ciphertexts. CCA security is necessary to simulate SessionStateReveal queries in the security proof. When we prove security in the case where ephemeral secret keys are leaked, the simulator needs to embed the challenge ciphertext in the ephemeral public key in the test session. Then, the static secret key to decrypt the challenge ciphertext is not known; that is, the simulator must respond to the SessionStateReveal query for a session owned by the same parties as the test session without knowing the static secret key. Hence, the simulator needs the power of the decryption oracle to obtain intermediate computation results corresponding to the SessionStateReveal query.

Generic Construction GC. The protocol of GC from KEMs (KeyGen, EnCap, DeCap) and (wKeyGen, wEnCap, wDeCap) is as follows.

Public Parameters. Let κ be the security parameter, $F : \{0,1\}^* \times \mathcal{FS} \to \mathcal{RS}_E$, $F' : \{0,1\}^* \times \mathcal{FS} \to \mathcal{RS}_E$, and $G : \{0,1\}^* \times \mathcal{FS} \to \{0,1\}^\kappa$ be pseudo-random functions, where $\mathcal{FS}$ is the key space of PRFs ($|\mathcal{FS}| = \kappa$), $\mathcal{RS}_E$ is the randomness space of encapsulation algorithms, and $\mathcal{RS}_G$ is the randomness space of key generation algorithms, and let $Ext : \mathcal{SS} \times \mathcal{KS} \to \mathcal{FS}$ be a strong randomness extractor with randomly chosen seed $s \in \mathcal{SS}$, where $\mathcal{SS}$ is the seed space and $\mathcal{KS}$ is the KEM key space. These are provided as some of the public parameters.

Secret and Public Keys. Party U_I randomly selects $\sigma_I \in \mathcal{FS}$ and $r_I \in \mathcal{RS}_G$, and runs the key generation algorithm $(ek_{I,1}, dk_{I,1}) \leftarrow \mathsf{KeyGen}(1^\kappa, r_I)$, where $\mathcal{RS}_G$ is the randomness space of KeyGen. Party U_I's static secret and public keys are $((dk_{I,1}, \sigma_I), ek_{I,1})$.

Key Exchange. Party U_A with secret and public keys $((dk_{A,1}, \sigma_A), ek_{A,1})$, and who is the initiator, and party U_B with secret and public keys $((dk_{B,1}, \sigma_B), ek_{B,1})$, and who is the responder, perform the following two-pass key exchange protocol.

1. Party U_A randomly chooses ephemeral secret keys $r_{A,1}, r'_{A,1} \in \mathcal{FS}$ and $r_{A,2} \in \mathcal{RS}_G$. Party U_A computes $(CT_{A,1}, K_{A,1}) \leftarrow \mathsf{EnCap}_{ek_{B,1}}(F_{\sigma_A}(r_{A,1}) \oplus F'_{r'_{A,1}}(\sigma_A))$ and $(ek_{A,2}, dk_{A,2}) \leftarrow \mathsf{wKeyGen}(1^\kappa, r_{A,2})$ and sends $(U_A, U_B, CT_{A,1}, ek_{A,2})$ to party U_B.

2. Upon receiving $(U_A, U_B, CT_{A,1}, ek_{A,2})$, party U_B chooses the ephemeral secret keys $r_{B,1}, r'_{B,1} \in \mathcal{FS}$ and $r_{B,2} \in \mathcal{RS}_E$, computes $(CT_{B,1}, K_{B,1}) \leftarrow \mathsf{EnCap}_{ek_{A,1}}(F_{\sigma_B}(r_{B,1}) \oplus F'_{r'_{B,1}}(\sigma_B))$ and $(CT_{B,2}, K_{B,2}) \leftarrow \mathsf{wEnCap}_{ek_{A,2}}(r_{B,2})$, and sends $(U_A, U_B, CT_{B,1}, CT_{B,2})$ to party U_A.
Party U_B computes $K_{A,1} \leftarrow \mathsf{DeCap}_{dk_{B,1}}(CT_{A,1})$, $K'_1 \leftarrow Ext(s, K_{A,1})$, $K'_2 \leftarrow Ext(s, K_{B,1})$ and $K'_3 \leftarrow Ext(s, K_{B,2})$, sets the session transcript $\mathsf{ST} = (U_A, U_B, ek_{A,1}, ek_{B,1}, CT_{A,1}, ek_{A,2}, CT_{B,1}, CT_{B,2})$ and the session key $SK = G_{K'_1}(\mathsf{ST}) \oplus G_{K'_2}(\mathsf{ST}) \oplus G_{K'_3}(\mathsf{ST})$, completes the session, and erases all session states.
3. Upon receiving $(U_A, U_B, CT_{B,1}, CT_{B,2})$, party U_A computes $K_{B,1} \leftarrow \mathsf{DeCap}_{dk_{A,1}}(CT_{B,1})$, $K_{B,2} \leftarrow \mathsf{wDeCap}_{dk_{A,2}}(CT_{B,2})$, $K'_1 \leftarrow Ext(s, K_{A,1})$, $K'_2 \leftarrow Ext(s, K_{B,1})$ and $K'_3 \leftarrow Ext(s, K_{B,2})$, sets the session transcript $\mathsf{ST} = (U_A, U_B, ek_{A,1}, ek_{B,1}, CT_{A,1}, ek_{A,2}, CT_{B,1}, CT_{B,2})$ and the session key $SK = G_{K'_1}(\mathsf{ST}) \oplus G_{K'_2}(\mathsf{ST}) \oplus G_{K'_3}(\mathsf{ST})$, completes the session, and erases all session states.

The session state of a session owned by U_A contains ephemeral secret keys $(r_{A,1}, r'_{A,1}, r_{A,2})$, KEM keys $(K_{A,1}, K_{B,1}, K_{B,2})$, outputs of the extractor (K'_1, K'_2, K'_3) and outputs of PRFs $(F_{\sigma_A}(r_{A,1}), F'_{r'_{A,1}}(\sigma_A), G_{K'_1}(\mathsf{ST}), G_{K'_2}(\mathsf{ST}), G_{K'_3}(\mathsf{ST}))$. Similarly, the session state of a session owned by U_B contains ephemeral secret keys $(r_{B,1}, r'_{B,1}, r_{B,2})$, decapsulated KEM keys $(K_{A,1}, K_{B,1}, K_{B,2})$, outputs of the extractor (K'_1, K'_2, K'_3) and outputs of PRFs $(F_{\sigma_B}(r_{B,1}), F'_{r'_{B,1}}(\sigma_B), G_{K'_1}(\mathsf{ST}), G_{K'_2}(\mathsf{ST}), G_{K'_3}(\mathsf{ST}))$.

Remark 1. Obviously, we can use arbitrary combinations of KEM schemes in the generic construction. This means that each party can rely on a different assumption from the peer. Since our construction does not contain any direct operation between derivatives of KEM schemes, it is no problem that randomness spaces, public keys, or ciphertext are distinct from each other.

Security. We show the following theorem.

Theorem 1. *If KEM* $(\mathsf{KeyGen}, \mathsf{EnCap}, \mathsf{DeCap})$ *is IND-CCA secure and is* κ*-min-entropy KEM, KEM* $(\mathsf{wKeyGen}, \mathsf{wEnCap}, \mathsf{wDeCap})$ *is IND-CPA secure and is* κ*-min-entropy KEM,* F, F', G *are PRFs, and Ext is a strong randomness extractor, then AKE scheme* GC *is* CK^+*-secure.*

Due to space limitations we defer the proof of Theorem 1 to the full version. Here, we give an overview of the security proof.

We have to consider the following four leakage patterns in the CK^+ security model (matching cases):

2-(c) the static secret key of the initiator and the ephemeral secret key of the responder
2-(d) both ephemeral secret keys
2-(e) both static secret keys

2-(f) the ephemeral secret key of the initiator and the static secret key of the responder

In case 2-(c), $K_{A,1}$ is protected by the security of $CT_{A,1}$ because $r'_{A,1}$ is not leaked; therefore, $F'_{r'_{A,1}}(\sigma_A)$ is hidden and $dk_{B,1}$ is not leaked. In case 2-(d), $K_{A,1}$ and $K_{B,1}$ are protected by the security of $CT_{A,1}$ and $CT_{B,1}$ because σ_A and σ_B are not leaked; therefore, $F_{\sigma_A}(r_{A,1})$ and $F_{\sigma_B}(r_{B,1})$ are hidden and $dk_{A,1}$ and $dk_{B,1}$ are not leaked. In case 2-(e), $K_{B,2}$ is protected by the security of $CT_{B,2}$ because $dk_{A,2}$ and $r_{B,2}$ are not leaked. In case 2-(f), $K_{B,1}$ is protected by the security of $CT_{B,1}$ because $r'_{B,1}$ is not leaked; therefore, $F'_{r'_{B,1}}(\sigma_B)$ is hidden and $dk_{A,1}$ is not leaked. Then, we transform the CK^+ security game since the session key in the test session is randomly distributed. First, we change part of the doubled PRF in the test session into a random function because the key of part of the doubled PRF is hidden from the adversary; therefore, the randomness of the protected KEM can be randomly distributed. Second, we change the protected KEM key into a random key for each pattern; therefore, the input of Ext is randomly distributed and has sufficient min-entropy. Third, we change the output of Ext into randomly chosen values. Finally, we change one of the PRFs (corresponding to the protected KEM) into a random function. Therefore, the session key in the test session is randomly distributed; thus, there is no advantage to the adversary. We can show a similar proof in non-matching cases.

4 Instantiations

4.1 Diffie-Hellman-Based

We can achieve various AKE schemes as concrete instantiations based on the hardness of the DH problem and its variants. These are derived from the generic construction GC in Section 3. For example, we can apply efficient IND-CCA KEM schemes to GC from the decisional DH [28,29] (DDH), computational DH [30,31], hashed DH [32] and bilinear DH assumptions [33].

We can easily show that these schemes have κ-min-entropy KEM keys. The KEM part of the Cramer-Shoup PKE consists of $g_1^{zr} \in G$, where G is a finite cyclic group of order prime p, g_1^z is part of ek, and r is uniformly chosen randomness, and $|r|$ is 2κ. Thus, g_1^{zr} has min-entropy larger than κ. Similarly, other schemes also have κ-min-entropy KEM keys.

The significant advantage of our instantiations in the StdM is reasonable assumption. First, HMQV satisfies the same security model as our construction. However, it requires the KEA1 assumption and relies on ROs. Since it has been criticised, in particular because the KEA1 assumption does not appear to be "efficiently falsifiable" as Naor put it [12], this assumption is quite undesirable. Also, it was shown that there exist some protocols that are secure in the ROM but are insecure if ROs are replaced by any specific function [34]. A disadvantage of our construction to HMQV is that HMQV is a one-round protocol but our scheme is not. One-round protocols mean that the initiator and the responder

Table 3. Comparison of previous DH-based schemes and an instantiation of our scheme

	Model	Resource	Assumption	Computation (#[multi,regular]-exp)	Communication complexity	
[3]	CK^+	ROM	gap DH & KEA1	[2, 2]	$2\|p\|$	512
[27]	eCK	StdM	DDH & πPRF	[6, 6]	$9\|p\|$	2304
[14]	CK & KCI	StdM	DDH	[4, 8]	$6\|p\|$	1536
Ours	CK^+	StdM	DDH	[4, 12]	$8\|p\|$	2048

For concreteness the expected ciphertext overhead for a 128-bit implementation is also given. Note that computational costs are estimated without any pre-computation technique.

can send their messages independently and simultaneously. Conversely, in our scheme, the responder must wait to receive the message from the initiator. Next, the AKE scheme by Okamoto [27] is secure in the StdM. However, it is not proved in the CK^+ model and needs to assume existence of πPRF. πPRF is a stronger primitive than ordinary PRF, and it is not known how to construct πPRF concretely. On the contrary, our instantiations only require the standard notions of KEM and pseudo-random function security. Moreover, the BCGNP construction [13,14] is secure in the StdM with standard assumption. However, the security is not proved in the CK^+ model.[4] Thus, DH-based AKE schemes from GC are first CK^+ secure schemes in the StdM with standard assumptions.

For example, our scheme can be instantiated with the Cramer-Shoup KEM [35] as an IND-CCA KEM, and with the ElGamal KEM as an IND-CPA KEM under the DDH assumption. Communication complexity (for two parties) of this instantiation is $8|p|$, where $|p|$ is the length of a group element. Computational complexity (for two parties) of this instantiation is 4 multi-exponentiations and 12 regular exponentiations (all symmetric operations such as hash function/KDF/PRF and multiplications are ignored). We show a comparison between this instantiation and previous schemes in Table 3.

4.2 Factoring-Based

We can achieve several new AKE protocols as concrete instantiations based on the hardness of integer factorization and its variants such as the RSA problem.

Some instantiations in the StdM are based on the hardness of the integer factorization problem. By applying the Hofheinz-Kiltz PKE [16] and the Mei-Li-Lu-Jia PKE [17], which are IND-CCA secure in the StdM under the factoring assumption to GC, we can obtain first CK^+ secure AKE protocols in the StdM

[4] The BCGNP construction with an additional exchange of a DH value (called Protocol 2 in [13,14]) can be proved in the CK model, and it satisfies wPFS and resistance to KCI. We can extend the security of Protocol 2 to the CK^+ security with the twisted PRF trick. If IND-CPA KEM in GC is instantiated with the ElGamal KEM, our scheme is the same as Protocol 2 with the twisted PRF trick. Thus, our scheme can also be seen as a generalization of the BCGNP construction.

under the integer factorization assumption. Also, we have other instantiations based on the hardness of RSA inversion. By applying the Chevallier-Mames-Joye PKE [36] and the Kiltz-Mohassel-O'Neill PKE [37], which are IND-CCA secure in the StdM under the instance-independent RSA assumption to GC, we can obtain first CK^+ secure AKE protocols in the StdM under the RSA-type assumption.

We can regard a message in PKE as a KEM key when the message space is larger than κ and messages are uniformly chosen randomness. In this case, it is obvious that such a KEM scheme has κ-min-entropy KEM keys.

4.3 Code-Based

We can achieve new AKE protocols as concrete instantiations based on code-based problems.

For the AKE protocol in the StdM, we can apply Dowsley et al.'s PKE [19] that is IND-CCA secure in the StdM under the McEliece and LPN assumptions to GC. (See Ref. [19] for definitions of these assumptions.) This is the first CK^+ secure AKE protocol without ROs based on a code-based problem.

As for factoring-based PKE, code-based PKE schemes also have κ-min-entropy KEM keys when the message space is larger than κ and messages are uniformly chosen randomness.

Remark 2. Bernstein et al. [38] estimated the size of a public key of the original McEliece at about 2 Mbits for 128-bit security. If we employ "wild" McEliece by Bernstein et al. [39] rather than the original McEliece PKE, the size of the public key is reduced to 750K bits. Our generic construction contains the public key of the KEM from the temporary key generation in the first round message. If the randomized McEliece PKE by Nojima et al. [40] is employed as the IND-CPA secure KEM, which is IND-CPA secure and requires the same size for the public key as the original, the communication complexity of the resultant AKE scheme is high. However, the way to construct an efficient and CK^+ secure AKE scheme from codes is an open problem.

4.4 Lattice-Based

We also achieve new concrete AKE protocols based on the worst-case hardness of the (ring-)LWE problems derived from our generic constructions.

PKE schemes [20,21,22,23,24,25,26,41] which are IND-CCA secure in the StdM are easily converted into IND-CCA secure KEM schemes. Also, PRFs are obtained from one-way functions [42,43,44,45] and directly constructed from the (ring-)LWE assumptions with sub-exponential parameters [46]. Thus, by applying these building blocks to GC, we can obtain first CK^+ secure AKE protocols in the StdM under the (ring-)LWE assumption. Unfortunately, the obtained AKE protocols are still theoretical since these PKE schemes require huge keys, say, of the quadratic or cubic order of the security parameter, and thus, an efficient and

direct construction of PRFs from the (ring-)LWE assumption with polynomial parameters has not yet been achieved.

As for factoring-based PKE, lattice-based PKE schemes also have κ-min-entropy KEM keys when the message space is larger than κ and messages are uniformly chosen randomness.

References

1. Bellare, M., Rogaway, P.: Entity Authentication and Key Distribution. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 232–249. Springer, Heidelberg (1994)
2. Canetti, R., Krawczyk, H.: Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 453–474. Springer, Heidelberg (2001)
3. Krawczyk, H.: HMQV: A High-Performance Secure Diffie-Hellman Protocol. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 546–566. Springer, Heidelberg (2005)
4. LaMacchia, B.A., Lauter, K., Mityagin, A.: Stronger Security of Authenticated Key Exchange. In: Susilo, W., Liu, J.K., Mu, Y. (eds.) ProvSec 2007. LNCS, vol. 4784, pp. 1–16. Springer, Heidelberg (2007)
5. Sarr, A.P., Elbaz-Vincent, P., Bajard, J.-C.: A New Security Model for Authenticated Key Agreement. In: Garay, J.A., De Prisco, R. (eds.) SCN 2010. LNCS, vol. 6280, pp. 219–234. Springer, Heidelberg (2010)
6. Jeong, I.R., Katz, J., Lee, D.-H.: One-Round Protocols for Two-Party Authenticated Key Exchange. In: Jakobsson, M., Yung, M., Zhou, J. (eds.) ACNS 2004. LNCS, vol. 3089, pp. 220–232. Springer, Heidelberg (2004)
7. Gennaro, R., Krawczyk, H., Rabin, T.: Okamoto-Tanaka Revisited: Fully Authenticated Diffie-Hellman with Minimal Overhead. In: Zhou, J., Yung, M. (eds.) ACNS 2010. LNCS, vol. 6123, pp. 309–328. Springer, Heidelberg (2010)
8. Cremers, C.J.F., Feltz, M.: One-round Strongly Secure Key Exchange with Perfect Forward Secrecy and Deniability. In: Cryptology ePrint Archive: 2011/300 (2011)
9. Cremers, C.J.F.: Session-state Reveal Is Stronger Than Ephemeral Key Reveal: Attacking the NAXOS Authenticated Key Exchange Protocol. In: Abdalla, M., Pointcheval, D., Fouque, P.-A., Vergnaud, D. (eds.) ACNS 2009. LNCS, vol. 5536, pp. 20–33. Springer, Heidelberg (2009)
10. Cremers, C.J.F.: Examining Indistinguishability-Based Security Models for Key Exchange Protocols: The case of CK, CK-HMQV, and eCK. In: ASIACCS 2011, pp. 80–91 (2011)
11. Damgård, I.: Towards Practical Public Key Systems Secure against Chosen Ciphertext Attacks. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 445–456. Springer, Heidelberg (1992)
12. Naor, M.: On Cryptographic Assumptions and Challenges. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 96–109. Springer, Heidelberg (2003)
13. Boyd, C., Cliff, Y., Gonzalez Nieto, J.M., Paterson, K.G.: Efficient One-Round Key Exchange in the Standard Model. In: Mu, Y., Susilo, W., Seberry, J. (eds.) ACISP 2008. LNCS, vol. 5107, pp. 69–83. Springer, Heidelberg (2008)
14. Boyd, C., Cliff, Y., González Nieto, J.M., Paterson, K.G.: One-round key exchange in the standard model. In: IJACT, vol. 1(3), pp. 181–199 (2009)

15. Gorantla, M.C., Boyd, C., González Nieto, J.M., Manulis, M.: Generic One Round Group Key Exchange in the Standard Model. In: Lee, D., Hong, S. (eds.) ICISC 2009. LNCS, vol. 5984, pp. 1–15. Springer, Heidelberg (2010)
16. Hofheinz, D., Kiltz, E.: Practical Chosen Ciphertext Secure Encryption from Factoring. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 313–332. Springer, Heidelberg (2009)
17. Mei, Q., Li, B., Lu, X., Jia, D.: Chosen Ciphertext Secure Encryption under Factoring Assumption Revisited. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 210–227. Springer, Heidelberg (2011)
18. McEliece, R.J.: A Public-Key Cryptosystem Based on Algebraic Coding Theory. In: Deep Space Network progress Report (1978)
19. Dowsley, R., Müller-Quade, J., Nascimento, A.C.A.: A CCA2 Secure Public Key Encryption Scheme Based on the McEliece Assumptions in the Standard Model. In: Fischlin, M. (ed.) CT-RSA 2009. LNCS, vol. 5473, pp. 240–251. Springer, Heidelberg (2009)
20. Peikert, C., Waters, B.: Lossy Trapdoor Functions and Their Applications. In: STOC 2008, pp. 187–196 (2008)
21. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: STOC 2009, pp. 333–342 (2009)
22. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai Trees, or How to Delegate a Lattice Basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010)
23. Agrawal, S., Boneh, D., Boyen, X.: Efficient Lattice (H)IBE in the Standard Model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010)
24. Agrawal, S., Boneh, D., Boyen, X.: Lattice Basis Delegation in Fixed Dimension and Shorter-Ciphertext Hierarchical IBE. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 98–115. Springer, Heidelberg (2010)
25. Stehlé, D., Steinfeld, R., Tanaka, K., Xagawa, K.: Efficient Public Key Encryption Based on Ideal Lattices. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 617–635. Springer, Heidelberg (2009)
26. Lyubashevsky, V., Peikert, C., Regev, O.: On Ideal Lattices and Learning with Errors over Rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (2010)
27. Okamoto, T.: Authenticated Key Exchange and Key Encapsulation in the Standard Model. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 474–484. Springer, Heidelberg (2007)
28. Cramer, R., Shoup, V.: A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
29. Kurosawa, K., Desmedt, Y.: A New Paradigm of Hybrid Encryption Scheme. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 426–442. Springer, Heidelberg (2004)
30. Hanaoka, G., Kurosawa, K.: Efficient Chosen Ciphertext Secure Public Key Encryption under the Computational Diffie-Hellman Assumption. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 308–325. Springer, Heidelberg (2008)
31. Haralambiev, K., Jager, T., Kiltz, E., Shoup, V.: Simple and Efficient Public-Key Encryption from Computational Diffie-Hellman in the Standard Model. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 1–18. Springer, Heidelberg (2010)

32. Kiltz, E.: Chosen-Ciphertext Secure Key-Encapsulation Based on Gap Hashed Diffie-Hellman. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 282–297. Springer, Heidelberg (2007)
33. Boyen, X., Mei, Q., Waters, B.: Direct chosen ciphertext security from identity-based techniques. In: ACM Conference on Computer and Communications Security 2005, pp. 320–329 (2005)
34. Canetti, R., Goldreich, O., Halevi, S.: The Random Oracle Methodology, Revisited (Preliminary Version). In: STOC 1998, pp. 131–140 (1998)
35. Cramer, R., Shoup, V.: Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. SIAM Journal on Computing 33, 167–226 (2004)
36. Chevallier-Mames, B., Joye, M.: Chosen-Ciphertext Secure RSA-Type Cryptosystems. In: Pieprzyk, J., Zhang, F. (eds.) ProvSec 2009. LNCS, vol. 5848, pp. 32–46. Springer, Heidelberg (2009)
37. Kiltz, E., Mohassel, P., O'Neill, A.: Adaptive Trapdoor Functions and Chosen-Ciphertext Security. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 673–692. Springer, Heidelberg (2010)
38. Bernstein, D.J., Lange, T., Peters, C.: Smaller Decoding Exponents: Ball-Collision Decoding. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 743–760. Springer, Heidelberg (2011)
39. Bernstein, D.J., Lange, T., Peters, C.: Wild McEliece. In: Biryukov, A., Gong, G., Stinson, D.R. (eds.) SAC 2010. LNCS, vol. 6544, pp. 143–158. Springer, Heidelberg (2011)
40. Nojima, R., Imai, H., Kobara, K., Morozov, K.: Semantic security for the McEliece cryptosystem without random oracles. Designs, Codes and Cryptography 49(1-3), 289–305 (2008)
41. Micciancio, D., Peikert, C.: Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012)
42. Ajtai, M.: Generating Hard Instances of Lattice Problems (Extended Abstract). In: STOC 1996, pp. 99–108 (1996); See also ECCC TR96-007
43. Micciancio, D., Regev, O.: Worst-Case to Average-Case Reductions Based on Gaussian Measures. SIAM Journal on Computing 37(1), 267–302 (2007); Preliminary version in FOCS 2004 (2004)
44. Lyubashevsky, V., Micciancio, D.: Generalized Compact Knapsacks Are Collision Resistant. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006, Part II. LNCS, vol. 4052, pp. 144–155. Springer, Heidelberg (2006)
45. Peikert, C., Rosen, A.: Efficient Collision-Resistant Hashing from Worst-Case Assumptions on Cyclic Lattices. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 145–166. Springer, Heidelberg (2006)
46. Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom Functions and Lattices. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 719–737. Springer, Heidelberg (2012)

Relatively-Sound NIZKs and Password-Based Key-Exchange*

Charanjit Jutla[1] and Arnab Roy[2]

[1] IBM T.J. Watson Research Center,
Yorktown Heights, NY 10598, USA
[2] Fujitsu Laboratories of America,
Santa Clara, CA 94058, USA

Abstract. We define a new notion of relatively-sound non-interactive zero-knowledge (NIZK) proofs, where a private verifier with access to a trapdoor continues to be sound even when the Adversary has access to simulated proofs and common reference strings. It is likely that this weaker notion of relative-soundness suffices in most applications that need simulation-soundness. We show that for certain languages which are diverse groups, and hence allow smooth projective hash functions, one can obtain more efficient single-theorem relatively-sound NIZKs as opposed to simulation-sound NIZKs. We also show that such relatively-sound NIZKs can be used to build rather efficient publicly-verifiable CCA2-encryption schemes.

By employing this new publicly-verifiable encryption scheme along with an associated smooth projective-hash, we show that a recent PAK-model single-round password-based key exchange protocol of Katz and Vaikuntanathan, Proc. TCC 2011, can be made much more efficient. We also show a new single round UC-secure password-based key exchange protocol with only a constant number of group elements as communication cost, whereas the previous single round UC-protocol required $\Omega(k)$ group elements, where k is the security parameter.

1 Introduction

Authentication based on passwords is a significant security paradigm in today's world. Security in this scenario has been a challenging problem to solve because passwords typically come from low-entropy domains resulting in insufficient randomness for generating cryptographically secure keys. Gong et al. [11] raised the problem of designing protocols resistant to offline password guessing attacks, where other than guessing the low-entropy password by an online attack, the protocol must otherwise provide strong security based on a security parameter. Beginning with the work of Bellovin and Merritt [2], there has been considerable theoretical work in formalizing and obtaining secure protocols in the setting where only passwords are shared by peers (e.g. [1]), referred to as the

* Authors were supported in part by the Department of Homeland Security under grant FA8750-08-2-0091.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 485–503, 2012.

PAK-security model. From [15] onwards, these protocols employ smooth projective hash functions which have been a standard tool in cryptography ever since Cramer and Shoup defined them to give an efficient chosen ciphertext secure (CCA2) encryption scheme [7].

As illustrated by Gennaro and Lindell [10], who call this the non-malleable commitment paradigm, these protocols require the two peers A and B to non-malleably commit to their password to their peer (say B), e.g. by CCA2 encrypting the password under a public key given as a common reference string (CRS). While, the peer B cannot decrypt this commitment, it might be able to compute a smooth projective-hash on this commitment using a smooth hash key that it generates. The projection of this smooth hash key is sent to peer A, and peer A can compute the same smooth hash using the witness it has for the commitment. The two peers then output a product of two such smooth hashes, one for its own commitment and one for its peer. The problem, however, is that smooth projective-hash for the language, which in this case is the CCA2-ciphertext encrypting a password, is not easy to define, and [10] requires an adaptive smooth hash key, which makes the key-exchange protocol a multi-round protocol.

Recently, Katz and Vaikuntanathan [16] gave a single round protocol for password-based authenticated key exchange, by utilizing a publicly-verifiable CCA2-encryption scheme of Sahai [19]. A publicly-verifiable encryption scheme allows a (non-interactive) public verification of well-formedness of the ciphertext, i.e. it returns TRUE if and only if the decryption oracle will not return an "invalid ciphertext" response when queried with this ciphertext. The public verification allows the smooth hash to be defined on only a part of the ciphertext, which in [16] happens to be two El-Gamal encryptions of the password. Such smooth projective hashes are easy to define and compute.

While the resulting protocol requires only a constant number of group elements, as it employs simulation-sound extensions of Groth-Sahai NIZKs [13], under the decisional linear assumption (DLIN [3]) it still requires each party to send 65 group elements (and the run-time is proportionately high).

In this paper we show that the above scheme can be made much more efficient by using a novel concept of *relatively-sound* NIZKs rather than using simulation-sound NIZKs. Simulation-Sound NIZKs were first defined by Sahai [19], where it was used to convert Naor-Yung [18] CCA1-encryption scheme into the aforementioned CCA2-encryption scheme. In simulation-sound NIZKs the NIZK (public) verifier continues to be sound even when the Adversary is given the simulated CRS and proofs. We notice that in most applications what is really required is that a (private) verifier with access to a trapdoor continues to be sound in the simulated world, as long as this private verifier is equivalent to the public verifier in the real-world. The novel relatively-sound NIZKs captures this idea[1]. While it is an open problem whether relatively-sound NIZKs are strictly weaker than adaptive simulation-sound NIZKs, we show that relatively-sound NIZKs imply soundness under simulation of proofs of random (false or true)

[1] Relatively-sound NIZKs can be considered a hybrid of designated-verifier simulation-sound NIZKs [9] and simulation-sound NIZKs.

statements. Since, for many applications (including the current) such non-adaptive (random) simulation-sound NIZKs suffice, relative-soundness can be seen as a useful abstraction and tool for obtaining the former.

While it is easy to check that relative-soundness suffices in Sahai's original proof, in this paper we consider a further optimized construction. We prove that an augmented El-Gamal encryption scheme (reminiscent of [8]), along with a labeled single-theorem relatively-sound NIZK leads to a publicly-verifiable CCA2-encryption scheme. In the augmented El-Gamal scheme the public key (under the DDH or SXDH assumptions) consists of g, g^a, g^k, and the encryption of m with randomness x is $g^x, g^{ax}, m \cdot g^{kx}$. The labeled relatively-sound NIZK proves that the first two elements of the ciphertext use the same randomness x, with the third element used as label.

While a single-theorem simulation-sound NIZK could also have been used above, we show that one can obtain single-theorem relatively-sound NIZK far more cheaply than simulation-sound NIZK for this language. We use the fact that the language is a finite diverse group, and hence allows simple 2-universal projective hash functions [7], which allows us to build a private verifier. Under the SXDH assumption [13], converting a NIZK for this language to a relatively-sound NIZK only requires two more group elements, whereas the best-known simulation-sound extension would require nine group elements. Similarly, under the DLIN assumption, our extension requires only three more elements, whereas a simulation-sound extension requires at least 18 more elements [16]. Overall under the DLIN assumption, our publicly-verifiable CCA2 ciphertexts have only 19 group elements versus the 47 group elements in the Sahai scheme [19].

We show that using the new encryption scheme in the PAK-model protocol of [16], leads to a new protocol which is two to three times more efficient (under both SXDH and DLIN assumptions), with the SXDH-based scheme requiring only 10 group elements to be communicated[2].

UC Security. Canetti et al. [6] proposed a definition of security for password-based key exchange protocols within the Universally Composable (UC) security framework [5], which has the benefit of the universal composition theorem and as such can be deployed as a part of larger security contexts. In addition, their definition of security considers the case of arbitrary and unknown password distributions.

Katz and Vaikuntanathan [16] also gave a single round UC-secure protocol for password-based authenticated key exchange. However, their single round UC protocol is still inefficient as it uses general purpose NIZKs (for NP languages), and further requires proof of knowledge NIZKs. Even if the language for which zero knowledge proofs are required can be made to be given by simple algebraic relations in bilinear groups, the proof of knowledge for exponents of elements as required in their protocol makes it rather expensive.

[2] It should be remarked that other efficient publicly-verifiable CCA2-encryption schemes such as [17], which allow hash proofs on the (proof-less) part of the ciphertext can also be used in [16].

A second main contribution of this paper is an efficient UC-secure single-round protocol for password based key exchange. The main new ideas required for this efficient protocol are as follows: (a) The shared secret key is obtained in the target group of the bilinear pairings used in the NIZKs which allows for efficient simulator-extraction of group elements corresponding to the smooth-hash trapdoor keys. Such an extraction is required for UC-simulatability. (b) The NIZK proof of knowledge (for extraction) requires the NIZKs to be unbounded simulation-sound. A general construction for unbounded simulation-soundness was given in [4] which is based on a construction due to Groth [12], both of which can be seen to be using relative-soundness implicitly. This leads us to give an optimized version of this general construction. (c) We continue to use the Damgard style [8] encryption scheme, which allows for even more optimization of the unbounded simulation-sound construction for this specific language.

As a result, we get a single-round UC-secure protocol, where under the DLIN-assumption, each party only communicates 63 group elements, which is as efficient as the PAK-model protocol described in [16]. Under the SXDH assumption, our UC-secure protocol only requires 33 group elements.

For sake of exposition, we focus on giving complete proofs only under the SXDH assumption. All of the protocols are also given under the DLIN assumption in the full paper [14].

2 NIZK Definitions

In this section we give some definitions related to Non Interactive Zero Knowledge (NIZK) proofs. We will assume familiarity with usual definitions of NIZKs (see e.g. [19,13]). A proof for a relation R consists of a key generation algorithm K which produces the CRS ψ, a probabilistic polynomial time (PPT) prover P and a PPT verifier V.

Zero-Knowledge. We call (K, P, V) a **NIZK** proof for R if there exists a poly-time simulator (S_1, S_2), such that for all non-uniform PPT adversaries $\mathcal{A}$ we have $\Pr[\psi \leftarrow K(1^m) : \mathcal{A}^{P(\psi,\cdot,\cdot)}(\psi) = 1] \approx \Pr[(\sigma, \tau) \leftarrow S_1(1^m) : \mathcal{A}^{S(\sigma,\tau,\cdot,\cdot)}(\sigma) = 1]$, where $S(\sigma, \tau, x, w) = S_2(\sigma, \tau, x)$ for $(x, w) \in R$ and both oracles output failure if $(x, w) \notin R$.

One-time Simulation Soundness. A NIZK proof is one-time simulation sound NIZK if for all non-uniform PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ we have
$\Pr[(\sigma, \tau) \leftarrow S_1(1^m); (x, s) \leftarrow \mathcal{A}_1(\sigma); \pi \leftarrow S_2(\sigma, \tau, x); (x', \pi') \leftarrow \mathcal{A}_2(x, \pi, \sigma, s) :$
$((x', \pi') \neq (x, \pi))$ and $\neg\exists w'$ s.t. $(x', w') \in R$, and $V(\sigma, x', \pi') = 1] \approx 0$.

Unbounded Simulation Sound Extractability (uSS-NIZK). Consider a NIZK proof (K, P, V, S_1, S_2) along with an initialization algorithm SE_1 and a knowledge extractor E_2, such that SE_1 outputs (σ, τ, ξ) with (σ, τ) identical to values output by S_1. Such a proof is said to have the Unbounded Simulation Sound Extractability property if for all non-uniform PPT adversaries $\mathcal{A}$ we have
$\Pr[(\sigma, \tau, \xi) \leftarrow \mathrm{SE}_1(1^k); (x, \pi) \leftarrow \mathcal{A}^{S_2(\sigma,\tau,\cdot)}(\sigma); w \leftarrow E_2(\sigma, \xi, x, \pi) \ :$
$(x, \pi) \notin Q$ and $(x, w) \notin R$ and $V(\sigma, x, \pi) = 1] \approx 0$

where Q is the set of simulation queries and responses (x_i, π_i). For some subset of witnesses the extractor E_2 may extract witnesses in polynomial time, which will be the focus in this paper.

2.1 Relative Soundness

We now define a novel *weaker notion of simulation soundness*, which might suffice for most applications, especially in the case of single theorem (or one-time) simulation. It is possible that this weaker notion may be more efficient to implement, as we demonstrate later for a particularly important language, where we also show that the weaker notion suffices for the application at hand. In a nutshell, the weaker notion allows for the simulator to have a private verifier of its own, with access to a trapdoor. Simulation-soundness is now defined with respect to simulator's private verifier, and hence the name *relative-soundness*. There is an important further stipulation in the definition that the zero-knowledge property should hold even when the Adversary is given oracle access to private verifier in the simulated world (and public verifier in real world).

Labeled Single-Theorem Relatively-Sound NIZK (l-SRS-NIZK). Consider a sound and complete (labeled) proof (K, P, V) for a relation R along with a PPT private-verifier W and a PPT simulator (S_1, S_2). In a labeled proof, the prover P takes an input label, in addition to the statement to be proven. The verifier takes a statement, a label, and a proof. Such a proof is called a **labeled single-theorem relatively-sound NIZK** for R if for all non-uniform PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4)$ we have

relative-ZK:

$$\Pr[(\psi) \leftarrow K(1^m);\ (x, w, \mathtt{lbl}, s) \leftarrow \mathcal{A}_1^{V(\psi,\cdot,\cdot,\cdot)}(\psi);\ \pi \leftarrow P(\psi, x, w, \mathtt{lbl}) : \mathcal{A}_2^{V(\psi,\cdot,\cdot,\cdot)}(\pi, s) = 1] \approx$$

$$\Pr[(\sigma, \tau) \leftarrow \mathrm{S}_1(1^m);\ (x, w, \mathtt{lbl}, s) \leftarrow \mathcal{A}_1^{W(\sigma,\tau,\cdot,\cdot,\cdot)}(\sigma);\ \pi \leftarrow S_2(\sigma, \tau, x, \mathtt{lbl}) : \mathcal{A}_2^{W(\sigma,\tau,\cdot,\cdot,\cdot)}(\pi, s) = 1],$$

for A_1 restricted to producing (x, w) satisfying R, **and**

relative-simulation-soundness:

$$\Pr[(\sigma, \tau) \leftarrow \mathrm{S}_1(1^m); (x, \mathtt{lbl}, s) \leftarrow \mathcal{A}_3^{W(\sigma,\tau,\cdot,\cdot,\cdot)}(\sigma); \pi \leftarrow S_2(\sigma, \tau, x, \mathtt{lbl}); (x', \mathtt{lbl}', \pi') \leftarrow \mathcal{A}_4^{W(\sigma,\tau,\cdot,\cdot,\cdot)}(\pi, s) : \quad ((x', \mathtt{lbl}', \pi') \neq (x, \mathtt{lbl}, \pi)) \text{ and } \neg\exists w' \text{ s.t. } R(x', w') = 1, \text{ and } W(\sigma, \tau, x', \mathtt{lbl}', \pi') = 1] \approx 0.$$

Note that there are no other requirements on W other that those listed above. It is critical that relative-ZK is required only w.r.t. adversaries $(\mathcal{A}_1)$ that produce language members. Otherwise, relative-simulation-soundness would already imply normal simulation-soundness. Although it remains an open problem whether relatively-sound NIZKs are *strictly* weaker than simulation-sound NIZKS, the following shows the relation to non-adaptive simulation soundness, i.e. where the statements for which the proofs need to be simulated are chosen randomly.

Relation to Simulation-Soundness. Consider the following variant of One-time Simulation Soundness defined in Section 2. A NIZK proof for language

$L \subseteq X$ is a **non-adaptive one-time simulation-sound NIZK** if for all non-uniform PPT adversaries $\mathcal{A} = (\mathcal{A}_3, \mathcal{A}_4)$ we have

$$\Pr[(\sigma,\tau) \leftarrow \mathrm{S}_1(1^m); x \xleftarrow{\$} X; (\mathtt{lbl}, s) \leftarrow \mathcal{A}_3(\sigma, x); \pi \leftarrow S_2(\sigma,\tau,x,\mathtt{lbl});$$
$$(x', \mathtt{lbl}', \pi') \leftarrow \mathcal{A}_4(\pi, s) : ((x', \mathtt{lbl}', \pi') \neq (x, \mathtt{lbl}, \pi))$$
$$\text{and } \neg\exists w' \text{ s.t. } R(x', w') = 1, \text{ and } V(\sigma, x', \mathtt{lbl}', \pi') = 1] \approx 0.$$

Now, assume that the language L is *efficiently witness-samplable*, i.e. there is PPT machine which can efficiently sample from L along with the witness for the language member. Also, a language L, subset of a domain X, is called *hard* if no PPT adversary can distinguish between a (uniformly) random element of L from a random element of X.

Lemma 1. *For a hard and efficiently witness-samplable language L, an l-SRS-NIZK for L also satisfies the non-adaptive labeled one-time simulation soundness property for L.*

The proof of this lemma uses standard arguments, and a version of this lemma for unbounded simulation soundness also holds.

3 Smooth Projective Hash Functions

Fix a cyclic group $G = \langle g, \cdot \rangle$ of prime order q, such that $1/q$ is a negligible function of the security parameter. We define the El-Gamal encryption function as follows. For K, m in G, and x, define

$$\mathsf{enc}^{\mathsf{eg}}_K(m; x) \;=\; \langle g^x, K^x \cdot m \rangle$$

For K and pwd in G, define $L_{K,\mathrm{pwd}} = \{c = \langle R, P\rangle \mid \exists x : c = \mathsf{enc}^{\mathsf{eg}}_K(\mathrm{pwd}; x)\} \cap G \times G$. A **projective hash function** [7] is a keyed family of functions mapping elements in some message space X to the group G, and is associated with a language. Further, it comes with a **projection function** $\alpha : \mathcal{K} \to S$, where $\mathcal{K}$ is the key space and S is the projected key space. For our hash family, the key space is $\mathbb{Z}_q \times \mathbb{Z}_q$, and the projected key space is G. The message space X is the space of ciphertexts. For $n, \hat{n}$ in $\mathbb{Z}_q$, c in G^2, and K, pwd in G, define the hash family $\mathcal{H}^{K,\mathrm{pwd}}$ associated with $L_{K,\mathrm{pwd}}$ by

$$\mathcal{H}^{\mathrm{pwd}}_{n,\hat{n}}(c = \langle R, P\rangle) = (P/\mathrm{pwd})^{\hat{n}} \cdot R^n, \qquad \alpha^{K,\mathrm{pwd}}(n, \hat{n}) = g^n \cdot (K)^{\hat{n}}.$$

It is straightforward to see that, if $c = \mathsf{enc}^{\mathsf{eg}}_K(\mathrm{pwd}; x)$ for some x, then $\mathcal{H}^{\mathrm{pwd}}_{n,\hat{n}}(c) = \alpha^{K,\mathrm{pwd}}(n, \hat{n})^x$.

For any K and pwd in G, $\mathcal{H}^{K,\mathrm{pwd}}$ is said to be **smooth** [7] w.r.t. $L = L_{K,\mathrm{pwd}}$, if for any c' in G^2, but *not* in L, the statistical distance between the distribution of the pair $(\mathcal{H}^{K,\mathrm{pwd}}_{n,\hat{n}}(c'),\ \alpha^{K,\mathrm{pwd}}(n.\hat{n}))$ and the pair (g^{d_1}, g^{d_2}) is negligible, where $n, \hat{n}, d_1, d_2$ are chosen randomly and independently from $\mathbb{Z}_q$. It is a simple exercise to see that $\mathcal{H}^{K,\mathrm{pwd}}$ is smooth with respect to $L_{K,\mathrm{pwd}}$.

We also define a projective hash function family associated with any language L to be **2-universal** [7] if for all $s \in S, x, x' \in X$, and $\pi, \pi' \in G$ with $x \notin L \cup \{x'\}$, it holds that $\Pr_k[H_k(x) = \pi \mid H_k(x') = \pi' \ \wedge\ \alpha(k) = s] \leq 1/q$.

4 Bilinear Assumptions

Throughout the paper, we use (bilinear) groups G_1, G_2, G_T each of prime order q, which allow an efficiently computable $\mathbb{Z}_q$-bilinear pairing map $e : G_1 \times G_2 \to G_T$.

SXDH: [13] The symmetric external decisional Diffie-Hellman (SXDH) assumption states that the decisional Diffie-Hellman (DDH) problem is hard in both groups G_1 and G_2.

DLIN: [3] In groups such that G_1 is same as G_2, the decisional linear (DLIN) assumption states that given $(\alpha\mathcal{P}, \beta\mathcal{P}, r\alpha\mathcal{P}, s\beta\mathcal{P}, t\mathcal{P})$ for random $\alpha, \beta, r, s \in \mathbb{Z}_q$, and arbitrary generator $\mathcal{P}$ of G_1, it is hard to distinguish between $t = r + s$ and a random t.

5 A Publicly-Verifiable CCA2-Encryption Scheme

In this section we describe a CCA2-Encryption scheme that has the property that a potential ciphertext can be publicly verified to be a valid ciphertext of some message. Note that Sahai [19] had previously given a publicly-verifiable CCA2-encryption scheme employing the Naor-Yung CCA1-scheme [18], but our scheme is simpler and more efficient.

One might be tempted to take the Cramer-Shoup encryption scheme, and extend the ciphertext by including a NIZK proof that the 2-universal smooth projective-hash [7] was correctly computed. However, since the NIZK scheme by itself may be malleable, this may render the scheme insecure in the CCA2-model. There are two potential fixes to this: (a) make the NIZK single theorem simulation-sound, or (b) include the NIZK commitments to the witness in the projective-hash. While it is not that difficult to see that (a) may lead to a correct publicly-verifiable CCA2-scheme (just as in [19]), the second idea (b) may seem far-fetched.

We now show that it suffices to make the NIZK proof a labeled single-theorem *relatively-sound* NIZK, and further one just needs to prove in this NIZK that the Diffie-Hellman tuple in the ciphertext is well-formed, i.e. it is of the form g^x, A^x. We later show that there exists a very efficient way to extend a single-theorem Groth-Sahai NIZK of this statement to be a relatively-sound proof, such that the resulting publicly-verifiable CCA2-scheme is just the idea (b) mentioned above.

To formally define publicly-verifiable CCA2-encryption schemes, one just extends the standard IND-CCA2 definition of encryption with a public verification function V which takes the public key and a potential ciphertext as arguments, and it returns true iff the decryption function when supplied with the same ciphertext does not return "invalid ciphertext".

For given g, A, let the relation $\mathcal{R} = \{((\rho, \hat{\rho}), x) \mid \rho = g^x, \hat{\rho} = A^x\}$. We now define a *labeled* publicly-verifiable public-key encryption scheme DHENC as follows:

Key Generation: Generate $g, A \xleftarrow{\$} G_1$, and $k \xleftarrow{\$} \mathbb{Z}_q$. Let $K = g^k$. Let ψ be the CRS for an l-SRS-NIZK. The public key is (g, A, K, ψ) and the private key is k.

Encrypt: Given plaintext $m \in G_1$, and label $\texttt{lbl}$. Choose $x \xleftarrow{\$} \mathbb{Z}_q$. Let the triple $\langle \rho, \hat{\rho}, \gamma \rangle$ be $\langle g^x, A^x, mK^x \rangle$. Let π be an l-SRS-NIZK proof of $((\rho, \hat{\rho}), x) \in \mathcal{R}$ with label $\gamma, \texttt{lbl}$. The ciphertext is $(\rho, \hat{\rho}, \gamma, \pi)$.

Decrypt: Given ciphertext $c = (\rho, \hat{\rho}, \gamma, \pi)$ and label $\texttt{lbl}$. Verify if π is an l-SRS-NIZK proof for $(\rho, \hat{\rho})$ and label $\gamma, \texttt{lbl}$. If verification fails output $\perp$. Otherwise output $m = \frac{\gamma}{\rho^k}$.

Verify: Given ciphertext $c = (\rho, \hat{\rho}, \gamma, \pi)$ and label $\texttt{lbl}$. Verify if π is an l-SRS-NIZK proof for $(\rho, \hat{\rho})$ and label $\gamma, \texttt{lbl}$. If verification fails output false else output true.

Theorem 1. *The scheme* DHENC *is publicly-verifiable (labeled) IND-CCA2 secure.*

The full proof of this theorem can be found in [14], but the main idea is that the decryption can be done as either γ/ρ^k, or as $\gamma/(\rho^{k'}\hat{\rho}^{k''})$, where the Simulator chooses the public key K as $g^{k'}A^{k''}$. The encryption oracle hides the message by employing DDH as follows: (1) The NIZK CRS in the original experiment is the binding-CRS, and the decryption oracle in the original experiment does a public verification of proofs in each adversarially supplied ciphertext. (2) The NIZK CRS is switched to be the hiding CRS, the proof switched to a simulator generated proof, and decryption oracle now uses private-verification. This is an indistinguishable change by the relative-ZK property of l-SRS-NIZK. Note, x is no more used in the simulated proof. (3) The decryption is done as $\gamma/(\rho^{k'}\hat{\rho}^{k''})$, which is equivalent because of relative-simulation soundness property of l-SRS-NIZK. (4) DDH is employed, as only $A(= g^a)$ is being used in the simulation, instead of a. This leads to A^x being replaced by an independent X'. (5) The decryption is done as γ/ρ^k, which is again equivalent by relative-soundness. (6) the message in the encryption can be switched by pairwise independence in k, and this step is information-theoretic. More precisely, $g^{xk'}(X')^{k''}$ is random and independent of g^x, X', K, A, as well as Adversary's coins with high probability. (7) Next we do all the above steps (2)-(5) in reverse.

6 l-SRS-NIZK for the DDH Language

Let G_1 and G_2 be two groups with a bilinear pairing $e : G_1 \times G_2 \to G_T$ and $|G_1| = |G_2| = |G_T| = q$, a prime number. Also assume that DDH is hard for both G_1 and G_2. Recall that this is the $\texttt{SXDH}$ assumption. Let $L_{g,A}$ be the language: $\{(\rho, \hat{\rho}) \in {G_1}^2 \mid \exists x.\ \rho = g^x \wedge\ \hat{\rho} = A^x\}$, with g, A in G_1.

Note that this language is actually a cyclic group with generator $\langle g, A \rangle$, and forms a diverse group system [7]. In [7], Cramer and Shoup show how to obtain 2-universal projective hash functions for such languages, and we use these hash functions for private-verification.

We construct an l-SRS-NIZK proof system for $L_{g,A}$, as follows:

CRS Generation: Generate $\mathcal{P} \xleftarrow{\$} G_2$ and $u, v, d_1, d_2, e_1, e_2 \xleftarrow{\$} \mathbb{Z}_q$. Compute $(P, Q, R, S, \mathbf{d}, \mathbf{e}) = (\mathcal{P}, \mathcal{P}^u, \mathcal{P}^v, \mathcal{P}^{uv+1}, g^{d_1}A^{d_2}, g^{e_1}A^{e_2})$. The CRS is $\psi =$

$(P, Q, R, S, \mathbf{d}, \mathbf{e})$. The first four elements are as in the Groth-Sahai NIZK for SXDH (*binding* CRS), and the last two are the projection keys for a 2-universal projective-hash for the DDH language (just as [7]), to be used in the relatively-sound system.

The simulation CRS σ is $(P, Q, R, S, \mathbf{d}, \mathbf{e}) = (\mathcal{P}, \mathcal{P}^u, \mathcal{P}^v, \mathcal{P}^{uv}, g^{d_1}A^{d_2}, g^{e_1}A^{e_2})$. This is the *hiding* CRS of GS-NIZK for SXDH along with $\mathbf{d}$ and $\mathbf{e}$ as above. The trapdoor is $\tau = (u, d_1, d_2, e_1, e_2)$.

Prover: Given witness x, candidate (g^x, A^x), and label $\texttt{lbl}$, construct proof as follows. Generate $s \xleftarrow{\$} \mathbb{Z}_q$. Compute $t \leftarrow H(g^x, A^x, Q^xP^s, S^xR^s, \texttt{lbl})$, where H is a collision resistant hash function. Then compute: $(\beta, c_1, c_2, \theta, \phi, \chi) \leftarrow ((\mathbf{de}^t)^x, Q^xP^s, S^xR^s, g^s, A^s, (\mathbf{de}^t)^s)$. Output proof $\pi = (\beta,\ c_1,\ c_2,\ \theta,\ \phi,\ \chi)$. The first element is a 2-universal projective-hash computed on the candidate with witness x. The last five elements can be interpreted as generated by the Groth-Sahai NIWI proof (which also happens to be a NIZK proof) for the language $\{\rho, \hat{\rho}, h \mid \exists x : \rho = g^x, \hat{\rho} = A^x, h = (\mathbf{de}^t)^x\}$, where t is a hash of ρ, $\hat{\rho}$, $\texttt{lbl}$, *and the commitment* to x in the NIWI system, i.e. Q^xP^s, S^xR^s.

Simulator: Given a candidate $(\rho, \hat{\rho})$, generate the proof as follows. Generate $s \xleftarrow{\$} \mathbb{Z}_q$ and compute $t \leftarrow H(\rho, \hat{\rho}, P^s, R^s, \texttt{lbl})$. Then compute

$$\pi = (\beta, c_1, c_2, \theta, \phi, \chi) = \left(\rho^{d_1}\hat{\rho}^{d_2}(\rho^{e_1}\hat{\rho}^{e_2})^t, P^s, R^s, \rho^{-u}g^s, \hat{\rho}^{-u}A^s, \beta^{-u}(\mathbf{de}^t)^s\right)$$

Public Verify: Given $\pi = (\beta, c_1, c_2, \theta, \phi, \chi)$ as a candidate proof of $(\rho, \hat{\rho})$ with label $\texttt{lbl}$, compute $t \leftarrow H(\rho, \hat{\rho}, c_1, c_2, \texttt{lbl})$. Then check the following equations:

$$\begin{pmatrix} e(g, c_1) \stackrel{?}{=} e(\rho, Q) \cdot e(\theta, P), & e(g, c_2) \stackrel{?}{=} e(\rho, S) \cdot e(\theta, R) \\ e(A, c_1) \stackrel{?}{=} e(\hat{\rho}, Q) \cdot e(\phi, P), & e(A, c_2) \stackrel{?}{=} e(\hat{\rho}, S) \cdot e(\phi, R) \\ e(\mathbf{de}^t, c_1) \stackrel{?}{=} e(\beta, Q) \cdot e(\chi, P), & e(\mathbf{de}^t, c_2) \stackrel{?}{=} e(\beta, S) \cdot e(\chi, R) \end{pmatrix}$$

Private Verify: Given $\pi = (\beta, c_1, c_2, \theta, \phi, \chi)$ as a candidate proof of $(\rho, \hat{\rho})$ with label $\texttt{lbl}$, compute $t \leftarrow H(\rho, \hat{\rho}, c_1, c_2, \texttt{lbl})$. Then first do public verification and if that succeeds then check the following equation: $\beta \stackrel{?}{=} \rho^{d_1}\hat{\rho}^{d_2}(\rho^{e_1}\hat{\rho}^{e_2})^t$. Note that this private verifier is well-defined in the real world as well. In addition, its trapdoor (d_1, d_2, e_1, e_2) is identically generated in both the real and the simulated worlds.

Theorem 2. *The above system is an l-SRS-NIZK proof system for* $L_{g,A}$.

Proof Sketch: We focus on Relative-ZK and Relative-SS properties. For the former, we need to show that the simulation CRS, and a proof for $(\rho, \hat{\rho})$ with label $\texttt{lbl}$ is computationally indistinguishable from the real CRS and a real proof, even when the Adversary has oracle access to respective verifiers. This is accomplished by a sequence of games, where the first game is same as the real world game. In the second game, the CRS and the proof remain the same but the verifier in the oracle is changed to be the private verifier, which in our case is well-defined in the real world. We need to show that public verification implies

private verification, but this follows from soundness of the Groth-Sahai NIZK, as well as the fact that on a *valid* DDH tuple the projection hash is same whether it is computed using the witness and the projection key or using the private hash keys. In the final game we switch to the simulation CRS and simulated proof, and indistinguishability follows from ZK property of Groth-Sahai NIZKs and the fact that the private verification trapdoor is independent of the Groth-Sahai NIZK CRS (hiding or binding).

The relative-simulation-soundness property is proven using the 2-universal property of the projective smooth hash (just as in [7]), but additionally using the fact that in Groth-Sahai NIZKs, once the commitments to the witnesses are fixed, there is a unique proof satisfying the linear equations of the type used in the above NIZK proof. This holds for both the SXDH and the DLIN assumptions. □

The l-SRS-NIZK proof for DDH language above consists of six group elements. The l-SRS-NIZK proof for the DLIN language (and under the DLIN assumption), given in the full paper [14], consists of 15 group elements.

7 Secure Protocol in the PAK Model

In this section we present a password-based key exchange protocol secure in the PAK model of security due to Bellare, Pointcheval and Rogaway [1]. We instantiate the single-round scheme due to Katz and Vaikuntanathan [16], which is described in Figure 1, with the more efficient publicly-verifiable CCA-secure encryption scheme DHENC of Section 5, which enables a more efficient hash proof as well. The common reference string (CRS) is just the public key of this scheme.

The projective-hash family used in this scheme is $\mathcal{H}^{\mathrm{pw}}$ along with the projection function $\alpha^{K,\mathrm{pw}}$ defined in Section 3, where K is from the public-key (i.e. CRS). Note that the input *label* to the hash function is ignored in $\mathcal{H}^{\mathrm{pw}}$. Also, α does not depend on pw.

CRS $= pk$

Party P_i	$\mathcal{A}$	Party P_j
$k_i \xleftarrow{\$}$ Hash-K; $s_i \leftarrow \alpha(k_i)$ $label_i \leftarrow (P_i, P_j, s_i)$ $C_i \leftarrow \mathsf{enc}_{pk}(label_i, pw)$	$\xrightarrow{label_i, C_i}$	$k_j \xleftarrow{\$}$ Hash-K; $s_j \leftarrow \alpha(k_j)$ $label_j \leftarrow (P_j, P_i, s_j)$ $C_j \leftarrow \mathsf{enc}_{pk}(label_j, pw)$
	$\xleftarrow{label'_j, C'_j}$	
	$\xleftarrow{label_j, C_j}$	
	$\xrightarrow{label'_i, C'_i}$	
Reject if C'_j is not a publicly verified ciphertext with label $label'_j$. $sk_i \leftarrow H_{k_i}(label'_j, C'_j, pw)$ $\cdot H_{k_j}(label_i, C_i, pw)$		Reject if C'_i is not a publicly verified ciphertext with label $label'_i$. $sk_j \leftarrow H_{k_j}(label'_i, C'_i, pw)$ $\cdot H_{k_i}(label_j, C_j, pw)$

Fig. 1. Single-round PAK-Model Secure Password-based Authenticated KE

Theorem 3. *Assume the existence of SXDH-hard groups G_1 and G_2. Then the protocol in Figure 1 is secure in the PAK model.*

The proof of this theorem is same as the proof in [16], as we have modularized the various constructs required in that proof. The main idea is that once the CCA2-encryption scheme is publicly verifiable, then the smooth hash needs to be just over the language $L_{K,\mathrm{pw}}$, which are CPA encryptions of password.

8 Secure Protocol in the UC Model

The essential elements of the Universal Composability framework can be found in [5]. We adopt the definition for password-based key exchange from Canetti et al [6]. The following description is a summary from [6]. The formal description is given in Figure 2.

Functionality $\mathcal{F}_{\mathbf{pwKE}}$

The functionality $\mathcal{F}_{\mathrm{pwKE}}$ is parameterized by a security parameter k. It interacts with an adversary S and a set of parties via the following queries:

Upon receiving a query $(\mathsf{NewSession}, sid, P_i, P_j, pw, role)$ **from** P_i**:** Send $(\mathsf{NewSession}, sid, P_i, P_j, role)$ to S. In addition, if this is the first NewSession query, or if this is the second NewSession query and there is a record (P_j, P_i, pw'), then record (P_i, P_j, pw) and mark this record fresh.

Upon receiving a query $(\mathsf{TestPwd}, sid, P_i, pw')$ **from the adversary** S**:** If there is a record of the form (P_i, P_j, pw) which is fresh, then do: If $pw = pw'$, mark the record compromised and reply to S with "correct guess". If $pw \neq pw'$, mark the record interrupted and reply with "wrong guess".

Upon receiving a query $(\mathsf{NewKey}, sid, P_i, sk)$ **from** S**, where** $|sk| = k$**:** If there is a record of the form (P_i, P_j, pw), and this is the first NewKey query for P_i, then:

- If this record is compromised, or either P_i or P_j is corrupted, then output (sid, sk) to player P_i.
- If this record is fresh, and there is a record (P_j, P_i, pw') with $pw' = pw$, and a key sk' was sent to P_j, and (P_j, P_i, pw) was fresh at the time, then output (sid, sk') to P_i.
- In any other case, pick a new random key sk' of length k and send (sid, sk') to P_i.

Either way, mark the record (P_i, P_j, pw) as completed.

Fig. 2. The password-based key-exchange functionality $\mathcal{F}_{\mathrm{pwKE}}$

Like the key exchange functionality, if both participating parties are not corrupted, then they receive the same uniformly distributed session key and the adversary learns nothing of the key except that it was generated. However, if

one of the parties is corrupted, then the adversary determines the session key. If the adversary makes a wrong password guess in a given session, then the session is marked interrupted and the parties are provided random and independent session keys. If the adversary makes a successful guess, then the session is marked compromised, and the Adversary gets the power to set the session key.

8.1 A Single Round UC Password-Based Key Exchange Protocol

The single-round UC protocol under the SXDH assumption uses labeled unbounded simulation sound G_2-extractable NIZKs (uSS-NIZK). Consider parties P_i and P_j involved in the protocol with SSID ssid. The CRS is three group elements $g, A(=g^a), K(=g^k)$ chosen randomly from G_1, another element $\mathcal{P}$ chosen randomly from G_2, and a uSS-NIZK CRS ψ. Since $g, \mathcal{P}$ are also part of the uSS-NIZK CRS, having chosen the NIZK CRS, g, $\mathcal{P}$ are already determined. The protocol is symmetric and asynchronous with each party computing a message to be sent, then receiving a corresponding message and computing a key. Therefore, we just describe it from the perspective of one party; the other is symmetric.

Party P_i generates $x \xleftarrow{\$} \mathbb{Z}_q$ and computes $c_1 = \langle g^x, A^x, K^x \cdot pw\rangle$. It also generates hash key $(n_1, \hat{n_1}) \xleftarrow{\$} (\mathbb{Z}_q)^2$ and computes the projection key $\eta_1 = \alpha^{K,\text{pwd}}(n_1, \hat{n_1}) = g^n \cdot K^{\hat{n}}$. Finally it computes a NIZK proof of consistency in the following way:

$$\pi_1 = \text{uSS-NIZK}_\psi(g^x, A^x, \eta_1; x, \mathcal{P}^{n_1}, \mathcal{P}^{\hat{n}_1}) \text{ with label } \langle P_i, P_j, \mathsf{ssid}\rangle$$

Note that π here denotes the commitments to the witnesses as well as the further proof as in the Groth-Sahai system. The NP language L for the NIZK is

$$L = \{\rho, \hat{\rho}, \eta \mid \exists x, N, \hat{N} \ : \ \rho = g^x, \hat{\rho} = A^x, \ e(\eta, \mathcal{P}) = e(g, N)e(K, \hat{N})\}$$

Now, the message sent by P_i is $\langle c_1, \eta_1, \pi_1\rangle$. Let the message received by P_i in this session, supposedly from P_j, be $\langle c_2', \eta_2', \pi_2'\rangle$. Let c_2' be parsed as $(\rho_2', \hat{\rho_2'}, \gamma_2')$. If any of $\rho_2', \hat{\rho}_2', \gamma_2', \eta_2'$ is not in $G_1\backslash\{1\}$, or uSS-NIZK-Verify$(\pi_2'; \rho_2', \hat{\rho}_2', \eta_2')$ with label $\langle P_j, P_i, \mathsf{ssid}\rangle$ turns out to be false, then it sets its session key sk_1 randomly from the target group of e, G_T. Otherwise it is computed as follows:

$$h_2' = (\frac{\gamma_2'}{\text{pwd}})^{\hat{n}_1}(\rho_2')^{n_1} \qquad h_1 = (\eta_2')^{x_1} \qquad h_3 = h_2' \cdot h_1 \qquad \text{sk}_1 = e(h_3, \mathcal{P}).$$

Theorem 4. *Assume the existence of a SXDH-hard group, a labeled unbounded simulation-sound G_2-extractable NIZK proof system. Then the protocol in Figure 3 securely realizes the $\widehat{\mathcal{F}}_{\text{PWKE}}$ functionality in the $\mathcal{F}_{\mathsf{crs}}$ hybrid model, in the presence of static corruption adversaries.*

In the next section we demonstrate a simulator which uses $\widehat{\mathcal{F}}_{\text{PWKE}}$ to simulate the protocol to an adversary, thus proving Theorem 4.

A more optimized version of such a general labeled unbounded simulation sound G_2-extractable NIZK [7] is given in the Appendix in Section A. In fact,

$\mathrm{CRS} = g, \mathcal{P}, A, K, \psi : \quad g, A, K \xleftarrow{\$} G_1 \quad \mathcal{P} \xleftarrow{\$} G_2 \quad \psi = \text{uSS-NIZK CRS}$

Party P_i		Adv $\mathcal{A}$
Input (NewSession, sid, ssid, P_i, P_j, pwd, $initiator/responder$)		
Choose $x_1, n_1, \hat{n}_1 \xleftarrow{\$} \mathbb{Z}_q$.		
Set $\rho_1 = g^{x_1}$, $\hat{\rho}_1 = (A)^{x_1}$, $\gamma_1 = \mathrm{pwd} \cdot K^{x_1}$, $\eta_1 = g^{n_1}(K)^{\hat{n}_1}$,	$\xrightarrow{c_1, \eta_1, \pi_1}$	$\mathcal{A}$
Let $c_1 = \langle \rho_1, \hat{\rho}_1, \gamma_1 \rangle$, and		
$\pi_1 = \text{uSS-NIZK}_\psi(\rho_1, \hat{\rho}_1, \eta_1; x_1, \mathcal{P}^{n_1}, \mathcal{P}^{\hat{n}_1})$ with label $\langle P_i, P_j, \mathsf{ssid} \rangle$.		
	$\xleftarrow{c_2', \eta_2', \pi_2'}$	$\mathcal{A}$
Let $c_2' = \langle \rho_2', \hat{\rho}_2', \gamma_2' \rangle$.		
If any of $\rho_2', \hat{\rho}_2', \gamma_2', \eta_2'$ is not in $G_1 \backslash \{1\}$, or		
not uSS-NIZK-Verify$(\pi_2; \rho_2', \hat{\rho}_2', \eta_2')$ with label $\langle P_j, P_i, \mathsf{ssid} \rangle$		
set $\mathrm{sk}_1 \xleftarrow{\$} G_T$, else		
compute $h_2' = (\frac{\gamma_2'}{\mathrm{pwd}})^{\hat{n}_1}(\rho_2')^{n_1}$, $h_1 = (\eta_2')^{x_1}$, $\mathrm{sk}_1 = e(h_2' \cdot h_1, \mathcal{P})$.		
Output $(sid, \mathsf{ssid}, \mathrm{sk}_1)$.		

Fig. 3. Single round UC-secure Password-based KE under SXDH Assumption

for the language above for which such a NIZK is required, we give a further optimization in [14]. Based on this optimized construction, the uSS-NIZK requires 29 group elements. A similar construction under the DLIN assumption, and for the DLIN based UC-secure PWKE-construction (given in the full paper [14]) requires 54 group elements.

8.2 The Simulator for the UC Protocol

The trapdoor keys a, k for the CRS are chosen differently by the simulator. Instead of choosing a, k randomly from $\mathbb{Z}_q$, the simulator chooses a, k', k'' from $\mathbb{Z}_q$ and sets $k = k' + a \cdot k''$. It outputs $A = g^a$ and $K = g^k = g^{k'}(g^a)^{k''}$ as before. Note that this does not change the distribution of A and K, as $\mathbb{Z}_q$ is a field. (We will continue to write k for $k' + ak''$, except when the simulation in some experiments needs to be done with g^a, instead of a).

Simulator S also invokes the initialization phase SE_1 of the labeled uSS-NIZK (with security parameter m) to obtain (σ, τ, ξ). S then gives A, K, and σ to the real world adversary $\mathcal{A}$ as the *common reference string*. Thereafter, the simulator S interacts with the environment $\mathcal{Z}$, the functionality $\widehat{\mathcal{F}}_{\mathrm{PWKE}}$, and uses $\mathcal{A}$ as a subroutine. The messages between $\mathcal{Z}$ and $\mathcal{A}$ are just forwarded by S.

The main difference in the simulation of the real world parties is that S uses a dummy message μ instead of the real password which it does not have access to. Further, it generates all proofs using the NIZK simulator S_2 instead of real prover.

New Session: Sending a message to $\mathcal{A}$. On message (NewSession, *sid*, ssid, i, j, role) from $\widehat{\mathcal{F}}_{\mathrm{PWKE}}$, S starts simulating a new session of the protocol Π for party P_i, peer P_j, session identifier ssid, and CRS $= (A, K, \psi)$. We will denote this session by (P_i, ssid). To simulate this session, S chooses x_1 at random, and

sets c_1 ($= \langle \rho_1, \hat{\rho}_1, \gamma_1 \rangle$) to $\langle g^{x_1}, A^{x_1}, \mu \cdot K^{x_1} \rangle$. It also chooses hash keys $n_1, \hat{n_1}$ at random, and computes the smooth-hash projected key η_1 as in the real protocol as well. S obtains a fake NIZK proof π_1 using the simulator S_2 of the NIZK, and the CRS σ, and simulation trapdoor τ. It then hands c_1, η_1, π_1 to $\mathcal{A}$ on behalf of this session.

On Receiving a Message from $\mathcal{A}$. On receiving a message c'_2, η'_2, π'_2 from $\mathcal{A}$ intended for this session (P_i, ssid), the simulator S makes the real world protocol checks including verifying the NIZK proof using the NIZK-verifier. If any of the checks fail, it issues a TestPwd call to $\widehat{\mathcal{F}}_{\text{PWKE}}$ with the dummy password μ, followed by a NewKey call with a random session key, which leads to the functionality issuing a random and independent session key to the party P_i (regardless of whether the session was interrupted or compromised).

Otherwise, it computes pwd′ by decrypting c'_2, i.e. setting it to $\gamma'_2/(\rho'_2)^k$. If the message received from $\mathcal{A}$ is same as message sent by S on behalf of peer P_j in session ssid, then S just issues a NewKey call for P_i. Otherwise, S calls $\widehat{\mathcal{F}}_{\text{PWKE}}$ with (TestPwd, ssid, P_i, pwd′). Regardless of the reply from $\mathcal{F}$, it then issues a NewKey call for P_i with key computed as follows (*this is different from the real-world protocol.*). This has the effect that if the pwd′ was same as the actual pwd in $\widehat{\mathcal{F}}_{\text{PWKE}}$ then the session key is determined by the Simulator, otherwise the session key is set to a random and independent value. Here is the complete simulator code (stated as it's overall experiment with $\mathcal{Z}$, including $\mathcal{F}$'s communication with $\mathcal{Z}$):

1. Let $c'_2 = \langle \rho'_2, \hat{\rho}'_2, \gamma'_2 \rangle$.
2. If any of $\rho'_2, \hat{\rho}'_2, \gamma'_2, \eta'_2$ is not in $G_1 \backslash \{1\}$, or *not* uSS-NIZK-Verify$(\pi'_2; \rho'_2, \hat{\rho}'_2, \eta'_2)$ with label $\langle P_j, P_i, \mathsf{ssid} \rangle$, output $\text{sk}_1 \xleftarrow{\$} G_T$, else compute as follows.
3. If msg rcvd == msg sent in same session (same SSID) by peer, set $\text{sk}_1 \xleftarrow{\$} G_T$, unless the peer also received a legitimate message and its key has already been set, in which case that same key is used to set sk_1.
4. Else, compute $N'_2, \hat{N}'_2$ from the proof π'_2, using the extraction trapdoor ξ.
5. Compute $\text{pwd}' = \gamma'_2/(\rho'_2)^k$. If (pwd′ $\neq$ pwd) then $\text{sk}_1 \xleftarrow{\$} G_T$, else
6. $h'_2 = (\frac{\gamma'_2}{\text{pwd}'})^{\hat{n}_1} (\rho'_2)^{n_1}, h_1 = (\eta'_2)^{x_1}$; set $\text{sk}_1 = e(h'_2, \mathcal{P}) \cdot e(h_1, \mathcal{P}) \cdot e(\mu/\text{pwd}, \hat{N}'_2)$.

Note that the main difference is the additional factor $e(\mu/\text{pwd}, \hat{N}'_2)$.

8.3 Proof of Indistinguishability for the UC Protocol

We now describe a series of experiments between the Simulator and the environment, starting with Expt_0 which is the same as the experiment described as the Simulator in Section 8.2 above, and ending with an experiment which is identical to the real world execution of the protocol in Fig 3. We will show that the environment has negligible advantage in distinguishing between these experiments, leading to a proof of realization of $\mathcal{F}_{\text{PWKE}}$ by the protocol Π.

For each instance, we will use subscript 2 along with a prime, to refer to variables after the reception of the message from $\mathcal{A}$, and use subscript 1 to refer

to variables computed before sending the message to $\mathcal{A}$. We will call a message legitimate if it was not altered by the adversary, and delivered in the correct session, and to the correct party.

Expt_1**:** The experiment Expt_1 is same as Expt_0 except for the following modified step 3 in the reception code: *If msg rcvd == msg sent in same session by peer, set* sk_1 to

$$e(\mathcal{H}^{\text{pwd}}_{n_1,\hat{n}_1}(\mathsf{enc}^{\mathsf{eg}}_{g^k}(\mu; x_2)) \cdot \mathcal{H}^{\text{pwd}}_{n_2,\hat{n}_2}(\mathsf{enc}^{\mathsf{eg}}_{g^k}(\mu; x_1)), \mathcal{P}).$$

Because the hash proof system is for languages with messages encrypting real password, the smooth-hash-proof yields random values from the adversary's point of view. Note that we only employ the hash proof system corresponding to n_1 and $\hat{n}_1$, and note that the second factor corresponding to n_2 and $\hat{n}_2$ is independent of the first. In step 6, n_1 and $\hat{n}_1$ are being used, but the code never gets there if the msg received is same as message sent by legitimate peer.

Expt_2: Next, we replace all occurrences of $e(h_1, \mathcal{P})$ $(= e((\eta'_2)^{x_1}, \mathcal{P}))$ in the computation of sk_1 in Step 6 of the reception code by $e(g, N'_2)^{x_1} \cdot e(K, (\hat{N}'_2)^{x_1})$, which is the same as $e(g^{x_1}, N') \cdot e(K^{x_1}, \hat{N}')$. This leads to an indistinguishable change as the simulator had verified the NIZK proofs, and the NIZK proofs have unbounded simulation extractability property, and thus $e(\eta'_2, \mathcal{P}) = e(g, N'_2)e(K, \hat{N}'_2)$.

Expt_3: The next change in simulation is to replace μ by the real password in the outgoing message element γ. However, since the simulator is employing k to compute pwd', one cannot directly employ DDH to replace μ by pwd in outgoing γ. However, since we are using an augmented El-Gamal encryption scheme, i.e. also including $\hat{\rho}$ in the outgoing message along with a proof of its relation to ρ, we can use the pairwise independence in k to accomplish our goal, just as in CCA2 scheme DHENC described in Section 5.

At this point, not only is the outgoing γ_1 being computed as $K^{x_1} \cdot \text{pwd}$, i.e. $c_1 = \mathsf{enc}^{\mathsf{eg}}_K(\text{pwd}; x_1)$, but also in the reception phase of the same (ssid, P_i), the term $e(\mu/\text{pwd}, \hat{N}'_2)$ has been replaced by 1. Recall that in Expt_2, $e(h_1, \mathcal{P})$ was replaced by $e(g^{x_1}, N') \cdot e(K^{x_1}, \hat{N}')$, and now $e(K^{x_1}, \hat{N}')$ has been replaced by $e(\text{pwd}/\mu \cdot K^{x_1}, \hat{N}')$, which is then equivalent to replacing $e(\mu/\text{pwd}, \hat{N}'_2)$ by 1 in Step 6. Further, if the message received was legitimate, then sk_1 is now set to

$$e(\mathcal{H}^{\text{pwd}}_{n_1,\hat{n}_1}(\mathsf{enc}^{\mathsf{eg}}_{g^k}(\mu; x_2)) \cdot \mathcal{H}^{\text{pwd}}_{n_2,\hat{n}_2}(\mathsf{enc}^{\mathsf{eg}}_{g^k}(\text{pwd}; x_1), \mathcal{P}).$$

Similarly, if the peer received a legitimate message, its computation of sk_1 has a similar change, i.e. its first factor has μ replaced by pwd. Thus, at the end of these sequence of hybrid experiments, if the message received was legitimate, then sk_1 is now set to $e(\mathcal{H}^{\text{pwd}}_{n_1,\hat{n}_1}(\mathsf{enc}^{\mathsf{eg}}_{g^k}(\text{pwd}; x_2)) \cdot \mathcal{H}^{\text{pwd}}_{n_2,\hat{n}_2}(\mathsf{enc}^{\mathsf{eg}}_{g^k}(\text{pwd}; x_1)), \mathcal{P})$.

Expt_4: In this experiment we drop the condition *if* $(pwd' \neq pwd)$ *then set* sk_1 *to random* in Step 5, and always output as follows

$$h'_2 = (\frac{\gamma'_2}{\text{pwd}})^{\hat{n}_1/\mathsf{ssid}}(\rho'_2)^{n_1},\ h_1 = (\eta'_2)^{x_1}; \text{set } \text{sk}_1 = e(h'_2, \mathcal{P}) \cdot e(g^{x_1}, N'_2) \cdot e(K^{x_1}, \hat{N}'_2).$$

This is accomplished by a series of hybrid experiments, one for each (ssid, P_i), we employ the hash proof smoothness, as $\text{pwd}' \neq \text{pwd}$ implies the tuple c'_2 is not in the language, and hence h'_2 is anyway random and independent.

Expt_5: In this experiment we set sk_1 in the last step as $e(h'_2, \mathcal{P}) \cdot e(\eta_2'^{x_1}, \mathcal{P})$. This change is indistinguishable as the simulator is checking the validity of the NIZK proofs, and by simulation-soundness extractability.

Expt_6: In this experiment we can drop the extraction of N'_2 and $\hat{N}'_2$, as they are no longer needed, and further we drop step 3. Note that currently that step is computing sk_1 as $e(\mathcal{H}^{\text{pwd}}_{n_1,\hat{n}_1}(\mathsf{enc}^{\mathsf{eg}}_{g^k}(\text{pwd}; x_2)) \cdot \mathcal{H}^{\text{pwd}}_{n_2,\hat{n}_2}(\mathsf{enc}^{\mathsf{eg}}_{g^k}(\text{pwd}; x_1)), \mathcal{P})$, but since $\eta'_2 = \eta_2$, and $c'_2 = c_2$ for this session, then the above expression is same as $e(h'_2, \mathcal{P}) \cdot e(\eta_2'^{x_1}, \mathcal{P})$. We replace all simulator generated proofs by proofs generated by real prover, and switch from the CRS generated by SE_1 to the real world CRS. Experiment Expt_6 is indistinguishable from the real-world experiment by completeness of the hash proof system, i.e. when the labeled tuple c, ssid is in the language, then the hash can be computed from the projection keys and the witness x_1 of c. This completes the proof of Theorem 4. □

Acknowledgments. The authors would like to thank the referees for several helpful comments.

References

1. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated Key Exchange Secure against Dictionary Attacks. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 139–155. Springer, Heidelberg (2000)
2. Bellovin, S.M., Merritt, M.: Encrypted key exchange: Password-based protocols secure against dictionary attacks. In: 1992 IEEE Symposium on Security and Privacy, pp. 72–84. IEEE Comp. Soc. Press (May 1992)
3. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
4. Camenisch, J., Chandran, N., Shoup, V.: A Public Key Encryption Scheme Secure against Key Dependent Chosen Plaintext and Adaptive Chosen Ciphertext Attacks. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 351–368. Springer, Heidelberg (2009)
5. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: 42nd FOCS, pp. 136–145. IEEE Comp. Soc. Press (October 2001)
6. Canetti, R., Halevi, S., Katz, J., Lindell, Y., MacKenzie, P.: Universally Composable Password-Based Key Exchange. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 404–421. Springer, Heidelberg (2005)
7. Cramer, R., Shoup, V.: Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)
8. Damgård, I.B.: Towards Practical Public Key Systems Secure against Chosen Ciphertext Attacks. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 445–456. Springer, Heidelberg (1992)

9. Elkind, E., Sahai, A.: A unified methodology for constructing public-key encryption schemes secure against adaptive chosen-ciphertext attacks. Cryptology ePrint Archive: Report 2002/042
10. Gennaro, R., Lindell, Y.: A Framework for Password-based Authenticated Key Exchange. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 524–543. Springer, Heidelberg (2003)
11. Gong, L., Lomas, T.M.A., Needham, R.M., Saltzer, J.H.: Protecting poorly chosen secrets from guessing attacks. IEEE JSAC 11(5), 648–656 (1993)
12. Groth, J.: Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 444–459. Springer, Heidelberg (2006)
13. Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
14. Jutla, C., Roy, A.: Relatively-sound NIZKs and password-based key-exchange. Cryptology ePrint Archive, Report 2011/507 (2011), http://eprint.iacr.org/
15. Katz, J., Ostrovsky, R., Yung, M.: Efficient Password-Authenticated Key Exchange Using Human-Memorable Passwords. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 475–494. Springer, Heidelberg (2001)
16. Katz, J., Vaikuntanathan, V.: Round-Optimal Password-Based Authenticated Key Exchange. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 293–310. Springer, Heidelberg (2011)
17. Kiltz, E.: Chosen-Ciphertext Security from Tag-Based Encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006)
18. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: 22nd ACM STOC. ACM Press (May 1990)
19. Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: 40th FOCS, pp. 543–553. IEEE Comp. Soc. Press (1999)

A More Efficient Unbounded Simulation Sound NIZKs

In [4], an unbounded simulation sound NIZK scheme is given for bilinear groups, building on the Groth-Sahai NIZKs and using Cramer-Shoup like CCA2 encryption schemes under K-linear assumptions. In this section we show various general optimizations for that construction, and further optimizations for specific languages involving generalized Diffie-Hellman tuples.

The general optimizations can be summarized as follows.

1. The scheme in [4] uses a one-time signature scheme. However, since it also uses a labeled CCA2 encryption scheme, the one-time signature scheme can be dropped, and one can use the label in the CCA2 scheme to get the signature property.
2. The scheme in [4] allows the simulator to generate a CCA2 encryption of u^x (for trapdoor x) along with a proof, instead of the proof of the statement. In order for the Adversary to cheat, it must also produce such an encryption, which is impossible under CCA2. However, one notices that since the simulator knows u^x, instead of a normal encryption, the simulator can hide u^x with just the smooth hash.

We now give this optimized version under the SXDH-assumption for groups (G_1, G_2, G_T), with a $\mathbb{Z}_q$-bilinear map e. We will write the bilinear map $e(A, B)$ in infix notation as $A \cdot B$. The group operation will be written in additive notation.

Languages for the simulation-sound NIZK can be specified by equations (relations) of the form $\boldsymbol{x} \cdot \boldsymbol{A} = T$, where $\boldsymbol{x}$ are variables from $\mathbb{Z}_q$, $\boldsymbol{A}$ are constants from G_2, and T is a constant from G_T, and thus $\boldsymbol{x}$ serves as witness for a member of a language specified by $\boldsymbol{A}$ and T. Languages can also be specified by equations of the form $\boldsymbol{B} \cdot \boldsymbol{\mathcal{Y}} = T_1 \cdot T_2$, where $\boldsymbol{B}$ are elements from G_1, $\boldsymbol{\mathcal{Y}}$ are variables from G_2, and T_1 and T_2 are constants from G_1 and G_2 resp. One can also consider languages with multiple such relations of both kinds.

Note that languages for which Groth-Sahai NIWI proofs can be given are more general, including equations like $\boldsymbol{x} \cdot \boldsymbol{A} + \boldsymbol{b} \cdot \boldsymbol{\mathcal{Y}} = T$, as well as quadratic equations.

The uss-NIZK CRS will consist of the usual Groth-Sahai NIWI CRS for SXDH, along with $g, A{=}g^a, \mathbf{k} = g^{k_1} A^{k_2}, \mathbf{d}{=}g^{d_1} A^{d_2}, \mathbf{e}{=}g^{e_1} A^{e_2}$, and $\mathbf{h}{=}g^x, \mathbf{u}{=}g^u$, with $g \in G_1$, and $a, k_1, k_2, d_1, d_2, e_1, e_2, x, u$ chosen at random from $\mathbb{Z}_q$. One could alternatively choose these values from G_2. Let H be a collision resistant hash function.

Given a set of relations as above, along with satisfying variables, the prover does the following:

1. – For each equation of the kind $\boldsymbol{x} \cdot \boldsymbol{A} = T$, it generates a modified equation $\boldsymbol{x} \cdot \boldsymbol{A} = \delta \cdot T$, where δ is a new global integer variable.
   – Get modified equations of the form $\boldsymbol{B} \cdot \boldsymbol{\mathcal{Y}} + T_1 \cdot \mathcal{V} = 0$, where $\mathcal{V}$ is a new variable representing elements from G_2, along with an additional equation $\mathcal{V} + (\delta - 1) \cdot T_2 = 0$ [13].
   – Generate an additional quadratic equation $\delta(1 - \delta) = 0$.
2. Produce a Groth-Sahai NIWI proof for the above modified set of equations, with δ set to 1. Call this proof, which includes all commitments to original variables as well as δ and $\mathcal{V}$, as π_1. Also append the original statement to be proven in π_1.
3. Generate $\rho = g^w$, $\hat{\rho} = A^w$, with w chosen at random.
4. Produce a Groth-Sahai NIWI proof of the following statements (using the same commitment to δ as in step 2, and w', x' committed to zero): $\rho^{1-\delta} = g^{w'}$, $\hat{\rho}^{1-\delta} = A^{w'}$, $\mathbf{h}^{1-\delta} = g^{x'}$. Call this proof along with commitments to x', w' as π_2.
5. Set $b = \mathbf{u} \cdot (\mathbf{kde}^t)^w$, where $t = H(\rho, \hat{\rho}, \pi_1, \pi_2)$.
6. Produce a Groth-Sahai NIWI proof of the following statement (using the same commitment to δ as in step 2, and same commitment for w', x' as in Step 4): $b^{1-\delta} = \mathbf{u}^{x'} \cdot (\mathbf{kde}^t)^{w'}$. Call this proof π_3.
7. The uss-NIZK proof consists of $(\pi_1, \pi_2, \pi_3, \rho, \hat{\rho}, b)$.

The proof of zero-knowledge is similar to the construction in [4]. The proof of unbounded simulation sound extractability is also similar to as in [4] but using the CCA2 encryption scheme (and its proof) as described in Section 5.

It is noteworthy that the uss-NIZK CRS can just give the product of $\mathbf{k}$ and $\mathbf{d}$, and it follows that $\mathbf{k}$ can be deleted altogether from the scheme. The above can also be made a labeled unbounded simulation-sound extractable NIZK, by including the label in the collision-resistance hash computation t in step 5.

Note that it takes 14 extra group elements to convert an SXDH based NIZK proof into a uSS-proof using this construction (and 28 elements for a DLIN based construction) [13]. For the language in Section 8.1, the NIZK proof requires 18 group elements. In the full paper [14] we show a further optimization for this specific language, which saves another 3 group elements, resulting in a total of 29 group elements for a uss-NIZK proof for the language.

Multi-location Leakage Resilient Cryptography

Ali Juma[1,⋆], Yevgeniy Vahlis[2], and Moti Yung[3]

[1] Mozilla Corporation
ajuma@mozilla.com
[2] AT&T Security Research Center
evahlis@att.com
[3] Google and Columbia University
my123@columbia.edu

Abstract. Understanding and modeling leakage in the context of cryptographic systems (connecting physical protection of keys and cryptographic operation) is an emerging area with many missing issues and hard to understand aspects. In this work we initiate the study of leakage out of cryptographic devices when the operation is inherently replicated in *multiple locations*. This setting (allowing the adversary access to leakage at different locations) arises naturally in cases like protocols, where different parties activate the same cryptographic function, or in the case of a global service providers (like cloud operators) which need to replicate the cryptographic function to allow for accessible and responsive services. We specifically deal with the theoretical setting of "leakage resilient cryptography," (modeling leakage as a bound associated with algorithmic steps), and in the most general model of continual leakage on memory, randomness (and thus computation) with periods of operation and refresh of private keys between them.

We first investigate public-key cryptography, and construct a multi-location leakage resilient signature scheme (with unbounded number of locations) with optimal (i.e., total $n(1 - o(1))$ leakage) in a period, and $O(\log n)$ leakage during updates (n is the key size). The new crucial issue behind our scheme is how to maintain leakage at each location at the level of key leakage in the single location variant, even under parallel adaptive leakage at the different locations. We then construct a shared-symmetric-key authenticated session protocol that is resilient to leakage on both the sender and the receiver, and tolerates $O(\log n)$ bits of leakage per computation. We construct and utilize a single-location pseudorandom generator which is the first to tolerate continual leakage with only an efficient pseudorandom function as a primitive component. This protocol highlights the importance of protocol level "per message synchronization" against leakage adversaries. Interestingly, the construction is secure in spite of the entire randomness used in the refresh processes being publicly available.

1 Introduction

When a cryptographic function/ service is performed at more than one location, and an adversary attacks it, if the adversary has only black-box access to it

⋆ This work was done while at the University of Toronto.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 504–521, 2012.

and the scheme is stateless or state-synchronized (for correctness), then from security point of view, it does not seem to matter (i.e., and it does not require a new model) whether the scheme is operated from a single location or multiple ones (since the black box information revealed in a sequence of cryptographic application is insensitive to the location). However, when leakage is allowed to be part of the outputs, the adversary gets this added side-channel information [23] and as a result, may have different power, depending on whether the leakage is at a single location or if it comes from multiple locations.

The above observation is the motivation to this work, since the sensitivity to multiple location leakage is important to various systems settings, and we investigate this issue from the "leakage resistance cryptography" perspective. Note that multi location is natural when two parties in a protocol operate the same cryptographic service or when the same function is inserted in various devices (e.g., different cloud servers, different mobile devices within the same organization), etc.

The theme of this work is the design of secure cryptosystems under the existence of multiple locations. We consider both *public key systems*: in particular a signature scheme (while the methods we design may apply in more generality to encryption, etc.), and *symmetric key systems*: in particular session authentication protocols providing sender continual authentication to the receiver, based on shared pseudorandomness. We consider the model of continual leakage with no relaxation, i.e., where the leakage function is not only a result of computation but can be a function of the memory (state) and the randomness (i.e., the computation) as well. In these models, the parties need to go through periods of operation where leakage is given to the adversary and once the accumulated amount of leakage is large enough, the private keys are refreshed between periods (obviously if refresh is not possible, continued leakage may reveal over time the entire key bits, say one by one). We note that the above model is the strongest, compared with more limited types of leakage models that have been considered as well in the literature (such as: leakage in the presence of leakage-free components, leakage where "only computations leak," and only memory leakage (without leaking the randomness used in the cryptographic computation)).

1.1 Multi-location Leakage Resilient Signature

Since we consider continual leakage we have to make sure that we have a scheme where the total leakage in a period is only a fraction of the state (if the state is l bits long we can allow at most $l(1 - o(1))$ bits of information about the secret to be given to the adversary. Indeed, a few recent schemes have achieved such leakage in a period, allowing logarithmic leakage in the refresh process (e.g. [8,5]). In this version we concentrate on the signature scheme of [28].

In a multi-location setting, the scheme is replicated in various locations to allow better accessibility o the signing service, say, and necessarily these locations contain related key information (since the verification key is identical regardless of location). If the adversary collects enough information at different locations, each of which by itself is too small to break the key, the cumulative effect may

nevertheless be that the adversary is in possession of enough bits to break the key (e.g., may collect all bits of a single replicated key). Thus, in the naive solution (which allows very limited leakage per location), we may restrict the amount of leakage at each location to be smaller than the total allowed per single location divided by the number of locations. This approach as we will show works sometimes, but sometimes fails!

We then turn into the challenging problem of constructing direct multi-location leakage resilient scheme which allows large leakges per location. Our starting point is a recent signature secure against continual leakage, where the crux behind this scheme is the fact that a key within a period is hidden within a large set of keys and the leakage within a period is simulatable by leakage correlated with a random value rather than the key [28]. Then, the key is refreshed to another value within a large set of keys. A crucial point behind extending the signature to a multi location scheme which is refreshed at all locations periodically (when a bound on signatures at any single location is reached) is extension of the space from which private keys are drawn, so that multiple location leakage will also be simulatable by random values taking the two dimensions of variety of keys, namely, "periods" and "locations," into account when building the space of keys.

1.2 Multi-location Symmetric-Key Authentication

We next design a session authentication protocol from symmetric key whose goal is to continuously authenticate the sender to the receiver. A natural way of doing it is to base this on a *stream cipher* (i.e., a pseudorandom generator) which is run by both parties. Dziembowski and Pietrzak [12] and Pietrzak [31] gave leakage-resilient stream ciphers in the *only-computation-leaks* model (where the adversary gets a bounded size (logarithmic, in fact) leakage bits each time). Their seminal constructions use two pieces of memory connected by a public channel, and computation alternates between the two pieces. For an authenticated session we have two parties, a sender A and a receiver B, where A is sending message pieces to B, and we wish to ensure that an adversary cannot modify or reorder messages pieces, or insert message pieces of his own without this being detected by B. The adversary obtains leakage from both parties. The existing security definitions of leakage-resilient stream ciphers do not deal with this case at all. In fact, in the existing ciphers, an adversary that can cause parties A and B to "get out of sync" can attack the system and eventually learn the cipher's entire state. This suggests that we need a way to somewhat synchronize the stream cipher computations performed by the two parties.

Our construction, in turn, builds on Pietrzak's stream cipher construction [31], and uses a *single* piece of memory along with a source of strings that are chosen according to distribution of high min-entropy but are not kept secret (*public min-entropy* source), and are rather communicated between the parties. Our stream cipher uses a pseudo-random function generator $F_s : \{0,1\}^n \to \{0,1\}^{2n}$. The initial secret state is randomly chosen $K_0 \in \{0,1\}^n$. For each $i > 0$, the i-th output is produced and the state is updated. A authenticates the message using

the output, while B generates the next round high entropy string and sends it back for synchronization and update to A. Of course, the adversary controls the public channel and may insert strings of his choice (purporting to be sent by the other party) to induce a party to continue its computation; we show that such tampering by the adversary will be detected by B when it attempts to verify the authenticity of the message pieces he receives; the construction allows continual leakage of logarithmic bits per round.

Remark: We note that Dodis *et al* [8] present a signature scheme which is multi-location leakage resilient (Theorem 7.6 in [8]), although they do not explicitly call it such. However, their scheme does not allow the adversary to obtain leakage on the randomness of signing. In contrast, in this work we define and present a signature scheme with multi-location resilience to full leakage (including signing randomness), and provide several generic theorems for obtaining multi-location leakage bounds for schemes that were intended to support only a single location.

1.3 Related Work

Side channel attacks [23] have often been shown to have devastating effects on the security of cryptographic schemes (some recent attacks that specifically pertain to general memory leakage are described in e.g., [17,32,36], and others). As a result a significant effort has been wielded to design cryptographic schemes that provably withstand large classes of such attacks.

The influential theoretical works of Ishai, Sahai, and Wagner [19] and Micali and Reyzin [30] enable us to construct schemes under the "any computation, and only computation, leak information," model, which has led to many recent achievements. In contrast, *memory leakage* [1] (which, in some sense, can be traced to the original works of Shamir [34] and Rivest [33]) are produced as a function of the memory state itself. This type of leakage is orthogonal to computational leakage: an adversary can get memory leakage by probing memories even if the memories are not currently used in any computation (e.g., the cold-boot attacks [17]). For example, the scheme of [9,8] is secure against memory attacks (even continual), but assumes that the signing process leaks no information. The most general model allows *full leakage* which includes leakage both from processing and memory.

The most demanding case for designing digital signature schemes seems to be the case of adaptive and continual full leakage that is available to the adversary from both computational and memory sources (without protection of sub-steps of computations). However, till recently there are no known schemes which achieve a digital signature scheme in this adversarial setting in the standard model. and without further relaxations. All known schemes with full (memory and processing) leakage either did not have a key update algorithm and thus are not continual (cf., [22]), have a key update algorithm but require some restrictions (e.g., [3,2] which requires an additional leakage-free master key), or are based on the random oracle model (with a relaxation of the definition of a "time period") [5]. Faust *et al* [13] construct signature schemes resilient to continual leakage in

the only computation leaks model. Recently, two schemes have appeared with continual full leakage [28,4].

In the private key setting, Dziembowski and Pietrzak [12], and Pietrzak [31] describe the first stream ciphers resilient to continual leakage in the only computation leaks model. Our private key construction uses the works of [12,31] as a starting point. Many other schemes dealt with the case of designing pseudorandom generation [12,31,39,11].

Faust *et al* [14] give a general compiler using secure hardware that protects an arbitrary circuit against continual leakage that can be modeled as a shallow (AC^0) boolean circuit. Juma and Vahlis [20], and separately Goldwasser and Rothblum [16], give compilers that protect any algorithm against continual leakage (without complexity restrictions), using secure hardware. Recently, Dodis and Pietrzak [11] show how to build continual leakage resilient pseudorandom functions that are secure against non-adaptive leakage. Finally, Lewko *et al* [26,25] show how to achieve leakage resilient Identity Based Encryption (IBE) and super-logarithmic leakage on key updates in signatures, and Chow *et al* [7] show an efficient leakage resilient IBE. A separate line of work studies strong leakage resilience in information theoretic implementation settings [35].

Finally, we mention a parallel rich line of work on tamper resistant cryptography [6,27,24,29,18,15]. Here, an adversary has the ability to modify, rather than observe, the state of the cryptographic primitive. Tamper resistant schemes provide security guarantees, even when the secret state is transformed through a tampering function adversarially chosen from a large class of functions. We remark that the techniques seem to be quite different from the ones employed to achieve leakage resilience. Indeed, studying the relation between tamper resistance and leakage resistance is an important direction.

Roadmap. In Section 2 we present our definitions of multi-location continuous leakage resilient signatures, and constructions. In Section 3 we present a shared key authenticated session protocol that is resilient to continuous leakage, and in Section 4 we present our variant of the Dziembowsky-Pietrzak leakage resilient stream cipher [12].

Notation. We write PPT to denote Probabilistic Polynomial Time. When we wish to fix the random bits of a PPT algorithm M to a particular value, we write $M(x;r)$ to denote running M on input x and randomness r. We write $time_n(M)$ to denote the running time of algorithm M on security parameter n. We use $x \in_{\mathrm{R}} S$ to denote the fact that x is sampled according to a distribution S. Similarly, when describing an algorithm we may write $x \leftarrow_{\mathrm{R}} S$ to denote the action of sampling an element from S and storing it in a variable x. For a randomized algorithm M, we denote by $\mathsf{Rnd}[M]$ the space of its random coins. Namely, if M uses at most n_M random bits in any execution, then $\mathsf{Rnd}[M] = \{0,1\}^{n_M}$.

2 Multi-location Leakage Resilience in the Public Key Setting

In a multi-location setting, an adversary (or perhaps multiple colluding adversaries) simultaneously mount side channel attacks on multiple devices. When the devices contain unrelated data, and perform unrelated computations, such an attack should be viewed as a single side channel attack on each of the devices, since none of the other devices can provide any information that would help the attacker. The problem becomes much more serious when the devices contain correlated secret data. As an extreme case, consider a situation where multiple identical copies of a secret key are stored on several servers. Even if the cryptographic scheme where the key is used remained secure when the key is partially leaked, an adversary in a multi-location setting may be able to reconstruct the entire key from partial leakage from each of the locations. In this scenario, surely all reasonable security properties of the scheme can be broken. One possible approach to dealing with this apparent limitation is to restrict the *total* amount of leakage that the adversary can obtain across *all* copies of the key. Indeed, for some primitives (such as encryption) we show a straightforward reduction from multi-location leakage resilience with a bound on the total amount of leakage to single location leakage resilience. We note that, perhaps surprisingly, the same straightforward reduction fails for other primitives, such as signature schemes.

Note however that ideally, we want to obtain transformations where the leakage per location remains as large as the leakage bound in the single location setting. In Section 2.1 we give constructions of signature and encryption schemes where the leakage bound per location does not decrease with the number of different locations that maintain an equivalent copy of the key.

Overall, we note that extending multi-location leakage resilience to the continuous setting introduces several subtle challenges that do not appear when considering leakage from only a single location. Before describing these issues, we describe (informally) a generic transformation of a continuous leakage resilient signature scheme into a multi-location variant of itself. This will serve two purposes: firstly, it will help us illustrate the definitional issues that arise in multi-location continual leakage resilience. Secondly, our actual constructions and transformations can all be viewed as variants of the general approach that we describe here.

Consider a signature scheme that is resilient to continual leakage in the single location setting. As we have already discussed, such a scheme must have a key refresh procedure (otherwise, an adversary can eventually obtain the entire key). Moreover, suppose that the refresh procedure produces a new signing key chosen uniformly from the set of all valid keys that correspond to the public verification key that is generated once at the beginning. Now consider the following initialization procedure for an n-location signature scheme:

1. A public-private key pair (vk, sk) is generated using the key generation procedure of the single location scheme.

2. The key refresh procedure is used to produce n random signing keys $sk_1, \ldots, sk_n$, all corresponding to the verification key vk.
3. Location i receives key sk_i. Whenever location i receives a request to update the key, it runs the refresh algorithm on its own key sk_i.

Essentially, each location maintains an independently chosen random signing key for vk, and when the leakage bound is reached for that specific location, the key is randomized. When $n = 1$, this is exactly what happens in the single location setting (and thus for $n = 1$ the security of the scheme trivially follows from its single location security). Consider now what happens when $n > 1$: at first glance it may seem that, because the keys at the different locations are independently chosen, leakage from one location would be completely useless in attacking another location. This turns out to be false for multiple reasons.

2.1 Signature Schemes

A *signature scheme with key update* $\mathcal{SGN}$ consists of four algorithms Kg, Sig, Ver, and Update. The inputs and outputs of Kg, Sig, and Ver are the same as in standard signature schemes. Update takes as input a secret key and a public key and outputs a new element of the secret key space. $\mathcal{SGN} = (\mathsf{Kg}, \mathsf{Sig}, \mathsf{Ver}, \mathsf{Update})$ has to satisfy the following property:

- (**Correctness**) For any integers $n, m, i \geq 0$ and any message M, if we compute $(pk, sk_1^{(0)}, \ldots, sk_m^{(0)}) \leftarrow \mathsf{Gen}(1^\kappa, m)$, $sk_0 \leftarrow sk_i^{(0)}$, $sk_1 \leftarrow \mathsf{Update}_{pk}(sk_0)$, $\ldots$, $sk_n \leftarrow \mathsf{Update}_{pk}(sk_{n-1})$, and $\sigma \leftarrow \mathsf{Sig}(sk_n, M)$, $\mathsf{Ver}(pk, M, \sigma) = 1$ always holds.

We now define multi-location leakage resilience for signatures. Intuitively, the definition is a natural extension of the definitions of leakage resilient signatures that appeared in [28,5,8,4,10]. Intuitively, the adversary can submit signature queries and leakage queries that are directed at a specific location. For example, the adversary may submit a query that is interpreted as "Have the ith signer sign message m, and obtain side-channel information $f(sk_i, r)$, where r is the randomness used during signing, along with the resulting signature". The other types of queries are location specific signature queries without leakage (to allow longer periods between updates, as discussed in the introduction), and update queries. For update queries, we distinguish between synchronized updates where all locations refresh their keys simultaneously and unsynchronized updates where the adversary instructs the signer at some location i to refresh his key. Finally, the adversary's goal is to produce a valid signature of a message that he has not submitted for signing in any of his queries.

Experiment $\mathsf{ExpMLSIG}(1^n, \mathcal{A}, \mathcal{SIG})$*:*

Setup. The adversary submits an integer m, and the challenger runs $\mathsf{Gen}(1^n, m)$ to obtain a public verification key pk, and m location secret keys $sk_1, \ldots, sk_m$.

Queries. $\mathcal{A}$ submits queries of the following three types:

- Update queries.

  Unsynchronized setting. Update queries of the form (update, f, i) where f is a circuit satisfying $|f(sk_i, R)| \leq \rho_U(|sk_i| + |R|)$ for any R. If $L_i + |f(sk_i, R)| \leq \rho_M |sk_i|$ holds, the challenger chooses $R \xleftarrow{\$} \mathsf{Rnd}[\mathsf{Update}]$ randomly, computes $sk_i \leftarrow \mathsf{Update}_{pk}(sk_i, R)$, sends $f(sk_i, R)$ back to $\mathcal{A}$, and sets $L_i \leftarrow |f(sk_i, R)|$. Otherwise, the challenger aborts.

  Synchronized setting. Update queries of the form $(\mathsf{update}, f_1, \ldots, f_n)$ where $|f_i(sk_i, R)| \leq \rho_U(|sk_i| + |R|)$ for any R. If $L_i + |f(sk_i, R)| \leq \rho_U |sk_i|$ holds, the challenger chooses $R_1, \ldots, R_n \xleftarrow{\$} \mathsf{Rnd}[\mathsf{Update}]$ randomly, computes $sk_i \leftarrow \mathsf{Update}_{pk}(sk_i; R_i)$, sends $(f_i(sk_i, R_i))_{i=1}^n$ back to $\mathcal{A}$, and sets $L_i \leftarrow |f_i(sk_i, R_i)|$. Otherwise, the challenger aborts.
- Memory leak queries (leak, f, i), where f is a circuit. If $L_i + |f(sk_i)| \leq \rho_M |sk_i|$ holds, the challenger sends $f(sk_i)$ to adversary and resets $L_i \leftarrow L_i + |f(sk_i)|$. Otherwise, the challenger aborts.
- Signing queries (sig, M, f, i) where f is a circuit with $|f(sk_i, R)| \leq \rho_S(|sk_i| + |R|)$ for any (sk_i, R). The challenger chooses $R \leftarrow \mathsf{Rnd}[\mathsf{Sig}]$ randomly, computes $\sigma \leftarrow \mathsf{Sig}(sk_i, M; R)$ and sends $(\sigma, f(sk_i, R))$ back to $\mathcal{A}$.

Challenge. Assuming the challenger did not abort, $\mathcal{A}$ outputs (M_*, σ_*). It succeeds if $\mathsf{Ver}(pk, M_*, \sigma_*) = 1$ holds and $\mathcal{A}$ never made query (sig, M_*, i) for any i.

Definition 1. Let ρ_G, ρ_U, ρ_M, and ρ_S be elements of the real range $[0, 1]$. We say that $\mathcal{SGN} = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver}, \mathsf{Update})$ is $(\rho_G, \rho_U, \rho_M, \rho_S)$- *EU-CMA-CML secure* (stand for existentially unforgeable under chosen message attack in the CML model) if no PPT adversary $\mathcal{A}$ succeeds in the experiment of $\mathsf{ExpMLSIG}$ with non-negligible probability. Here $\mathsf{Rnd}[\mathsf{Algo}]$ denote the set of randomnesses for algorithm Algo.

In the full version of this paper [21] we show several negative results regarding generic transformations of single to multi location signature schemes, as well as a simple transformation that does work, under some restrictions on the base signature scheme. We now turn to a direct construction that achieves optimal leakage bounds.

Direct Multi-Location Leakage Resilience. The simple generic transformation (described in [21]) may not be satisfactory if the number of locations is very large. For instance, for a key of length 256 bits, even an optimally leakage resilient scheme that is transformed to a multi-location setting with more than 256 locations would be able to withstand less than one bit of leakage per location before the key has to be refreshed. This would require an extremely high refresh rate if even a small (but unknown) number of locations are suspected to leak information.

To address this, we turn to constructing signature and encryption schemes directly, that will withstand large amounts of leakage per location, and will allow the total amount of leakage among different locations to exceed the length of the key between updates. At the core of our constructions is a strengthening of the Leakage Resilient Subspaces Lemma from [5]. On a high level, the BKKV

lemma can be described as follows: let $\mathcal{K}$ be an n-dimensional vector space, and let $Z_1, \ldots, Z_l$, $1 \leq l < k$ be random elements in $\mathcal{K}$. Then, no adversary (even a computationally unbounded one) can distinguish between leakage from random samples from $\mathsf{Span}(Z_1, \ldots, Z_l)$ and random samples from $\mathcal{K}$. The key difference between the lemma in [5] and the one we present here is the ability of the adversary to leak on samples in *parallel* rather than sequentially: we show that even if the adversary breaks his leakage on a given sample into several rounds, where at each round he chooses the leakage function adaptively based on leakage from other samples, he is still unable to distinguish between random samples from $\mathcal{K}$ and from the l dimensional subspace. We next describe the parallel leakage resilient subspace game:

Parallel-leakage resilient subspaces. Let $b \in \{0,1\}$, n, ℓ, m, λ be integers satisfying $n \geq \ell > m \geq 2$, p be a prime, and $\mathcal{K}$ be a n-dimensional vector space over $\mathbb{Z}_p$. The following is the parallel leakage resilient subspace game, played with a computationally unbounded adversary $\mathcal{D}$:

1. Let $Z_1, \ldots, Z_\ell \xleftarrow{\$} \mathcal{K}$. Initially a set $\Gamma = \{\Gamma_1, \ldots, \Gamma_m\}$ of size m is sampled uniformly at random from $\mathcal{K}$ if $b = 1$ and from $\mathsf{Span}(Z_1, \ldots, Z_\ell)$ if $b = 0$.
2. The adversary can make leakage queries: (leak, i, F) where $i \in [m]$ and $F : \mathcal{K} \to \{0,1\}^{\lambda_F}$, $\mathcal{P}_F \subseteq \mathcal{P}$; and refresh queries: $\mathsf{refresh}$. For a leakage query, the adversary is given $F(\Gamma_i)$, as long as $\sum_F \lambda_F \leq \lambda$ where the sum is over all the leakage functions F that are applied to Γ_i between two refresh queries. When the adversary submits a $\mathsf{refresh}$ query, $(\Gamma_1, \ldots, \Gamma_m)$ are assigned a random values from $\mathcal{K}$ if $b = 1$ and random values from $\mathsf{Span}(Z_1, \ldots, Z_\ell)$ if $b = 0$.
3. Finally, $\mathcal{D}$ is given $Z_1, \ldots, Z_\ell$, and it outputs a bit b'.

We denote by $\mathsf{ExpLRS}(b, \mathcal{D})$ the above experiment with an adversary $\mathcal{D}$, and with the bit b specified as a parameter. The output of $\mathsf{ExpLRS}(b, \mathcal{D})$ is defined to be the output of $\mathcal{D}$ at the end of the experiment. We now state the central parallel leakage resilient subspaces lemma:

Lemma 1. *Let $\mathcal{D}$ be an adversary for the above game. Then, for all $\delta \geq 0$, if $2^{\lambda_{\text{total}}} \leq p^{\ell - m - 1}\delta^2/q^2$, then*

$$|\Pr[\mathsf{ExpLRS}(0, \mathcal{D}) = 0] - \Pr[\mathsf{ExpLRS}(1, \mathcal{D}) = 0]| \leq \delta.$$

The proof of Lemma 1 appears in the full version of this paper [21].

2.2 Construction

We present a simple adaptation of the signature scheme of [28] to the multi-location setting. The modified construction allows us to achieve optimal leakage resilience, even when multiple versions of the key leak simultaneously. That is, the total amount of leakage across all locations between updates significantly exceeds the length of a single complete key (this is in contrast to the simple

generic transformation, where the amount of leakage per location decreases as the number of locations increases to guarantee that the total amount does not exceed the size of a key). The changes required to the scheme and analysis are quite minimal. Indeed, the only substantial modification to the analysis is the use of the parallel leakage-resilient subspaces lemma (Lemma 1). For completeness, we describe the complete scheme here, and give a high level overview of the necessary modifications to the security analysis.

Our construction relies on the Symmetric External DDH assumption in bilinear groups (details of the assumption are given in [21]). The description of our scheme is as follows. Let $n \geq 3$ and m be integers. Let Setup be a polytime algorithm that generates a group description $gk = (p, \mathbb{G}, \mathbb{H}, \mathbb{T}, \mathsf{e})$, as discussed above, where $\mathsf{e} : \mathbb{G} \times \mathbb{H} \to \mathbb{T}$. For $\mathcal{H} = (\boldsymbol{H}_0, \boldsymbol{H}_1, \ldots, \boldsymbol{H}_m) \in (\mathbb{H}^2)^{m+1}$ and $M \in \{0,1\}^m$, we define a Water's hash function [38] h as

$$h_{gk}(\mathcal{H}, M) = \boldsymbol{H}_0 + \sum_{k \in [m]} M_k \boldsymbol{H}_k,$$

where M_k is the k-th bit of M. Let Prf and Vrf be the proof algorithm and the verification algorithm of the Groth-Sahai proof system (reviewed in [21]). Our signature scheme $\mathcal{SGN} = (\mathsf{Kg}, \mathsf{Update}, \mathsf{Sig}, \mathsf{Ver})$ works as follows.

Key Generation $\mathsf{Gen}(1^\kappa, m)$**:** $gk \leftarrow (p, \mathbb{G}, \mathbb{H}, \mathbb{T}, \mathsf{e}) \leftarrow \mathsf{Setup}(1^\kappa)$, $\boldsymbol{G} \leftarrow \mathbb{H}^2$, $\mathcal{H} \leftarrow (\boldsymbol{H}_0, \boldsymbol{H}_1, \ldots, \boldsymbol{H}_m) \leftarrow (\mathbb{H}^2)^{m+1}$.
Randomly select $A \xleftarrow{\$} \mathbb{G}$, $Q \xleftarrow{\$} \mathbb{H}$, and $\boldsymbol{a}, \boldsymbol{q} \xleftarrow{\$} \mathbb{Z}_p^n$ satisfying $\langle \boldsymbol{a}, \boldsymbol{q} \rangle = 0$ and compute $\boldsymbol{A} \leftarrow \boldsymbol{a}A$, $\boldsymbol{Q} \leftarrow \boldsymbol{q}Q$. Select $\boldsymbol{W}^{[0]} \xleftarrow{\$} \mathbb{H}^n$ randomly, compute $T \leftarrow \mathsf{e}(\boldsymbol{A}, \boldsymbol{W}^{[0]})$. Then, the location specific keys are generated as: choose $s_i \xleftarrow{\$} \mathbb{Z}_p$, and set $\boldsymbol{W}_i^{[0]} \leftarrow \boldsymbol{W}^{[0]} + s_i \boldsymbol{Q}$. Outputs $pk \leftarrow (gk, \boldsymbol{G}, \mathcal{H}, \boldsymbol{A}, T, \boldsymbol{Q})$ and location specific private keys $(sk_i^{[0]})_{i \in [m]}$.

Key Update $\mathsf{Update}_{pk}(sk^{[i]})$**:** Parse pk and $sk^{[i]}$ as $(gk, \boldsymbol{G}, \mathcal{H}, \boldsymbol{A}, T, \boldsymbol{Q})$ and $\boldsymbol{W}^{[i]}$ respectively, select $s \xleftarrow{\$} \mathbb{Z}_p$ randomly, and output $sk^{[i+1]} \leftarrow \boldsymbol{W}^{[i+1]} \leftarrow \boldsymbol{W}^{[i]} + s\boldsymbol{Q}$.

Signing $\mathsf{Sig}(sk^{[i]}, M)$ **for** $M \in \{0,1\}^m$**:** Parse pk and $sk^{[i]}$ as $(gk, \boldsymbol{G}, \mathcal{H}, \boldsymbol{A}, T, \boldsymbol{Q})$ and $\boldsymbol{W}^{[i]}$. Compute $\boldsymbol{H}_M \leftarrow h_{gk}(\mathcal{H}, M)$, set $crs_M \leftarrow (\boldsymbol{G}, \boldsymbol{H}_M)$, and $\sigma \leftarrow \mathsf{Prf}(gk, crs_M, (\boldsymbol{A}, T), \boldsymbol{W}^{[i]})$ and output σ.

Verification $\mathsf{Ver}(pk, M, \sigma)$**:** Parse pk as $(gk, \boldsymbol{G}, \mathcal{H}, \boldsymbol{A}, T, \boldsymbol{Q})$, compute $\boldsymbol{H}_M \leftarrow h_{gk}(\mathcal{H}, M)$, and set $crs_M \leftarrow (\boldsymbol{G}, \boldsymbol{H}_M)$. If $\mathsf{Ver}(gk, crs_M, (\boldsymbol{A}, T), \sigma) = 1$, output 1. Otherwise, output 0.

Theorem 2. *For any constants $c > 0$ and any $\gamma = \Theta(1/\sqrt{\kappa})$, the proposed scheme $\mathcal{SIG}$ is $(\rho_G, \rho_U, \rho_M, \rho_S)$-EU-CMA-CML secure under the SXDH assumption. Here*

$$(\rho_G, \rho_U, \rho_M, \rho_S) = \left(\frac{c \cdot \log k}{n \log p}, \frac{c \cdot \log k}{n \log p}, 1 - \frac{2+\gamma}{n}, 1 - \frac{2+\gamma}{n} \right).$$

We can achieve the fraction $1 - o(1)$ of leakage in signing and in memory by setting $n = \kappa$.

Overview of the modifications to the analysis of the MTVY scheme. Essentially the analysis of the above scheme proceeds similarly to the analysis of the original (single-location) variant of [28]. The main modification to the argument is to replace the use of the original leakage resilient subspace lemma from [5] with our parallel version, given by lemma 1. Specifically, in the proof of [28, Lemma 9] we replace the use of [28, Proposition 10] with our Lemma 1. Once we use Lemma 1, the subspace $\mathcal{W}$ remains information theoretically hidden from the adversary throughout the security, and therefore any successful forgery would yield a successful attack on the Independent Pre-Image Resistant Hash Function described in [28, Section 3]. We leave the full details of the analysis to the full version of this paper.

3 Authenticated Session Protocols

Next, we describe our definition and construction of a leakage resilient authenticated session protocol in the private key setting.

3.1 Security Definition

The intuitive goal of an authenticated session protocol involving two parties, the sender A and the receiver B, where A is sending message pieces $m_1, m_2, \ldots,$ to B, is that B can verify that the message pieces he receives are indeed those sent by A, in the same order. This should hold even when all message pieces m_i sent by A are adversarially chosen. Of course, the adversary has complete control of the public channel over which A and B are communicating. This means that he controls the timing and contents of all communication. In the leakage-resilient case, we strengthen the adversary by allowing him to obtain leakage on *both* parties. We are interested in the *continual* leakage setting, where the adversary obtains some bounded amount of leakage on each computation by each party but the total amount of leakage obtained by the adversary over the course of the execution of the protocol is unbounded. The leakage on each computation is computed by an adversarially-chosen function is applied to the inputs and randomness involved in the computation along with the *entire* state of the party performing the computation. This means that we do *not* rely on the only-computation-leaks assumption. In our case, we further strengthen the adversary by giving him all the entropy used by each party after the initial state. Equivalently, we require that A and B are deterministic but each have access to a (separate) source of *public min-entropy*; whenever a party obtains a string its source of high min-entropy strings, this string is also given to the adversary.

We begin by formally defining session protocols (we restrict our definition to protocols as ours with two flows per message, but the idea can be extended).

Definition 3. (Shared-private-key session protocol with public min-entropy) *A* shared-private-key session protocol with public min-entropy *(which we will henceforth simply refer to as a* session protocol*) consists of deterministic*

polytime algorithms EvalB_1 *(producing message from B),* EvalA *(receiving the emssage from B), and* EvalB_2 *(producing the message received from after evaluation), polynomials* $s_B(n)$, $\ell_B(n)$, $s_A(n)$, *and* $\ell_A(n)$, *and distribution ensembles* $\{Z_n^A\}$ *and* $\{Z_n^B\}$ *that satisfy the following properties for all* $n \in \mathbb{N}$*:*

1. Z_n^A *is a distribution over strings of length* $s_A(n)$ *such that* $\mathbf{H}_\infty(Z_n^A) \geq \log^2(n)$. *Similarly,* Z_n^B *is a distribution over strings of length* $s_B(n)$ *such that* $\mathbf{H}_\infty(Z_n^B) \geq \log^2(n)$.

2. EvalB_1 *takes as input* $K_B \in \{0,1\}^n$ *and* $r_B \in \{0,1\}^{s_B(n)}$, *and outputs* $\beta \in \{0,1\}^{\ell_B(n)}$ *and* $K'_B \in \{0,1\}^n$ *such that* β *has prefix* r_B.

   Informally, the strings K_B *and* K'_B *are the state of party B before and after it executes* EvalB_1, r_B *is the public min-entropy used by* EvalB_1, *and* β *is a flow from party B to party A.*

3. EvalA *takes as input* $K_A \in \{0,1\}^n$, $m \in \{0,1\}^n$, $\beta \in \{0,1\}^{\ell_B(n)}$, *and* $r_A \in \{0,1\}^{s_A(n)}$, *and outputs* $e \in \{0,1\}^{\ell_A(n)}$ *and* $K'_A \in \{0,1\}^n$ *such that* e *has prefix* r_A.

   Informally, the strings K_A *and* K'_A *are the state of party A before and after it executes* EvalA, m *is a message piece that party A would like to send to party B,* β *is a flow from party B to party A,* r_A *is the public min-entropy used by* EvalA, *and* e *is a flow from party A to party B.*

4. EvalB_2 *takes as input* $K_B \in \{0,1\}^n$, $r_B \in \{0,1\}^{s_B(n)}$, *and* $e \in \{0,1\}^{\ell_A(n)}$, *and outputs either* $m \in \{0,1\}^n$ *and* $K'_B \in \{0,1\}^n$ *or a special message* `Fail`.

   Informally, the strings K_B *and* K'_B *are the state of party B before and after it executes* EvalB_2, r_B *is the public min-entropy used by the immediately preceding run of* EvalB_1, e *is a flow from party A to party B, and* m *is a message piece received by party B.*

5. *For all* $K \in \{0,1\}^n$, *every polynomial* $p(n)$, *all* $r_{A,1}, r_{A,2}, \ldots, r_{A,p(n)} \in \{0,1\}^{s_A(n)}$, *all* $r_{B,1}, r_{B,2}, \ldots, r_{B,p(n)} \in \{0,1\}^{s_B(n)}$, *and all sequences of message pieces* $m_1, m_2, \ldots, m_{p(n)} \in \{0,1\}^n$, *if we define* $K_{A,0} = K_{B,0} = K$ *and, for* $1 \leq i \leq p(n)$, *we iteratively define* $K_{A,i}, K'_{B,i}, K_{B,i}, e_i, \beta_i, m'_i$ *in the following manner:*

$$
\begin{aligned}
(\beta_i, K'_{B,i}) &\leftarrow \mathsf{EvalB}_1(K_{B,i-1}, r_{B,i}) \\
(e_i, K_{A,i}) &\leftarrow \mathsf{EvalA}(K_{A,i-1}, m_i, \beta_i, r_{A,i}) \\
(m'_i, K_{B,i}) &\leftarrow \mathsf{EvalB}_2(K'_{B,i}, r_{B,i}, e_i)
\end{aligned}
$$

   then $m'_i = m_i$ *for all* $1 \leq i \leq p(n)$.

   Informally, this means that in the absence of an adversary, the message pieces output by party B are exactly those sent by party A, in the same order.

We now define the security experiment for leakage-resilient authenticated session protocols. The adversary will be a family of polynomial-size circuits $C = \{C_n\}$.

Letting $\lambda : \mathbb{N} \to \mathbb{N}$ be a function, we will say that an adversary C is $\lambda(n)$-bounded if the leakage functions produced by C_n over the course of the security experiment each have output length $\lambda(n)$. Fixing a session protocol

$$(\mathsf{EvalB}_1, \mathsf{EvalA}, \mathsf{EvalB}_2, s_B(n), \ell_B(n), s_A(n), \ell_A(n), \{Z_n^A\}, \{Z_n^B\})$$

a function $\lambda : \mathbb{N} \to \mathbb{N}$, a $\lambda(n)$-bounded adversary $C = \{C_n\}$, and $n \in \mathbb{N}$, the security experiment proceeds as follows.

A string $K \in \{0,1\}^n$ is randomly chosen. We define $K_{A,0} = K_{B,0} = K$. Then, C_n is allowed to run EvalA, EvalB_1, and EvalB_2 in the following manner. C_n may run these algorithms as many times as he wishes and in any order of his choice as long as for every $i > 0$, the $(i+1)$-st invocation of EvalB_1 does not occur before the i-th invocation of EvalB_2, and the i-th invocation of EvalB_2 does not occur before the i-th invocation of EvalB_1. (This restriction captures the fact that even though the adversary controls the public channel, party B will still alternate between executing EvalB_1 and executing EvalB_2.) We now describe what happens when the adversary C_n runs each algorithm.

- For $i > 0$, the i-th invocation of EvalB_1 proceeds as follows. C_n produces the description of a circuit $f_{B1,i} : \{0,1\}^n \times \{0,1\}^{s_B(n)} \to \{0,1\}^{\lambda(n)}$. Then, $r_{B,i} \leftarrow Z_n^B$ is chosen. Next, $(\beta_i, K'_{B,i}) \leftarrow \mathsf{EvalB}_1(K_{B,i-1}, r_{B,i})$ and $leak_{B1,i} \leftarrow f_{B1,i}(K_{B,i-1}, r_{B,i})$ are computed. Finally, C_n is given β_i and $leak_{B1,i}$.
- For $i > 0$, the i-th invocation of EvalA proceeds as follows. C_n produces $m_i \in \{0,1\}^n$ and $\beta'_i \in \{0,1\}^{\ell_B(n)}$, and the description of a circuit $f_{A,i} : \{0,1\}^n \times \{0,1\}^{s_A(n)} \to \{0,1\}^{\lambda(n)}$. Then, $r_{A,i} \leftarrow Z_n^A$ is randomly chosen. Next, $(e_i, K_{A,i}) \leftarrow \mathsf{EvalA}(K_{A,i-1}, m_i, \beta'_i, r_{A,i})$ and $leak_{A,i} \leftarrow f_{A,i}(K_{A,i-1}, r_{A,i})$ are computed[1]. Finally, C_n is given e_i and $leak_{A,i}$.
- For $i > 0$, the i-th invocation of EvalB_2 proceeds as follows. C_n produces a string $e'_i \in \{0,1\}^{\ell_A(n)}$ and the description of a circuit $f_{B2,i} : \{0,1\}^n \to \{0,1\}^{\lambda(n)}$. Then, $(m'_i, K_{B,i}) \leftarrow \mathsf{EvalB}_2(K'_{B,i}, r_{B,i}, e'_i)$ and $leak_{B2,i} \leftarrow f_{B2,i}(K'_{B,i})$ are computed[2]; if EvalB_2 outputs `Fail`, the experiment ends immediately. If the i-th invocation of EvalA has previously occurred and $m'_i = m_i$, C_n is given $leak_{B2,i}$; otherwise, the experiment ends immediately.

Say that the final invocation of EvalB_2 is the j-th invocation. Define $q_C(n)$ to be the probability that the j-th invocation of EvalB_2 does not output `Fail` and either EvalA has been invoked fewer than j times or $m'_j \neq m_j$.

Definition 4. (Leakage-resilient authenticated session protocol) *Let $\lambda : \mathbb{N} \to \mathbb{N}$ be a function. A session protocol is a* $\lambda(n)$-leakage-resilient authenticated

[1] It is not necessary to provide m_i or β'_i as inputs to $f_{A,i}$ since C_n chose these values himself and hence he can simply hardcode them into $f_{A,i}$ if he wishes.

[2] It is not necessary to provide $r_{B,i}$ to $f_{B2,i}$ since this was previously provided to C_n as the prefix of β_i, and it is not necessary to provide e'_i to $f_{B2,i}$ since C_n chose this value himself.

session protocol *if for every $\lambda(n)$-bounded adversary C as above, we have $q_C(n) \leq 1/n^d$ for all d and sufficiently large n.*

3.2 Our Construction

In our construction, only party B requires a source of public min-entropy. Accordingly, to simplify notation, we use Z_n rather that Z_n^B to denote the high min-entropy distribution used by B.

Given pseudo-random function generators $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{2n}$ and $F' : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$, and given a distribution ensemble $\{Z_n\}$ such that for all n, Z_n is a distribution over $\{0,1\}^n$ and $\mathbf{H}_\infty(Z_n) \geq \log^2(n)$, we construct a leakage-resilient authenticated session protocol SP as follows.

EvalB_1**:** On input (K_B, r_B), where $K_B \in \{0,1\}^n$ and $r_B \in \{0,1\}^n$, EvalB_1 lets $K'_B = K_B$ and $\beta = r_B$, and outputs (β, K'_B).

EvalA**:** On input (K_A, m, β), where $K_A, m \in \{0,1\}^n$ and $\beta \in \{0,1\}^n$, EvalA computes $K'_A||X_A \leftarrow F_{K_A}(\beta)$ (where $|K'_A| = |X_A| = n$) and $\alpha = F'_{X_A}(m)$, lets $e = \langle m, \alpha \rangle$, and outputs (e, K'_A).

EvalB_2**:** On input (K_B, r_B, e'), where $K_B \in \{0,1\}^n$, $r_B \in \{0,1\}^n$, and $e' \in \{0,1\}^{2n}$, EvalB_2 parses $\langle m', \alpha' \rangle \leftarrow e'$, computes $K'_B||X_B \leftarrow F_{K_B}(r_B)$ (where $|K'_B| = |X_B| = n$), and $\alpha = F'_{X_B}(m')$. If $\alpha' = \alpha$, EvalB_2 outputs (m', K'_B); otherwise, EvalB_2 outputs `Fail`.

It is not hard to see that SP satisfies the definition of a session protocol. The idea is that parties A and B both run a stream cipher (see Section 4) starting from the same key and using the same inputs, and use the i-th output X_i to compute a signature $F'_{X_i}(m_i)$ of the i-th message piece m_i.

Theorem 5. *For all $c > 0$, SP is a $c \log n$-leakage-resilient authenticated session protcol.*

For the details of the proof of Theorem 5 we direct the reader to [21].

4 Stream Cipher Construction

In this section, we present our modified version of Pietrzak's stream cipher. The main purpose of our construction is to prove Theorem 6, which in turn is used in the proof of Theorem 5. Our construction uses only a single piece of memory but requires a public source of min-entropy. We believe that the construction below and its analysis are of independent interest due to the involved analysis of the leakage resilient stream cipher with public randomness.

The construction. Let $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{2n}$ be a pseudo-random function. Let $\{Z_n\}$ be such that for all n, Z_n is a distribution over strings of length n and $\mathbf{H}_\infty(Z_n) \geq \log^2(n)$. The initial state is K_0, where $K_0 \in \{0,1\}^n$ is randomly chosen. For each $i > 0$, the i-th round consists of:

1. $R_i \leftarrow Z_n$ is chosen.
2. $K_i || X_i \leftarrow F_{K_{i-1}}(R_i)$.
3. The new state is K_i.

The adversary's interaction. Fix $c > 0$. A $(c \log n)$-bounded adversary interacts as follows. For each $i > 0$:

1. Before round i, the adversary outputs the description of a function $f_i : \{0,1\}^{2n} \to \{0,1\}^{c \log n}$.
2. After round i, the adversary sees $R_i, X_i, f_i(K_{i-1}, R_i)$.

4.1 Security Analysis

We begin by defining some notation.

For an adversary A, we will use $\texttt{real}_i$ to denote the adversary's view after the first i rounds along with the corresponding X_j. That is,

$$\texttt{real}_i = \langle R_1, f_1(K_0, R_1), X_1, R_2, f_2(K_1, R_2), X_2, \ldots, R_i, f_i(K_{i-1}, R_i), X_i \rangle$$

Note that the f_j are not fixed functions, but rather are chosen adaptively by the adversary A as described in Section 4.

We will also define a version of $\texttt{real}_i$ that includes an additional round where there is no leakage. Specifically, we define

$$\texttt{real}_i^+ = \langle \texttt{real}_i, R_{i+1}, K_{i+1}, X_{i+1} \rangle$$

That is, $\texttt{real}_i^+$ includes the inputs and outputs of an additional leak-free round along with the entire state at the end of that round.

Theorem 6. *Let $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{2n}$ be a pseudo-random function. Let K' and X' be independent random variables that are each uniformly distributed over $\{0,1\}^n$. Let $\{Z_n\}$ be such that for all n, Z_n is a distribution over strings of length n and $\mathbf{H}_\infty(Z_n) \geq \log^2(n)$. For all $c > 0, d > 0, e > 0$, every function $p : \mathbb{N} \to \mathbb{N}$, sufficiently large n, all $(c \log n)$-bounded adversaries A interacting as described in section 4 and obtaining leakage for $p(n)$ rounds, and all adversaries D such that $2 \cdot \texttt{size}(A) + \texttt{size}(D) + p(n)\texttt{size}(F) \leq n^e$,*

$$\left| \Pr\left[D(\texttt{real}^+_{p(n)}) = 1 \right] - \Pr\left[D(\texttt{real}_{p(n)}, R_{p(n)+1}, K', X') = 1 \right] \right| \leq \frac{6p(n)+6}{n^d}.$$

Specifically, for sufficiently large n (depending only on c, d, and e), if there exists an adversary D breaking the above, then there exists an adversary of size $n^{e+8d+2c+8}$ breaking F with advantage $1/n^{5d+2c+3}$.

The details of the proof of Theorem 6 are given in [21]. The high-level approach is similar to that of Pietrzak [31], but there are differences in the details, due to the differences in our security models.

References

1. Akavia, A., Goldwasser, S., Vaikuntanathan, V.: Simultaneous Hardcore Bits and Cryptography Against Memory Attacks. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 474–495. Springer, Heidelberg (2009)
2. Alwen, J., Dodis, Y., Naor, M., Segev, G., Walfish, S., Wichs, D.: Public-Key Encryption in the Bounded-Retrieval Model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 113–134. Springer, Heidelberg (2010)
3. Alwen, J., Dodis, Y., Wichs, D.: Leakage-Resilient Public-Key Cryptography in the Bounded-Retrieval Model. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 36–54. Springer, Heidelberg (2009)
4. Boyle, E., Segev, G., Wichs, D.: Fully Leakage-Resilient Signatures. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 89–108. Springer, Heidelberg (2011)
5. Brakerski, Z., Kalai, Y.T., Katz, J., Vaikuntanathan, V.: Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In: 51st Annual Symposium on Foundations of Computer Science, pp. 501–510. IEEE Computer Society Press (2010)
6. Choi, S.G., Kiayias, A., Malkin, T.: BiTR: Built-in Tamper Resilience. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 740–758. Springer, Heidelberg (2011), Cryptology ePrint Archive, Report 2010/503 (2010), http://eprint.iacr.org/
7. Chow, S.S.M., Dodis, Y., Rouselakis, Y., Waters, B.: Practical leakage-resilient identity-based encryption from simple assumptions. In: Al-Shaer, E., Keromytis, A.D., Shmatikov, V. (eds.) ACM CCS 2010: 17th Conference on Computer and Communications Security, pp. 152–161. ACM Press (October 2010)
8. Dodis, Y., Haralambiev, K., López-Alt, A., Wichs, D.: Cryptography against continuous memory attacks. In: 51st Annual Symposium on Foundations of Computer Science, pp. 511–520. IEEE Computer Society Press (2010)
9. Dodis, Y., Haralambiev, K., López-Alt, A., Wichs, D.: Efficient Public-Key Cryptography in the Presence of Key Leakage. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 613–631. Springer, Heidelberg (2010)
10. Dodis, Y., Lewko, A., Waters, B., Wichs, D.: Storing secrets on continually leaky devices. Cryptology ePrint Archive, Report 2011/369 (2011), http://eprint.iacr.org/; To appear in FOCS 2011
11. Dodis, Y., Pietrzak, K.: Leakage-Resilient Pseudorandom Functions and Side-Channel Attacks on Feistel Networks. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 21–40. Springer, Heidelberg (2010)
12. Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: 49th Annual Symposium on Foundations of Computer Science, pp. 293–302. IEEE Computer Society Press (October 2008)
13. Faust, S., Kiltz, E., Pietrzak, K., Rothblum, G.N.: Leakage-Resilient Signatures. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 343–360. Springer, Heidelberg (2010)
14. Faust, S., Rabin, T., Reyzin, L., Tromer, E., Vaikuntanathan, V.: Protecting Circuits from Leakage: the Computationally-Bounded and Noisy Cases. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 135–156. Springer, Heidelberg (2010)

15. Gennaro, R., Lysyanskaya, A., Malkin, T., Micali, S., Rabin, T.: Algorithmic Tamper-Proof (ATP) Security: Theoretical Foundations for Security against Hardware Tampering. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 258–277. Springer, Heidelberg (2004)
16. Goldwasser, S., Rothblum, G.N.: Securing Computation against Continuous Leakage. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 59–79. Springer, Heidelberg (2010)
17. J. Alex Halderman, Schoen, S.D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J.A., Feldman, A.J., Appelbaum, J., Felten, E.W.: Lest we remember: Cold boot attacks on encryption keys. In: van Oorschot, P.C. (ed.) USENIX Security Symposium, pp. 45–60. USENIX Association (2008)
18. Ishai, Y., Prabhakaran, M., Sahai, A., Wagner, D.: Private Circuits II: Keeping Secrets in Tamperable Circuits. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 308–327. Springer, Heidelberg (2006)
19. Ishai, Y., Sahai, A., Wagner, D.: Private Circuits: Securing Hardware against Probing Attacks. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 463–481. Springer, Heidelberg (2003)
20. Juma, A., Vahlis, Y.: Protecting Cryptographic Keys against Continual Leakage. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 41–58. Springer, Heidelberg (2010)
21. Juma, A., Vahlis, Y., Yung, M.: Multi-Location Leakage Resilient Cryptography. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 504–521. Springer, Heidelberg (2012), http://eprint.iacr.org/
22. Katz, J., Vaikuntanathan, V.: Signature Schemes with Bounded Leakage Resilience. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 703–720. Springer, Heidelberg (2009)
23. Kocher, P.C.: Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996)
24. Komano, Y., Ohta, K., Miyake, H., Shimbo, A.: Algorithmic Tamper Proof (ATP) Counter Units for Authentication Devices Using PIN. In: Abdalla, M., Pointcheval, D., Fouque, P.-A., Vergnaud, D. (eds.) ACNS 2009. LNCS, vol. 5536, pp. 306–323. Springer, Heidelberg (2009)
25. Lewko, A., Lewko, M., Waters, B.: How to leak on key updates. Cryptology ePrint Archive, Report 2010/562 (2010), http://eprint.iacr.org/
26. Lewko, A., Rouselakis, Y., Waters, B.: Achieving Leakage Resilience through Dual System Encryption. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 70–88. Springer, Heidelberg (2011)
27. Liu, F.-H., Lysyanskaya, A.: Algorithmic Tamper-Proof Security under Probing Attacks. In: Garay, J.A., De Prisco, R. (eds.) SCN 2010. LNCS, vol. 6280, pp. 106–120. Springer, Heidelberg (2010)
28. Malkin, T., Teranishi, I., Vahlis, Y., Yung, M.: Signatures Resilient to Continual Leakage on Memory and Computation. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 89–106. Springer, Heidelberg (2011)
29. Mateus, P., Vaudenay, S.: On Tamper-Resistance from a Theoretical Viewpoint. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 411–428. Springer, Heidelberg (2009)
30. Micali, S., Reyzin, L.: Physically Observable Cryptography (Extented Abstract). In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 278–296. Springer, Heidelberg (2004)

31. Pietrzak, K.: A Leakage-Resilient Mode of Operation. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 462–482. Springer, Heidelberg (2009)
32. Ristenpart, T., Tromer, E., Shacham, H., Savage, S.: Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In: Al-Shaer, E., Jha, S., Keromytis, A.D. (eds.) ACM CCS 2009: 16th Conference on Computer and Communications Security, pp. 199–212. ACM Press (November 2009)
33. Rivest, R.L.: All-or-Nothing Encryption and the Package Transform. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 210–218. Springer, Heidelberg (1997)
34. Shamir, A.: How to share a secret. Communications of the Association for Computing Machinery 22(11), 612–613 (1979)
35. Standaert, F.-X., Malkin, T.G., Yung, M.: A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 443–461. Springer, Heidelberg (2009)
36. Tromer, E., Osvik, D.A., Shamir, A.: Efficient cache attacks on AES, and countermeasures. Journal of Cryptology 23(1), 62–74 (2010)
37. Veyrat-Charvillon, N., Standaert, F.-X.: Generic Side-Channel Distinguishers: Improvements and Limitations. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 354–372. Springer, Heidelberg (2011)
38. Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
39. Yu, Y., Standaert, F.-X., Pereira, O., Yung, M.: Practical leakage-resilient pseudorandom generators. In: Al-Shaer, E., Keromytis, A.D., Shmatikov, V. (eds.) ACM CCS 2010: 17th Conference on Computer and Communications Security, pp. 141–151. ACM Press (October 2010)

On Definitions of Selective Opening Security

Florian Böhl, Dennis Hofheinz, and Daniel Kraschewski

Karlsruhe Institute of Technology
{Florian.Boehl,Dennis.Hofheinz,Daniel.Kraschewski}@kit.edu

Abstract. Assume that an adversary observes many ciphertexts, and may then ask for openings, i.e. the plaintext and the randomness used for encryption, of some of them. Do the unopened ciphertexts remain secure? There are several ways to formalize this question, and the ensuing security notions are not known to be implied by standard notions of encryption security. In this work, we relate the two existing flavors of selective opening security. Our main result is that indistinguishability-based selective opening security and simulation-based selective opening security do not imply each other.

We show our claims by counterexamples. Concretely, we construct two public-key encryption schemes. One scheme is secure under selective openings in a simulation-based sense, but not in an indistinguishability-based sense. The other scheme is secure in an indistinguishability-based sense, but not in a simulation-based sense.

Our results settle an open question of Bellare et al. (Eurocrypt 2009). Also, taken together with known results about selective opening secure encryption, we get an almost complete picture how the two flavors of selective opening security relate to standard security notions.

Keywords: security definitions, selective opening security, public-key encryption.

1 Introduction

Security under Selective Openings. Assume that an adversary observes many ciphertexts, and may then ask for openings of some of them. Do the unopened ciphertexts remain secure? Somewhat surprisingly, security in this setting is not known to be implied by standard security notions for encryption schemes (such as IND-CPA security). In fact, very recently, Bellare et al. [2] showed that a whole class of IND-CPA secure public-key encryption schemes do not achieve a simulation-based notion of selective open security.

To date, there are two flavors of definitions to capture security under selective openings: simulation-based selective opening security (SIM-SO security, [8, 1]) and indistinguishability-based selective opening security (IND-SO security, [1]). There are indications that SIM-SO and IND-SO security constitute orthogonal requirements. For instance, when looking at selective opening security for commitment schemes, Bellare et al. prove that any statistically hiding commitment scheme is IND-SO secure; however, there are severe limitations on the

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 522–539, 2012.

construction of SIM-SO secure commitment schemes from a number of standard assumptions [1]. Nonetheless, in case of encryption schemes (which are the focus of this paper), no similar result is known.

We will now describe the existing security notions for selective opening security, along with known results.

Simulation-Based Selective Opening Security (SIM-SO-CPA). An encryption scheme is called SIM-SO-CPA secure, if anything an adversary can compute from a vector of ciphertexts *and* openings of a subset of these ciphertexts, can also be computed by a simulator that only sees the opened messages (but no ciphertexts at all). SIM-SO-CPA security dates back to Dwork et al. [8], who consider the same security notion for commitments. In the encryption context, SIM-SO-CPA security has been investigated by Bellare et al. [1], who also observe that Goldasser-Micali encryption [10] achieves SIM-SO-CPA security. Later on, several other SIM-SO-CPA secure encryption schemes have been constructed [9, 13, 14].

However, all known SIM-SO-CPA secure encryption schemes are comparatively inefficient: they either encrypt messages bitwise [10, 9], or they are based on assumptions related to Paillier encryption [13, 14]. There is no known efficient SIM-SO-CPA secure encryption scheme based on, say, the DDH problem in a suitable cyclic group. One key difficulty seems to be that SIM-SO-CPA security essentially requires that the encryption is non-committing, such that a ciphertext can be efficiently opened to any message [4, 5, 9] (possibly using a special trapdoor). In fact, Bellare et al. [2] use this property in a clever way to construct an encryption scheme that is IND-CPA secure, but not SIM-SO-CPA secure.

Indistinguishability-Based Selective Opening Security (IND-SO-CPA). An encryption scheme is called IND-SO-CPA secure, if no adversary, after given a vector of ciphertexts and openings of a subset of these ciphertexts, can distinguish the unopened messages from fresh messages. There is one subtlety here. Namely, in most applications, the initially received ciphertext vector may contain encryptions of related messages (e.g., encryptions of shares of a secret value). Hence, the "fresh" messages that the adversary must distinguish from the actually encrypted (but unopened) messages must be *conditioned* on the already opened messages. Note that depending on the underlying distribution of message vectors, conditioning on an arbitrary subset of messages can be an inefficient process. In particular, the IND-SO-CPA security experiment may be inefficient.

This subtlety has led to two different IND-SO-CPA variations. *Full IND-SO-CPA* security requires exactly what we have sketched above, with a potentially inefficient security experiment. The problem with full IND-SO-CPA security is that there are no known fully IND-SO-CPA secure encryption schemes.[1]

[1] We mention that for *commitments*, the situation is less problematic: every statistically commitment scheme is (fully) IND-SO secure [1]. However, a moment of reflection shows that there can be no statistically hiding *encryption* scheme. The closest we can get to statistically hiding encryption is lossy encryption, which is only known to imply *weak* IND-SO-CPA security.

On the other hand, *weak IND-SO-CPA* security requires the above, but only for distributions of message vectors that are efficiently re-samplable. Here, efficiently re-samplable means that the message distribution can be efficiently sampled, even when fixing any subset of messages to a particular value.[2] The advantage of weak IND-SO-CPA security is that any lossy encryption scheme [17] is already weakly IND-SO-CPA secure [1]. In particular, there are very efficient weakly IND-SO-CPA secure encryption schemes based on standard assumptions. This is also an important advantage over full IND-SO-CPA security for which no realizations are known yet.

The main disadvantage of weak IND-SO-CPA security is that it is obviously only useful in settings in which the joint distribution of all encrypted messages actually is efficiently re-samplable. Many conceivable settings (e.g., when commitments or other non-invertible functions of other messages are encrypted) do not conform to such a re-samplability condition.

The Current Situation. So far, we can summarize that SIM-SO-CPA as well as (full or weak) IND-SO-CPA security both have advantages and disadvantages. It depends on the concrete setting and requirements which notion is to prefer. However, so far little is known about the *relations* among those notions of selective opening security. A few implications are trivial or at least follow with little effort: full IND-SO-CPA security obviously implies weak IND-SO-CPA security, and it is not hard to see that SIM-SO-CPA security implies weak IND-SO-CPA security. However, otherwise the relation in particular between full IND-SO-CPA security and SIM-SO-CPA security is not known. (We again stress that for *commitments*, the situation is a little different, as sketched above; however, these results do not apply to encryption schemes.)

Our Contribution. This paper attempts to fill this gap: we relate full IND-SO-CPA security and SIM-SO-CPA security. Our results show that full IND-SO-CPA security does not imply SIM-SO-CPA security, and vice versa. We give concrete counterexamples, i.e., encryption schemes that are fully IND-SO-CPA secure, but not SIM-SO-CPA secure (and the other way around). In a sense, our results further isolate full IND-SO-CPA security from other notions of encryption scheme security. Thus, there is even less motivation to study full IND-SO-CPA security. Figure 1 depicts the relations of the different flavors of selective opening security to one another and to IND-CPA security.

We now provide some more technical background on our results.

Our First Counterexample. We first construct a scheme that is SIM-SO-CPA secure, but not fully IND-SO-CPA secure. The basic idea is to take any SIM-SO-CPA secure scheme, and modify it such that it becomes vulnerable to a full IND-SO-CPA attack (while preserving its SIM-SO-CPA security, of course). Our modification is simple: we add a tuple

$$((g^s u^t)^M, (h^s v^t)) \tag{1}$$

[2] For instance, the distribution of message tuples (x, x) is efficiently re-samplable, while the distribution (x, g^x) is not (where $x \in \mathbb{Z}_{|\mathbb{G}|}$ is uniform, and $g \in \mathbb{G}$ for some group $\mathbb{G}$ in which discrete logarithms are hard to compute).

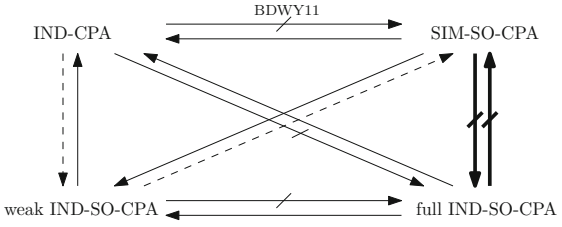


Fig. 1. Relations of different definitions of selective opening security and IND-CPA. The bold arrows illustrate the results of our work while BDWY11 is the main result of [2]. Crossed arrows symbolize concrete counterexamples and dashed arrows stand for open questions. All other arrows are implications that are pretty much straightforward or follow directly from the already settled relations. Note that the question whether weak IND-SO-CPA security implies SIM-SO-CPA security is settled negatively if a fully IND-SO-CPA secure encryption scheme exists.

to each ciphertext, where M is the encrypted message, s, t are random exponents, and g, h, u, v are group elements that are part of the public key. In the scheme, $(g, h, u, v) = (g, h, g^\omega, h^\omega)$ is a Diffie-Hellman tuple, such that (1) is a perfectly binding commitment to M. However, during the proof that the modified scheme is still SIM-SO-CPA secure, we will switch (g, h, u, v) to a non-Diffie-Hellman tuple. Then, (1) becomes a perfectly hiding commitment, which can actually be equivocated arbitrarily. (Note that this added commitment really only is an instance of the dual-mode commitment schemes from Damgård and Nielsen [6].) This allows to open ciphertexts in our modified scheme arbitrarily, and shows the modified scheme SIM-SO-CPA secure.

To prove that the modified scheme is not fully IND-SO-CPA secure, we first define a suitable distribution dist of message tuples (x, z), such that re-sampling dist essentially requires computing a discrete logarithm. Concretely, we define dist such that $(x, z) = (x, g^x)$, resp. $(x, z) = (x, h^x)$ (for uniform x and g, h from the scheme's public key) with probability $1/2$ each. Now suppose an adversary starts off with two ciphertexts, one for x and one for $z = g^x$. He then chooses to open the second ciphertext (for $z = g^x$), which fixes the second component of the ciphertext vector. (However, note that the adversary does not know x at this point.)

Assume, when invoked with the challenge message vector, he then gets a first component y, sampled from dist conditioned on the second component z. By our definition of dist, with a probability of $1/2$, the adversary then does not get $y = x$, but the unique y with $z = h^y$. Note that then, $x = y \cdot \mathrm{dlog}_g h$. Using this relation, the adversary can recognize that the first unopened ciphertext (with commitment $((g^s u^t)^x, (h^s v^t)))$ really contained x. This check works *only* if re-sampling occurred, and hence the adversary successfully distinguishes authentic from re-sampled messages. As SIM-SO-CPA security implies IND-CPA security, this counterexample also shows that IND-CPA security does not imply full IND-SO-CPA security.

Our Second Counterexample. We proceed to construct a scheme that is fully IND-SO-CPA secure, but not SIM-SO-CPA secure. Again, we simply modify an assumed fully IND-SO-CPA secure scheme to make a SIM-SO-CPA attack possible. Concretely, we add a statistically hiding commitment $\mathsf{Com}(M)$ to each ciphertext, where M is the encrypted message. (In fact, we will require non-interactive statistically hiding commitments without any kind of setup, which can be built from collision-resistant hash functions. See Section 4 for details.) This makes the encryption scheme binding (i.e., a public key and a ciphertext form a binding commitment to the message). Hence, applying a general result due to Bellare et al. [2] shows that the scheme is not SIM-SO-CPA secure.

To show that the modified scheme is still fully IND-SO-CPA secure, we show that any IND-SO-CPA adversary A' on the modified scheme can be mapped to an IND-SO-CPA adversary A on the old scheme. The problem for A is that it must present (an internal simulation of) A' with ciphertexts with added commitments $\mathsf{Com}(M_i)$, and later open some of those commitments to the right M_i. In this, A must not know any of the M_i in advance. Our solution to this commitment problem is to embed the $\mathsf{Com}(M_i)$ into A's message distribution. (That is, if A''s message distribution over the M_i is dist', then A's message distribution is dist, which is the same as dist', only with added commitments to the M_i.) Hence, A can go ahead and open all $\mathsf{Com}(M_i)$-encryptions (and only those) in advance to be able to prepare authentic commitments for A'. The remaining translation between A''s and A's IND-SO-CPA experiment is then straightforward.

The technical difficulty in pushing this line of proof through is that by initially opening commitments $\mathsf{Com}(M_i)$ to *all* messages, A may slightly skew a later re-sampling of the M_i. If the used commitment scheme is *perfectly* hiding, this is a non-issue: then, $\mathsf{Com}(M_i)$ reveals no information about M_i, and conditioning on $\mathsf{Com}(M_i)$ does not change the distribution of M_i. However, the most interesting candidates for non-interactive statistically hiding commitment schemes are only statistically, but not perfectly hiding. We thus need to show that conditioning on a statistically hiding commitment does not significantly change a message distribution. This in fact turns out to be surprisingly nontrivial. Specifically, the statement only holds for *bit* messages M_i, but not necessarily for messages, say, from $\{0,1\}^k$. See Section 4 for details.

Outline. We start by recalling some notation and definitions (including the definitions of selective opening security) in Section 2. We present our counterexamples in Section 3 and Section 4. In Appendix A, we prove a technical lemma that is necessary for our second counterexample.

2 Preliminaries

Notation. For $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$. Throughout the paper, $k \in \mathbb{N}$ denotes the security parameter. For a finite set $\mathcal{S}$, we denote by $s \leftarrow \mathcal{S}$ the process of sampling s uniformly from $\mathcal{S}$. For a distribution X, we denote by $x \leftarrow X$ the process of sampling x from X. For a probabilistic algorithm A, we denote with $y \leftarrow A(x; R)$ the process of running A on input x and with randomness R, and

assigning y the result. We let $\mathcal{R}_A$ denote the randomness space of A; we require $\mathcal{R}_A$ to be of the form $\mathcal{R}_A = \{0,1\}^r$. We write $y \leftarrow A(x)$ for $y \leftarrow A(x;R)$ with uniformly chosen $R \in \mathcal{R}_A$. If A's running time is polynomial in k, then A is called probabilistic polynomial-time (PPT). Two sequences of random variables $X = (X_k)_{k\in\mathbb{N}}$ and $Y = (Y_k)_{k\in\mathbb{N}}$ are *computationally indistinguishable* (denoted $X \overset{c}{\approx} Y$) iff for any PPT algorithm D, the probability $\Pr\left[D(1^k, X_k) = 1\right] - \Pr\left[D(1^k, Y_k)\right]$ is negligible in k. The statistical distance of X_k and Y_k is defined as $\mathsf{SD}\left(X_k\,;\,Y_k\right) := \frac{1}{2}\sum_s |\Pr\left[X_k = s\right] - \Pr\left[Y_k = s\right]|$.

DDH Assumption. The *decisional Diffie-Hellman (DDH) assumption* over a group $\mathbb{G}$ (that may depend on the security parameter k) stipulates that

$$(g, g^a, g^b, g^{ab}) \quad \overset{c}{\approx} \quad (g, g^a, g^b, g^c),$$

where $g \leftarrow \mathbb{G}$ and $a, b, c \leftarrow [|\mathbb{G}|]$ are uniformly distributed.

PKE Schemes. A public-key encryption (PKE) scheme consists of three PPT algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. Key generation $\mathsf{Gen}(1^k)$ outputs a public key pk and a secret key sk. Encryption $\mathsf{Enc}(pk, M)$ takes a public key pk and a message M, and outputs a ciphertext C. Decryption $\mathsf{Dec}(sk, C)$ takes a secret key sk and a ciphertext C, and outputs a message M. For correctness, we want $\mathsf{Dec}(sk, C) = M$ for all M, all $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$, and all $C \leftarrow \mathsf{Enc}(pk, M)$.

Definition of Selective Opening Security. We present and discuss three definitions for security under selective openings that capture security of an encryption scheme under adaptive attacks. Two definitions are indistinguishability-based, following the IND-SO-COM, resp. IND-SO-ENC definitions by Bellare et al. [1]. These definitions demand that even an adversary that gets to see a vector of ciphertexts cannot distinguish the true contents of the ciphertexts from independently sampled plaintexts. While one of these definitions, called weak IND-SO-CPA here, only considers *efficiently re-samplable* message distributions, the other one, dubbed full IND-SO-CPA, does not restrict the considered message distributions in this way. The third definition, dubbed SIM-SO-CPA by us, resembles the SEM-SO-COM, resp. SEM-SO-ENC definitions from [1] (which in turn follow Dwork et al. [8]). This definition is simulation-based and does not have to cope with different strategies to handle re-sampling.

Definition 1 (Efficiently re-samplable). *Let $N = N(k) > 0$, and let* dist *be a joint distribution over $(\{0,1\}^k)^N$. We say that* dist *is* efficiently re-samplable *if there is a PPT algorithm $\mathsf{ReSamp}_{\mathsf{dist}}$ such that for any $\mathcal{I} \subseteq [N]$ and any partial vector $\mathbf{M}'_{\mathcal{I}} := (M'^{(i)})_{i\in\mathcal{I}} \in (\{0,1\}^k)^{|\mathcal{I}|}$, $\mathsf{ReSamp}_{\mathsf{dist}}(\mathbf{M}'_{\mathcal{I}})$ samples from $\mathsf{dist} \mid \mathbf{M}_{\mathcal{I}}$, i.e., from the distribution dist, conditioned on $M^{(i)} = M'^{(i)}$ for all $i \in \mathcal{I}$.*

Opening Oracles. In our definitions of selective opening security we provide the adversary with an *opening oracle* to allow adaptive queries. Such an oracle is a stateful functionality that takes one argument. When queried with a set of indexes, it responds with the corresponding openings. When queried with the string `get queries`, it returns the set of all indexes it has provided openings for since its instantiation.

Definition 2 (Weak indistinguishability-based selective opening security). *For a PKE scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*, a polynomially bounded function* $N = N(k) > 0$*, an opening oracle* $\mathcal{O}$ *and a stateful PPT adversary* A*, consider the following experiment:*

Experiment $\mathsf{Exp}^{\mathsf{weak\text{-}ind\text{-}so}}_{\mathsf{PKE},A}$
$b \leftarrow \{0,1\}$
$(pk, sk) \leftarrow \mathsf{Gen}(1^k)$
$(\mathsf{dist}, \mathsf{ReSamp}_{\mathsf{dist}}) \leftarrow A(pk)$
$\mathbf{M}_0 := (M^{(i)})_{i\in[n]} \leftarrow \mathsf{dist}$
$\mathbf{R} := (R^{(i)})_{i\in[n]} \leftarrow (\mathcal{R}_{\mathsf{Enc}})^N$
$\mathbf{C} := (C^{(i)})_{i\in[n]} := (\mathsf{Enc}(pk, M^{(i)}; R^{(i)}))_{i\in[n]}$
$\mathbf{O} := (M^{(i)}, R^{(i)})_{i\in[n]}$
$A^{\mathcal{O}(\cdot)}(\mathtt{select}, \mathbf{C})$
$\mathcal{I} := \mathcal{O}(\mathtt{get\ queries})$
$\mathbf{M}_1 \leftarrow \mathsf{dist} \mid \mathbf{M}_{\mathcal{I}}$
$out_A \leftarrow A(\mathtt{output}, \mathbf{M}_b)$
return 1 if $out_A = b$, 0 otherwise

We only allow A *that always output efficiently re-samplable distributions* dist *over* $(\{0,1\}^k)^N$ *with corresponding efficient re-sampling algorithms* $\mathsf{ReSamp}_{\mathsf{dist}}$*. We say that* PKE *is* weakly IND-SO-CPA *secure, if*

$$\mathsf{Adv}^{\mathsf{w\text{-}ind\text{-}so}}_{\mathsf{PKE},A}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{weak\text{-}ind\text{-}so}}_{\mathsf{PKE},A}(k) = 1\right] - \frac{1}{2}$$

is negligible.

There are some minor technical differences between Definition 2 and the IND-SO-ENC definition from [1]: IND-SO-ENC security universally quantifies over all (efficiently re-samplable) message distributions dist. We let A choose dist instead, e.g., to allow a message distribution that depends on the public key pk. (In fact, otherwise it is not even clear that the resulting definition implies IND-CPA security.) Besides, our definition grants the adversary multiple, possibly adaptive openings, whereas IND-SO-ENC security only allows for a one-shot opening phase. We believe that multiple openings are more realistic in view of a scenario with adaptive party corruptions.

Definition 3 (Full indistinguishability-based selective opening security). *For a PKE scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*, a polynomially bounded function* $N = N(k) > 0$*, a stateful opening oracle* $\mathcal{O}$ *and a stateful PPT adversary* A*, we define the experiment* $\mathsf{Exp}^{\mathsf{full\text{-}ind\text{-}so}}_{\mathsf{PKE},A}$ *analogously to* $\mathsf{Exp}^{\mathsf{weak\text{-}ind\text{-}so}}_{\mathsf{PKE},A}$ *but do not require the adversary to provide an algorithm for re-sampling, i.e.,* $A(pk)$ *just outputs a message distribution* dist*. We say that* PKE *is* fully IND-SO-CPA *secure if*

$$\mathsf{Adv}^{\mathsf{s\text{-}ind\text{-}so}}_{\mathsf{PKE},A}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{full\text{-}ind\text{-}so}}_{\mathsf{PKE},A}(k) = 1\right] - \frac{1}{2}.$$

is negligible.

Definition 3 resembles the IND-SO-COM definition from [1], only for encryption instead of commitments, and with the same syntactic differences as above. (We note that [1] only consider efficiently re-samplable message spaces in their results about encryption schemes. In their results about selective opening secure commitments, the involved message spaces are arbitrary, as in Definition 3.)

Definition 4 (simulation-based selective opening security). *For a PKE scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*, a polynomially bounded function* $N = N(k) > 0$*, and a stateful PPT adversary* A*, consider the following experiments:*

Experiment $\mathsf{Exp}^{\mathsf{sim\text{-}so\text{-}real}}_{\mathsf{PKE},A}$	**Experiment** $\mathsf{Exp}^{\mathsf{sim\text{-}so\text{-}ideal}}_{S}$
$(pk, sk) \leftarrow \mathsf{Gen}(1^k)$	$\mathsf{dist} \leftarrow S()$
$\mathsf{dist} \leftarrow A(pk)$	$\mathbf{M} := (M^{(i)})_{i\in[n]} \leftarrow \mathsf{dist}$
$\mathbf{M} := (M^{(i)})_{i\in[n]} \leftarrow \mathsf{dist}$	$out_S \leftarrow S^{\mathcal{O}(\cdot)}(\texttt{select})$
$\mathbf{R} := (R^{(i)})_{i\in[n]} \leftarrow (\mathcal{R}_{\mathsf{Enc}})^N$	$\mathcal{I} := \mathcal{O}(\texttt{get queries})$
$\mathbf{C} := (C^{(i)})_{i\in[n]} := (\mathsf{Enc}(pk, M^{(i)}; R^{(i)}))_{i\in[n]}$	return $(\mathbf{M}, \mathsf{dist}, \mathcal{I}, out_S)$
$\mathbf{O} := (M^{(i)}, R^{(i)})_{i\in[n]}$	
$out_A \leftarrow A^{\mathcal{O}(\cdot)}(\texttt{select}, \mathbf{C})$	
$\mathcal{I} := \mathcal{O}(\texttt{get queries})$	
return $(\mathbf{M}, \mathsf{dist}, \mathcal{I}, out_A)$	

We say that the scheme is SIM-SO-CPA secure iff for every aversary A *there is a PPT algorithm, the simulator,* S *such that the distributions induced by the experiments* $\mathsf{Exp}^{\mathsf{sim\text{-}so\text{-}real}}_{\mathsf{PKE},A}$ *and* $\mathsf{Exp}^{\mathsf{sim\text{-}so\text{-}ideal}}_{S}$ *are computationally indistinguishable.*

Apart from the differences mentioned above, Definition 4 is identical to the SEM-SO-ENC definition from [1].

3 SIM-SO-CPA Security Does Not Imply Full IND-SO-CPA Security

We prove by counterexample that there are SIM-CO-CPA secure PKE schemes that are not fully IND-SO-CPA secure. Let $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme with message space $\{0,1\}^k$ that is SIM-SO-CPA secure[3]. From PKE we construct a scheme $\mathsf{PKE}' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ (see Figure 2) that is still SIM-SO-CPA secure, which is what we prove first, but not fully IND-SO-CPA secure.

For the construction of PKE' (see Figure 2) we use a cyclic DDH group $\mathbb{G}$ of prime order. We assume that the underlying SIM-SO-CPA secure scheme PKE can encrypt elements of $\mathbb{G}$ and $\mathbb{G}$-exponents.[4] The idea of our modification is to

[3] Such schemes exists under reasonable assumptions, see [1, 9] for example.

[4] Specifically, in the term $(g^s u^t)^M$ used in Enc', the message M can be a group element. We thus implicitly assume a suitable encoding of group elements as (nonzero) $\mathbb{G}$-exponents; depending on $\mathbb{G}$, this may additionally require application of a collision-resistant hash function H, so that the term becomes $(g^s u^t)^{H(M)}$. We stress that our results do not depend on the used encoding or hash function.

$\mathsf{Gen}'\ (1^k)$	$\mathsf{Enc}'\ (pk', M)$	$\mathsf{Dec}'\ (sk, C)$
$(pk, sk) \leftarrow \mathsf{Gen}(1^k)$	$((pk, g, h, u, v) := pk'$	$(C_1, C_2) := C$
$g \leftarrow \mathbb{G}, h \leftarrow \mathbb{G}$	$s \leftarrow [\lvert\mathbb{G}\rvert], t \leftarrow [\lvert\mathbb{G}\rvert]$	$M := \mathsf{Dec}(sk, C_1)$
$\omega \leftarrow [\lvert\mathbb{G}\rvert]$	$C_1 \leftarrow \mathsf{Enc}(pk, M)$	return M
$u := g^\omega, v := h^\omega$	$C_2 := ((g^s u^t)^M, h^s v^t)$	
return $((pk, g, h, u, v), sk)$	return (C_1, C_2)	

Fig. 2. PKE', a scheme which is SIM-SO-CPA but not fully IND-SO-CPA secure

extend the ciphertext by a "dual-mode" commitment (in the spirit of [6]). If the public key is generated honestly, the commitment is perfectly binding. However, in the course of the proof of Lemma 1, we will swap the public key. Thereby we switch to the alternative mode where the commitment is equivocable with the help of a trapdoor. Finally, in the proof of Lemma 2, we can use the commitment to show that PKE' is not fully IND-SO-CPA secure.

For a ciphertext $C \leftarrow \mathsf{Enc}'(pk, M)$ under PKE' we write $(M, (r, s, t))$ for the corresponding opening. (r, s, t) resembles the randomness used to generate c: r is the randomness used by Enc and s and t are the coins for the commitment (see Figure 2).

3.1 PKE' is SIM-SO-CPA Secure

Lemma 1. PKE' *is SIM-SO-CPA secure.*

Proof. Let A' be an adversary for PKE'. Our goal is to construct a simulator S such that $\mathsf{Exp}^{\mathsf{sim\text{-}so\text{-}real}}_{\mathsf{PKE}',A'}$ and $\mathsf{Exp}^{\mathsf{sim\text{-}so\text{-}ideal}}_{S}$ are computationally indistinguishable. Towards this goal we first construct an adversary A that uses A' to attack PKE. Then we show the indistinguishability of $\mathsf{Exp}^{\mathsf{sim\text{-}so\text{-}real}}_{\mathsf{PKE}',A'}$ and $\mathsf{Exp}^{\mathsf{sim\text{-}so\text{-}real}}_{\mathsf{PKE},A}$ and finally use the SIM-SO-CPA security of PKE to obtain S.

The SIM-SO-CPA-real experiment calls A twice, once to obtain the message distribution dist, and once to obtain the output of the adversary after the opening phase. Based on these calls we define A as follows:

Message distribution. A uniformly picks g, h from $\mathbb{G}$ and $\omega_u \neq \omega_v$ from $[\lvert\mathbb{G}\rvert]$. It then computes $u := g^{\omega_u}$, $v := h^{\omega_v}$ and returns $A'((pk, g, h, u, v))$.

Opening queries. A uniformly picks vectors $\mathbf{S}, \mathbf{T} \leftarrow [\lvert\mathbb{G}\rvert]^N$ of values and computes $C_1^{(i)} := C^{(i)}$, $C_2^{(i)} := (u^{\mathbf{S}^{(i)}}, v^{\mathbf{T}^{(i)}})$ and $\mathbf{C}' := (C_1^{(i)}, C_2^{(i)})_{i \in [\lvert\mathbf{C}\rvert]}$. Next A constructs an opening oracle $\mathcal{O}'$ that works as follows: If called with an index i, it fetches the corresponding opening $(M, R) := \mathcal{O}(i)$ from $\mathcal{O}$ and computes

$$s := \omega_u \omega_v (\mathbf{S}^{(i)} - T^{(i)} M) / (\omega_u M - \omega_v M)$$

and

$$t := \mathbf{T}^{(i)} - s/\omega_v$$

which yield the opening $(M, (R, s, t))$ for $C'^{(i)}$. Note that we have $(g^s u^t)^{M_i} = u^{\mathbf{S}^{(i)}}$ and $h^s v^t = v^{\mathbf{T}^{(i)}}$. A returns $A'^{\mathcal{O}'(\cdot)}(\mathtt{select}, \mathbf{C}')$.

We now provide a sequence of games that shows the computational indistinguishability of $\mathsf{Exp}^{\text{sim-so-real}}_{\mathsf{PKE}',A'}$ and $\mathsf{Exp}^{\text{sim-so-real}}_{\mathsf{PKE},A}$. **Game** 1 is simply the real SIM-SO-CPA experiment with A' and PKE'. In **Game** 2 the experiment runs with a modified public key: Let $pk' = (pk, g, h, u, v)$ denote the public key generated by Gen'. The experiment in Game 2 uniformly picks $\omega_u \neq \omega_v$ from $[\#\mathbb{G}]$ and sends the tuple $(pk, g, h, g^{\omega_u}, h^{\omega_v})$ instead of pk' to A'. Every efficient algorithm that could distinguish the distribution generated by Game 1 from that generated by Game 2 with non-negligible probability would win the DDH-experiment with non-negligible probability. In **Game** 3 we remove the information about the encrypted message from the commitment part of the ciphertext. For each ciphertext $C = (\mathsf{Enc}(pk, M), ((g^s u^t)^M, h^s v^t))$ in $\mathbf{C}$ the experiment picks s and t uniformly from $[|\mathbb{G}|]$ and replaces C_2 by (u^s, v^t). If A' wishes to open the ciphertext, the experiment computes an opening as described in the definition of A above using the knowledge of ω_u and ω_v. The distributions of Game 2 and Game 3 are identical: The commitment part of the ciphertext consists of $((g^s u^t)^M, h^s v^t)$ for uniform s and t. Since $\omega_u = \log_g(u) \neq \log_h(v) = \omega_v$, its distribution is identical to[5] (g^{aM}, g^b) for uniformly random a and b and hence obviously identical to (u^s, v^t) for random s, t. Similarly we can see that the random values in the openings are still distributed uniformly as well.

The situation in Game 3 is identical to running the SIM-SO-CPA-real experiment with A and PKE. Since A is SIM-SO-CPA secure there is a simulator S such that $\mathsf{Exp}^{\text{sim-so-real}}_{\mathsf{PKE},A} \overset{c}{\approx} \mathsf{Exp}^{\text{sim-so-ideal}}_{S}$. Altogether we find $\mathsf{Exp}^{\text{sim-so-real}}_{\mathsf{PKE}',A'} \overset{c}{\approx} \mathsf{Exp}^{\text{sim-so-real}}_{\mathsf{PKE},A} \overset{c}{\approx} \mathsf{Exp}^{\text{sim-so-ideal}}_{S}$. Hence S simulates A' which concludes our proof.

3.2 PKE' Is Not Fully IND-SO-CPA Secure

Lemma 2. PKE' *is not fully IND-SO-CPA secure.*

Proof. We construct an adversary A that wins the full IND-SO-CPA experiment with non-negligible probability. Basically, A benefits from the fact that the experiment conditions the distribution of messages dist on the choice of openings $\mathcal{I}$ to sample $\mathbf{M}_1$ even if this re-sampling could not be done efficiently by A. In the course of this proof we will see that A can therefore utilize the experiment to compute a discrete logarithm which helps A to learn the experiment's choice b.

We now describe the adversary A.

Message distribution. When A receives the public key $pk' = (pk, g, h, u, v)$ it responds with a distribution of tuples $(x, z) \in \mathbb{Z}_{|\mathbb{G}|} \times \mathbb{G}$ determined by the following algorithm:

Distribution dist
$b \leftarrow \{0, 1\}$
$x \leftarrow [|\mathbb{G}|]$
if $b = 0$ then return (x, g^x) otherwise return (x, h^x)

[5] Recall that we have assumed an encoding of M that does not map to 0.

Intuitively, this algorithm draws a random element z from $\mathbb{G}$ and returns either $(\log_g z, z)$ or $(\log_h z, z)$.

Challenge ciphertexts. A receives $\mathbf{C} \leftarrow (\mathsf{Enc}'(pk', x), \mathsf{Enc}'(pk', z))$ for some x and $z = g^x$ or $z = h^x$. Let $(\mathsf{Enc}(pk, x), ((g^s u^t)^x, h^s v^t)) = C^{(1)}$.

Opening queries. A calls $\mathcal{O}(2)$ to open the second component of $\mathbf{C}$. The return value of this call is of no interest for A here. However, it is important that the value of z is fixed for the re-sampling of $\mathbf{M}_1$.

Challenge messages. Finally, A receives a message vector $\mathbf{M}_b = (y, z)$ from the experiment. If

$$(h^s v^t)^y = (g^s u^t)^x \tag{2}$$

then A returns 1 and 0 otherwise.

Analysis. **Game** 1 is the full IND-SO-CPA experiment $\mathsf{Exp}^{\mathsf{full\text{-}ind\text{-}so}}_{\mathsf{PKE},A}$. In **Game** 2 the experiment calls $\mathsf{Gen}(1^k)$ to generate the public key $(pk, g, h, u, v) = pk' \leftarrow \mathsf{Gen}(1^k)$ until $g \neq h$ and $gh \neq 1$ before sending pk' to A. The statistical distance of the two distributions of public keys is $\frac{2}{|\mathbb{G}|}$ and hence negligible.

We now analyze the advantage of A in Game 2. By opening the second component of the ciphertext vector A fixes its value, i.e. $z := \mathbf{M}_0^{(2)} = \mathbf{M}_1^{(2)}$. However, since the value of z does not determine whether the first component of $\mathbf{M}_b$ contains the logarithm to base g or to base h, this is decided only when $\mathbf{M}_1$ is sampled. An adversary A benefits from this re-sampling if $\mathbf{M}_0 = (x = \log_g(z), z)$, $\mathbf{M}_1 = (y = \log_h(z), z)$ and $b = 1$. In this case A learns y and only then[6] we have that equation 2 holds.

We now show that the advantage of A is non-negligible. We define the three events

- B: The experiment samples $b = 1$.
- $M0$: The experiment samples $\mathbf{M}_0 = (x, g^x)$ (i.e. the first message vector contains a logarithm to base g).
- $M1$: The experiment samples $\mathbf{M}_1 = (y, h^y = z)$ for a fixed z (i.e. the second message vector contains a logarithm to base h).

Let $\overline{E}$ denote the complementary event for an event E. We observe that A outputs 1 if $B \wedge M0 \wedge M1$ and 0 if $\overline{B \wedge M0 \wedge M1}$. Hence

$$\begin{aligned}
\Pr\left[\mathsf{Exp}^{\mathsf{full\text{-}ind\text{-}so}}_{\mathsf{PKE},A} = 1\right] &= \Pr\left[\overline{B} \wedge \overline{B \wedge M0 \wedge M1}\right] + \Pr\left[B \wedge (B \wedge M0 \wedge M1)\right] \\
&\overset{(*)}{=} \Pr\left[\overline{B} \vee (\overline{B} \wedge (\overline{M0} \vee \overline{M1}))\right] + \Pr\left[B \wedge M0 \wedge M1\right] \\
&= \Pr\left[\overline{B}\right] + \Pr\left[B\right]\Pr\left[M0\right]\Pr\left[M1\right] \\
&= \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{5}{8},
\end{aligned}$$

where $(*)$ uses that B, $M0$ and $M1$ are independent events. Altogether, the adversary's advantage in Game 2 is

[6] Since $g \neq h$ and $gh \neq 1$.

$$\mathsf{Adv}^{\text{s-ind-so}}_{\mathsf{PKE},A} = \Pr\left[\mathsf{Exp}^{\text{full-ind-so}}_{\mathsf{PKE},A} = 1\right] - \frac{1}{2} = \frac{1}{8}$$

which is non-negligible.

4 Full IND-SO-CPA Does Not Imply SIM-SO-CPA

4.1 Outline

We will now construct a fully IND-SO-CPA secure PKE scheme that is *not* SIM-SO-CPA secure. To this end, we will start from a fully IND-SO-CPA secure scheme PKE.[7] We will then add a commitment (to the encrypted message) to each PKE ciphertext, such that the resulting scheme PKE' becomes committing. The result of Bellare et al. [2] then implies that PKE' is not SIM-SO-CPA secure.

The heart of our argument will thus be to show that PKE' is still fully IND-SO-CPA secure. We will reduce the IND-SO-CPA security of PKE' to that of PKE. Concretely, assume an IND-SO-CPA adversary A' on PKE'. We need to construct an IND-SO-CPA adversary A on PKE. Of course, A will internally run A' and try to map PKE ciphertexts and openings to those of PKE'.

The concrete problem for A is that initially, A' expects a vector of PKE' ciphertexts, which contain commitments to each message. Because these commitments do not appear in PKE ciphertexts, A will have to make up those commitments for A' *without knowing the respective messages*. Later on, however, when A' requests openings, A will have to also open those commitments to messages not known in advance (to A). In other words, A will have to equivocate commitments for A'.

This seems like an insurmountable problem: we need PKE' to be committing, in order to derive (using [2]) that PKE' is not SIM-SO-CPA secure. However, if PKE' is committing, then how could A possibly equivocate commitments? Our solution is to abuse the (possibly inefficient) re-sampling that occurs during the IND-SO-CPA experiment. Namely, observe that statistically hiding commitments can always be equivocated *inefficiently* (at least with high probability). In fact, equivocating a commitment $com = \mathsf{Com}(M; R)$ (with message M and randomness R) can be formulated as re-sampling from the message distribution $(M, R, \mathsf{Com}(M; R))$, conditioned on a fixed value com for the third component. This will essentially allow our adversary A to formulate the necessary equivocations as a re-sampling of suitable message distribution.

4.2 Non-interactive Statistically Hiding Commitments

As a technical tool for our separation, we will require the notion of suitable commitments. To allow for (inefficient) equivocation, we will require that the commitments are statistically hiding. Additionally, for the use in a PKE scheme,

[7] To date, there is no PKE scheme that is known to be fully IND-SO-CPA secure. However, in case no IND-SO-CPA secure scheme exists, of course no separating example can be constructed.

the commitments should be non-interactive. Finally, we stress that we do not allow any public parameters (such as a common reference string).

Definition 5 (NISHCOMs). *A* non-interactive statistically hiding commitment scheme (NISHCOM) *is a PPT algorithm* Com *that takes as input a message* $M \in \{0,1\}$ *and outputs a commitment* $com \in \{0,1\}^*$*. We require the following properties:*

Statistical hiding. *The statistical distance* $\mathsf{SD}(\mathsf{Com}(0); \mathsf{Com}(1))$ *is negligible in* k*.*

Binding. *For every PPT* A*, the following probability is negligible (in* k*):*

$$\Pr\left[\mathsf{Com}(0; R_0) = \mathsf{Com}(1; R_1) \mid (R_0, R_1) \leftarrow A(1^k)\right].$$

While one-way functions imply statistically hiding commitments [12], we cannot expect to construct NISHCOMs even from trapdoor one-way permutations [11]. In fact, there can be no NISHCOM that is binding against *non-uniform* adversaries. (The statistical hiding property implies that for each k, there exist many tuples (R_0, R_1) with $\mathsf{Com}(0; R_0) = \mathsf{Com}(1; R_1)$. We can always hardcode one such tuple into a non-uniform A.) However, under specific assumptions, we *can* construct NISHCOMs:

NISHCOMs from CRHFs. Assume a collision-resistant hash function $H : \{0,1\}^* \to \{0,1\}^k$. We stress that H is not keyed but fixed. (In particular, we can only hope for collision-resistance against uniform adversaries.) Instantiated with such an H, Naor and Yung [15], and Damgård et al. [7] yield several constructions of NISHCOMs. For instance, implicit in [15] is the NISHCOM

$$\mathsf{Com}(M; (X, h)) := (H(X), h, h(X) \oplus M)$$

for $M \in \{0,1\}$, $X \in \{0,1\}^\ell$ for suitably large ℓ, and a suitable randomness extractor h.

NISHCOMs from Fixed Groups. Let $(\mathbb{G}_k, g_k, h_k)_{k\in\mathbb{N}}$ be a family of finite groups, one for each value of the security parameter k, along with (fixed) generators g_k, h_k of $\mathbb{G}_k$. If we assume that the problem of computing $\mathrm{dlog}_{g_k}(h_k)$ is computationally infeasible, then Pedersen's commitment [16] (i.e., $\mathsf{Com}(M; R) := g_k^M h_k^R$) is a NISHCOM that is even perfectly hiding.

4.3 The Separating Scheme

We are now ready to describe our scheme. We assume a fully IND-SO-CPA secure scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\{0,1\}$, as well as a NISHCOM Com. In our scheme, depicted in Figure 3, we simply append to each ciphertext a commitment to the encrypted message. This commitment is never checked or opened during execution of the scheme; it only serves as a means to make the scheme committing in the sense of Bellare et al. [2].

$\mathsf{Gen}'(1^k)$	$\mathsf{Enc}'(pk', M)$	$\mathsf{Dec}'(sk', C')$
$(pk, sk) \leftarrow \mathsf{Gen}(1^k)$	$C \leftarrow \mathsf{Enc}(pk, M)$	$(C, com) := C'$
return (pk, sk)	$com \leftarrow \mathsf{Com}(M)$	$M \leftarrow \mathsf{Dec}(sk', C)$
	return $C' := (C, com)$	return M

Fig. 3. PKE' — a fully IND-SO-CPA, but not SIM-SO-CPA secure PKE scheme

4.4 SIM-SO-CPA Insecurity of the Scheme

First, we note that because of our use of Com, scheme PKE' is a binding CE ("committing encryption") scheme in the sense of Bellare et al. [2]. Concretely, opening a ciphertext (by releasing the encryption randomness) as an honest encryption in two different ways (i.e., for two different messages) requires breaking the binding property of Com. Hence, we can apply [2, Theorem 4.1][8], and we get:

Theorem 1. PKE' *as depicted in Figure 3 is not SIM-SO-CPA secure.*

4.5 Full IND-SO-CPA Security of the Scheme

The main part of our work is to prove that PKE' is fully IND-SO-CPA secure. As explained above, our intuition will be to use the (potentially inefficient) message re-sampling in the full IND-SO-CPA experiment to equivocate Com commitments.

Theorem 2. PKE' *as depicted in Figure 3 is fully IND-SO-CPA secure, provided that* PKE *is fully IND-SO-CPA secure, and* Com *is a NISHCOM.*

Proof. Given an IND-SO-CPA adversary A' on PKE', we construct an IND-SO-CPA adversary on PKE with roughly the same complexity and success. Concretely, A proceeds as follows:

Message distribution. When invoked with a PKE public key pk, A sets $pk' := pk$ and runs $\mathsf{dist}' \leftarrow A'(pk')$ to obtain an N'-message distribution dist'. Then A creates and outputs its own N-message distribution (for $N := 3N'$) dist as follows:

Distribution dist
$(M'_i)_{i\in[N']} \leftarrow \mathsf{dist}'$
$(R_i^{\mathsf{Com}})_{i\in[N']} \leftarrow (\mathcal{R}_{\mathsf{Com}})^{N'}$
$(com_i)_{i\in[N']} := (\mathsf{Com}(M'_i; R_i^{\mathsf{Com}}))_{i\in[N']}$
return $(M'_1, R_1^{\mathsf{Com}}, com_1, \ldots, M'_{N'}, R_{N'}^{\mathsf{Com}}, com_{N'})$

[8] Note that there is an important difference between our SIM-SO-CPA definition and the one from [2]: In [2] the simulator and the adversary are allowed a common *auxiliary input* which is of importance for Theorem 4.1. However, it is easy to verify that all of our proofs concerning SIM-SO-CPA security are still valid in presence of an auxiliary input, which we omitted for the sake of simplicity.

Challenge ciphertexts. When receiving an N-ciphertext vector $(C_i)_{i\in[N]}$, A prepares an N'-ciphertext vector $(C'_i)_{i\in[N']}$ for A' as follows. First, A asks its own IND-SO-CPA experiment for openings of $C_3, C_6, \ldots, C_N$ to obtain the commitments com_i (for $i \in [N']$). It then sets $C'_i := (C_i, com_i)$ for all i and hands $(C'_i)_{i\in[N']}$ to D. Note that this results in a challenge ciphertext for D that is perfectly distributed as in D's own IND-SO-CPA experiment. Furthermore, because Com is statistically hiding, opening the encrypted commitments does not fix any of the encrypted messages.

Opening queries. When A' wants a ciphertext C'_i opened, A asks for an opening of C_{3i-2} and C_{3i-1}. The opening of C_{3i-2} yields a properly distributed opening of the PKE part C_i of $C'_i = (C_i, com_i)$. On the other hand, the opening of C_{3i-1} reveals the randomness R_i^{Com} of the corresponding commitment com_i. Together, this forms a perfectly distributed opening of C'_i, which A then hands to A'.

Challenge messages. Finally, when A' is finished asking for openings and requests challenge messages, A does the same and hands the corresponding M'_i (for $i \in [N']$) to A'. When A' outputs a decision bit b', then A outputs the same bit.

To analyze this A, first note that up to the challenge message, A provides a perfect internal simulation of A' running in its own IND-SO-COM experiment with PKE'. In particular, both challenge ciphertexts and openings are exactly distributed as with PKE'. For the eventual challenge message (and A''s decision bit), we make the following case distinction:

When A's experiment tosses $b = 0$ (i.e., no re-sampling). In this case, A eventually obtains the initially sampled plaintext vector with all M'_i, R_i^{Com}, com_i. In particular, A' gets the messages M'_i just as it would have in its own IND-SO-CPA experiment with PKE'. We get:

$$\Pr\left[\mathsf{Exp}^{\mathsf{full\text{-}ind\text{-}so}}_{\mathsf{PKE},A}(k) = 1 \mid b = 0\right] = \Pr\left[\mathsf{Exp}^{\mathsf{full\text{-}ind\text{-}so}}_{\mathsf{PKE}',A'}(k) = 1 \mid b = 0\right]. \quad (3)$$

When A's experiment tosses $b = 1$ (i.e., re-sampling occurs). In this case, A eventually obtains a plaintext vector that has been re-sampled from dist, conditioned on all opened messages M'_i (along with the corresponding R_i^{Com}), *and all commitments* com_i. In particular, A' gets a re-sampled message vector that is additionally conditioned on *all* com_i. This marks a difference to what A' would have gotten in its IND-SO-CPA experiment with PKE': there, A' would have gotten M'_i that are only conditioned on the so far opened messages, but not on all com_i. However, recall that Com is statistically hiding, and thus the distribution of the com_i is statistically close to, say, commitments to all-zero strings. Thus, we will now prove that

$$\Pr\left[\mathsf{Exp}^{\mathsf{full\text{-}ind\text{-}so}}_{\mathsf{PKE},A}(k) = 1 \mid b = 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{full\text{-}ind\text{-}so}}_{\mathsf{PKE}',A'}(k) = 1 \mid b = 1\right]. \quad (4)$$

is negligible in k, using a sequence of Games.

Game 1 is simply the IND-SO-CPA experiment with A and PKE as described above, but with b fixed to 1.

In **Game** 2, we substitute all $com_i \leftarrow \mathsf{Com}(M'_i)$ by $com_i \leftarrow \mathsf{Com}(0)$. We stress that during the resampling operation, we still condition on the com_i being output as M_i-commitments. Note that this conditioning operation may fail, e.g., when some M_i has been opened as $M_i = 1$, but com_i lies not in the range of $\mathsf{Com}(1)$. However, this can happen only with negligible probability by the hiding property of Com. Namely, note that for each sampled message vector $(M'_i)_{i\in[N']}$, we can view the whole experiment (including A''s output) as a probabilistic function of the commitments com_i. If any commitment randomness R_i^{Com} is to be revealed, this randomness can be — inefficiently — generated from com_i and the corresponding M_i. Since Com is statistically hiding, we know that hence, A''s output does not significantly change compared to Game 1.

In **Game** 3, we no longer condition on the com_i during re-sampling. (Of course, we still condition on the so far opened M'_i.) Lemma 3 in Appendix A shows that this has no significant effect on the experiment's output. Concretely, note that we can view both Game 2 and Game 3 (including A) as an unbounded algorithm that

- gets a vector $(com_i)_{i\in[n]}$ of 0-commitments as input,
- then deterministically[9] selects a message distribution $\widetilde{\mathsf{dist}}$ over $\{0,1\}^n$ (that internally corresponds to dist', conditioned on all opened messages),
- and finally gets a sample from either $\widetilde{\mathsf{dist}}$, or $\widetilde{\mathsf{dist}}$ conditioned on all commitments com_i. With a $\widetilde{\mathsf{dist}}$-sample, this results in Game 3, whereas with a sample from $\widetilde{\mathsf{dist}} \mid (com_i)_i$, this yields an execution of Game 2.

Applying Lemma 3 yields that the output in Game 3 does not significantly differ from that in Game 2. (Somewhat surprisingly, the same statement would not hold if the M_i were not bits but, say, k-bitstrings. See the full version [3] for details.) At first glance, it might seem like we only need a non-adaptive version of Lemma 3, in which the adversary chooses the distribution ahead of time. However, such a non-adaptive Lemma would not be sufficient in our case, because the distribution $\widetilde{\mathsf{dist}}$ depends on the adversary's opening requests and thus may depend on the commitments com_i.

Finally, in **Game** 4, we replace all $com_i \leftarrow \mathsf{Com}(0)$ again by $com_i \leftarrow \mathsf{Com}(M'_i)$. Like in Game 2, this has no significant effect on the output of the experiment.

Now note that in Game 4, re-sampled message vectors (M'_i) are no longer conditioned on the com_i, and are hence distributed exactly as in $\mathsf{Exp}^{\text{full-ind-so}}_{\mathsf{PKE}',A'}$ with $b = 1$. Also, commitments and openings are distributed exactly as with PKE'. We obtain (4).

Taking (3,4) together, we get that

$$\mathsf{Adv}^{\text{s-ind-so}}_{\mathsf{PKE},A}(k) - \mathsf{Adv}^{\text{s-ind-so}}_{\mathsf{PKE}',A'}(k)$$

is negligible, which proves the theorem.

[9] At this point, we can assume without loss of generality that the experiment, including A', is unbounded, and can thus choose its own random coins deterministically.

References

[1] Bellare, M., Hofheinz, D., Yilek, S.: Possibility and Impossibility Results for Encryption and Commitment Secure under Selective Opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (2009)

[2] Bellare, M., Dowsley, R., Waters, B., Yilek, S.: Standard Security Does Not Imply Security against Selective-Opening. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 645–662. Springer, Heidelberg (2012)

[3] Böhl, F., Hofheinz, D., Kraschewski, D.: On definitions of selective opening security. IACR Cryptology ePrint Archive 678 (2011)

[4] Canetti, R., Feige, U., Goldreich, O., Naor, M.: Adaptively secure multi-party computation. In: 28th Annual ACM Symposium on Theory of Computing, pp. 639–648. ACM Press (May 1996)

[5] Canetti, R., Dwork, C., Naor, M., Ostrovsky, R.: Deniable Encryption. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 90–104. Springer, Heidelberg (1997)

[6] Damgård, I., Nielsen, J.B.: Perfect Hiding and Perfect Binding Universally Composable Commitment Schemes with Constant Expansion Factor. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 581–596. Springer, Heidelberg (2002)

[7] Damgård, I., Pedersen, T.P., Pfitzmann, B.: On the existence of statistically hiding bit commitment schemes and fail-stop signatures. Journal of Cryptology 10(3), 163–194 (1997)

[8] Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.J.: Magic functions. In: 40th Annual Symposium on Foundations of Computer Science, pp. 523–534. IEEE Computer Society Press (October 1999)

[9] Fehr, S., Hofheinz, D., Kiltz, E., Wee, H.: Encryption Schemes Secure against Chosen-Ciphertext Selective Opening Attacks. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 381–402. Springer, Heidelberg (2010)

[10] Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)

[11] Haitner, I., Hoch, J.J., Reingold, O., Segev, G.: Finding collisions in interactive protocols - a tight lower bound on the round complexity of statistically-hiding commitments. In: 48th Annual Symposium on Foundations of Computer Science, pp. 669–679. IEEE Computer Society Press (October 2007)

[12] Haitner, I., Nguyen, M.-H., Ong, S.J., Reingold, O., Vadhan, S.P.: Statistically hiding commitments and statistical zero-knowledge arguments from any one-way function. SIAM J. Comput. 39(3), 1153–1218 (2009)

[13] Hemenway, B., Libert, B., Ostrovsky, R., Vergnaud, D.: Lossy Encryption: Constructions from General Assumptions and Efficient Selective Opening Chosen Ciphertext Security. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 70–88. Springer, Heidelberg (2011)

[14] Hofheinz, D.: All-But-Many Lossy Trapdoor Functions. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 209–227. Springer, Heidelberg (2012)

[15] Naor, M., Yung, M.: Universal one-way hash functions and their cryptographic applications. In: 21st Annual ACM Symposium on Theory of Computing, pp. 33–43. ACM Press (May 1989)

[16] Pedersen, T.P.: Non-interactive and Information-Theoretic Secure Verifiable Secret Sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992)

[17] Peikert, C., Vaikuntanathan, V., Waters, B.: A Framework for Efficient and Composable Oblivious Transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008)

A A Technical Lemma

For a concise presentation, in the following lemma we represent
- the distributions of 0-commitments and 1-commitments by two probability mass functions γ_0, γ_1,
- the initially given commitment vector $(com_i)_{i\in[n]}$ by a random variable $\mathbf{C}$,
- by a family of probability mass functions $\beta_{\mathbf{c}}$ we represent how the message distribution $\widetilde{\mathsf{dist}}$ is generated from the initially given commitment vector,
- and the two possible sample distributions by two random variables $\mathbf{M}, \mathbf{M}'$.

Lemma 3. *Fix the following parameters:*
- *message space* $\{0,1\}$ *and some countable commitment space* $\mathcal{C}$
- *a tuple* $(\gamma_m)_{m\in\{0,1\}}$*, consisting of two probability mass functions over* $\mathcal{C}$
- *some* $n \in \mathbb{N}$ *and a family* $(\beta_{\mathbf{c}})_{\mathbf{c}\in\mathcal{C}^n}$ *of probability mass functions over* $\{0,1\}^n$

In this setting let the random variables $\mathbf{C} = (C_i)_{i\in[n]} \in \mathcal{C}^n$ *and* $\mathbf{M} = (M_i)_{i\in[n]} \in \{0,1\}^n$ *and* $\mathbf{M}' = (M'_i)_{i\in[n]} \in \{0,1\}^n$ *be given, distributed as follows:*

$$\begin{aligned}
\Pr[\mathbf{C} = \mathbf{c}] &= \prod\nolimits_{i\in[n]} \gamma_0(c_i) \\
\Pr[\mathbf{M} = \mathbf{m} \mid \mathbf{C} = \mathbf{c}] &= \beta_{\mathbf{c}}(\mathbf{m}) \\
\Pr[\mathbf{M}' = \mathbf{m}' \mid \mathbf{C} = \mathbf{c}] &= \frac{\beta_{\mathbf{c}}(\mathbf{m}') \cdot \prod_{i\in[n]} \gamma_{m'_i}(c_i)}{\sum_{\mathbf{m}\in\{0,1\}^n} \beta_{\mathbf{c}}(\mathbf{m}) \cdot \prod_{i\in[n]} \gamma_{m_i}(c_i)}
\end{aligned}$$

Let $\mu := \mathsf{SD}(\gamma_0\,;\,\gamma_1)$ *in slight abuse of notation. Now, if* $(1+\sqrt{\mu})^n < 2$*, it holds:*

$$\mathsf{SD}((\mathbf{C},\mathbf{M})\,;\,(\mathbf{C},\mathbf{M}')) \;\le\; 2n\big(\sqrt{\mu}+\mu\big) + \frac{1}{2-(1+\sqrt{\mu})^n} - 1$$

In particular, if μ *is negligible and* n *is polynomially bounded, then the statistical distance* $\mathsf{SD}((\mathbf{C},\mathbf{M})\,;\,(\mathbf{C},\mathbf{M}'))$ *is also negligible.*

In the lemma we implicitly assume that the distribution $\widetilde{\mathsf{dist}}$ conditioned on the initially given commitments $(com_i)_{i\in[n]}$ is well defined in the sense that it assigns a non-zero probability to some message vector for which $(com_i)_{i\in[n]}$ is a possible commitment vector. This corresponds to the assumption that in Theorem 2, opening a 0-commitment as a commitment to M'_i does not fail. In particular, this assumption may be violated with at most negligible probability by the statistical hiding property of the commitment.

For the proof of Lemma 3 and a discussion why it only holds for small message space we refer to the full version [3].

New Definitions and Separations for Circular Security

David Cash[1,⋆], Matthew Green[2,⋆⋆], and Susan Hohenberger[2,⋆⋆⋆]

[1] IBM T.J. Watson Research Center
[2] Johns Hopkins University

Abstract. Traditional definitions of encryption security guarantee secrecy for any plaintext that can be computed by an outside adversary. In some settings, such as anonymous credential or disk encryption systems, this is not enough, because these applications encrypt messages that depend on the secret key. A natural question to ask is do standard definitions capture these scenarios? One area of interest is *n-circular security* where the ciphertexts $E(pk_1, sk_2), E(pk_2, sk_3), \ldots, E(pk_{n-1}, sk_n), E(pk_n, sk_1)$ must be indistinguishable from encryptions of zero. Acar et al. (Eurocrypt 2010) provided a CPA-secure public key cryptosystem that is not 2-circular secure due to a distinguishing attack.

In this work, we consider a natural relaxation of this definition. Informally, a cryptosystem is *n-weak circular secure* if an adversary given the cycle $E(pk_1, sk_2), E(pk_2, sk_3), \ldots, E(pk_{n-1}, sk_n), E(pk_n, sk_1)$ has no significant advantage in the regular security game, (e.g., CPA or CCA) where ciphertexts of chosen messages must be distinguished from ciphertexts of zero. Since this definition is sufficient for some practical applications and the Acar et al. counterexample no longer applies, the hope is that it would be easier to realize, or perhaps even implied by standard definitions. We show that this is unfortunately not the case: even this weaker notion is not implied by standard definitions. Specifically, we show:

- For symmetric encryption, under the minimal assumption that one-way functions exist, n-weak circular (CPA) security is not implied by CCA security, for any n. In fact, it is not even implied by authenticated encryption security, where ciphertext integrity is guaranteed.

⋆ This work was performed at the University of California, San Diego, supported in part by NSF grant CCF-0915675.

⋆⋆ Supported in part by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under contract FA8750-11-2-0211, the Office of Naval Research under contract N00014-11-1-0470, NSF grant CNS-1010928 and HHS 90TR0003/01. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the HHS.

⋆⋆⋆ Supported in part by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under contract FA8750-11-2-0211, the Office of Naval Research under contract N00014-11-1-0470, NSF CNS 1154035, a Microsoft Faculty Fellowship and a Google Faculty Research Award. Applying to all authors, the views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 540–557, 2012.

– For public-key encryption, under a number-theoretic assumption, 2-weak circular security is not implied by CCA security.

In both of these results, which also apply to the stronger circular security definition, *we actually show for the first time an attack in which the adversary can recover the secret key of an otherwise-secure encryption scheme after an encrypted key cycle is published.* These negative results are an important step in answering deep questions about which attacks are prevented by commonly-used definitions and systems of encryption. They say to practitioners: if key cycles may arise in your system, then even if you use CCA-secure encryption, your system may break catastrophically; that is, a passive adversary might be able to recover your secret keys.

Keywords: Encryption, Definitions, Circular Security, Counterexamples.

1 Introduction

Encryption is one of the most fundamental cryptographic primitives. Most definitions of encryption security [22,19,35] follow the seminal notion of Goldwasser and Micali which guarantees indistinguishability of encryptions for messages chosen by the adversary [22]. However, Goldwasser and Micali wisely warned to be careful when using a system proven secure within this framework on messages that the adversary cannot derive himself.

Over the past several years, there has been significant interest in designing schemes secure against *key-dependent message attacks*, e.g., [15,11,31,3,27,29,13,14,5,2], where the system must remain secure even when the adversary is allowed to obtain encryptions of messages that depend on the secret keys themselves. In this work, we are particularly interested in circular security [15]. A public-key cryptosystem is *n-circular secure* if the ciphertexts $E(pk_1, sk_2), E(pk_2, sk_3), \ldots, E(pk_{n-1}, sk_n), E(pk_n, sk_1)$, as well as ciphertexts of chosen messages, cannot be distinguished from encryptions of zero, for independent key pairs. Either by design or accident, these key cycles naturally arise in many applications, including storage systems such as BitLocker [13], anonymous credentials [15], the study of "axiomatic security" [31,3] and more. See [13] for a discussion of the applications.

Until recently, few positive or negative results regarding circular security were known outside of the random oracle model. On one hand, no n-circular secure cryptosystems were known for $n > 1$. On the other hand, no counterexamples existed for $n > 1$ to separate the definitions of circular and CPA security; that is, as far as anyone knew the CPA-security definition already captured circular security for any cycle larger than a self-loop.

Recently, this gap has been closing in two ways. On the positive side, several circular-secure schemes have been proposed [13,5,14]. The focus of the current work is on negative results – namely, investigating whether standard notions of encryption are "safe" for circular applications.

In 2008, Boneh, Halevi, Hamburg and Ostrovsky proved, by counterexample, that *one-way* security does not imply circular security [13]. Recently, Acar, Beleniky, Bellare and Cash [2] proved that, under an assumption in bilinear groups, CPA-security does not imply circular security.

Our Results. We narrow this gap even further by studying the extent to which standard definitions (e.g., CPA, CCA) imply a *weak* form of circular security. Our results are primarily negative.

1. Relaxing the Circular Security Notion. Perhaps the current formulation of circular security is "too strong"; that is, perhaps there is a relaxed notion of this definition which simultaneously satisfies many practical applications and yet is also *already* captured by standard security notions. This is an area worth investigating. We begin by proposing a natural relaxation called *weak circular security* where the adversary is handed an encrypted cycle $E(pk_1, sk_2), E(pk_2, sk_3), \ldots, E(pk_{n-1}, sk_n), E(pk_n, sk_1)$ along with the public keys and then proceeds to play the CPA or CCA security game as normal (where these ciphertexts are also off-limits for the decryption oracle). We stress here that the encrypted cycle is *always* generated as described, and is never changed to encryptions of zero. This definition is intriguing, and perhaps of independent interest, for two reasons.

First, the Acar et al. [2] counterexample does *not* apply to it. That construction uses the bilinear map to test whether a sequence of ciphertexts contain a cycle or zeros. Here the adversary knows he's getting an encrypted cycle, but then must extract some knowledge from this that helps him distinguish two messages of his choosing.

Second, this definition appears sufficient for some practical settings. Using a weak circular secure encryption scheme, Alice and Bob could exchange keys with each other over an insecure channel knowing that: (1) Eve can detect that they did so, but (2) Eve cannot learn anything about their other messages. Similarly, an adversary scanning over a user's BitLocker storage may detect that her drive contains an encrypted cycle, but cannot read anything on her drive. In an anonymous credential system of Camenisch and Lysyanskaya [15], a user has multiple keys. To participate in the system, the user must encrypt them in a cycle, provide this cycle to the other users, and prove that she has done this correctly. Then, if she shares one key, she automatically shares all her keys. In their application, *detection* of a cycle is actually desirable, provided that subsequent encryptions remain secure.

2. Symmetric-Key Counterexamples. In the symmetric setting, we show that standard notions do not imply n-circular security for any positive n. Specifically, given any $n \geq 1$, we show how to construct a secure authenticated encryption scheme (which is necessarily CCA-secure; see Section 2) that is not n-weak circular secure, under the minimal assumption that secure authenticated encryption schemes exist, which are equivalent to one-way functions.

The main technical ingredient in our counterexample is a lemma showing that it is provably hard for an adversary to compute an encrypted key cycle itself,

assuming that the symmetric scheme under attack is a secure authenticated encryption scheme (or CCA secure). We stress that this lemma does not hold if the encryption scheme is only CPA secure.

Our lemma gives us leverage in constructing a counterexample because it means the adversary is given strictly more power in the weak circular security game than in the standard security game. Specifically, the adversary is given an encrypted key cycle in the weak circular security game that it could not have computed itself, and we design a scheme to help such an adversary without affecting regular security.

3. Public-Key Counterexamples. We show that neither CPA nor CCA-security imply (even) weak circular security for cycles of size 2. That is, we show secure systems that are totally compromised when the independently-generated ciphertexts $E(pk_A, sk_B)$ and $E(pk_B, sk_A)$ are released. This is a difficult task, because the system must remain secure if either one, but only one, of these ciphertexts are released. Moreover, this counterexample requires new ideas. We cannot use the common trick in self-loop counterexamples that test if the message is the secret key corresponding to the public key, since there is no way for the encryption algorithm with public key pk_A to distinguish, say, sk_B from any other valid message. Specifically, we show that:

If there exists an algebraic setting where the Symmetric External Diffie-Hellman[1] (SXDH) assumption holds, then there exists a CPA-secure cryptosystem which is *not* 2-weak circular secure. The proposed scheme is particularly interesting in that it breaks *catastrophically* in the presence of a 2-cycle — revealing the secret keys of both users.

Moreover, if simulation-sound non-interactive zero- knowledge (NIZK) proof systems exist for NP and there exists an algebraic setting where the Symmetric External Diffie-Hellman (SXDH) assumption holds, then there exists a CCA-secure cryptosystem which is *not* 2-weak circular secure. This is also the first separation of CCA security and (regular) circular security.

These results deepen our understanding of how to define "secure" encryption and which practical attacks are captured by the standard definitions. They also provide additional justification for the ongoing effort, e.g. [13,14,5], to develop cryptosystems which are provably circular secure.

1.1 Related Work

In 2001, Camenisch and Lysyanskaya [15] introduced the notion of *circular security* and used it in their anonymous credential system to discourage users from delegating their secret keys. They also showed how to construct a circular-secure cryptosystem from any CPA-secure cryptosystem in the random oracle model.

[1] The SXDH assumption states that there is a bilinear setting $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ where the DDH assumption holds in both $\mathbb{G}_1$ and $\mathbb{G}_2$. It has been extensively studied and used e.g., [21,38,32,12,8,6,24,9,25], perhaps most notably as a setting of the Groth-Sahai NIZK proof system [25].

Independently, Abadi and Rogaway [1] and Black, Rogaway, Shrimpton [11] introduced the more general notion of *key-dependent message* (KDM) security, where the encrypted messages might depend on an arbitrary function of the secret keys. Black et al. showed how to realize this notion in the random oracle model.

Halevi and Krawczyk [27] extended the work of Black et al. to look at KDM security for deterministic secret-key functions such as pseudorandom functions (PRFs), tweakable blockciphers, and more. They give both positive and negative results, including some KDM-secure constructions in the standard model for PRFs. In the symmetric setting, Hofheinz and Unruh [29] showed how to construct circular-secure cryptosystems in the standard model under relaxed notions of security. Backes, Pfitzmann and Scedrov [7] presented stronger notions of KDM security (some in the random oracle model) and discussed the relationships among these notions.

In the public-key setting, Boneh, Halevi, Hamburg and Ostrovsky [13] presented the first cryptosystem which is simultaneously CPA-secure and n-circular-secure (for any n) in the standard model, based on either the DDH or Decision Linear assumptions. As mentioned earlier, Boneh et al. [13] also proved, by counterexample, that *one-way* security does not imply circular security. One-way encryption is a very weak notion, which informally states that given $(pk, E(pk, m))$, the adversary should not be able to recover m. Given any one-way encryption system, they constructed a one-way encryption system that is not n-circular secure (for any n). Their system generates two key pairs from the original and sets $PK = pk_1$ and $SK = (sk_1, sk_2)$. A message (m_1, m_2) is encrypted as $(m_1, E(pk_1, m_2))$. In the event of a 2-cycle, the values $\mathsf{Enc}(pk_A, sk_B) = (sk_{B,1}, E(pk_{A,1}, sk_{B,2}))$ and $\mathsf{Enc}(pk_B, sk_A) = (sk_{A,1}, E(pk_{B,1}, sk_{A,2}))$ provide the critical secret key information $(sk_{B,1}, sk_{A,1})$ in the clear.

Subsequently, Applebaum, Cash, Peikert and Sahai [5] adapted the circular-secure construction of [13] into the lattice setting. Camenisch, Chandran and Shoup [14] extended [13] to the first cryptosystem which is simultaneously CCA-secure and n-circular-secure (for any n) in the standard model, by applying the "double encryption" paradigm of Naor and Yung [34]. (Interestingly, we use this same approach in Section 4.4 to extend our public-key counterexample from CPA to CCA security.)

Haitner and Holenstein [26] recently provided strong impossibility results for KDM-security *with respect to 1-key cycles* (a.k.a., self-loops.) They study the problem of building an encryption scheme where it is secure to release $E(k, g(k))$ for various functions g. First, they show that there exists no fully-black-box reduction from a KDM-secure encryption scheme to one-way permutations (or even some families of trapdoor permutations) if the adversary can obtain encryptions of $g(k)$, where g is a poly(n)-wise independent hash function. Second, there exists no reduction from an encryption scheme secure against key-dependent messages to, essentially, any cryptographic assumption, if the adversary can obtain an encryption of $g(k)$ for an *arbitrary* g, as long as the security reduction treats both the adversary and the function g as black boxes. These results address

the possibility of achieving strong single-user KDM-security via reductions to cryptographic assumptions. The results in this paper study a version of KDM security that is in one sense weaker – we only allow a narrow class of functions g – but also stronger because it considers multiple users. Our results also address a different question regarding KDM security. We study whether or not KDM security is always implied by regular security while Haitner and Holenstein study the possibility of achieving strong single-user KDM security via specialized constructions.

Recently, Acar et al. [2] demonstrated both public and private key encryption systems that are provably CPA-secure and yet also demonstrably *not* 2-circular secure. Their counterexample does not apply to CCA or weak circular security.

There is also a relationship to recent work on *leakage resilient* and *auxiliary input* models of encryption, which mostly falls into the "self-loop" category. In leakage resilient models, such as those of Akavia, Goldwasser and Vaikuntanathan [4] and Naor and Segev [33], the adversary is given some function h of the secret key, not necessarily an encryption, such that it is *information theoretically* impossible to recover sk. The auxiliary input model, introduced by Dodis, Kalai and Lovett [18], relaxes this requirement so that it only needs to be difficult to recover sk.

Self-Loops. In sharp contrast to all $n \geq 2$, the case of 1-circular security is fairly well understood. A folklore counterexample shows that CPA-security does not directly imply 1-circular security. Given any encryption scheme (G, E, D), one can build a second scheme (G, E', D') as follows: (1) $E'(pk, m)$ outputs $E(pk, m)||0$ if $m \neq sk$ and $m||1$ otherwise, (2) $D'(sk, c||b)$ outputs $D(sk, m)$ if $b = 0$ and sk otherwise. It is easy to show that if (G, E, D) is CPA-secure, then (G, E', D') is CPA-secure. When $E'(pk, sk) = sk||1$ is exposed, then there is a complete break. Conversely, given any CPA-secure system, one can build a 1-circular secure scheme in the standard model [13].

2 Definitions of Security

A *public-key encryption system* Π is a tuple of algorithms $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, where KeyGen is a key-generation algorithm that takes as input a security parameter λ and outputs a public/secret key pair (pk, sk); $\mathsf{Enc}(pk, m)$ encrypts a message m under public key pk; and $\mathsf{Dec}(sk, c)$ decrypts ciphertext c with secret key sk. A *symmetric-key encryption system* is a public-key encryption system, except that it always outputs $pk = \perp$, and the encryption algorithm computes ciphertexts using sk, i.e. by running $\mathsf{Enc}(sk, m)$. In the symmetric case we will sometimes write K instead of sk. As in most other works, we assume that all algorithms implicitly have access to shared public parameters establishing a common algebraic setting.

Our definitions of security will associate a message space, denoted M, with each encryption scheme. Throughout this paper, we assume that the space of possible secret keys output by KeyGen is a subset of the message space M and

$\underline{\mathsf{IND\text{-}CPA}(\Pi, \mathcal{A}, \lambda)}$
$b \xleftarrow{r} \{0,1\}$
$(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$
$(m_0, m_1, z) \leftarrow \mathcal{A}_1(pk)$
$y \leftarrow \mathsf{Enc}(pk, m_b)$
$\hat{b} \leftarrow \mathcal{A}_2(y, z)$
Output $(\hat{b} \stackrel{?}{=} b)$

$\underline{\mathsf{AE}(\Pi, \mathcal{A}, \lambda)}$
$b \xleftarrow{r} \{0,1\}$
$K \leftarrow \mathsf{KeyGen}(1^\lambda)$
$\hat{b} \leftarrow \mathcal{A}^{\mathcal{E}^{\mathsf{ae}}_{K,b}(\cdot,\cdot), \mathcal{D}^{\mathsf{ae}}_{K,b}(\cdot)}(1^\lambda)$
Output $(\hat{b} \stackrel{?}{=} b)$.

Fig. 1. Experiments for Definitions 1 and 3

thus any secret key can be encrypted using any public key. For symmetric encryption schemes we will always have $M \subset \{0,1\}^*$.

By $\nu(k)$ we denote some *negligible* function, i.e., one such that, for all $c > 0$ and all sufficiently large k, $\nu(k) < 1/k^c$. We abbreviate probabilistic polynomial time as PPT.

2.1 Standard Security Definitions

Public-key encryption. We recall the standard notion of indistinguishability of encryptions under a chosen-plaintext attack due to Goldwasser and Micali [22].

Definition 1 (IND-CPA)**.** *Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme for the message space M. For $b \in \{0,1\}$, $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and $\lambda \in \mathbb{N}$, let the random variable $\mathsf{IND\text{-}CPA}(\Pi, \mathcal{A}, \lambda)$ be defined by the probabilistic algorithm described on the left side of Figure 1. We denote the* IND-CPA advantage of $\mathcal{A}$ *by $\mathsf{Adv}^{\mathsf{cpa}}_{\Pi,\mathcal{A}}(\lambda) = 2 \cdot \Pr[\mathsf{IND\text{-}CPA}(\Pi, \mathcal{A}, \lambda) = 1] - 1$. We say that Π is* IND-CPA secure *if $\mathsf{Adv}^{\mathsf{cpa}}_{\Pi,\mathcal{A}}(\lambda)$ is negligible for all PPT $\mathcal{A}$.*

We also consider the indistinguishability of encryptions under chosen-ciphertext attacks [34,35,19].

Definition 2 (IND-CCA)**.** *Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme for the message space M. Let the random variable $\mathsf{IND\text{-}CCA}(\Pi, \mathcal{A}, \lambda)$ be defined by an algorithm identical to $\mathsf{IND\text{-}CPA}(\Pi, \mathcal{A}, \lambda)$ above, except that both $\mathcal{A}_1$ and $\mathcal{A}_2$ have access to an oracle $\mathsf{Dec}(sk, \cdot)$ that returns the output of the decryption algorithm and $\mathcal{A}_2$ cannot query this oracle on input y. We denote the* IND-CCA advantage of $\mathcal{A}$ *by $\mathsf{Adv}^{\mathsf{cca}}_{\Pi,\mathcal{A}}(\lambda) = 2 \cdot \Pr[\mathsf{IND\text{-}CCA}(\Pi, \mathcal{A}, \lambda) = 1] - 1$. We say that Π is* IND-CCA secure *if $\mathsf{Adv}^{\mathsf{cca}}_{\Pi,\mathcal{A}}(\lambda)$ is negligible for all PPT $\mathcal{A}$.*

Symmetric-key authenticated encryption. We recall the definition of secure authenticated (symmetric-key) encryption due to [36], except that we will not require pseudorandom ciphertexts. Bellare and Namprempre [10] showed that AE implies IND-CCA, and is in fact strictly stronger. For our counterexample, we target this very strong definition of security in order strengthen our results by showing that even this does not imply weak circular security.

$\underline{\mathsf{IND\text{-}CIRC\text{-}CPA}^n(\Pi, \mathcal{A}, \lambda)}$
$b \xleftarrow{r} \{0,1\}$
For $i = 1$ to n:
  $(pk_i, sk_i) \leftarrow \mathsf{KeyGen}(1^\lambda)$
If $b = 1$ then
  $\mathbf{y} \leftarrow \mathsf{EncCycle}(\mathbf{pk}, \mathbf{sk})$
Else
  $\mathbf{y} \leftarrow \mathsf{EncZero}(\mathbf{pk}, \mathbf{sk})$
$\hat{b} \leftarrow \mathcal{A}(\mathbf{pk}, \mathbf{y})$
Output $(\hat{b} \stackrel{?}{=} b)$

$\underline{\mathsf{EncCycle}(\mathbf{pk}, \mathbf{sk})}$
For $i = 1$ to n
  $y_i \leftarrow \mathsf{Enc}(pk_i, sk_{(i \bmod n)+1})$
Output $\mathbf{y}$

$\underline{\mathsf{IND\text{-}WCIRC\text{-}CPA}^n(\Pi, \mathcal{A}, \lambda)}$
$b \xleftarrow{r} \{0,1\}$
For $i = 1$ to n:
  $(pk_i, sk_i) \leftarrow \mathsf{KeyGen}(1^\lambda)$
$\mathbf{y} \leftarrow \mathsf{EncCycle}(\mathbf{pk}, \mathbf{sk})$
$(j, m_0, m_1, z) \leftarrow \mathcal{A}_1(\mathbf{pk}, \mathbf{y})$
$y \leftarrow \mathsf{Enc}(pk_j, m_b)$
$\hat{b} \leftarrow \mathcal{A}_2(y, z)$
Output $(\hat{b} \stackrel{?}{=} b)$

$\underline{\mathsf{EncZero}(\mathbf{pk}, \mathbf{sk})}$
For $i = 1$ to n
  $y_i \leftarrow \mathsf{Enc}(pk_i, 0^{|sk_{(i \bmod n)+1}|})$
Output $\mathbf{y}$

Fig. 2. Experiments for Definitions 4 and 5. Each is defined with respect to a message space M, and we assume that $m_0, m_1 \in M$ always. We write $\mathbf{pk}$, $\mathbf{sk}$, and $\mathbf{y}$ for $(pk_1, \ldots, pk_n)$, $(sk_1, \ldots, sk_n)$ and $(y_1, \ldots, y_n)$ respectively

Definition 3 (AE). *Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a symmetric-key encryption scheme for the message space M. Let the random variable $\mathsf{AE}(\Pi, \mathcal{A}, \lambda)$ be defined by the probabilistic algorithm described on the right side of Figure 1. In the experiment, the oracle $\mathcal{E}^{\mathrm{ae}}_{K,b}(\cdot,\cdot)$ takes as input a pair of equal-length messages (m_0, m_1) and computes $\mathsf{Enc}(K, m_b)$. The oracle $\mathcal{D}^{\mathrm{ae}}_{K,b}(\cdot)$ takes as input a ciphertext c and computes $\mathsf{Dec}(K, c)$ if $b = 1$ and always returns $\perp$ if $b = 0$. The adversary is not allowed to submit any ciphertext to $\mathcal{D}^{\mathrm{ae}}_{K,b}(\cdot)$ that was previously returned by $\mathcal{E}^{\mathrm{ae}}_{K,b}(\cdot,\cdot)$. We denote the AE advantage of $\mathcal{A}$ by $\mathsf{Adv}^{\mathsf{ae}}_{\Pi,\mathcal{A}}(\lambda) = 2 \cdot \Pr[\mathsf{AE}(\Pi, \mathcal{A}, \lambda) = 1] - 1$. We say that Π is AE secure if $\mathsf{Adv}^{\mathsf{ae}}_{\Pi,\mathcal{A}}(\lambda)$ is negligible for all PPT $\mathcal{A}$.*

2.2 Circular Security Definitions

We next give definitions for circular security of public-key and symmetric-key encryption. These definitions are variants of the Key-Dependent Message (KDM) security notion of Black et al. [11]. By restricting the adversary's power, we make it significantly harder for us to devise a counterexample and thus prove a stronger negative result.[2]

Definition 4 ($\mathsf{IND\text{-}CIRC\text{-}CPA}^n$). *Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme for the message space M. For $b \in \{0,1\}$, integer $n > 0$, adversary $\mathcal{A}$ and $\lambda \in \mathbb{N}$, let the random variable $\mathsf{IND\text{-}CIRC\text{-}CPA}^n(\Pi, \mathcal{A}, \lambda)$ be*

[2] If we allowed the adversary to obtain encryptions of any affine function of the secret keys, as is done in [27,13], then we could devise a trivial counterexample where the adversary uses 1-cycles to break the system.

defined by the probabilistic algorithm on the left side of Figure 2. We denote the $\mathsf{IND\text{-}CIRC\text{-}CPA}^n$ advantage of $\mathcal{A}$ *by*

$$\mathsf{Adv}^{n\text{-}\mathsf{circ}\text{-}\mathsf{cpa}}_{\Pi,\mathcal{A}}(\lambda) = 2 \cdot \Pr[\mathsf{IND\text{-}CIRC\text{-}CPA}^n(\Pi, \mathcal{A}, \lambda) = 1] - 1.$$

We say that Π *is* $\mathsf{IND\text{-}CIRC\text{-}CPA}^n$ secure *if* $\mathsf{Adv}^{n\text{-}\mathsf{circ}\text{-}\mathsf{cpa}}_{\Pi,\mathcal{A}}(\lambda)$ *is negligible for all PPT* $\mathcal{A}$*.*

One could augment this definition by modifying the $\mathsf{IND\text{-}CIRC\text{-}CPA}^n$ experiment to allow for a challenge "left-or-right" query as in $\mathsf{IND\text{-}CPA}$. While this is a quite natural modification, it only strengthens the definition, and we are interested in studying the weakest notions for which we can give a separation. Next we give a definition of *weak* circular security of public-key encryption.

Definition 5 ($\mathsf{IND\text{-}WCIRC\text{-}CPA}^n$)**.** *Let* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public-key encryption scheme for the message space* M*. For* $b \in \{0,1\}$*, integer* $n > 0$*, adversary* $\mathcal{A}$ *and* $\lambda \in \mathbb{N}$*, let the random variable* $\mathsf{IND\text{-}WCIRC\text{-}CPA}^n(\Pi, \mathcal{A}, \lambda)$ *be defined by probabilistic algorithm on the center of Figure 2. We denote the* $\mathsf{IND\text{-}WCIRC\text{-}CPA}^n$ advantage of $\mathcal{A}$ *by*

$$\mathsf{Adv}^{n\text{-}\mathsf{wcirc}\text{-}\mathsf{cpa}}_{\Pi,\mathcal{A}}(\lambda) = 2 \cdot \Pr[\mathsf{IND\text{-}WCIRC\text{-}CPA}^n(\Pi, \mathcal{A}, \lambda) = 1] - 1.$$

We say that Π *is* $\mathsf{IND\text{-}WCIRC\text{-}CPA}^n$ secure *if the function* $\mathsf{Adv}^{n\text{-}\mathsf{wcirc}\text{-}\mathsf{cpa}}_{\Pi,\mathcal{A}}(\lambda)$ *is negligible for all PPT* $\mathcal{A}$*.*

Finally, we give a definition of weak circular security for symmetric encryption. We will abuse notation and also call this $\mathsf{IND\text{-}WCIRC\text{-}CPA}^n$ security, since it will be clear from the context whether or not we mean public-key and symmetric-key.

Definition 6 ($\mathsf{IND\text{-}WCIRC\text{-}CPA}^n$)**.** *Let* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *be a symmetric-key encryption scheme for the message space* M*. For* $b \in \{0,1\}$*, integer* $n > 0$*, adversary* $\mathcal{A}$ *and* $\lambda \in \mathbb{N}$*, let* $\mathsf{IND\text{-}WCIRC\text{-}CPA}^n(\Pi, \mathcal{A}, \lambda)$ *be defined by the following probabilistic algorithm:*

$\underline{\mathsf{IND\text{-}WCIRC\text{-}CPA}^n_b(\Pi, \mathcal{A}, \lambda)}$
$b \xleftarrow{r} \{0,1\}$
For $i = 1$ *to* n*:*
$K_i \leftarrow \mathsf{KeyGen}(1^\lambda)$
$\mathbf{y} \leftarrow \mathsf{EncCycle}(\mathbf{K})$
$\hat{b} \leftarrow \mathcal{A}^{\widetilde{\mathsf{Enc}}(\cdot,\cdot,\cdot)}(\mathbf{y})$
Output $(\hat{b} \stackrel{?}{=} b)$

$\underline{\mathsf{EncCycle}(\mathbf{K})}$
For $i = 1$ *to* n
$y_i \leftarrow \mathsf{Enc}(K_i, K_{(i \bmod n)+1})$
Output $\mathbf{y}$

$\underline{\widetilde{\mathsf{Enc}}(j, m_0, m_1)}$
Return $\mathsf{Enc}(K_j, m_b)$

We denote the $\mathsf{IND\text{-}WCIRC\text{-}CPA}^n$ advantage of $\mathcal{A}$ *by*

$$\mathsf{Adv}^{n\text{-}\mathsf{wcirc}\text{-}\mathsf{cpa}}_{\Pi,\mathcal{A}}(\lambda) = 2 \cdot \Pr[\mathsf{IND\text{-}WCIRC\text{-}CPA}^n(\Pi, \mathcal{A}, \lambda) = 1] - 1.$$

We say that Π *is* $\mathsf{IND\text{-}WCIRC\text{-}CPA}^n$ secure *if* $\mathsf{Adv}^{n\text{-}\mathsf{wcirc}\text{-}\mathsf{cpa}}_{\Pi,\mathcal{A}}(\lambda)$ *is negligible for all PPT* $\mathcal{A}$*.*

Discussion. In both the IND-CPA and IND-CIRC-CPA notions, the adversary must distinguish an encryption (or encryptions) of a special message from the encryption of zero. This choice of the message zero is arbitrary. We keep it in the statement of our definition to be consistent with [13]; however, it is important to note, for systems such as ours where zero is not in the message space, that zero can be replaced by any constant message for an equivalent definition. Acar et al. [2] use an equivalent definition where zero is replaced by a fresh random message.

We will not need to define a notion of security to withstand *circular and chosen-ciphertext attacks*, because we are able to show a stronger negative result. In Section 4.4, we provide an IND-CCA-secure cryptosystem, which is provably not IND-CIRC-CPA-secure. In other words, we are able to devise a peculiar cryptosystem: one that withstands all chosen-ciphertext attacks, and yet breaks under a weak circular attack which does not require a decryption oracle.

3 Counterexample for Symmetric Encryption

Encryption Scheme Π_{ae}. Let $\Pi'_{\mathsf{ae}} = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ be a secure authenticated encryption scheme. To simplify our results, we assume that $\mathsf{KeyGen}'(1^\lambda)$ outputs a uniformly random key K in $\{0,1\}^\lambda$, that the message space $M' = \{0,1\}^*$, and that ciphertexts output by $\mathsf{Enc}'(K,m)$ are always in $\{0,1\}^{p(|m|)}$, where p is some polynomial that depends on λ. We also assume that the first λ bits of a ciphertext are *never* equal to K. All of these assumptions can be removed via straightforward and standard modifications to our arguments below.

Fix a positive integer n. We now construct our counterexample scheme, denoted $\Pi_{\mathsf{ae}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. We will take $\mathsf{KeyGen} = \mathsf{KeyGen}'$, i.e., Π_{ae} also uses keys randomly chosen from $\{0,1\}^\lambda$. The message-space of Π_{ae} will consist of $M = \{0,1\}^\lambda \cup \{0,1\}^{np(\lambda)}$, bit strings of length either λ or $np(\lambda)$. The algorithms Enc and Dec are defined as follows.

$\underline{\mathsf{Enc}(K,m)}$
If $\mathsf{IsCycle}(K,m)$ then
  Output $K \parallel m$
Else
  Output $\mathsf{Enc}'(K,m)$

$\underline{\mathsf{Dec}(K,c)}$
If $c = K \parallel \tilde{m}$ then
  Output $\tilde{m}$
Else
  Output $\mathsf{Dec}'(K,c)$

$\underline{\mathsf{IsCycle}(K,m)}$
If $|m| \neq np(\lambda)$
  Return false
Parse m as $(c_1, \ldots, c_n)$
$K_2 \leftarrow \mathsf{Dec}'(K, c_1)$
For $i = 2$ to n
  $K_{i \bmod n+1} \leftarrow \mathsf{Dec}'(K_i, c_i)$
Return $(K_1 \stackrel{?}{=} K)$

Decryption is always correct. This follows from our assumption that Enc' will never output a ciphertext that contains K as a prefix. We first establish the AE security of our scheme.

Theorem 1. *Encryption scheme* Π_{ae} *is* AE *secure whenever* Π'_{ae} *is* AE *secure. (Proof in the full version of this work [17].)*

The proof proceeds by showing that computing an encrypted key-cycle during the AE game is equivalent to recovering the secret key. From there we can reduce the security of Π_{ae} to Π'_{ae} easily.

Curiously, Theorem 1 is no longer true if one replaces AE security with a symmetric version of IND-CPA security for both Π_{ae} and Π'_{ae}. Namely, some type of chosen-ciphertext security is required on Π'_{ae} to prove even chosen-plaintext security of Π_{ae}. Intuitively, this is because it might be possible for an adversary to compute an encrypted key-cycle on its own if the scheme is only IND-CPA-secure, but *not* if the scheme is AE-secure. In fact, the work of Boneh et al. [13] gives an explicit example of a scheme where the adversary can compute a cycle himself.

The Attack. We now show that Π_{ae} is not circular-secure for n cycles, even in a weak sense.

Theorem 2. Π_{ae} *is not* IND-WCIRC-CPA^n *secure.*

Proof. We give an explicit adversary $\mathcal{A}$ that has advantage negligibly close to 1. The adversary takes as input the encrypted key-cycle $\mathbf{y}$ in the IND-WCIRC-CPA^n game. It queries $\widetilde{\mathsf{Enc}}(1, m_0, m_1)$, where $m_0 = \mathbf{y}$ and m_1 is a random message of the same length. Let y be the ciphertext returned by the oracle.

At this point, there are many ways to proceed; perhaps the simplest is to observe that the *length* of y depends on the challenge bit b. This is because, if $b = 0$, then $m_0 = \mathbf{y}$ was encrypted, resulting in $y = K \parallel \mathbf{y}$, which is $\lambda + np(\lambda)$ bits long. If $b = 1$ then y was computed by running $\mathsf{Enc}'(K, m_1)$, which will be $p(|m_1|) = p(np(\lambda))$ bits long *if* $\mathsf{IsCycle}(K, m_1)$ *returns* false. Thus, as long as $\mathsf{IsCycle}(K, m_1)$ returns false, $\mathcal{A}_2$ can compute the value of b by measuring y's length.

But why should $\mathsf{IsCycle}(K, m_1)$ return false? This follows from the AE security of Π'_{ae}. Let us parse m_1 into $(c_1, \ldots, c_n)$, where each $c_i \in \{0,1\}^{p(\lambda)}$ is random. When $\mathsf{IsCycle}(K, m_1)$ returns true, it must be that $\mathsf{Dec}'(K, c_1)$ did not return $\perp$. But if this happens, then we can construct an adversary to break the AE security of Π'_{ae}. The adversary simply queries $\mathcal{D}^{\mathrm{ae}}_{K,b}(\cdot)$ at a random point, observes if it returns $\perp$ or not, and outputs $\hat{b} = 0$ or 1 depending on this observation.

We note that we could design an encryption scheme that does not have this type of ciphertext-length behavior by giving a different attack that abuses the fact that K is present in the ciphertext in one case, but not the other. We have chosen to present the attack this way for simplicity only.

4 Counterexamples for Public-Key Encryption

4.1 Preliminaries and Algebraic Setting

Bilinear Groups. We work in a bilinear setting where there exists an efficient mapping function $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ involving groups of the same prime order p.

Two algebraic properties required are that: (1) if g generates $\mathbb{G}_1$ and h generates $\mathbb{G}_2$, then $e(g,h) \neq 1$ and (2) for all $a, b \in \mathbb{Z}_p$, it holds that $e(g^a, h^b) = e(g,h)^{ab}$.

Decisional Diffie-Hellman Assumption (DDH): Let $\mathbb{G}$ be a group of prime order $p \in \Theta(2^\lambda)$. For all PPT adversaries $\mathcal{A}$, the following probability is $1/2$ plus an amount negligible in λ:

$$\Pr\left[\begin{matrix} g, z_0 \leftarrow \mathbb{G}; a, b \leftarrow \mathbb{Z}_p; z_1 \leftarrow g^{ab}; d \leftarrow \{0,1\}; \\ d' \leftarrow \mathcal{A}(g, g^a, g^b, z_d) : d = d' \end{matrix}\right].$$

Strong External Diffie-Hellman Assumption (SXDH): Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be bilinear groups. The SXDH assumption states that the DDH problem is hard in both $\mathbb{G}_1$ and in $\mathbb{G}_2$. This implies that there does *not* exist an efficiently computable isomorphism between these two groups. The SXDH assumption appears in many prior works, such as [21,38,32,12,8,6,24,9,25,2].

Indistinguishability and Pseudorandom Generators.

Definition 7 (Indistinguishability). *Two ensembles of probability distributions $\{X_k\}_{k\in\mathbb{N}}$ and $\{Y_k\}_{k\in\mathbb{N}}$ with index set $\mathbb{N}$ are said to be* computationally indistinguishable *if for every polynomial-size circuit family $\{D_k\}_{k\in\mathbb{N}}$, there exists a negligible function ν such that*

$$|\Pr[x \leftarrow X_k : D_k(x) = 1] - \Pr[y \leftarrow Y_k : D_k(y) = 1]|$$

is less than $\nu(k)$. We denote such sets $\{X_k\}_{k\in\mathbb{N}} \overset{c}{\approx} \{Y_k\}_{k\in\mathbb{N}}$.

Definition 8 (Pseudorandom Generator [30]). *Let U_x denote the uniform distribution over $\{0,1\}^x$. Let $\ell(\cdot)$ be a polynomial and let G be a deterministic polynomial-time algorithm such that for any input $s \in \{0,1\}^n$, algorithm G outputs a string of length $\ell(n)$. We say that G is a* pseudorandom generator *if the following two conditions hold:*

- *(Expansion:) For every n, it holds that $\ell(n) > n$.*
- *(Pseudorandomness:) For every n, $\{U_{\ell(n)}\}_n \overset{c}{\approx} \{s \leftarrow U_n : G(s)\}_n$.*

The constructions of Section 4.2 use a PRG where the domain of the function is an exponentially-sized cyclic group.

4.2 Encryption Scheme Π_{cpa}

We now describe an encryption scheme $\Pi_{\mathsf{cpa}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. It is set in asymmetric bilinear groups $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ of prime order p where we assume that the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ are distinct and that the DDH assumption holds in both. We assume that a single set of group parameters $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, h)$, where $\mathbb{G}_1 = \langle g \rangle, \mathbb{G}_2 = \langle h \rangle$, will be shared across all keys generated at a given security level and are implicitly provided to all algorithms.

The message space is $\mathcal{M} = \{0,1\} \times \mathbb{Z}_p^* \times \mathbb{Z}_p^*$. Let $\mathsf{encode} : \mathcal{M} \to \{0,1\}^{\ell(\lambda)}$ and $\mathsf{decode} : \{0,1\}^{\ell(\lambda)} \to \mathcal{M}$ denote an invertible encoding scheme where $\ell(\lambda)$ is the polynomial length of the encoded message. Let $F : \mathbb{G}_T \to \{0,1\}^{\ell(\lambda)}$ be a pseudorandom generator secure under the Decisional Diffie Hellman assumption. (Recall that pseudorandom generators can be constructed from any one-way function [28].)

$\mathsf{KeyGen}(1^\lambda)$. The key generation algorithm selects a random bit $\beta \leftarrow \{0,1\}$ and random values $a_1, a_2 \leftarrow \mathbb{Z}_p^*$. The secret key is set as $sk = (\beta, a_1, a_2)$. We note that $sk \in \mathcal{M}$. The public key is set as:

$$pk = \begin{cases} (0, e(g,h)^{a_1}, g^{a_2}) \in \{0,1\} \times \mathbb{G}_T \times \mathbb{G}_1 & \text{if } \beta = 0 \\ (1, e(g,h)^{a_1}, h^{a_2}) \in \{0,1\} \times \mathbb{G}_T \times \mathbb{G}_2 & \text{if } \beta = 1. \end{cases}$$

$\mathsf{Encrypt}(pk, M)$. The encryption algorithm parses the public key $pk = (\beta, Y_1, Y_2)$, where Y_2 may be in $\mathbb{G}_1$ or $\mathbb{G}_2$ depending on the structure of the public key, and message $M = (\alpha, m_1, m_2) \in \mathcal{M}$. Note that m_1 and m_2 cannot be zero, but these values can be easily included in the message space by a proper encoding.
Select random $r \leftarrow \mathbb{Z}_p$ and $R \leftarrow \mathbb{G}_T$. Set $I = F(R) \oplus \mathsf{encode}(M)$.
Output the ciphertext C as:

$$C = \begin{cases} (g^r,\ R \cdot Y_1^r,\ Y_2^{rm_2} \cdot g^{m_1},\ I) & \text{if } \beta = 0; \\ (h^r,\ R \cdot Y_1^r,\ Y_2^{rm_2},\ I) & \text{if } \beta = 1. \end{cases}$$

We note that in the first case, $C \in \mathbb{G}_1 \times \mathbb{G}_T \times \mathbb{G}_1 \times \{0,1\}^{\ell(\lambda)}$, while in the second $C \in \mathbb{G}_2 \times \mathbb{G}_T \times \mathbb{G}_2 \times \{0,1\}^{\ell(\lambda)}$.

$\mathsf{Decrypt}(sk, C)$. The decryption algorithm parses the secret key $sk = (\beta, a_1, a_2)$ and the ciphertext $C = (C_1, C_2, C_3, C_4)$. Next, it computes:

$$R = \begin{cases} (C_2 / e(C_1, h))^{a_1} & \text{if } \beta = 0; \\ (C_2 / e(g, C_1))^{a_1} & \text{if } \beta = 1. \end{cases}$$

Then it computes $M' = F(R) \oplus C_4 \in \{0,1\}^{\ell(\lambda)}$ and outputs the message $M = \mathsf{decode}(M')$.

Discussion. Like the circular-secure scheme of Boneh et al. [13], the above cryptosystem is a variation on El Gamal [20]. It is a practical system, which on first glance might be somewhat reminiscent of schemes the readers are used to seeing in the literature. The scheme includes a few "artificial" properties: (1) placing a public key in either $\mathbb{G}_1$ or $\mathbb{G}_2$ at random and (2) the fact that the ciphertext value C_3 is unused in the decryption algorithm. We will shortly see that these features are "harmless" in a semantic-security sense, but very useful for recovering the secret keys of the system in the presence of a two cycle. While it is not unusual for counterexamples to have artificial properties (e.g., [16,23]), we

can address these points as well.[3] In the full version of this work [17], we show that property (1) can be removed by doubling the length of the ciphertext. For property (2), we observe that many complex protocols such as group signatures (e.g., [12]) combine ciphertexts with other components that are unused in decryption but are quite important to the protocol as a whole. Thus, we believe our counterexample is not that far fetched. It is possible that such an attack could exist on one of today's commonly-used encryption algorithms.

We first show that Π_{cpa} meets the standard notion of CPA security.

Theorem 3. *Encryption scheme* Π_{cpa} *is* IND-CPA *secure under the Decisional Diffie-Hellman Assumption in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *(SXDH).*

The proof is given in the full version of this work [17]. It is relatively standard and involves repeated applications of the DDH assumption and PRG security.

4.3 The Attack

Despite being IND-CPA-secure, cryptosystem Π_{cpa} is not even weakly circular secure for 2-cycles. Specifically, given a circular encryption of two keys, we show that an adversary can distinguish another ciphertext with advantage 1/2. Our adversary actually does much more than this: with probability 1/2 over the coins used in key generation, *it can recover both secret keys.*

This is the first circular attack that allows the adversary to recover the secret keys. (In the full version of this work [17], we discuss how to improve these probabilities to almost 1.) Our attack combines elements of both ciphertexts in an attempt to recover sk_A, which can then be used to decrypt the first ciphertext and obtain sk_B. It is counterintuitive that this is possible, given that it is easy to see that IND-CPA-security guarantees that it is safe for *one* of them to send their message.

Theorem 4. Π_{cpa} *is not* IND-WCIRC-CPA^2*-secure.*

Proof. We give PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ such that $\mathsf{Adv}^{\text{2-wcirc-cpa}}_{\Pi_{\mathsf{cpa}},\mathcal{A}}(\lambda)$ is equal to 1/2. Since IND-WCIRC-CPA security requires that this advantage be negligible, this attack breaks security. The adversary proceeds as follows. The first stage of the adversary, $\mathcal{A}_1$, obtains the two public keys, which we will write as pk_A and pk_B, and an encrypted cycle, which we will write as (C_A, C_B).

If both keys have $\beta = 0$ or $\beta = 1$ (call this event E_1), the adversary aborts and instructs the second stage ($\mathcal{A}_2$) to output a random bit. Since the two keys are independently generated by the challenger, this event will occur with probability exactly 1/2. Below we will condition on E_1 not happening, and wlog assume that $pk_A = (0, e(g,h)^{a_1}, g^{a_2})$ and $pk_B = (1, e(g,h)^{b_1}, h^{b_2})$. The corresponding secret keys $sk_A = (0, a_1, a_2)$, $sk_B = (1, b_1, b_2)$ are not known to the adversary.

[3] While our scheme is different from that of Acar et al. [2], that scheme also has similar artificial properties such as the presence of values that are not used in decryption.

We write the given ciphertexts $C_A = (c_{A,1}, c_{A,2}, c_{A,3}, c_{A,4})$ and $C_B = (c_{B,1}, c_{B,2}, c_{B,3}, c_{B,4})$. $\mathcal{A}_1$ will output two arbitrary distinct messages, and request that the challenge use pk_A. For the state passed to $\mathcal{A}_2$, it now computes:

$$X := c_{B,2} \cdot \frac{e(c_{A,1}, c_{B,3})}{e(c_{A,3}, c_{B,1})}.$$

$\mathcal{A}_1$ sets $\hat{sk}_A = \mathsf{decode}(c_{B,4} \oplus F(X))$ and passes this with the challenge messages as state to $\mathcal{A}_2$.

$\mathcal{A}_2$ receives a ciphertext y and the passed state. It parses $\hat{sk}_A$ as a secret key for Π_{cpa} and computes $\mathsf{Dec}(\hat{sk}_A, y)$, and tests if this is equal to either of the challenge messages. If so, it outputs the corresponding bit. Otherwise it outputs a random bit.

Let's explore why this test works. Write $C_A = \mathsf{Enc}(pk_A, sk_B)$ and $C_B = \mathsf{Enc}(pk_B, sk_A)$. Then:

$$\begin{aligned} C_A &= (c_{A,1}, c_{A,2}, c_{A,3}, c_{A,4}) \\ &= (g^r, R \cdot e(g,h)^{ra_1}, g^{ra_2b_2+b_1}, F(R) \oplus \mathsf{encode}(sk_B)) \\ C_B &= (c_{B,1}, c_{B,2}, c_{B,3}, c_{B,4}) \\ &= (h^s, S \cdot e(g,h)^{sb_1}, h^{sa_2b_2}, F(S) \oplus \mathsf{encode}(sk_A)) \end{aligned}$$

for some $r, s \in \mathbb{Z}_p$ and $R, S \in \mathbb{G}_T$. Then we have that:

$$\begin{aligned} X := c_{B,2} \cdot \frac{e(c_{A,1}, c_{B,3})}{e(c_{A,3}, c_{B,1})} &= S \cdot e(g,h)^{sb_1} \cdot \frac{e(g^r, h^{sa_2b_2})}{e(g^{ra_2b_2+b_1}, h^s)} \\ = S \cdot e(g,h)^{sb_1} \cdot \frac{e(g,h)^{rsa_2b_2}}{e(g,h)^{rsa_2b_2} \cdot e(g,h)^{sb_1}} &= S. \end{aligned}$$

Thus, $\mathcal{A}_1$ recovers $\hat{sk}_A = sk_A$ as $\mathsf{decode}(c_{B,4} \oplus F(S))$, and $\mathcal{A}_2$ will correctly guess bit b in this case.

Write $\hat{b}$ for the output of $\mathcal{A}_2$. We have

$$\begin{aligned} \mathsf{Adv}^{\text{2-wcirc-cpa}}_{\Pi_{\mathsf{cpa}},\mathcal{A}}(\lambda) &= 2\Pr[\hat{b} = b] - 1 \\ &= 2(\Pr[\hat{b} = b | E_1] \Pr[E_1] + \\ &\qquad \Pr[\hat{b} = b | \neg E_1] \Pr[\neg E_1]) - 1 \\ &= 2(1 \cdot 1/2 + 1/2 \cdot 1/2) - 1 \\ &= 1/2 \end{aligned}$$

This completes the proof.

4.4 Extension: A Counterexample for CCA Security

We show that there exists an IND-CCA-secure cryptosystem, which suffers a complete break when Alice and Bob trade secret keys over an insecure channel;

i.e., transmit the two-key cycle $E(pk_A, sk_B)$ and $E(pk_B, sk_A)$. Our construction follows the "double-encryption" approach to building IND-CCA systems from IND-CPA systems as pioneered by Naor and Yung [34] and refined by Dolev, Dwork and Naor [19] and Sahai [37]. Our building blocks will be:

1. The IND-CPA-secure cryptosystem $\Pi_{\mathsf{cpa}} = (G, E, D)$ from Section 4. Let $E(pk, m; r)$ be the encryption of m under public key pk with randomness r.
2. An adaptively non-malleable non-interactive zero-knowledge (NIZK) proof system with unpredictable simulated proofs and uniquely applicable proofs for the language L of consistent pairs of encryptions, defined as:

$$L = \left\{ \begin{array}{ll} (e_0, e_1, c_0, c_1) : & \exists m, r_0, r_1 \in \{0,1\}^* \text{ s.t.} \\ c_0 = E(e_0, m; r_0) \text{ and } c_1 = E(e_1, m; r_1) & \end{array} \right\}.$$

A proof system for L can be realized under relatively mild assumptions, such as the difficulty of factoring Blum integers (e.g., [37]). One complication is that the secret keys for this cryptosystem now change and the construction must be adapted accordingly, so that the secret key can still be recovered by the adversary during a circular attack. We show that this is possible.

5 Conclusion and Open Problems

In this work, we presented a natural relaxation of the circular security definition, which may prove interesting for positive results in its own right. We demonstrated that its guarantees are *not* already captured by standard definitions of encryption. To do this, we presented symmetric and public-key encryption systems that are secure in the IND-CPA and IND-CCA sense, but fail catastrophically in the presence of an encrypted cycle. This provides the first answer to the foundational question on whether IND-CCA-security captures (weak or regular) circular security for all cycles larger than self-loops. In either case, it does not.

Our work leaves open the interesting problem of finding a public-key counterexample for cycles of size ≥ 3. Secondly, while our symmetric counterexample depended only on the existence of AE-secure symmetric encryption, our public-key counterexample, like that of Acar et al. [2], required a specific bilinear map assumption. It would be highly interesting to find a counterexample assuming only that IND-CPA- or IND-CCA-secure systems exist.

Finally, we observe that our public-key counterexample contains a novel and curious property – *certain combinations of independently generated ciphertexts trigger the release of their underlying plaintext.* From Rabin's $\frac{1}{2}$-OT system to DH-DDH gap groups, the cryptographic community has a strong history of turning such oddities to an advantage. If we view a cryptosystem with this property as a new primitive, what new functionalities can be realized using it?

Acknowledgments. The authors thank Ronald Rivest for the suggestion to view the public key counterexample in Section 4 as a potential building block for other functionalities.

References

1. Abadi, M., Rogaway, P.: Reconciling two views of cryptography (the computational soundness of formal encryption). J. Cryptology 15(2), 103–127 (2002)
2. Acar, T., Belenkiy, M., Bellare, M., Cash, D.: Cryptographic Agility and Its Relation to Circular Encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 403–422. Springer, Heidelberg (2010)
3. Adão, P., Bana, G., Herzog, J., Scedrov, A.: Soundness of Formal Encryption in the Presence of Key-Cycles. In: de Capitani di Vimercati, S., Syverson, P.F., Gollmann, D. (eds.) ESORICS 2005. LNCS, vol. 3679, pp. 374–396. Springer, Heidelberg (2005)
4. Akavia, A., Goldwasser, S., Vaikuntanathan, V.: Simultaneous Hardcore Bits and Cryptography against Memory Attacks. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 474–495. Springer, Heidelberg (2009)
5. Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 595–618. Springer, Heidelberg (2009)
6. Ateniese, G., Camenisch, J., de Medeiros, B.: Untraceable RFID tags via insubvertible encryption. In: CCS 2005, pp. 92–101 (2005)
7. Backes, M., Pfitzmann, B., Scedrov, A.: Key-dependent message security under active attacks -BRSIM/UC-soundness of Dolev-Yao-style encryption with key cycles. J. of Comp. Security 16(5), 497–530 (2008)
8. Ballard, L., Green, M., de Medeiros, B., Monrose, F.: Correlation-resistant storage. Technical Report TR-SP-BGMM-050705, Johns Hopkins University, CS Dept, 2005. http://spar.isi.jhu.edu/~mgreen/correlation.pdf.
9. Belenkiy, M., Chase, M., Kohlweiss, M., Lysyanskaya, A.: P-signatures and Noninteractive Anonymous Credentials. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 356–374. Springer, Heidelberg (2008)
10. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. J. Cryptology 21(4), 469–491 (2008)
11. Black, J., Rogaway, P., Shrimpton, T.: Encryption-Scheme Security in the Presence of Key-Dependent Messages. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 62–75. Springer, Heidelberg (2003)
12. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
13. Boneh, D., Halevi, S., Hamburg, M., Ostrovsky, R.: Circular-Secure Encryption from Decision Diffie-Hellman. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 108–125. Springer, Heidelberg (2008)
14. Camenisch, J., Chandran, N., Shoup, V.: A Public Key Encryption Scheme Secure against Key Dependent Chosen Plaintext and Adaptive Chosen Ciphertext Attacks. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 351–368. Springer, Heidelberg (2009)
15. Camenisch, J.L., Lysyanskaya, A.: An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer, Heidelberg (2001)
16. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. J. of the ACM 51(4), 557–594 (2004)
17. Cash, D., Green, M., Hohenberger, S.: New definitions and separations for circular security. Cryptology ePrint Archive, Report 2010/144 (2012), http://eprint.iacr.org/2010/144

18. Dodis, Y., Kalai, Y.T., Lovett, S.: On cryptography with auxiliary input. In: STOC 2009, pp. 621–630 (2009)
19. Dolev, D., Dwork, C., Naor, M.: Nonmalleable cryptography. SIAM J. Computing 30(2), 391–437 (2000)
20. El Gamal, T.: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (1985)
21. Galbraith, S.D.: Supersingular Curves in Cryptography. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 495–513. Springer, Heidelberg (2001)
22. Goldwasser, S., Micali, S.: Probabilistic encryption. J. Comput. Syst. Sci. 28(2), 270–299 (1984)
23. Goldwasser, S., Kalai, Y.T.: On the (In)security of the Fiat-Shamir Paradigm. In: FOCS 2003, p. 102 (2003)
24. Green, M., Hohenberger, S.: Universally Composable Adaptive Oblivious Transfer. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 179–197. Springer, Heidelberg (2008)
25. Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
26. Haitner, I., Holenstein, T.: On the (Im)Possibility of Key Dependent Encryption. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 202–219. Springer, Heidelberg (2009)
27. Halevi, S., Krawczyk, H.: Security under key-dependent inputs. In: ACM CCS 2007, pp. 466–475 (2007)
28. Hastad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM J. Computing 28(4), 1364–1396 (1999)
29. Hofheinz, D., Unruh, D.: Towards Key-Dependent Message Security in the Standard Model. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 108–126. Springer, Heidelberg (2008)
30. Katz, J., Lindell, Y.: Introduction to Modern Cryptography. Chapman & Hall/CRC (2008)
31. Laud, P., Corin, R.: Sound Computational Interpretation of Formal Encryption with Composed Keys. In: Lim, J.-I., Lee, D.-H. (eds.) ICISC 2003. LNCS, vol. 2971, pp. 55–66. Springer, Heidelberg (2004)
32. McCullagh, N., Barreto, P.S.L.M.: A New Two-Party Identity-Based Authenticated Key Agreement. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 262–274. Springer, Heidelberg (2005)
33. Naor, M., Segev, G.: Public-Key Cryptosystems Resilient to Key Leakage. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 18–35. Springer, Heidelberg (2009)
34. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: STOC 1990, pp. 427–437 (1990)
35. Rackoff, C., Simon, D.R.: Non-interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)
36. Rogaway, P., Shrimpton, T.: A Provable-Security Treatment of the Key-Wrap Problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 373–390. Springer, Heidelberg (2006)
37. Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: FOCS 1999, pp. 543–553 (1999)
38. Scott, M.: Authenticated id-based key exchange and remote log-in with simple token and pin number (2002), http://eprint.iacr.org/2002/164

Correlated Product Security from Any One-Way Function

Brett Hemenway[1,⋆], Steve Lu[2,⋆⋆], and Rafail Ostrovsky[3,⋆⋆⋆]

[1] University of Michigan
[2] Stealth Software Technologies, Inc.
[3] UCLA

Abstract. It is well-known that the k-wise product of one-way functions remains one-way, but may no longer be when the k inputs are correlated. At TCC 2009, Rosen and Segev introduced a new notion known as Correlated Product secure functions. These functions have the property that a k-wise product of them remains one-way even under correlated inputs. Rosen and Segev gave a construction of injective trapdoor functions which were correlated product secure from the existence of Lossy Trapdoor Functions (introduced by Peikert and Waters in STOC 2008).

In this work we continue the study of correlated product security, and find many differences depending on whether the functions have trapdoors.

The first main result of this work shows that a family of correlated product secure functions (without trapdoors) can be constructed from any one-way function. Because correlated product secure functions are trivially one-way, this shows an equivalence between the existence of these two cryptographic primitives.

In the second main result of this work, we consider a natural decisional variant of correlated product security. Roughly, a family of functions is Decisional Correlated Product (DCP) secure if $f_1(x_1), \ldots, f_k(x_1)$ is indistinguishable from $f_1(x_1), \ldots, f_k(x_k)$ when $x_1, \ldots, x_k$ are chosen uniformly at random.

When considering DCP secure functions with trapdoors, we give a construction based on Lossy Trapdoor Functions, and show that any

⋆ bhemen@umich.edu

⋆⋆ steve@stealthsoftwareinc.com

⋆⋆⋆ R. Ostrovsky, University of California Los Angeles, Department of Computer Science and Department of Mathematics, 3732D Boelter Hall, Los Angeles CA 90095-1596, U.S., email: rafail@cs.ucla.edu. Supported in part by NSF grants 0830803, 09165174, 1065276, 1118126 and 1136174, US-Israel BSF grant 2008411, OKAWA Foundation Research Award, IBM Faculty Research Award, Xerox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, and Lockheed-Martin Corporation Research Award. This material is based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11-1-0392. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 558–575, 2012.

DCP secure function family with trapdoor satisfies the security requirements for Deterministic Encryption as defined by Bellare, Boldyreva and O'Neill in CRYPTO 2007. In fact, we also show that definitionally, DCP secure functions with trapdoors are a strict subset of Deterministic Encryption functions by showing an example of a Deterministic Encryption function which according to the definition is not a DCP secure function.

Keywords: Correlated Product Security, Lossy Trapdoor Functions, Deterministic Encryption.

1 Introduction

If f and g are one-way functions on some domain X, it follows immediately that $(x, y) \mapsto (f(x), g(y))$ is a one-way function. On the other hand, it is well-known that $x \mapsto (f(x), g(x))$ may not be. The RSA function provides a simple example of this observation. The RSA assumption posits that $x \mapsto x^e \mod n$ is a one-way function. Given $x^{e_1} \mod n$, and $x^{e_2} \mod n$, the extended euclidean algorithm provides an efficient means of computing $x^{\gcd(e_1,e_2)} \mod n$, so if $\gcd(e_1, e_2) = 1$, the map $x \mapsto (x^{e_1}, x^{e_2}) \mod n$, is trivially invertible, even though its constituents are believed to be one-way.

In [RS09], Rosen and Segev formalized the notion of Correlated Product (CP) Security. They called a family of one-way trapdoor functions CP secure if they remained one-way when evaluated on correlated (and in particular, repeated) inputs. Rosen and Segev were motivated by the construction of IND-CCA secure encryption based on Lossy Trapdoor Functions (LTDFs) given by Peikert and Waters in [PW08]. Rosen and Segev showed that CP security is exactly the property needed to prove security of the Peikert and Waters construction.

Correlated Product security is an appealing notion because it is easy to define and appears to be a significantly weaker property than the *statistical* lossiness requirement of Lossy Trapdoor Functions. Despite this appearance of relative simplicity there have been few examples of correlated product secure functions that are not Lossy Trapdoor Functions. The notable exceptions are the constructions given in [Pei09] and [FGK+10].

This work continues the study of Correlated Product Secure Functions. We introduce a natural decisional variant of correlated product security, and show how this notion of Decisional Correlated Product Security provides connections to many areas in cryptography.

1.1 Our Results

In this work, we introduce (in Section 3) the notion of Decisional Correlated Product (DCP) security, which strengthens the definition of Rosen and Segev. We argue that this is a natural stepping-stone between Lossy Trapdoor Functions and Correlated Product secure functions. Intuitively, these are families of functions such that for any k functions $f_1, \ldots, f_k$, the distributions $\{(f_1(x_1), \ldots, f_k(x_1))\}$ and $\{(f_1(x_1), \ldots, f_k(x_k))\}$ are indistinguishable

when $x_1, \ldots, x_k$ are chosen uniformly at random. Like correlated product security, decisional correlated product security can be defined for distributions other than the repetition distribution. We have focused on the case of the repetition distribution because it is conceptually simple while still capturing the essence of the problem. The repetition distribution is also the distribution that is necessary for applications to IND-CCA encryption [PW08, RS09].

Our results can be divided into three categories.

1. **Connections to Correlated Product Security:**
   We begin by examining the connections between Correlated Product (CP) and Decisional Correlated Product (DCP) security.

   From the definition of DCP security, it is clear that a family of constant functions is DCP secure, so for non-trivial results, we either specify that the functions be (individually) one-way or that they be injective with large domain. It turns out that, under either one of these assumptions, these families can be shown to also be Correlated Product secure. This is proven in Section 4 as the following lemmas:

   Lemma 2. If $\mathcal{F} = (G, F)$ is a family of k-DCP secure functions with super-polynomial size domain that are injective, then $\mathcal{F}$ is k-correlated product secure.

   Lemma 3. If $\mathcal{F} = (G, F)$ is a family of k-DCP secure one-way functions, then $\mathcal{F}$ is k-correlated product secure.
2. **DCP Secure Functions Without Trapdoors:**
   Our first main result considers families of one-way functions that are DCP secure. We show that such families are automatically (plain) Correlated Product secure, and demonstrate a construction from any pseudorandom function family. Due to the celebrated fact that a PRF family can be constructed from any one-way function ([GGM86, ILL89, HILL99]), we obtain an equivalence between the existence of one-way functions, DCP secure one-way function families, and CP secure function families. This is proven in Section 5 as the following theorem:

   Theorem 1. The following statements are equivalent:
   (a) One-way functions exist.
   (b) k-DCP secure families of one-way functions exist.
   (c) k-CP secure families of one-way functions exist.

   Theorem 1 shows that without a trapdoor, correlated product security essentially, is no stronger than simple one-wayness. This is somewhat surprising given the results of Vahlis [Vah10] that show that Correlated Product secure functions with trapdoor cannot be constructed from enhanced one-way trapdoor permutations. It is also somewhat surprising since lossy functions (without trapdoor) have not proven to be significantly easier to construct than lossy trapdoor functions.
3. **DCP Secure Functions With Trapdoors:**
   Our second main result considers DCP secure function families which also have trapdoor. We investigate the connection between this and other primitives. In Section 6, we show a construction of these one-way trapdoor DCP

secure families from sufficiently lossy LTDFs. This is stated as the following theorem:

Theorem 2. Let $\epsilon(\lambda)$ be any function such that $1/2^{\epsilon(\lambda)}$ is negligible in λ. Let $\mathcal{F} = (G, F)$ be a family of LTDFs on domain $\{0,1\}^\lambda$, with residual leakage[1] at most $\frac{\lambda+2-2\log(1/\epsilon)}{k}$. Then functions of the form $F_s(h(x))$ form a family of k-DCP trapdoor functions, where h is an injective pairwise independent hash function.

Finally, in Section 7, we show that these families definitionally satisfy the security requirements of Deterministic Encryption, but the converse is not true in general. Using the notion of *PRIV1 security* for Deterministic Encryption, which we will recall later, we have:

Theorem 3. DCP secure function families with trapdoor are also PRIV1 secure deterministic encryption schemes.

1.2 Previous Work

In [PW08] Peikert and Waters introduced a new paradigm for constructing IND-CCA secure encryption based on the newly defined primitive Lossy Trapdoor Functions (LTDFs). Their construction of IND-CCA was natural and appealing, but LTDFs proved difficult to construct because of their strong statistical lossiness properties. Despite the power of LTDFs, in [PW08] they were able to give constructions from DDH and Lattice-based assumptions, and the authors of [BFO08] and [RS08, FGK+10] (independently) found identical efficient constructions of LTDFs from Paillier's Decisional Composite Residuousity Assumption.

In [RS09], Rosen and Segev examined which properties of LTDFs were necessary to construct IND-CCA secure encryption via the methods in [PW08]. With this goal, they defined Correlated Product secure functions, and gave a construction of IND-CCA secure encryption from Correlated Product secure functions with trapdoor paralleling the construction in [PW08]. One of the primary difficulties in constructing Lossy Trapdoor Functions is creating functions the necessary *statistical* lossiness property (i.e. that the image of the function is significantly smaller than the domain). Correlated Product secure functions do not have these statistical requirements, and thus should be easier to construct than LTDFs. This intuition was reinforced in [RS09] where they showed that LTDFs are Correlated Product secure, and showed a black-box separation in the opposite direction. Correlated Product secure functions remain difficult to realize, however, and the recent results of Vahlis [Vah10], show a black-box separation between (enhanced) one-way trapdoor permutations and Correlated Product Secure functions.

In 2007, Bellare, Boldyreva, and O'Neill [BBO07] introduced a new notion known as Deterministic Encryption (DE). The deterministic property of the

[1] Recall that the residual leakage is defined to be the average number of bits leaked about the input when the function is in lossy mode. In particular, the residual leakage is defined to be the log of the size of the image of the function in lossy mode.

encryption affords the scheme many practical applications, such as searchable encryption, but at the same time requires new security definitions. Subsequent works [BFO08, BFOR08] demonstrate equivalences between various definitions of DE and show that the existence of a sufficiently lossy LTDFs imply the existence of deterministic encryption, which in turn implies the existence of IND-CCA secure cryptosystems.

The works [BFO08, BFOR08] show many different relationships between DE and other primitives. Indeed, they show that any LTDF is almost immediately a DE scheme, and show how a weaker notion of DE can be constructed from any one-way trapdoor permutation.

In [ABBC10] Acar et al. studied the notion of *cryptographic agility*, where families of cryptographic primitives are said to be agile if they remain secure when the same key is re-used across families. While this is also a notion regarding correlated security, it does not appear to be connected to DCP security. Cryptographic agility refers to the security of correlated *keys* across *different* families of primitives, while DCP security refers to the one-wayness of functions *from the same family* when evaluated on correlated *inputs*.

The notion of security under correlated inputs has been studied in other contexts as well. In [IKNP03], Ishai et al. defined the notion of *correlation robustness* and used correlation robust functions to efficiently extend the number of independent oblivious transfer pairs available in a secure multiparty protocol. Correlation robustness was then used to create cryptosystems secure under related key attacks [AHI11, GOR11]. The notion of correlation robustness is distinct from the notion of correlated product security that is studied in this work. Correlation robustness studies the security of *a single function* applied on correlated inputs, while correlated product security studies the notion of *different* functions applied to correlated inputs. This distinction makes the constructions and applications quite different between the two areas.

2 Preliminaries

If A is a PPT machine, then we use $a \xleftarrow{\$} A$ to denote running the machine A and obtaining an output, where a is distributed according to the internal randomness of A. For a PPT machine A, we use $\mathsf{coins}(A)$ to denote the distribution of the internal randomness of A. So the distributions $\{a \xleftarrow{\$} A\}$ and $\{r \xleftarrow{\$} \mathsf{coins}(A) : a = A(r)\}$ are identical. If R is a set, we use $r \xleftarrow{\$} R$ to denote sampling uniformly from R. If X and Y are families distributions indexed by a security paramete λ, we use $X \approx_s Y$ to mean the two distributions are statistically close i.e. the statistical distance between X and Y is negligible in λ. We use $X \approx_c Y$ to mean that the distributions are computationally close, i.e. no PPT distinguisher with oracle access to the distribution has a non-negligible distinguishing advantage. We will need an extension of the leftover hash lemma known as the Crooked Leftover Hash Lemma [BFO08].

Lemma 1 (Crooked Leftover Hash Lemma [BFO08]). *Let $\mathcal{H}$ be a pairwise independent hash family, such that for all $h \in \mathcal{H}$, $h : X \to X$. Let $f : X \to Y$, and let Z be any random variable independent of h and D_X a distribution over X such that the min entropy $\tilde{H}_\infty(D_X|Z) \geq \log|Y| + 2\log(1/\epsilon) - 2$. Then if we define $\Lambda_1 = \{h \xleftarrow{\$} \mathcal{H}; x \xleftarrow{\$} D_X : (h, f(h(x)), Z)\}$, and $\Lambda_2 = \{h \xleftarrow{\$} \mathcal{H}; y \xleftarrow{\$} Y : (h, f(h(U_X), Z))\}$, we have $\Delta(\Lambda_1, \Lambda_2) \leq \epsilon$.*

Notice that the Crooked Leftover Hash Lemma does *not* imply that $h(D_X)$ is close to U_X, and indeed this may not be the case.

2.1 Correlated Product Security

In this section, we review the definition of Correlated Product security, first defined in [RS09]. We begin by defining the k-wise product of a Function Family.

Definition 1 (k-wise product). *Let $\mathcal{F} = (G, F)$ be a collection of efficiently computable functions. G is a (randomized) algorithm which takes as input a size parameter 1^λ and generates a key (or seed) s for F. Each function $F(s, \cdot)$ takes as input an element of some domain X and outputs some value in the range Y, both of which implicitly depend on the parameter λ. For notational purposes, we also write $F_s(\cdot) = F(s, \cdot)$.*
For $k \geq 2$, we define a family of k-wise products $\mathcal{F}^k = (G^k, F^k)$ as follows:

- ***Key Generation:***
  $G^k(1^\lambda)$ independently generates $s_i \xleftarrow{\$} G(1^\lambda)$, for $i = 1, \ldots, k$.
- ***Evaluation:***
  To evaluate F^k on input $((s_1, \ldots, s_k), (x_1, \ldots, x_k))$, we define

$$F^k((s_1, \ldots, s_k), (x_1, \ldots, x_k)) = (F_{s_1}(x_1), \ldots, F_{s_k}(x_k)).$$

Definition 2 (Correlated Product Security). *Let $\mathcal{F} = (G, F)$ be a collection of efficiently computable functions. Let $C_k = C_k(1^\lambda)$ be a distribution. We say that $\mathcal{F}$ is secure under C_k-correlated products if $\mathcal{F}^k$ is one-way with respect to the input distribution C_k.*

We remark that if the function family is very small, e.g. if it consists of only a single function, then correlated product security can be trivially satisfied, since $s_1 = \cdots = s_k$ and hence $F_{s_1}(x) = \cdots = F_{s_k}(x)$. This degenerate case only arises when considering CP security for functions without trapdoor. Throughout this work, we will focus on *decisional* correlated product security (Definition 3). We note that a family with fewer than k functions can *never* be k-DCP secure. Similarly, functions with trapdoor must also belong to a large (super-polynomial size) family. Since all of our results deal with DCP security or DCP security with trapdoor, we do not find it necessary to amend the definition of CP security explicitly require the size of the function family to be large.

For the remainder of the paper, we will focus on the case where C_k is the uniform k-repetition distribution, i.e. k copies of a uniformly chosen input. We refer

the reader to the Appendix for reminders of the definitions of the Discrete Log and DDH assumptions, Deterministic Encryption, Lossy Trapdoor Functions, and Pseudorandom Functions.

3 Decisional Correlated Product Security

In this work we introduce the notion of Decisional Correlated Product (DCP) security, which can be viewed as the decisional variant of Correlated Product security introduced in [RS09]. In [RS09], Rosen and Segev focused on the case where C_k was the uniform k-repetition distribution, i.e. C_k uniformly samples x and outputs k copies of x. We will also focus on the k-repetition distribution, although we will consider a decisional variant of the problem.

First, we remark that Correlated Product security seems to be a much stronger notion than simply one-wayness. For example, the map $f_e : x \mapsto x^e \mod n$, is one-way trapdoor permutation under the RSA assumption. However, given $f_{e_1}(x), f_{e_2}(x)$, if $\gcd(e_1, e_2) = 1$, we can immediately recover x, by using the extended Euclidean algorithm to calculate s, t such that $se_1 + te_2 = 1$, and noticing that $(x^{e_1})^s (x^{e_2})^t = x$. This example also shows that Decisional Correlated Product security does not follow immediately from Computational Correlated Product security, because if d_1, d_2, d_3 are relatively prime, and $e_i = ed_i$ for some fixed e, then $f_{e_1}, f_{e_2}, f_{e_3}$ will be Computationally Correlated Product secure under the RSA assumption, but will not be Decisional Correlated Product secure by a similar argument.

Definition 3 (Decisional Correlated Product Security). *Let $\mathcal{F} = (G, F)$ be a collection of efficiently computable functions. We say that $\mathcal{F}^k$ is* k-wise Decisional Correlated Product secure *if for all efficient PPT adversaries A,*

$$\left|\Pr\left[A^{\mathsf{indepdist}} = 1\right] - \Pr\left[A^{\mathsf{repdist}} = 1\right]\right| < \nu$$

for some negligible function ν, and where the games indepdist *and* repdist *are defined as in Figure 1.*

Independent	**Repetition**
$s_1 \xleftarrow{\$} G(1^\lambda), \ldots, s_k \xleftarrow{\$} G(1^\lambda)$	$s_1 \xleftarrow{\$} G(1^\lambda), \ldots, s_k \xleftarrow{\$} G(1^\lambda)$
$x_1 \xleftarrow{\$} X, \ldots, x_k \xleftarrow{\$} X$	$x \xleftarrow{\$} X$
$b \xleftarrow{\$} A(s_1, \ldots, s_k, F_{s_1}(x_1), \ldots, F_{s_k}(x_k))$	$b \xleftarrow{\$} A(s_1, \ldots, s_k, F_{s_1}(x), \ldots, F_{s_k}(x))$
Return b	Return b

Fig. 1. Decisional Correlated Product Security

To illustrate the power of this definition, we construct a very natural IND-CPA secure encryption from any family of 2-DCP secure injective trapdoor functions. Let the public key be F_1, F_2, h where h is a pairwise independent hash function.

Define encryption as $E(m, r) = (F_1(r), h(F_2(r)) \oplus m)$. To decrypt, we simply invert F_1 to recover r, from this we can recover $h(F_2(r))$ and recover the message. If F_i have domain $\{0,1\}^\lambda$, and h maps from the range of F_i to $\{0,1\}^{\lambda/2}$, then the leftover hash lemma tells us that $(F_1(r_1), h(F_2(r_2)) \oplus m)$ is statistically close to $(F_1(r_1), h(F_2(r_2)))$. So if y_0, y_1 are chosen from the repetition-distribution $(y_0, h(y_1) \oplus m)$ is a valid ciphertext, while if (y_0, y_1) are chosen from the independent distribution $(y_0, h(y_1) \oplus m)$ is independent of m, thus this scheme will be IND-CPA secure. We emphasize that this is not one of our main results, but simply an illustration of a natural construction that follows from this definition.

Remark. One of the appealing properties of the notion of k-DCP security is that it abstracts one of the most important properties of the DDH assumption. To see the parallel, recall a simple DDH-based PRG. The description of the function is the group $\mathcal{G}$, and two elements g, g^a, and $f(b) = (g^b, (g^a)^b)$. The first element of the output will be uniform if b is uniform, and the pair is indistinguishable from uniform by the DDH assumption. Now, it is easy to see that this construction will go through as before with an injective k-DCP family of functions. In particular, the description of the PRG will be $\mathcal{F}, s_1, \ldots, s_k$, and $f(x) = F_{s_1}(x), \ldots, F_{s_k}(x)$. If $F_{s_i}(\cdot)$ is a permutation, f will be a PRG with no modification. If the $F_{s_i}(\cdot)$ are merely injective, we will have to apply an extractor to "smooth" the output, but the proof of security remains exactly the same as in the DDH case. In fact, this observation can be generalized, the full version of this work contains a more detailed discussion of the parallel between DCP security and the DDH assumption.

The notion of *Decisional* Correlated Product security is clearly a stronger notion than the (Computational) Correlated Product security defined in [RS09] for *injective functions*. In the next section, we examine under what conditions DCP security implies CP security.

4 Relations to (Computational) Correlated Product Security

The notion of k-DCP security seems like a stronger requirement than Computational Correlated Product security, but we observe that if we do not put any requirements on the functions, then k-DCP security may be satisfied by trivial functions. For example the constant functions are trivially k-DCP for any $k \geq 2$. The following lemmas give sufficient conditions for when a k-DCP secure family is k-correlated product secure.

Lemma 2. *If $\mathcal{F} = (G, F)$ is a family of k-DCP secure functions with super-polynomial size domain and are injective, then $\mathcal{F}$ is k-correlated product secure.*

Proof. Let A be an efficient adversary that given $s_1, \ldots, s_k$, and $(F_{s_1}(x), \ldots, F_{s_k}(x))$, finds the inverse $(x'_1, \ldots, x'_k) = (x, x, \ldots, x)$ with non-negligible probability ϵ, we exhibit an efficient distinguisher D that uses A to break the k-DCP security of $\mathcal{F}$.

Algorithm 1. $D(s_1, \ldots, s_k, y_1, \ldots, y_k)$

$(x'_1, \ldots, x'_k) \xleftarrow{\$} A(s_1, \ldots, s_k, y_1, \ldots, y_k)$
if $x'_1 = x'_2 = \cdots = x'_k$ and $F_{s_i}(x'_i) = y_i$ for $i \in [k]$ **then**
  return 1
else
  return 0
end if

We must analyze the probability that D outputs 1 in the repdist and indepdist games.

$$\begin{aligned}\Pr[D^{\mathsf{repdist}} = 1] &= \Pr[x'_1 = \cdots = x'_k \wedge F_{s_i}(x'_i) = y_i | \\ &\qquad x \xleftarrow{\$} X, s_i \xleftarrow{\$} G(1^\lambda), y_i = F_{s_i}(x), \{x'_i\}_{i=1}^k \xleftarrow{\$} A(\{s_i\}_{i=1}^k, \{y_i\}_{i=1}^k)] \\ &= \Pr[A \text{ successfully inverts}] = \epsilon.\end{aligned}$$

$$\begin{aligned}\Pr[D^{\mathsf{indepdist}} = 1] &= \Pr[x'_1 = \cdots = x'_k \wedge F_{s_i}(x'_i) = y_i | \\ &\qquad x_i \xleftarrow{\$} X, s_i \xleftarrow{\$} G(1^\lambda), y_i = F_{s_i}(x_i), \{x'_i\}_{i=1}^k \xleftarrow{\$} A(\{s_i\}_{i=1}^k, \{y_i\}_{i=1}^k)] \\ &= \Pr[x'_1 = \cdots = x'_k \wedge x'_i = x_i | \\ &\qquad x_i \xleftarrow{\$} X, s_i \xleftarrow{\$} G(1^\lambda), y_i = F_{s_i}(x_i), \{x'_i\}_{i=1}^k \xleftarrow{\$} A(\{s_i\}_{i=1}^k, \{y_i\}_{i=1}^k)] \\ &\leq \Pr[x_1 = x_2 | x_i \xleftarrow{\$} X] \leq \frac{1}{|X|}.\end{aligned}$$

Thus the difference $|\Pr[D^{\mathsf{repdist}} = 1] - \Pr[D^{\mathsf{indepdist}} = 1]| \geq \epsilon - \frac{1}{|X|}$ is non-negligible, as $|X|$ is super-polynomial.

Next, we show that if a family $\mathcal{F} = (G, F)$ is a DCP secure, and each function is *individually* one-way, then the family is also Correlated Product secure.

Lemma 3. *If $\mathcal{F} = (G, F)$ is a family of k-DCP secure one-way functions, then $\mathcal{F}$ is k-correlated product secure.*

Proof. Suppose on the contrary that they were not. Let A be a PPT algorithm that breaks the correlated product security of (G, F), in particular given $\{s_1, \ldots, s_k, F_{s_1}(x_1), \ldots, F_{s_k}(x_1)\}$ A is able to find a pre-image $(x'_1, \ldots, x'_k)$ with some non-negligible probability ϵ, where the s_i are generated by G at random, and x_1 is chosen uniformly at random. We use A to build a PPT distinguisher D that can win in the k-DCP game.

We analyze the probability that D outputs 1. If indeed the inputs are correlated, i.e. $y_i = F_{s_i}(x_1)$, then A succeeds with probability ϵ and so D will output 1 with that probability.

On the other hand, if the inputs are random and independent, i.e. $y_i = F_{s_i}(x_i)$, then $(x_1, \ldots, x_k)$ is a uniformly chosen input from the product space. Because

Algorithm 2. $D(s_1, \ldots, s_k, y_1, \ldots, y_k)$

$(x'_1, \ldots, x'_k) \xleftarrow{\$} A(s_1, \ldots, s_k, y_1, \ldots, y_k)$
if $F_{s_i}(x'_i) = y_i$ for $i \in [k]$ **then**
  return 1
else
  return 0
end if

each $F_{s_i}(\cdot)$ is a one-way function, the product function $(F_{s_1}(\cdot), \ldots, F_{s_k}(\cdot))$ is also one-way. Since the inputs are uncorrelated, the probability that A inverts it on a random value is negligible. Thus, in this case, D outputs 1 with only negligible probability.

This contradicts the k-DCP security of (G, F).

Many of the results in this work will focus on the case where the family $\mathcal{F}$ are in fact injective, or injective with trapdoor, and so the Correlated Product security will follow immediately from the DCP security of $\mathcal{F}$.

5 Equivalence of OWF and (Decisional) Correlated Product Secure Families of OWFs

In this section, we aim to prove the main theorem relating the existence of OWFs to that of (Decisional) Correlated Product secure OWF families.

Theorem 1. *The following statements are equivalent:*

1. *One-way functions exist.*
2. *k-DCP secure families of one-way functions exist.*
3. *k-CP secure families of one-way functions exist.*

To do this, we first show how to construct a DCP secure family of one-way functions from any pseudorandom function family. The idea is that a PRF family becomes DCP secure if we swap what we call the seed, and what we call the input. This idea has also been used in the past by Luby and Rackoff [LR89] to show the one-wayness of the UNIX-like password hash. If the PRF output is sufficiently long, then the resulting functions are also one-way, thus we have a family of DCP secure one-way functions. The exact lengths necessary are given in Lemma 5.

We then show that DCP secure one-way function families are also (ordinary) CP secure. This will follow directly from the fact that a product of one-way functions remain one-way under uniform independent inputs (Lemma 3). Finally, CP secure OWF families obviously are one-way, which completes the cycle of implications.

Let $(\mathsf{PRFGen}, \mathsf{PRF})$ be a PRF family, such that if $s \xleftarrow{\$} \mathsf{PRFGen}(1^\lambda)$, with $s \in \{0,1\}^{w(\lambda)}$ then the domain of

$$\mathsf{PRF}(s, \cdot) : \{0,1\}^{n(\lambda)} \to \{0,1\}^{\ell(\lambda)}.$$

We can define a DCP family (G, F), by

- **Sampling:** $G(1^\lambda)$ outputs a uniform value $s \in \{0,1\}^{n(\lambda)}$.
- **Evaluation:** For any $s \in \{0,1\}^{n(\lambda)}$,

$$F_s(\cdot) : \{0,1\}^{w(\lambda)} \to \{0,1\}^{\ell(\lambda)}$$
$$x \mapsto \mathsf{PRF}(x, s).$$

Lemma 4. *(G, F) forms a k-Decisional Correlated Product secure function family for any $k = poly(\lambda)$.*

Proof. Define the distributions Λ_0, Λ_1 by sampling $s_1, \ldots, s_k \xleftarrow{\$} G(1^\lambda)$, and $x_1, \ldots, x_k \xleftarrow{\$} \{0,1\}^{w(\lambda)}$

$$\Lambda_0 = \{s_1, \ldots, s_k, F_{s_1}(x_1), \ldots, F_{s_k}(x_1)\}$$
$$\Lambda_1 = \{s_1, \ldots, s_k, F_{s_1}(x_1), F_{s_2}(x_2), \ldots, F_{s_k}(x_k)\}$$

Thus we must show that any adversary who can distinguish Λ_0 from Λ_1 can distinguish the underlying Pseudorandom Function from a truly random function.

Now, by the definition of F, we have

$$\Lambda_0 = \{s_1, \ldots, s_k, F_{s_1}(x_1), \ldots, F_{s_k}(x_1)\} = \{s_1, \ldots, s_k, \mathsf{PRF}(x_1, s_1), \ldots, \mathsf{PRF}(x_1, s_k)\},$$
$$\Lambda_1 = \{s_1, \ldots, s_k, F_{s_1}(x_1), \ldots, F_{s_k}(x_k)\} = \{s_1, \ldots, s_k, \mathsf{PRF}(x_1, s_1), \ldots, \mathsf{PRF}(x_k, s_k)\}.$$

Now, it is clear that the security of the Pseudorandom Function gives

$$\Lambda_0 \approx_c \{s_1, \ldots, s_k, U_{\ell(\lambda)}, \ldots, U_{\ell(\lambda)}\} \approx_c \Lambda_1,$$

which gives the result.

Lemma 5. *If the size of the key space of F is a negligible fraction of the size of the output space, i.e. $1/2^{\ell(\lambda)-w(\lambda)}$ is negligible in λ, then (G, F) forms a family of one-way functions.*

Proof. Suppose to the contrary that for some key s, the function $F_s(\cdot)$ was not one-way. Let A be a PPT inverter that succeeds with non-negligible probability ϵ, i.e.

$$\Pr_x[F_s(z) = F_s(x) | z \leftarrow A(F_s(x))] = \epsilon$$

We use A to construct a PPT algorithm B that distinguishes between oracle access to PRF (with a randomly chosen seed x) and a truly random function $\mathcal{RO}$. The algorithm queries s on the oracle, and receives y, which is either $y = \mathsf{PRF}(x, s) = F_s(x)$ for some x, or a truly random value. The distinguisher B runs A on y, and receives some output x'. If it is the case that $F_s(x') = y$, then B outputs 1, otherwise B outputs 0.

We analyze the probabilities $\Pr[B^{\mathcal{RO}(\cdot)} = 1]$ and $\Pr_x[B^{\mathsf{PRF}(x,\cdot)} = 1]$. In the former case, the probability that a random value is in the range of $\mathsf{PRF}(s,\cdot)$ is $\frac{|Range|}{2^\ell} \leq \frac{2^w}{2^\ell}$ which we assumed to be negligible. On the other hand,

$$\begin{aligned}\Pr_x[B^{\mathsf{PRF}(x,\cdot)} = 1] &= \Pr_x[\mathsf{PRF}(z,s) = y | z \leftarrow A(y)] \\ &= \Pr_x[\mathsf{PRF}(z,s) = \mathsf{PRF}(x,s) | z \leftarrow A(\mathsf{PRF}(x,s))] \\ &= \Pr_x[F_s(z) = F_s(x) | z \leftarrow A(F_s(x))] = \epsilon\end{aligned}$$

This contradicts the pseudorandomness of PRF.

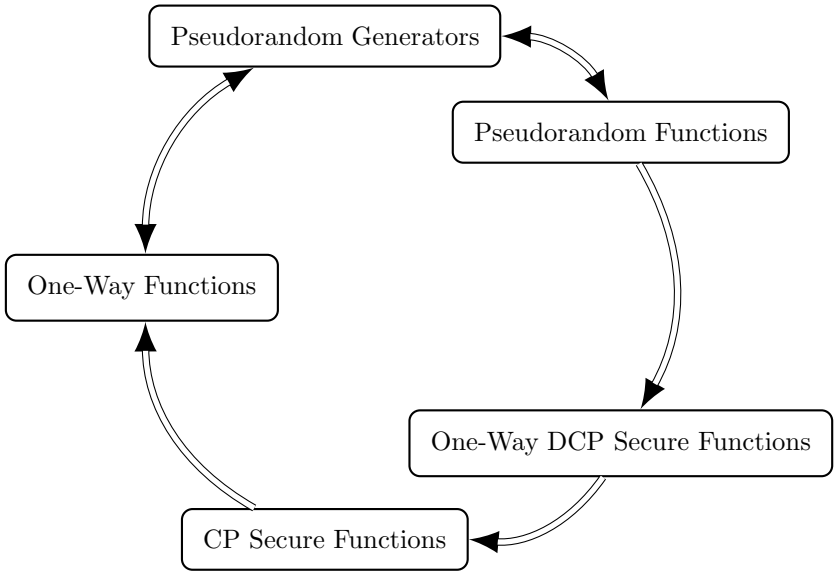
Corollary 1. *One-way functions imply k-DCP secure one-way function families.*

Proof. In Hastad, Impagliazzo, Levin and Luby [HILL99] it was shown that one-way functions imply PRGs, and in Goldreich, Goldwasser, Micali [GGM86] it was shown that PRGs imply the existence of PRF families with sufficiently long output, thus combining these results with our result, we have one-way functions imply k-DCP secure one-way functions.

Corollary 2. *One-way functions imply k-CP secure function families.*

Proof. This follows immediately from applying Lemma 3 to Corollary 1.

Since every Correlated Product secure function family is trivially a one-way function family, we have



Since pseudo-random synthesizers [NR95, Rei98] are equivalent to one-way functions, we also achieve an equivalence between DCP secure functions and synthesizers. In the full version of this work, we give a direct proof that every family of pseudo-random synthesizers is immediately DCP secure.

In [BHK11], Braverman, Hassidim and Kalai introduced the notion of leakage-resilient random-input PRFs. A leakage-resilient random-input PRF is a pseudo-random function which remains pseudo-random when queried on *random* inputs (i.e. it is a weak PRF) even when partial information about the seed is leaked. Applying our construction to a leakage-resilient random-input PRF, we obtain a family of functions which is decisionally correlated product secure for any distribution $(X_1, \ldots, X_n)$ where that satisfies $\tilde{H}_\infty(X_i|X_1, \ldots, X_{i-1}) > \lambda$. Notice that the repetition distribution does not have this property, so by applying our construction to leakage-resilient random-input PRFs, we achieve DCP security for a completely different class of distributions.

6 DCP with Trapdoor from Lossy Trapdoor Functions

In the preceding sections, we examined DCP secure functions without trapdoors, and showed that one-way DCP secure functions *without trapdoor* could be constructed from any one-way function. Now, we show constructions of DCP with trapdoor. In particular, in this section, we show that lossy trapdoor functions with sufficient lossiness imply DCP secure injective trapdoor functions.

Theorem 2. *Let $\mathcal{H}$ be a family of invertible*[2] *pairwise independent hash functions with $h : \{0,1\}^\lambda \to \{0,1\}^\lambda$. Let $\epsilon(\lambda)$ be any function such that $1/2^{\epsilon(\lambda)}$ is negligible in λ. Let $\mathcal{F} = (G, F)$ be a family of LTDFs on domain $\{0,1\}^\lambda$, where the lossy mode has residual leakage $r \leq \frac{\lambda+2-2\log(1/\epsilon)}{k}$, for some integer k. Define $\hat{\mathcal{F}} = (\hat{G}, \hat{F})$ by*

- *$\hat{G}(1^\lambda)$, samples $s \xleftarrow{\$} G(1^\lambda)$, and $h \xleftarrow{\$} \mathcal{H}$, and outputs the function index h, s.*
- *Given a function index (h, s) and an input x, $\hat{F}_{h,s}(x) = F_s(h(x))$.*

Then $\hat{\mathcal{F}}$ is a k-DCP secure injective trapdoor function.

Proof. To prove the claim, we must show that distributions

$$\{h_1, s_1, \ldots, h_k, s_k, \hat{F}_{h_1,s_1}(x_1), \ldots, \hat{F}_{h_k,s_k}(x_1)\}$$
and
$$\{h_1, s_1, \ldots, h_k, s_k, \hat{F}_{h_1,s_1}(x_1), \hat{F}_{h_k,s_k}(x_k)\}$$

are computationally indistinguishable, where $h_1, s_1, \ldots, h_k, s_k \xleftarrow{\$} \hat{G}(1^\lambda)$, and $x_1, \ldots, x_k$ are sampled uniformly at random from the domain $\{0,1\}^\lambda$.

[2] We remark that this is not a strong restriction, and the natural construction $h(x) = ax + b$ over a finite field yields a collection of invertible pairwise independent hash functions.

If the function $F_s(\cdot)$ is in lossy mode, it has image size at most $2^{\frac{\lambda+2-2\log(1/\epsilon)}{k}}$, so if x is chosen uniformly from $\{0,1\}^\lambda$, then

$$\begin{aligned}\tilde{H}_\infty(x|\hat{F}_{h_1,s_1}(x),\ldots,\hat{F}_{h_{k-1},s_{k-1}}(x)) &\geq \lambda-(k-1)\frac{\lambda+2-2\log(1/\epsilon)}{k}\\ &= \lambda-(\lambda+2-2\log(1/\epsilon))+\frac{\lambda+2-2\log(1/\epsilon)}{k}\\ &= \frac{\lambda+2-2\log(1/\epsilon)}{k}+2\log(1/\epsilon)-2.\end{aligned}$$

By the Crooked Leftover Hash Lemma, we have

$$\begin{aligned}\Delta\Big(&\big\{h_1,s_1,\ldots,h_k,s_k,F_{s_1}(h_1(x_1)),\ldots,F_{s_k}(h_k(x))\big\},\\ &\big\{h_1,s_1,\ldots,h_k,s_k,F_{s_1}(h_1(U_\lambda)),F_{s_2}(h_2(x)),\ldots,F_{s_k}(h_k(x))\big\}\Big)<\epsilon.\end{aligned}$$

Repeating this argument a total of k times, we have

$$\begin{aligned}\Delta\Big(&\big\{h_1,s_1,\ldots,h_k,s_k,F_{s_1}(h_1(x)),\ldots,F_{s_k}(h_k(x))\big\},\\ &\big\{h_1,s_1,\ldots,h_k,s_k,F_{s_1}(h_1(U_\lambda)),\ldots,F_{s_k}(h_k(U_\lambda))\big\}\Big)<k\epsilon.\end{aligned}$$

Since ϵ was assumed to be negligible, so is $k\epsilon$. Thus when the s_i are chosen to be lossy keys, the two distributions $\{h_1,s_1,\ldots,h_k,s_k,\hat{F}_{h_1,s_1}(x_1),\ldots,\hat{F}_{h_k,s_k}(x_1)\}$ and $\{h_1,s_1,\ldots,h_k,s_k,\hat{F}_{h_1,s_1}(x_1),\hat{F}_{h_k,s_k}(x_k)\}$ are statistically indistinguishable. The computational indistinguishability of lossy and injective keys implies that when the s_i are injective keys, the two distributions are computationally indistinguishable. Thus $(\hat{G},\hat{F})$ forms a family of k-DCP secure trapdoor functions.

7 Decisional Correlated Product Security Is Deterministic Encryption

In this section, we examine the consequences of DCP secure functions, again *with trapdoor*. We show that any 2-DCP secure functions with trapdoor are – almost without modification – a PRIV1 secure uniform deterministic encryption. The notion of PRIV1 security is the original definition of security for deterministic encryption put forward in [BBO07]. PRIV1 security is the natural relaxation of the notion of semantically secure encryption to the deterministic setting. Recall that a cryptosystem is semantically secure if for any function $f(\cdot)$, an adversary's probability of calculating $f(m)$ remains essentially unchanged if the adversary is given access to an encryption $E(m)$. PRIV1 security requires that for any function $f(\cdot)$ *which is independent of the public key*, an adversary's ability to calculate $f(m)$ remains essentially unchanged whether he has access to the public key, or the public key *and* an encryption $E(m)$. See [BBO07] for the formal definition of PRIV1 security.

We follow the terminology of [BFOR08], where a uniform deterministic encryption is one which is only guaranteed to be secure against message adversaries that choose messages from the uniform distribution, instead of simply any high min-entropy distribution.

Let $\mathcal{F} = (G, F)$ be a family of 2-Decisional Correlated Product secure Functions.

We can define a (Uniform) Deterministic Encryption by

KeyGen:	**Encryption:**	**Decryption:**
$(s,t) \xleftarrow{\$} G(1^\lambda)$	$E(pk, m) = F_{pk}(m)$	$D(sk, c) = F_t^{-1}(c)$
$pk = s$, $sk = t$		

Fig. 2. Decisional Correlated Product Secure functions with trapdoor are PRIV1 secure

Theorem 3. *The scheme outlined in Figure 2 is BB-CSS secure.*

Proof. First, we recall the notion of BB-CSS (Balanced Boolean Comparison-based Semantic Security) as defined in [BFOR08]. This is similar to the Comparison Semantic Security PRIV1, outlined by the games privreal and prividеal, except that the side information t is required to be a balanced boolean function, i.e. $\Pr[t = 0] \approx \Pr[t = 1] \approx \frac{1}{2}$.

For simplicity, we assume that $\Pr[t = 0] = \Pr[t = 1] = \frac{1}{2}$, but it is easy to see that if the distributions are only negligibly close to $\frac{1}{2}$ then the argument goes through as well.

Notice that in this setting *any* adversary has a $\frac{1}{2}$ chance of winning in the prividеal game since his view is independent of the actual side information, thus it is enough to consider the adversary's probability of winning in the privreal game.

Now, suppose there exists an adversary $A = (A_m, A_g)$, such that $(m, t) \xleftarrow{\$} A_m(1^\lambda)$, where m is uniform on X the domain of f_s, and t is uniform on $\{0, 1\}$. The guessing adversary A_g on input pk, c outputs a guess t'. If $c = E(Pk, m)$, then $\Pr[t = t'] = \frac{1}{2} + \epsilon$.

We show how to use A to create a distinguisher D that can distinguish the 2-repetition distribution from the 2-independent distribution. The algorithm D takes as input the description of two functions s_0, s_1, and two outputs y_0, y_1, which come from either the repetition distribution (in which case $y_i = F_{s_i}(x)$) or the independent distribution (in which case $y_i = F_{s_i}(x_i)$, for two independently sampled x_i). The distinguisher D is described by Algorithm 3.

Now, we must analyze the probability that D succeeds. If y_0, y_1 were generated from the repetition distribution, then since A_g succeeds with probability $\frac{1}{2} + \epsilon$, the probability that D guesses "repetition" is $(\frac{1}{2} + \epsilon)^2 + (\frac{1}{2} - \epsilon)^2 = \frac{1}{2} + 2\epsilon^2$. If y_0, y_1 were generated from the independent distribution, because the side information is a balanced boolean function, the probability that the t_0, t_1 that would have been generated by A_m are equal is $\frac{1}{2}$. Intuitively, this should mean

Algorithm 3. $D(s_0, s_1, y_0, y_1)$

$t'_0 \xleftarrow{\$} A_g(s_0, y_0)$
$t'_1 \xleftarrow{\$} A_g(s_1, y_1)$
if $t'_0 = t'_1$ **then**
  return Repetition
else
  return Independent
end if

the probability that D correctly guesses "independent" is just $\frac{1}{2}$. This is in fact the case, because

$$\begin{aligned}
&\Pr[D \text{ correctly guesses independent}] \\
&\quad = \frac{1}{2}\Pr[D \text{ guesses independent} | t_0 = t_1] + \frac{1}{2}\Pr[D \text{ guesses independent} | t_0 \neq t_1] \\
&\quad = \frac{1}{2}\left(2\left(\frac{1}{2}+\epsilon\right)\left(\frac{1}{2}-\epsilon\right)\right) + \frac{1}{2}\left(\left(\frac{1}{2}+\epsilon\right)^2 + \left(\frac{1}{2}-\epsilon\right)^2\right) = \frac{1}{2}.
\end{aligned}$$

Thus the probability that D is correct is $\frac{1}{2} + \epsilon^2$.

Corollary 3. *The scheme outlined above is PRIV1 secure.*

Proof. In [BFOR08], they show that BB-CSS security (Comparison based Semantic Security against Balanced Boolean side information) implies B-CSS security (Comparison based Semantic Security against any Boolean side information), which in turns implies A-CSS which is security against Arbitrary side information. A-CSS security is the terminology in [BFOR08] for PRIV1 security. The only thing to do is to notice that both proofs in [BFOR08] go through unchanged when the adversaries are restricted to be uniform adversaries.

Remark. We note that if the function family $\mathcal{F} = (G, F)$ were assumed to be Decisional Correlated Product (DCP) secure when the inputs were chosen not uniformly, but simply from some high min-entropy distribution, the same proof would go through to show PRIV1 security against any (not necessarily uniform) adversary A_m.

Remark. On the other hand, there is an example (outlined below) of a PRIV1 secure uniform DE scheme that is not n-DCP secure (treating the public key as the seed, key generation as G, and encryption as F), where n is the size of the message. This does not preclude the construction of a DCP secure family from such a DE scheme, but instead shows that these two notions are not *definitionally* equivalent. To see that a PRIV1 secure DE need not be n-DCP secure, take any IND-CPA secure (randomized) encryption scheme, and transform it into a "leaky" scheme that leaks the first bit of randomness used in encryption by simply taking an extra dummy bit of randomness and revealing it in the ciphertext. The construction of uniform DE from one-way trapdoor permutations given in [BFOR08] makes use of an IND-CPA secure (randomized) encryption scheme.

Without fully reproducing the [BFOR08] construction, we only need to point out that the first bit of randomness is the hard-core predicate defined by the dot product of the message and a vector from the public key. If the "leaky" encryption of the same message under n different public keys is revealed, the message can be reconstructed using linear algebra. This immediately breaks (Decisional) Correlated Product security.

8 Conclusion and Open Problems

In this work we suggested a new primitive, the decisional variant of Correlated Product (DCP) secure functions. We argue that this primitive has many appealing properties. To this end, we show a parallel between Correlated Product security and DCP and the Discrete Log Problem and its decisional variant DDH. We also show how to construct simple primitives from DCP such as PRGs and IND-CPA secure encryption.

Our main results examine two main cases: DCP functions with trapdoor and without trapdoor. We show that DCP secure functions (and CP secure functions) without trapdoor are equivalent to one-way functions. This is a somewhat surprising result since notions of correlated product security appear to be much stronger than simple one-wayness. When examining DCP secure functions with trapdoor, we show that they are implied by Lossy Trapdoor Functions, and that DCP secure functions are immediately a Deterministic Encryption scheme.

An interesting line of future research would be to develop further constructions of DCP secure functions with trapdoor. A second line of research would be a closer examination of the connections between DCP security and deterministic encryption. For example, we know that DCP secure functions are deterministic encryption, but it would be interesting to see how the security is affected by auxiliary information, e.g. along the lines of [BS11].

References

[ABBC10] Acar, T., Belenkiy, M., Bellare, M., Cash, D.: Cryptographic Agility and Its Relation to Circular Encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 403–422. Springer, Heidelberg (2010)

[AHI11] Applebaum, B., Harnik, D., Ishai, Y.: Semantic security under related-key attacks and applications. In: ITCS 2011 (2011)

[BBO07] Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and Efficiently Searchable Encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 535–552. Springer, Heidelberg (2007)

[BFO08] Boldyreva, A., Fehr, S., O'Neill, A.: On Notions of Security for Deterministic Encryption, and Efficient Constructions without Random Oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 335–359. Springer, Heidelberg (2008)

[BFOR08] Bellare, M., Fischlin, M., O'Neill, A., Ristenpart, T.: Deterministic Encryption: Definitional Equivalences and Constructions without Random Oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 360–378. Springer, Heidelberg (2008)

[BHK11] Braverman, M., Hassidim, A., Kalai, Y.T.: Leaky Pseudo-entropy functions. In: ITCS 2011 (2011)

[BS11] Brakerski, Z., Segev, G.: Better Security for Deterministic Public-Key Encryption: The Auxiliary-Input Setting. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 543–560. Springer, Heidelberg (2011)

[FGK+10] Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More Constructions of Lossy and Correlation-Secure Trapdoor Functions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 279–295. Springer, Heidelberg (2010)

[GGM86] Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. Journal of the ACM 33(4), 792–807 (1986)

[GOR11] Goyal, V., O'Neill, A., Rao, V.: Correlated-Input Secure Hash Functions. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 182–200. Springer, Heidelberg (2011)

[HILL99] Hastad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM J. Comput. 28(4), 1364–1396 (1999)

[IKNP03] Ishai, Y., Kilian, J., Nissim, K., Petrank, E.: Extending Oblivious Transfers Efficiently. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 145–161. Springer, Heidelberg (2003)

[ILL89] Impagliazzo, R., Levin, L.A., Luby, M.: Pseudo-random generation from one-way functions (extended abstract). In: STOC 1989, pp. 12–24 (1989)

[LR89] Luby, M., Rackoff, C.: A study of password security. Journal of Cryptology 1(3), 151–158 (1989)

[NR95] Naor, M., Reingold, O.: Synthesizers and their application to the parallel construction of psuedo-random functions. In: FOCS 1995, pp. 170–181 (1995)

[Pei09] Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: STOC 2009: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, pp. 333–342. ACM, New York (2009)

[PW08] Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: STOC 2008: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, pp. 187–196. ACM, New York (2008)

[Rei98] Reingold, O.: Pseudo-Random Synthesizers Functions and Permutations. PhD thesis, The Weizmann Institute of Science (1998)

[RS08] Rosen, A., Segev, G.: Efficient lossy trapdoor functions based on the composite residuosity assumption (2008), http://eprint.iacr.org/2008/134

[RS09] Rosen, A., Segev, G.: Chosen-Ciphertext Security via Correlated Products. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 419–436. Springer, Heidelberg (2009)

[Vah10] Vahlis, Y.: Two Is a Crowd? A Black-Box Separation of One-Wayness and Security under Correlated Inputs. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 165–182. Springer, Heidelberg (2010)

Relations between Constrained and Bounded Chosen Ciphertext Security for Key Encapsulation Mechanisms

Takahiro Matsuda[1,⋆], Goichiro Hanaoka[1], and Kanta Matsuura[2]

[1] National Institute of Advanced Industrial Science and Technology, Japan
{t-matsuda,hanaoka-goichiro}@aist.go.jp
[2] The University of Tokyo, Japan
kanta@iis.u-tokyo.ac.jp

Abstract. In CRYPTO 2007, Hofheinz and Kiltz formalized a security notion for key encapsulation mechanisms (KEMs), called *constrained chosen ciphertext* (CCCA) security, which is strictly weaker than ordinary chosen ciphertext (CCA) security, and showed a new composition paradigm for CCA secure hybrid encryption. Thus, CCCA security of a KEM turned out to be quite useful. However, since the notion is relatively new and its definition is slightly complicated, relations among CCCA security and other security notions have not been clarified well. In this paper, in order to better understand CCCA security and the construction of CCCA secure KEMs, we study relations between CCCA and *bounded CCA* security, where the latter notion considers security against adversaries that make a-priori bounded number of decapsulation queries, and is also strictly weaker than CCA security. Specifically, we show that in most cases there are separations between these notions, while there is some unexpected implication from (a slightly stronger version of) CCCA security to a weak form of 1-bounded CCA security. We also revisit the construction of a KEM from a hash proof system (HPS) with computational security properties, and show that the HPS-based KEM, which was previously shown CCCA secure, is actually 1-bounded CCA secure as well. This result, together with the above general implication, suggests that 1-bounded CCA security can be essentially seen as a "necessary" condition for a CCCA secure KEM.

Keywords: key encapsulation mechanism, constrained CCA security, bounded CCA security, hash proof system.

1 Introduction

Background and Motivation. Studies on constructing and understanding practical public key encryption (PKE) schemes secure against chosen ciphertext attacks (CCA) [26,11] are important research topics in the area of cryptography.

⋆ Takahiro Matsuda is supported by JSPS Research Fellowships for Young Scientists.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 576–594, 2012.

Among several approaches towards CCA secure PKE schemes, one of the promising approaches is to construct a PKE scheme via the hybrid encryption methodologies using a key encapsulation mechanism (KEM) which encapsulates (i.e. encrypts) a random session-key, and a data encapsulation mechanism (DEM) which encrypts an actual message using the session-key. Cramer and Shoup [10] show that if we combine a CCA secure KEM and a CCA secure DEM, we obtain a hybrid PKE scheme which is CCA secure. Abe et al. [1] show yet another hybrid encryption paradigm from a Tag-KEM, which is an extension of a KEM, and a passively secure DEM.

In CRYPTO 2007, Hofheinz and Kiltz [17] formalized a security notion for KEMs called *constrained chosen ciphertext* (CCCA) security, which is strictly weaker than ordinary CCA security. Then, they show that a CCA secure PKE scheme can be constructed by combining a CCCA secure KEM and a DEM satisfying the security of (one-time) authenticated encryption [4]. Therefore, CCCA security turned out to be a quite useful security notion for constructing a CCA secure PKE scheme.

However, the notion of CCCA security is relatively new, and the definition of CCCA security is slightly technically complicated compared to other existing security notions for KEMs, such as (ordinary) CCA security. Therefore, the relations between CCCA security and other security notions have not been studied and clarified well. Especially, "how" CCCA security is weak, compared to ordinary CCA security, seems not to have been understood well previously. It is naturally expected that the better we understand CCCA security itself, the higher the possibility we will come up with practical CCCA secure KEMs becomes, which will also lead to practical CCA secure PKE schemes.

So far, there are several positive and negative results regarding how close CCCA security and CCA security for KEMs are: Baek et al. [2] show that a CCCA secure KEM can be generically converted into a CCA secure one by using a one-time secure message authentication code. Hanaoka and Kurosawa [13] show that in fact, a CCCA secure KEM can be turned into a CCA secure one even without using any other additional building block, by using a part of the session-key (for a DEM) to check the consistency of a ciphertext in the decapsulation process. These results make us believe that CCCA security and CCA security for KEMs are in fact very close.

On the other hand, Choi et al. [7] show that the well-known KEM by Kurosawa and Desmedt [21], which was shown to be CCCA secure under the decisional Diffie-Hellman (DDH) assumption in [17], is not `OW-2-CCA` secure. That is, the session-key hidden in a ciphertext of the Kurosawa-Desmedt KEM can be recovered if an adversary can submit two decapsulation queries of its choice. This result, in contrast to the above positive results [2,13], makes us think that CCCA security is far from CCA security.

These previous results may illustrate that it is difficult to grasp what is actually achieved by CCCA security and what is not. The motivation of this work is to clarify the relations between CCCA security and other security notions, so that it leads to better understanding of CCCA security itself and also leads to

insights for constructing practical CCCA secure KEMs in the future. For that purpose, we study relations between CCCA security and *bounded CCA security* [8], which only captures security against adversaries that make a-priori bounded number of decapsulation queries (denoted by "q-CCA" for q queries), and is also strictly weaker than CCA security in a different sense from CCCA security.

It is known that we can construct a "q-bounded CCA" secure KEM whose ciphertext consists of only one group element (and thus "optimal" ciphertext size as a KEM) under the DDH assumption [8], for any predetermined polynomial q. On the other hand, the best known CCCA secure KEMs under the DDH assumption (or weaker assumptions) [21,17,12] have at least two group elements in a ciphertext. If we can construct a CCCA secure KEM under the DDH assumption with just one group element ciphertext, it will lead to (by combining it with a DEM satisfying the security of authenticated encryption) the best DDH-based PKE scheme in terms of the ciphertext overhead[1], i.e. one group element plus the ciphertext overhead caused by the DEM, which can be as small as k-bit for k-bit security. We believe that studying relations between CCCA security and bounded CCA security will also lead to important insights for the possibility of such "space-efficient" CCA secure PKE schemes (under DDH and weaker assumptions). Especially, understanding "how hard" it is to construct a CCCA secure KEM compared to a KEM with bounded CCA security will benefit the future designers of CCCA secure KEMs.

Our Contribution. Firstly, in Section 3 we investigate relations between CCCA security and bounded CCA security, i.e. implications/separations between these two security notions. One might expect that there is always a separation of CCCA security from bounded CCA security, and vice versa. As expected, we show that in most cases we have separations in both directions, and thus our contribution regarding this result is to give formal proofs, together with some basic ideas, for the separations. In particular, we show that IND-CCCA security does not imply OW-2-CCA or IND-1-CCA security (here, OW and IND stand for "one-wayness" and "indistinguishability", respectively, and the formal definitions for security notions are given in Section 2). Perhaps somewhat surprisingly, however, it turns out that there is an implication from CCCA security to the weakest form of bounded CCA security, namely, OW-1-CCA, *if we slightly change the definition of a valid CCCA adversary.* The change we make to show the implication is regarding the definition of *uncertainty* that plays an important role in defining CCCA security, but is quite subtle. The proof for this result involves some unusual treatment (at least in the context of security proofs of CCCA/bounded CCA security) of an adversary, and might be of independent interest. For more details, see Section 3.2. We also show the separation of the opposite direction (bounded CCA security does not imply CCCA security) in terms of the number of queries allowed for an adversary. Specifically, we show that for any polynomial $q \geq 0$, IND-q-CCA security does not imply OW-$(q+1)$-CCCA security.

[1] Ciphertext overhead is the difference between the size of a ciphertext and the size of its plaintext.

Then, in Section 4 we revisit the construction of KEMs from a *hash proof system* (HPS) [9,17], and show that the HPS-based KEM, which was shown to be IND-CCCA secure [17] (under some computational security requirements), actually satisfies IND-1-CCA security under the same assumptions used to show its IND-CCCA security. This result should be contrasted with the above mentioned separation of IND-1-CCA from IND-CCCA. Given the hybrid encryption paradigm by Cramer and Shoup [10], the result here implies that if we combine a HPS-based KEM (e.g. the Kurosawa-Desmedt KEM [21]) with a CCA secure redundancy-free DEM (e.g. a strong pseudorandom permutation [25]), the resulting PKE scheme still provides IND-1-CCA security. (As mentioned above, OW-2-CCA attack on this KEM is possible, and thus this is the optimal security result for the Kurosawa-Desmedt KEM, in terms of bounded CCA security.) Given the fact that using computational HPS is one of the major methodologies for constructing a practical CCCA secure KEM, we see that IND-1-CCA security can essentially be viewed as a "necessary" condition for a CCCA secure KEM. This result on the HPS-based KEM, together with the above general implication to OW-1-CCA security, also suggests that constructing a CCCA secure KEM is harder than constructing a 1-bounded CCA secure one. To the best of our knowledge, such insights have not been known previously.

Although it might be hard to imagine that 1-bounded CCA security (i.e. OW-1-CCA security and IND-1-CCA security) plays a practical role in real world applications in which KEMs (and PKE schemes) are used,[2] we stress that our aim in this paper is not to emphasize the importance of such security in practice, but rather to give better understanding of CCCA security itself, and we believe that our results give insights for constructing CCCA secure KEMs, and are useful for the future users/designers of CCCA secure KEMs.

Due to space limitation, the full proofs of the theorems in this paper will be given in the full version. We instead give proof sketches for each theorem.

Related Work. After Hofheinz and Kiltz [17] defined CCCA security, several practical CCCA secure KEMs have been proposed [6,12,20,13,14]. Hiwatari et al. [16] extended the CCCA secure KEM by Hanaoka and Kurosawa [12] to a CCCA secure multi-recipient KEM. Sakai et al. [27] used a OW-CCCA secure KEM which has reproducibility to construct a CCA secure KEM whose ciphertext length is shorter than that of the building block KEM, using a random oracle.

Bellare et al. [3] formalized the security notions for PKE schemes in a systematic way and showed the relations among security notions. For KEMs, Nagao et al. [23] and Herranz et al. [15] showed the relations among security notions. Moreover, Herranz et al. investigated the security notions achieved by hybrid encryption from a KEM and a DEM with several different levels of security.

[2] Very recently, Hohenberger et al. [18] used an IND-1-CCA secure PKE scheme as one of building blocks to construct a (fully) CCA secure PKE scheme. Although their construction still does not yield a practical scheme (at least compared to the concrete schemes, e.g. [10,21,17,6,12]), it would be interesting to seek for another application of 1-(or more-)bounded CCA secure schemes.

Bellare and Sahai [5] and later Pass, Shelat, and Vaikuntanathan [24] investigated the relations among several types of non-malleability [11]. Cramer et al. [8] introduced bounded CCA security, and show that non-malleability is separated from bounded CCA security. Matsuda and Matsuura [22] considered parallel decryption queries (which was originally introduced by Bellare and Sahai [5] in the context of non-malleability of PKE schemes) in bounded CCA security of PKE schemes and KEMs, and show several general implication/separation results. We note that the relations among security notions we show in this paper are not covered by these previous works.

2 Preliminaries

In this section, we review the basic notation and the definitions for a KEM.

Basic Notation. $\mathbb{N}$ denotes the set of all natural numbers, and if $n \in \mathbb{N}$ then $[n] = \{1, \ldots, n\}$. "$x \leftarrow y$" denotes that x is chosen uniformly at random from y if y is a finite set, x is output from y if y is a function or an algorithm, or y is assigned to x otherwise. If x and y are strings, then "$|x|$" denotes the bit-length of x, "$\mathsf{msb}(x)$" denotes the most significant bit of x, and "$x||y$" denotes a concatenation x and y. "PPTA" denotes a *probabilistic polynomial time algorithm*. If $\mathcal{A}$ is a probabilistic algorithm then $y \leftarrow \mathcal{A}(x; r)$ denotes that $\mathcal{A}$ computes y as output by taking x as input and using r as randomness. $\mathcal{A}^{\mathcal{O}}$ denotes an algorithm $\mathcal{A}$ with oracle access to $\mathcal{O}$. A function $f(k) : \mathbb{N} \to [0, 1]$ is said to be *negligible* if for all positive polynomials $p(k)$ and all sufficiently large $k \in \mathbb{N}$, we have $f(k) < 1/p(k)$.

Syntax of KEMs. A key encapsulation mechanism (KEM) Γ consists of the following three PPTAs $(\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$:

- KG: The key generation algorithm that takes 1^k (security parameter k) as input, and outputs a public/secret key pair (pk, sk).
- Enc: The encapsulation algorithm that takes pk as input, and outputs a ciphertext c and a session-key $K \in \mathcal{K}$ (where $\mathcal{K}$ is the session-key space specified by pk).
- Dec: The (deterministic) decapsulation algorithm that takes sk and c as input, and outputs a session-key K which could be a special symbol $\perp$ meaning "c is an invalid ciphertext".

We require $\mathsf{Dec}(sk, c) = K$ for all (pk, sk) output by KG and all (c, K) output by $\mathsf{Enc}(pk)$.

Security Notions for KEMs. Typically, security notions for KEMs are expressed by the combination of a security goal (`GOAL`) and an adversary's attack type (`ATK`). In this paper, we will treat *indistinguishability* (`IND`) and *one-wayness* (`OW`) as security goals, and *chosen plaintext attacks* (`CPA`), *q-bounded chosen ciphertext attacks* (q-`CCA`) [8], *constrained CCA* (`CCCA`) [17], and its q-bounded analogue, namely, *q-bounded CCCA* (q-`CCCA`) as an adversary's attack types, where $q \geq 0$ is an integer.

$\mathsf{Expt}^{\mathtt{IND\text{-}ATK}}_{\Gamma,\mathcal{A}}(k)$:
$(pk, sk) \leftarrow \mathsf{KG}(1^k)$; $b \leftarrow \{0,1\}$;
$(c^*, K_1^*) \leftarrow \mathsf{Enc}(pk)$; $K_0^* \leftarrow \mathcal{K}$;
$b' \leftarrow \mathcal{A}^{\mathcal{O}}(pk, c^*, K_b^*)$;
`If` $b' = b$ `then return` 1 `else return` 0

$\mathsf{Expt}^{\mathtt{OW\text{-}ATK}}_{\Gamma,\mathcal{A}}(k)$:
$(pk, sk) \leftarrow \mathsf{KG}(1^k)$; $(c^*, K^*) \leftarrow \mathsf{Enc}(pk)$;
$K' \leftarrow \mathcal{A}^{\mathcal{O}}(pk, c^*)$;
`If` $K' = K^*$ `then`
`return` 1 `else return` 0

Fig. 1. The security experiment for indistinguishability (`IND-ATK` experiment) (left) and that for one-wayness (`OW-ATK` experiment) (right)

For a KEM $\Gamma = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$, we define the experiment $\mathsf{Expt}^{\mathtt{IND\text{-}ATK}}_{\Gamma,\mathcal{A}}(k)$ in which an adversary $\mathcal{A}$ attacks indistinguishability of Γ under the attack type `ATK`, and the experiment $\mathsf{Expt}^{\mathtt{OW\text{-}ATK}}_{\Gamma,\mathcal{A}}(k)$ in which $\mathcal{A}$ attacks one-wayness of Γ under `ATK`, as in Fig. 1.

In the experiments, how the oracle $\mathcal{O}$ is defined and how it is available for $\mathcal{A}$ is determined depending on `ATK` in the following ways:

- If $\mathtt{ATK} = \mathtt{CPA}$, the oracle is unavailable and thus $\mathcal{A}$ cannot make any query.
- If $\mathtt{ATK} = q\text{-}\mathtt{CCA}$, the oracle is the decapsulation oracle $\mathcal{O}(\cdot) = \mathsf{Dec}(sk, \cdot)$, and $\mathcal{A}$ can submit at most q queries. Furthermore, $\mathcal{A}$ is not allowed to submit the challenge ciphertext c^* to $\mathcal{O}$.
- If $\mathtt{ATK} \in \{\mathtt{CCCA}, q\text{-}\mathtt{CCCA}\}$, the oracle is the *constrained decapsulation (CDEC) oracle* $\mathcal{O}_{cdec}(\cdot,\cdot)$, which takes a predicate $\mathsf{pred} : \mathcal{K} \to \{0,1\}$ and a ciphertext c as input, and returns a response as follows:

$$\mathcal{O}_{cdec}(\mathsf{pred}(\cdot), c) = \begin{cases} K & \text{If } \mathsf{Dec}(sk, c) = K \neq \perp \wedge \mathsf{pred}(K) = 1 \\ \perp & \text{Otherwise} \end{cases}$$

  Moreover, $\mathcal{A}$ is not allowed to submit a query containing c^* to $\mathcal{O}_{cdec}$. Additionally, if $\mathtt{ATK} = q\text{-}\mathtt{CCCA}$, $\mathcal{A}$ can submit at most q queries (as in q-`CCA`).

For a KEM Γ and $\mathtt{GOAL} \in \{\mathtt{IND}, \mathtt{OW}\}$, let $\mathcal{A}$ be an adversary that runs in $\mathsf{Expt}^{\mathtt{GOAL\text{-}CCCA}}_{\Gamma,\mathcal{A}}(k)$ and makes in total Q queries, and let $(\mathsf{pred}_i(\cdot), c_i)$ be $\mathcal{A}$'s i-th CDEC query. "*The running time of* $\mathcal{A}$ *in the* `GOAL-CCCA` *experiment*" is defined as the sum of $\mathcal{A}$'s running time and the total of maximum running time for evaluating each pred_i submitted by $\mathcal{A}$. "*The running time of the* `GOAL-CCCA` *experiment*" is defined as the total running time of the whole experiment $\mathsf{Expt}^{\mathtt{GOAL\text{-}CCCA}}_{\Gamma,\mathcal{A}}(k)$ minus "the running time of $\mathcal{A}$ in the `GOAL-CCCA` experiment". For a CCCA adversary $\mathcal{A}$ and an experiment $\mathcal{E}$ (not necessarily $\mathsf{Expt}^{\mathtt{GOAL\text{-}CCCA}}_{\Gamma,\mathcal{A}}(k)$) that $\mathcal{A}$ runs in, we define the parameter called (plaintext) *uncertainty* $\mathsf{uncert}_{\mathcal{A},\mathcal{E}}(k)$ by:

$$\mathsf{uncert}_{\mathcal{A},\mathcal{E}}(k) = \frac{1}{Q} \sum_{i \in [Q]} \Pr[\mathcal{E}; K \leftarrow \mathcal{K} : \mathsf{pred}_i(K) = 1]$$

Finally, we say that an adversary $\mathcal{A}$ is a *valid* `GOAL-CCCA` *adversary* if (1) "the running time of $\mathcal{A}$ in the `GOAL-CCCA` experiment" is polynomial in k, and (2) $\mathsf{uncert}_{\mathcal{A},\mathcal{E}}(k)$ is negligible for all experiments $\mathcal{E}$ whose running time is at most

"the running time of GOAL-CCCA experiment" that $\mathcal{A}$ runs in. We define the notion of a "valid GOAL-q-CCCA adversary" in exactly the same way as above.

For a KEM Γ, an adversary $\mathcal{A}$, and $\mathtt{ATK} \in \{\mathtt{CPA}, q\text{-}\mathtt{CCA}, \mathtt{CCCA}, q\text{-}\mathtt{CCCA}\}$, we define IND-ATK advantage $\mathsf{Adv}^{\mathtt{IND\text{-}ATK}}_{\Gamma,\mathcal{A}}(k)$ and OW-ATK advantage $\mathsf{Adv}^{\mathtt{OW\text{-}ATK}}_{\Gamma,\mathcal{A}}(k)$ by:

$$\mathsf{Adv}^{\mathtt{IND\text{-}ATK}}_{\Gamma,\mathcal{A}}(k) = |\Pr[\mathsf{Expt}^{\mathtt{IND\text{-}ATK}}_{\Gamma,\mathcal{A}}(k) = 1] - \frac{1}{2}|$$
$$\mathsf{Adv}^{\mathtt{OW\text{-}ATK}}_{\Gamma,\mathcal{A}}(k) = \Pr[\mathsf{Expt}^{\mathtt{OW\text{-}ATK}}_{\Gamma,\mathcal{A}}(k) = 1]$$

Definition 1. *Let* $\mathtt{GOAL} \in \{\mathtt{IND}, \mathtt{OW}\}$ *and* $q \in \mathbb{N}$. *We say that a KEM* Γ *is* GOAL-CPA *(resp.* GOAL-q-CCA*) secure if* $\mathsf{Adv}^{\mathtt{GOAL\text{-}CPA}}_{\Gamma,\mathcal{A}}(k)$ *(resp.* $\mathsf{Adv}^{\mathtt{GOAL}\text{-}q\text{-}\mathtt{CCA}}_{\Gamma,\mathcal{A}}(k)$*) is negligible for any PPTA* $\mathcal{A}$. *We say that a KEM* Γ *is* GOAL-CCCA *(resp.* GOAL-q-CCCA*) secure if* $\mathsf{Adv}^{\mathtt{GOAL\text{-}CCCA}}_{\Gamma,\mathcal{A}}(k)$ *(resp.* $\mathsf{Adv}^{\mathtt{GOAL}\text{-}q\text{-}\mathtt{CCCA}}_{\Gamma,\mathcal{A}}(k)$*) is negligible for any valid* GOAL-CCCA *(resp.* GOAL-q-CCCA*) adversary* $\mathcal{A}$.

3 Relations between Constrained and Bounded Chosen Ciphertext Security

In this section, we investigate relations between constrained and bounded CCA security. One might expect that there is always a separation of CCCA security from bounded CCA security, and vice versa. It is actually the case, and we formally show that for most cases we have separations in both directions. Perhaps somewhat surprisingly, however, it turns out that there is an implication from IND-CCCA security to the weakest form of bounded CCA security, namely, OW-1-CCA, *if we slightly change the definition of a valid CCCA adversary.*

The rest of this section is organized as follows: In Section 3.1, we show the separations between CCCA and bounded CCA security. Then, in Section 3.2 we introduce slightly stronger CCCA security and its implication to OW-1-CCA security.

3.1 Separations

Basic Ideas for Separations. Notice that a CDEC query by a valid CCCA adversary $\mathcal{A}$ is answered with a value that is not $\perp$ only when $\mathcal{A}$ already has some "non-trivial" knowledge about the decapsulation result, where the non-triviality is captured by the condition that $\mathcal{A}$ has to control the uncertainty negligible. We note that CDEC queries made by a valid CCCA adversary $\mathcal{A}$ cannot (except with negligible probability) reveal information on the decapsulation result that is hard to guess and is independent from $\mathcal{A}$'s view, because otherwise $\mathcal{A}$'s uncertainty cannot be negligible. We use this idea for showing the separations of bounded CCA security from CCCA security.

On the other hand, CDEC queries by a valid adversary $\mathcal{A}$ can reveal (while controlling $\mathcal{A}$'s uncertainty negligible) information that is dependent on some part of a public key, even if the decapsulation result itself is hard to guess, as

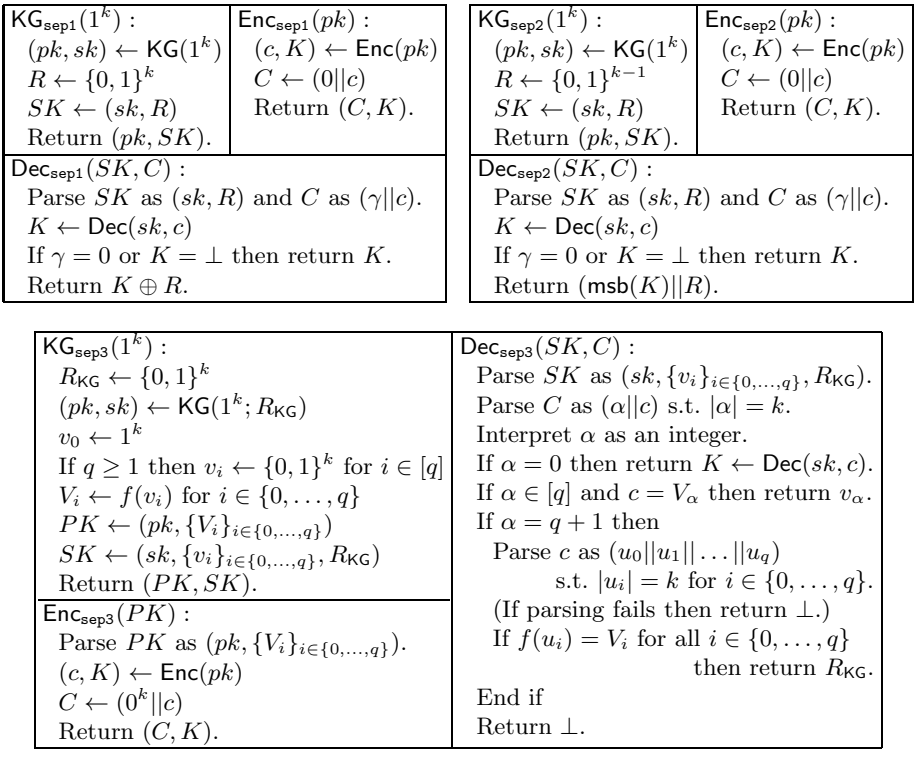


Fig. 2. The KEM Γ_{sep1} that separates OW-2-CCA from IND-CCCA (upper-left), the KEM Γ_{sep2} that separates IND-1-CCA from IND-CCCA (upper-right), and the KEM Γ_{sep3} that separates OW-$(q+1)$-CCCA from IND-q-CCA (bottom). In Γ_{sep3}, f is a one-way function.

long as it is non-trivial. This idea is later used to separate CCCA security from bounded CCA security in terms of the number of queries.

For simplicity, in this subsection we assume that the session-key space of a KEM is $\{0,1\}^k$ when the key generation algorithm is run with input 1^k.

IND-CCCA *vs.* OW-2-CCA. Choi et al. [7] showed that the KEM part of the Kurosawa-Desmedt PKE scheme [21], which was shown to be IND-CCCA secure under the DDH assumption in [17], is not OW-2-CCA secure. This result implies that if there is a group with prime order in which the DDH assumption holds, then there exists a KEM which is IND-CCCA secure but is not OW-2-CCA secure (and thus IND-CCCA security does not imply OW-2-CCA security, under the DDH assumption). We remove the DDH assumption from this statement, and show that in general IND-CCCA security does not imply OW-2-CCA security.

Theorem 1. *If there exists an* IND-CCCA *secure KEM, then there exists a KEM which is* IND-CCCA *secure but is not* OW-2-CCA *secure. Moreover, the* OW-2-CCA *attack for the latter KEM succeeds even if an adversary has to make two decapsulation queries parallelly (i.e. non-adaptively).*

Proof Sketch. Let $\Gamma = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CCCA secure KEM. Using the KEM Γ, we construct another KEM $\Gamma_{\mathsf{sep1}} = (\mathsf{KG}_{\mathsf{sep1}}, \mathsf{Enc}_{\mathsf{sep1}}, \mathsf{Dec}_{\mathsf{sep1}})$ for the separation as in Fig. 2 (upper-left).

The OW-2-CCA attack for Γ_{sep1} is easy: Consider the following OW-2-CCA adversary $\mathcal{A}$. Given $(pk, C^* = (0||c^*))$, $\mathcal{A}$ computes $(c', K') \leftarrow \mathsf{Enc}(pk)$, and submits ciphertexts $C_1 = (1||c^*)$ and $C_2 = (1||c')$ parallelly (i.e. non-adaptively) as decapsulation queries. According to the definition of $\mathsf{Dec}_{\mathsf{sep1}}$, $\mathcal{A}$ receives $K_1 = K^* \oplus R$ and $K_2 = K' \oplus R$, respectively, from the decapsulation oracle. Then $\mathcal{A}$ calculates $R \leftarrow K_2 \oplus K'$ and $K^* \leftarrow K_1 \oplus R$, and terminates with output K^*. It is easy to see that $\mathcal{A}$'s OW-2-CCA advantage is 1.

In order to show that Γ_{sep1} is IND-CCCA secure based on IND-CCCA security of the building block KEM Γ, consider the following sequence of games:

Game 1. This is the original IND-CCCA experiment, i.e. $\mathsf{Expt}^{\text{IND-CCCA}}_{\Gamma_{\mathsf{sep1}},\mathcal{A}}(k)$.

Game 2. Same as Game 1, except that any CDEC query containing a ciphertext of the form $C = (1||c)$ is answered with $\perp$.

Let $\mathcal{A}$ be any valid IND-CCCA adversary that makes in total Q CDEC queries. Then the difference in Game 1 and Game 2 can occur only when $\mathcal{A}$ submits a CDEC query (pred, C) satisfying $C = (1||c)$, $\mathsf{Dec}_{\mathsf{sep1}}(SK, C) = K \neq \perp$, and $\mathsf{pred}(K) = 1$. (In Game 1, it is answered with K, while in Game 2 it is answered with $\perp$.) By definition of $\mathsf{Dec}_{\mathsf{sep1}}$, if $C = (1||c)$ and $\mathsf{Dec}(sk, c) \neq \perp$, then $\mathsf{Dec}_{\mathsf{sep1}}(SK, C) = \mathsf{Dec}(sk, c) \oplus R$. However, notice that the information on R is information-theoretically hidden from $\mathcal{A}$'s view in Game 2. Moreover, R is chosen uniformly from $\{0,1\}^k$, and thus the decapsulation result $\mathsf{Dec}(sk, c) \oplus R$ of the query of the above type is also uniformly random and independent of $\mathcal{A}$'s view in Game 2. Then, the probability that some of $\mathcal{A}$'s CDEC queries of the form $(\mathsf{pred}, C = (1||c))$ satisfies $\mathsf{Dec}_{\mathsf{sep1}}(SK, C) = K' \neq \perp$ and $\mathsf{pred}(K') = 1$ will be upperbounded by $Q \cdot \mathsf{uncert}_{\mathcal{A},\mathbf{Game\ 2}}(k)$, which is negligible due to the fact that $\mathcal{A}$ is a valid IND-CCCA adversary. Moreover, Game 2 can be perfectly simulated by another valid IND-CCCA adversary for the building block KEM Γ, which means that $\mathcal{A}$'s advantage in Game 2 is negligible. In summary, $\mathcal{A}$'s IND-CCCA advantage is upperbounded to be negligible. □

IND-CCCA *vs. Non-malleability.* In the above theorem, to break OW-2-CCA security of the KEM Γ_{sep1}, the two decapsulation queries can be made parallelly. Hence, due to the equivalence of non-malleability under chosen plaintext attack and indistinguishability under one parallel decapsulation query [5,23,15], and the transitivity of the implication of security notions, it follows that IND-CCCA security does not imply non-malleability (under chosen plaintext attack).

IND-CCCA *vs.* IND-1-CCA. We next show that if the security goal is IND, then even IND-1-CCA security is in general separated from IND-CCCA.

Theorem 2. *If there exists an* IND-CCCA *secure KEM, then there exists a KEM which is* IND-CCCA *secure but is not* IND-1-CCA *secure.*

Proof Sketch. Let $\Gamma = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CCCA secure KEM. Using the KEM Γ, we construct another KEM $\Gamma_{\mathtt{sep2}} = (\mathsf{KG}_{\mathtt{sep2}}, \mathsf{Enc}_{\mathtt{sep2}}, \mathsf{Dec}_{\mathtt{sep2}})$ for the separation as in Fig. 2 (upper-right).

The IND-1-CCA attack for $\Gamma_{\mathtt{sep2}}$ is quite easy to see. Consider the following IND-1-CCA adversary $\mathcal{A}$. Given $(pk, C^* = (0||c^*), K_b^*)$, $\mathcal{A}$ submits a decapsulation query $C = (1||c^*)$, and receives the result K, which must be of the form $K = (\mathsf{msb}(K_1^*)||R)$ according to the definition of $\mathsf{Dec}_{\mathtt{sep2}}$. Then $\mathcal{A}$ checks if $\mathsf{msb}(K_b^*) = \mathsf{msb}(K)$, and outputs $b' = 1$ if this is the case, and outputs $b' = 0$ otherwise. A simple calculation shows that $\mathcal{A}$'s IND-1-CCA advantage is $1/4$.

The proof of IND-CCCA security of $\Gamma_{\mathtt{sep2}}$ based on IND-CCCA security of the building block KEM Γ proceeds almost in the same way as that of $\Gamma_{\mathtt{sep1}}$, considering the two games Game 1 ($\mathsf{Expt}^{\mathtt{IND\text{-}CCCA}}_{\Gamma_{\mathtt{sep2}},\mathcal{A}}(k)$ itself) and Game 2 (in which every CDEC query containing a ciphertext of the form $C = (1||c)$ is rejected). Game 1 and Game 2 are identical unless a valid IND-CCCA adversary $\mathcal{A}$ makes a CDEC query (pred, C) satisfying $C = (1||c)$, $\mathsf{Dec}_{\mathtt{sep2}}(SK, C) = K \neq \perp$, and $\mathsf{pred}(K) = 1$. The decapsulation result of such a ciphertext is of the form $(\mathsf{msb}(\mathsf{Dec}(sk, c))||R)$ where R is the value in the secret key (if c is not invalid). However, recall that this R is chosen uniformly at random, and is information-theoretically hidden from $\mathcal{A}$ and independent of $\mathcal{A}$'s view in Game 2. Therefore, the predicates contained in $\mathcal{A}$'s CDEC queries of the above type are almost never satisfied by the corresponding decapsulation results due to the condition that $\mathcal{A}$ has to control its uncertainty negligible, which implies that the difference between $\mathcal{A}$'s success probability (in guessing the challenge bit) in Game 1 and that in Game 2 is negligible. More specifically, in the full proof, we show that the difference in $\mathcal{A}$'s success probability in these games is upperbounded by $2Q \cdot \mathsf{uncert}_{\mathcal{A},\mathbf{Game\ 2}}(k)$ where Q is the total number of $\mathcal{A}$'s CDEC queries (the reason why "2" appears is because the value R in $\Gamma_{\mathtt{sep2}}$ is not k-bit but $(k-1)$-bit, and we lose the factor 2 when relating it with the uncertainty that considers whether the predicates are satisfied by a k-bit randomness). The fact that $\mathcal{A}$'s advantage in Game 2 is negligible follows from the IND-CCCA security of the building block KEM Γ, as in $\Gamma_{\mathtt{sep1}}$. □

IND-q-CCA *vs.* OW-$(q+1)$-CCCA. The above separations show that CCCA security does not imply bounded CCA security in most cases. Here, we show the separation of the opposite direction: if there is no trivial implication in terms of the number of queries, CCCA security is separated from bounded CCA security.

Theorem 3. *For any polynomial $q \geq 0$, if there exists an* IND-q-CCA *secure KEM, then there exists a KEM which is* IND-q-CCA *secure but is not* OW-$(q+1)$-CCCA *secure.*

Proof Sketch. Fix $q \geq 0$. Let $\Gamma = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-q-CCA secure KEM. Here, without loss of generality, we assume that the randomness space of KG is $\{0,1\}^k$. Moreover, let $f : \{0,1\}^* \to \{0,1\}^*$ be a one-way function (OWF), whose existence is guaranteed by the existence of Γ. Using the KEM Γ and the OWF f, we construct another KEM $\Gamma_{\mathtt{sep3}} = (\mathsf{KG}_{\mathtt{sep3}}, \mathsf{Enc}_{\mathtt{sep3}}, \mathsf{Dec}_{\mathtt{sep3}})$ for the

separation as in Fig. 2 (bottom). In the following, whenever we treat an integer as a k-bit string, we use "hat" (e.g. $\widehat{1}$ is the k-bit representation of 1).

The $\mathtt{OW}$-$(q+1)$-$\mathtt{CCA}$ attack against the KEM Γ_{sep3} is as follows (if $q = 0$, then we skip this part and goes to the $(q+1)$-th query below): Given $(PK, C^* = (0^k||c^*))$, for $i \in [q]$, a $\mathtt{OW}$-$(q+1)$-$\mathtt{CCCA}$ adversary $\mathcal{A}$ defines the predicate pred_i by "$\mathsf{pred}_i(K) = 1$ iff $f(K) = V_i$", and submits the i-th CDEC query of the form $(\mathsf{pred}_i, C_i = (\widehat{i}||V_i))$. Since $\mathsf{Dec}_{\mathsf{sep3}}(SK, C_i) = v_i$ by definition, $\mathcal{A}$ receives v_i from the oracle. After obtaining $v_1, \ldots, v_q$, $\mathcal{A}$ defines the $(q+1)$-th predicate pred_{q+1} by "$\mathsf{pred}_{q+1}(K) = 1$ iff $\mathsf{KG}(1^k; K) = (pk, *)$", sets $C_{q+1} \leftarrow (\widehat{q+1}||v_0||v_1||\ldots||v_q)$, and submits $(\mathsf{pred}_{q+1}, C_{q+1})$ to the oracle as the $(q+1)$-th CDEC query. Since $\mathsf{Dec}(SK, C_{q+1}) = R_{\mathsf{KG}}$, $\mathcal{A}$ receives R_{KG} as a response. $\mathcal{A}$ can then compute sk from R_{KG} and decrypt c^*, and thus $\mathcal{A}$'s $\mathtt{OW}$-$(q+1)$-$\mathtt{CCCA}$ advantage is 1.

Here, we also have to show that the above $\mathcal{A}$ is a valid $\mathtt{OW}$-$(q+1)$-$\mathtt{CCCA}$ adversary. We have to be careful because we have to show that $\mathcal{A}$'s uncertainty is negligible for *any* experiment $\mathcal{E}$ that is as efficient as the original $\mathtt{OW}$-$(q+1)$-$\mathtt{CCCA}$ experiment.[3] Fortunately, we can use the following statistical property that is satisfied by any OWF. (The proof is given in the full version.)

Lemma 1. *If f is a OWF, then* $\Pr[x \leftarrow \{0,1\}^k : f(x) = y]$ *is negligible for any string* $y \in \{0,1\}^*$.

This guarantees that, for $i \in [q]$, whatever value is assigned to V_i by an experiment $\mathcal{E}$, the probability that pred_i is satisfied by a random K is negligible. Furthermore, recall that the key generation algorithm of any secure (at least $\mathtt{OW}$-$\mathtt{CPA}$ secure) KEM can be viewed as a OWF whose domain is the randomness space of KG and whose image is pk (sk is discarded). Then we can use Lemma 1 also for the $(q+1)$-th CDEC predicate pred_{q+1}, and conclude that $\mathsf{uncert}_{\mathcal{A},\mathcal{E}}(k)$ is negligible for any experiment $\mathcal{E}$.

$\mathtt{IND}$-q-$\mathtt{CCA}$ security of Γ_{sep3} is explained as follows. Let $\mathcal{A}$ be any $\mathtt{IND}$-q-$\mathtt{CCA}$ adversary against Π_{sep3}. Recall that a decapsulation query of the form $C = (\widehat{q+1}||c)$ is answered with R_{KG} only when all preimages $v_0, v_1, \ldots, v_q$ are known to $\mathcal{A}$. Since v_0 is the fixed value 1^k, $\mathcal{A}$ actually needs to find q preimages $v_1, \ldots, v_q$. However, due to one-wayness of f, it is hard to find v_i without making a decapsulation query of the form $C = (\widehat{i}||V_i)$. But since $\mathcal{A}$ can make only q queries, if $\mathcal{A}$ makes q queries to obtain $(v_1, \ldots, v_q)$, $\mathcal{A}$ can no longer use the decapsulation oracle. This means that unless $\mathcal{A}$ breaks the OWF, $\mathcal{A}$ cannot make a decapsulation query of the form $C = (\widehat{q+1}||c)$ that reveals R_{KG}. Then, in order to break $\mathtt{IND}$-q-$\mathtt{CCA}$ security of Γ_{sep3}, $\mathcal{A}$ has to essentially break $\mathtt{IND}$-q-$\mathtt{CCA}$ security of the building block KEM Γ, which is hard by assumption. □

3.2 Slightly Stronger CCCA Security and Its Implication

In the previous subsection, we have seen that $\mathtt{IND}$-$\mathtt{CCCA}$ security does not imply $\mathtt{OW}$-2-$\mathtt{CCA}$ or $\mathtt{IND}$-1-$\mathtt{CCA}$ security. Then, a natural question would be whether

[3] For example, $\mathcal{A}$'s uncertainty has to be negligible in which PK and/or C^* are generated incorrectly (as long as the experiment is efficient).

IND-CCCA security implies (or does not imply) OW-1-CCA security, which is the weakest bounded CCA security for KEMs. Actually, we could not show implication/separation from IND-CCCA. Alternatively, however, we find that if we consider a slightly stronger definition for IND-CCCA, we actually have an implication. The modification we will make is in the definition of uncertainty, and is quite subtle. We explain this in this subsection.

Note that the IND-CCCA experiment is fixed if we fix the following: (a) the randomness for key generation ($(pk, sk) \leftarrow \mathsf{KG}(1^k)$), (b) the randomness for challenge ciphertext/session-keys ($(c^*, K_1^*) \leftarrow \mathsf{Enc}(pk)$ and $K_0^* \leftarrow \mathcal{K}$), (c) the challenge bit ($b \leftarrow \{0,1\}$), and (d) the randomness for an adversary. We denote the process of randomly picking these randomness and fixing the IND-CCCA experiment by "$\mathcal{E} \leftarrow \mathsf{Expt}_{\Gamma,\mathcal{A}}^{\texttt{IND-CCCA}}(k)$". We introduce the following definition.

Definition 2. *Let Γ be a KEM and $\mathcal{A}$ be an* IND-CCCA *adversary (against Γ) that makes Q CDEC queries. Let pred_i be the predicate contained in $\mathcal{A}$'s i-th CDEC query. We define the* average uncertainty $\mathsf{uncert}_{\mathcal{A}}^{ave}(k)$ *of $\mathcal{A}$ by:*

$$\begin{aligned}\mathsf{uncert}_{\mathcal{A}}^{ave}(k) &= \mathop{\mathbf{E}}_{\mathcal{E} \leftarrow \mathsf{Expt}_{\Gamma,\mathcal{A}}^{\texttt{IND-CCCA}}(k)} [\mathsf{uncert}_{\mathcal{A},\mathcal{E}}(k)] \\ &= \mathop{\mathbf{E}}_{\mathcal{E} \leftarrow \mathsf{Expt}_{\Gamma,\mathcal{A}}^{\texttt{IND-CCCA}}(k)} \left[\frac{1}{Q} \sum_{i \in [Q]} \Pr[\mathcal{E}; K \leftarrow \mathcal{K} : \mathsf{pred}_i(K) = 1] \right]\end{aligned}$$

Furthermore, we say that $\mathcal{A}$ is a valid IND-CCCA* adversary *(against Γ) if (1) the running time of $\mathcal{A}$ in the* IND-CCCA *experiment is polynomial in k, and (2)* $\mathsf{uncert}_{\mathcal{A}}^{ave}(k)$ *is negligible.*

Using average uncertainty, we define IND-CCCA* security of a KEM as follows:

Definition 3. *We say that a KEM Γ is* IND-CCCA* *secure if* $\mathsf{Adv}_{\Gamma,\mathcal{A}}^{\texttt{IND-CCCA}}(k)$ *is negligible for any valid* IND-CCCA* *adversary $\mathcal{A}$.*

We define OW-CCCA*, IND-q-CCCA*, and OW-q-CCCA* security in exactly the same way as above.

Note that to define IND-CCCA* security, we have not changed anything about the definition of IND-CCCA advantage $\mathsf{Adv}_{\Gamma,\mathcal{A}}^{\texttt{IND-CCCA}}(k)$. The only difference between IND-CCCA* security defined here and the original IND-CCCA security in [17] is for which class of adversaries we require the advantage to be negligible. In order for a CCCA adversary $\mathcal{A}$ to be valid as an IND-CCCA* adversary, $\mathcal{A}$ only needs to control his uncertainty in the original IND-CCCA experiment to be negligible on an average, and thus for example, its uncertainty can be 1 accidentally (as long as it is negligible on an average). On the other hand, the original IND-CCCA security definition requires that $\mathcal{A}$'s uncertainty to be negligible for any experiment whose running time is at most that of the original IND-CCCA experiment. Therefore, if $\mathcal{A}$ is a valid IND-CCCA adversary, then it is a valid IND-CCCA* adversary as well. Since IND-CCCA security requires the IND-CCCA advantage to be negligible for adversaries of a smaller class, IND-CCCA* security implies IND-CCCA security.

Although the difference between IND-CCCA* and IND-CCCA security seems quite subtle and small, so far we are not sure if the latter implies (or is separated from) the former, and we would like to leave it as an open problem.

Now, we show the implication that bridges CCCA and bounded CCA security.

Theorem 4. *If a KEM is* IND-1-CCCA* *secure, then it is* OW-1-CCA *secure.*

Proof Sketch. Without loss of generality, a OW-1-CCA adversary $\mathcal{A}$ can be divided into two stages $(\mathcal{A}_1, \mathcal{A}_2)$ so that the OW-1-CCA experiment is rewritten as:

$$(pk, sk) \leftarrow \mathsf{KG}(1^k);\ (c^*, K^*) \leftarrow \mathsf{Enc}(pk);\ (\widehat{c}, \mathsf{st}) \leftarrow \mathcal{A}_1(pk, c^*);\ \widehat{K} \leftarrow \mathsf{Dec}(sk, \widehat{c});$$
$$K' \leftarrow \mathcal{A}_2(\widehat{K}, \mathsf{st});\ \texttt{If}\ K' = K^*\ \texttt{then return}\ 1\ \texttt{else return}\ 0$$

where $\widehat{c}$ represents $\mathcal{A}$'s decapsulation query (which can be made only once). Moreover, we can assume that $\mathcal{A}_2$ is deterministic because in case $\mathcal{A}_2$ needs randomness, it can be chosen by $\mathcal{A}_1$ and passed via st. Now, using a OW-1-CCA adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we construct an IND-1-CCCA adversary $\mathcal{B}$ as follows:

On input (pk, c^*, K_b^*) (where b is $\mathcal{B}$'s challenge bit), $\mathcal{B}$ runs $(\widehat{c}, \mathsf{st}) \leftarrow \mathcal{A}_1(pk, c^*)$. Then $\mathcal{B}$ defines pred by "$\mathsf{pred}(K) = 1$ iff $\mathcal{A}_2(K, \mathsf{st}) = K_b^*$" and submits a CDEC query $(\mathsf{pred}, \widehat{c})$. If the answer from $\mathcal{O}_{cdec}$ is not $\perp$, $\mathcal{B}$ outputs 1. Otherwise $\mathcal{B}$ checks if $\mathcal{A}_2(\perp, \mathsf{st}) = K_b^*$, and returns 1 if the check holds or returns 0 otherwise.

Assume that $\mathcal{A}$ breaks OW-1-CCA security with non-negligible advantage. Since $\mathcal{A}_2$, which is given a correct decapsulation $\mathsf{Dec}(sk, \widehat{c})$ during the evaluation of pred in $\mathcal{O}_{cdec}$, outputs $K_1^* = \mathsf{Dec}(sk, c^*)$ with non-negligible probability, $\mathcal{B}$ outputs 1 with non-negligible probability when $b = 1$. (The check "$\mathcal{A}_2(\perp, \mathsf{st}) = K_b^*$" performed by $\mathcal{B}$ covers the case in which $\mathsf{Dec}(sk, \widehat{c}) = \perp$.) On the other hand, K_0^* is information-theoretically hidden from $\mathcal{A}$'s view, and $\mathcal{A}_2$ can output it only with negligible probability. Thus, when $b = 0$, pred is almost never satisfied and $\mathcal{B}$ outputs 1 only with negligible probability. Therefore, there is a non-negligible difference in the probabilities that $\mathcal{B}$ outputs 1 between the cases $b = 1$ and $b = 0$, namely, $\mathcal{B}$ has non-negligible IND-1-CCCA advantage. The idea of using an adversary in the predicate in a CDEC query might be of independent interest.

Note that $\mathcal{B}$'s uncertainty depends on $\mathcal{A}$, and we could not rule out the possibility that there is an experiment (which is as efficient as the IND-1-CCCA experiment) such that when $\mathcal{B}$ (which internally runs $\mathcal{A}$) is run, $\mathcal{B}$'s uncertainty is non-negligible. However, it is possible, using IND-1-CCCA* security of the KEM itself, to show that $\mathcal{B}$'s average uncertainty is negligible, and thus $\mathcal{B}$ is shown to be a valid IND-1-CCCA* adversary. We can show this roughly because the definition of average uncertainty considers the distribution of the public key and the challenge ciphertext/session-key pair (rather than fixed values for these), which makes it possible to use security of the KEM itself. Specifically, in the full proof we show that if the KEM is OW-CPA secure (which is trivially satisfied by the IND-1-CCCA* security of the KEM), then $\mathcal{B}$ is a valid IND-1-CCCA* adversary. □

4 KEMs from Computational Hash Proof Systems, Revisited

In this section, we revisit the construction of KEMs from a HPS [17] for which we only require computational security properties, as opposed to the information-theoretic ones in [9,21]. More concretely, we show that if a HPS satisfies the computational security requirements defined in [17], then the KEM constructed based on the HPS satisfies not only IND-CCCA security but also IND-1-CCA security. (In particular, our result implies that the Kurosawa-Desmedt KEM [21] is IND-1-CCA secure under the DDH assumption.) This result should be contrasted with the separation of IND-1-CCA from IND-CCCA security in Section 3.2.

Below, we review the definitions of computational HPS in Section 4.1, and we show that the HPS-based KEM satisfies IND-1-CCA security in Section 4.2.

4.1 Definitions for Computational HPS

Here, we review the definition of hash proof systems as defined by Cramer and Shoup [9,21,17]. (We mainly borrow the notations from [17], which we customize slightly for our purpose.)

Let $\mathcal{C}$, $\mathcal{K}$, $\mathcal{S}$, and $\mathcal{P}$ be sets, and $\mathcal{V}$ be the set of "languages" satisfying $\mathcal{V} \subset \mathcal{C}$. Let $\mathsf{D}_{sk} : \mathcal{C} \to \mathcal{K}$ be a hash function indexed by $sk \in \mathcal{S}$. Informally speaking, a HPS is a special type of a designated-verifier proof system for a "subset membership problem" (i.e. whether a "statement" $c \in \mathcal{C}$ satisfies $c \in \mathcal{V}$). A hash function D_{sk} is said to be *projective* if there exists an efficiently computable projection $\mu : \mathcal{S} \to \mathcal{P}$ such that $pk = \mu(sk) \in \mathcal{P}$ defines the action of D_{sk} over the subset $\mathcal{V}$. That is, for every $c \in \mathcal{V}$, the value $K = \mathsf{D}_{sk}(C)$ is uniquely determined by $\mu(sk)$ and c. In the context of the HPS-based KEM that will be explained later, we will identify $\mathcal{C}$ as the ciphertext space, $\mathcal{V}$ as the set of all valid ciphertexts, $\mathcal{S}$ as the secret key space, $\mathcal{P}$ as the public key space, $\mathcal{K}$ as the session-key space, $\mu(\cdot)$ as the key generation algorithm, and $\mathsf{D}_{sk}(\cdot)$ as the decapsulation algorithm. Taking this into account, hereafter we call an element $c \in \mathcal{C}$ *valid* if $c \in \mathcal{V}$ and *invalid* if $c \in \mathcal{C} \backslash \mathcal{V}$. As usual, we require: (1) $\mathcal{C}$ is efficiently recognizable, (2) a valid element $c \in \mathcal{V}$ can be efficiently sampled together with a witness w about the fact that $c \in \mathcal{V}$, and (3) we can sample elements from $\mathcal{C} \backslash \mathcal{V}$, $\mathcal{S}$, and $\mathcal{K}$ efficiently and (statistically close to) uniformly.

The above are the description of the parameters for a HPS. For simplicity, we assume that the definitions of the sets and the functions we described above are generated and determined by a probabilistic algorithm HGen. Formally, a HPS Π consists of the following three PPTAs $(\mathsf{HGen}, \mathsf{Pub}, \mathsf{Priv})$:

HGen: The parameter generation algorithm for HPS which takes 1^k as input, and outputs parameters $\mathsf{pub} = (\mathcal{C}, \mathcal{V}, \mathcal{K}, \mathcal{S}, \mathcal{P}, \mathsf{D}_{(\cdot)} : \mathcal{C} \to \mathcal{K}, \mu : \mathcal{S} \to \mathcal{P})$. For notational convenience, we assume that pub is provided as input to the following algorithms Pub and Priv, and do not write it explicitly.

Pub: The (deterministic) public evaluation algorithm which takes $pk = \mu(sk) \in \mathcal{P}$, a valid element/witness pair (c, w) (where w is about the fact that $c \in \mathcal{V}$) as input, and outputs a hash value $K = \mathsf{D}_{sk}(c)$.

$\mathsf{Expt}^{\mathtt{CU}_2}_{\Pi,\mathsf{A}}(k)$:
$\mathsf{pub} \leftarrow \mathsf{HGen}(1^k)$;
$sk \leftarrow \mathcal{S}$; $pk \leftarrow \mu(sk)$;
$c^* \leftarrow \mathcal{C}\backslash\mathcal{V}$; $K^* \leftarrow \mathsf{D}_{sk}(c^*)$;
$(c', \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathcal{O}}(\mathsf{pub}, pk, c^*, K^*)$;
$K_1' \leftarrow \mathsf{D}_{sk}(c')$; $K_0' \leftarrow \mathcal{K}$; $b \leftarrow \{0,1\}$;
$b' \leftarrow \mathcal{A}_2(K_b', \mathsf{st})$;
`If` $b' = b$ `then return` 1 `else return` 0

$\mathsf{Expt}^{\mathtt{CU}_1}_{\Pi,\mathsf{A}}(k)$:
$\mathsf{pub} \leftarrow \mathsf{HGen}(1^k)$;
$sk \leftarrow \mathcal{S}$; $pk \leftarrow \mu(sk)$; $c^* \leftarrow \mathcal{C}\backslash\mathcal{V}$;
$K_1^* \leftarrow \mathsf{D}_{sk}(c^*)$; $K_0^* \leftarrow \mathcal{K}$; $b \leftarrow \{0,1\}$;
$b' \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pub}, pk, c^*, K_b^*)$;
`If` $b' = b$ `then return` 1 `else return` 0

The definition of the oracle $\mathcal{O}$ in $\mathsf{Expt}^{\mathtt{CU}_2}_{\Pi,\mathcal{A}}$ and $\mathsf{Expt}^{\mathtt{CU}_1}_{\Pi,\mathcal{A}}$:

$$\mathcal{O}(c) = \begin{cases} \mathsf{D}_{sk}(c) & \text{If } c \in \mathcal{V} \\ \perp & \text{Otherwise} \end{cases}$$

Fig. 3. The CU_2 experiment (left), the CU_1 experiment (upper-right), and the definition of the oracle (lower-right)

Priv: The (deterministic) private evaluation algorithm which takes $sk \in \mathcal{S}$ and an element $c \in \mathcal{C}$ as input, and outputs a hash value $K = \mathsf{D}_{sk}(c)$.

For all $\mathsf{pub} \leftarrow \mathsf{HGen}(1^k)$, we require the following: (1) for all $c \in \mathcal{C}$ and all $sk \in \mathcal{S}$, it holds that $\mathsf{Priv}(sk, c) = \mathsf{D}_{sk}(c)$, and (2) for all $c \in \mathcal{V}$ with the corresponding witness w (about the fact that $c \in \mathcal{V}$), and all $sk \in \mathcal{S}$, it holds that $\mathsf{Pub}(\mu(sk), c, w) = \mathsf{Priv}(sk, c) = \mathsf{D}_{sk}(c)$.

Security Requirements. As usual, we define the subset membership problem for a HPS Π and its hardness.

Definition 4. *We say that the subset membership problem in a HPS Π is hard if the following advantage function $\mathsf{Adv}^{\mathtt{SM}}_{\Pi,\mathcal{A}}(k)$ is negligible for any PPTA $\mathcal{A}$:*

$$\mathsf{Adv}^{\mathtt{SM}}_{\Pi,\mathcal{A}}(k) = |\Pr[\mathsf{pub} \leftarrow \mathsf{HGen}(1^k);\ b \leftarrow \{0,1\};\ c_1^* \leftarrow \mathcal{V};\ c_0^* \leftarrow \mathcal{C}\backslash\mathcal{V};\ b' \leftarrow \mathcal{A}(\mathsf{pub}, c_b^*) : b' = b] - \frac{1}{2}|$$

Hofheinz and Kiltz [17] defined the computational analogue of *strong universal*$_2$ that is defined in [21] for a HPS, called *computational universal*$_2$ (CU_2 security, for short), which we recall here. The CU_2 experiment $\mathsf{Expt}^{\mathtt{CU}_2}_{\Pi,\mathcal{A}}(k)$ for a HPS Π that an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ runs in is defined as in Fig. 3 (left). In the experiment, it is required that $\mathcal{A}_1$'s output c' satisfy $c' \in \mathcal{C}\backslash\mathcal{V}$ and $c' \neq c^*$.

Definition 5. *We say that a HPS Π is CU_2 secure if the advantage function $\mathsf{Adv}^{\mathtt{CU}_2}_{\Pi,\mathcal{A}}(k) = |\Pr[\mathsf{Expt}^{\mathtt{CU}_2}_{\Pi,\mathcal{A}}(k) = 1] - 1/2|$ is negligible for any PPTA $\mathcal{A}$.*

Moreover, we define a *universal*$_1$-analogue of CU_2 security, which we call *computational universal*$_1$ (CU_1 security, for short). We define the CU_1 experiment $\mathsf{Expt}^{\mathtt{CU}_1}_{\Pi,\mathcal{A}}(k)$ for a HPS Π that an adversary $\mathcal{A}$ runs in as in Fig. 3 (upper-right).

Definition 6. *We say that a HPS Π is CU_1 secure if the advantage function $\mathsf{Adv}^{\mathtt{CU}_1}_{\Pi,\mathcal{A}}(k) = |\Pr[\mathsf{Expt}^{\mathtt{CU}_1}_{\Pi,\mathcal{A}}(k) = 1] - 1/2|$ is negligible for any PPTA $\mathcal{A}$.*

$\mathsf{KG}(1^k)$:	$\mathsf{Enc}(PK)$:	$\mathsf{Dec}(sk, c)$:
$\mathsf{pub} \leftarrow \mathsf{HGen}(1^k)$	Pick $c \in \mathcal{V}$ uniformly	$K \leftarrow \mathsf{Priv}(sk, c)$
$sk \leftarrow \mathcal{S}$; $pk \leftarrow \mu(sk)$	together with a witness w.	Return K.
$PK \leftarrow (\mathsf{pub}, pk)$	$K \leftarrow \mathsf{Pub}(pk, c, w)$	
Return (PK, sk).	Return (c, K).	

Fig. 4. The KEM Γ_Π based on a HPS Π

Although CU_1 security is not explicitly defined in [17], it seems to us that this security is implicitly used for proving the CCCA security of the HPS-based KEM. Jiang and Wang [19] defined a slightly stronger version of CU_2 security which allows the second stage adversary $\mathcal{A}_2$ to have access to the oracle $\mathcal{O}$. This version of CU_2 security is satisfied by all known instantiations of HPS (see the following paragraph), and implies CU_1 security defined here. Thus, CU_1 security is not an additional security requirement for a HPS in practice. The reason why we introduce CU_1 security separately is that we believe that it makes our security analysis clearer. We also remark that CU_1 security is strictly weaker than "smoothness" defined in [19, Def. 7].

Concrete Instantiations of HPS. There are several known concrete instantiations of computational HPS that satisfy the above security requirements. The Kurosawa-Desmedt HPS [21,17] based on the DDH assumption, and its n-linear variant under the n-linear assumption [17], both of which are discrete logarithm-type constructions. Meanwhile, we also have a computational HPS based on the Paillier's decision composite residuosity assumption [9]. For more details, see [9,17] and the references therein.

4.2 HPS-Based KEM and Bounded CCA Security

Let $\Pi = (\mathsf{HGen}, \mathsf{Pub}, \mathsf{Priv})$ be a HPS. Then, the HPS-based KEM $\Gamma_\Pi = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ [21,17] is constructed as in Fig. 4.

The following was shown by Hofheinz and Kiltz [17].

Theorem 5. *([17]) If the subset membership problem of Π is hard, Π satisfies CU_2 and CU_1 security, then the HPS-based KEM Γ_Π is* `IND-CCCA` *secure.*

We show that under the same assumptions on the HPS used to prove its `IND-CCCA` security, the KEM Γ_Π satisfies `IND-1-CCA` security as well.

Theorem 6. *If the subset membership problem of Π is hard, Π satisfies CU_2 and CU_1 security, then the HPS-based KEM Γ_Π is* `IND-1-CCA` *secure.*

Intuition. CU_1 security of Π guarantees that, under the situation where the challenge ciphertext c^* is sampled from invalid elements (i.e. $c^* \leftarrow \mathcal{C} \backslash \mathcal{V}$), the real challenge session-key $K_1^* = \mathsf{D}_{sk}(c^*)$ looks random to $\mathcal{A}$, as long as $\mathcal{A}$'s decapsulation query is a valid one. However, $\mathcal{A}$ is free to choose a ciphertext for a decapsulation query, and in particular, it can be invalid. This is the place where

CU_2 security comes into play. CU_2 security of Π guarantees that, even if $\mathcal{A}$'s decapsulation query c is an invalid one, $\mathcal{A}$ gets no significant information from the response, compared to just receiving a random value in $\mathcal{K}$, as long as $\mathcal{A}$ does not make any query after this query (and it is guaranteed because $\mathcal{A}$ is an IND-1-CCA adversary). Therefore, CU_1 and CU_2 together guarantee that $\mathcal{A}$'s decapsulation query essentially gives no significant information for distinguishing the real challenge session-key K_1^* from a random. Although CU_1 and CU_2 security are guaranteed only when the challenge ciphertext is an invalid one, the hardness of the subset membership problem in Π guarantees that $\mathcal{A}$'s behavior cannot be non-negligibly different between the case in which the challenge ciphertext is a valid one (i.e. in the original IND-1-CCA experiment), and the case in which the challenge ciphertext is an invalid one (and thus we can use CU_1 and CU_2 security of Π).

Proof Sketch. Let $\mathcal{A}$ be any PPTA IND-1-CCA adversary against the HPS-based KEM Γ_Π. Consider the following sequence of games.

Game 1. This is the original IND-1-CCA experiment, i.e. $\mathsf{Expt}^{\texttt{IND-1-CCA}}_{\Gamma_\Pi,\mathcal{A}}(k)$.
Game 2. Same as Game 1, except that K_1^* is generated by $K_1^* \leftarrow \mathsf{Priv}(sk, c^*)$.
Game 3. Same as Game 2, except that c^* is picked uniformly from $\mathcal{C}\backslash\mathcal{V}$.
Game 4. Same as Game 3, except that if $\mathcal{A}$'s decapsulation query c satisfies $c \in \mathcal{C}\backslash\mathcal{V}$, then it is answered with a uniformly random value $K \in \mathcal{K}$.

For $i \in [4]$, let S_i be the event that $\mathcal{A}$ succeeds in guessing the challenge bit (i.e. $b' = b$ occurs) in Game i. $\mathcal{A}$'s IND-1-CCA advantage can be estimated as follows:

$$\mathsf{Adv}^{\texttt{IND-1-CCA}}_{\Gamma_\Pi,\mathcal{A}}(k) = |\Pr[\mathsf{S}_1] - \frac{1}{2}| \leq \sum_{i\in[3]} |\Pr[\mathsf{S}_i] - \Pr[\mathsf{S}_{i+1}]| + |\Pr[\mathsf{S}_4] - \frac{1}{2}|$$

The proof is completed by upperbounding each term in the right hand side of the above inequality to be negligible. We have $\Pr[\mathsf{S}_1] = \Pr[\mathsf{S}_2]$ due to the correctness of Π. $|\Pr[\mathsf{S}_2] - \Pr[\mathsf{S}_3]|$ is negligible due to the hardness of the subset membership problem in Π. $|\Pr[\mathsf{S}_3] - \Pr[\mathsf{S}_4]|$ and $|\Pr[\mathsf{S}_4] - 1/2|$ can be shown to be negligible by using CU_2 security and CU_1 security of Π, respectively. □

References

1. Abe, M., Gennaro, R., Kurosawa, K., Shoup, V.: Tag-KEM/DEM: A New Framework for Hybrid Encryption and A New Analysis of Kurosawa-Desmedt KEM. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 128–146. Springer, Heidelberg (2005)
2. Baek, J., Galindo, D., Susilo, W., Zhou, J.: Constructing Strong KEM from Weak KEM (or How to Revive the KEM/DEM Framework). In: Ostrovsky, R., De Prisco, R., Visconti, I. (eds.) SCN 2008. LNCS, vol. 5229, pp. 358–374. Springer, Heidelberg (2008)
3. Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations among Notions of Security for Public-Key Encryption Schemes. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 26–45. Springer, Heidelberg (1998)

4. Bellare, M., Namprempre, C.: Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 531–545. Springer, Heidelberg (2000)
5. Bellare, M., Sahai, A.: Non-malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 519–536. Springer, Heidelberg (1999); The revised version is available in Cryptology ePrint Archive (Report 2006/228)
6. Cash, D., Kiltz, E., Shoup, V.: The Twin Diffie-Hellman Problem and Applications. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 127–145. Springer, Heidelberg (2008)
7. Choi, S.G., Herranz, J., Hofheinz, D., Hwang, J.Y., Kiltz, E., Lee, D.H., Yung, M.: The Kurosawa-Desmedt key encapsulation is not chosen-ciphertext secure. Inf. Process. Lett. 109(16), 897–901 (2009)
8. Cramer, R., Hanaoka, G., Hofheinz, D., Imai, H., Kiltz, E., Pass, R., Shelat, A., Vaikuntanathan, V.: Bounded CCA2-Secure Encryption. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 502–518. Springer, Heidelberg (2007)
9. Cramer, R., Shoup, V.: Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)
10. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM J. Computing 33(1), 167–226 (2003)
11. Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography. In: STOC 1991, pp. 542–552 (1991)
12. Hanaoka, G., Kurosawa, K.: Efficient Chosen Ciphertext Secure Public Key Encryption under the Computational Diffie-Hellman Assumption. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 308–325. Springer, Heidelberg (2008)
13. Hanaoka, G., Kurosawa, K.: Between hashed DH and computational DH: Compact encryption from weaker assumption. IEICE Transactions E93-A(11), 1994–2006 (2010)
14. Haralambiev, K., Jager, T., Kiltz, E., Shoup, V.: Simple and Efficient Public-Key Encryption from Computational Diffie-Hellman in the Standard Model. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 1–18. Springer, Heidelberg (2010)
15. Herranz, J., Hofheinz, D., Kiltz, E.: Some (in)sufficient conditions for secure hybrid encryption. Inf. Comput. 208(11), 1243–1257 (2010)
16. Hiwatari, H., Tanaka, K., Asano, T., Sakumoto, K.: Multi-recipient Public-Key Encryption from Simulators in Security Proofs. In: Boyd, C., González Nieto, J. (eds.) ACISP 2009. LNCS, vol. 5594, pp. 293–308. Springer, Heidelberg (2009)
17. Hofheinz, D., Kiltz, E.: Secure Hybrid Encryption from Weakened Key Encapsulation. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 553–571. Springer, Heidelberg (2007)
18. Hohenberger, S., Lewko, A., Waters, B.: Detecting Dangerous Queries: A New Approach for Chosen Ciphertext Security. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 663–681. Springer, Heidelberg (2012); The full version is available in Cryptology ePrint Archive (Report 2012/006)
19. Jiang, S., Wang, H.: Plaintext-Awareness of Hybrid Encryption. In: Pieprzyk, J. (ed.) CT-RSA 2010. LNCS, vol. 5985, pp. 57–72. Springer, Heidelberg (2010)

20. Kiltz, E., Pietrzak, K., Stam, M., Yung, M.: A New Randomness Extraction Paradigm for Hybrid Encryption. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 590–609. Springer, Heidelberg (2009)
21. Kurosawa, K., Desmedt, Y.: A New Paradigm of Hybrid Encryption Scheme. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 426–442. Springer, Heidelberg (2004)
22. Matsuda, T., Matsuura, K.: Parallel Decryption Queries in Bounded Chosen Ciphertext Attacks. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 246–264. Springer, Heidelberg (2011)
23. Nagao, W., Manabe, Y., Okamoto, T.: On the equivalence of several security notions of KEM and DEM. IEICE Transactions E91-A(1), 283–297 (2008)
24. Pass, R., Shelat, A., Vaikuntanathan, V.: Relations Among Notions of Non-malleability for Encryption. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 519–535. Springer, Heidelberg (2007)
25. Phan, D.H., Pointcheval, D.: About the Security of Ciphers (Semantic Security and Pseudo-Random Permutations). In: Handschuh, H., Hasan, M.A. (eds.) SAC 2004. LNCS, vol. 3357, pp. 182–197. Springer, Heidelberg (2004)
26. Rackoff, C., Simon, D.R.: Non-interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)
27. Sakai, Y., Hanaoka, G., Kurosawa, K., Ohta, K.: A Generic Method for Reducing Ciphertext Length of Reproducible KEMs in the RO Model. In: Echizen, I., Kunihiro, N., Sasaki, R. (eds.) IWSEC 2010. LNCS, vol. 6434, pp. 55–69. Springer, Heidelberg (2010)

Solving a Discrete Logarithm Problem with Auxiliary Input on a 160-Bit Elliptic Curve

Yumi Sakemi[1,⋆], Goichiro Hanaoka[2], Tetsuya Izu[1], Masahiko Takenaka[1], and Masaya Yasuda[1]

[1] FUJITSU LABORATORIES Ltd.,
4-1-1, Kamikodanaka, Nakahara-ku, Kawasaki, 211-8588, Japan
{sakemi,izu,takenaka,myasuda}@labs.fujitsu.com
[2] Research Institute for Secure Systems (RISEC),
National Institute of Advanced Industrial Science and Technology (AIST),
Central 2, 1-1-1, Umezono, Tsukuba, 305-8568, Japan
hanaoka-goichiro@aist.go.jp

Abstract. A discrete logarithm problem with auxiliary input (DLPwAI) is a problem to find α from G, αG, $\alpha^d G$ in an additive cyclic group generated by an element G of prime order r, and a positive integer d satisfying $d|(r-1)$. The infeasibility of this problem assures the security of some cryptographic schemes. In 2006, Cheon proposed a novel algorithm for solving DLPwAI (Cheon's algorithm). This paper reports our experimental results of Cheon's algorithm by implementing it with some speeding-up techniques. In fact, we have succeeded to solve DLPwAI on a pairing-friendly elliptic curve of 160-bit order in 1314 core days. Implications of our experiments on cryptographic schemes are also discussed.

Keywords: DLPwAI (DLP with Auxiliary Input), Barreto-Naehrig pairing-friendly elliptic curve, Cheon's algorithm.

1 Introduction

Let $\mathbb{G}$ be an additive cyclic group generated by an element G of prime order r. A discrete logarithm problem (DLP) is a problem to find α from G and αG. In the general setting, DLP is considered to be infeasible, and the infeasibility of DLP assures the security of some cryptographic schemes such as ECDH and ECDSA. When $\mathbb{G}$ is defined on elliptic curves over finite fields, the currently best algorithms for solving DLP require exponential time with regard to r, namely, $O(\sqrt{r})$. In fact, Shanks' baby-step giant-step (BSGS) method [20] requires $O(\sqrt{r})$ group operations in time and $O(\sqrt{r})$ group elements in space. On the other hand, Pollard's ρ-method also requires $O(\sqrt{r})$ in time, but much smaller elements in space. Since the state-of-the-art record of solving DLP on elliptic curves is 112-bit [7], 160-bit elliptic curves have been used as a secure parameter.

⋆ This research was done while she was a doctor student in Okayama University.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 595–608, 2012.

Table 1. Required time for solving DLPwAI

	$\log_2 r$ (in bit)	Required Time (by a single core)	Sub-algorithm
Jao, Yoshida [17]	60	3 hours	ρ-method
Izu, Takenaka, Yasuda [15,16]	83	14 hours	BSGS method
Sakemi et al. [21]	128	131 hours	BSGS method
Sakemi et al. [22]	128	136 hours	$\boldsymbol{\rho}$-method
This paper	**160**	**1314 days**	$\boldsymbol{\rho}$-method

At the beginning of 2000's, bilinear maps were introduced to establish efficient cryptographic schemes with new functions, whose security rely on the infeasibility of newly proposed mathematical problems such as Bilinear Diffie-Hellmann Problem (BDHP) [4], ℓ-Strong Diffie-Hellmann Problem (ℓ-SDHP) [2], ℓ-Bilinear Diffie-Hellmann Inversion Problem (ℓ-BDHIP) [1], ℓ-simplified Strong Diffie-Hellmann Problem (ℓ-sSDHP) [3], and ℓ-BDHEP [5]. In 2006, Cheon defined the discrete logarithm problem with auxiliary input (DLPwAI) as a generarization of some mathematical problems in the above [8]: find α from $G,\ \alpha G,\ \alpha^d G \in \mathbb{G}$ and a positive integer d satisfying $d|(r-1)$. Cheon also proposed a novel algorithm for solving DLPwAI [8,9]. The time complexity of Cheon's algorithm is $O\left(\sqrt{(r-1)/d} + \sqrt{d}\right)$, and especially when d can be chosen as $d \approx \sqrt{r}$, the complexity becomes $O(\sqrt[4]{r})$, which is more efficient than that for solving DLP in general groups (which requires $O(\sqrt{r})$). Thus, it is indispensable to evaluate the infeasibility of DLPwAI from implementational viewpoints in order to adopt cryptographic schemes based on such new mathematical problems in practice.

In this paper, we investigate useful techniques for speeding up Cheon's algorithm, and demonstrate that it is possible to solve 160-bit DLPwAI over a pairing-friendly elliptic curve within *a practical time*. Specifically, we clarify that Cheon's algorithm effectively works by using some accelerating techniques such as a precomputation table technique effective for scalar multiplications needed for the algorithm, the automorphism technique, and parallelization (see section 3 for details). In fact, we have successfully solved a DLPwAI in 25 days with about 160 cores (1314 days with a single core), which amounts USD 3,150 in Amazon EC2, in a group with 160-bit order defined on the pairing-friendly elliptic curve proposed by Barreto and Naehrig [6].

As far as the authors know, this is the largest result of solving DLPwAI by Cheon's algorithm (see Table 1). Note that solving DLP on this 160-bit elliptic curve is regarded to be infeasible. Our result implies that, if USD 1,000,000 is available, a DLPwAI on the 192-bit Barreto-Naehrig elliptic curve can be solved. Implications of our experimental results to the security of some cryptographic schemes are also discussed in this paper.

Algorithm 1. Cheon's Algorithm [8,9]

Require: : $G,\ G_1 = \alpha G,\ G_d = \alpha^d G \in \mathbb{G}$, d dividing $r-1$
Ensure: : $\alpha \in \mathbb{Z}/r\mathbb{Z}$
1: Find a generator $\zeta \in (\mathbb{Z}/r\mathbb{Z})^*$
2: Set $\zeta_d \leftarrow \zeta^d$
3: [Step 1] Find $0 \le k_1 < (r-1)/d$ such that $G_d = \zeta_d^{k_1} G$
4: Set $\zeta_e \leftarrow \zeta^{(r-1)/d}$, $G_e \leftarrow \zeta^{-k_1} G_1$
5: [Step 2] Find $0 \le k_2 < d$ such that $G_e = \zeta_e^{k_2} G$
6: Output $\zeta^{k_1 + k_2 (r-1)/d}$

2 Preliminaries

This section introduces Cheon's algorithm for solving DLPwAI [8,9] and ρ-method [19].

2.1 Cheon's Algorithm

Let $\mathbb{G} = \langle G \rangle$ be an additive cyclic group generated by an element G of prime order $r > 2$. The discrete logarithm problem with auxiliary input (DLPwAI) is a problem to find α on input $G,\ G_1 = \alpha G,\ G_d = \alpha^d G \in \mathbb{G}$ and an integer d dividing $r-1$. In 2006, Cheon proposed a novel algorithm for solving DLPwAI (Cheon's algorithm, [8,9]), which is the center topic of this paper. Cheon's algorithm requires $O\left(\sqrt{(r-1)/d} + \sqrt{d}\right)$ group operations in time. Especially, when $d \approx \sqrt{r}$, it only requires $O(\sqrt[4]{r})$ operations, which is much smaller than required in the baby-step giant-step (BSGS) method or in the ρ-method for solving DLP.

Let us briefly describe how Cheon's algorithm works. A goal of Cheon's algorithm is to find an integer $k \in \mathbb{Z}/r\mathbb{Z}$ such that $\alpha = \zeta^k$ for a generator ζ of the multiplicative group $(\mathbb{Z}/r\mathbb{Z})^*$ (Note that the generator ζ can be found efficiently). Here, such k is uniquely determined. In order to find k, Cheon's algorithm searches two integers $k_1,\ k_2$ such that $k = k_1 + k_2(r-1)/d$ satisfies $0 \le k_1 < (r-1)/d,\ 0 \le k_2 < d$ in two steps (see Algorithm 1). Step 1 searches an integer k_1 such that $G_d = \zeta_d^{k_1} G$, since k_1 satisfies $\alpha^d = \zeta_d^{k_1}$ for $\zeta_d = \zeta^d$. Similarly, Step 2 searches an integer k_2 such that $G_e = \zeta_e^{k_2} G$, since k_2 satisfies $\alpha = \zeta^{k_1} \zeta_e^{k_2}$ for $\zeta_e = \zeta^{(r-1)/d}$ and $G_e = \zeta^{-k_1} G_1$.

In Cheon's algorithm, searching k_1 (resp. k_2) in Step 1 (resp. Step 2) requires another sub-algorithm. Since these problems are very similar to DLP in the general setting, the baby-step giant-step method [20] or the ρ-method [19] can be used as a sub-algorithm. Since this paper is interested in Cheon's algorithm combined only with the ρ-method, we briefly describe its outline in the next subsection.

2.2 Pollard's ρ-Method

Pollard's ρ-method is one of the algorithms for solving DLP [19], which finds a solution α, from $G,\ \alpha G \in \mathbb{G}$ of prime order r, whose time complexity is $O(\sqrt{r})$

because of the birthday paradox. Let us describe the outline in the context of Cheon's algorithm. Especially, since Step 1 and Step 2 of Cheon's algorithm (Algorithm 1) are almost the same, we focus only on Step 1.

The idea of the ρ-method in Step 1 of Cheon's algorithm is to find a collision $F^{(i)}(G_d) = F^{(j)}(G)$ for a given function $F : \mathbb{G} \to \mathbb{G}$, where $F^{(i)}(P) = F(F^{(i-1)}(P))$ and $F^{(0)}(P) = P$. For an efficient evaluation, the function $F(P)$ is desired to be (i) random as possible, and (ii) of the form $F(P) = \zeta^{f(P)}P$ for some function f on $\mathbb{G}$. Such a function F is called a *random-walk function*. In our experiment, we use

$$F(P) : P \mapsto \zeta_d^{f_e(P)}P$$

with a pseudo-random function $f_e : \mathbb{G} \to \mathbb{Z}/e\mathbb{Z}$, where $e = (r-1)/d$ and $\zeta_d = \zeta^d$. By definition, we have

$$F^{(i)}(G_d) = \zeta_d^{\sum_{l=0}^{i-1} f_e(F^{(l)}(G_d))}G_d \quad \text{and} \quad F^{(j)}(G) = \zeta_d^{\sum_{l=0}^{j-1} f_e(F^{(l)}(G))}G.$$

Thus, one can find k_1 by computing

$$k_1 = \sum_{l=0}^{i-1} f_e(F^{(l)}(G_d)) - \sum_{l=0}^{j-1} f_e(F^{(l)}(G)) \bmod (r-1)/d$$

from a collision $F^{(i)}(G_d) = F^{(j)}(G)$. Since the image of the function F has $(r-1)/d$ elements, the time complexity of Step 1 is $O(\sqrt{(r-1)/d})$ (if the KKM method [18] is used, which will be described later).

In the ρ-method, the distinguished element technique [23] reduces the number of elements to be stored. An element which satisfies the specific condition (the least significant 6 bits of an element are zero, for example) is called a distinguished element. With this technique, one has to store elements $F^{(l)}(G_d)$ and $F^{(l)}(G)$ only when they are distinguished elements. Note that there exists a collision on the distinguished elements: in fact, for a collision $F^{(i)}(G_d) = F^{(j)}(G)$, we have $F^{(i+1)}(G_d) = F^{(j+1)}(G)$, $F^{(i+2)}(G_d) = F^{(j+2)}(G)$, $\cdots$, and thus, we eventually have a collision $F^{(i+w)}(G_d) = F^{(j+w)}(G)$ on the distinguished elements for an integer w. The space complexity (also the number of elements) can be reduced to $1/w$ with arbitrary parameter w, while the time complexity is increased to $O\left(\sqrt{(r-1)/d} + w\right)$. However, the increase can be neglected since $w \ll (r-1)/d$ in practice.

3 Implementation

This section describes our strategy for implementing Cheon's algorithm.

3.1 Evaluating $F(X)$

In Cheon's algorithm, the most computationally heavy operation is the evaluation of the function $F^{(l)}(P) = F(F^{(l-1)}(P))$, which consists of

1. Evaluate $f_e(F^{(l-1)}(P))$,
2. Compute $\zeta_d^{f_e(F^{(l-1)}(P))}$ as an exponentiation in $(\mathbb{Z}/r\mathbb{Z})^*$,
3. Compute $\zeta_d^{f_e(F^{(l-1)}(P))}P$ as a scalar multiplication in $\mathbb{G}$.

In our implementation, an element $P \in \mathbb{G}$ is represented by a pair of x-coordinate and y-coordinate, and we used the pseudo-random function $f_e(P) = x(P) \bmod e$, where $x(P)$ is the x-coordinate of P. Thus, procedure 1 is negligible compared to procedure 2 and 3.

Procedure 3 computes a scalar multiplication of a fixed element P independent from l, so that a precomputation table for scalar multiplications is significantly effective (KKM method, [18]). Let us describe the KKM method for a scalar multiplication δP ($\delta \in \mathbb{Z}/r\mathbb{Z}$, $P \in \mathbb{G}$). For a fixed integer c (which will be optimized later) and $n = \lceil \sqrt[c]{r} \rceil$, obtain the n-array expansion of the scalar $\delta = \sum_{l=0}^{c-1} \delta_l n^l$ ($0 \le \delta_l < n$). For all $0 \le l < c$ and $0 \le l' < n$, compute $S(l, l') = l' n^l P$ and store them in a table in advance to the scalar multiplications. Then, the scalar multiplication δP is computed by

$$\begin{aligned}\delta P &= \delta_0 P + \delta_1 nP + \cdots + \delta_{c-1} n^{c-1} P \\ &= S(0, \delta_0) + S(1, \delta_1) + \cdots + S(c-1, \delta_{c-1}).\end{aligned}$$

Note that the precomputation table can be computed by at most cn additions.

Similar to procedure 3, procedure 2 also computes an exponentiation of a fixed element ζ_d independent from l, so that the KKM method can be applied to procedure 2 in the same way.

3.2 Using Automorphisms

If there exists an efficiently computable automorphism $\phi : \mathbb{G} \to \mathbb{G}$ of order m on a group $\mathbb{G}$ satisfying the condition $\phi(P) = \zeta_d^s P$ for an integer s, the random-walk function $F(P) : \mathbb{G} \to \mathbb{G}$ can be extended to the random-walk function $\tilde{F}(P) : \mathbb{G}/\sim_\phi \to \mathbb{G}/\sim_\phi$ on the set $\mathbb{G}/\sim_\phi$ of the equivalence classes. Here, two elements P, Q are in the same equivalence class if and only if there exists an integer l such that $P = \phi^{(l)}(Q)$ ($0 \le l < m$). Since the number of elements in $\mathbb{G}/\sim_\phi$ is reduced to $1/m$, the ρ-method can be sped-up by a factor of $\sqrt{m}$.

In our experiment, the pairing-friendly elliptic curve introduced by Barreto-Naehrig (BN curve, [6]) is used. The BN curve is an elliptic curve $y^2 = x^3 + b$ ($b \in \mathbb{F}_p$) defined over a prime field $\mathbb{F}_p$ satisfying $3|(p-1)$. On the BN curve, there exist the negation map ($P \mapsto -P$) which is an automorphism of order 2, and, in addition, the automorphism of order 3 [13]. For an element $P = (x, y) \in \mathbb{G}$, the map $\phi_3(P) = (\epsilon x, y) = \gamma P$ is an automorphism of order 3, where ϵ is a fixed primitive cube root of a unity in $\mathbb{F}_p$ and $\gamma \in (\mathbb{Z}/r\mathbb{Z})^*$ satisfies $\gamma^2 + \gamma + 1 \equiv 0 \bmod r$, i.e. γ is a primitive cube root of unity in $(\mathbb{Z}/r\mathbb{Z})^*$. Such automorphism ϕ_3 can be computed with one multiplication in $\mathbb{F}_p$ only.

Let us consider when these automorphisms satisfy the condition $\phi(P) = \zeta_d^s P$ on the BN curve.

– Negation map: Since $-1 = \zeta^{(r-1)/2} \in (\mathbb{Z}/r\mathbb{Z})^*$, the negation map satisfies the condition if $2d|(r-1)$. The ρ-method can be sped-up by $\sqrt{2}$ with the negation map.
– Automorphism ϕ_3: Since γ is a primitive cubic root of a unity in $(\mathbb{Z}/r\mathbb{Z})^*$, γ can be represented by $\gamma = \zeta^{(r-1)/3}$. Thus, the automorphism satisfies the condition if $3d|(r-1)$. The ρ-method can be sped-up by $\sqrt{3}$ with the automorphism ϕ_3.

As a result of the above analysis, the time complexity of Cheon's algorithm can be reduced to $\tilde{T} = O\left(\sqrt{e/\gcd(d,6)} + \sqrt{d/\gcd(e,6)}\right)$ by using the automorphism technique.

For a random-walk function $\tilde{F}$ of additive type such as Teske's adding walk [24] or the function proposed by [12], the function $\tilde{F}$ on $\mathbb{G}/\sim_\phi$ can fall into short cycles, which are called "fruitless cycles", and hence the optimal speed-up cannot be expected in general [11,12]. However, since our random-walk function is of multiplicative type, our function on $\mathbb{G}/\sim_\phi$ rarely falls into fruitless cycles. Therefore, using both the negation map and the automorphism ϕ_3, the time complexity of the algorithm can be reduced to $\tilde{T} = O\left(\sqrt{e/\gcd(d,6)} + \sqrt{d/\gcd(e,6)}\right)$.

When the above automorphism technique is used, all elements have to be converted to the representative elements of equivalence classes. In our implementation, the representative element is the smallest element when a concatenation $x(P)||y(P)$ is regarded as an integer. Since there are at most 6 elements in an equivalence class, and x-coordinates of a half coincide with those of another half, only one multiplication in $\mathbb{F}_p$ is enough to compute the representative element.

3.3 Parallelization

The ρ-method can be sped-up by parallelization. However, in order to make paths different, initial elements are randomized in the following way [9]: when a core computes $F^{(l)}(G_d)$, $F^{(l)}(G)$ for Step 1, two random integers $c_\mathrm{L}, c_\mathrm{R}$ are assigned to this core and initial points are converted to $G'_d = \zeta_d^{c_\mathrm{L}} G_d$ and $G' = \zeta_d^{c_\mathrm{R}} G$. Then, one can find k_1 by computing

$$k_1 = \left(\sum_{l=0}^{i-1} f_e(F^{(l)}(G'_d)) + c_\mathrm{L}\right) - \left(\sum_{l=0}^{j-1} f_e(F^{(l)}(G') + c_\mathrm{R}\right) \bmod (r-1)/d$$

from a collision $F^{(i)}(G'_d) = F^{(j)}(G')$. Note that since all converted initial points can be regarded as scalar multiple point of G, G_d, or G_e, the KKM method can be applied to the conversion.

In our experiment, we also developed a management system for parallelized ρ-method. Outline of the system is described in the appendix.

4 Experimental Results

This section reports our experimental results of Cheon's algorithm for the pairing-friendly elliptic curve with 160-bit order. We have successfully solved a DLPwAI

on this curve in 1314 core days. We emphasize that DLP on the same elliptic curve has been believed to be secure.

4.1 Parameters

We used an additive cyclic group $\mathbb{G}$ with order r on the pairing-friendly elliptic curve $E : y^2 = x^3 + 3$ over a prime field $\mathbb{F}_p$ introduced by Barreto-Naehrig [6]. Concrete values of these parameters are summarized in the following:

$$\begin{aligned} p &= 1461501624496790265145448589920785493717258890819 \text{ (160-bit)} \\ \#\mathbb{G} &= 1461501624496790265145447380994971188499300027613 \text{ (160-bit, prime)} \\ r &= 1461501624496790265145447380994971188499300027613 \text{ (160-bit)} \\ r - 1 &= 2^2 \cdot 3 \cdot 12132793 \cdot 164442871007 \cdot 448873741399 \cdot 135993458106516349 \end{aligned}$$

where $\#\mathbb{G}$ denotes the number of elements in the additive group $\mathbb{G} = E(\mathbb{F}_p)$. In our implementation, we used the following parameters:

$$\begin{aligned} d &= 2 \cdot 3 \cdot 12132793 \cdot 135993458106516349 \text{ (84-bit)} \\ e &= (r-1)/d = 2 \cdot 164442871007 \cdot 448873741399 \text{ (77-bit)} \\ \zeta &= 2 \end{aligned}$$

where the generator $\zeta \in (\mathbb{Z}/r\mathbb{Z})^*$ was selected as the smallest one. With these parameters, Step 1 can be sped-up by $\sqrt{2}$, and Step 2 can be sped-up by $\sqrt{6}$ with the automorphism technique, and the estimated time complexity is $2^{40.5}$.

We selected the solution α as 49 decimal places of the circle ratio π:

$$\alpha = 1415926535897932384626433832795028841971693993751 \text{ (160-bit)}$$

We used a base point G whose x-coordinate coincides 48 decimal places of the Napier's constant. Then, coordinates of G, $G_1 = \alpha G$, $G_d = \alpha^d G$ are as follows:

$$\begin{aligned} x(G) &= 718281828459045235360287471352662497757247093699 \\ y(G) &= 267920135876087743710291823125072055976344820822 \\ x(G_1) &= 673981942030616258426617938323441969041367773762 \\ y(G_1) &= 1145655312172916339251351940414297415585122330072 \\ x(G_d) &= 113217660152885721186991580289363094493274367616 2 \\ y(G_d) &= 948528425611362859774760991656937949436755965122 \end{aligned}$$

With these parameters, we have optimized $n = 2^{20}, c = 8$ for the KKM method. With this optimization, one F-evaluation requires 24 μseconds (on Intel core i7 2.93GHz) while 980 μseconds without KKM.

4.2 Results

This section reports experimental results for solving DLPwAI. Required resources are summarized in Table 2.

Table 2. Required resource for solving a 160-bit DLPwAI

CPU (Hz)	# of PCs	# of cores	Time
Step 1			
Q9450 (2.66GHz)	8	32	7 days
Step 2			
Q9450 (2.66GHz)	8	32	18 days
Q9450 (3.00GHz)	8	32	13 days
X3460 (2.80GHz)	10	80*	1 day
Pentium D (3.40GHz)	9	18	1 day

*Hyper-Threading is used

Step 1. Step 1 required 7 days with 8 PCs (Intel Core2 Quad CPU Q9450 2.66GHz), namely 32 cores: 16 cores are used for computing $F^{(l)}(G'_d)$ while other 16 cores are for computing $F^{(l)}(G')$. The required storage for distinguished elements was 53.3 MByte in total. Obtained partial solution was $k_1 = 108516124982482634887141$.

Step 2. Step 2 was estimated to require 4.7 times more cores compared to Step 1. Thus, we used many PCs with different specifications as in Table 2. Thanks to the flexibility of our management system for the parallelization, PCs are invested one-by-one (see Figure 1). In total, Step 2 required 18 days with 35 PCs (162 cores), more precisely 1090 core days. The required storage for distinguished elements was 256 MByte in total. Obtained partial solution was $k_2 = 60161665500021502744798 50$ and k is obtained by

$$\begin{aligned} k &= k_1 + k_2(r-1)/d \\ &= 888155679312448193339542847931449754121424529241. \end{aligned}$$

Consequently, the final solution α is obtained by

$$\begin{aligned} \alpha &= \zeta^k \bmod r \\ &= 141592653589793238462643383279502884197169399375 1, \end{aligned}$$

which required 1314 core days in total.

4.3 Discussion

Let us estimate the required monetary cost of our experiment on Amazon Elastic Compute Cloud (Amazon EC2), a service to provide resizable computing environment in the cloud. In Amazon EC2, various instances corresponding to CPU power, memory size, and storage size are available. For our experiments, high-spec CPUs and large memory (for the KKM method) are required. Thus, the high-CPU extra-large instance (Memory: 7 GB, Cores: 8 (in virtual)) is adapted, which requires USD 0.1 per hour per 1 virtual core. Since our experiment required 1314 core days ($T = 2^{40.5}$), it is estimated to cost USD 3,150 in Amazon

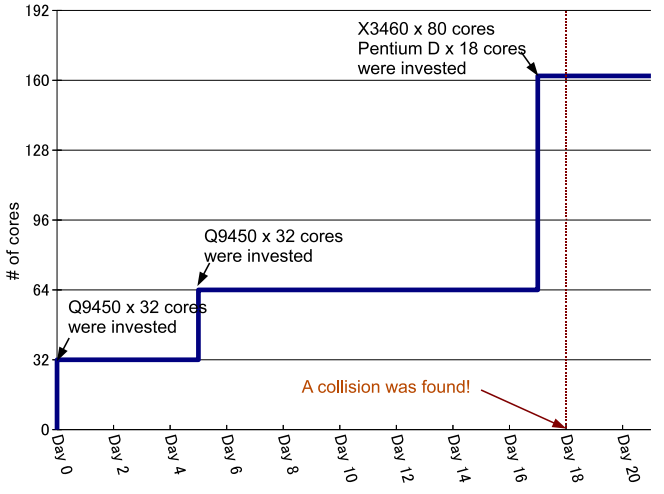


Fig. 1. One-by-one investment in Step 2

EC2. If USD 1,000,000 is available, it is estimated to use 320 times more PCs ($T = 2^{48.3}$) than the experiment. With this environment and if the parameter d can be selected as $d \approx \sqrt[4]{r}$, it is possible to solve a DLPwAI on an elliptic curve with 193.2-bit order.

On the other hand, if USD 1,000,000 is available for the same parameters (namely, a group with 160-bit order), the parameter d can be reduced to $d = 2^{64}$, while d was optimized as $d \approx \sqrt[4]{r}$ in our experiment. Effects of this reduction will be discussed in the next section.

5 Feedback to Cryptographic Schemes

In recently proposed cryptographic schemes, the infeasibility of new mathematical problems are assumed. For example, ℓ-BDHEP is used in Boneh, Gentry, and Waters' broadcast encryption system [5], where ℓ-BDHEP is the problem to find $e(G, \hat{G})^{\alpha^{\ell+1}}$ for a given bilinear map $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ on input $G,\ \alpha G, \ldots, \alpha^{\ell} G,\ \alpha^{\ell+2} G, \ldots,\ \alpha^{2\ell} G \in \mathbb{G}$ and $\hat{G} \in \hat{\mathbb{G}}$, where $\mathbb{G} = \langle G \rangle$, $\hat{\mathbb{G}} = \langle \hat{G} \rangle$, and $\mathbb{G}_T$ is a multiplicative group with order r. Let d be the largest divisor of $(r-1)$ among $2, 3, \ldots, \ell, \ell+2, \ldots, 2\ell$. As shown in section 4.3, if the parameter d can be selected as $d \approx 2^{80}$ and a 160-bit elliptic curve is used, Cheon's algorithm can solve a DLPwAI. In addition, if USD 1,000,000 is available, the parameter d can be reduced to $d = 2^{64}$ with a 160-bit elliptic curve. Therefore, if the parameter ℓ is chosen to be larger than 2^{80} (or 2^{64}), Cheon's algorithm can solve the ℓ-BDHEP and thus break the scheme: by finding α as a DLPwAI, a solution of ℓ-BDHEP is obtained. Thus, when such cryptographic schemes are

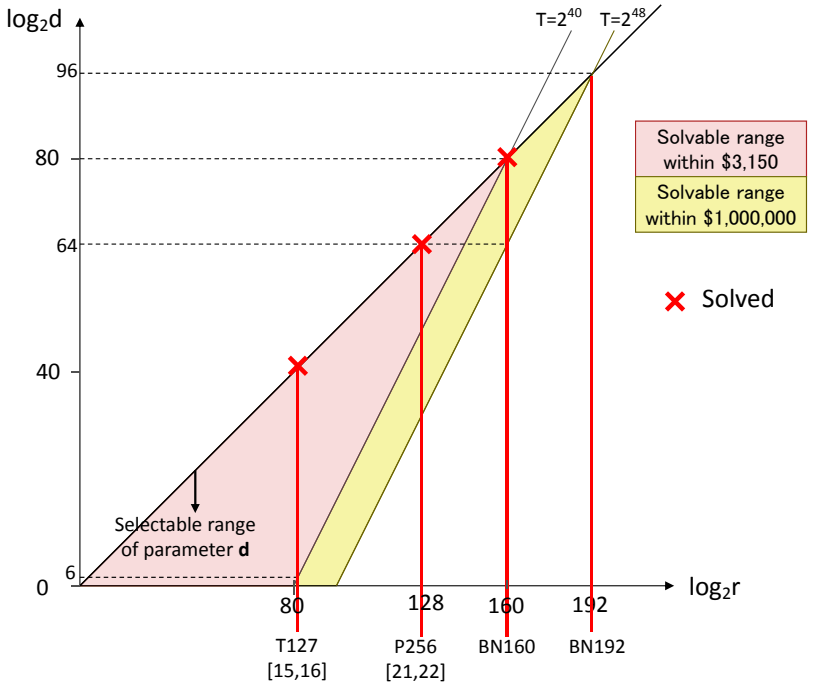


Fig. 2. Selectable range of d

implemented with a 160-bit elliptic curve, the parameter ℓ should be smaller than 2^{80} (or 2^{64}).

In this section, we discuss feedbacks of our experiments on a 160-bit elliptic curve to some cryptographic schemes including Boneh, Gentry, and Waters' broadcast encryption scheme [5], Boneh and Boyen's ID-based encryption [1], and Boneh and Boyen's signature scheme [2].

5.1 Boneh, Gentry, and Waters' Broadcast Encryption Scheme

Boneh, Gentry, and Waters' broadcast encryption scheme is provably secure under an assumption that ℓ-BDHEP is infeasible [5], where ℓ is the number of users (receivers) in the broadcast encryption scheme. In the special construction, the sender publishes his public key as

$$\mathsf{pk} = (G, \alpha G, \ldots, \alpha^{\ell} G, \alpha^{\ell+2} G, \ldots, \alpha^{2\ell} G, \gamma G) \in \mathbb{G}^{2\ell+1}$$

where $\gamma \in \mathbb{Z}/r\mathbb{Z}$ is a random number. Thus, when the broadcast encryption scheme is implemented with a 160-bit elliptic curve, ℓ should be chosen smaller than 2^{80} (or 2^{64}) to avoid Cheon's algorithm for DLPwAI.

However, restricting $\ell < 2^{80} \approx 10^{24}$ has almost no effect on the scheme in practice since 10^{24} is far beyond the population on the earth. Even if USD 1,000,000,000 is available, ℓ can be chosen as $2^{64} \approx 10^{19.2}$ so that the restriction has little effect.

5.2 Boneh and Boyen's ID-Based Encryption Scheme

Boneh and Boyen's ID-based encryption scheme is proved to be IND-sID-CCA secure under an assumption that ℓ-BDHIP is infeasible [1], where ℓ is the number of queries to the key generation algorithm. Here, ℓ-BDHIP is a problem to find $e(G,G)^{1/\alpha} \in \mathbb{G}_T$ on input $G,\ \alpha G, \ldots, \alpha^{\ell} G \in \mathbb{G}$. Thus, when the ID-based encryption scheme is implemented with a 160-bit elliptic curve, ℓ should be smaller than 2^{80} (or 2^{64}) to avoid Cheon's algorithm for DLPwAI. In the ID-based encryption scheme, queries to the key generation algorithm will be online so that such queries are almost impossible for adversaries. Note that the same discussion can be applied to some ID-based encryption schemes [3,14].

5.3 Boneh and Boyen's Signature Scheme

Boneh and Boyen's signature scheme is provable secure under the assumption that ℓ-SDHP is infeasible [2] (moreover, it is proven that the infeasibility and the unforgeability is equivalent [17]), where ℓ is the number of queries to the signing algorithm. Here, ℓ-SDHP is the problem to find a pair $(a, \frac{1}{a+\alpha} G) \in \mathbb{Z}/r\mathbb{Z} \times \mathbb{G}$ on input $G,\ \alpha G, \ldots, \alpha^{\ell} G \in \mathbb{G}$ and $\hat{G} \in \hat{\mathbb{G}}$. Thus, when the signature scheme is implemented with a 160-bit elliptic curve, ℓ should be smaller than 2^{80} (or 2^{64}) to avoid Cheon's algorithm for DLPwAI. The effect of this restriction depends on how the signing algorithm is implemented. If it is implemented online similar to Boneh and Boyen's ID-based encryption scheme, this restriction has almost no effect. However, if the query to the signing algorithm can be offline (for example, the case where the signing algorithm is implemented in IC chip), more queries will be available compared to the online case. Thus, this case is the most attackable for adversaries with Cheon's algorithm.

6 Concluding Remarks

This paper successfully solved a discrete logarithm problem with auxiliary input (DLPwAI) in 1314 core days over a 160-bit pairing-friendly elliptic curve. If cryptographic schemes based on mathematical problems such as ℓ-BDEP, ℓ-SDHP, ℓ-sSDHP, or ℓ-BDHIP are implemented, such weak parameters should be avoided. However, there are pairing-based cryptographic schemes which are not affected by Cheon's algorithm such as Boneh and Franklin's ID-based encryption scheme.

Acknowledgements. We wish to thank anonymous reviewers for their helpful comments and suggestions. We also thank Professor Yoshitaka Morikawa, Professor Yasuyuki Nogami, and Naoki Takahashi for their generous support.

References

1. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
2. Boneh, D., Boyen, X.: Short Signatures Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004)
3. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical Identity Based Encryption with Constant Size Ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
4. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
5. Boneh, D., Gentry, C., Waters, B.: Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005)
6. Barreto, P.S.L.M., Naehrig, M.: Pairing-Friendly Elliptic Curves of Prime Order. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 319–331. Springer, Heidelberg (2006)
7. Bos, J.W., Kaihara, M.E., Kleinjung, T., Lenstra, A.K., Montgomery, P.L.: PlayStation 3 Computing Breaks 2^{60} Barrier 112-bit Prime ECDLP Solved (2009), http://lacal.epfl.ch/112bit_prime
8. Cheon, J.H.: Security Analysis of the Strong Diffie-Hellman Problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 1–11. Springer, Heidelberg (2006)
9. Cheon, J.H.: Discrete Logarithm Problems with Auxiliary Inputs. Journal of Cryptology 23(3), 457–476 (2010)
10. Distributed.net, http://www.distributed.net/Main_Page/
11. Galbraith, S.D., Ruprai, R.S.: Using Equivalence Classes to Accelerate Solving the Discrete Logarithm Problem in a Short Interval. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 368–383. Springer, Heidelberg (2010)
12. Gallant, R.P., Lambert, R.J., Vanstone, S.A.: Improving the Parallelized Pollard Lambda Search on Binary Anomalous Curves. Math. Comp. 69, 1699–1705 (2000)
13. Gallant, R.P., Lambert, R.J., Vanstone, S.A.: Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 190–200. Springer, Heidelberg (2001)
14. Gentry, C.: Practical Identity-Based Encryption Without Random Oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
15. Izu, T., Takenaka, M., Yasuda, M.: Experimental Results on Cheon's Algorithm. In: WAIS 2010, Proceedings of ARES 2010, pp. 625–630. IEEE Computer Science (2010)
16. Izu, T., Takenaka, M., Yasuda, M.: Experimental Analysis of Cheon's Algorithm against Pairing-friendly Curves. Journal of Information Processing 19, 441–450 (2011)
17. Jao, D., Yoshida, K.: Boneh-Boyen Signatures and the Strong Diffie-Hellman Problem. In: Shacham, H., Waters, B. (eds.) Pairing 2009. LNCS, vol. 5671, pp. 1–16. Springer, Heidelberg (2009)
18. Kozaki, S., Kutsuma, T., Matsuo, K.: Remarks on Cheon's Algorithms for Pairing-Related Problems. In: Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T. (eds.) Pairing 2007. LNCS, vol. 4575, pp. 302–316. Springer, Heidelberg (2007)

19. Pollard, J.: Monte Carlo Methods for Index Computation (mod p). Math. Comp. 32, 918–924 (1978)
20. Shanks, D.: Class Number, a Theory of Factorization, and Genera. In: Proc. of Symp. Math. Soc., vol. 20, pp. 41–440 (1971)
21. Sakemi, Y., Izu, T., Takenaka, M., Yasuda, M.: Solving DLP with Auxiliary Input over an Elliptic Curve Used in TinyTate Library. In: Ardagna, C.A., Zhou, J. (eds.) WISTP 2011. LNCS, vol. 6633, pp. 116–127. Springer, Heidelberg (2011)
22. Sakemi, Y., Izu, T., Takenaka, M., Yasuda, M.: Solving a DLP with Auxiliary Input with the ρ-Algorithm. In: Jung, S., Yung, M. (eds.) WISA 2011. LNCS, vol. 7115, pp. 98–108. Springer, Heidelberg (2012)
23. Teske, E.: Speeding Up Pollard's Rho Method for Computing Discrete Logarithms. In: Buhler, J.P. (ed.) ANTS III. LNCS, vol. 1423, pp. 541–554. Springer, Heidelberg (1998)
24. Teske, E.: On Random Walks for Pollard's Rho Method. Math. Comp. 70, 809–825 (2001)

A Large Scale Solving System

This appendix describes the management system "Large Scale Solving System (LSSS)" for the parallelized ρ-method for Cheon's algorithm dedicated to the large-scale experiment.

For such a large-scale parallelized experiment, the distributed.net is used worldwide [10], which supports to solve large scale problems using idle PCs, CPUs or GPUs in everywhere in the world. For example, the distributed.net has broken RC5-64 (64-bit RC5), and is trying to break RC5-72 (72-bit RC5) currently. Anyone can join to the distributed.net simply by downloading and executing a client program. A server of the distributed.net system distributes a "key-block" to each client, and each client exhaustively searches the correct key. In the distributed.net system, the HTTP protocol over proxy-server is used for the communication between the server and clients.

the distributed.net has high scalability and is suitable for large-scale experiment. However, since the connection between the server and clients in the distributed.bet is loose, the system is not efficient. Thus, we have designed more tightly-connected and more efficient but less scalable system in our experiment. An overall design of LSSS is shown in Figure 3. We have also adopted the HTTP protocol over proxy-server for the communication between the server and clients so that any clients of any organizations can join to LSSS at any time (this is very important when the experiments are conducted in academic organizations and private companies).

For solving a 160-bit DLPwAI by Cheon's algorithm in our experiment, two LSSSs are used as in Figure 4. Each LSSS consists of one server, numerous calculating clients, and one DB organizer. A calculating client evaluates the random-walk function and outputs a result if it is the distinguished element. Every client sends distinguished elements to the server and the server catches the received distinguished elements. A DB organizer obtains the distinguished elements from the server and establishes a DB of these distinguished elements. One LSSS is dedicated to evaluate $F^{(l)}(G'_d)$, while another LSSS is to evaluate

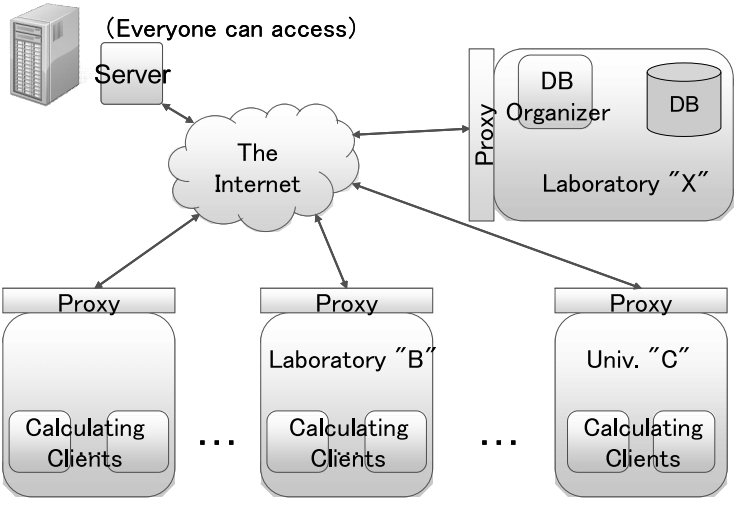


Fig. 3. Grand design of LSSS

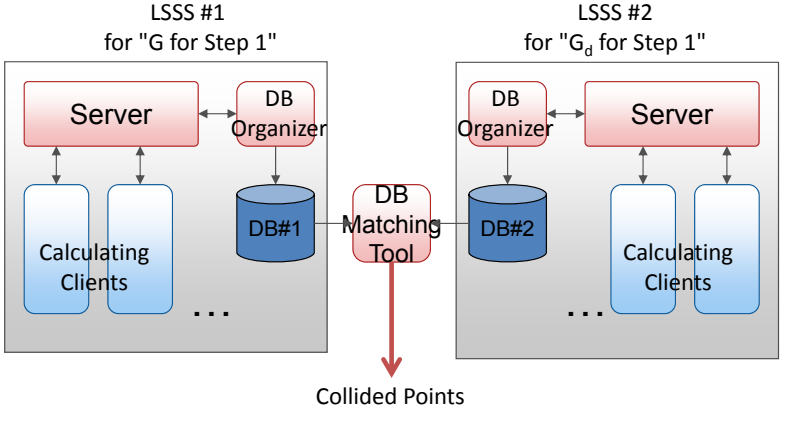


Fig. 4. Constitution of the solving system of a 160-bit DLPwAI by Cheon's algorithm with two LSSSs

$F^{(l)}(G')$ for Step 1 of Cheon's algorithm. These two DBs are compared by a DB matching tool periodically. If a collision $F^{(i)}(G'_d) = F^{(j)}(G')$ is found in these DBs, the tool output the collision. In LSSS, all functions work on Windows and Linux (and perhaps other UNIX OSs) to utilize any platforms.

A calculating client has the common communication unit and the calculating unit. Since the common communication unit is independent from the target parameter, and APIs between the communication unit and the calculating unit is very simple, a user has to change the calculating unit only for a new experiment. Because of this construction, LSSS can be used not only for Cheon's algorithm but also for solving ECDLP with ρ-method and other similar problems.

Inferring Sequences Produced by Nonlinear Pseudorandom Number Generators Using Coppersmith's Methods

Aurélie Bauer[1], Damien Vergnaud[2,⋆], and Jean-Christophe Zapalowicz[3,⋆⋆]

[1] Agence Nationale de la Sécurité des Systèmes d'Information
51 Boulevard de la Tour-Maubourg - 75700 Paris 07 SP, France
aurelie.bauer@ssi.gouv.fr
[2] École Normale Supérieure – C.N.R.S. – I.N.R.I.A.
45, rue d'Ulm, F-75230 Paris Cedex 05, France
[3] INRIA Rennes – Bretagne Atlantique
Campus de Beaulieu, 35042, Rennes, France
jean-christophe.zapalowicz@inria.fr

Abstract. *Number-theoretic* pseudorandom generators work by iterating an algebraic map F (public or private) over a residue ring $\mathbb{Z}_N$ on a secret random initial seed value $v_0 \in \mathbb{Z}_N$ to compute values $v_{n+1} = F(v_n) \bmod N$ for $n \in \mathbb{N}$. They output some consecutive bits of the state value v_n at each iteration and their efficiency and security are thus strongly related to the number of output bits. In 2005, Blackburn, Gomez-Perez, Gutierrez and Shparlinski proposed a deep analysis on the security of such generators. In this paper, we revisit the security of number-theoretic generators by proposing better attacks based on Coppersmith's techniques for finding small roots on polynomial equations. Using intricate constructions, we are able to significantly improve the security bounds obtained by Blackburn *et al.*.

Keywords: Nonlinear Pseudorandom number generators, Euclidean lattice, LLL algorithm, Coppersmith's techniques, Unravelled linearization.

1 Introduction

This paper aims to present new cryptanalytic results on some nonlinear number-theoretic pseudorandom number generators. We show that several generators are insecure if sufficiently many bits are output at each clocking cycle. In particular, this provides an upper bound on the generators' security. The attacks used the well-known Coppersmith methods for finding small roots on polynomial equations and outperform previously known results [2,3,4,10,11].

Prior work. One of the most fundamental cryptographic primitives is the *pseudorandom bit generator*. It is a deterministic algorithm that expands a few truly

⋆ This author was supported in part by the European Commission through the ICT Program under contract ICT-2007-216676 ECRYPT II.
⋆⋆ Work done while at Agence Nationale de la Sécurité des Systèmes d'Information.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 609–626, 2012.

random bits to a longer sequence of bits that cannot be distinguished from uniformly random bits by a computationally bounded algorithm. It has numerous uses in cryptography, e.g. in signature schemes or public-key encryption schemes.

Number-theoretic pseudorandom generators work by iterating an algebraic map F (public or private) over a residue ring $\mathbb{Z}_N$ on a secret random initial seed value $v_0 \in \mathbb{Z}_N$ to compute the intermediate state values $v_{i+1} = F(v_i) \bmod N$ for $i \in \mathbb{N}$ and outputting (some consecutive bits of) the state value v_i at each iteration. The input v_0 of the generator (and possibly the description of F) is called the *seed* and the output is called the *pseudorandom sequence*. The case where F is an affine function is known as the *linear congruential generator*. This generator is efficient and has good statistical properties. Unfortunately, it is cryptographically insecure: Boyar [7] proved that - with a sufficiently long run of the pseudorandom sequence - one can recover the seed in time polynomial in the bit-size of N and Stern [17] proved that this is also the case even if one outputs only the most significant bits of each v_i (see also [6,15]).

It was suggested to use a non-linear algebraic map F in order to avoid these attacks but several works [2,3,4,10,11] showed that not too many bits can be output at each stage. Blackburn, Gomez-Perez, Gutierrez and Shparlinski [3,4] proved that some generators are polynomial time predictable if sufficiently many bits of some consecutive values of the pseudorandom sequence are revealed (even when F is kept private).

Blackburn et al.'s results are based on a lattice basis reduction attack, using a certain linearization technique. A natural idea – already stated in [3] – is instead of using only linear relations in the attack, to use also relations that are derived by taking products of them. This technique was proposed by Coppersmith to find small roots on polynomial equations [8,9]. In Coppersmith's method, a family of polynomials is first derived from the polynomial whose root is wanted. This family naturally gives a lattice basis and short vectors of this lattice possibly provide the wanted root. Blackburn *et al.* claimed that "this approach does not seem to provide any advantages" and that "it may be very hard to give any precise rigorous or even convincing heuristic analysis of this approach". Our goal in this paper is to investigate this issue.

Our contributions. We show that if a sufficient number of the most significant bits of several consecutive values v_i of non-linear algebraic pseudorandom generator are given, one can recover the seed v_0 (even in the case where the coefficients of F are unknown). We tackle these issues with Coppermith's lattice-based technique for calculating the small roots of multivariate polynomials modulo an integer. This method is heuristic, which is also the case of some arguments of Blackburn *et al.* showing that their basic results could be strengthened if the number of pseudorandom bits known to the attacker is increased. If F is a polynomial of degree d known to the attacker, Blackburn *et al.*'s result [4] proved that the generator can be predicted if one outputs a proportion $(d^2 - 1)/d^2$ of the most significant bits of two consecutive intermediate state values. We improve this result (*cf.* Section 3) by showing that this is also the case if one outputs a proportion as large as

$d/(d+1)$ of the most significant bits of two consecutive intermediate state values (or $(d-1)/d$ for sufficiently many consecutive intermediate state values).

Blackburn *et al.* [2,3] then focused on the well-known following number-theoretic pseudorandom generators (where p is a prime, $a \in \mathbb{Z}_p^*$ and $b \in \mathbb{Z}_p$):

- The *Quadratic generator* corresponding to the map $F(x) = ax^2 + b \bmod p$
- The *Pollard generator*, a special case of the quadratic generator when $a = 1$
- The *Inversive generator* corresponding to the map $F(x) = ax^{-1} + b \bmod p$

Our generic results apply to these settings and improve the previous bounds. The theoretical data complexity (*i.e.* the minimum keystream length) of our attack is decreased compared to the attack from [2,3,4,10,11]. Therefore a secure use of these generators requires the output of much fewer bits at each iteration and the efficiency of the schemes is thus degraded.

The table below shows a comparison between our results and what is known in the literature. It gives the proportion of most significant bits output from each consecutive state values necessary to break the generator in (heuristic) polynomial time. The *basic proportion* corresponds to the case where the adversary knows bits coming from the minimum number of intermediate states leading to a feasible attack; while the *asymptotic proportion* corresponds to the case when the bits known by the adversary come from an infinite number of values.

		Basic proportion		**Asymptotic proportion**	
		Prior result	**Our result**	**Prior result**	**Our result**
Quadratic	a,b known	**3/4**	**2/3**	**2/3**	**1/2**
generator	a,b unknown	**18/19**	**11/12**	**11/12**	**2/3**
Pollard	b known	**9/14**	**3/5**	**9/14**	**1/2**
generator	b unknown	**3/4**	**5/7**	**2/3**	**3/5**
Inversive	a,b known	**3/4**	**2/3**	**2/3**	**1/2**
generator	a,b unknown	**14/15**	**11/12**	**11/12**	**2/3**

The results on the quadratic generator (and the inversive generator) are described in Section 3.3 (and Section 3.4) and are direct applications of our general results. Those on the Pollard generator relies on the *unravelled linearization* technique introduced by Hermann and May in 2009 [12] and are described in Section 4.

2 Preliminaries

2.1 Lattices

Definition. If $(b_1, \ldots, b_d)$ are d linearly independent vectors over $\mathbb{Z}^n$, then the lattice $\mathcal{L} = \langle b_1, \ldots, b_d \rangle$ generated by these vectors is defined as the set of all integer linear combination of the b_i's. The set $B = \{b_1, \ldots, b_d\}$ is called a *basis* of $\mathcal{L}$ and d is the *dimension* of $\mathcal{L}$. We restrict ourselves to *full-rank* lattices corresponding to the particular case $d = n$. The quantity $|\det(B)|$ is called the *determinant* of the lattice $\mathcal{L}$.

LLL-reduced bases. In 1982, Lenstra, Lenstra and Lovász [16] defined *LLL-reduced* bases of lattices and presented a deterministic polynomial-time algorithm, called *LLL* to compute such a basis. If $(b_1, \ldots, b_n)$ is an LLL-reduced basis of $\mathcal{L}$, the first vector b_1 is close to be the shortest non-zero vector of the lattice. Moreover, if $(b_1^\star, \ldots, b_n^\star)$ are the corresponding vectors coming from Gram-Schmidt's orthogonalization, then:

$$\|b_n^\star\|_2 \geq 2^{-(n-1)/4} (\det \mathcal{L})^{1/n} \tag{1}$$

2.2 Coppersmith's Techniques

In 1996, Coppersmith introduced lattice-based techniques [8,9] for finding small roots on univariate and bivariate polynomial equations. As these techniques had a wide range of cryptanalytic applications, some reformulations and generalizations to more variables have been proposed [1,5,13,14].

All these methods have allowed to attack many instances of public-key cryptosystems (*e.g.* [12,15]). In the following, we give more details explaining how such techniques work in practice for the multivariate modular case.

Definition of the Problem. Let $f(y_1, \ldots, y_n)$ be an irreducible multivariate polynomial defined over $\mathbb{Z}$, having a root $(x_1, \ldots, x_n)$ modulo a known integer N such that $|x_1| < X_1, \ldots, |x_n| < X_n$. The question is to determine the bounds X_i allowing to recover the desired root in polynomial time.

Collection of Polynomials. One has to generate a collection of polynomials $f_1, \ldots, f_r$ having $(x_1, \ldots, x_n)$ as a modular root. Usually, we consider multiples and powers of the polynomial f, namely $f_\ell = y_1^{\alpha_1^{(\ell)}} \ldots y_n^{\alpha_n^{(\ell)}} f^{k_\ell}$, for ℓ in $\{1, \ldots, r\}$. By definition, such polynomials satisfy the relation $f_\ell(x_1, \ldots, x_n) \equiv 0 \mod N^{k_\ell}$, *i.e.* there exists an integer c_ℓ such that $f_\ell(x_1, \ldots, x_n) = c_\ell N^{k_\ell}$. From now, let us denote as M the set of monomials appearing in the collection $\{f_1, \ldots, f_r\}$. We then construct a matrix $\mathcal{M}$ by extracting the polynomial coefficients as follows:

$$\mathcal{M} = \left(\begin{array}{c|c} \begin{array}{c} 1 \\ X_1^{-1} \\ \ddots \\ X_1^{-a_1} \ldots X_n^{-a_n} \\ \hline 0 \end{array} & \begin{array}{c} f_1 \ \ldots \ f_r \\ \downarrow \ \ \downarrow \ \ \downarrow \\ \\ \\ \\ \hline N^{k_1} \\ \ddots \\ N^{k_r} \end{array} \end{array}\right) \begin{array}{c} 1 \\ y_1 \\ \vdots \\ y_1^{a_1} \ldots y_n^{a_n} \end{array}$$

Every row of the upper part is related to one monomial of the set M. The left-hand side contains the bounds corresponding to these monomials (*e.g.* the coefficient $X_1^{-1} X_2^{-2}$ is put in the row related to the monomial $y_1 y_2^2$). Each column of the right-hand side contains a vector coming from the initial collection $\{f_1, \ldots, f_r\}$. We define as $\mathcal{L}$ the lattice generated by $\mathcal{M}$'s rows and we have:

$$|\det(\mathcal{L})| = \frac{N^{k_1 + \cdots + k_r}}{\prod_{(y_1^{a_1} \ldots y_n^{a_n} \in M)} X_1^{a_1} \ldots X_n^{a_n}}.$$

A Short Vector in the Lattice $\mathcal{L}$. Let us consider the vectors r_0 and s_0 defined by $r_0 = (1, x_1, \ldots, x_1^{a_1} \ldots x_n^{a_n}, -c_1, \ldots, -c_r)$ and $s_0 = \mathcal{M} \cdot v_0 \in \mathcal{L}$, such that

$$s_0 = (1, (x_1/X_1), \ldots, (x_1/X_1)^{a_1} \ldots (x_n/X_n)^{a_n}, 0, \ldots, 0).$$

One has $\|s_0\|_2 \leq \sqrt{\#M}$ and the knowledge of s_0 is sufficient to compute the root of f. Since in practice, we will not always recover s_0, the method consists in looking for a vector which is orthogonal to it. We compute an LLL-reduced basis $B = (b_1, \ldots, b_t)$ of (a sublattice of) $\mathcal{L}$ and a Gram-Schmidt's orthogonalization on B. As s_0 belongs to $\mathcal{L}$, it can be expressed as a linear combination of the $b_i^\star$'s and if its norm is smaller than those of $b_t^\star$, then the dot product $\langle s_0, b_t^\star \rangle = 0$.

Extracting the coefficients in $b_t^\star$ leads to a polynomial p_1 defined over M such that $p_1(x_1, \ldots, x_n) = 0$ and iterating the process with $b_{t-1}^\star, \ldots, b_{t-n+1}^\star$, one gets a multivariate polynomial system $\{p_1(x_1, \ldots, x_n) = 0, \ldots, p_n(x_1, \ldots, x_n) = 0\}$. Under the (heuristic) assumption that these polynomials are algebraically independent, the system can be solved in polynomial time.

Conditions on the Bounds X_i's. Since s_0 is small and we have an upper bound on $\|b_t^\star\|_2$, (*cf.* (1)), the condition $\sqrt{\#M} < 2^{-(t-1)/4}(\det(\mathcal{L}))^{1/t}$ implies $\langle s_0, b_t^\star \rangle = 0$. Removing parameters that do not influence the asymptotic result, this relation can be simplified to $|\det(\mathcal{L})| > 1$, leading to the following final condition:

$$\prod_{(y_1^{a_1} \ldots y_n^{a_n} \in M)} X_1^{a_1} \ldots X_n^{a_n} < N^{k_1 + \cdots + k_r} \tag{2}$$

The most complex step of the method is the choice of the collection of polynomials, what could be a difficult task when working with multiple polynomials.

3 Attacking a Non-linear Generator

For N an integer of size π, we denote by $\mathbb{Z}_N$ the residue ring of N elements. A pseudorandom non-linear generator can be defined by the following recurrence sequence:

$$v_{i+1} = F(v_i) \mod N \tag{3}$$

where $F(X) = \sum_{j=0}^{d} c_j X^j$ is a polynomial of degree d in $\mathbb{Z}_N[X]$ and v_0 is the secret seed. We assume that this generator outputs the k most significant bits of v_i at each iteration (with $k \in \{1, \ldots, \pi\}$), i.e. if $v_i = 2^{\pi-k} w_i + x_i$, w_i is output by the generator and $x_i < 2^{\pi-k} = N^\delta$ stays unknown. We want to recover $x_i < N^\delta$ for some $i \in \mathbb{N}$ from consecutive values of the pseudorandom sequence (with δ as large as possible) knowing F or not.

3.1 Case F Known

Any non-linear pseudorandom generator defined by a known iteration function F can be broken when sufficiently many bits are output at each iteration. In the following, we determine that amount of output bits when two (Theorem 1) then more (Theorem 2) consecutive outputs are known to the attacker.

Theorem 1 (Two consecutive outputs). *Let $\mathcal{G}$ be a non-linear pseudorandom generator defined by a known iteration function $F(X)$ of degree d. If an adversary has access to two consecutive outputs of $\mathcal{G}$ then it will be able to predict the entire sequence that follows ; under the condition that at least $\frac{d}{d+1}\pi$ most significant bits are output at each iteration, that is:*

$$\delta < \frac{1}{d+1}$$

Proof. Suppose the adversary is given two approximations w_0 and w_1 of two consecutive values v_0 and v_1 that satisfy (3). By denoting v_0 as $2^{\pi-k}w_0 + x_0$ and $v_1 = 2^{\pi-k}w_1 + x_1$, we obtain:

$$2^{\pi-k}w_1 + x_1 - \sum_{j=0}^{d} c_j(2^{\pi-k}w_0 + x_0)^j = 0 \mod N$$

Let $f(y_0, y_1)$ be the polynomial $y_1 + a_0 + a_1 y_0 + \cdots + a_d y_0^d$ defined by this equation, where the values a_i, that explicitly depend on w_0, w_1 and the coefficients c_i, are known to the adversary. The goal is to compute efficiently the (small) modular root (x_0, x_1) of $f(y_0, y_1)$. To do so, let us consider the following collection of polynomials:

$$\{y_0^j f^i(y_0, y_1) \quad | \quad di + j \leq dm \quad \wedge \quad i > 0\}$$

where $m \geq 1$ is a fixed integer. Knowing the shape of f, the list of monomials appearing within this collection can be described as:

$$\{y_1^i y_0^j \quad | \quad di + j \leq dm\}$$

Using Coppersmith's method, the right-hand side (*resp.* the left-hand side) of (2) is then equal to:

$$\prod_{i=1}^{m} \prod_{j=0}^{d(m-i)} N^i = N^{\frac{1}{6}m(m+1)(dm-d+3)} \quad \left(\textit{resp.} \prod_{i=0}^{m} \prod_{j=0}^{d(m-i)} N^{i\delta} N^{j\delta} \right).$$

Thus, the algorithm (heuristically) outputs the root of f in polynomial time as soon as:

$$\delta < \frac{\frac{1}{6}m(m+1)(dm-d+3)}{\frac{1}{12}m(m+1)(2d^2m + 2dm + 6 + d^2 + d)} \xrightarrow[m \to +\infty]{} \frac{1}{d+1} \tag{4}$$

□

This bound is better than those previously obtained by Blackburn *et al.* [3]. Indeed, their result was approximately $\delta < 1/d^2$ when two consecutive outputs are known to the attacker.

Theorem 2 (More consecutive outputs). *Let $\mathcal{G}$ be a non-linear pseudorandom generator defined by a known iteration function $F(X)$ of degree d. If an*

adversary has access to $n+2$ (with $n \geq 1$) consecutive outputs of $\mathcal{G}$ then it will be able to predict the entire sequence that follows ; under the condition that at least $\frac{d^{n+2}-d^{n+1}}{d^{n+2}-1}\pi$ most significant bits are output at each iteration, that is:

$$\delta < \frac{d^{n+1}-1}{d^{n+2}-1}$$

Proof. Let us assume that the attacker knows $n+2$ consecutive outputs of the generator $w_0, \ldots, w_{n+1}$. Writing v_i as $2^{\pi-k}w_i + x_i$ (for $i \in \{0, \ldots, n+1\}$), we want to recover the solution $(x_0, \ldots, x_{n+1})$ of the multivariate polynomial system:

$$\begin{cases} f_0(y_0, y_1) & = y_1 + a_{00} + a_{01}y_0 + \cdots + a_{0d}y_0^d \mod N \\ & \vdots \\ f_n(y_n, y_{n+1}) & = y_{n+1} + a_{n0} + a_{n1}y_n + \cdots + a_{nd}y_n^d \mod N \end{cases}$$

where each polynomial f_i is constructed in the same way as for the "two consecutive outputs" case. From now, we use the following collection of polynomials:

$$\left\{ y_0^j f_0^{i_0} \ldots f_n^{i_n} \quad | \quad d(i_0 + di_1 + \cdots + d^n i_n) + j \leq dm \quad \wedge \quad i_0 + \cdots + i_n > 0 \right\}$$

where $m \geq 1$ is a fixed integer. As it seems to be a difficult task to describe the set of monomials appearing in that collection for the general case, we first focus on what happens with two polynomials f_0 and f_1. In that case, the set can be described by the powers of these polynomials, that is

$$\left\{ (y_0^j y_1^i) \cdot (y_1^k y_2^l) \quad | \quad di + j \leq dm \quad \wedge \quad dl + k \leq dm - di - j \right\}$$

Another way of expressing this set is $\left\{ y_0^j y_1^i y_2^l \quad | \quad di + j + dl \leq dm \right\}$. From that point, by induction on n, we can show that the monomials appearing in the collection can be described as:

$$\left\{ y_0^j y_1^{i_0} \ldots y_{n+1}^{i_n} \quad | \quad d(i_0 + di_1 + \cdots + d^n i_n) + j \leq dm \right\}$$

The right-hand side and the left-hand side of (2) is then equal to $N^{A(m,n)}$ and $N^{B(m,n)}$ respectively, where:

$$A(m,n) = \sum_{i_0=0}^{m} \sum_{i_1=0}^{\lfloor \frac{m-i_0}{d} \rfloor} \cdots \sum_{j=0}^{d(m-\sum_{p=0}^{n} d^p i_p)} i_0 + \cdots + i_n$$

$$B(m,n) = \sum_{i_0=0}^{m} \sum_{i_1=0}^{\lfloor \frac{m-i_0}{d} \rfloor} \cdots \sum_{j=0}^{d(m-\sum_{p=0}^{n} d^p i_p)} i_0 + \cdots + i_n + j$$

Our goal is to obtain an asymptotic expression of the multiples sums $A(m,n)$ and $B(m,n)$ which depends on the number of outputs n, when m goes to $+\infty$. It is quite clear that the floor function appearing in the upper bound of the sums

can be omitted and we will use several times a trick from [12] which consists in letting indices of a sum run over a larger range in order to obtain a symmetric formula that is easier to evaluate. Basically, it relies on the following observation which holds for any function f:

$$\sum_{i=0}^{N} f(i) = \frac{1}{d} \sum_{i=0}^{dN} f(\lfloor \frac{i}{d} \rfloor).$$

Applying this trick n times on $A(m,n)$, one obtains:

$$\begin{aligned} A(m,n) &\simeq \frac{1}{d} \cdots \frac{1}{d^n} \sum_{i_0=0}^{m} \sum_{i_1=0}^{m-i_0} \cdots \sum_{j=0}^{d(m-\sum_{p=0}^{n} i_p)} i_0 + \frac{1}{d} i_1 + \cdots + \frac{1}{d^n} i_n \\ &\simeq d \cdot \frac{1}{d} \cdots \frac{1}{d^n} \sum_{i_0=0}^{m} \sum_{i_1=0}^{m-i_0} \cdots \sum_{j=0}^{m-\sum_{p=0}^{n} i_p} i_0 + \frac{1}{d} i_1 + \cdots + \frac{1}{d^n} i_n \end{aligned}$$

and similarly

$$B(m,n) = d \cdot \frac{1}{d} \cdots \frac{1}{d^n} \sum_{i_0=0}^{m} \sum_{i_1=0}^{m-i_0} \cdots \sum_{j=0}^{m-\sum_{p=0}^{n} i_p} i_0 + \frac{1}{d} i_1 + \cdots + \frac{1}{d^n} i_n + dj.$$

We get for $A(m,n)$ and $B(m,n)$:

$$A(m,n) \simeq \frac{1}{d^2} \cdots \frac{1}{d^n} \left(\frac{d^{n+1}-1}{d^n(d-1)} \right) p_1 \text{ and } B(m,n) \simeq \frac{1}{d^2} \cdots \frac{1}{d^n} \left(\frac{d^{n+2}-1}{d^n(d-1)} \right) p_1$$

where

$$p_1 = \sum_{i_0=0}^{m} \sum_{i_1=0}^{m-i_0} \cdots \sum_{j=0}^{m-\sum_{p=0}^{n} i_p} i_0.$$

We obtain in consequence the following bound:

$$\delta < \frac{A(m,n)}{B(m,n)} \simeq \frac{d^{n+1}-1}{d^{n+2}-1}$$

□

When the number of consecutive values known by the adversary tends to infinity, this condition becomes $\delta < 1/d$. Knowing that d is the degree of the iteration function, this result seems to be the optimal one when using Coppersmith's technique.

3.2 Case F Unknown

We show that a non-linear pseudorandom generator defined by an unknown iteration function F can also be broken. In order to apply Coppersmith's technique,

one needs to construct a polynomial P (from the unknown iteration function F) with a root encoding the secret seed. We will see in the forthcoming sections that one could use elimination techniques to find such a P. Let us denote D the degree of P (depending on $d = \deg F$ and on the elimination technique used) and we consider a monomial order such that the leading coefficient[1] of P is equal to 1 modulo N. Since there are $d+1$ unknown coefficients in F, one requires $d+2$ consecutive equations of the form $v_{i+1} = F(v_i) \bmod N$, and thus $d+3$ consecutive outputs of the generator.

Theorem 3 ($d+3$ consecutive outputs). *Let $\mathcal{G}$ be a non-linear pseudorandom generator defined by an unknown iteration function $F(X)$ of degree d. We consider an adversary that has access to $d+3$ consecutive outputs of $\mathcal{G}$ and can compute a polynomial P of degree D and a monomial order as above.*

It will be able to predict the entire sequence that follows ; under the condition that at least $\frac{D^2(d+3)-1}{D(d+3)}\pi$ most significant bits are output at each iteration, that is $\delta < \frac{1}{D^2(d+3)}$. Moreover, if one assumes that the degree of the leading monomial of P is equal to D, then this bound can be improved to:

$$\delta < \frac{1}{D(d+3)}.$$

Proof. Let us assume that the adversary knows $w_0, \ldots, w_{d+2}$. By manipulating the system $\left(v_{i+1} = F(v_i) \mod N, i \in \{0, \ldots, d+1\}\right)$ one obtains a polynomial P satisfying $P(x_0, \ldots, x_{d+2}) = 0 \mod N$. Since the shape of P and its degree D both depend on the technique used to manipulate the initial system, describing the monomials appearing in P and therefore in P^m is an impossible task. Consequently, the only way to perform Coppersmith's method is to choose a simpler but larger set of monomials which necessarily contains those of P^m:

$$\left\{ y_0^{j_0} \ldots y_{d+2}^{j_{d+2}} \quad | \quad j_0 + j_1 + \cdots + j_{d+2} \leq Dm \right\}$$

The leading monomial of P, $LM(P)$, can be described as $y_0^{\alpha_0} \ldots y_{d+2}^{\alpha_{d+2}}$ where at least one of the α_i is non negative. Without loss of generality, we can assume for now that $\alpha_0 > 0$. In that case, one can apply Coppersmith's method on the following collection of polynomials:

$$\left\{ y_1^{j_1} \ldots y_{d+2}^{j_{d+2}} P^i \quad | \quad Di + j_1 + \cdots + j_{d+2} \leq Dm \ \wedge \ 1 \leq i \leq m \right\}$$

As y_0 only comes from the powers of P, the prohibition of the multiplication by y_0 ensures that the collection of polynomials will be linearly independent. The right-hand side (*resp.* the left-hand side) of (2) is then equal to N to the power:

$$\sum_{1 \leq i \leq m} \ \sum_{j_1 + \cdots + j_{d+2} \leq Dm - Di} i \qquad \left(\textit{resp.} \sum_{j_0 + \cdots + j_{d+2} \leq Dm} \delta(j_0 + \cdots + j_{d+2}) \right).$$

[1] In the general case, this condition is almost always satisfied and this is obviously true when N is prime

We can show that this formula leads to the following condition:

$$\delta < \frac{1}{D^2(d+3)}$$

In fact, this result can be improved if one assumes that the degree of $LM(P)$ is equal to D. Indeed, this monomial can be described as $y_0^{\alpha_0} \dots y_{d+2}^{\alpha_{d+2}}$ with $\sum_{i=0}^{d+2} \alpha_i = D$. In order to keep the linear independency between the polynomials, one should only consider polynomials of the form $Mon \times P^i$ such that $Mon \neq LM(P)$. This leads to the following collection:

$$\left\{ y_0^{j_0} y_1^{j_1} \dots y_{d+2}^{j_{d+2}} P^i \;\middle|\; \begin{array}{rl} & Di + j_0 + j_1 + \dots + j_{d+2} \leq Dm \\ \wedge & 1 \leq i \leq m \\ \wedge & (j_0 < \alpha_0) \cup \dots \cup (j_{d+2} < \alpha_{d+2}) \end{array} \right\}$$

Using the same kind of tricks as in the proof of Theorem 2, the resulting asymptotic bound becomes:

$$\delta < \alpha_0 \frac{1}{D^2(d+3)} + \dots + \alpha_{d+2} \frac{1}{D^2(d+3)} = \frac{1}{D(d+3)}$$

□

More consecutive outputs. We want to generalize the previous attack when the adversary is given access to more consecutive outputs. Let us assume, for instance, that it has access to $d+n+2$ consecutive values $w_0, \dots, w_{d+1+n}$; its goal is then to compute the (small) solution $(x_0, \dots, x_{n+d+1})$ of the multivariate polynomial system $(P_1(y_0, \dots, y_{d+2}), \dots, P_n(y_{n-1}, \dots, y_{n+d+1}))$ where the polynomials P_i of degree D, are defined as in the previous section. As before, finding a general description of the monomials appearing in these polynomials is a challenging task. Thus we consider a larger set of monomials, easier to describe:

$$\left\{ y_0^{j_0} \dots y_{d+1+n}^{j_{d+1+n}} \quad | \quad j_0 + j_1 + \dots + j_{d+1+n} \leq Dm \right\}$$

Let us express the leading monomial of P_1 as $y_0^{\alpha_0} \dots y_{d+2}^{\alpha_{d+2}}$ with at least one of the $\alpha_i \geq 1$, the leading monomial of P_2 as $y_1^{\alpha_0} \dots y_{d+3}^{\alpha_{d+2}}$ and those of P_n as $y_{n-1}^{\alpha_0} \dots y_{n+d+1}^{\alpha_{d+2}}$, using a monomial order such as *lex* or *hlex* with $y_0 < \dots < y_{d+1+n}$. Without loss of generality, we can assume that $\alpha_0 > 0$. From that, one can apply Coppersmith's method on the following collection of polynomials:

$$\left\{ y_n^{j_1} \dots y_{n+d+1}^{j_{d+2}} P_1^{i_1} \dots P_n^{i_n} \;\middle|\; \begin{array}{rl} & D(i_1 + \dots + i_n) + j_1 + \dots + j_{d+2} \leq Dm \\ \wedge & 1 \leq i_1 + \dots + i_n \leq m \end{array} \right\}$$

The prohibition of the multiplication by $y_0, \dots, y_{n-1}$ ensures that all the polynomials of the collection are linearly independent. Thus, the right-hand side (*resp.* the left-hand side) of (2) is equal to N to the power:

$$\sum_{\substack{1 \leq i_1 + \dots + i_n \leq m \\ j_1 + \dots + j_{d+2} \leq Dm - D(i_1 + \dots + i_n)}} i_1 + \dots + i_n \quad \left(\textit{resp.} \sum_{j_0 + \dots + j_{n+d+1} \leq Dm} \delta(j_0 + \dots + j_{n+d+1}) \right).$$

We can show that the resulting asymptotic bound is $\delta < \dfrac{n}{D^{n+1}(n+d+2)}$ (details can be found in the full version).

Remark 1. This bound is not interesting as its value decreases when the adversary is given access to more outputs. However, we are convinced that it can significantly be improved. Indeed, using the same kind of techniques as in the previous case, we might be able to gain a factor D for each involved polynomial and get:

$$\delta < \frac{n}{D(n+d+2)} \xrightarrow[m \to +\infty]{} \delta < \frac{1}{D}$$

In practice we notice that this conjecture seems to be true, see for instance the analysis of the quadratic generator in Section 3.3.

3.3 Application: Attacking the Quadratic Generator

For p a prime of size π, the notation $\mathbb{Z}_p$ refers to the field of p elements. The quadratic generator is defined by the following recurrence sequence:

$$v_i = a v_{i-1}^2 + b \mod p \tag{5}$$

In that particular case, the iteration function $F(x)$ is defined as $F(x) = ax^2 + b$ where $a \in \mathbb{Z}_p^*$ and $b \in \mathbb{Z}_p$ are constant values. Exactly as before, we denote the secret seed as $v_0 \in \mathbb{Z}_p$ and we assume that the generator outputs the k most significant bits of v_i at each iteration (with $k \in \{1, \ldots, \pi\}$). In other words, each value v_i can be written as $2^{\pi-k} w_i + x_i$ where w_i is output by the generator and $x_i < 2^{\pi-k} = p^\delta$ stays unknown. Our goal consists in recovering the value $x_i < p^\delta$ for some $i \in \mathbb{N}$ by using some consecutive values output by the pseudorandom sequence (with δ as large as possible).

Case *F* Known. If the adversary is given access to two consecutive outputs of the generator, then it can break the scheme under the condition that sufficiently many bits are output by the generator at each iteration. More precisely, for a fixed value m (that will define the size of the corresponding lattice), the bound on δ should respect the following condition, directly coming from Equation (4) in Theorem 1:

$$\delta < \frac{1}{6} \cdot \frac{2m+1}{m+1}$$

In particular, taking $m = 1$ leads to the bound $\delta < 1/4$ previously reached by Blackburn *et al.* [3]. This bound can be improved to $\delta < 1/3$ when the quantity m goes to infinity. This value is exactly the same as those previously obtained by Blackburn *et al.* [3] when the authors assume that the adversary is given access to an infinite number of outputs, whereas it only requires here two outputs of the generator. Finally, when increasing the number of known outputs to infinity, the condition becomes $\delta < 1/2$ (see Theorem 2).

Case *F* Unknown. Knowing that the coefficients a and b appearing in the iteration function $F(x) = ax^2 + b$ are unknown to the attacker, the first step consists in expressing the relations between the outputs of the generator exclusively in terms of known quantities. More precisely, by using four consecutive outputs, the adversary is able to eliminate the quantities a and b by considering the following polynomial P of degree 3:

$$P = c + c_0 y_0 + c_1 y_1 + c_2 y_2 + c_3 y_3 + d_0(y_0^2 - y_1^2) + d_1(y_2 y_0 - y_0 y_3) + d_2(y_1^2 - 3y_2^2) \\ +2d_2 y_1 y_2 + d_3(y_2^2 - 3y_1^2 + 2y_1 y_3) + e(y_2^2 y_1 - y_1^3 + y_1^2 y_3 - y_2^3 - y_0^2 y_3 + y_0^2 y_2) \bmod p$$

As each coefficient in this polynomial is inversible modulo p, one can consider that $LM(P)) = 1$. Thus, applying Theorem 3, one reaches the bound $\delta < 1/15$, knowing that the degree d of the iteration function F is equal to 2 and those of the polynomial P is 3. In fact, this bound can be improved as the coefficient related to x in the iteration function, is equal to zero. Indeed, the denominator in the formula given by Theorem 3 can in fact be expressed as $D.\ell(n)$ where $\ell(n)$ is the number of required outputs. In this particular case, as $\ell(n)$ is equal to four, the bound thus becomes $\delta < 1/12$. In the same scenario, Blackburn *et al.* [3] reached the value $\delta < 1/19$.

We assume that the adversary is given access to more consecutive outputs and generalize the previous construction using the fact that the iteration function F contains one zero coefficient. In that case, if the set of monomials stays easy to formulate, namely $\{y_0^{j_0} \dots y_{n+2}^{j_{n+2}} \mid j_0 + \dots + j_{n+2} \leq 3m\}$, this is not the case for the collection of polynomials which becomes:

$$\left\{ y_0^{j_0} y_1^{j_1} y_2^{a_2} \dots y_{n+1}^{a_{n+1}} y_{n+2}^{j_{n+2}} P_1^{i_1} \dots P_n^{i_n} \;\middle|\; \begin{array}{l} 0 < i_1 + \dots + i_n \leq m \\ 0 \leq a_\ell \leq \min(2, 3m - 3\sum_{t=1}^{n} i_p - \sum_{t=2}^{\ell-1} a_p) \\ (\text{for } \ell \in \{2, \dots, n+1\}) \\ j_0 + j_1 + j_{n+2} \leq 3m - 3(i_1 + \dots + i_n) \\ \qquad\qquad - (a_2 + \dots + a_{n+1}) \end{array} \right\}$$

The estimation of the "weight" of these two sets allows to reach the asymptotic bound $\delta < 1/3$ when both m (related to the dimension of the involved lattice) and n go to infinity (*cf.* the full version of the paper). This value seems to confirm the conjecture $\delta < 1/D$ discussed in Remark 1. Moreover, it significantly improves the bound $\delta < 1/12$ previously obtained by Blackburn *et al.* in [3] and it provides interesting asymptotic bounds for small values of n (when m goes to infinity):

Number of outputs	4	5	6	7	8	9	10	11	12
Asymptotic bound	1/12	2/15	1/6	4/21	5/24	2/9	7/30	8/33	1/4

3.4 The Inversive Generator

The inversive generator is defined by the recurrence sequence $v_i = av_{i-1}^{-1} + b \bmod p$ where p is a prime and a,$b \in \mathbb{Z}_p$. As usual, we assume that this generator outputs the k most significant bits at each iteration. When a and b are known, the polynomial $h(y_0, y_1) = y_0 y_1 + c_0 y_0 + c_1 y_1 + c$ can be constructed, using two consecutive outputs, where c, c_0, c_1 are constant values.

Let us now look at the link between the geometrical representation of the polynomial $h(y_0, y_1)$, namely a square, and those of $f(y_0, y_1) = y_1 - c_0 y_0 - a y_0^2 + c \bmod p$, which corresponds to the polynomial defined for the quadratic generator with two outputs when a and b are known, that can be represented as a triangle. The denominator appearing in the value δ, coming from Equation (2), can be

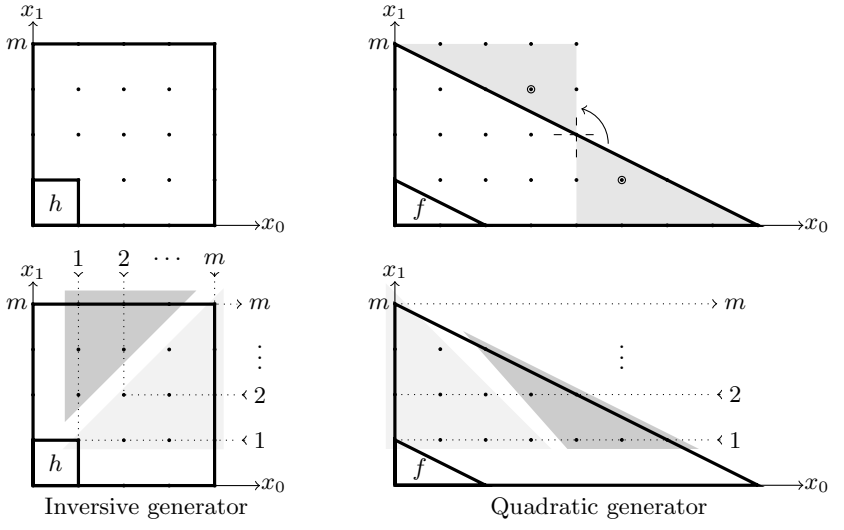


Fig. 1. Geometrical link between f and h

seen as the sum of the coordinates of each point belonging to the form defined by the polynomial. For the inversive generator, this sum can be expressed as:

$$\sum_{x_0=0}^{m} \sum_{x_1=0}^{m} x_0 + x_1 = m(m+1)^2 = \sum_{x_1=0}^{m} \sum_{x_0=0}^{2m-x_1} x_0 + x_1$$

The collection of polynomials involved in the quadratic generator case, gives the following formula, corresponding to the numerator:

$$\sum_{x_1=1}^{m} \sum_{x_0=0}^{2(m-x_1)} x_1 = \frac{1}{6} m(m+1)(2m+1) = \sum_{x_1 x_0=1}^{m} \sum_{x_1=0}^{m-x_1 x_0} x_1 x_0 + \sum_{x_1 x_0=1}^{m-1} \sum_{x_0=1}^{m-x_1 x_0} x_1 x_0.$$

Figure 1 shows the geometrical link between these two generators (on the top, the set of monomials ; on the bottom, the collection of polynomials). When working with more polynomials, the situation is identical. Moreover, when a and b are unknown, the polynomial used to build the collection in the inversive generator case is similar to those used in the quadratic generator's one. The obtained bound is also $1/12$ and with more consecutive outputs, it tends to $1/3$, similarly to the quadratic generator.

Table 1. Some bounds for the inversive generator, a,b known

	2 outputs	3 outputs	4 outputs	5 outputs	6 outputs	7 outputs
previous bound	0.25	0.286	0.3	0.308	0.313	0.316
new achievable bound	0.321	0.39	0.40	0.401	0.401	0.401
	$m = 13$	$m = 9$	$m = 8$	$m = 8$	$m = 8$	$m = 8$
new asymptotic bound	1/3	3/7	7/15	15/31	31/63	63/127

4 The Pollard Generator

The recursive sequence of the Pollard generator is defined as $v_i = v_{i-1}^2 + b \mod p$ with $b \in \mathbb{Z}_p$ (*i.e.* it is a particular instance of the quadratic generator where the constant a is equal to 1). As a consequence, the attack scenario is exactly the same as in the previous section when b is known to the attacker. However, if one takes advantage of the fact that a is fixed to 1, a specific analysis can be made and a better bound can be obtained. To reach such a result, we use a novel technique, called unravelled linearization whose description is provided below.

Unravelled linearization. In 2009, Hermann and May introduced a new technique called *unravelled linearization* [12] that allows to work with smaller lattices by optimizing the way the initial polynomial is written. It consists in improving the bounds, see Equation (2), by reducing the number of monomials in M while keeping the same amount of powers of N in the right hand side of the equation.

Let us show what happens on a toy example, say $f(x, y) = x^2 + x + y$ having a root (x_0, y_0) modulo N where $|x_0| < X$ and $|y_0| < Y$ with $X = Y$. The idea is to find the better way of linearizing f before proceeding to Coppersmith's construction. If we fix $u = x^2$, the polynomial f becomes $g(u, x, y) = u+x+y$ and the bounds on the root can be determined by the following formula $UXY < N$. Knowing that $U = X^2$, this leads to $X = N^{1/4}$. Now, let us take another smarter linearization, say $u = x^2+y$, leading to the polynomial $g(u, x) = u+x$. This time, the formula becomes $UX < N$, what leads to the improved bound $X = N^{1/3}$. In this case, the "weight" of y is hidden in u by the weight of x^2.

One need to use another tricky manipulation to conclude. Let us go back to our toy example $g(u, x) = u + x$ and construct the original matrix defined by Coppersmith taking the collection $\{g, g^2\}$. This leads to the matrix $\mathcal{M}$ that follows, thus reaching the asymptotic bound $U^4X^4 < N^3$, what gives $X < N^{1/4}$.

$$\underbrace{\left(\begin{array}{ccccc|cc}
1/U & & & & & 1 & 0 \\
 & 1/X & & & & 1 & 0 \\
 & & 1/X^2 & & & 0 & 1 \\
 & & & 1/UX & & 0 & 2 \\
 & & & & 1/U^2 & 0 & 1 \\
\hline
 & & & 0 & & N & \\
 & & & & & & N^2
\end{array}\right)}_{\mathcal{M}}
\begin{array}{c} u \\ x \\ x^2 \\ ux \\ u^2 \end{array}
\qquad
\underbrace{\left(\begin{array}{ccccc|cc}
1/U & & & & & 1 & 1 \\
 & 1/X & & & & 1 & 0 \\
 & & \mathbf{1/Y} & & & 0 & -1 \\
 & & & 1/UX & & 0 & 2 \\
 & & & & 1/U^2 & 0 & 1 \\
\hline
 & & & 0 & & N & \\
 & & & & & & N^2
\end{array}\right)}_{\mathcal{M}'}
\begin{array}{c} u \\ x \\ \boldsymbol{y} \\ ux \\ u^2 \end{array}$$

But here is the point: by definition of u, the monomial x^2 can easily be written as $u - y$, thus allowing to express the polynomial g^2 as $g^2 = u^2 + 2ux + u - y$.

Such a manipulation leads to the matrix on the right hand side, say $\mathcal{M}'$. In this case, the obtained bounds on the root can be reformulated as $U^4X^2Y < N^3$ what gives the improved result $X < N^{3/11}$. This benefit can be understood by the fact that we have managed to decrease the weight of the monomials in the set M by 1 while keeping the exact number of powers of N appearing in the right hand side of Equation (2). Such manipulations are quite hard to proceed, they strongly rely on the linearization chosen for the initial polynomial f (a more detailed discussion on the importance of the choice of the linearization can be found in the full version of the paper).

4.1 Case F Known

Attack with Two Consecutive Outputs. Let us first assume that the adversary is given access to two consecutive outputs of the generator, namely w_0 and w_1. Knowing that $v_0 = 2^{\pi-k}w_0 + x_0$ and $v_1 = 2^{\pi-k}w_1 + x_1$, we reach the same relation as those previously obtained for the quadratic case:

$$x_1 - 2^{\pi-k+1}w_0x_0 - x_0^2 - b + 2^{\pi-k}w_1 - 4^{\pi-k}w_0^2 = 0 \mod p$$

Let us denote by $f(y_0, y_1)$ the polynomial $y_1 - c_0y_0 - y_0^2 + d_0$ where the coefficients $c_0 = 2^{\pi-k+1}w_0$ and $d_0 = -b + 2^{\pi-k}w_1 - 4^{\pi-k}w_0^2$ are known to the attacker. As usual, its goal consists in recovering the small modular root (x_0, x_1) of $f(y_0, y_1)$.

To solve this problem, we use the unravelled linearization technique. As already stated, the first step consists in choosing a good linearization for f. In this particular case, we set $u = y_1 - y_0^2$, what leads to the following polynomial $g(y_0, u) = u - c_0y_0 + c \mod p$. In that case, the bound on u can thus be expressed as $U = X_0^2$.

Let us now consider the collection of polynomials defined as $y_0^j g^i(y_0, u)$ with $i + j \leq m$ and $i > 0$. The list of monomials appearing in that collection can be described as $M = \left\{ y_0^j u^i \quad | \quad i + j \leq m \right\}$. Initially, we use this set of polynomials to construct the matrix defined by Coppersmith, as in Section 2.2. In that case, the right-hand side (*resp.* the left-hand side of) of (2) can easily be expressed as p to the power

$$\sum_{i=1}^{m} \sum_{j=0}^{m-i} i = \frac{1}{6}m^3 + o(m^3) \qquad \left(\textit{resp. } \delta \sum_{i=0}^{m} \sum_{j=0}^{m-i} 2i + j = \frac{\delta}{2}m^3 + o(m^3) \right)$$

The idea of the unravelled linearization technique is to improve the bound on δ by decreasing the weight of the monomials. To do so, one should proceed to a "back-substitution" in the constructed matrix, as explained in the previous section. In that particular case, knowing that $y_0^2 = y_1 - u$, the following replacement

is done (for all monomials μ such that $\mu \cdot y_0^2 \in M$): $\mu \cdot y_0^2 \to \mu \cdot y_1 - \mu \cdot u$. It is obvious that the presence of $\mu \cdot y_0^2$ in the set M implies those of $\mu \cdot u$. As a consequence, doing such a manipulation allows to replace the quantity $\mu \cdot y_0^2$ by $\mu \cdot y_1$ thus decreasing by "1" the weight on the monomials. If we express the collection M as $M = \left\{ y_0^{2b+a} u^i \quad | \quad a \in \{0,1\} \wedge a + 2b + i \leq m \right\}$, after the back-substitution, we obtain the set $M' = \left\{ y_1^b y_0^a u^i \quad | \quad a \in \{0,1\} \wedge a + 2b + i \leq m \right\}$. In that case, the *new* left-hand side in Equation (2) becomes p raised to the power:

$$\delta \sum_{a=0}^{1} \sum_{i=0}^{m-a} \sum_{b=0}^{\lfloor \frac{m-i-a}{2} \rfloor} (a + b + 2i) = \delta \frac{5}{12} m^3 + o(m^3)$$

Thus, the corresponding asymptotic bound on δ becomes:

$$\delta < \frac{1/6m^3 + o(m^3)}{5/12m^3 + o(m^3)} \xrightarrow[m \to +\infty]{} \frac{2}{5}.$$

This bound is better than $\delta < 5/14$, previously obtained by Blackburn et al. [11] when working with one polynomial. One can also notice that $2/5$ is exactly the bound obtained in [12] for attacking the Blum-Blum Shub generator.

More Consecutive Outputs. In that case, one can easily generalize the method explained before in the same way as what has been done for the Blum-Blum Shub generator, thus reaching the bound $\delta < 1/2$. Details are left to the reader.

4.2 Case F Unknown

Attack with three consecutive outputs. Let us now consider the case of an adversary having access to three consecutive outputs of the generator. In that case, writing two consecutive recurrence relations and subtracting both of them leads to:

$$-x_1^2 + x_0^2 + x_2 + \underbrace{2^{\pi-k+1} w_0}_{c_0} x_0 - \underbrace{(2^{\pi-k+1} w_1 + 1)}_{c_1} x_1 + \underbrace{2^{\pi-k} w_2 - 2^{\pi-k} w_1 + 4^{\pi-k} w_0^2 - 4^{\pi-k} w_1^2}_{c} = 0 \mod p.$$

The adversary wants to recover the small modular root (x_0, x_1, x_2) of the polynomial $f(y_0, y_1, y_2) = -y_1^2 + y_0^2 + y_2 + c_0 y_0 - c_1 y_1 + c$. To do so, we use again the unravelled linearization technique. To linearize the polynomial f, we set $u = -y_1^2 + y_0^2 + y_2$, reaching the new following expression $g(u, y_0, y_1) = u + c_0 y_0 - c_1 y_1 + c$. Let us now consider the collection of polynomials defined as $y_0^k y_1^j g^i$ with $i + j + k \leq m$ and $i > 0$. In that case, the list of involved monomials can easily be expressed as $M = \left\{ u^i y_1^j y_0^k \quad | \quad i + j + k \leq m \right\}$. Thus, the right-hand side of Coppersmith's Equation (2) is given by p to the power:

$$\sum_{i=1}^{m}\sum_{j=0}^{m-i}\sum_{k=0}^{m-i-j} i = \frac{1}{24}m^4 + o(m^4).$$

Before evaluating the weight of the monomials in M, we perform some back-substitutions. In this case, the rule given by the linearization is (for all monomials μ such that $\mu \cdot y_1^2 \in M$): $\mu \cdot y_1^2 \rightarrow \mu \cdot y_0^2 + \mu \cdot y_2 - \mu \cdot u$. One can notice that the presence of the monomial $\mu \cdot y_1^2$ in the set M automatically implies those of $\mu \cdot y_0^2$ and $\mu \cdot u$. Thus, each monomial of the form $\mu \cdot y_1^2$ can be replaced by one of those $\mu \cdot y_2$ in the constructed matrix, again decreasing by "1" the weight on the involved monomials. The shape of the *new* constructed set M is then:

$$\left\{u^i y_2^b y_1^a y_0^k \quad | \quad a \in \{0,1\} \wedge i + k + a + 2b \leq m\right\}$$

In that case, the *new* left hand side of Equation (2) becomes:

$$\delta \sum_{a=0}^{1}\sum_{i=0}^{m-a}\sum_{b=0}^{\lfloor\frac{m-i-a}{2}\rfloor}\sum_{k=0}^{m-i-a-2b}(a+b+2i+k) = \frac{7\delta}{48}m^4 + o(m^4)$$

which leads to the following bound on δ:

$$\delta < (1/24m^4 + o(m^4))/(7/48m^4 + o(m^4)) \xrightarrow[m\rightarrow+\infty]{} 2/7.$$

More Consecutive Outputs. Let us assume that the attacker knows $n+2$ consecutive outputs, for $n \geq 2$. We denote f_i the relation between two outputs:

$$f_i = 2^{\pi-k}w_i + y_i - (2^{\pi-k}w_{i-1} + y_{i-1})^2 - b \mod p \quad i \in \{1,\ldots,n\}$$

These polynomials have (x_i, x_{i-1}) as a root modulo p and denoting $g_i = f_{i+1} - f_i$ for $i \in \{1,\ldots,n\}$, we have $g_i = -y_i^2 + y_{i-1}^2 + y_{i+1} + c_i y_{i-1} - d_i y_i + e_i \mod p$ for some constants c_i, d_i, e_i known to the adversary.

Knowing the set of polynomials $\{g_1,\ldots,g_n\}$, the attacker wants to recover the unknown values x_i. To do so, we use again the unravelled linearization technique by choosing $u_i = -y_i^2 + y_{i-1}^2 + y_{i+1}$, what leads to: $g_i = u_i + c_i y_{i-1} - d_i y_i + e_i$. Such polynomials allows us to reach the asymptotic following bound $\delta < \frac{2}{5}$ (details will be given in the full version). In that particular case, we think this bound could be improved to $\delta < 1/2$, following the discussion from Remark 1.

Table 2. Theoretical and experimental bounds for the Pollard generator, b unknown

Number of outputs	3	4	5	6	7	8
Previous bound [3]	0.261	0.286	0.3	0.308	0.313	0.316
Our achievable bound	0.278	0.319	0.324	0.324	0.324	0.324
Our asymptotic bound	2/7	6/17	14/37	30/77	62/157	126/317

References

1. Bauer, A., Joux, A.: Toward a Rigorous Variation of Coppersmith's Algorithm on Three Variables. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 361–378. Springer, Heidelberg (2007)
2. Blackburn, S.R., Gomez-Perez, D., Gutierrez, J., Shparlinski, I.E.: Predicting the Inversive Generator. In: Paterson, K.G. (ed.) Cryptography and Coding 2003. LNCS, vol. 2898, pp. 264–275. Springer, Heidelberg (2003)
3. Blackburn, S.R., Gomez-Perez, D., Gutierrez, J., Shparlinski, I.E.: Predicting nonlinear pseudorandom number generators. Math. Comput. 74(251), 1471–1494 (2005)
4. Blackburn, S.R., Gomez-Perez, D., Gutierrez, J., Shparlinski, I.E.: Reconstructing noisy polynomial evaluation in residue rings. J. Algorithms 61(2), 47–59 (2006)
5. Blömer, J., May, A.: A Tool Kit for Finding Small Roots of Bivariate Polynomials over the Integers. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 251–267. Springer, Heidelberg (2005)
6. Boyar, J.: Inferring sequences produced by a linear congruential generator missing low-order bits. Journal of Cryptology 1(3), 177–184 (1989)
7. Boyar, J.: Inferring sequences produced by pseudo-random number generators. J. ACM 36(1), 129–141 (1989)
8. Coppersmith, D.: Finding a Small Root of a Bivariate Integer Equation; Factoring with High Bits Known. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 178–189. Springer, Heidelberg (1996)
9. Coppersmith, D.: Finding a Small Root of a Univariate Modular Equation. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 155–165. Springer, Heidelberg (1996)
10. Gomez, D., Gutierrez, J., Ibeas, Á.A.: Cryptanalysis of the Quadratic Generator. In: Maitra, S., Veni Madhavan, C.E., Venkatesan, R. (eds.) INDOCRYPT 2005. LNCS, vol. 3797, pp. 118–129. Springer, Heidelberg (2005)
11. Gomez, D., Gutierrez, J., Ibeas, Á.A.: Attacking the Pollard generator. IEEE Transactions on Information Theory 52(12), 5518–5523 (2006)
12. Herrmann, M., May, A.: Attacking Power Generators Using Unravelled Linearization: When Do We Output Too Much? In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 487–504. Springer, Heidelberg (2009)
13. Howgrave-Graham, N.: Finding Small Roots of Univariate Modular Equations Revisited. In: Darnell, M.J. (ed.) Cryptography and Coding 1997. LNCS, vol. 1355, pp. 131–142. Springer, Heidelberg (1997)
14. Jochemsz, E., May, A.: A Strategy for Finding Roots of Multivariate Polynomials with New Applications in Attacking RSA Variants. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 267–282. Springer, Heidelberg (2006)
15. Joux, A., Stern, J.: Lattice reduction: A toolbox for the cryptanalyst. Journal of Cryptology 11(3), 161–185 (1998)
16. Lenstra, A.K., Lenstra, H.W.J., Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann. 261(4), 515–534 (1982)
17. Stern, J.: Secret linear congruential generators are not cryptographically secure. In: FOCS, pp. 421–426. IEEE (1987)

Extended-DDH and Lossy Trapdoor Functions

Brett Hemenway[1] and Rafail Ostrovsky[2,⋆]

[1] University of Michigan
bhemen@umich.edu
[2] UCLA

Abstract. Lossy Trapdoor Functions (LTFs) were introduced by Peikert and Waters in STOC '08 and since then have found many applications and have proven to be an extremely useful and versatile cryptographic primitive. Lossy trapdoor functions were used to build the first injective trapdoor functions based on DDH, the first IND-CCA cryptosystems based on lattice assumptions, and they are known to imply deterministic encryption, collision resistant hash-functions, oblivious transfer and a host of other important primitives. While LTFs can be instantiated under most known cryptographic hardness assumptions, no constructions until today existed based on generic cryptographic primitives. In this work, we show that any Homomorphic Smooth Hash Proof System, introduced by Cramer and Shoup in EUROCRYPT '02, can be used to construct LTFs. In addition to providing a connection between two important cryptographic primitives – our construction implies the first construction of LTFs based on the QR assumption.

Smooth Hash Proof Systems (SHPs) can be seen as a generalization of the DDH assumption, yet can be built on other cryptographic assumptions, such as the DCR or QR assumptions. Yet, until today, a "translation" of results proven secure under DDH to results under DCR or QR has always been fraught with difficulties. Thus, as our second goal of this paper, we ask the following question: is it possible to streamline such translations from DDH to QR and other primitives? Our second result formally provides this connection. More specifically, we define an Extended Decisional Diffie Hellman (EDDH) assumption, which is a simple and natural generalization of DDH. We show that EDDH can be instantiated under both the DCR and QR assumptions. This gives a much simpler connection between the DDH and the DCR and QR assumptions and provides an easy way to translate proofs from DDH to DCR or QR.

⋆ R. Ostrovsky, University of California Los Angeles, Department of Computer Science and Department of Mathematics, 3732D Boelter Hall, Los Angeles CA 90095-1596, U.S., email: rafail@cs.ucla.edu. Supported in part by NSF grants 0830803, 09165174, 1065276, 1118126 and 1136174, US-Israel BSF grant 2008411, OKAWA Foundation Research Award, IBM Faculty Research Award, Xerox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, and Lockheed-Martin Corporation Research Award. This material is based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11-1-0392. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 627–643, 2012.

That is, the advantage of the EDDH assumption is that most schemes (including LTFs) proven secure under the DDH assumption can easily be instantiated under the DCR and QR assumptions with almost no change to their proofs of security.

1 Introduction

The first practical IND-CCA secure cryptosystem was built by Cramer and Shoup under the Decisional Diffie-Hellman (DDH) assumption [CS98]. In a follow up work, Cramer and Shoup introduced projective hash proofs as a means of generalizing their original DDH-based construction [CS02]. This generalization allowed them to create unified constructions of IND-CCA secure cryptosystems based on Paillier's Decisional Composite Residuosity (DCR) assumption and the Quadratic Residuosity (QR) assumption.

Since their introduction, projective hash proof systems have proven to be an effective tool for generalizing constructions that were originally proven secure under the DDH assumption. Indeed, many important results use the framework of projective hash proofs to take a system built using the DDH assumption and instantiate it using the DCR or QR assumptions.

Cramer and Shoup [CS02] converted the DDH-based construction of IND-CCA encryption [CS98] to one based on the DCR or QR assumptions. Kalai and Halevi [Kal05, HK07] converted the DDH-based construction of OT given by Naor and Pinkas [NP01] to one based on the DCR or QR assumptions. Brakerski and Goldwasser [BG10] converted the DDH-based construction of circular secure encryption given by Boneh, Halevi, Hamburg and Ostrovsky [BHHO08] to one based on the DCR or QR assumptions[1].

This series of works generalizing DDH-based constructions suggests the heuristic that "anything that can be done with DDH can be done with DCR or QR." Like any heuristic it is not completely accurate, but it appears to provide the right intuition.

While projective hash proof systems suggest a means for converting a DDH-based scheme to a DCR or QR based scheme, the generality of projective hash proof systems framework often means that converting the actual proofs of security can be fairly technical. This is evidenced in the works of [CS02, Kal05, HK07, BG10] which provided significant technical contributions beyond the original constructions of [CS98, NP01, BHHO08].

This work makes two contributions: First, we show that Lossy Trapdoor Functions (LTFs) of Peikert and Waters [PW08] can be built under general assumptions, namely any homomorphic smooth hash proof system. This provides a connection between two important cryptographic primitives. Second, we introduce the *Extended Decisional Diffie-Hellman (EDDH)* assumption, and show how it can be instantiated using the DCR and QR assumptions. This second result provides a justification for the heuristic noted above that the DCR and

[1] Brakerski and Goldwasser did not explicitly use the language of projective hash proofs, but their construction fits the framework exactly.

QR assumptions "imply" the DDH assumption. While the EDDH assumption does not appear to be as general as the notion of projective hash proof systems, its simplicity gives it some advantages. In particular, the EDDH assumption provides a much simpler method for identifying which DDH-based constructions can be instantiated under the DCR or QR assumptions, and proofs of security under the EDDH assumption are almost identical to those under the DDH assumption. Using the framework of EDDH, it becomes almost immediate that the DDH constructions of [NP01, BHHO08, PW08] can be instantiated under the DCR or QR assumptions with almost no modifications to the proofs of security.

As mentioned above, our first result is a construction of lossy trapdoor functions (LTFs) from general assumptions. Lossy trapdoor functions were introduced by Peikert and Waters [PW08]. LTFs provided the first injective trapdoor functions based on the Decisional Diffie-Hellman (DDH) assumption, and the first chosen ciphertext (IND-CCA) secure cryptosystem based on lattice assumptions. In addition to providing natural constructions of injective trapdoor functions and IND-CCA secure cryptosystems, Peikert and Waters went on to show that LTFs provide very natural constructions of many cryptographic primitives, including pseudo-random generators, collision-resistant hash functions, and oblivious transfer. The extremely intuitive nature of these many constructions provided early evidence of the value of LTFs as a cryptographic primitive. Since the original work of Peikert and Waters, lossy trapdoor functions have been shown to imply many other important cryptographic primitives. In [BFO08], Boldyreva, Fehr and O'Neill showed that LTFs imply deterministic encryption. Deterministic encryption was introduced in [BBO07], and captures the strongest notion of security possible for a deterministic function. In contrast to one-way functions, which do leak the parity of a random subset of the bits of its input [GL89], deterministic encryption does not leak *any fixed function*[2] of its input. Deterministic encryption has applications to efficiently searchable encryption, and securing legacy systems. Lossy trapdoor functions were then shown to imply correlated product secure functions by Rosen and Segev in [RS09]. Roughly a family of correlated product secure functions is a family of functions that remain one-way even when the output of multiple functions is given *on the same input*. In [MY09], Mol and Yilek introduced a relaxation of lossy trapdoor functions called *slightly lossy trapdoor functions*, and showed that even slightly lossy trapdoor functions are sufficient to achieve correlated product secure functions. Lossy functions, (without the need for a trapdoor) have been shown to imply leaky pseudo-entropy functions [BHK11].

Lossy trapdoor functions have been constructed from a variety of concrete hardness assumptions. In [PW08], Peikert and Waters constructed LTFs from the DDH assumption and lattice assumptions, and an efficient construction of LTFs from Paillier's Decisional Composite Residuosity (DCR) assumption was given independently in [BFO08] and [RS08]. In concurrent, independent work, Freeman et al. [FGK+10] give constructions of LTFs from the D-Linear Assumption and constructions of slightly lossy trapdoor functions from the QR assumption.

[2] Independent of the choice of the key for the deterministic encryption.

While we have seen a wide variety of important consequences of lossy trapdoor functions, there remains a lack of general constructions. This work provides the first constructions of LTFs from generic primitives (in this case homomorphic smooth hash proof systems, and diverse group systems) as well as the first construction of fully lossy trapdoor functions from the well-known Quadratic Residuosity (QR) assumption.

This result has a number of other consequences. Applying our construction to the results of [BFO08], we achieve the first construction of deterministic encryption from smooth homomorphic hash proof systems. Applying our results to those of [RS09], we give the only known construction of correlated product secure functions from a generic primitive other than lossy trapdoor functions,[3] and the first known construction of correlated product secure functions from the QR assumption.[4] Applying the separation of Rosen and Segev, we provide a black-box separation of smooth homomorphic hash proof systems and one-way trapdoor permutations.

The second contribution of this work is a development of the connection between the DDH, DCR and QR assumptions. Projective hash proof systems [CS02] showed that many properties of DDH-based protocols could be achieved using the DCR or QR assumptions. In this work, we introduce the Extended DDH (EDDH) assumption, and show how the EDDH assumption is implied by the DDH, DCR and QR assumptions. One formulation of the DDH assumption is that the distributions $\{g, g^a, g^b, g^{ab}\}$, $\{g, g^a, g^b, g^c\}$ are computationally indistinguishable. Equivalently, $\{g, g^a, g^b, g^{ab}\} \approx_c \{g, g^a, g^b, g^{ab}r\}$ for some uniformly chosen element r in the group. The EDDH assumption is the same, except that r is chosen from a subgroup instead of the entire group. Thus the EDDH assumption states that $\{g, g^a, g^b, g^{ab}\}$ and $\{g, g^a, g^b, g^{ab}r\}$ are computationally indistinguishable when r is chosen uniformly from a given subgroup of the universe group. See Definition 6 for the formal definition. The value of the EDDH assumption is that it provides a very simple method for converting constructions based on the DDH assumption into constructions which can be proven secure under the DCR or QR assumptions. Since the semantics of the EDDH assumption are very similar to those of the DDH assumption in many cases proofs of security under the DDH assumption go through almost unchanged under the EDDH assumption.

1.1 Previous Work

Lossy Trapdoor Functions (LTFs) were introduced by Peikert and Waters in [PW08], simultaneously providing the first construction of one-way trapdoor

[3] There are two concrete constructions of correlated product secure functions that are not lossy trapdoor functions. A construction based on the Learning With Error (LWE) problem given by Peikert in [Pei09], and a construction based on the hardness of syndrome decoding given by Freeman et al. in [FGK+10].

[4] A completely different construction of correlated product secure functions from the QR assumption is given in the concurrent, independent work of Freeman et al. [FGK+10].

functions from the Decisional Diffie Hellman and the first IND-CCA secure cryptosystem based on lattice assumptions.

Roughly, a family of lossy trapdoor functions is a family of functions with two computationally indistinguishable branches. An injective branch with a trapdoor, and a lossy branch which statistically loses information about its input, in particular the image size of the lossy branch is required to be much smaller than its domain size. If the lossy branch is lossy enough, this immediately implies that the injective branch is an injective one-way trapdoor function. Peikert and Waters gave constructions of lossy trapdoor functions from the DDH assumption and lattice-based assumptions. In [BFO08], [RS08], Boldyreva et al. and Rosen and Segev gave efficient constructions of lossy trapdoor functions from Paillier's DCR assumption. A construction of lossy trapdoor functions from the D-Linear assumption, and slightly lossy trapdoor functions from the QR assumption are given in the concurrent, independent work of [FGK+10].

Lossy trapdoor functions are known to imply IND-CCA secure encryption. In addition to IND-CCA secure encryption, LTFs were shown to imply collision-resistant hash functions [PW08], deterministic encryption [BFO08], lossy encryption [PVW08] and correlated product secure functions [RS09].

Projective Hash Proof Systems were introduced by Cramer and Shoup in [CS02], generalizing their construction of IND-CCA encryption from the Decisional Diffie-Hellman (DDH) assumption given in [CS98]. In [CS02], Cramer and Shoup defined two types of hash proof systems, smooth projective hash families, which immediately implied IND-CPA secure encryption, and universal hash families, which could be used as a type of designated verifier proof system for the specific class of language given by smooth projective hash families. They went on to show that universal hash proof systems imply smooth projective hash proof systems, so it was sufficient to construct only universal hash proof systems. Their general construction, however, was fairly inefficient, and in all of their constructions they were able to avoid the general construction of smooth projective hash proof systems, and create efficient smooth projective hash proof systems directly. In this work, we will deal only with smooth projective hash proof systems.

In order to construct explicit hash proof systems, Cramer and Shoup defined another primitive called a *Diverse Group System*. Diverse Group Systems seemed to capture the essential part of the algebraic structure of a cyclic group, and they gave a very natural construction of projective hash proof systems from Diverse Group Systems. They went on to construct diverse group systems from the DDH assumption, the Quadratic Residuosity (QR) assumption and the Decisional Composite Residuosity (DCR) assumption.

The first result of this work is a proof that smooth homomorphic hash proof systems imply lossy trapdoor functions. By providing a link between smooth homomorphic hash proof systems, and lossy trapdoor functions, we provide a number of new connections as well. This work provides the first construction of lossy trapdoor functions from a generic primitive. Additionally, it provides

the first construction of deterministic encryption from smooth homomorphic projective hash proof systems.

Our first result uses the framework of smooth projective hashing to generalize the DDH-based construction of LTFs from [PW08]. Smooth projective hash proof systems have been used to generalize DDH-based constructions in the past. Kalai and Halevi [Kal05, HK07] used them to generalize Naor and Pinkas's OT protocol [NP01], and Brakerski and Goldwasser [BG10] generalized the circular secure encryption of Boneh, Halevi, Hamburg and Ostrovsky [BHHO08] using the same framework. This series of results indicates a close relationship between the DDH, DCR and QR assumptions.

The second result of this work is a development of the connection between the DDH, DCR and QR assumptions. One of the most useful features of projective hash proof systems is that they provide a framework for converting cryptographic schemes designed under the DDH assumption into cryptographic schemes that are provably secure under the DCR or QR assumptions. While projective hash proof systems showed a close connection between the DDH, DCR and QR assumptions, generality of projective hash proof systems makes this connection difficult to see. To make the connection between these three hardness assumptions clearer, we introduce the EDDH assumption and show how it can be realized under the DCR and QR assumptions. The benefit of the EDDH assumption is that it is semantically very similar to the DDH assumption, so many existing constructions whose security rests on the DDH assumption (including the construction of LTFs by Peikert and Waters) can immediately be instantiated under the DCR or QR assumptions. In particular, we note that the proof of [PW08] can be instantiated using the EDDH assumption. This gives a novel construction of LTFs from the DCR assumption and the first construction of LTFs from the QR assumption.

1.2 Our Contributions

In this work, we show that smooth homomorphic hash proof systems imply lossy trapdoor functions (LTFs). It was shown in [BFO08] that lossy trapdoor functions imply deterministic encryption, so our results give the first construction of deterministic encryption from smooth homomorphic hash proof systems.

In [RS09], Rosen and Segev introduced correlated product secure functions, and showed that lossy trapdoor functions are correlated product secure. Applying their results to our construction, we have a construction of correlated product secure functions from smooth homomorphic hash proof systems. Finally, combining our results with the black-box separations of Rosen and Segev [RS09], we find that there is a black-box separation between one-way trapdoor permutations and smooth homomorphic hash proof systems.

Our primary results are summarized as follows:

Theorem. *Smooth Homomorphic Projective Hash Proof Systems imply Lossy Trapdoor Functions.*

This theorem has a number of immediate Corollaries. Since Boldyreva et al. [BFO08] showed that LTFs imply deterministic encryption (as defined in [BBO07]), we have

Corollary. *Smooth Homomorphic Projective Hash Proof Systems imply deterministic encryption.*

Since Rosen and Segev [RS09] showed that LTFs imply correlated product secure encryption, and a black-box separation between one-way trapdoor permutations and lossy trapdoor functions, we have

Corollary. *Smooth Homomorphic Projective Hash Proof Systems imply correlated product secure functions.*

Corollary. *There is a black-box separation between Smooth Homomorphic Projective Hash Proof Systems and one-way trapdoor permutations, i.e. there exists an oracle, relative to which the latter exists but the former does not.*

In addition to the new constructions outlined above, in Section 4 we introduce the Extended Decisional Diffie Hellman (EDDH) assumption, which provides a simple way to achieve a DDH-like property under the DCR and QR assumptions. This serves to unify many of the previous constructions (e.g. [NP01] and [Kal05, HK07], [BHHO08] and [BG10]), and provides a more familiar alternative to projective hash proof systems.

Applying these results yields lossy trapdoor functions from the DDH, DCR and QR assumptions. When applied to DDH, the construction achieved in this way is identical to the construction of LTFs given by Peikert and Waters in [PW08], however the constructions from the DCR and QR assumptions are new. While our construction of LTFs from the DCR assumption is less efficient than that given by [BFO08] and [RS08], our results provide the first construction of lossy trapdoor functions from the QR assumption.

2 Preliminaries

2.1 Notation

If A is a Probabilistic Polynomial Time (PPT) machine, then we use $a \xleftarrow{\$} A$ to denote running the machine A and obtaining an output, where a is distributed according to the internal randomness of A. If R is a set, we use $r \xleftarrow{\$} R$ to denote sampling uniformly from R.

We use the notation

$$\Pr[r \xleftarrow{\$} R; x \xleftarrow{\$} X : A(x, r) = c],$$

to denote the probability that A outputs c when x is sampled uniformly from X and r is sampled uniformly from R. We define the statistical distance between two distributions X, Y to be

$$\Delta(X, Y) = \frac{1}{2} \sum_{x} |\Pr[X = x] - \Pr[Y = x]|$$

If X and Y are families of distributions indexed by a security parameter λ, we use $X \approx_s Y$ to mean the distributions X and Y are statistically close, *i.e.*, for all polynomials p and sufficiently large λ, we have $\Delta(X,Y) < \frac{1}{p(\lambda)}$. We use $X \approx_c Y$ to mean X and Y are computationally close, *i.e.*, for all PPT adversaries A, for all polynomials p, then for all sufficiently large λ, we have $|\Pr[A^X = 1] - \Pr[A^Y = 1]| < 1/p(\lambda)$.

2.2 Lossy Trapdoor Functions

We briefly recall the definition of lossy trapdoor functions given in [PW08].

A tuple $(S_{\text{ltdf}}, F_{\text{ltdf}}, F^{-1}_{\text{ltdf}})$ of PPT algorithms is called a family of (n,k)-Lossy Trapdoor Functions if the following properties hold:

- **Sampling Injective Functions:** $S_{\text{ltdf}}(1^\lambda, 1)$ outputs s,t where s is a function index, and t its trapdoor. We require that $F_{\text{ltdf}}(s,\cdot)$ is an injective deterministic function on $\{0,1\}^n$, and $F^{-1}_{\text{ltdf}}(t, F_{\text{ltdf}}(s,x)) = x$ for all x.
- **Sampling Lossy Functions:** $S_{\text{ltdf}}(1^\lambda, 0)$ outputs $(s, \perp)$ where s is a function index and $F_{\text{ltdf}}(s,\cdot)$ is a function on $\{0,1\}^n$, where the image of $F_{\text{ltdf}}(s,\cdot)$ has size at most 2^{n-k}.
- **Indistinguishability:** The first outputs of $S_{\text{ltdf}}(1^\lambda, 0)$ and $S_{\text{ltdf}}(1^\lambda, 1)$ are computationally indistinguishable.

2.3 Subset Membership Problems

In this section we recall the definition of of a subset membership problem as formalized in [CS02]. Roughly, given sets $L \subset X$, we want L and X to be computationally indistinguishable.

Formally, given a family of sets (X, L, W) indexed by a security parameter λ, we require $L \subset X$, and there is a binary relation $\mathcal{R} : X \times W \to \{0,1\}$. If $\mathcal{R}(x,w) = 1$, we say that w is a witness for x. In this work, we will restrict our attention to relations $\mathcal{R}$ such that for all $x \in L$, there exists a $w \in W$ such that $\mathcal{R}(x,w) = 1$, and for all $x \notin L$, and all $w \in W$, $\mathcal{R}(x,w) = 0$.

We also need the following efficient sampling algorithms.

- **Instance Sampling:** Given a security parameter λ, we can sample (X, L, W) and $\mathcal{R}$.
- **Sampling Without Witness:** Given (X, L, W) we can sample (statistically-close to) uniformly on X.
- **Sampling With Witness:** Given (X, L, W) we can sample x (statistically-close to) uniformly on L, along with a witness w such that $\mathcal{R}(x,w) = 1$.

Definition 1. *A subset membership problem is called* hard *if for all PPT distinguishers,*

$$|\Pr[x \xleftarrow{\$} X : D(x) = 1] - \Pr[x \xleftarrow{\$} L : D(x) = 1]| < \nu(\lambda),$$

for some negligible function ν.

As in [CS02], the security of all of our constructions will rely on the security of some underlying hard subset membership problem. In fact, the hardness assumptions DDH, DCR and QR all have natural formulations in terms of hard subset membership problems [CS02].

2.4 Smooth Hash Proof Systems

We briefly recall the notion of *smooth projective hash* families as defined by Cramer and Shoup in [CS02]. Let H be a function family indexed by keys in the a keyspace K, *i.e.* for each $k \in K$, $H_k : X \to \Pi$. Let $L \subset X$ and a "projection" $\alpha : K \to S$. We require efficient evaluation algorithms such that, for any $x \in X$, $H_k(x)$ is efficiently computable using $k \in K$. Using the terminology of [CS02], this is called the *private evaluation algorithm*. Finally we require efficient sampling algorithms to sample uniformly from X, uniformly from K, and uniformly from L *along with a witness*. The security properties of the system will follow from the indistinguishability of X and L.

Definition 2. *The set* $\mathsf{HPS} = (H, K, X, L, \Pi, S, \alpha)$ *is a* projective hash family *if, for all* $k \in K$*, the action of* H_k *on the subset* L *is completely determined by* $\alpha(k)$*.*

For a projective hash family, $\alpha(k)$ determines the output of H_k on L. Additionally, if $x \in L$ and a witness w for $x \in L$ is known, then we require that $H_k(x)$ is efficiently computable given $x, w, \alpha(k)$. This is called the *public evaluation algorithm*. A *smooth* projective hash family is one in which α does not encode any information about the action of H_k on $X \setminus L$.

Definition 3. *Let* $(H, K, X, L, \Pi, S, \alpha)$ *be a projective hash family, and define two distributions* Z_1, Z_2 *taking values on the set* $X \setminus L \times S \times \Pi$*. For* Z_1*, we sample* $k \xleftarrow{\$} K$*,* $x \xleftarrow{\$} X \setminus L$*, and set* $s = \alpha(k)$*,* $\pi = H_k(x)$*, for* Z_2 *we sample* $k \xleftarrow{\$} K$*,* $x \xleftarrow{\$} X \setminus L$*, and* $\pi \xleftarrow{\$} \Pi$*, and set* $s = \alpha(k)$*. The projective hash family is called* ν-smooth *if* $\Delta(Z_1, Z_2) < \nu$*.*

This means that, given $\alpha(k)$ and $x \in X \setminus L$, $H_k(x)$ is statistically close to uniform on Π.

In [CS02], they showed that smooth projective hash families immediately imply IND-CPA secure encryption by taking $sk = k$, $pk = \alpha(k)$, and to encrypt a message $m \in \Pi$, we sample $x \in L$ along with randomness and output $E(m) = (x, H_k(x) + m)$.

We extend the definition of smooth projective hash proof systems slightly

Definition 4. *If* $\mathsf{HPS} = (H, K, X, L, \Pi, S, \alpha)$ *is a projective hash family, we say that* HPS *is a* homomorphic projective hash family *if* X *is a group, and for all* $k \in K$*, and* $x_1, x_2 \in X$*, we have* $H_k(x_1) + H_k(x_2) = H_k(x_1 + x_2)$*, that is to say* H_k *is a homomorphism for each* k*.*

In [CS02] Cramer and Shoup provide smooth homomorphic projective hash families based on the DDH, DCR and QR assumptions.

3 Lossy Trapdoor Functions from Smooth Homomorphic Hash Proof Systems

Peikert and Waters [PW08] gave a construction of lossy trapdoor functions from the Decisional Diffie-Hellman (DDH) assumption. In this section, we show that a similar construction goes through with smooth homomorphic hash proof systems. This extends the intuition given in [CS02] that projective hashing provides a good generalization of the DDH assumption. We note, however, that although our construction is very similar that of [PW08], the proofs of security are quite different.

Let (X, L, W) be a hard subset membership problem. For notational convenience, we suppress the dependence on the security parameter λ. Let $\mathbf{H} = (H, K, X, L, \Pi, S, \alpha)$ be an associated smooth homomorphic projective hash family.

- **Key Generation:**
  Pick $x_1, \ldots, x_n \in L$.
  Fix $b \in \Pi \setminus \{0\}$.
  Generate the matrix $B = (B_{ij}) \subset \Pi^{n \times n}$, where $B_{ij} = 0$ if $i \neq j$, and
  In lossy mode $B_{ii} = 0$ for all i.
  In injective mode $B_{ii} = b$.

  Sample $k_1, \ldots, k_n \leftarrow K$, and output

$$R = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \qquad A = \begin{pmatrix} H_{k_1}(x_1) + B_{11} & \cdots & H_{k_1}(x_n) + B_{1n} \\ \vdots & \ddots & \vdots \\ H_{k_n}(x_1) + B_{n1} & \cdots & H_{k_n}(x_n) + B_{nn} \end{pmatrix}$$

  The trapdoor will be $(k_1, \ldots, k_n)$.

- **Evaluation:**
  Given a message $z = z_1, \ldots, z_n \in \{0,1\}^n$
  Given a function index R, A, calculate

$$F_{R,A}(z) = (Rz, Az) = \left(\sum_{i=1}^{n} z_i x_i, \begin{pmatrix} \sum_{i=1}^{n} z_i (H_{k_1}(x_i) + B_{1i}) \\ \vdots \\ \sum_{i=1}^{n} z_i (H_{k_n}(x_i) + B_{ni}) \end{pmatrix} \right).$$

- **Trapdoor:**
  Given a value (Rz, Az), and a trapdoor $(k_1, \ldots, k_n)$, we begin by noting that the homomorphic property of H_k guarantees that

$$F_{R,A}(z) = (Rz, Az) = \left(\sum_{i=1}^{n} z_i x_i, \begin{pmatrix} \sum_{i=1}^{n} z_i (H_{k_1}(x_i) + B_{1i}) \\ \vdots \\ \sum_{i=1}^{n} z_i (H_{k_n}(x_i) + B_{ni}) \end{pmatrix} \right)$$

$$= \left(\sum_{i=1}^{n} z_i x_i, \begin{pmatrix} H_{k_1}\left(\sum_{i=1}^{n} z_i x_i\right) + \sum_{i=1}^{n} z_i B_{1i}) \\ \vdots \\ H_{k_n}\left(\sum_{i=1}^{n} z_i x_i\right) + \sum_{i=1}^{n} z_i B_{ni}) \end{pmatrix} \right)$$

Since $\sum_{i=1}^{n} z_i x_i$, and k_i is known, we can calculate $H_{k_i}\left(\sum_{i=1}^{n} z_i x_i\right)$ and subtract it from each component to recover the vector

$$\left(\sum_{i=1}^{n} z_i B_{1i}, \cdots, \sum_{i=1}^{n} z_i B_{ni} \right)^t .$$

Now, in injective mode, $B_{ij} = 0 \in \Pi$ for $i \neq j$, and $B_{ij} = b$ for $i = j$, so

$$\left(\sum_{i=1}^{n} z_i B_{1i}, \cdots, \sum_{i=1}^{n} z_i B_{ni} \right)^t = (z_1 b, \cdots, z_n b) .$$

Since the $z_i \in \{0, 1\}$, and since b is known, we can recover the z_i by inspection.

Remark: Notice that we do not make use of the projection α in our construction, it will appear, however, in the proof of security. Unlike in [CS02], we do not require that α be efficiently computable, merely that it exists.

We now examine the security of this construction.

Lemma 1. *In Lossy Mode, the image of F has size at most $|X|$.*

Proof. Notice that in Lossy Mode, since $B_{ij} = 0$ for all i, j,

$$F_{R,A}(z) = \left(\sum_{i=1}^{n} z_i x_i, \begin{pmatrix} H_{k_1}\left(\sum_{i=1}^{n} z_i x_i\right)) \\ \vdots \\ H_{k_n}\left(\sum_{i=1}^{n} z_i x_i\right)) \end{pmatrix} \right)$$

which depends only on the sum $\sum_{i=1}^{n} z_i x_i \in X$. Thus the size of the image is bounded by $|X|$.

Thus by taking $n > \log(|X|)$, we can make the lossy mode of F as lossy as desired.

Lemma 2. *The Injective and Lossy Modes are computationally indistinguishable.*

The proof can be found in the full version of this work. We remark that this construction *does not* make use of the projection α. The projective property *is* used, however, since we condition on $H_k(x)$ for $x \in L$, which leaves at least as much entropy in k as conditioning on $\alpha(k)$, since $\alpha(k)$ determines $H_k(x)$.

A similar construction and proof goes through for Diverse Group Systems (see the full version of this work for details). Thus we arrive at

Theorem 1. *Smooth Homomorphic Projective Hash Proof Systems imply Lossy Trapdoor Functions, and Diverse Group Systems imply Lossy Trapdoor Functions.*

This theorem has a number of immediate Corollaries. Since Boldyreva et al. [BFO08] showed that LTFs imply deterministic encryption (as defined in [BBO07]), we have Corollary 1. Since Rosen and Segev [RS09] showed that LTFs imply correlated product secure encryption, we have Corollary 2. Since Rosen and Segev showed a black-box separation between one-way trapdoor permutations and lossy trapdoor functions, we have Corollary 3.

Corollary 1. *Smooth Homomorphic Projective Hash Proof Systems imply deterministic encryption.*

Corollary 2. *Smooth Homomorphic Projective Hash Proof Systems imply correlated product secure functions.*

Corollary 3. *There is a black-box separation between Smooth Homomorphic Projective Hash Proof Systems and one-way trapdoor permutations, i.e. there exists an oracle, relative to which the latter exists but the former does not.*

4 The Extended DDH Assumption

In this section, we introduce the Extended Decisional Diffie Hellman (EDDH) assumption. Let $\mathbb{G}$ be commutative group (written multiplicatively). The DDH assumption states that

Definition 5 (The DDH Assumption). *Assume $\mathbb{G}$ is a group with an efficient sampling algorithm, and $K = \{1, \ldots, |\mathbb{G}|\}$. Then the DDH assumption states that*

$$\{(g, g^a, g^b, g^{ab}) : g \xleftarrow{\$} G, a, b \xleftarrow{\$} K\} \approx_c \{(g, g^a, g^b, g^c) : g \xleftarrow{\$} G, a, b, c \xleftarrow{\$} K, \}$$

When $\mathbb{G}$ is a cyclic group, this can be rephrased as

$$\{(g, g^a, g^b, g^{ab}) : g \xleftarrow{\$} G, a, b \xleftarrow{\$} K\} \approx_c \{(g, g^a, g^b, g^{ab}h) : g \xleftarrow{\$} G, a, b \xleftarrow{\$} K, h \xleftarrow{\$} \mathbb{G}\}$$

We introduce a slight modification of the DDH assumption, called the *Extended Decisional Diffie Hellman (EDDH)* assumption.

Definition 6 (The EDDH Assumption). *For a group $\mathbb{G}$, and a (samplable) subgroup $\mathbb{H} \triangleleft \mathbb{G}$, the* extended decisional diffie hellman (EDDH) *problem is said to be hard if there exists a samplable set $G \subset \mathbb{G}$ and samplable sets $K \subset \mathbb{Z}$ such that the following two distributions are computationally indistinguishable:*

$$\{(g, g^a, g^b, g^{ab}) : g \xleftarrow{\$} G, a, b \xleftarrow{\$} K\} \approx_c \{(g, g^a, g^b, g^{ab}h) : g \xleftarrow{\$} G, a, b \xleftarrow{\$} K, h \xleftarrow{\$} \mathbb{H}\}$$

It is not hard to see:

Lemma 3. *If $K = \{1, \ldots, |\mathbb{G}|\}$, and $\mathbb{H} = \mathbb{G}$, then the EDDH assumption is just the DDH assumption in the group $\mathbb{G}$.*

The utility of this assumption is that it extracts the essential properties of the DDH assumption, yet it can be instantiated under the QR assumption and the DCR assumption. See the full version of this work for example applications of the EDDH assumption.

We begin by showing that the DCR assumption [Pai99] implies the EDDH assumption.

Theorem 2 (DCR implies EDDH). *Let p, q be safe primes*[5] *and define:*

- $N = pq$,
- $\mathbb{G} = \{x : x \xleftarrow{\$} \mathbb{Z}^*_{N^2}, \left(\frac{x}{N}\right) = 1\}$,
- $G = \{g^{2N} \mod N^2 : g \xleftarrow{\$} \mathbb{Z}_{N^2}\}$,
- $K = \{0, \ldots, \lfloor N^2/4 \rfloor\} = \{0, \ldots, (N^2-1)/4\}$,
- $\mathbb{H} = \{(1+aN) : a \in \mathbb{Z}_N\} = \{(1+N)^a \mod N^2 : a \in \mathbb{Z}_N\}$.

Then under the DCR assumption the EDDH assumption is hard in the group $\mathbb{G}$.

Proof. Define the following distributions Let $\hat{G} = \{g^{2N}(1+N) \mod N^2 : g \xleftarrow{\$} \mathbb{Z}_{N^2}\}$.

$$\begin{aligned}
\Lambda_1 &= \{(g, g^a, g^b, g^{ab}) : g \xleftarrow{\$} G, a \xleftarrow{\$} K, b \xleftarrow{\$} K\} \\
\Lambda_2 &= \{(g, x, g^b, x^b) : g \xleftarrow{\$} G, x \xleftarrow{\$} \hat{G}, b \xleftarrow{\$} K\} \\
\Lambda_3 &= \{(g, x, g^b, x^b h) : g \xleftarrow{\$} G, x \xleftarrow{\$} \hat{G}, b \xleftarrow{\$} K, h \xleftarrow{\$} \mathbb{H}\} \\
\Lambda_4 &= \{(g, g^a, g^b, g^b h) : g \xleftarrow{\$} G, a \xleftarrow{\$} K, b \xleftarrow{\$} K, h \xleftarrow{\$} \mathbb{H}\}
\end{aligned}$$

1. The DCR assumption says $\{g^2 \mod N^2 : g \xleftarrow{\$} \mathbb{Z}_{N^2}\} \approx_c \{g^{2N} \mod N^2 : g \xleftarrow{\$} \mathbb{Z}_{N^2}\}$. Thus

$$\begin{aligned}
G &= \{g^{2N} \mod N^2 : g \xleftarrow{\$} \mathbb{Z}_{N^2}\} \\
&\approx_c \{g^2 \mod N^2 : g \xleftarrow{\$} \mathbb{Z}_{N^2}\} \\
&= \{g^2(1+N) \mod N^2 : g \xleftarrow{\$} \mathbb{Z}_{N^2}\} \\
&\approx_c \{g^{2N}(1+N) \mod N^2 : g \xleftarrow{\$} \mathbb{Z}_{N^2}\} \\
&= \hat{G}.
\end{aligned}$$

   Now, notice that for a fixed generator g of G,

$$\{g^a \mod N^2 : a \xleftarrow{\$} K\} \approx_s \{g^a \mod N^2 : a \xleftarrow{\$} \{0, 1, \ldots, \varphi(N)/4\}\} \approx_s G$$

[5] Choosing p, q safe primes makes the analysis slightly simpler. See the full version of this work for a complete discussion.

(See the full version of this work for a rigorous proof of this fact). We also know that with all but negligible probability a uniformly chosen element $g \xleftarrow{\$} G$ will be a generator for G, so this implies $\Lambda_1 \approx_c \Lambda_2$.

2. If $x = g_1^{2N}(1+N)$, then $x^b = g_1^{2Nb}(1+N)^b = g_1^{2N(b \mod N\varphi(N)/4)}(1+N)^{b \mod N} \mod N^2$. Since the distribution of b is statistically close to uniform modulo $N\varphi(N)/4$, we have that b is statistically close to uniform modulo N even conditioned on any value of b modulo $\varphi(N)/4$. Since the order of g is $\varphi(N)/4$, the distribution of b modulo N is statistically close to uniform conditioned on g^b. Thus, even conditioned on g^b, the distribution of x^b is statistically close to $g_1 h$ where $g_1 \xleftarrow{\$} G$, and $h \xleftarrow{\$} \mathbb{H}$, which shows $\{(g, x, g^b, x^b)\} \approx_s \{(g, x, g^b, x^b h)\}$. Thus $\Lambda_2 \approx_s \Lambda_3$.
3. We have already observed that $G \approx_c \hat{G}$, so $\Lambda_3 \approx_c \Lambda_4$.

It is standard to conserve randomness by sampling $a \xleftarrow{\$} \{0, \ldots, (N-1)/4\}$, and $b \xleftarrow{\$} \{0, \ldots, (N^2-1)/4\}$. It is easy to see that security is preserved in this case as well. Since the exposition is cleaner if they are sampled from the same space, and a few DDH applications require it, our scheme samples them from the same larger space.

Next, we show that the QR assumption implies the EDDH assumption.

Theorem 3 (QR Implies EDDH). *Let p, q be safe primes with $p = q = 3 \mod 4$, and define:*

- $N = pq$,
- $\mathbb{G} = \{x : x \xleftarrow{\$} \mathbb{Z}_N^*, \left(\frac{x}{N}\right) = 1\}$,
- $G = \{g^2 \mod N : g \xleftarrow{\$} \mathbb{Z}_N\}$,
- $K = \{0, \ldots, \lfloor N/2 \rfloor\}$,
- $\mathbb{H} = \{\pm 1\}$.

Then under the QR assumption the EDDH assumption is hard in the group $\mathbb{G}$.

Proof. Since $p = q = 3 \mod 4$, -1 is a quadratic non-residue modulo N with jacobi symbol 1.

Define the following distributions

$$\Lambda_1 = \{(g, g^a, g^b, g^{ab}) : g \xleftarrow{\$} G, a \xleftarrow{\$} K, b \xleftarrow{\$} K\}$$
$$\Lambda_2 = \{(g, x, g^b, x^b) : g \xleftarrow{\$} G, x \xleftarrow{\$} \mathbb{G}, b \xleftarrow{\$} K\}$$
$$\Lambda_3 = \{(g, x, g^b, x^b h) : g \xleftarrow{\$} G, x \xleftarrow{\$} \mathbb{G}, b \xleftarrow{\$} K, h \xleftarrow{\$} \mathbb{H}\}$$
$$\Lambda_4 = \{(g, g^a, g^b, g^b h) : g \xleftarrow{\$} G, a \xleftarrow{\$} K, b \xleftarrow{\$} K, h \xleftarrow{\$} \mathbb{H}\}$$

1. The QR assumption says

$$\mathbb{G} = \{x : x \xleftarrow{\$} \mathbb{Z}_N^*, \left(\frac{x}{N}\right) = 1\} \approx_c \{g^2 \mod N : g \xleftarrow{\$} \mathbb{Z}_N\} = G$$

Now, notice that for a fixed generator g of G,

$$\{g^a \mod N : a \xleftarrow{\$} K\} \approx_s \{g^a \mod N : a \xleftarrow{\$} \{0, 1, \ldots, \varphi(N)/4\}\} \approx_s G$$

(See the full version for a rigorous proof of this fact.) We also know that with all but negligible probability a uniformly chosen element $g \xleftarrow{\$} G$ will be a generator for G, so this implies $\Lambda_1 \approx_c \Lambda_2$.

2. If $x = -g_1^2$, then $x^b = g_1^{2b}(-1)^b = g_1^{2(b \mod \varphi(N)/4)}(-1)^{b \mod 2} \mod N$. Since the distribution of b is statistically close to uniform modulo $\varphi(N)/2$, we have that b is statistically close to uniform modulo 2 even conditioned on any value of b modulo $\varphi(N)/4$. Since the order of g is $\varphi(N)/4$, the distribution of b modulo 2 is statistically close to uniform conditioned on g^b. Thus, even conditioned on g^b, the distribution of x^b is statistically close to $g_1 h$ where $g_1 \xleftarrow{\$} G$, and $h \xleftarrow{\$} \{\pm 1\}$, which shows $\{(g, x, g^b, x^b)\} \approx_s \{(g, x, g^b, x^b h)\}$. Thus $\Lambda_2 \approx_s \Lambda_3$.
3. We have already observed that $G \approx_c \mathbb{G}$, so $\Lambda_3 \approx_c \Lambda_4$.

As in the case of the DCR based schemes, it is standard to conserve randomness by sampling a from a smaller space than b. In particular, we can sample $a \xleftarrow{\$} \{0, \ldots, (N-1)/4\}$, and $b \xleftarrow{\$} \{0, \ldots, (N^2-1)/4\}$. For the reasons outlined above we present this simpler (though slightly less efficient) variant.

It is not too hard to see that the construction of LTFs given by Peikert and Waters in [PW08] carries through under the EDDH assumption. This immediately gives new constructions of LTFs based on the QR assumption and the DCR assumption. See the full version of this work for details.

This provides the first construction of full LTFs from the QR assumption, and a novel construction of LTFs from the DCR assumption.

5 Conclusion

In this work, we showed that the intuition that hash proof systems are a natural generalization of the Decisional Diffie-Hellman (DDH) assumption holds in the case of lossy trapdoor functions as well. In particular, we showed that the construction of lossy trapdoor functions from DDH given in [PW08] can be made to work with any smooth homomorphic projective hash (or any diverse group system). This shows an interesting connection between these two powerful primitives and provides the first generic[6] construction of lossy trapdoor functions from *any* primitive.

When applied to the results of [BFO08], we obtain the first construction of deterministic encryption from smooth homomorphic hash proof systems. Combining our work with the negative results of [RS09], we obtain a black-box separation between one-way trapdoor permutations and smooth homomorphic hash proof systems.

[6] i.e. not based on specific number theoretic assumptions

To reinforce the intuition that the DCR and QR assumptions can be used to replace the DDH assumption, we introduced the Extended Decisional Diffie Hellman (EDDH) assumption and showed that the DCR and QR assumptions imply the EDDH assumption. This provides a simple method for converting most DDH-based protocols into protocols whose security can be based on either the DCR or QR assumptions. In particular, this framework gives novel constructions of LTFs from the DCR assumption, and the first known constructions of fully lossy trapdoor functions from the QR assumption.

References

[BBO07] Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and Efficiently Searchable Encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 535–552. Springer, Heidelberg (2007)

[BFO08] Boldyreva, A., Fehr, S., O'Neill, A.: On Notions of Security for Deterministic Encryption, and Efficient Constructions without Random Oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 335–359. Springer, Heidelberg (2008)

[BG10] Brakerski, Z., Goldwasser, S.: Circular and Leakage Resilient Public-Key Encryption under Subgroup Indistinguishability. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 1–20. Springer, Heidelberg (2010)

[BHHO08] Boneh, D., Halevi, S., Hamburg, M., Ostrovsky, R.: Circular-Secure Encryption from Decision Diffie-Hellman. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 108–125. Springer, Heidelberg (2008)

[BHK11] Braverman, M., Hassidim, A., Kalai, Y.T.: Leaky pseudo-entropy functions. In: ICS 2011 (2011)

[CS98] Cramer, R., Shoup, V.: A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)

[CS02] Cramer, R., Shoup, V.: Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002); Full version available at `http://eprint.iacr.org` Cryptology ePrint Archive, Report 2001/085

[FGK+10] Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More Constructions of Lossy and Correlation-Secure Trapdoor Functions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 279–295. Springer, Heidelberg (2010)

[GL89] Goldreich, O., Levin, L.: A hard-core predicate for all one-way functions. In: STOC 1989, pp. 25–32. ACM (1989)

[HK07] Halevi, S., Kalai, Y.T.: Smooth projective hashing and two-message oblivious transfer. Cryptology ePrint Archive, Report 2007/118 (2007), `http://eprint.iacr.org/2007/118`

[Kal05] Kalai, Y.T.: Smooth Projective Hashing and Two-Message Oblivious Transfer. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 78–95. Springer, Heidelberg (2005)

[MY09] Mol, P., Yilek, S.: Chosen-ciphertext security from slightly lossy trapdoor functions (2009), `http://eprint.iacr.org/2009/524`

[NP01] Naor, M., Pinkas, B.: Efficient Oblivious Transfer Protocols. In: SODA 2001, pp. 448–457. ACM/SIAM (2001)

[Pai99] Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)

[Pei09] Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: STOC 2009: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, pp. 333–342. ACM, New York (2009)

[PVW08] Peikert, C., Vaikuntanathan, V., Waters, B.: A Framework for Efficient and Composable Oblivious Transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008)

[PW08] Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: STOC 2008: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, pp. 187–196. ACM, New York (2008)

[RS08] Rosen, A., Segev, G.: Efficient lossy trapdoor functions based on the composite residuosity assumption (2008), http://eprint.iacr.org/2008/134

[RS09] Rosen, A., Segev, G.: Chosen-Ciphertext Security via Correlated Products. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 419–436. Springer, Heidelberg (2009)

DDH-Like Assumptions Based on Extension Rings

Ronald Cramer[1], Ivan Damgård[2], Eike Kiltz[3], Sarah Zakarias[2], and Angela Zottarel[2,⋆]

[1] CWI and Leiden University
[2] Aarhus University
[3] RU Bochum

Abstract. We introduce and study a new type of DDH-like assumptions based on groups of prime order q. Whereas standard DDH is based on encoding elements of $\mathbb{F}_q$ "in the exponent" of elements in the group, we ask what happens if instead we put in the exponent elements of the extension ring $R_f = \mathbb{F}_q[X]/(f)$ where f is a degree-d polynomial. The decision problem that follows naturally reduces to the case where f is irreducible. This variant is called the d-DDH problem, where 1-DDH is standard DDH. We show in the generic group model that d-DDH is harder than DDH for $d > 1$ and that we obtain, in fact, an infinite hierarchy of progressively weaker assumptions whose complexities lie "between" DDH and CDH. This leads to a large number of new schemes because virtually all known DDH-based constructions can very easily be upgraded to be based on d-DDH. We use the same construction and security proof but get better security and moreover, the amortized complexity (e.g, computation per encrypted bit) is the same as when using DDH. We also show that d-DDH, just like DDH, is easy in bilinear groups. We therefore suggest a different type of assumption, the d-vector DDH problems (d-VDDH), which are based on $f(X) = X^d$, but with a twist to avoid problems with reducible polynomials. We show in the generic group model that d-VDDH is hard in bilinear groups and that the problems become harder with increasing d. We show that hardness of d-VDDH implies CCA-secure encryption, efficient Naor-Reingold style pseudorandom functions, and auxiliary input secure encryption. This can be seen as an alternative to the known family of k-LIN assumptions.

1 Introduction

The computational Diffie-Hellman assumption (CDH, proposed by Diffie and Hellman in [DH76]), says that if one chooses random g in a finite group $\mathbb{G}$ and random exponents a, b, then given g, g^a, g^b it is hard to compute g^{ab}. The assumption was introduced as basis for the well-known Diffie-Hellman key exchange.

⋆ The second, fourth and fifth author acknowledge support from the Danish National Research Foundation and The National Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the Theory of Interactive Computation, and also from the CFEM research center (supported by the Danish Strategic Research Council) within which part of this work was performed.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 644–661, 2012.

However, to get efficient cryptographic constructions one needs the stronger Decisional Diffie-Hellman assumption (DDH, studied by Naor and Reingold in [NR97]). It says that given g, g^a, g^b, the group element g^{ab} is pseudorandom, i.e., cannot be efficiently distinguished from g^c for a random c. In some groups, the DDH assumption is clearly false, but it is widely conjectured to hold when $\mathbb{G}$ is, for instance, a large prime order subgroup of $\mathbb{F}_p^*$ or an elliptic curve group.

DDH has been used as the basis for a very wide range of efficient cryptographic primitives, such as pseudorandom functions (PRF) [NR97], hash-proof systems and CCA-secure public-key encryption [CS98], leakage resilient cryptography (in particular, auxiliary input security [DGK+10]), and circular secure encryption [BHHO08].

Similar efficient constructions are not known under the weaker CDH assumption (unless one assumes random oracles) and this has motivated a large body of research studying weaker variants of DDH that would still enable cryptographic constructions. A well-known example is a family of assumptions called the k-LIN assumptions (where $k = 1$ is simply the standard DDH assumption) [BBS04, HK07, Kil07, Sha07]. In the generic group model, these assumptions are known to become progressively weaker for increasing k.

In this paper we initiate a study of a new family of assumptions that form natural extensions of DDH in prime order groups: if $\mathbb{G}$ has prime order q, and we fix a generator h, then an element $g \in \mathbb{G}$ "encodes" an element $a \in \mathbb{F}_q$ namely the a for which $g = h^a$. Intuitively we can think of a copy of $\mathbb{F}_q$ sitting in the exponent, and we can add field elements by multiplying in $\mathbb{G}$, and multiply by known constants by doing exponentiation. However, if CDH is hard, we cannot do general multiplication, i.e., compute g^{ab} from g^a, g^b. If DDH is hard, we cannot even distinguish the correct result from random. Now, let us instead consider the extension ring $R_f = \mathbb{F}_q[X]/(f)$ where f is a degree-d polynomial. It is well-known that an element $\mathbf{w} \in R_f$ can be represented as a vector $(w_0, ..., w_{d-1}) \in \mathbb{F}_q^d$. We can therefore represent $\mathbf{w}$ by a tuple of d group elements $(h^{w_0}, ..., h^{w_{d-1}}) \in \mathbb{G}^d$. Addition in R_f now becomes multiplication in $\mathbb{G}^d$, and multiplication by a known constant $\mathbf{a} \in R_f$ can be done (as we shall see) by applying a linear function in the exponent. This is simply because in R_f multiplication by a constant $\mathbf{a}$ acts as a linear mapping on the vector $(w_0, ..., w_{d-1})$. More details will be given below, but the essence is that if we set $\mathbf{g} = (h^{w_0}, ..., h^{w_{d-1}}) \in \mathbb{G}^d$ and take any $\mathbf{a} \in R_f$, we can define $\mathbf{g^a}$ in a completely natural way, namely as the d-tuple of elements in $\mathbb{G}$ that represent $\mathbf{wa}$. This leads to defining the f-DDH problem as follows: given $(\mathbf{g}, \mathbf{g^a}, \mathbf{g^b}, \mathbf{g^c})$, where $\mathbf{a,b,c} \in \mathbb{F}_{q^d}, \mathbf{g} \in \mathbb{G}^d$, decide if $\mathbf{c}$ is random or $\mathbf{c} = \mathbf{ab}$.

It is not hard to see that f_1-DDH and f_2-DDH are equivalent whenever R_{f_1} is isomorphic to R_{f_2}, and also that f_2-DDH is no harder than f_1-DDH where f_1 is an irreducible factor in f_2. So it is natural to consider only the case where f is irreducible of degree d, in which case $R_f = \mathbb{F}_{q^d}$. This variant is called d-DDH[1]. We show that if d_1 divides d_2, so that $\mathbb{F}_{q^{d_1}}$ is a subfield of $\mathbb{F}_{q^{d_2}}$, then d_2-DDH is

[1] The d-DDH assumption should not be confused with the previously known k-DDH assumption which is completely different and is *stronger* than DDH (see, e.g., [BB04, DY05, BMR10] for details on and applications of this assumption).

at least as hard as d_1-DDH. Conversely, we show in the generic group model that d-DDH for $d > 1$ is *harder* than DDH, and that d_2-DDH is harder than d_1-DDH if $d_1|d_2$ and $d_2 > 4(3d_1 - 2)$. Thus we get an infinite hierarchy of progressively weaker assumptions whose complexities lie "between" DDH and CDH.

From a basic research point of view, we believe this result is interesting because it contributes to understanding a very natural class of assumptions. Moreover, the proof is interesting from a technical point of view: proofs in the generic group model usually work by arguing that the adversary fails because he cannot compute expressions "in the exponent" of sufficiently high degree. This approach completely fails in our case, instead we have to solve a much harder task, namely we show that the ability to verify whether certain degree-2 equations are satisfied, does not allow verification of a different class of degree-2 equations.

From a more practical point of view, d-DDH gives us a large number of new schemes "for free" because virtually all known cryptographic constructions based on DDH can very easily be upgraded to be based on d-DDH: exactly the same construction and security proof applies but we get better security. Moreover, the amortized complexity of resulting schemes (e.g., computation per encrypted bit) is *the same* as when using DDH. We explain this in more detail in Section 5. In contrast, using the family of k-LIN assumptions is less attractive: The known DDH-based primitives have to be generalized to k-LIN and reproved from scratch, and one suffers a loss of efficiency that increases with k (also in the amortized sense).

How significant is the security advantage of using d-DDH? Given that in appropriately chosen groups, we do not know how to attack even the weakest variant, this can only be a matter of opinion. One may of course take the position that extending DDH is not useful: one can choose to believe that if DDH turns out to be easy, the algorithm will "probably" be so general that it can solve d-DDH for any d. This, on the other hand, is an argument that can made in exactly the same way against any known class of assumptions that generalize DDH, such as the k-LIN assumptions. With current state of the art, there is no way to settle this question. What our result does guarantee, however, is that if someone finds an efficient algorithm for DDH, even a non-generic one, there is no generic black-box reduction that turns it into an algorithm for 2-DDH, for instance. To render the d-DDH assumptions useless, one needs to solve the entire hierarchy using a non-generic reduction or a completely general algorithm.

We believe that in applications of cryptography, one should always minimize the risk of ones assumption being broken. And if the risk can potentially be made smaller at very little extra cost by modifying the application, there is good reason to do this. We therefore believe that using, e.g, 2-DDH instead of DDH is a 'good deal" in practice.

Everything we said so far applies to groups where no bilinear map is available, such as prime order subgroups of $\mathbb{Z}_p^*$ or compact elliptic curve based groups. In bilinear groups, however, it turns out that d-DDH, just like DDH, is easy. This fact motivates our suggestion of an alternative family of problems: we observe that by omitting some group elements from an instance of f-DDH, one can

obtain a problem that is hard, even if f is reducible. Based on this, we propose the d-vector DDH (d-VDDH) assumptions, based on $f(X) = X^d$. We show in the generic group model that the d-VDDH assumption holds even in bilinear groups. In fact, it holds even given a d-linear map, which can be thought of as an oracle allowing the adversary to compute expressions of degree d in the exponent. This means that the d-VDDH assumptions become progressively weaker for increasing d. We show that the d-VDDH assumption implies CCA-secure encryption and efficient Naor-Reingold style pseudorandom functions. We also construct another cryptosystem based on the d-VDDH assumption, very similar to the BHHO scheme [BHHO08]. We show that this scheme is auxiliary input secure, a strong form of leakage resilience where full information on the secret key can be leaked, as long as the key remains hard to compute.

In bilinear groups, the family of d-VDDH assumptions can therefore be seen as an alternative to the (incomparable) family of k-linear assumptions.

A final related work that should be mentioned is [HYZX08] in which an assumption called EDDH is proposed, which is our 2-DDH assumption. This is the only prior work we know of that mentions a DDH variant based on ring extensions. It is claimed in [HYZX08] that DDH reduces to EDDH and that in the generic group model EDDH is hard, even in bilinear groups. The first result is correct, but we could not verify the proof. In this paper, we give a different proof of a more general statement. The second claim is false, and is refuted by our result that d-DDH for any d is easy in bilinear groups.

2 Preliminaries

2.1 Notation

If S is a set, we write $x \leftarrow S$ meaning that x is sampled uniformly from S. If $\mathbf{x} \in \mathbb{F}_q^m$ is a vector, we write $x[i]$ for the ith entry of $\mathbf{x}$. We say that a function $f : \mathbb{N} \to \mathbb{R}$ is negligible if, for every polynomial p, there exists an integer $n_p \in \mathbb{N}$ such that $f(n) < 1/p(n)$ for every $n > n_p$. If X and Y are two random variables, we say that X and Y are computationally indistinguishable ($X \overset{c}{\approx} Y$) if their computational distance is negligible. Furthermore, throughout the paper, vectors are denoted by bold lowercase letters.

A d-linear map $e : \mathbb{G}^d \to \mathbb{G}_T$ is an efficiently computable map such that $e(g, \ldots, g) \neq 1$ and $e(g_1^{a_1}, \ldots, g_d^{a_d}) = e(g_1, \ldots, g_n)^{\prod a_i}$, for all g_i in $\mathbb{G}$ and for all a_i in $\mathbb{F}_q$. A d-linear group $\mathbb{G}$ is a group $\mathbb{G}$ together with a d-linear map.

3 Extension Rings and DDH

We consider here a finite field $\mathbb{F}_q$ of prime order q and its extension with a polynomial f of degree d. By this we obtain the ring $R_f = \mathbb{F}_q[X]/(f)$, where an element $\mathbf{v}$ can be written as $v_0 + \cdots + v_{d-1}X^{d-1} + (f)$. However, we can also represent $\mathbf{v}$ by the matrix $V = v_0\mathbf{I}_d + v_1A_f + \cdots + v_{d-1}A_f^{d-1}$, where $\mathbf{I}_d$ is the d-dimensional identity matrix and A_f is the so-called companion matrix of f. The

companion matrix of a monic polynomial $f = X^d + \alpha_{d-1}X^{d-1} + \cdots + \alpha_1 X^1 + \alpha_0$ is given by the $d \times d$ matrix

$$A_f = \begin{pmatrix} 0 & 0 & \cdots & 0 & -\alpha_0 \\ 1 & 0 & \cdots & 0 & -\alpha_1 \\ 0 & 1 & \cdots & 0 & -\alpha_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -\alpha_{d-1} \end{pmatrix}.$$

Action of Matrices on $\mathbb{G}^d$. Given a group $\mathbb{G}$ of order q and a tuple of elements $\mathbf{g} = (g_0, \ldots, g_{d-1}) \in \mathbb{G}^d$, any matrix $M = (m_{ij})$ of dimension $n \times d$ defines a mapping $\mathbb{G}^d \to \mathbb{G}^n$ as follows:

$$\mathbf{g}^M := \left(\prod_j^d g_j^{m_{1j}}, \ldots, \prod_j^d g_j^{m_{nj}} \right). \tag{1}$$

In particular this means that R_f can act on $\mathbb{G}^d$: we write the element $\mathbf{v} \in R_f$ in its matrix representation V and compute $\mathbf{g}^{\mathbf{v}} := \mathbf{g}^V$ as above. It is straightforward to verify that this map behaves according to the standard rules for exponentiation:

$$(\mathbf{g}^{\mathbf{a}})^{\mathbf{b}} = \mathbf{g}^{\mathbf{ab}}, \quad \mathbf{g}^{\mathbf{a}}\mathbf{g}^{\mathbf{b}} = \mathbf{g}^{\mathbf{a+b}}.$$

Note that this action can also be understood as implementing a product in R_f in a slightly different way: if we choose a generator h of $\mathbb{G}$, then we can write any $\mathbf{g}$ as $(g_0, \ldots, g_{d-1}) = (h^{w_0}, \ldots, h^{w_{d-1}})$. Once we fix h, we can therefore think of $\mathbf{g}$ as representing an element $\mathbf{w}$ in R_f, namely $\mathbf{w} = w_{d-1}X^{d-1} + \ldots + w_0 + (f)$. We will write this as $\mathbf{g} = h(\mathbf{w})$. It now turns out that we have

$$\mathbf{g}^{\mathbf{v}} = h(\mathbf{w})^{\mathbf{v}} = h(\mathbf{wv}).$$

This follows because we can think of R_f as a d-dimensional vector space over $\mathbb{F}_q$. In that interpretation, multiplication by $\mathbf{v}$ is a linear mapping which has a matrix, namely V. Since the action $\mathbf{g}^{\mathbf{v}}$ is defined to be multiplication by V "in the exponent", it follows that by computing $\mathbf{g}^{\mathbf{v}} = (h^{w_0}, \ldots, h^{w_{d-1}})^{\mathbf{v}}$, we are in fact multiplying $\mathbf{w}$ by $\mathbf{v}$.

3.1 The f-DDH Problem

Given the above, we can now define an new variant of the DDH problem:

Definition 1 (The f-DDH Problem). *Let f be a d-degree polynomial. Let $\mathcal{G}$ be a PPT algorithm, which given the security parameter λ, outputs the description of a group $\mathbb{G}$ of order $q = q(1^\lambda)$. Let $\mathcal{A}$ be a probabilistic algorithm that takes as input (a description of) $\mathbb{G}$ and a 4-tuple of elements in $\mathbb{G}^d$, and outputs 0 or 1. We say that $\mathcal{A}$ solves the f-DDH problem with advantage $\varepsilon_{\mathcal{A}}(\lambda)$, where*

$$\varepsilon_{\mathcal{A}}(\lambda) = |Pr[\mathcal{A}(\mathbb{G}, (\boldsymbol{g}, \boldsymbol{g^a}, \boldsymbol{g^b}, \boldsymbol{g^c})) = 1] - Pr[\mathcal{A}(\mathbb{G}, (\boldsymbol{g}, \boldsymbol{g^a}, \boldsymbol{g^b}, \boldsymbol{g^{ab}})) = 1]|$$

where $\boldsymbol{g} \leftarrow \mathbb{G}^d$ *and* $\boldsymbol{a} \leftarrow R_f, \boldsymbol{b} \leftarrow R_f, \boldsymbol{c} \leftarrow R_f$. *In other words, given* $(\boldsymbol{g}, \boldsymbol{g^a}, \boldsymbol{g^b}, \boldsymbol{g^c})$, *the problem is to decide whether* $\boldsymbol{c} = \boldsymbol{ab}$ *or* $\boldsymbol{c}$ *is a random element in* R_f.

Equivalently, we can think of the problem instance as being given in the alternative representation $(h(\mathbf{w}), h(\mathbf{wa}), h(\mathbf{wb}), h(\mathbf{wc}))$. This makes no difference to the adversary, as he would not be given $\mathbf{w}$ – but he knows that such a $\mathbf{w}$ exists. From the above we construct the following assumption.

Definition 2 (The f-DDH Assumption). *For any probabilistic polynomial time algorithm* $\mathcal{A}$ *as in Definition 1, it holds that* $\varepsilon_{\mathcal{A}}(\lambda)$ *is negligible as a function of* λ.

Note that this is a generalization of the DDH problem: for a polynomial f of degree 1, $R_f = \mathbb{F}_q$ and f-DDH is just the standard DDH problem in $\mathbb{G}$.

Now we look a bit closer at the polynomial f. We can distinguish between two different cases: one where f is reducible and one where f is irreducible. For the first case we have the following theorem:

Theorem 1 (f-DDH for Reducible f). *Let* f *be a* d*-degree reducible polynomial and suppose* f_0 *divides* f*, then solving* f*-DDH is polynomial time reducible to solving* f_0*-DDH.*

Proof. Let d_0 and d be the degrees of f_0 and f respectively. Let us consider an element $\mathbf{w}$ in R_f. We know that $\mathbf{w}$ can be written as $w_{d-1}x^{d-1} + \cdots + w_0 + (f)$. If we map $\mathbf{w}$ to R_{f_0} by reducing modulo f_0 we get an element $\mathbf{v} = v_{d_0-1}x^{d_0-1} + \cdots + v_0 + (f_0)$. In fact, reduction modulo f_0 is a ring homomorphism $\phi : R_f \to R_{f_0}$. It particular, it is linear and therefore has a matrix M. By (1) we can let M act on $\mathbf{w}$, so we get $h(\mathbf{w})^M = h(\phi(\mathbf{w})) = h(\mathbf{v})$. Hence, M can be used to efficiently map an f-DDH instance $(h(\mathbf{w}), h(\mathbf{wa}), h(\mathbf{wb}), h(\mathbf{wc}))$ to an f_0-DDH instance $(h(\phi(\mathbf{w})), h(\phi(\mathbf{wa})), h(\phi(\mathbf{wb})), h(\phi(\mathbf{wc}))) = (h(\mathbf{v}), h(\mathbf{v}\phi(\mathbf{a})), h(\mathbf{v}\phi(\mathbf{b})), h(\mathbf{v}\phi(\mathbf{c})))$. If $\mathbf{c} = \mathbf{ab}$, then $\phi(\mathbf{c}) = \phi(\mathbf{a})\phi(\mathbf{b})$, while if $\mathbf{c}$ is uniform in R_f, then $\phi(\mathbf{c})$ is uniformly chosen in R_{f_0}. Thus, if we can solve f_0-DDH, we can solve f-DDH with the same advantage.

4 The d-DDH Problem

Theorem 1 implies that f-DDH is no harder than f_0-DDH, where f_0 is the smallest irreducible factor in f. The natural conclusion is therefore that we should only look at the irreducible polynomials. In this case we know that our ring R_f is a field, namely the extension field $\mathbb{F}_{q^d}$ where d is the degree of f. In fact, since all fields with q^d elements are isomorphic, f-DDH is equivalent f'-DDH for any f' which is also irreducible and of the same degree as f. This is because the isomorphism can be implemented as a linear mapping in the same fashion as in the proof of Theorem 1. We can thus efficiently map an f-DDH instance to an f'-DDH instance and hence the only thing that may matter to the hardness of the problem is the degree of the extension. In the following, we will talk about d-DDH. In this definition we do not fix f; we can use any d-degree irreducible polynomial and otherwise the game is the same as in Definition 1.

Theorem 2. *Let d_1 divide d_2, so $\mathbb{F}_{q^{d_1}}$ is a subfield of $\mathbb{F}_{q^{d_2}}$, then d_1-DDH is no harder than d_2-DDH.*

The proof can be found in the full version [CDK+11]. We now show that d-DDH for $d > 1$ is, in fact, harder than DDH in the generic group model. Moreover, we show that if d_1 divides d_2 and $d_2 > 4(3d_1 - 2)$, then d_2-DDH is generically harder than d_1-DDH, giving in this way a hierarchy of progressively strictly weaker assumptions. For this, we need two auxiliary results. The first is a standard result, known as the Schwartz-Zippel lemma [Sch80, Zip79]:

Theorem 3. *For a non-zero multivariate polynomial over a finite field K of degree at most t, if uniformly random and independent values are assigned to the variables, the probability that this produces a root is at most $t/|K|$.*

The second is our main technical result supporting the hardness of d-DDH. In the following, for $\mathbf{a}, \mathbf{b} \in \mathbb{F}_{q^{d_1}}$ we will use $C_k(\mathbf{a}, \mathbf{b}) \in \mathbb{F}$ to denote the k-th component of the product $\mathbf{ab} \in \mathbb{F}_{q^{d_1}}$. Moreover, for ease of notation, whenever we have $P_1, \ldots, P_d$ affine functions from $(\mathbb{F}_{q^{d_2}})^3$ to $\mathbb{F}_q$, we will denote by P the vector consisting of all the P_i's. Namely $P(X, Y, Z) = (P_1(X, Y, Z), \ldots, P_d(X, Y, Z))$. Note that here we think of $\mathbb{F}_{q^{d_2}}$ as a d_2-dimensional vector space over $\mathbb{F}$. With this notation, an expression like $C_k(P, T)(X, Y, Z)$ can be understood in a natural way as a degree-2 polynomial in the $3d_2$ coordinates of X, Y and Z.

Theorem 4. *For $i = 1, \ldots, d_1$, let $P_i, R_i, S_i, T_i : (\mathbb{F}_{q^{d_2}})^3 \to \mathbb{F}_q$ be affine functions, with $d_2 > 4(3d_1 - 2)$. Assume $F_k(P, R, S, T)(X, Y, XY) := (C_k(P, T) - C_k(R, S))(X, Y, XY)$ is the zero polynomial. Then also $F_k(P, R, S, T)(X, Y, Z)$ is the zero polynomial. In particular, if $d_1 = 1$, the above is true for any $d_2 > 1$.*

The point of this theorem is that $X, Y, Z \in \mathbb{F}_{q^{d_2}}$ represent the input that the adversary gets in the generic group model game. Once he receives these inputs, the P, R, S, T represent new group elements he can compute. They are affine functions since the adversary can only compute sums and scalar multiplications "in the exponent". The adversary is trying to decide whether $Z = XY$ or if Z is random. He can try to do this by submitting a tuple of group elements (represented by P, R, S, T) to the oracle which answers back whether this tuple is an $\mathbb{F}_{q^{d_1}}$-DDH tuple or not. In the theorem, the functions F_k represent the oracle's answer, as for each component $k = 1, \ldots, d_1$, F_k tests if the tuple is "good" or not. What the theorem says is that, no matter how the adversary computes his oracle queries, if the tuple he is submitting is "good", this was already obvious without asking the oracle because the corresponding polynomials F_k are identically zero.

The idea behind the proof is writing the functions P_i, R_i, S_i, T_i from $(\mathbb{F}_{q^{d_2}})^3$ to $\mathbb{F}_q$ as sums of affine functions mapping $\mathbb{F}_{q^{d_2}}$ to $\mathbb{F}_q$. Such affine functions can be expressed via the trace function $\mathrm{Tr} : \mathbb{F}_{q^{d_2}} \to \mathbb{F}_q$, leading to an expression which is much easier to handle. We can then start looking at the implications of $F_k(P, R, S, T)(X, Y, XY)$ being zero. We show that $F_k(P, R, S, T)(X, Y, XY)$ vanishing in $(\mathbb{F}_{d_2})^2$ implies several terms of $F_k(P, R, S, T)(X, Y, Z)$ vanish as

well. Proceeding in this way, we simplify our expression further and obtain a polynomial which is a sum of products of trace functions. We show that the intersection of the kernels of these trace functions is not empty, and thus we prove that the last term surviving in $F_k(P,R,S,T)(X,Y,Z)$ actually does not depend on Z and so must be zero as well.

The complete proof of the theorem can be found in the full version [CDK+11].

Theorem 5. *In the generic group model, the d_2-DDH assumption holds even when the adversary is given an oracle allowing him to solve the d_1-DDH problem, for $d_2 > 4(3d_1 - 2)$. In particular, if $d_1 = 1$, we have that d_2-DDH holds even when an adversary has access to a DDH oracle, for any $d_2 > 1$.*

Proof. Recall that an instance to the d_2-DDH problem can be written as $(h(\mathbf{w}), h(\mathbf{wa}), h(\mathbf{wb}), h(\mathbf{wc}))$ for a fixed generator h of $\mathbb{G}$ and random $\mathbf{w}, \mathbf{a}, \mathbf{b}, \mathbf{c}$ in $\mathbb{F}_{q^{d_2}}$. We will show, in the generic group model, that the problem remains hard even if the adversary is given $\mathbf{w}$. From $\mathbf{w}$, it is easy to compute $\mathbf{w}^{-1}$. So we can equivalently think of the problem as being given instead as $(h(\mathbf{x}), h(\mathbf{y}), h(\mathbf{z}))$, where the adversary now has to decide whether $\mathbf{z} = \mathbf{xy}$.

We will assume that a random bit b is chosen by the simulator, and when $b = 0$ the adversary sees $\mathbf{z} = \mathbf{xy}$, while if $b = 1$, the adversary will see a uniform $\mathbf{z}$. The theorem is proved if we can show that a polynomial-time adversary cannot guess b with non-negligible advantage over $1/2$.

Let $\mathcal{A}$ be a polynomial-time generic group adversary. As usual, $\mathcal{A}$ has access to an oracle computing the group operation and inversion. In our case, we also give $\mathcal{A}$ access to an oracle solving d_1-DDH problem. More formally, on input $g^{w_0}, \ldots, g^{w_{d_1-1}}, g^{a_0}, \ldots, g^{a_{d_1-1}}, g^{b_0}, \ldots, g^{b_{d_1-1}}, g^{c_0}, \ldots, g^{c_{d_1-1}}$, the oracle outputs 1 if $\mathbf{w}^2\mathbf{c} = \mathbf{wawb}$ in $\mathbb{F}_{q^{d_1}}$.

We consider an algorithm $\mathcal{B}$ playing the following game with $\mathcal{A}$. Algorithm $\mathcal{B}$ chooses $3d_2 + 2$ bit strings $\sigma_0, \ldots, \sigma_{3d_2+1}$ uniformly in $\{0,1\}^m$, for a sufficiently large m. These strings represent the encoded elements which algorithm $\mathcal{A}$ will work with. Internally, $\mathcal{B}$ keeps track of the encoded elements using polynomials in the ring $\mathbb{F}_q[X_1, \ldots, X_{d_2-1}, Y_0, \ldots, Y_{d_2-1}, Z_0, \ldots, Z_{d_2-1}, T_0]$. Externally, the elements that $\mathcal{B}$ gives to $\mathcal{A}$ are just bit strings in $\{0,1\}^m$. To maintain consistency, $\mathcal{B}$ creates a list L consisting of pairs (F, σ) where F is a polynomial in the ring specified above and σ is a bit string. Initially, L is set to $\{(1, \sigma_0), (X_1, \sigma_1), \ldots, (X_{d_2-1}, \sigma_{d_2-1}), (Y_0, \sigma_{d_2}), \ldots, (Y_{d_2-1}, \sigma_{2d_2-1}), (Z_0, \sigma_{2d_2}), \ldots, (Z_{d_2-1}, \sigma_{3d_2-1})\}$.
Algorithm $\mathcal{B}$ starts the game providing $\mathcal{A}$ with $\sigma_0, \ldots, \sigma_{3d_2-1}$. The simulation of the oracles goes as follows:

Group Action: Given two strings σ_i, σ_j, $\mathcal{B}$ recovers the corresponding polynomials F_i and F_j and computes $F_i + F_j$. If $F_i + F_j$ is already in L, $\mathcal{B}$ returns to $\mathcal{A}$ the corresponding bit string; otherwise it returns a uniform element σ in $\{0,1\}^m$ and stores $(F_i + F_j, \sigma)$ in L.

Inversion: Given an element σ in $\mathbb{G}$, $\mathcal{B}$ recovers its internal representation F and computes $-F$. If the polynomial $-F$ is already in L, $\mathcal{B}$ returns the corresponding bit string; otherwise it returns a uniform string σ and stores $(-F, \sigma)$ in L.

d_1-DDH: Given $4d_1$ strings $\pi_1, \ldots, \pi_{d_1}, \rho_1, \ldots, \rho_{d_1}, \sigma_1, \ldots, \sigma_{d_1}, \tau_1, \ldots, \tau_{d_1}$ in $\mathbb{G}$, adversary $\mathcal{B}$ recovers the polynomials $P_1, \ldots, P_{d_1}, R_1, \ldots, R_{d_1}, S_1, \ldots, S_{d_1}$, $T_1, \ldots, T_{d_1}$ and returns 1 iff $C_i(P_1, \ldots, P_{d_1}, T_1, \ldots, T_{d_1}) = C_i(R_1, \ldots, R_{d_1}, S_1, \ldots, S_{d_1})$ for every $i = 1, \ldots, d_1$, where C_i represents the i-th component of the product in $\mathbb{F}_{q^{d_1}}$.

After $\mathcal{A}$ queried the oracles, it outputs a bit b'. At this point, $\mathcal{B}$ chooses uniform values $\mathbf{x} = (x_1, \ldots, x_{d_2-1})$, $\mathbf{y} = (y_0, \ldots, y_{d_2-1})$, $\mathbf{z} = (z_0, \ldots, z_{d_2-1})$ in $\mathbb{F}_{q_d^2}$ and sets $X_1 = x_1, \ldots, X_{d_2-1} = x_{d_2-1}$, $Y_0 = y_0, \ldots, Y_{d_2-1} = y_{d_2-1}$. Finally $\mathcal{B}$ chooses a bit b and, if $b = 1$ it sets $Z_0 = z_0, \ldots, Z_{d_2-1} = z_{d_2-1}$, otherwise it sets $Z_0 = C_0(\mathbf{x},\mathbf{y}), ..., Z_{d_2-1} = C_{d_2}(\mathbf{x},\mathbf{y})$.

If the simulation provided by $\mathcal{B}$ is consistent, it reveals nothing about b. This means that the probability of $\mathcal{A}$ guessing the correct value for b is $1/2$. The only way in which the simulation could be inconsistent is if, after we choose value for $\mathbf{x}, \mathbf{y}, \mathbf{z}$, either two different polynomials in L happen to produce the same value or some query to the d_1-DDH oracle is such that $C_i(P_1, \ldots, P_{d_1}, T_1, \ldots, T_{d_1}) - C_i(R_1, \ldots, R_{d_1}, S_1, \ldots, S_{d_1})$ is not the 0 polynomial, but produces 0 after assigning values.

If $b = 1$, all values for $\mathbf{x}, \mathbf{y}, \mathbf{z}$ are chosen independently, so Theorem 3 applies to show that for a single oracle query $C_i(P_1, \ldots, P_{d_1}, T_1, \ldots, T_{d_1}) - C_i(R_1, \ldots, R_{d_1}, S_1, \ldots, S_{d_1})$ or a single difference $F_i - F_j$, the probability of having 0 after assigning values is negligible because q is exponentially large and all polynomials involved have degree at most 2. Further, by the union bound, since we only have a polynomial number of polynomials to consider, the overall probability of having 0 after assigning values is also negligible.

If $b = 0$, there are two extra possibilities for inconsistency between simulation and real attack. The first is if some query to the d_1-DDH oracle satisfies that

$$C_i(P_1, \ldots, P_{d_1}, T_1, \ldots, T_{d_1}) - C_i(R_1, \ldots, R_{d_1}, S_1, \ldots, S_{d_1})(X, Y, Z) \neq 0,$$

but

$$C_i(P_1, \ldots, P_{d_1}, T_1, \ldots, T_{d_1}) - C_i(R_1, \ldots, R_{d_1}, S_1, \ldots, S_{d_1})(X, Y, XY)$$

is the 0-polynomial. This is ruled out by Theorem 4, since all the polynomials involved have degree at most 1 and can therefore be thought of as affine functions. The second potential inconsistency is if two distinct polynomials F_i, F_j in L satisfy that $(F_i - F_j)(X, Y, XY)$ is the 0 polynomial. To see that this cannot happen, note that since each F_i has degree at most 1, it can be decomposed uniquely as as $F_i(X, Y, Z) = F_i^x(X) + F_i^y(Y) + F_i^z(Z) + c_i$ for a constant c_i and polynomials $F_i^x(X), F_i^y(Y), F_i^z(Z)$ of degree at most 1 and constant term 0. A collision as described here can only happen if $(F_i^z - F_j^z)(Z) \neq 0$, but $(F_i^z - F_j^z)(XY) = 0$. This leads to a contradiction: we can assign values $Y_0 = 1$,

$Y_1 = 0, ..., Y_{d-1} = 0$, corresponding to the 1-element in $\mathbb{F}_{q^d}$. With this assignment, we get that $(F_i^z - F_j^z)(X) = 0$, contradicting that $(F_i^z - F_j^z)(Z) \neq 0$. Having ruled out these two possibilities for inconsistency, the only remaining possibility is that an unfortunate choice of values for the variables lead to collisions, as in the $b = 1$ case. Again by Theorem 3, this happens with negligible probability since the involved polynomials have degree at most 4.

We now look at what happens to d-DDH in a bilinear group. In such a group it is well-known that DDH is easy, and we show that this is also the case for d-DDH. The EDDH assumption presented in [HYZX08] is equivalent to d-DDH for $d = 2$. It was claimed that EDDH is hard also in generic bilinear groups, which is however refuted by the following result:

Theorem 6. *d-DDH over any bilinear group can be solved in polynomial time.*

Proof. We assume that the extension field $\mathbb{F}_{q^d}$ has been constructed using some fixed irreducible polynomial f. Consider any two elements $\mathbf{x},\mathbf{y} \in \mathbb{F}_{q^d}$ as vectors $\mathbf{x} = (x_0, ..., x_{d-1}), \mathbf{y} = (y_0, ..., y_{d-1})$ and write the product as $\mathbf{xy} = (z_0, ..., z_{d-1})$. Now, multiplication of $\mathbf{x}$ and $\mathbf{y}$ takes place by multiplying the polynomials $x_0 + ... + x_{d-1}X^{d-1}$ and $y_0 + ... + y_{d-1}X^{d-1}$ and reducing modulo $f(X)$. From this it follows that we can write

$$z_k = \sum \alpha_{ij}^k x_i y_j$$

for coefficients $\alpha_{ij}^k \in \mathbb{F}_q$ that depend only on $f(x)$. Now, if we are given d-tuples $h(\mathbf{x}), h(\mathbf{y})$, it follows from the above that we can efficiently compute a representation in the target group G_T of $\mathbf{xy}$. Namely, for every k, we have

$$e(h,h)^{z_k} = \prod_{ij} (e(h,h)^{x_i y_j})^{\alpha_{ij}^k} = \prod_{ij} e(h^{x_i}, h^{y_j})^{\alpha_{ij}^k}$$

and h^{x_i}, h^{y_j} can be taken directly from $h(\mathbf{x}), h(\mathbf{y})$. So if we define

$$e(h,h)(\mathbf{xy}) = (e(h,h)^{z_0}, ..., e(h,h)^{z_{d-1}})$$

what we have shown is that we can compute $e(h,h)(\mathbf{xy})$ efficiently from $h(\mathbf{x}), h(\mathbf{y})$.

Now, consider an input instance of d-DDH, in the form $h(\mathbf{w}), h(\mathbf{wa}), h(\mathbf{wb}), h(\mathbf{wc})$. Observe that we have $\mathbf{c} = \mathbf{ab}$ if and only if $\mathbf{wa\ wb} = \mathbf{w\ wc} = \mathbf{w}^2\mathbf{ab}$. It now follows immediately from the above that we can decide if $\mathbf{ab} = \mathbf{c}$ by computing $e(h,h)(\mathbf{wa\ wb})$ and $e(h,h)(\mathbf{w\ wc})$ and comparing the two.

Although of course not all groups are bilinear, this result nevertheless motivates looking for alternative assumptions with similar properties that can be assumed to be hard in bilinear groups. We do this in Section 6.

5 Applications of d-DDH

In this section we present a number of applications for the d-DDH assumption.

5.1 Pseudorandom Functions

We construct pseudorandom functions (PRF) from d-DDH by taking the construction from [NR97] and showing that the natural modification where we work in the extension field also gives a PRF.

Definition 3. *Let $F = \{F_k\}$ be a family of keyed functions where $F_k : A_k \to B_k$, for every k in the key space $\mathcal{K}$. We say that F is a family of pseudorandom functions if for all PPT algorithms $\mathcal{D}$, any polynomial p and large enough λ,*

$$|\Pr[\mathcal{D}^{F_k(\cdot)}(1^\lambda) = 1] - \Pr[\mathcal{D}^{f(\cdot)}(1^\lambda) = 1]| < 1/p(\lambda),$$

where k is chosen uniformly in $\mathcal{K}$ and f is chosen uniformly from the set of functions mapping A_k to B_k.

PRF Construction. We construct a function family $F = \{f_k\}$ as follows. The index k specifies a tuple $(q, \mathbb{G}^d, \mathbf{g}, \mathbf{a}_0, \ldots, \mathbf{a}_n)$ where q is a prime number, $\mathbb{G}$ is a group of order q, $\mathbf{g}$ is an element of $\mathbb{G}^d$ and $\mathbf{a}_0, \ldots, \mathbf{a}_n$ are random in $\mathbb{F}_{q^d}$. Finally, we define $f_k : \{0,1\}^n \to \mathbb{G}^d$, $f_k(x_1, \ldots, x_n) = \mathbf{g}^{\mathbf{a}_0 \prod_{x_i=1} \mathbf{a}_i}$.

Theorem 7. *Under the d-DDH assumption, the family $F = \{f_k\}$ defined above is a family of pseudorandom functions.*

The proof of the theorem follows the exact same line as in [NR97]. Essentially the proof is done by a hybrid argument in which we define a sequence of functions $\{h_i\}$ where h_0 is f_k and h_n is a uniformly random function. An adversary that distinguishes between h_0 and h_n will also distinguish between h_i and h_{i+1}, for some i, which reduces to the d-DDH problem.

5.2 Public Key Encryption

We now apply d-DDH to public key encryption. If we modify in the natural way the Elgamal [Gam84] scheme, we obtain CPA secure encryption based on d-DDH.

- $\mathsf{Gen}(1^\lambda)$: Let $\mathbb{G} \leftarrow \mathcal{G}(1^\lambda)$. Choose a random element $\mathbf{g} \leftarrow \mathbb{G}^d$ and random $\mathbf{x} \leftarrow \mathbb{F}_{q^d}$. Compute $\mathbf{h} = \mathbf{g}^{\mathbf{x}}$. The secret key is then $sk = \mathbf{x}$ and the public key is $pk = (\mathbf{h})$, where $\mathbb{G}$ can be considered a public parameter.
- $\mathsf{Enc}(pk, M)$: Let the message be $M \in \mathbb{G}^d$. Choose randomly $\mathbf{r} \leftarrow \mathbb{F}_{q^d}$. Output the ciphertext $CT = (\mathbf{g}^{\mathbf{r}}, \mathbf{h}^{\mathbf{r}} \cdot M)$.
- $\mathsf{Dec}(sk, CT)$: Write the ciphertext as $CT = (\mathbf{e}, \mathbf{c})$. Output $M' = c \cdot (\mathbf{e}^{\mathbf{x}})^{-1}$

The proof of correctness and security follows immediately as for standard Elgamal.

5.3 Applications in General

Having seen the two examples above, it should not be surprising that all DDH-based cryptographic schemes we are aware of can be based on d-DDH instead. This is basically because all involved algorithms (such as key generation, encryption, and security reduction) will work given only black-box access to a group $\mathcal{G}$ and a finite field K. We just need that for $g \in \mathcal{G}$ and $x \in K$, $g^x \in \mathcal{G}$ is well-defined and standard "axioms" such as $g^{x+y} = g^x g^y$ and $(g^x)^y = g^{xy}$ hold. The exact same scheme and security proof can be run, based on $(\mathcal{G}, K) = (\mathbb{G}, \mathbb{F}_q)$ or based on $(\mathcal{G}, K) = (\mathbb{G}^d, \mathbb{F}_{q^d})$. The only difference is that we need the d-DDH assumption in the latter case. Thus, for instance, CCA secure encryption [CS98] and circular secure or auxiliary input secure encryption [BHHO08] follow immediately from d-DDH.

5.4 Efficiency

For all constructions mentioned here, we can define a notion of amortized complexity. For a PRF, this is the computation time needed to produce a single pseudorandom group element; for an encryption scheme it is the computation time needed to encrypt a group element.

An important point is that in all applications we are aware of, the amortized complexity is essentially the same for constructions based on DDH and on d-DDH. This is because for $\mathbf{g} \in \mathbb{G}^d$ and $\mathbf{a} \in \mathbb{F}_{q^d}$, $\mathbf{g}^{\mathbf{a}}$ corresponds to a tuple of length d where each entry is an expression of the form $\prod g_i^{\alpha_i}$. By a well-known algorithm (see [Pip76]) such a value can be computed in time roughly what you need for a single exponentiation in $\mathbb{G}$.

As a concrete example, computing the PRF defined above requires essentially a single exponentiation: $\mathbf{g}^{(\mathbf{a}_0 \prod_{x_i=1} \mathbf{a}_i)}$. This produces d pseudorandom elements at amortized cost roughly 1 exponentiation in $\mathbb{G}$, which is the same cost as the DDH based version.

Various optimizations are known that save computation in the constructions we consider here. However, all the optimizations we are aware of can be applied to both variants based on DDH and d-DDH, and therefore do not affect our conclusion on the amortized complexities.

6 The Vector DDH Problem

The main observation in this section is that we can construct a problem that is generically harder than DDH by revealing only the last entry of the final vector in an f-DDH instance. In the following, we study in detail what happens if we choose f to be x^d. It turns out that there is a simple way of expressing products in $R_d = \mathbb{F}_q[X]/(x^d)$. If we take $\mathbf{x} = (x_0, \ldots, x_{d-1})$ and $\mathbf{y} = (y_0, \ldots, y_{d-1})$ in R_d, we have:

$$\mathbf{xy} = \left(x_0 y_0, \ldots, \sum_{k=0}^{i-1} x_k y_{i-1-k}, \ldots, \sum_{k=0}^{d-1} x_k y_{d-1-k} \right). \tag{2}$$

We define the d-VDDH problem just like d-DDH, except that the problem instance is now of the form $(h(\mathbf{w}), h(\mathbf{wa}), h(\mathbf{wb}), h(\mathbf{wc})[d])$, where we recall that $\mathbf{x}[d]$ is the dth entry of the vector $\mathbf{x}$, that is x_{d-1} if we start numbering from 0.

Definition 4 (The d-VDDH Problem). *Let d be an integer. Let $\mathcal{G}$ be a PPT algorithm, which given the security parameter λ, outputs the description of a group $\mathbb{G}$ of order $q = q(1^\lambda)$. Let $\mathcal{A}$ be a probabilistic algorithm that takes as input (a description of) $\mathbb{G}$ and a 3-tuple in $\mathbb{G}^d$ plus an element in $\mathbb{G}$, and outputs 0 or 1.*

We say that $\mathcal{A}$ solves the d-VDDH problem with advantage $\varepsilon_{\mathcal{A}}(\lambda)$, where

$$\varepsilon_{\mathcal{A}}(\lambda) = |Pr[\mathcal{A}(\mathbb{G}, (\boldsymbol{g}, \boldsymbol{g}^{\boldsymbol{a}}, \boldsymbol{g}^{\boldsymbol{b}}, \boldsymbol{g}^{\boldsymbol{c}}[d])) = 1] - Pr[\mathcal{A}(\mathbb{G}, (\boldsymbol{g}, \boldsymbol{g}^{\boldsymbol{a}}, \boldsymbol{g}^{\boldsymbol{b}}, \boldsymbol{g}^{\boldsymbol{ab}}[d])) = 1]|$$

where $\boldsymbol{g} \leftarrow \mathbb{G}^d$ and $\boldsymbol{a} \leftarrow R_d, \boldsymbol{b} \leftarrow R_d, \boldsymbol{c} \leftarrow R_d$.

Definition 5 (The d-VDDH Assumption). *For any probabilistic polynomial time algorithm $\mathcal{A}$ as in Definition 4, it holds that $\varepsilon_{\mathcal{A}}(\lambda)$ is negligible as a function of λ.*

Recall the notation from Section 3: $\mathbf{g} = (g_0 \ldots, g_{d-1}) = (h^{w_0}, \ldots, h^{w_{d-1}})$. Note that we WLOG can choose $w_0 = 1$, so $h(\mathbf{w}) = (g_0, g_0^{w_1} \ldots, g_0^{w_{d-1}})$. To prove that d-VDDH is generically hard, even in d-linear groups, it is useful to do the following parameter substitution: set $\mathbf{x} = \mathbf{wa}$, $\mathbf{y} = \mathbf{wb}$. The d-VDDH problem now becomes deciding whether the last element is the dth coordinate of $\mathbf{xyw}^{-1}$ or is random.

Now, set $\mathbf{w}^{-1} = (z_0, z_1, ..., z_{d-1})$ and consider the z_i as unknowns. Since $\mathbf{ww}^{-1} = \mathbf{1} = (1, 0, ..., 0)$ we get $d-1$ equations involving the z_i's, using the product introduced in (2):

$$z_0 = 1,\ z_1 = -w_1,\ \ldots,\ z_i = -w_i - \sum_{j+l=i} z_l w_j,\ \ldots,\ z_{d-1} = -w_{d-1} - \sum_{j+l=d-1} z_l w_j$$

In particular, $z_i = -w_1 z_{i-1} - \cdots - w_{i-1} z_1 - w_i$. Hence, it can be proved by simple induction that z_i has degree i as a function of the w_j's. Now, let $p_i(\mathbf{w},\mathbf{x},\mathbf{y})$ be the ith entry of $\mathbf{w}^{-1}\mathbf{xy}$. Then $p_d(\mathbf{w},\mathbf{x},\mathbf{y})$ has degree $d+1$ in $\mathbf{w}, \mathbf{x}$, and $\mathbf{y}$.
We are now ready to prove the generic hardness of d-VDDH.

Theorem 8. *Even given a d-linear mapping, the d-VDDH holds in the generic group model.*

The proof can be found in the full paper [CDK+11].

Later, we will need a lemma stating that a generalization of the d-VDDH which considers several generators is equivalent to the original assumption.

Lemma 1. *If d-VDDH is hard for $\mathcal{G}$, then for any positive integer m*

$$\left\{(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_m, \boldsymbol{g}_1^{\boldsymbol{r}}[d], \ldots, \boldsymbol{g}_m^{\boldsymbol{r}}[d]) \mid \boldsymbol{g}_i \leftarrow \mathbb{G}^d, \boldsymbol{r} \leftarrow R_d\right\} \overset{c}{\approx} \tag{3}$$

$$\left\{(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_m, \boldsymbol{g}_1^{\boldsymbol{r}_1}[d], \ldots, \boldsymbol{g}_m^{\boldsymbol{r}_m}[d]) \mid \boldsymbol{g}_i \leftarrow \mathbb{G}^d, \boldsymbol{r}_i \leftarrow R_d\right\}. \tag{4}$$

The proof can be found in the full paper [CDK+11].

7 Applications of d-VDDH

In this section we discuss a number of natural application of our d-VDDH assumption. Throughout this section we will use the ring $R_d = \mathbb{F}_q[X]/(f)$ for $f = X^d$.

7.1 Public Key Encryption

It is immediate how to construct a CPA-secure encryption schemes from the d-VDDH assumption family. We now show how to extend them to chosen-ciphertext (CCA) secure schemes. Let us first recall the definition of chosen-ciphertext security for encryption schemes.

Definition 6. *A scheme* PKE *is CCA secure if for any PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, any polynomial* p *and large enough* λ*,*

$$\mathsf{Adv}_{\mathcal{A},h} := \left|\Pr[\mathsf{CCA}_0(\mathsf{PKE}, \mathcal{A}, 1^\lambda)] - \Pr[\mathsf{CCA}_1(\mathsf{PKE}, \mathcal{A}, 1^\lambda)]\right| < 1/p(\lambda),$$

where $\mathsf{CCA}_b(\mathsf{PKE}, \mathcal{A}, 1^\lambda)$ *is output from the following experiment:*

$$\begin{aligned}(pk, sk) &\leftarrow \mathcal{G}(1^\lambda)\\ (m_0, m_1, \mathsf{state}) &\leftarrow \mathcal{A}_1^{\mathsf{Dec}(sk,\cdot)}(1^\lambda, pk) \text{ with } |m_0| = |m_1|\\ CT^* &\leftarrow \mathsf{Enc}_{pk}(m_b),\\ \text{Output } b' &\leftarrow \mathcal{A}_2^{\mathsf{Dec}(sk,\cdot)}(1^\lambda, \mathsf{state}, CT^*)\end{aligned}$$

In the second phase the decryption oracle $\mathsf{Dec}(sk, \cdot)$ returns $\perp$ when queried on the challenge ciphertext CT^*.

We now give the construction of our CCA secure encryption scheme. Let (E, D) be a symmetric encryption scheme with key-space $K \in \mathbb{G}$. Let $\mathsf{T} : \mathbb{G}^d \to \mathbb{F}_q$ be a target collision resistant hash function (see [HK07] for a definition) and define $\hat{\mathsf{T}}(\mathbf{x}) := (\mathsf{T}(x), 0, \ldots, 0) \in R_d$. (Note that for two elements $\mathbf{x} \neq \mathbf{y}$ we have that $\hat{\mathsf{T}}(\mathbf{x}) - \hat{\mathsf{T}}(\mathbf{y})$ is invertible in R_d unless $\mathsf{T}(\mathbf{x}) = \mathsf{T}(\mathbf{y})$).

- $\mathsf{Gen}(1^\lambda)$: Let $\mathbb{G} \leftarrow \mathcal{G}(1^\lambda)$. Choose a random generator $\mathbf{g}_1 \leftarrow \mathbb{G}^d$ and random indices $\mathbf{w}, \mathbf{x}, \mathbf{y} \leftarrow R_d$. Compute $\mathbf{g}_2 = \mathbf{g}^{\mathbf{w}}, \mathbf{u} = \mathbf{g}^{\mathbf{x}}, \mathbf{v} = \mathbf{g}^{\mathbf{y}}$. The secret key is then $sk = \mathbf{w}, \mathbf{x}, \mathbf{y}$ and the public key is $pk = (\mathbb{G}, \mathbf{g}_1, \mathbf{g}_2, \mathbf{u}, \mathbf{v})$.
- $\mathsf{Enc}(pk, M)$: Choose randomly $\mathbf{r} \leftarrow R_d$. Compute $\mathbf{c}_1 = \mathbf{g}^{\mathbf{r}}$ and $\mathbf{c}_2 = (\mathbf{u}^{\mathbf{t}}\mathbf{v})^{\mathbf{r}}$, where $\mathbf{t} = \hat{\mathsf{T}}(\mathbf{c}_1) \in R_d$. Compute the symmetric part as $C = \mathsf{E}_K(m)$, where $K = \mathbf{g}_2^{\mathbf{r}}[d]$. Output the ciphertext $CT = (\mathbf{c}_1, \mathbf{c}_2, C)$.
- $\mathsf{Dec}(sk, CT)$: Write the ciphertext as $CT = (\mathbf{c}_1, \mathbf{c}_2, C)$. If $\mathbf{c}_1^{\mathbf{x}\cdot\mathbf{t}+\mathbf{y}} \neq \mathbf{c}_2$ then return $\perp$. Otherwise return $\mathsf{D}_K(C)$, where $K = \mathbf{c}_1^{\mathbf{w}}[d]$.

It is easy to see that correctness follows by the definition of the public/secret key and by the correctness of the symmetric scheme. To prove the theorem we need that symmetric scheme is secure in the sense of authenticated encryption. That is, it acts as a one-time pad plus any decryption query (with respect to a uniform random key) is rejected. We refer again to [HK07] for a formal definition.

Theorem 9. *If* (E, D) *is a symmetric encryption scheme secure in the sense of authenticated encryption,* T *is a target collision resistant hash function and the d-VDDH holds in* $\mathbb{G}$*, then the encryption scheme is IND-CCA secure.*

The proof is exactly the same as Theorem 2 in [HK07] where an encryption scheme is proved CCA secure from the DDH assumption. We give some intuition about the proof.

The difficulty in the security reduction is that an adversary against the d-VDDH assumption has to answer the decryption queries and hence has to distinguish between consistent ciphertexts (i.e., ciphertexts for that $\mathbf{c}_1^{\mathbf{xt}+\mathbf{y}} = \mathbf{c}_2$ holds) and inconsistent ones, without knowing $\mathbf{w} = \log_{\mathbf{g}_1} \mathbf{g}_2$. The simulator inputs $(\mathbf{g}_1, \mathbf{g}_2, \mathbf{c}_1^* = \mathbf{g}_1^{\mathbf{r}}, K^*)$ and wants to distinguish $K^* = \mathbf{g}_2^{\mathbf{r}^*}[d]$ from a uniform element in $\mathbb{G}$. In the simulation the values $\mathbf{u}, \mathbf{v}$ from the public-key are set-up such that the tuple $\mathbf{c}_1^*, \mathbf{c}_2^*$ can be used as the challenge ciphertext for some efficiently computable $\mathbf{c}_2^*$ and the value K^* as the symmetric key. More precisely, we define $\mathbf{u} = \mathbf{g}_1^{\mathbf{x}_1}\mathbf{g}_2^{\mathbf{x}_2}, \mathbf{v} = \mathbf{g}_1^{\mathbf{y}_1}\mathbf{g}_2^{-\mathbf{t}^* \cdot \mathbf{x}_2}$ for uniform $\mathbf{x}_1, \mathbf{y}_1 \in R_d$, $\mathbf{x}_2 \in R_d^*$ and $\mathbf{t}^* = \hat{\mathsf{T}}(\mathbf{c}_1^*)$. By construction, the corresponding real session key is $\mathbf{g}_2^{\mathbf{r}^*}[d]$ so breaking the indistinguishability of the scheme is equivalent to solving the d-VDDH problem. It leaves to deal with the decryption queries for $CT = (\mathbf{c}_1, \mathbf{c}_2, C)$. The simulator is not able to distinguish consistent from inconsistent ciphertexts. However, for ciphertexts with $\mathbf{t} = \hat{\mathsf{T}}(\mathbf{c}_1) \neq \mathbf{t}^*$ (these are the interesting cases) the simulator implements an alternative decryption algorithm by computing the symmetric key as $K = (\mathbf{c}_1\mathbf{c}_2^{-\mathbf{x}_1\mathbf{t}+\mathbf{y}_1})^{(\mathbf{x}_2(\mathbf{t}-\mathbf{t}^*))^{-1}}[d]$. (Note that by the properties of $\hat{\mathsf{T}}$, $\mathbf{x}_2(\mathbf{t} - \mathbf{t}^*) \in R^*$ so its inverse is well-defined.) This has the following consequences.

It is easy to verify that if the queried ciphertext is consistent then the alternative decryption algorithm yields the correct symmetric key $K = \mathbf{c}_1^{\mathbf{w}}$. If the queried ciphertext is inconsistent then the alternative decapsulation algorithm yields one single symmetric key K that is uniformly distributed over $\mathbb{G}$. (The probability space is taken over all possible $\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1$ that yield $\mathbf{u}, \mathbf{v}$ from the public-key given to the adversary.) Returning this key K to the adversary would completely determine the simulator's secret key and hence also the virtual symmetric key K' for the next decapsulation query. However, this key K is used to decrypt the symmetric part C of the decryption query and by the authenticity property of the latter this will always lead to a rejection. Hence the decryption query is answered correctly and no information about the secret key is leaked which makes it possible to apply the same argument again.

7.2 Generalized BHHO Encryption

In this section we define a public-key encryption scheme which is heavily inspired by the scheme in [BHHO08]. Here, however, the cryptosystem is based on d-VDDH, instead of DDH.

Let λ be the security parameter and $m = m(\lambda)$ be a parameter of the scheme. The encryption scheme is $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

- $\mathsf{Gen}(1^\lambda)$: Let $\mathbb{G} \leftarrow \mathcal{G}(1^\lambda)$. Choose a vector of uniformly random generators $\mathbf{g} = (\mathbf{g}_1, \ldots, \mathbf{g}_m), \mathbf{g}_i \leftarrow \mathbb{G}^d$ and random bit string $\mathbf{s} = (s_1, \ldots, s_m) \leftarrow \{0,1\}^m$. Compute $\mathbf{y} = \prod_{i=1}^m \mathbf{g}_i^{(s_i,0,\ldots,0)}$, where $(s_1, 0, \ldots, 0)$ is viewed as an element in R_d. The secret key is then $sk = \mathbf{s}$ and the public key is $pk = (\mathbb{G}, \mathbf{g}, \mathbf{y})$, where $\mathbb{G}$ and $\mathbf{g}$ can be considered public parameters.
- $\mathsf{Enc}(pk, M)$: Let the message be $M \in \mathbb{G}$. Choose randomly $\mathbf{r} \leftarrow R_d$. Compute $f_i = \mathbf{g}_i^{\mathbf{r}}[d]$ and output the ciphertext $CT = (f_1, \ldots, f_m, \mathbf{y}^{\mathbf{r}}[d] \cdot M)$.
- $\mathsf{Dec}(sk, CT)$: Write the ciphertext as $CT = (f_1, \ldots, f_m, c)$. Output $M' = c \cdot \left(\prod_{i=1}^m f_i^{s_i}\right)^{-1}$

Correctness of decryption follows since

$$\begin{aligned}\prod_{i=1}^m f_i^{s_i} &= \prod_{i=1}^m \left(\mathbf{g}_i^{\mathbf{r}}[d]\right)^{s_i} = \prod_{i=1}^m \left(g_{i1}^{r_d} \cdot g_{i2}^{r_{d-1}} \cdots g_{id}^{r_1}\right)^{s_i} = \prod_{i=1}^m \left(g_{i1}^{r_d s_i} \cdot g_{i2}^{r_{d-1} s_i} \cdots g_{id}^{r_1 s_i}\right) \\ &= \prod_{i=1}^m \left(g_{i1}^{s_i}, \ldots, g_{id}^{s_i}\right)^{(r_1,\ldots,r_d)}[d] = \prod_{i=1}^m \left(g_{i1}, \ldots, g_{id}\right)^{(s_i,0\ldots,0)(r_1,\ldots,r_d)}[d] \\ &= \prod_{i=1}^m \mathbf{g}_i^{(s_i,0,\ldots,0)\mathbf{r}}[d] = \mathbf{y}^{\mathbf{r}}[d]\end{aligned}$$

CPA security in the usual sense follows immediately from Lemma 1. We will, however, argue that the scheme is also leakage resilient in the auxiliary input model.

Auxiliary Input Security The definition of security w.r.t auxiliary inputs is exactly as in [DGK+10].

Definition 7. *A scheme* PKE *is CPA secure w.r.t. auxiliary inputs from a function class* $\mathcal{H}$ *if for any function* $h \in \mathcal{H}$*, any PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, any polynomial* p *and large enough* λ*,*

$$\mathsf{Adv}_{\mathcal{A},h} := \left|\Pr[\mathsf{CPA}_0(\mathsf{PKE}, \mathcal{A}, 1^\lambda, h)] - \Pr[\mathsf{CPA}_1(\mathsf{PKE}, \mathcal{A}, 1^\lambda, h)]\right| < 1/p(\lambda),$$

where $\mathsf{CPA}_b(\mathsf{PKE}, \mathcal{A}, 1^\lambda, h)$ *is output from the following experiment:*

$$\begin{aligned}(pk, sk) &\leftarrow \mathsf{Gen}(1^\lambda) \\ (m_0, m_1, \mathsf{state}) &\leftarrow \mathcal{A}_1(1^\lambda, pk, h(sk, pk)) \text{ with } |m_0| = |m_1| \\ CT^* &\leftarrow \mathsf{Enc}_{pk}(m_b), \\ \text{Output } b' &\leftarrow \mathcal{A}_2(1^\lambda, \mathsf{state}, CT^*)\end{aligned}$$

The functions we will consider are those where the secret key is hard to compute even given the leakage. More precisely, $\mathcal{H}_{ow}(f(k))$ consists of all PT functions $h : \{0,1\}^{|sk|+|pk|} \to \{0,1\}^*$ s.t. given $h(sk, pk)$ (for $(sk, pk) \leftarrow \mathsf{Gen}(1^\lambda)$), no PPT algorithm can find sk with probability greater that $f(k)$. A scheme secure w.r.t auxiliary inputs from $\mathcal{H}_{ow}(f(k))$ is called $f(k)$-AI-CPA secure.

We are now ready to state the theorem about the security of our scheme.

Theorem 10. *Let $m = (4 \log q^d)^{1/\varepsilon}$, for some $\epsilon > 0$. Assuming that d-VDDH is hard for $\mathcal{G}$, the scheme above is $\left(2^{-m^{\varepsilon}}\right)$-AI-CPA secure.*

The complete details of the proof of Theorem 10 are given in the appendix of the full version [CDK+11]. Based on Lemma 1, it follows the exact same lines as in the proof in [DGK+10].

There is a trade-off between the ciphertext size and the hardness of the leakage functions that we can protect against. Obtaining security against functions that are $2^{-m^{\varepsilon}}$-hard to invert, requires that $m = (4 \log q^d)^{1/\varepsilon}$ instead of $m = (4 \log q)^{1/\varepsilon}$, which is a polynomial overhead in the ciphertext size.

We point out that, even though this generalized version of BHHO schemes is auxiliary input secure, KDM security does not follow using our implementation with d-VDDH assumption.

7.3 Pseudorandom Functions

In this section we present a construction for pseudorandom functions (see Definition 3) based on the d-VDDH assumption. This construction is a modification of the DDH-based one in [NR97].

PRF Construction. We construct a function family $F = \{f_k\}$ as follows. The index k specifies a tuple $(q, \mathbb{G}, g_1, g_2, e, \mathbf{a}_0, \ldots, \mathbf{a}_n)$ where q is a prime number, $\mathbb{G}$ is a group of order q, g_1, g_2 are two generators of $\mathbb{G}$, $e : \mathbb{G}^2 \to \mathbb{G}_T$ is a bilinear map and $\mathbf{a}_0, \ldots, \mathbf{a}_n$ are random in R_2. For any such index k we denote $t_1 = e(g_1, g_1), t_2 = e(g_2, g_1)$ and $\mathbf{t} = (t_1, t_2)$. Finally, we define $f_k : \{0,1\}^n \to \mathbb{G}_T$, $f_k(x_1, \ldots, x_n) = \mathbf{t}^{\mathbf{a}_0 \prod_{x_i=1} \mathbf{a}_i}[2]$.

Theorem 11. *Under the 2-VDH assumption, the family $F = \{f_k\}$ defined above is a family of pseudorandom functions.*

We refer to the full version [CDK+11] for the proof of this theorem.

References

[BB04] Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)

[BBS04] Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)

[BHHO08] Boneh, D., Halevi, S., Hamburg, M., Ostrovsky, R.: Circular-Secure Encryption from Decision Diffie-Hellman. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 108–125. Springer, Heidelberg (2008)

[BMR10] Boneh, D., Montgomery, H.W., Raghunathan, A.: Algebraic pseudorandom functions with improved efficiency from the augmented cascade. In: Al-Shaer, E., Keromytis, A.D., Shmatikov, V. (eds.) ACM Conference on Computer and Communications Security, pp. 131–140. ACM (2010)

[CDK+11] Cramer, R., Damgaard, I., Kiltz, E., Zakarias, S., Zottarel, A.: Ddh-like assumptions based on extension rings. Cryptology ePrint Archive, Report 2011/280 (2011), http://eprint.iacr.org/

[CS98] Cramer, R., Shoup, V.: A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)

[DGK+10] Dodis, Y., Goldwasser, S., Kalai, Y.T., Peikert, C., Vaikuntanathan, V.: Public-Key Encryption Schemes with Auxiliary Inputs. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 361–381. Springer, Heidelberg (2010)

[DH76] Diffie, W., Hellman, M.E.: New Directions in Cryptography. IEEE Transactions on Information Theory IT-22(6), 644–654 (1976)

[DY05] Dodis, Y., Yampolskiy, A.: A Verifiable Random Function with Short Proofs and Keys. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 416–431. Springer, Heidelberg (2005)

[Gam84] El Gamal, T.: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (1985)

[HK07] Hofheinz, D., Kiltz, E.: Secure Hybrid Encryption from Weakened Key Encapsulation. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 553–571. Springer, Heidelberg (2007)

[HYZX08] Huang, H., Yang, B., Zhu, S., Xiao, G.: Generalized ElGamal Public Key Cryptosystem Based on a New Diffie-Hellman Problem. In: Baek, J., Bao, F., Chen, K., Lai, X. (eds.) ProvSec 2008. LNCS, vol. 5324, pp. 1–21. Springer, Heidelberg (2008)

[Kil07] Kiltz, E.: Chosen-Ciphertext Secure Key-Encapsulation Based on Gap Hashed Diffie-Hellman. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 282–297. Springer, Heidelberg (2007)

[NR97] Naor, M., Reingold, O.: Number-theoretic constructions of efficient pseudo-random functions. In: FOCS, pp. 458–467 (1997)

[Pip76] Pippenger, N.: On the evaluation of powers and related problems (preliminary version). In: FOCS, pp. 258–263 (1976)

[Sch80] Schwartz, J.T.: Fast probabilistic algorithms for verification of polynomial identities. J. ACM 27, 701–717 (1980)

[Sha07] Shacham, H.: A Cramer-Shoup encryption scheme from the Linear Assumption and from progressively weaker Linear variants. Cryptology ePrint Archive, Report 2007/074 (February 2007), http://eprint.iacr.org/

[Zip79] Zippel, R.: Probabilistic Algorithms for Sparse Polynomials. In: Ng, K.W. (ed.) EUROSAM 1979 and ISSAC 1979. LNCS, vol. 72, pp. 216–226. Springer, Heidelberg (1979)

Security of Blind Signatures Revisited

Dominique Schröder[1,⋆] and Dominique Unruh[2]

[1] University of Maryland, USA
[2] University of Tartu, Estonia

Abstract. We revisit the definition of unforgeability of blind signatures as proposed by Pointcheval and Stern (Journal of Cryptology 2000). Surprisingly, we show that this established definition falls short in two ways of what one would intuitively expect from a secure blind signature scheme: It is not excluded that an adversary submits the same message m twice for signing, and then produces a signature for $m' \neq m$. The reason is that the forger only succeeds if *all* messages are distinct. Moreover, it is not excluded that an adversary performs k signing queries and produces signatures on $k+1$ messages as long as *each* of these signatures does not pass verification with probability 1.

Finally, we propose a new definition, honest-user unforgeability, that covers these attacks. We give a simple and efficient transformation that transforms any unforgeable blind signature scheme (with deterministic verification) into an honest-user unforgeable one.

1 Introduction

Blind signature schemes have been suggested by Chaum [12,13]. Roughly speaking, this widely-studied primitive allows a signer to interactively issue signatures for a user such that the signer learns nothing about the message being signed (*blindness*) while the user cannot compute any additional signature without the help of the signer (*unforgeability*). Typical applications of blind signatures include e-cash, where a bank signs coins withdrawn by users, and e-voting, where an authority signs public keys that voters later use to cast their votes. Another application of blind signature schemes are anonymous credentials, where the issuing authority blindly signs a key [9,10]. Very recently, Microsoft introduced a new technology called U-Prove to "overcome the long standing dilemma between identity assurance and privacy" [6,29]. Their technology uses as a central building block blind signatures [6,8].

There are two main security requirements for blind signature schemes. First, the scheme should be blind. That is, a malicious signer should not be able to link the final signatures output by the user to the individual interactions with the user. In other words, the signer cannot tell which session of the signing protocol corresponds to which message. Second, the scheme should be unforgeable. That is, an adversary, even if he can impersonate the user and interact freely with the signer, should not be able to produce signatures on messages except for those

⋆ Supported in part by a DAAD postdoctoral fellowship.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 662–679, 2012.

that the signer signed. It is the notion of unforgeability we are concerned with in this paper.

A formal definition of the unforgeability of blind signatures schemes (or generally interactive signature schemes) has been proposed by [25]. Roughly, their definition states that an adversary that interacts k times with the adversary cannot produce valid signatures on more than k different messages.[1] At this point, one may wonder why the definition of unforgeability does not just require that the adversary cannot output a signature for m unless there was an interaction with the signer in which m was queried. The reason is that in general, it is not well-defined which message is queried in a given interaction. The message is not sent in clear, and it might be even information-theoretically impossible to tell from an interaction which message is being signed.[2] Thus, in order to be able to tell which message is signed in a given interaction, we would have to add some kind of extractability to the security definition; this would be an additional requirement on the protocols and make them more complex.

Insecurity of Unforgeable Blind Signatures Schemes. Unfortunately, however, the definition of unforgeability might not cover all cases in which one would intuitively expect unforgeability to be sufficient. We illustrate this by the following toy protocol:

Consider the setting of an online video store such as Netflix. In our setting, we assume that the store is implemented via two entities, the content provider and the reseller. We assume that the contract between client and reseller is a flatrate that allows the client to download a fixed number of movies. For privacy reasons, we do not wish the reseller to know which movies the client actually watches. On the other hand, we wish to ensure that underage clients can only download movies suitable for their age. To achieve this, we introduce another (trusted) entity, the parental control server whose job it is to work as a proxy between reseller and client and to ensure that the client only obtains appropriate movies. Then, to download a movie X, the client first sends her name and X to the parental control server. If X is appropriate for the client, the

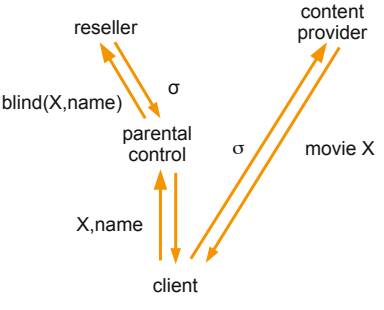


Fig. 1. Setting of an online video store

[1] There is also a variant called strong unforgeability which requires that the adversary cannot produce more than k different message/signature pairs. In particular, this means that the adversary wins even if he produces additional signatures for an already signed message. Since most known blind signature schemes (e.g., [20,15,3,26,19,18]) do not satisfy strong unforgeability, in this work we focus on the weaker notion.

[2] This might be the case when signing a message m is implemented by signing an information-theoretically hiding commitment on m.

parental control server then runs a blind signature scheme with the reseller to obtain a signature σ on (X, name) (the blind signature is used to protect the privacy of the client, there is no need for the reseller to know which movies the client watches). Then σ is sent to the client, and the client uses σ to download X from the content provider. (We assume that all communication is suitably authenticated.)

At a first glance, it seems that this protocol is secure. In particular, the client will not be able to download a movie that is not approved by the parental control server. It turns out, however, that the client can cheat the parental control server: Assume the client twice requests a signature on some harmless movie X. He will then obtain two signatures σ_1, σ_2 on X from the parental control server. Then, given σ_1 and σ_2, the client might be able to compute a signature on an adult movie Y that has not been approved by the parental control server.

It seems that unforgeability should forbid the possibility of such an attack. But it does not. From the point of view of the signer, two signing queries have been performed, and finally signatures on two different messages X and Y have been produced. This does not violate the definition of unforgeability. In fact, we show in Section 4.2 that blind signature schemes exist that allow such attacks but that are still unforgeable.

What went wrong? The definition of unforgeability covers *only partially* the case that the user of the scheme is honest. It only ensures that the number of signed messages is not greater than the number of interactions with the signer. Only considering the number of messages but not their content is fine from the signer's point of view who is not allowed to know the messages anyway. It is not, however, fine from the user's point of view. If the user signs some messages $m_1, \ldots, m_k$ (by interacting with the signer), he expects that no signature on some different message m' can be computed from his signatures. We believe that settings in which the user is honest are natural, and that the definition of unforgeability should cover this case. We thus propose a new *game-based* definition, honest-user unforgeability, which is a strengthening of unforgeability. Alternatively, one could also define an ideal functionality (see [14,4]) that covers these attacks, but schemes that achieve such strong security properties are usually less efficient.

Definition 1 (Honest-User Unforgeability – Informal). *If an adversary performs k direct interactions with the signer, and requests signatures for the message $m_1, \ldots, m_n$ from the user (which produces these signatures by interacting with the signer), then the adversary cannot produce signatures for pairwise distinct messages $m_1^*, \ldots, m_{k+1}^*$ with $\{m_1^*, \ldots, m_{k+1}^*\} \cap \{m_1, \ldots, m_n\} = \varnothing$.*

Notice that this definition also covers the hybrid case in which the adversary interacts with an honest user and the signer simultaneously. Alternatively, one could also require that security in each of the setting individually: Security when there is no honest user (that is, the normal definition of unforgeability), and security when the adversary may not query the signer directly (we call this $\mathcal{S}+\mathcal{U}$-unforgeability). We show in the full version of this paper [28] that requiring these variants of security individually leads to a strictly weaker security notion. Notice

that $\mathcal{S}+\mathcal{U}$-unforgeability would be sufficient to solve the problem in our video store example. It seems, however, restrictive to assume that in all protocols, there will always be only either queries from honest users or only from dishonest users but never from both in the same execution.

Achieving Honest-User Unforgeability. We show that any unforgeable blind signature scheme can be converted into an honest-user unforgeable blind signature scheme. The transformation is very simple and efficient: Instead of signing a message m, in the transformed scheme the user signs the message (m, r) where r is some randomness. Furthermore, we show that if a scheme is already strongly unforgeable, then it is strongly honest-user unforgeable (as long as the original scheme is *randomized* which holds for most signature schemes).

Insecurity with Probabilistic Verification. Most interactive and non-interactive signature schemes have a deterministic verification algorithm. In general, however, having a deterministic verification is not a necessity. Yet, when we allow a probabilistic verification algorithm (and this is usually not excluded), both the definition of unforgeability as well as the definition of honest-user unforgeability are subject to an attack: Consider again our video store example. Let λ denote the security parameter. Fix a polynomial $p = p(\lambda) > \lambda$. Assume that the parental control server and the client are malicious and collude. The parental control server interacts with the reseller λ times, and produces p "half-signatures" on movie names $X_1, \ldots, X_p$. Here, a half-signature means a signature that passes verification with probability $\frac{1}{2}$. Then the client can download the movies $X_1, \ldots, X_n$ from the content provider. (If in some download request, a half-signature does not pass verification, the client just retries his request.) Thus the client got p movies, even if his flatrate only allows for downloading λ movies.

Can this happen? It seems that unforgeability would exclude this because $p > \lambda$ signatures were produced using λ queries to the signer. In the definition of unforgeability, however, the adversary succeeds if it outputs $p > \lambda$ signatures such that *all* signatures pass verification. However, the signatures that are produced are half-signatures: That is, the probability that all $p > \lambda$ signatures pass the verification simultaneously is negligible! Thus, producing more than λ half-signatures using λ queries would not be considered an attack by the definition of unforgeability. In Section 5, we show that blind signature schemes exist that allow such attacks but that satisfy the definition of unforgeability. The same applies to honest-user unforgeability as described so far; we thus need to augment the definition further.

There are two solutions to this problem. One is to explicitly require that the verification algorithm is deterministic. Since most schemes have deterministic verification, this is not a strong restriction. To cover the case of probabilistic verification, we propose an augmented definition of honest-user unforgeability in Section 5: This definition considers a list of signatures as a successful forgery if each of them would pass verification with noticeable probability (roughly speaking).

We do not propose a generic transformation that makes schemes with probabilistic verification secure according to our definition. Yet, since most schemes have a deterministic verification anyway; these schemes will automatically satisfy our augmented definition.

Related Work. Many blind signature schemes have been proposed in the literature, these schemes differ in their round complexity, their underlying computational assumptions, and the model in which the proof of security is given. For example, some schemes rely on the random oracle heuristic [25,2,5,7,4], some constructions are secure in the standard model [11,24,21,23,17,3,27] ([17,3] assume the existence of a common reference string), and some constructions are based on general assumptions [22,14,20,18,27]. Only a few works consider the security of blind signatures [22,25,15] or their round complexity [16].

Notations. Before presenting our results we briefly recall some basic definitions. In what follows we denote by $\lambda \in \mathbb{N}$ the security parameter. Informally, we say that a function is *negligible* if it vanishes faster than the inverse of any polynomial. We call a function non-negligible if it is not negligible. If S is a set, then $x \xleftarrow{\$} S$ indicates that x is chosen uniformly at random over S (which in particular assumes that S can be sampled efficiently).

2 Blind Signatures

To define blind signatures formally we introduce the following notation for interactive executions between algorithms $\mathcal{X}$ and $\mathcal{Y}$. By $(a, b) \leftarrow \langle \mathcal{X}(x), \mathcal{Y}(y) \rangle$ we denote the joint execution of $\mathcal{X}$ and $\mathcal{Y}$, where x is the private input of $\mathcal{X}$ and y defines the private input of $\mathcal{Y}$. The private output of $\mathcal{X}$ equals a and the private output of $\mathcal{Y}$ is b. We write $\mathcal{Y}^{\langle \mathcal{X}(x), \cdot \rangle^{\infty}}(y)$ if $\mathcal{Y}$ can invoke an unbounded number of executions of the interactive protocol with $\mathcal{X}$ in arbitrarily interleaved order. Accordingly, $\mathcal{X}^{\langle \cdot, \mathcal{Y}(y_0) \rangle^1, \langle \cdot, \mathcal{Y}(y_1) \rangle^1}(x)$ can invoke arbitrarily ordered executions with $\mathcal{Y}(y_0)$ and $\mathcal{Y}(y_1)$, but interact with each algorithm only once.

The invoking oracle machine does not see the private output of the invoked machine. In the above definition this means that $\mathcal{Y}$ does not learn a and $\mathcal{X}$ does not learn b_0 (resp. b_1).

Definition 2 (Interactive Signature Scheme). *We define an interactive signature scheme as a tuple of efficient*[3] *algorithms* $\mathsf{BS} = (\mathsf{KG}, \langle \mathcal{S}, \mathcal{U} \rangle, \mathsf{Vf})$ *(the key-generation algorithm* KG*, the signer* $\mathcal{S}$*, the user* $\mathcal{U}$*, and the verification algorithm* Vf*) where*

Key Generation. $\mathsf{KG}(1^\lambda)$ *for parameter* λ *generates a key pair* (sk, pk)*.*

[3] More precisely, KG and Vf run in polynomial-time in the total length of their inputs. The total running time of $\mathcal{S}$ is polynomial in the total length of its input (sk) plus the total length of its incoming messages. The total running time of $\mathcal{U}$ is polynomial in the total length of its input (pk, m). (But the running time of $\mathcal{U}$ may not depend on its incoming messages.) The asymmetry between the running time of $\mathcal{S}$ and $\mathcal{U}$ is necessary to ensure that (a) an interaction between $\mathcal{U}$ and $\mathcal{S}$ always runs in polynomial-time, and (b) that the running-time of $\mathcal{S}$ may depend on the length of the message m that only $\mathcal{U}$ has in its input.

Signature Issuing. *The execution of algorithm* $\mathcal{S}(sk)$ *and algorithm* $\mathcal{U}(pk, m)$ *for message* $m \in \{0,1\}^*$ *generates an output* σ *of the user (and some possibly empty output out for the signer.),* $(out, \sigma) \leftarrow \langle \mathcal{S}(sk), \mathcal{U}(pk, m) \rangle$.

Verification. $\mathsf{Vf}(pk, m, \sigma)$ *outputs a bit.*

It is assumed that the scheme is complete*, i.e., for any function* f*, with overwhelming probability in* $\lambda \in \mathbb{N}$ *the following holds: when executing* $(sk, pk) \leftarrow \mathsf{KG}(1^\lambda)$*, setting* $m := f(\lambda, pk, sk)$*, and letting* σ *be the output by* $\mathcal{U}$ *in the joint execution of* $\mathcal{S}(sk)$ *and* $\mathcal{U}(pk, m)$*, then we have* $\mathsf{Vf}(pk, m, \sigma) = 1$.

3 Security of Blind Signatures

Security of blind signature schemes is defined by unforgeability and blindness [22,25].

Unforgeability. An adversary $\mathcal{U}^*$ against unforgeability tries to generate $k+1$ valid message/signatures pairs with different messages after at most k completed interactions with the honest signer, where the number of executions is adaptively determined by $\mathcal{U}^*$ during the attack. To identify completed sessions we assume that the honest signer returns a special symbol ok when having sent the final protocol message in order to indicate a completed execution (from its point of view). We remark that this output is "atomically" connected to the final transmission to the user.

Definition 3 (Unforgeability). *An interactive signature scheme* $\mathsf{BS} = (\mathsf{KG}, \langle \mathcal{S}, \mathcal{U} \rangle, \mathsf{Vf})$ *is called* unforgeable *if for any efficient algorithm* $\mathcal{A}$*(the malicious user) the probability that experiment* $\mathsf{Unforge}^{\mathsf{BS}}_{\mathcal{A}}(\lambda)$ *evaluates to* 1 *is negligible (as a function of* λ*) where*

Experiment $\mathsf{Unforge}^{\mathsf{BS}}_{\mathcal{A}}(\lambda)$
$(sk, pk) \leftarrow \mathsf{KG}(1^\lambda)$
$((m_1^*, \sigma_1^*), \ldots, (m_{k+1}^*, \sigma_{k+1}^*)) \leftarrow \mathcal{A}^{\langle \mathcal{S}(sk), \cdot \rangle^\infty}(pk)$
Return 1 *iff*
$m_i^* \neq m_j^*$ *for* i, j *with* $i \neq j$*, and*
$\mathsf{Vf}(pk, m_i^*, \sigma_i^*) = 1$ *for all* i*, and*
$\mathcal{S}$ *has returned* ok *in at most* k *interactions.*

An interactive signature scheme is *strongly unforgeable* if the condition "$m_i^* \neq m_j^*$ for i, j with $i \neq j$" in the above definition is substituted by "$(m_i^*, \sigma_i^*) \neq (m_j^*, \sigma_j^*)$ for i, j with $i \neq j$".

Observe that the adversary $\mathcal{A}$ does not learn the private output *out* of the signer $\mathcal{S}(sk)$. We assume schemes in which it can be efficiently determined from the interaction between signer and adversary whether the signer outputs ok. If this is not the case, we need to augment the definition and explicitly give the adversary access to the output *out* since *out* might leak information that the adversary could use to produce forgeries.

Blindness. The blindness condition says that it should be infeasible for a malicious signer $\mathcal{S}^*$ to decide which of two messages m_0 and m_1 has been signed first in two executions with an honest user $\mathcal{U}$. This condition must hold, even if $\mathcal{S}^*$ is allowed to choose the public key maliciously [1]. If one of these executions has returned $\bot$ then the signer is not informed about the other signature either.

Definition 4 (Blindness). *A blind signature scheme* $\mathsf{BS} = (\mathsf{KG}, \langle \mathcal{S}, \mathcal{U} \rangle, \mathsf{Vf})$ *is called* blind *if for any efficient algorithm* $\mathcal{S}^*$ *(working in modes* find*,* issue *and* guess*) the probability that the following experiment* $\mathsf{Blind}^{\mathsf{BS}}_{\mathcal{S}^*}(\lambda)$ *evaluates to* 1 *is negligibly close to* $1/2$*, where*

Experiment $\mathsf{Blind}^{\mathsf{BS}}_{\mathcal{S}^*}(\lambda)$
$(pk, m_0, m_1, \mathsf{st}_{\mathsf{find}}) \leftarrow \mathcal{S}^*(\mathsf{find}, 1^\lambda)$
$b \xleftarrow{\$} \{0,1\}$
$\mathsf{st}_{\mathsf{issue}} \leftarrow \mathcal{S}^{*\langle \cdot, \mathcal{U}(pk,m_b)\rangle^1, \langle \cdot, \mathcal{U}(pk,m_{1-b})\rangle^1}(\mathsf{issue}, \mathsf{st}_{\mathsf{find}})$
and let σ_b, σ_{1-b} *denote the (possibly undefined) local outputs of* $\mathcal{U}(pk, m_b)$ *resp.* $\mathcal{U}(pk, m_{1-b})$*.*
set $(\sigma_0, \sigma_1) = (\bot, \bot)$ *if* $\sigma_0 = \bot$ *or* $\sigma_1 = \bot$
$b^* \leftarrow \mathcal{S}^*(\mathsf{guess}, \sigma_0, \sigma_1, \mathsf{st}_{\mathsf{issue}})$
return 1 *iff* $b = b^*$*.*

4 Honest-User Unforgeability

In this section we introduce a stronger notion of unforgeability that we call *honest-user unforgeability*. In the traditional definition of unforgeability due to [22,25], the adversary fulfills the role of the user. This means that the attacker may choose all messages that are exchanged during the signature issue protocol at will. In particular, the attacker may sample random message *without* fixing a specific message and a certain randomness for the user algorithm. Even if the adversary runs the honest user algorithm, due to the blindness, it is impossible to tell which message has been used. Thus, from a definitional perspective, one has to count the number of executions and produced signatures in order to determine the success condition for the attacker.

This, however, might not be sufficient. Consider an attacker that queries twice the same message m (through, say, some third party honestly implementing the user's algorithm) and is then able to compute a valid signature on some message $m' \neq m$. Since this adversary queried twice the same message, it *still* has to output three distinct messages in order to succeed in the unforgeability game.

In this section we show that giving the attacker, in addition to controlling the user, access to a protocol oracle (that takes as input a message and returns the signature and the user's transcript) yields a strictly stronger definition.

4.1 Defining Honest-User Unforgeability

Before proposing the new definition, we fix some notation. Let $\mathcal{P}(sk, pk, \cdot)$ be an oracle that on input a message m executes the signature issue protocol

$\langle \mathcal{S}(sk), \mathcal{U}(pk, m) \rangle$ obtaining a signature σ. Let trans denote the transcript of the messages exchanges in that interaction. We assume that the transcript consists of all messages exchanged between the parties.[4] This oracle then returns (σ, trans).

Definition 5 (Honest-User Unforgeability). *An interactive signature scheme* $\mathsf{BS} = (\mathsf{KG}, \langle \mathcal{S}, \mathcal{U} \rangle, \mathsf{Vf})$ *is* honest-user unforgeable *if* Vf *is deterministic and the following holds: For any efficient algorithm* $\mathcal{A}$ *the probability that experiment* $\mathsf{HUnforge}^{\mathsf{BS}}_{\mathcal{A}}(\lambda)$ *evaluates to* 1 *is negligible (as a function of* λ*) where*

Experiment $\mathsf{HUnforge}^{\mathsf{BS}}_{\mathcal{A}}(\lambda)$
$(sk, pk) \leftarrow \mathsf{KG}(1^\lambda)$
$((m_1^*, \sigma_1^*), \ldots, (m_{k+1}^*, \sigma_{k+1}^*)) \leftarrow \mathcal{A}^{\langle \mathcal{S}(sk), \cdot \rangle^\infty, \mathcal{P}(sk, pk, \cdot)}(pk)$
Let $m_1, \ldots, m_n$ *be the messages queried to* $\mathcal{P}(sk, pk, \cdot)$.
Return 1 *iff*
$m_i^* \neq m_j$ *for all* i, j
$m_i^* \neq m_j^*$ *for* i, j *with* $i \neq j$, *and*
$\mathsf{Vf}(pk, m_i^*, \sigma_i^*) = 1$ *for all* i, *and*
$\mathcal{S}$ *has returned* ok *in at most* k *interactions.*

(When counting the interactions in which $\mathcal{S}$ *returns* ok*, we do not count the interactions simulated by* $\mathcal{P}$*.)*

An interactive signature scheme is *strongly honest-user unforgeable* if the condition "$m_i^* \neq m_j$ for all i, j" in the above definition is substituted by "$(m_i^*, \sigma_i^*) \neq (m_j, \sigma_j)$ for all i, j" and if we change the condition "$m_i^* \neq m_j^*$ for i, j with $i \neq j$" to "$(m_i^*, \sigma_i^*) \neq (m_j^*, \sigma_j^*)$ for i, j with $i \neq j$".

Notice that we require Vf to be deterministic. When we drop this requirement, the definition does not behave as one would intuitively expect. We explain this problem in detail in Section 5. Note further that this definition can be further strengthened by giving the adversary also the randomness of the honest user. Notice that all our results and proofs also hold for this stronger definition.

4.2 Unforgeability Does Not Imply Honest-User Unforgeability

We show that unforgeability does not imply honest-user unforgeability. The high-level idea of our counterexample is to change the verification algorithm of an interactive signature scheme such that it accepts a message m' if it obtains as input two distinct and valid signatures on some message $m \neq m'$ (in addition to accepting honestly generated signatures). More precisely, fix an unforgeable and blind signature scheme $\mathsf{BS} = (\mathsf{KG}, \langle \mathcal{S}, \mathcal{U} \rangle, \mathsf{Vf})$ that is strongly unforgeable. Fix some efficiently computable injective function $f \neq id$ on bitstrings (e.g., $f(m) := 0 \| m$). We construct a blind signature scheme $\mathsf{BS}_1 = (\mathsf{KG}_1, \langle \mathcal{S}_1, \mathcal{U}_1 \rangle, \mathsf{Vf}_1)$ as follows:

[4] The definition of honest-user unforgeability could be easily strengthened by including the randomness of $\mathcal{U}$ in trans. Our results also hold with respect to that strengthened definition. However, it is not clear that giving the honest-user's randomness to the adversary models any realistic attacks.

- $\mathsf{KG}_1 := \mathsf{KG}$, $\mathcal{S}_1 := \mathcal{S}$, and $\mathcal{U}_1 := \mathcal{U}$.
- $\mathsf{Vf}_1(pk, m, \sigma)$ executes the following steps:
  - Invoke $v := \mathsf{Vf}(pk, m, \sigma)$. If $v = 1$, return 1.
  - Otherwise, parse σ as (σ^1, σ^2). If parsing fails or $\sigma^1 = \sigma^2$, return 0.
  - Invoke $v_i := \mathsf{Vf}(pk, f(m), \sigma^i)$ for $i = 1, 2$. If $v_1 = v_2 = 1$, return 1. Otherwise return 0.

Lemma 6. *If* BS *is complete, strongly unforgeable, and blind, then* BS_1 *is complete, unforgeable, and blind.*

We omit both the proof of blindness and completeness of BS_1 since they follow directly from the blindness and completeness of BS. The unforgeability follows directly from the unforgeability of the underlying scheme. The main idea behind unforgeability is the following: The only possibility for the adversary to forge a signature is to obtain two different signatures σ_1, σ_2 on the same message $f(m)$. Then (σ_1, σ_2) is a valid signature on m. However, since the underlying scheme BS is strongly unforgeable, the adversary can only get σ_1, σ_2 by performing two signing queries. Thus, using two queries, the adversary gets two signatures on the message $f(m)$ and one on m. This is not sufficient to break the unforgeability of BS_1 since the adversary would need to get signatures on three different messages for that. The full proof is given in [28].

Before proving the next lemma, we need to define what a randomized (interactive) signature is. Roughly speaking, schemes that have this property output the same signature in two independent executions with same message only with negligible probability.

Definition 7 (Randomized Signature Scheme). *An interactive signature scheme* $\mathsf{BS} = (\mathsf{KG}, \langle \mathcal{S}, \mathcal{U} \rangle, \mathsf{Vf})$ *is* randomized *if with overwhelming probability in* $\lambda \in \mathbb{N}$ *the following holds: for any* (sk, pk) *in the range of* $\mathsf{KG}(1^\lambda)$*, any message* $m \in \{0,1\}^*$*, we have* $\sigma_1 \neq \sigma_2$ *where* $\sigma_1 \leftarrow \langle \mathcal{S}(sk), \mathcal{U}(pk, m) \rangle$ *and* $\sigma_2 \leftarrow \langle \mathcal{S}(sk), \mathcal{U}(pk, m) \rangle$*.*

Note that any scheme can easily be modified such that is satisfies this definition by letting the user algorithm pick some random value r, setting $m' \leftarrow m \| r$, and by including r in the signature. It is easy to see that, given any randomized interactive signature scheme, we can construct an adversary that queries the oracle $\mathcal{P}$ twice on some message m with $f(m) \neq m$, receives two signatures, $\sigma_1, \neq \sigma_2$ and outputs the pair $(m, (\sigma_1, \sigma_2))$. This pair is a valid forgery for the message $f(m)$ because our adversary has never queried this message to $\mathcal{P}$ and never invoked $\mathcal{S}$ directly. Thus, we immediate get the following lemma (the full proof can be found in [28]).

Lemma 8. *If* BS *is complete and randomized, then* BS_1 *is not honest-user unforgeable.*

By Lemmas 6 and 8 we immediately get:

Theorem 9. *If complete, blind, and strongly unforgeable interactive signature schemes exist, then there are complete, blind, and unforgeable interactive signature schemes that are not honest-user unforgeable.*

Strong Honest-User Unforgeability. The following lemma shows that strong unforgeability implies strong honest-user unforgeability.

Lemma 10. *Assume that* BS *is complete*,[5] *randomized, and strongly unforgeable. Then* BS *is strongly honest-user unforgeable.*

The full proof is delegated to [28]. This lemma shows that for strongly unforgeable schemes, the traditional (non-honest-user) definition of unforgeability is sufficient. Note, however, that most known blind signature schemes (e.g., [20,15,3,26,19,18]) are not strongly unforgeable. It can also easily be shown that strong unforgeability is strictly stronger than honest-user unforgeability. The separating example appends a bit b to the signature that is ignored by the verification algorithm. Then the signature can easily be changed by flipping the bit. Thus honest-user unforgeability lies strictly between unforgeability and strong unforgeability.

5 Probabilistic Verification

In this section we show that, if we allow for a probabilistic verification algorithm, both the definition of honest-user unforgeability, as well as the usual definition of unforgeability will consider schemes to be secure that do not meet the intuitive notion of unforgeability.

One may argue that discussing problems in the definition of blind signature schemes in the case of probabilistic verification is not necessary because one can always just use schemes with deterministic verification. We disagree with this point of view: Without understanding why the definition is problematic in the case of probabilistic verification, there is no reason to restrict oneself to schemes with deterministic verification. Only the awareness of the problem allows us to circumvent it. We additionally give a definition that works in the case of probabilistic verification. This is less important than pointing out the flaws, since in most cases one can indeed use schemes with deterministic verification. But there might be (rare) cases where this is not possible (note that no generic transformation outside the random oracle model is known that makes the verification deterministic).

First, we give some intuition for our counterexample and formalize it afterwards. Assume an interactive signature scheme BS_3 that can distinguishes two kinds of signatures: A full-signature that will pass verification with probability 1, and a half-signature that passes verification with probability $\frac{1}{2}$. An honest interaction between the signer $\mathcal{S}_3$ and the user $\mathcal{U}_3$ will always produce a full-signature.

[5] Completeness is actually necessary to show this lemma: For example, let BS' be a scheme derived from a complete and strongly unforgeable scheme BS in the following way: All machines except for the user are the same in BS and BS'. When the user $\mathcal{U}'$ should sign a message m, he signs $m+1$ instead. Since the user does not occur in the definition of strong unforgeability, the strong unforgeability of BS implies the strong unforgeability of BS'. Yet BS' is not strongly honest-user unforgeable: By performing a signature query for m from the user $\mathcal{U}'$, the adversary can get a valid signature for $m+1$.

A malicious user, however, may interact with the signer to get a half-signature for arbitrary messages. Furthermore, the malicious user may, by sending λ half-signatures to the signer (λ is the security parameter) and executing a special command, get two half-signatures instead of one. ("Buy $\lambda+1$ signatures, get one free.") At the first glance, one would expect that such a scheme cannot be honest-user unforgeable or even unforgeable. But in fact, the adversary has essentially two options: First, he does not request λ half-signatures. Then he will not get a signature for free and thus will not win in the honest-user unforgeability game. Second, he does request λ half-signatures and then performs the extra query and thus gets $\lambda+2$ half-signatures using $\lambda+1$ queries. Then, to win, he needs that all $\lambda+2$ signatures pass verification (since the definition of unforgeability/honest-user unforgeability requires that $\mathsf{Vf}_3(pk, m_i^*, \sigma_i^*)$ evaluates to 1 for all signatures (m_i^*, σ_i^*) output by the adversary) However, since each half-signature passes verification with probability $\frac{1}{2}$, the probability that all signatures pass verification is negligible ($\leq 2^{-\lambda}$). Thus, the adversary does not win, and the scheme is honest-user unforgeable. Clearly, this is not what one would expect; so Definition 5 should not be applied to the case where the verification is probabilistic (and similarly the normal definition of unforgeability should not be applied either in that case).

More precisely, let $\mathsf{BS} = (\mathsf{KG}, \langle \mathcal{S}, \mathcal{U} \rangle, \mathsf{Vf})$ be a randomized, complete, blind, and honest-user unforgeable interactive signature scheme. Let Q be an efficiently decidable set such that the computation of arbitrarily many bitstrings $m \in Q$ and $m' \notin Q$ is efficiently feasible.

We define the scheme $\mathsf{BS}_3 = (\mathsf{KG}_3, \langle \mathcal{S}_3, \mathcal{U}_3 \rangle, \mathsf{Vf}_3)$ as follows:

- $\mathsf{KG}_3 := \mathsf{KG}$.
- $\mathcal{S}_3(sk)$ behaves like $\mathcal{S}(sk)$, except when the first message from the user is of the form $(\texttt{extrasig}, m_1^\circ, \ldots, m_\lambda^\circ, \sigma_1^\circ, \ldots, \sigma_\lambda^\circ, m_1', \ldots, m_q')$ where λ is the security parameter. Then $\mathcal{S}_3$ executes the following steps:
  - Check whether $m_1^\circ, \ldots, m_\lambda^\circ \in Q$ are pairwise distinct messages, and for all $i = 1, \ldots, q$ we have $m_i' \notin Q$, and for all $i = 1, \ldots, \lambda$ we have $\mathsf{Vf}(pk, 1\|m_i^\circ, \sigma_i^\circ) = 1$.[6] If not, ignore the message.
  - If the check passes, run $\langle \mathcal{S}(sk), \mathcal{U}(pk, 1\|m_i') \rangle$ for each $i = 1, \ldots, q$, resulting in signatures $\tilde{\sigma}_i$, and set $\sigma_i' := 1\|\tilde{\sigma}_i$.
  - Then $\mathcal{S}_3$ sends $(\sigma_1', \ldots, \sigma_n')$ to the user, outputs ok and does not react to any further messages in this session.
- $\mathcal{U}_3(pk, m)$ runs $\sigma \leftarrow \mathcal{U}(pk, 0\|m)$ and returns $0\|\sigma$.
- $\mathsf{Vf}_3(pk, m, \sigma)$ performs the following steps:
  - If $\sigma = 0\|\sigma'$ and $\mathsf{Vf}(pk, 0\|m, \sigma') = 1$, Vf_3 returns 1.
  - If $\sigma = 1\|\sigma'$ and $\mathsf{Vf}(pk, 1\|m, \sigma) = 1$, Vf_3 returns 1 with probability $p := \frac{1}{2}$ and 0 with probability $1 - p$.
  - Otherwise, Vf_3 returns 0.

Lemma 11. *If* BS *is blind and complete, so is* BS_3.

[6] Without loss of generality, we assume that the public key pk can efficiently be computed from the secret key sk.

Proof. Blindness and completeness of BS_3 follow directly from that of BS. The only difference between the schemes is that instead of a message m, a message $0\|m$ is signed and 0 is prepended to the signatures (as long as the user is honest as is the case in the definitions of blindness and completeness).

Lemma 12. *If* BS *is honest-user unforgeable, so is* BS_3.

The proof idea was already explained at the beginning of this section. The complete proof is given in [28].
The following lemma shows that, although BS_3 is honest-user unforgeable (and thus also unforgeable), it should not be considered secure! Namely, an adversary can, given λ queries, produce $\lambda+1$ message/signature pairs, each of which passes verification with probability $\frac{1}{2}$. In particular in a setting where the machine which verifies the signatures is stateless and where the adversary may thus just resubmit a rejected signature, such signatures are as good as signatures that pass verification with probability 1. Thus, the adversary has essentially forged one signature.

An adversary that queries the signer λ times on distinct messages (from Q) is able to execute the special command that allows to produce an arbitrary number of half-signatures. Thus, we immediate get (see [28] for the full proof):

Lemma 13. *We call* (m, σ) *a half-signature (with respect to some implicit public-key* pk*) if the probability that* $\mathsf{Vf}(pk, m, \sigma) = 1$ *is* $1/2$*. If* BS *is complete, then for any polynomial* p*, there is an adversary* $\mathcal{A}$ *that performs* $\lambda + 1$ *interactions with* $\mathcal{S}_3$ *and does not query* $\mathcal{P}$ *and that, with overwhelming probability, outputs* $p(\lambda)$ *half-signatures* $(m_1^*, \sigma_1^*), \ldots, (m_{p(\lambda)}^*, \sigma_{p(\lambda)}^*)$ *such that all* m_i^* *are distinct.*

5.1 Adapting the Definition

We have shown that, if we allow for a probabilistic verification algorithm in the definition of honest-user unforgeability (and similarly in the definition of unforgeability), schemes that are intuitively insecure will be considered secure by the definition. There are two possible ways to cope with this problem.

The simplest solution is to require that the verification algorithm is deterministic. This is what we did in Section 4.1 (Definition 5). This choice is justified since almost all known blind signature schemes have a deterministic verification algorithm anyway. Thus restricting the verification algorithm to be deterministic may be preferable to getting a more complicated definition.[7]

In some cases, however, it might not be possible to make the verification deterministic. In such cases, it is necessary to strengthen the definition of honest-user unforgeability. Looking back at our counterexample, the problem was the following: If the adversary produces many signatures that each pass verification

[7] Notice that one could weaken the requirement and only require that two invocations of the verification algorithm output the same value with overwhelming probability. This would allow for verification algorithms that essentially compute a deterministic function but have to solve problems in BPP during that computation.

with non-negligible but not overwhelming probability, this is not considered an attack: The probability that all signatures pass verification simultaneously is negligible. In order to fix this problem, we thus need to change the definition in such a way that a signature that is accepted with non-negligible probability is always considered a successful forgery. More precisely, if a signature passes verification at least once when running the verification algorithm a polynomial number of times, then the signature is considered valid. This idea leads to the following definition:

Definition 14 (Honest-User Unforgeability with Probabilistic Verification). *Given a probabilistic algorithm* Vf *and an integer* t*, we define* Vf^t *as follows:* $\mathsf{Vf}^t(pk, m, \sigma)$ *runs* $\mathsf{Vf}(pk, m, \sigma)$ t*-times. If one of the invocations of* Vf *returns* 1*,* Vf^t *returns* 1*. If all invocations of* Vf *return* 0*,* Vf^t *returns* 0*.*

A blind signature scheme $\mathsf{BS} = (\mathsf{KG}, \langle \mathcal{S}, \mathcal{U} \rangle, \mathsf{Vf})$ *is called* honest-user unforgeable (with probabilistic verification) *if the following holds: For any efficient algorithm* $\mathcal{A}$ *and any polynomial* p*, the probability that experiment* $\mathsf{HUnforge}^{\mathsf{BS}}_{\mathcal{A}}(\lambda)$ *evaluates to* 1 *is negligible (as a function of* λ*) where*

Experiment $\mathsf{HUnforge}^{\mathsf{BS}}_{\mathcal{A}}(\lambda)$
$(sk, pk) \leftarrow \mathsf{KG}(1^\lambda)$
$((m_1^*, \sigma_1^*), \ldots, (m_{k+1}^*, \sigma_{k+1}^*)) \leftarrow \mathcal{A}^{\langle \mathcal{S}(sk), \cdot \rangle^\infty, \mathcal{P}(sk, pk, \cdot)}(pk)$
Let $m_1, \ldots, m_n$ *be the messages queried to* $\mathcal{P}(sk, pk, \cdot)$*.*
Return 1 *iff*
$m_i^* \neq m_j$ *for all* i, j
$m_i^* \neq m_j^*$ *for* i, j *with* $i \neq j$*, and*
$\mathsf{Vf}^{p(\lambda)}(pk, m_i^*, \sigma_i^*) = 1$ *for all* i*, and*
$\mathcal{S}$ *has returned* ok *in at most* k *interactions.*

(When counting the interactions in which $\mathcal{S}$ *returns* ok*, we do not count the interactions simulated by* $\mathcal{P}$*.)*

Notice that the only difference to Definition 5 is that we additionally quantify over a polynomial p, and use $\mathsf{Vf}^{p(\lambda)}$ instead of Vf. If a signature is accepted with non-negligible probability, then there is a polynomial p such that $\mathsf{Vf}^{p(\lambda)}$ will accept that signature with overwhelming probability. (For our counterexample BS_3, one can choose $p(\lambda) := \lambda$ to show that it does not satisfy Definition 14.)

Notice that there is no obvious transformation for taking a signature scheme satisfying the regular unforgeability definition and constructing a scheme secure with respect to Definition 14 out of it. One obvious approach would be to include the randomness for verification in the message and thus to make the scheme deterministic. This might, however, make the scheme totally insecure because in this case a forger might include just the right randomness to get a signature accepted (if that signature would be accepted with negligible but non-zero probability otherwise). Another obvious approach would be to change the verification algorithm such that it verifies each signature p times (for a suitable polynomial p) and only accepts when all verifications succeed. This would make, e.g., half-signatures into signatures with negligible acceptance probability. But also this

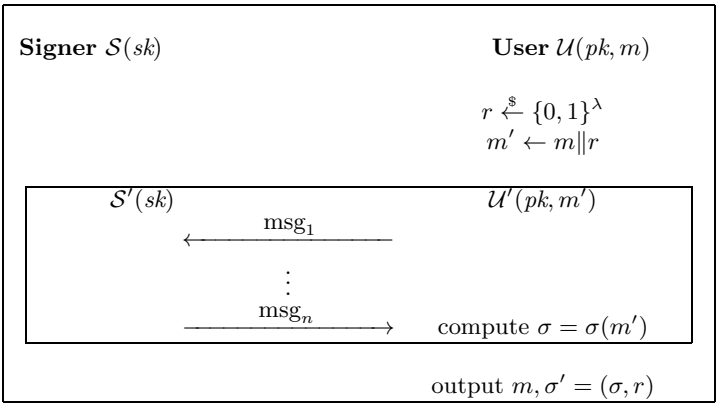


Fig. 2. Issue protocol of the blind signature scheme

approach does not work in general: For any p, the adversary might be able to produce signatures that fails each individual verification with probability $1/2p$ and thus passes the overall verification with constant probability.

6 From Unforgeability to Honest-User Unforgeability

In this section we show how to turn any unforgeable interactive signature scheme into an honest-user unforgeable one. Our transformation is extremely efficient as it only adds some randomness to the message. Therefore, it not only adds a negligible overhead to original scheme, but it also preserves all underlying assumptions. The construction is formally defined in Construction 1 and depicted in Figure 2.

Construction 1. *Let* $\mathsf{BS}' = (\mathsf{KG}', \langle \mathcal{S}', \mathcal{U}' \rangle, \mathsf{Vf}')$ *be an interactive signature scheme and define the signature scheme* BS *through the following three procedures:*

Key Generation. *The algorithm* $\mathsf{KG}(1^\lambda)$ *runs* $(sk', pk') \leftarrow \mathsf{KG}'(1^\lambda)$ *and returns this key-pair.*

Signature Issue Protocol. *The interactive signature issue protocol for message* $m \in \{0,1\}^*$ *is described in Figure 2.*

Signature Verification. *The input of the verification algorithm* Vf *is a public key* pk*, a message* m*, and a signature* $\sigma' = (\sigma, r)$*. It sets* $m' \leftarrow (m \| r)$ *and returns the result of* $\mathsf{Vf}'(pk, m \| r, \sigma)$*.*

We first show that our transformation preserves completeness and blindness.

Lemma 15. *If* BS' *is a complete and blind interactive signature scheme, so is* BS*.*

Since the proof follows easily, we omit it here.

Now, we prove that our construction turns any unforgeable scheme into an honest-user unforgeable one.

Lemma 16. *If* BS' *is an unforgeable interactive signature scheme, then* BS *is secure with respect to Definition 5.*

Proof. Assume for the sake of contradiction that BS is not honest-user unforgeable. Then there exists an efficient adversary $\mathcal{A}$ that wins the honest-user unforgeability game with non-negligible probability. We then show how to build an attacker $\mathcal{B}$ that breaks the unforgeability of BS'.

The input of the algorithm $\mathcal{B}$ is a public *pk*. It runs a black-box simulation of $\mathcal{A}$ and simulates the oracles as follows. Whenever $\mathcal{A}$ engages in an interactive signature issue protocol with the signer, i.e., when the algorithm $\mathcal{A}$ plays the role of the user, then $\mathcal{B}$ relays all messages between $\mathcal{A}$ and the signer. If $\mathcal{A}$ invokes the oracle $\mathcal{P}$ on a message m, then $\mathcal{B}$ picks a random $r \xleftarrow{\$} \{0,1\}^\lambda$, sets $m' \leftarrow m\|r$, and engages in an interactive signature issue protocol where $\mathcal{B}$ runs the honest user algorithm $\mathcal{U}'$. At the end of this protocol, the algorithm $\mathcal{B}$ obtains a signature σ on the message m'. It sets $\sigma' \leftarrow (\sigma, r)$, stores the pair (m', σ') in a list L and returns σ' together with the corresponding transcript trans to the attacker $\mathcal{A}$.

Eventually, the algorithm $\mathcal{A}$ stops, outputting a sequence of message/signature pairs $(m_1^*, \sigma_1^*), \ldots, (m_{k+1}^*, \sigma_{k+1}^*)$. In this case, $\mathcal{B}$ recovers all message/signature pairs $(m_1', \sigma_1'), \ldots, (m_n', \sigma_n')$ stored in L, it parses σ_i^* as (σ_i', r_i'), it sets $\widetilde{m}_i \leftarrow m_i^*\|r_i^*$ and $\widetilde{\sigma} \leftarrow \sigma_i'$ for all $i = 1, \ldots, k+1$ and outputs $(m_1', \sigma_1'), \ldots, (m_n', \sigma_n')$, $(\widetilde{m}_1, \widetilde{\sigma}_1), \ldots, (\widetilde{m}_{k+1}, \widetilde{\sigma}_{k+1})$.

Analysis. For the analysis first observe that $\mathcal{B}$ runs in polynomial time because $\mathcal{A}$ is efficient and because the handling of all queries can be done efficiently. Suppose that $\mathcal{A}$ succeeds with non-negligible probability. Then it outputs $(k+1)$ message/signature pairs that verify under Vf. Since $\mathcal{B}$ runs the honest user algorithm to compute the signatures $\sigma_1', \ldots, \sigma_n'$ it follows (from the completeness) that all message/signature pairs that $\mathcal{B}$ returns, verify with overwhelming probability. It is left to show that a) the algorithm $\mathcal{B}$ output one more message/signature pair (than queries to the signing oracle with output ok took place) and b) all messages are distinct.

The distinctness property follows immediately from the definition of the success probability in the honest-user unforgeability game and from the construction. More precisely, consider the messages $(m_1', \ldots, m_n')$ and $(\widetilde{m}_1, \ldots, \widetilde{m}_{k+1})$, where $m_i' = m_i\|r_i$ and $\widetilde{m}_j = m_j^*\|r_j^*$. According to our assumption that $\mathcal{A}$ succeeds, it follows that all message pairs m_r^* and m_s^* (for all $r \neq s$) differ from each other. But then it follows easily that $\widetilde{m}_r^*$ and $\widetilde{m}_s^*$ are also distinct (for all $r \neq s$). Since the r_i are chosen randomly, the messages $(m_1', \ldots, m_n')$ also differ from each other with overwhelming probability. Now, consider the messages $(m_1, \ldots, m_n)$ that $\mathcal{A}$ sends to the oracle $\mathcal{P}$. Note that all these messages must differ from the messages $(m_1^*, \ldots, m_{k+1}^*)$ returned by $\mathcal{A}$ by definition. This means, however, that $\widetilde{m}_r^*$ differs from m_i' for all i, r.

Finally we have to show that $\mathcal{B}$ returns one more message/signature pair (property (a)) than protocol executions with the signer $\mathcal{S}'$ took place (and that produced output ok). Since $\mathcal{A}$ wins the game, it follows that in at most k of the protocol executions that $\mathcal{B}$ forwarded between $\mathcal{A}$ and its external signer, the signer returned ok. $\mathcal{B}$ itself has executed n user instances to simulate the oracle $\mathcal{P}$. Since $\mathcal{A}$ outputs $k+1$ message signature pair (s.t. $m_i \neq m_j^*$ for all i, j) it follows that $\mathcal{B}$ has asked at most $n+k$ queries in which the signer $\mathcal{S}'$ returned ok, but $\mathcal{B}$ returned $n+k+1$ message/signature pairs. This, however, contradicts the assumption that BS is unforgeable.

Putting together the above results, we get the following theorem.

Theorem 17. *If complete, blind, and unforgeable interactive signature schemes exist, then there are complete, blind, unforgeable, and honest-user unforgeable interactive signature schemes (with respect to Definition 5).*

The proof of this theorem follows directly from Lemmas 15 and 16.

Acknowledgments. We thank the anonymous reviewers for valuable comments. This research was supported by the Cluster of Excellence "Multimodal Computing and Interaction", by the European Social Fund's Doctoral Studies and Internationalisation Programme DoRa, by the European Regional Development Fund through the Estonian Center of Excellence in Computer Science, EXCS, and by European Social Fund through the Estonian Doctoral School in Information and Communication Technology.

References

1. Abdalla, M., Namprempre, C., Neven, G.: On the (Im)possibility of Blind Message Authentication Codes. In: Pointcheval, D. (ed.) CT-RSA 2006. LNCS, vol. 3860, pp. 262–279. Springer, Heidelberg (2006)
2. Abe, M.: A Secure Three-Move Blind Signature Scheme for Polynomially Many Signatures. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 136–151. Springer, Heidelberg (2001)
3. Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-Preserving Signatures and Commitments to Group Elements. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 209–236. Springer, Heidelberg (2010)
4. Abe, M., Ohkubo, M.: A Framework for Universally Composable Non-committing Blind Signatures. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 435–450. Springer, Heidelberg (2009)
5. Bellare, M., Namprempre, C., Pointcheval, D., Semanko, M.: The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. Journal of Cryptology 16(3), 185–215 (2003)
6. Bjones, R.: U-prove technology overview (October 2010), http://www.itforum.dk/downloads/Ronny_Bjones_Uprove.pdf
7. Boldyreva, A.: Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 31–46. Springer, Heidelberg (2002)

8. Brands, S., Paquin, C.: U-prove cryptographic specification v1.0 (March 2011), http://connect.microsoft.com/site642/Downloads/DownloadDetails.aspx?DownloadID=26953
9. Brands, S.A.: Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy. MIT Press, Cambridge (2000)
10. Camenisch, J., Groß, T.: Efficient attributes for anonymous credentials. In: Ning, P., Syverson, P.F., Jha, S. (eds.) ACM CCS 2008: 15th Conference on Computer and Communications Security, Alexandria, Virginia, USA, October 27-31, pp. 345–356. ACM Press (2008)
11. Camenisch, J., Koprowski, M., Warinschi, B.: Efficient Blind Signatures Without Random Oracles. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 134–148. Springer, Heidelberg (2005)
12. Chaum, D.: Blind signatures for untraceable payments. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) Advances in Cryptology – CRYPTO 1982, Santa Barbara, CA, USA, pp. 199–203. Plenum Press, New York (1983)
13. Chaum, D.: Blind signature system. In: Chaum, D. (ed.) Advances in Cryptology – CRYPTO 1983, Santa Barbara, CA, USA, p. 153. Plenum Press, New York (1984)
14. Fischlin, M.: Round-Optimal Composable Blind Signatures in the Common Reference String Model. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 60–77. Springer, Heidelberg (2006)
15. Fischlin, M., Schröder, D.: Security of Blind Signatures under Aborts. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 297–316. Springer, Heidelberg (2009)
16. Fischlin, M., Schröder, D.: On the Impossibility of Three-Move Blind Signature Schemes. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 197–215. Springer, Heidelberg (2010)
17. Fuchsbauer, G.: Automorphic signatures in bilinear groups and an application to round-optimal blind signatures. Cryptology ePrint Archive, Report 2009/320 (2009), http://eprint.iacr.org/
18. Garg, S., Rao, V., Sahai, A., Schröder, D., Unruh, D.: Round Optimal Blind Signatures. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 630–648. Springer, Heidelberg (2011)
19. Ghadafi, E., Smart, N.P.: Efficient two-move blind signatures in the common reference string model. Cryptology ePrint Archive, Report 2010/568 (2010), http://eprint.iacr.org/
20. Hazay, C., Katz, J., Koo, C.-Y., Lindell, Y.: Concurrently-Secure Blind Signatures Without Random Oracles or Setup Assumptions. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 323–341. Springer, Heidelberg (2007)
21. Horvitz, O., Katz, J.: Universally-Composable Two-Party Computation in Two Rounds. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 111–129. Springer, Heidelberg (2007)
22. Juels, A., Luby, M., Ostrovsky, R.: Security of Blind Digital Signatures (Extended Abstract). In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 150–164. Springer, Heidelberg (1997)
23. Kiayias, A., Zhou, H.-S.: Equivocal Blind Signatures and Adaptive UC-Security. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 340–355. Springer, Heidelberg (2008)
24. Okamoto, T.: Efficient Blind and Partially Blind Signatures Without Random Oracles. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 80–99. Springer, Heidelberg (2006)

25. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. Journal of Cryptology 13(3), 361–396 (2000)
26. Rückert, M.: Lattice-Based Blind Signatures. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 413–430. Springer, Heidelberg (2010)
27. Schröder, D., Unruh, D.: Round optimal blind signatures. Cryptology ePrint Archive, Report 2011/264 (2011), http://eprint.iacr.org/
28. Schröder, D., Unruh, D.: Security of blind signatures revisited. Cryptology ePrint Archive, Report 2011/316 (2011), http://eprint.iacr.org/
29. MICROSOFT U-PROVE. Microsoft u-prove ctp release 2 (March 2011), http://connect.microsoft.com/site642/Downloads/DownloadDetails.aspx?DownloadID=26953

Efficient Network Coding Signatures in the Standard Model

Dario Catalano[1], Dario Fiore[2,⋆], and Bogdan Warinschi[3]

[1] Dipartimento di Matematica e Informatica, Università di Catania, Italy
catalano@dmi.unict.it
[2] Department of Computer Science, New York University, USA
fiore@cs.nyu.edu
[3] Dept. Computer Science, University of Bristol, UK
bogdan@cs.bris.ac.uk

Abstract. Network Coding is a routing technique where each node may actively modify the received packets before transmitting them. While this departure from passive networks improves throughput and resilience to packet loss it renders transmission susceptible to *pollution attacks* where nodes can misbehave and change in a malicious way the messages transmitted. Nodes cannot use standard signature schemes to authenticate the modified packets: this would require knowledge of the original sender's signing key. Network coding signature schemes offer a cryptographic solution to this problem. Very roughly, such signatures allow signing vector spaces (or rather bases of such spaces), and these signatures are homomorphic: given signatures on a set of vectors it is possible to create signatures for any linear combination of these vectors. Designing such schemes is a difficult task, and the few existent constructions either rely on random oracles or are rather inefficient. In this paper we introduce two new network coding signature schemes. Both of our schemes are provably secure in the standard model, rely on standard assumptions, *and* are in the same efficiency class as previous solutions based on random oracles.

1 Introduction

Network Coding [1,23] is an elegant and novel routing approach that is alternative to traditional routing where each node simply stores and forwards the incoming packets. The main difference is that in Network Coding intermediate nodes can modify data packets in transit, still allowing the final recipients to obtain the original information.

More specifically, we consider a network setting where a *source* node wants to transmit a piece of information (a file) to a set of *target* nodes. The source node splits the file into m network packets and sends them to its neighboring nodes. An intermediate node who receives a set of packets from its incoming links, modifies them and sends the resulting packets into the network through its outgoing edges. In *Linear Network Coding* packets are seen as vectors in a

⋆ Work done while at École Normale Supérieure.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 680–696, 2012.

linear space over some field and the modifications by the intermediate nodes are linear combinations of these vectors. Such linear combinations can be performed by using ad-hoc coefficients (e.g., fixed by the application or defined by a central authority), or random coefficients chosen by the intermediate nodes in a suitable domain. The latter case is referred to as *Random (Linear) Network Coding*. In addition to offering a more decentralized approach, random network coding has been shown to perform almost as well as network coding with ad-hoc coefficients [12,16,18]. One important aspect of linear network coding is that it enables target nodes to recover the original information with high probability if they receive sufficiently many correct packets. Interestingly, the target nodes can do so without knowledge of the coefficients chosen by the intermediate nodes. We give a more detailed description of these techniques in Section 2.2.

The original motivation for network coding was to increase throughput in decentralized networks and indeed, the technique performs well in wireless/ad-hoc network topologies where a centralized control may not be available. For example, it has been suggested as a good means to improve file sharing in peer-to-peer networks [22], and digital content distribution over the Internet [15].

The main issue of (random) linear network coding is its susceptibility to *pollution attacks* in which malicious nodes (or simple network error transmission) may inject into the network invalid packets to prevent the target nodes from reconstructing the original information. In the specific setting of linear network coding, an invalid packet is simply a vector outside the space spanned by the initial m vectors sent by the source node. In turn, intermediary nodes can later use the invalid incoming vectors thus generating even more invalid packets. This means that errors may dramatically propagate through the network, and adversaries might easily mount a Denial of Service attack to prevent the file from being reconstructed by only injecting *a few* invalid packets.

Two main approaches have been proposed to deal with this problem. One is information-theoretic and uses error-correction techniques [17,18,20]. Unfortunately, this introduces redundant information that badly affects the communication efficiency. The other approach (the one considered in our work) relies on computational assumptions and uses cryptographic techniques. Here, the main idea is to provide a way to authenticate valid vectors. However, standard authentication techniques, such as MACs or digital signatures, do not solve the problem as we want to grant the intermediate nodes some malleability on the received vectors.

The main tool that has been proposed to achieve this goal employs *network coding signature* schemes [7]. In a few words, a network coding signature allows to sign a linear subspace $\mathcal{W} \subset \mathbb{F}^N$ in such a way that a signature σ on $\mathcal{W}$ is verified only by those vectors $w \in \mathcal{W}$.

These schemes can be constructed either from *homomorphic hash functions*, or from *homomorphic signatures*. Very briefly, a homomorphic hash function H satisfies the property that for any vectors a, b and scalar coefficients α and β, it holds that $H(\alpha a + \beta b) = H(a)^{\alpha} H(b)^{\beta}$. Constructions based on homomorphic hashing [22,16,7,14] are less recent and their security can be based on well-established

assumptions in the standard model, such as solving discrete log or factoring. The main drawback of this approach is that the public key and the authentication information that has to be sent along with the packets are linear in the size m of the vector space and thus defeats the purpose of increasing the throughput. Furthermore, the sender has to know the entire file before sending the first packet (which is undesirable for example in the ubiquitous streaming applications).

In contrast, solutions based on homomorphic signatures [7,14,3,11] are more communication-efficient, even though they are computationally somewhat more expensive than those built from homomorphic hashing. In a nutshell, a homomorphic signature is a special type of signature scheme that enjoys a linear homomorphic property: for any vectors a, b and scalar coefficients α and β, it holds that $\mathsf{Sign}(\alpha a + \beta b) = \mathsf{Sign}(a)^{\alpha}\mathsf{Sign}(b)^{\beta}$. More formally, this means that the scheme is equipped with a $\mathsf{Combine}$ algorithm that given μ signatures $\sigma_1, \ldots, \sigma_\mu$ on vectors $w_1, \ldots, w_\mu$ respectively, and scalar coefficients $\alpha_1, \ldots, \alpha_\mu$, it can compute a signature σ which is valid with respect to the vector $w = \sum_{i=1}^{\mu} \alpha_i \cdot w_i$. Importantly, the combination operation does not require the secret key. The security notion for this primitive requires that an adversary who receives signatures on a set of vectors $w_1, \ldots, w_m$ should be able to generate only signatures on vectors that lie in the linear span of $(w_1, \ldots, w_m)$. It should be clear at this point how this primitive can be used to secure the network coding-based application (see Section 2.4 for a detailed description) and, more generally, enable authenticated computation of linear functions of signed data [2].

Related Work. Since our work focuses on homomorphic network coding signatures, in this section we describe the most significant works in this topic. The notion of homomorphic signature was first introduced by Johnson, Molnar, Song and Wagner in a more general setting [21] and only recently adapted to the particular application for network coding by Boneh, Freeman, Katz and Waters [7]. In their work, Boneh *et al.* propose an efficient construction over bilinear groups and prove its security from the CDH assumption in the random oracle model. One year later, Gennaro, Katz, Krawkzyk and Rabin [14] proposed another implementation of homomorphic network coding signatures based on RSA in the random oracle model. Moreover, as an additional contribution, they showed that even if the homomorphic signature works over a large finite field (or over the integers), it is possible to use small coefficients in the linear combinations, and this significantly improves the efficiency at the intermediate nodes in the network coding application. In [9] Boneh and Freeman give the construction of a homomorphic network coding signature based on lattices. As a new property, their scheme allows to authenticate vectors defined over binary fields, and is based on the problem of finding short vectors in integer lattices. The security of this construction relies on the random oracle heuristic. In addition, the same paper shows a scheme in the standard model, but this scheme is only k-time secure (a signing key can be used to issue only k signatures, where k is fixed in advance). In a subsequent work [8], Boneh and Freeman proposed the notion of homomorphic signatures for polynomial functions. While all previous works considered schemes whose homomorphic property allows to compute only linear

functions on the signed data, the scheme in [8] is capable to evaluate multivariate polynomials. Their construction uses ideal lattices and its security is proven in the random oracle model.

The problems associated to the use of the random oracles are well-known and significant research effort is invested in devising implementations that do not rely on this heuristic. For network coding such constructions proved elusive – and we are only aware of two such proposals [3,11][1].

In [3] Attrapadung and Libert give an implementation over bilinear groups of composite order, using the dual system techniques of Waters [24] to carry on the security proof. Unfortunately the scheme relies on the setting of composite order groups and is thus highly inefficient. Furthermore, even if the scheme were to be converted to gropus of prime order (as suggested, but not fully described in [3]), the efficiency gap between the resulting construction and those in the random oracle solutions is still significant.

The most recent proposal is by Catalano, Fiore and Warinschi who propose a homomorphic network coding signature as an application of the notion of Adaptive Pseudo-Free groups [11]. In particular, the concrete implementation is secure in the standard model under the Strong RSA assumption. While from the point of view of computation the efficiency of this scheme is not far from that of the random oracle construction of Gennaro *et al.* which also works in the RSA group, the signature's size in [11] is much worse than that in [14], as it is very affected by the large random exponent s (that is 1346 bits long if one considers 80 bits of security).

Our Contribution. In this work we design two new homomorphic network coding signatures with security proofs in the standard model. Our realizations outperform in efficiency the two currently known constructions in the standard model [3,11] and achieve computational and communication efficiency comparable to those of the random oracle implementations [7,14].

Our first scheme works over asymmetric bilinear groups of prime order p, and is secure under the q-Strong Diffie Hellman assumption (q-SDH for short) introduced by Boneh and Boyen [6]. The construction adapts ideas from the signature by Hofheinz and Kiltz [19] which in turn is based on the concept of Programmable Hash Functions. There, a signature is a random $r \in \mathbb{Z}_p$ and a group element X that is a solution of $X^{z+r} = H(M)$, where z is the secret key, and H is the programmable hash function. To obtain a solution for signing vector spaces along the same lines, we developed some non-trivial extensions which roughly speaking deal with the fact that in our case the same random exponent has to be reused for several signatures. In our construction, a signature on a vector $w = (u, v) \in \mathbb{F}_p^{m+n}$ consists of a random element $s \in \mathbb{Z}_p$ and the

[1] We mention that the random oracle based solution given in [7] might be turned into a scheme secure in the standard model if one is willing to give up the homomorphic property. This makes the resulting solution much less interesting in practice as the signer would need to sign all the vectors in the given subspace at once.

solution X to the following equation:

$$X^{z+\mathsf{fid}} = h^s h_1^{u_1} \cdots h_m^{u_m} g_1^{v_1} \cdots g_n^{v_n}$$

where $\mathsf{fid} \in \mathbb{Z}_p$ represents the random file identifier and z is the secret key. We can therefore achieve rather short signatures: one group element plus an element of $\mathbb{Z}_p$, that is, about 512 bits for 128 bits of security.

Our second realization works over $\mathbb{Z}_N^*$ where N is the product of two safe primes pq. The scheme can be seen as an optimization of the construction by Catalano-Fiore-Warinschi where the random exponent s can now be taken as small as $2k$ bits (where k denotes the desired bit security). The signature on a vector $w = (u, v) \in \mathbb{F}^{m+n}$ is a random integer $s \in \mathbb{Z}_e$ and the solution x to the equation

$$x^e = g^s h_1^{u_1} \cdots h_m^{u_m} g_1^{v_1} \cdots g_n^{v_n} \bmod N$$

where e is a random prime representing the file identifier, and $g, h_1, \ldots, h_m$, $g_1, \ldots, g_n \in \mathbb{Z}_N^*$ are in the public key. As an additional improvement, we show how to do linear combinations $\pmod e$, allowing for the signature scheme to be used in networks with paths of any lengths. This was not the case in [11] and [14] where the parameters have to be set according to a bound L on the maximum length of a path between the source and the target nodes in the network.

A more detailed efficiency analysis of our schemes as well as comparisons with previous solutions, are given in Section 5.

Concurrent Work. In concurrent and independent work Freeman has proposed a semi-generic transformation for building linearly-homomorphic signatures from standard signature schemes [13]. This transformation yields new linearly homomorphic signature schemes that are secure in the standard model under a new security notion (introduced in [13]) which is slightly stronger than the one considered in our work. Our schemes are different from the ones obtained in [13] enjoy better efficiency. It is of future interest to check whether they also satisfy the stronger notion of security proposed in [13].

2 Background and Definitions

In what follows we will denote with $k \in \mathbb{N}$ a security parameter. We say that a function $\epsilon : \mathbb{N} \to \mathbb{R}^+$ is negligible if and only if for every positive polynomial $p(k)$ there exists a $k_0 \in \mathbb{N}$ such that for all $k > k_0$: $\epsilon(k) < 1/p(k)$. If S is a set, we denote with $x \xleftarrow{\$} S$ the process of selecting x uniformly at random in S. Let $\mathcal{A}$ be a probabilistic algorithm. We denote with $x \xleftarrow{\$} \mathcal{A}(\cdot)$ the process of running $\mathcal{A}$ on some appropriate input and assigning its output to x.

2.1 Computational Assumptions

An integer N is called *RSA modulus* if it is the product of two distinct prime numbers pq. The Strong RSA Assumption was introduced by Baric and Pfitzmann in [4]. Informally, the assumption states that given a public RSA modulus

N, and a random value $z \in \mathbb{Z}_N$, any PPT adversary cannot compute an e-th root of z for an $e \neq 1$ of its choice.

Definition 1 (Strong RSA Assumption). *Let N be a random RSA modulus of length k where $k \in \mathbb{N}$ is the security parameter, and z be a random element in $\mathbb{Z}_N$. Then we say that the Strong RSA assumption holds if for any PPT adversary $\mathcal{A}$ the probability*

$$Pr[(y, e) \leftarrow \mathcal{A}(N, z) : y^e = z \bmod N \wedge e \neq 1]$$

is negligible in k.

Let $\mathbb{G}, \mathbb{G}'$ and $\mathbb{G}_T$ be bilinear groups of prime order p such that $e : \mathbb{G} \times \mathbb{G}' \to \mathbb{G}_T$ is a bilinear map. The q-Strong Diffie-Hellman Assumption (q-SDH for short) was introduced by Boneh and Boyen in [5] and it is defined as follows.

Definition 2 (q-SDH Assumption). *Let $k \in \mathbb{N}$ be the security parameter, $p > 2^k$ be a prime, and $\mathbb{G}, \mathbb{G}', \mathbb{G}_T$ be bilinear groups of the same order p such that g and g' are the generators of $\mathbb{G}$ and $\mathbb{G}'$ respectively. Then we say that the q-SDH Assumption holds in $\mathbb{G}, \mathbb{G}', \mathbb{G}_T$ if for any PPT algorithm $\mathcal{A}$ and any $q = poly(k)$, the following probability (taken over the random choice of x and the random coins of $\mathcal{A}$) is negligible in k*

$$\Pr[\mathcal{A}(g, g^x, g^{x^2}, \cdots, g^{x^q}, g', (g')^x) = (c, g^{1/(x+c)})]$$

2.2 Background on Linear Network Coding

In linear network coding [1,23] a file to be transmitted is viewed as a set of n-dimensional vectors $(v^{(1)}, \ldots, v^{(m)})$ defined over the integers or over some finite field. To transmit a file $\mathcal{V} = (v^{(1)}, \ldots, v^{(m)})$ the source node creates m augmented vectors $(w^{(1)}, \ldots, w^{(m)})$ where each $w^{(i)}$ is obtained by prepending to $v^{(i)}$ a vector $u^{(i)}$ of length m, i.e., $w^{(i)} = (u^{(i)}, v^{(i)})$. Precisely, $(u^{(1)}, \ldots, u^{(m)})$ represents the canonical basis of $\mathbb{Z}^m$, that is $u^{(i)}$ is the i-th unitary vector, with 1 in position i and 0 elsewhere. This way, the vectors $w^{(1)}, \ldots, w^{(m)}$ form a basis of a subspace $\mathcal{W} \subset \mathbb{F}^{m+n}$. Vectors $w^{(i)}$ of the above form are called *properly augmented vectors* while $(w^{(1)}, \ldots, w^{(m)})$ is a *properly augmented basis*.

In this setting, the *source* node sends these vectors as packets in the network. Whenever a node in the network receives $(w^{(1)}, \ldots, w^{(\mu)})$ on its μ incoming edges, it computes a linear combination $\hat{w}$ of the received vectors and transmits $\hat{w}$ in the network through its outgoing edges. The coefficients used in the linear combination can be fixed by the application, established by a central authority, or they can be randomly chosen by each node. The latter is the case considered in our work and it is called "random network coding". As shown in [12,16,18], random network coding performs almost as well as linear network coding with ad-hoc coefficients. To recover the original file a node must receive m (valid) vectors $\hat{w}^{(1)}, \ldots, \hat{w}^{(m)}$ of the form described before, i.e., $\hat{w}^{(i)} = (\hat{u}^{(i)}, \hat{v}^{(i)})$. In particular, in order for the file to be reconstructed, the vectors $(\hat{u}^{(1)}, \ldots, \hat{u}^{(m)})$

need to be linearly independent. Let denote with $\hat{U}$ the matrix whose rows are the vectors $(\hat{u}^{(1)}, \ldots, \hat{u}^{(m)})$ and with $\hat{\mathcal{V}}$ the matrix whose rows are the vectors $(\hat{v}^{(1)}, \ldots, \hat{v}^{(m)})$. Then, the original file can be retrieved by computing

$$\mathcal{V} = \hat{U}^{-1} \cdot \hat{\mathcal{V}}.$$

Although the above described approach solves the problem of recovering the information in network coding, as we mentioned in the introduction, the main issue in this approach is that it is susceptible to *pollution attacks* where malicious nodes may inject invalid packets in the network so that the reconstruction of the original file becomes impossible. This is particularly sensitive also because a single error introduced by a (malicious) node can be propagated by honest nodes.

Before describing solutions, we observe how two trivial approaches do not solve the problem. First, the source node cannot simply sign the transmitted packets as the receivers are likely to get modified versions of them (by the effect of the linear combinations). Second, the source could sign the entire file. This would prevent the receivers to accept incorrect files, but it does not provide an efficient way for the receivers to recover the correct file as malicious nodes can still inject invalid packets to mount a DoS attack.

To mitigate the effect of pollution attacks two main approaches have been proposed. They can be divided into two categories: *information-theoretic* and *computational*.

Information theoretic approaches [17,18,20] use error-correction techniques to introduce redundancy in the transmitted vectors so that it is possible to reconstruct the original file as long as the number of compromised vectors is not too big. These methods have the advantage of not relying on computational assumptions, but, unfortunately, they introduce a significant overhead in the communication.

On the other hand, approaches based on computational assumptions use cryptographic techniques to provide a way for honest nodes to verify that the received packets are correct. The main tool to achieve this goal are *network coding signature schemes*. Roughly speaking, the basic requirement of such schemes is that they allow to efficiently check if a given vector is valid, i.e., it has been generated as linear combination of initial (valid) vectors $w^{(1)}, \ldots, w^{(m)}$. Two classes of network coding signatures are known: those based on homomorphic hashing [22,16,7], and those using homomorphic signatures [21,7,14,11].

In our work, we focus on the second class of schemes, that is homomorphic network coding signatures. We give relevant definitions in the following section.

2.3 Network Coding Signatures

In this section we give the definition of a network coding signature scheme and its security notion, as done by Boneh *et al.* in [7]. As we mentioned before, a network coding signature scheme allows to sign a subspace $\mathcal{W} \subset \mathbb{F}^N$ so that any vector $w \in \mathcal{W}$ is accepted, whereas vectors $w \notin \mathcal{W}$ are rejected. In particular, in

our work we focus on subspaces $\mathcal{W}$ that are described by a properly augmented basis.

We assume that a file is associated with a file identifier fid that is chosen by the source node before the transmission. In general, such fid can be the filename. Though, in our systems we need such file identifiers to be randomly chosen by the source node. Thus we think of fid as an element of an efficiently samplable set $\mathcal{I}$.

Definition 3 (Network Coding Signatures). *A* network coding signature *is defined by a triple of algorithms (*$\mathsf{NetKG}, \mathsf{NetSign}, \mathsf{NetVer}$*) such that:*

$\mathsf{NetKG}(1^k, m, n)$ *On input the security parameter k and two integers m, n, this algorithm outputs* $(\mathsf{vk}, \mathsf{sk})$ *where* sk *is the secret signing key and* vk *is the public verification key. m defines the dimension of the vector spaces while n is an upper bound to the size of the signed vectors. We assume that the public key implicitly defines the field $\mathbb{F}$ over which vectors and linear combinations are defined.*

$\mathsf{NetSign}(\mathsf{sk}, \mathsf{fid}, \mathcal{W})$ *The signing algorithm takes as input the secret key* sk*, a random file identifier* fid *and a properly augmented basis of a m-dimensional subspace $\mathcal{W} \subset \mathbb{F}^{m+\ell}$ (with $1 \leq \ell \leq n$), and it outputs a signature σ.*

$\mathsf{NetVer}(\mathsf{vk}, \mathsf{fid}, w, \sigma)$ *Given the public key* vk*, a file identifier* fid*, a vector $w \in \mathbb{F}^{m+\ell}$ (for $1 \leq \ell \leq n$) and a signature σ, the algorithm outputs 0 (reject) or 1 (accept).*

For correctness, we require that for all honestly generated key pairs $(\mathsf{vk}, \mathsf{sk})$, all identifiers $\mathsf{fid} \in \mathcal{I}$, all $1 \leq \ell \leq n$, and all $\mathcal{W} \subset \mathbb{F}^{m+\ell}$, if $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{fid}, \mathcal{W})$ then $\mathsf{Ver}(\mathsf{vk}, \mathsf{fid}, w, \sigma) = 1 \ \forall w \in \mathcal{W}$.

Security of Network Coding Signatures. The security notion of network coding signatures is defined by the following game between a challenger and an adversary $\mathcal{A}$:

Setup. The adversary chooses positive integers m, n and gives them to the challenger. The challenger runs $(\mathsf{vk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{NetKG}(1^k, m, n)$ and gives vk to $\mathcal{A}$.

Signing queries. The adversary can ask signatures on vector spaces $\mathcal{W}_i \subset \mathbb{F}^{m+\ell}$ (with $\ell \leq n$) of its choice, specified by giving to the challenger a properly augmented basis describing $\mathcal{W}_i$. The challenger chooses a random file identifier fid_i, runs $\sigma_i \xleftarrow{\$} \mathsf{NetSign}(\mathsf{sk}, \mathsf{fid}_i, \mathcal{W}_i)$ and returns σ_i to $\mathcal{A}$.

Forgery. The adversary outputs a tuple $(\mathsf{fid}^*, w^*, \sigma^*)$.

We say that the adversary wins this game if $\mathsf{NetVer}(\mathsf{vk}, \mathsf{fid}^*, v^*, \sigma^*) = 1$ and either one of the following cases holds: (1) $\mathsf{fid}^* \neq \mathsf{fid}_i$ for all i (*type-I forgery*); (2) $\mathsf{fid}^* = \mathsf{fid}_i$ for some i, but $w^* \notin \mathcal{W}_i$ (*type-II forgery*).

We define the advantage of $\mathcal{A}$ into breaking a network coding signature scheme, $\mathbf{Adv}^{NC}(\mathcal{A})$, as the probability that $\mathcal{A}$ wins the above security game, and we say that a network coding signature is secure if for any PPT $\mathcal{A}$, $\mathbf{Adv}^{NC}(\mathcal{A})$ is at most negligible in the security parameter.

Finally, we give the formal definition of *homomorphic network coding signature*.

Definition 4 (Homomorphic Network Coding Signatures). *A* homomorphic network coding signature scheme *is defined by a 4-tuple of algorithms (*$\mathsf{NetKG}, \mathsf{NetSign}, \mathsf{NetVer}, \mathsf{Combine}$*) such that:*

$\mathsf{NetKG}(1^k, m, n)$**.** *On input the security parameter k and two integers $m, n \geq 1$, this algorithm outputs $(\mathsf{vk}, \mathsf{sk})$ where sk is the secret signing key and vk is the public verification key. Here, m defines the dimension of the vector spaces and $n + m$ is an upper bound to the size of the signed vectors. We assume that the public key implicitly defines the field $\mathbb{F}$ over which vectors and linear combinations are defined, and that it contains the description of an efficiently samplable distribution for fid.*

$\mathsf{NetSign}(\mathsf{sk}, \mathsf{fid}, w)$**.** *The signing algorithm takes as input the secret key sk, a file identifier in the support of fid and a vector $w \in \mathbb{F}^{\ell+m}$ (with $1 \leq \ell \leq n$) and outputs a signature σ.*

$\mathsf{NetVer}(\mathsf{vk}, \mathsf{fid}, w, \sigma)$**.** *Given the public key vk, a file identifier fid, a vector $w \in \mathbb{F}^{\ell}$ and a signature σ, the algorithm outputs 0 (reject) or 1 (accept).*

$\mathsf{Combine}(\mathsf{vk}, \mathsf{fid}, \{(w^{(i)}, \alpha_i, \sigma_i)\}_{i=1}^{\mu})$**.** *This algorithm takes as input the public key vk, a file identifier fid, and a set of tuples $(w^{(i)}, \alpha_i, \sigma_i)$ where σ_i is a signature, $w^{(i)} \in \mathbb{F}^{\ell}$ is a vector and $\alpha_i \in \mathbb{F}$ is a scalar. This algorithm outputs a new signature σ such that: if each σ_i is a valid signature on vector $w^{(i)}$, then σ is a valid signature for w obtained from the linear combination $\sum_{i=1}^{\mu} \alpha_i \cdot w^{(i)}$.*

For correctness, we require that for all $m, n \geq 1$, all honestly generated pairs of keys $(\mathsf{vk}, \mathsf{sk}) \stackrel{\$}{\leftarrow} \mathsf{NetKG}(1^k, m, n)$ the following hold:

- For all $\mathsf{fid} \in \mathcal{I}$ and all $w \in \mathbb{F}^{m+\ell}$, if $\sigma \stackrel{\$}{\leftarrow} \mathsf{NetSign}(\mathsf{sk}, \mathsf{fid}, w)$, then $\mathsf{NetVer}(\mathsf{vk}, \mathsf{fid}, w, \sigma) = 1$.
- For all $\mathsf{fid} \in \mathcal{I}$, any $\mu > 0$, and all sets of triples $\{(w^{(i)}, \alpha_i, \sigma_i)\}_{i=1}^{\mu}$, if $\mathsf{NetVer}(\mathsf{vk}, \mathsf{fid}, w^{(i)}, \sigma_i) = 1$ for all i, then it must be the case that

$$\mathsf{NetVer}(\mathsf{vk}, \mathsf{fid}, \sum \alpha_i w^{(i)}, \mathsf{Combine}(\mathsf{vk}, \mathsf{fid}, \{(w^{(i)}, \alpha_i, \sigma_i)\}_{i=1}^{\mu}) = 1.$$

As noticed by Boneh et al. [7], homomorphic network coding signatures are a special case of network coding signatures.

2.4 An Efficient Linear Network Coding Scheme

In this section we specify the linear network coding scheme considered in our work. Basically, it is the random network coding solution described in the previous section except that we consider some optimizations recently proposed by Gennaro *et al.* in [14]. The scheme works as follows.

The application specifies four global parameters $m, n, M, p' \in \mathbb{N}$ such that $m, n \geq 1$, and p' is a prime. In this setting, a file $\mathcal{V}$ to be transmitted is always encoded as a set of m vectors $(v^{(1)}, \ldots, v^{(m)})$ where each $v^{(i)}$ takes values in $\mathbb{F}_M^{\ell}$

where M is a bound on the initial magnitude of each coordinate and $\ell \leq n$. Since m is fixed in advance by the application, at the time of the transmission, once the size of the file $\mathcal{V}$ is known, the total length of information in every vector $v^{(i)}$ is determined. Thus, ℓ can be chosen accordingly as any number between 1 and n. The freedom in choosing ℓ is important as different choices have different impact on the efficiency of the scheme: a smaller ℓ saves bandwidth, while a larger ℓ saves computation (see [14] for more details). The parameter p' specifies the domain $P = \{0, \ldots, p'-1\}$ from which the network nodes sample the coefficients for the linear combination. Linear combinations can then be performed either over the integers, or modulo some large prime p (which is specified by the application or by the signature scheme). Gennaro et *al.* show that taking a small p' (e.g., $p' = 257$) allows to improve the performances of the network coding scheme as well as to keep a good decoding probability. In particular, they show that this holds in both cases when the linear combinations are done over the integers, or over some large prime $p > M$. Precisely, in the latter case, the performances remain better (than the case when coefficients are chosen in $\mathbb{F}_p$) as long as the bit-size of p' is negligible compared to the bit-size k of the prime p.

Global application parameters: $m, n, M, p' \in \mathbb{N}$ as specified above.

Key Generation: Each source node generates a pair of keys $(\mathsf{vk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{NetKG}(1^k, m, n)$ of a homomorphic network coding signature scheme.

File transmission: On input a file $\mathcal{V}$ represented by m vectors $v^{(1)}, \ldots, v^{(m)} \in \mathbb{F}_M^\ell$ (with $\ell \leq n$), the source node generates augmented vectors $w^{(1)}, \ldots, w^{(m)}$, i.e., $w^{(i)} = (u^{(i)}, v^{(i)})$ where $u^{(i)}$ is the i-th unity vector. Next, it chooses a random file identifier $\mathsf{fid} \xleftarrow{\$} \mathcal{I}$ (recall that $\mathcal{I}$ is specified by vk), and for $i = 1$ to m, it generates $\sigma_i \xleftarrow{\$} \mathsf{NetSign}(\mathsf{sk}, \mathsf{fid}, w^{(i)})$. Finally, it sends the tuples $(\mathsf{fid}, w^{(i)}, \sigma_i)$ on its outgoing edges.

Intermediate nodes: When a node receives μ vectors $w^{(1)}, \ldots, w^{(\mu)}$ and signatures $\sigma_1, \ldots, \sigma_\mu$, all corresponding to file fid, it proceeds as follows. First, it checks that $\mathsf{NetVer}(\mathsf{vk}, \mathsf{fid}, w^{(i)}, \sigma_i) = 1$, for $i = 1$ to μ. It discards all the vectors (and signatures) that did not pass the check. For the remaining vectors (for simplicity, let they be $w^{(1)}, \ldots, w^{(\mu)}$), the node chooses $\alpha_1, \ldots, \alpha_\mu \xleftarrow{\$} P$, and computes: $w = \sum_{i=1}^{\mu} \alpha_i \cdot w^{(i)}$, $\sigma \leftarrow \mathsf{Combine}(\mathsf{vk}, \mathsf{fid}, \{(w^{(i)}, \alpha_i, \sigma_i)\}_{i=1}^{\mu})$. Finally, the node sends $(\mathsf{fid}, w, \sigma)$ on its outgoing edges.

Target node: Once a node obtains linearly independent vectors $w^{(1)}, \ldots, w^{(m)}$ together with the respective signatures and the same file identifier fid, it first checks that they are all valid, i.e., it verifies that $\mathsf{NetVer}(\mathsf{vk}, \mathsf{fid}, w^{(i)}, \sigma_i) = 1$, $\forall i = 1, \ldots, m$. Given m valid vectors, the node can reconstruct the original file $(v^{(1)}, \ldots, v^m)$ as described in Section 2.2.

3 A Construction Based on SDH

In this section we propose the construction of a network coding homomorphic signature based on the Strong Diffie-Hellman assumption.

Recall that we are in the setting of the linear network coding application described in the previous section. A file $\mathcal{V}$ is represented as a set of m vectors $(v^{(1)}, \ldots, v^{(m)})$ such that each $v^{(i)} \in \mathbb{F}_p^\ell$ where p is a (publicly known) prime specified by the key generation algorithm and $\ell \leq n$. Notice that all the operations with the vectors are thus defined over the finite field $\mathbb{F}_p$, i.e., $\bmod p$. Moreover, the space for file identifiers is the set $\mathbb{Z}_p^*$ where p is the same prime specified in the key generation.

Below we give a precise description of the scheme's algorithms[2]:

$\mathsf{NetKG}(1^k, n, m)$**:** Let $\mathbb{G}, \mathbb{G}', \mathbb{G}_T$ be bilinear groups of prime order p such that $e : \mathbb{G} \times \mathbb{G}' \to \mathbb{G}_T$ is a bilinear map and $g \in \mathbb{G}, g' \in \mathbb{G}'$ are two generators. Pick a random $z \xleftarrow{\$} \mathbb{Z}_p$ and set $Z = (g')^z$. Choose random elements $h, h_1, \ldots, h_m, g_1, \ldots, g_n \xleftarrow{\$} \mathbb{G}$. Output the public verification key $\mathsf{vk} = (p, g, g', Z, h, h_1, \ldots, h_m, g_1, \ldots, g_n)$ and the secret key $\mathsf{sk} = z$.

$\mathsf{NetSign}(\mathsf{sk}, \mathsf{fid}, w)$**:** Let $w = (u, v) \in \mathbb{F}_p^{m+n}$ be a properly augmented vector, and let fid be randomly chosen in $\mathbb{Z}_p^*$. The signing algorithm proceeds as follows. Pick a random $s \xleftarrow{\$} \mathbb{Z}_p$ and compute

$$X = \left(h^s \prod_{i=1}^{m} h_i^{u_i} \prod_{i=1}^{n} g_i^{v_i} \right)^{\frac{1}{z+\mathsf{fid}}}$$

Finally, output $\sigma = (X, s)$.

$\mathsf{NetVer}(\mathsf{vk}, \mathsf{fid}, w, \sigma)$**:** Let $\sigma = (X, s) \in \mathbb{G} \times \mathbb{Z}_p$. This algorithm checks whether σ is a valid signature on a vector $w = (u, v)$ w.r.t. the file identifier fid. If the following equation holds, then output 1, otherwise output 0:

$$e(X, Z \cdot (g')^{\mathsf{fid}}) = e(h^s \prod_{i=1}^{m} h_i^{u_i} \prod_{i=1}^{n} g_i^{v_i}, g').$$

$\mathsf{Combine}(\mathsf{vk}, \mathsf{fid}, \{w^{(i)}, \alpha_i, \sigma_i\}_{i=1}^{\mu})$**:** Recall that $w^{(i)} = (u^{(i)}, v^{(i)})$ where $u^{(i)} \in \mathbb{F}_p^m$ and $v^{(i)} \in \mathbb{F}_p^n$, and that $\alpha_i \in \mathbb{F}_p$ is a randomly chosen coefficient, for all $i \in \{1, \ldots, \mu\}$. Moreover, recall that in our application this algorithm is run when every σ_i has been verified as a valid signature on $w^{(i)}$ w.r.t. fid. The algorithm computes

$$X = \prod_{i=1}^{\mu} (X_i)^{\alpha_i}, \quad s = \sum_{i=1}^{\mu} \alpha_i \cdot s_i \bmod p$$

and outputs $\sigma = (X, s)$.

Efficiency. A signature consists of one element of $\mathbb{G}$ and one element of $\mathbb{Z}_p$. Signing costs a multi-exponentiation in $\mathbb{G}$, whereas verification needs to compute two pairings, one exponentiation in $\mathbb{G}'$, i.e., $(g')^{\mathsf{fid}}$, and one multi-exponentiation.

[2] For ease of exposition, in our description we assume that the vectors w have the maximum length $m+n$. In fact, in our scheme any shorter vector with $\ell < n$ can be augmented by appending $n - \ell$ zeros.

We state the following theorem (for lack of space its proof appears in the full version of this paper [10])

Theorem 1. *If the q-SDH assumption holds in* $(p, \mathbb{G}, \mathbb{G}', \mathbb{G}_T)$ *for any polynomial* q*, then the scheme described above is a secure network coding signature.*

4 A (Strong) RSA Based Realization

In this section we describe our strong-RSA based implementation. We stress that the file to be signed is encoded as a set of vectors $(v^{(1)}, \ldots, v^{(m)})$ of ℓ components each where $\ell \leq n$ for some pre-specified bound n. Before being signed and transmitted, such vectors will be prepended with m unitary vectors $u^{(i)}$ (each having m components). We denote with $w^{(i)}$ the resulting vectors. Our implementation uses a parameter λ to specify the space $\mathcal{I}$ for the file identifiers. If M is the bound on the initial magnitude of each vector component, then $2^\lambda > M$ and $\mathcal{I}$ is the set of prime numbers of (exactly) $\lambda + 1$ bits, greater than 2^λ.

Finally, we notice that in this scheme the exact finite field over which are done the linear combinations is different for each file. In particular, it will be $\mathbb{F}_e$ where $e = \mathsf{fid}$ (e is a prime number) is the file identifier chosen by the sender. More precisely, this means that whenever a vector space $\mathcal{W}$ has to be signed, a file identifier $\mathsf{fid} = e$ is chosen (as a sufficiently large prime) and it is associated to $\mathcal{W}$. Thus, linear combinations are done $\bmod e$ and $w \notin \mathcal{W}$ implies that w cannot be written as a linear combination $\bmod e$ of vectors of $\mathcal{W}$.

A precise description of our network coding scheme $\mathsf{NetPFSig} = (\mathsf{NetKG}, \mathsf{NetSign}, \mathsf{NetVer}, \mathsf{Combine})$ follows.

$\mathsf{NetKG}(1^k, \lambda, m, n)$**.** The NetKG algorithm chooses two random (safe) primes p, q of length $k/2$ each. It sets $N = pq$ and proceeds by choosing $g, g_1, \ldots, g_n$, $h_1, \ldots, h_m$ at random (in $\mathbb{Z}_N^*$). In addition to k, here we assume an additional security parameter λ which specifies the space $\mathcal{I}$ of file identifiers as described before. The public key is set as $(N, g, g_1, \ldots, g_n, h_1, \ldots, h_m)$, while the secret key is (p, q).

$\mathsf{NetSign}(\mathsf{sk}, \mathsf{fid}, w)$**.** The signing algorithm proceeds as follows. Let $w = (u, v) \in \mathbb{F}_M^{m+n}$ and let fid be a random file identifier, which is a prime number of the form specified before. For ease of exposition, let $e = \mathsf{fid}$. The signer chooses a random element $s \in \mathbb{Z}_e$ and uses its knowledge of p and q to solve the following equation

$$x^e = g^s \prod_{j=1}^{m} h_j^{u_j} \prod_{j=1}^{n} g_j^{v_j} \bmod N$$

Finally, it outputs the signature $\sigma = (s, x)$.

$\mathsf{NetVer}(\mathsf{vk}, \mathsf{fid}, w, \sigma)$**.** To verify a signature $\sigma = (s, x)$ on a vector w, the verification algorithm proceeds as follows. Let $e = \mathsf{fid}$.

- Check that e is an odd number of the right size (i.e. $\lambda + 1$ bits).
- Check that all the u's, v's and s are in $\mathbb{Z}_e$.

– Check that the equation $x^e = g^s \prod_{j=1}^{m} h_j^{u_j} \prod_{j=1}^{n} g_j^{v_j} \bmod N$ is satisfied by the given x.
– If all the checks above are satisfied, output 1, otherwise 0.

$\mathsf{Combine}(\mathsf{vk}, \mathsf{fid}, \{w^{(i)}, \alpha_i, \sigma_i\}_{i=1}^{\mu})$. To combine signatures σ_i, corresponding to vectors $w^{(i)}$ sharing the same fid, the algorithm proceeds as follows.

– It computes

$$w = \sum_{i=1}^{\mu} \alpha_i \cdot w^{(i)} \bmod e, \; w' = (\sum_{i=1}^{\mu} \alpha_i \cdot w^{(i)} - w)/e$$

$$s = \sum_{i=1}^{\mu} \alpha_i s_i \bmod e, \; s' = (\sum_{i=1}^{\mu} \alpha_i s_i - s)/e$$

Let $w' = (u', v')$. It outputs $\sigma = (s, x)$ where x is obtained by computing:

$$x = \frac{\prod_{i=1}^{\mu} x_i^{\alpha_i}}{g^{s'} \prod_{j=1}^{m} h_j^{u'_j} \prod_{j=1}^{n} g_j^{v'_j}} \bmod N$$

To complete the description of the scheme we show its correctness. In particular, while the correctness of the signatures returned by the signing algorithm can be easily checked by inspection, we pause to show that also the signatures obtained from the $\mathsf{Combine}$ algorithm are correct. Assume that for $i = 1$ to μ, $\sigma_i = (x_i, s_i)$ is a valid signature on the vector $w^{(i)} = (u^{(i)}, v^{(i)})$, and let α_i be the integer coefficients of the linear combination. Let $\sigma = (x, s)$ be the signature as computed by $\mathsf{Combine}(\mathsf{vk}, \mathsf{fid}, \{w^{(i)}, \alpha_i, \sigma_i\}_{i=1}^{\mu})$. We have that:

$$x^e = \frac{\prod_{i=1}^{\mu} (x_i^e)^{\alpha_i}}{(g^{s'} \prod_{j=1}^{m} h_j^{u'_j} \prod_{j=1}^{n} g_j^{v'_j})^e} \tag{1}$$

$$= \frac{g^{\sum_{i=1}^{\mu} s_i \alpha_i} \prod_{j=1}^{m} h_j^{\sum_{i=1}^{\mu} u_j^{(i)} \alpha_i} \prod_{j=1}^{n} g_j^{\sum_{i=1}^{\mu} v_j^{(i)} \alpha_i}}{(g^{s'} \prod_{j=1}^{m} h_j^{u'_j} \prod_{j=1}^{n} g_j^{v'_j})^e} \tag{2}$$

$$= g^{(\sum_{i=1}^{\mu} s_i \alpha_i - s'e)} \prod_{j=1}^{m} h_j^{(\sum_{i=1}^{\mu} u_j^{(i)} \alpha_i - u'_j e)} \prod_{j=1}^{n} g_j^{(\sum_{i=1}^{\mu} v_j^{(i)} \alpha_i - v'_j e)} \tag{3}$$

$$= g^s \prod_{j=1}^{m} h_j^{u_j} \prod_{j=1}^{n} g_j^{v_j} \tag{4}$$

which shows correctness as desired. Above, equation (2) is justified by that each σ_i is valid, and equation (4) follows from the definition of s' and $w' = (u', v')$ as computed in the $\mathsf{Combine}$ algorithm.

Efficiency. Each signature consists of an element of $\mathbb{Z}_N$ and one integer of λ bits. Signing costs one full exponentiation and one multi-exponentiation in $\mathbb{Z}_N$ with

λ-bits exponents, plus the sampling of a random prime number (which is dominated by the cost of prime verification). The verification needs an exponentiation with a $(\lambda + 1)$-bits prime, x^e, and one multi-exponentiation with λ-bits exponents.

Here we state the following theorem (again for lack of space the proof appears in [10]).

Theorem 2. *Under the Strong-RSA assumption, the scheme described above is a secure homomorphic network coding signature.*

5 Efficiency and Comparisons

In this section we discuss the efficiency of our two constructions and compare it to that of other known homomorphic network coding signatures. As we already mentioned, there are not that many schemes in the literature realizing this primitive: a few constructions [7,14,9,8] rely on random oracles, and a couple of more recent schemes [3,11,13] are proven secure in the standard model. We should also mention that there are other schemes in the standard model based on homomorphic hashing. However these are less appealing in practice mainly because the basis vectors have to be signed all at once, which means that in the network coding application the source node must know the entire file before sending the first packet. This is not desirable in several applications, e.g. a source node which is a sensor collecting data in some time interval, or streaming applications. Moreover, the authentication information to be sent along with the packets is quite long.

Therefore, we compare our constructions with the schemes in the standard model, and later in this section we will briefly discuss a comparison with the random oracle based ones.

In the scheme by Attrapadung and Libert [3] a signature consists of three group elements where the bilinear groups have composite order N, with N product of three primes. To compute a signature, the scheme needs to perform two multi-exponentiations and one exponentiation, whereas the verification time is dominated by the computation of four pairings in such composite order groups. Even if one applies standard techniques to convert the scheme in prime order groups (as suggested in [3]), the overhead would still remain significant.

In [13] Freeman proposes a general framework, that can be seen as a generalization of the Attrapadung and Libert methodology, for converting signature schemes with certain properties into linearly homomorphic ones. There are two appealing features in Freeman's work. First, his model allows for a stronger adversary than the one we consider. Second, the proposed approach is general enough to work with several currently known signature schemes. However, all the resulting (linearly homomorphic) signatures are less efficient than those given in this paper.

In the scheme by Catalano, Fiore and Warinschi [11] each signature consists of an element of $\mathbb{Z}_N^*$ and an integer s of $\lambda_s = 3k+|N|$ bits, where k is the security parameter and $|N|$ is the bit size of the RSA modulus N (which is related to k). Signing and verifying both need one multi-exponentiation (where all exponents have size λ, except one of size λ_s) and one exponentiation. Since in this scheme the linear combinations are done over the integers, it can support only a limited number of linear combinations, that in the network coding application translates to supporting only networks with paths of predetermined bounded length. Technically, the reason of such bound is that the vector coordinates cannot be let grow more than the size of the prime e.

In this scenario, our solution based on q-SDH seems the most efficient in terms of both bandwidth and computation. In fact, recall that in our case a signature is one group element plus one element of $\mathbb{Z}_p$: 512 bits in total, if one considers $k = 128$ bits of security and asymmetric pairings. The operations for signing and verifying are similar in all the schemes, but our SDH construction has the advantage that such operations can be performed over prime order groups. Our RSA realization, can be seen as a significant optimization of the Catalano-Fiore-Warinschi's scheme [11]. There are two main improvements. First, our scheme allows for a much smaller exponent s. In fact, in our case s can be of λ bits, that is even more than 10 times shorter than in [11], if one considers 128 bits of security. Intuitively, the reason of using a large s in [11] is that in the real scheme s is truly random, while in the simulation it is used to hide some information of $2k + |N|$ bits, which decreases its entropy down to k bits. So, there s is taken sufficiently large to keep it within negligible statistical distance from a uniform value of λ_s bits. In our case, s is in $\mathbb{Z}_e$, and we take advantage of modular reduction to obtain a uniformly distributed s also in the simulation. Notice that having such a short s saves in both bandwidth and computation. Second, our idea of computing all the linear combinations $\pmod e$ avoids the problem that the vector coordinates may grow beyond e. In this way we can support networks with paths of any lengths, which was not the case in the previous RSA-based schemes [11] and [14].

Finally, we consider the schemes in the random oracle model that work over similar algebraic settings, i.e., bilinear groups [7] and RSA [14]. Compared to them, our solutions are (not surprisingly) slightly worse. The main difference is the size of the public key that in our case is linear in $m+n$, whereas in [7,14] it is constant (because $O(m + m)$ group elements are generated on-the-fly using the random oracle). On the other hand, the size of a signature and the time needed to sign and verify are somewhat comparable. In this sense, we believe that our solutions offer a good compromise if one does not want to rely on the random oracle heuristic.

Acknowledgements. The work described in this paper has been supported in part by the European Commission through the ICT programme under contract ICT-2007-216676 ECRYPT II. The authors would like to thank Dennis Hofheinz and Eike Kiltz for helpful discussions in the early stage of this work.

References

1. Ahlswede, R., Cai, N., Li, S., Yeung, R.W.: Network information flow. IEEE Transactions on Information Theory 46(4), 1204–1216 (2000)
2. Ahn, J.H., Boneh, D., Camenisch, J., Hohenberger, S., Shelat, A., Waters, B.: Computing on authenticated data. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 1–20. Springer, Heidelberg (2012), http://eprint.iacr.org/2011/096
3. Attrapadung, N., Libert, B.: Homomorphic Network Coding Signatures in the Standard Model. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 17–34. Springer, Heidelberg (2011)
4. Barić, N., Pfitzmann, B.: Collision-Free Accumulators and Fail-Stop Signature Schemes without Trees. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 480–494. Springer, Heidelberg (1997)
5. Boneh, D., Boyen, X.: Short Signatures Without Random Oracles. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004)
6. Boneh, D., Boyen, X.: Short signatures without random oracles and the SDH assumption in bilinear groups. Journal of Cryptology 21(2), 149–177 (2008)
7. Boneh, D., Freeman, D., Katz, J., Waters, B.: Signing a Linear Subspace: Signature Schemes for Network Coding. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 68–87. Springer, Heidelberg (2009)
8. Boneh, D., Freeman, D.M.: Homomorphic Signatures for Polynomial Functions. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 149–168. Springer, Heidelberg (2011)
9. Boneh, D., Freeman, D.M.: Linearly Homomorphic Signatures over Binary Fields and New Tools for Lattice-Based Signatures. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 1–16. Springer, Heidelberg (2011)
10. Catalano, D., Fiore, D., Warinschi, B.: Efficient network coding signatures in the standard model. Cryptology ePrint Archive, http://eprint.iacr.org/2011/696
11. Catalano, D., Fiore, D., Warinschi, B.: Adaptive Pseudo-free Groups and Applications. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 207–223. Springer, Heidelberg (2011)
12. Chou, P.A., Wu, Y., Jain, K.: Practical network coding. In: 41st Allerton Conference on Communication, Control and Computing (2003)
13. Freeman, D.M.: Improved Security for Linearly Homomorphic Signatures: A Generic Framework. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 697–714. Springer, Heidelberg (2012), http://eprint.iacr.org/2012/060
14. Gennaro, R., Katz, J., Krawczyk, H., Rabin, T.: Secure Network Coding over the Integers. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 142–160. Springer, Heidelberg (2010)
15. Gkantsidis, C., Rodriguez, P.: Network coding for large scale content distribution. In: Proc. of IEEE INFOCOM 2005, pp. 2235–2245 (2005)
16. Ho, T., Koetter, R., Médard, M., Karger, D., Effros, M.: The benefit of coding over routing in a randomized setting. In: Proc. of International Symposium on Information Theory (ISIT), p. 442 (2003)
17. Ho, T., Leong, B., Koetter, R., Médard, M., Effros, M., Karger, D.: Byzantine modification detection in multicast networks using randomized network coding. In: Proc. of International Symposium on Information Theory (ISIT), pp. 144–152 (2004)

18. Ho, T., Médard, M., Koetter, R., Karger, D.R., Effros, M., Shi, J., Leong, B.: A random linear network coding approach to multicast. IEEE Transactions on Information Theory 52, 4413–4430 (2006)
19. Hofheinz, D., Kiltz, E.: Programmable Hash Functions and Their Applications. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 21–38. Springer, Heidelberg (2008)
20. Jaggi, S., Langberg, M., Katti, S., Ho, T., Katabi, D., Médard, M., Effros, M.: Resilient network coding in the presence of byzantine adversaries. IEEE Transactions on Information Theory 54, 2596–2603 (2008)
21. Johnson, R., Molnar, D., Song, D., Wagner, D.: Homomorphic Signature Schemes. In: Preneel, B. (ed.) CT-RSA 2002. LNCS, vol. 2271, pp. 244–262. Springer, Heidelberg (2002)
22. Krohn, M., Freedman, M., Mazieres, D.: On the-fly verification of rateless erasure codes for efficient content distribution. In: 2004 IEEE Symposium on Security and Privacy, Berkeley, California, USA, May 9–12, pp. 226–240. IEEE Computer Society Press (2004)
23. Robert-Li, S.-Y., Yeung, R.Y., Cai, N.: Linear network coding. IEEE Transactions on Information Theory 49(2), 371–381 (2003)
24. Waters, B.: Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)

Improved Security for Linearly Homomorphic Signatures: A Generic Framework

David Mandell Freeman*

Stanford University
dfreeman@cs.stanford.edu

Abstract. We propose a general framework that converts (ordinary) signature schemes having certain properties into linearly homomorphic signature schemes, i.e., schemes that allow authentication of linear functions on signed data. The security of the homomorphic scheme follows from the same computational assumption as is used to prove security of the underlying signature scheme. We show that the following signature schemes have the required properties and thus give rise to secure homomorphic signatures in the standard model:

- The scheme of Waters (Eurocrypt 2005), secure under the computational Diffie-Hellman asumption in bilinear groups.
- The scheme of Boneh and Boyen (Eurocrypt 2004, *J. Cryptology* 2008), secure under the q-strong Diffie-Hellman assumption in bilinear groups.
- The scheme of Gennaro, Halevi, and Rabin (Eurocrypt 1999), secure under the strong RSA assumption.
- The scheme of Hohenberger and Waters (Crypto 2009), secure under the RSA assumption.

Our systems not only allow weaker security assumptions than were previously available for homomorphic signatures in the standard model, but also are secure in a model that allows a stronger adversary than in other proposed schemes.

Our framework also leads to efficient linearly homomorphic signatures that are secure against our stronger adversary under weak assumptions (CDH or RSA) in the random oracle model; all previous proofs of security in the random oracle model break down completely when faced with our stronger adversary.

Keywords: Homomorphic signatures, standard model, bilinear groups, CDH, RSA.

1 Introduction

Suppose Alice has some set of data $m_1, \ldots, m_k$ that she signs with a digital signature and stores in a database. At some later point in time Bob queries the database for the

* This material is based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11-1-0382. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the Department of Defense or the U.S. Government.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 697–714, 2012.

mean $\overline{m}$ of the data. Since Bob suspects the database might be malicious, he also wants Alice's signature on $\overline{m}$ to prove that the mean was computed correctly. Bob's bandwidth is limited, so he can't simply download the whole database, verify the signature, and compute the mean himself. Or maybe he has the bandwidth but Alice has requested that the data be kept private, with only the mean to be made public. What is Bob to do?

Homomorphic signatures [19,5,14,7,2,6] are a cryptographic primitive that addresses this problem. In a homomorphic signature scheme, a user signs messages $m_1, \ldots, m_k$ in some message space $\mathcal{M}$, producing signatures $\sigma_1, \ldots, \sigma_k$; verification is performed as usual for a signature scheme. The "homomorphic" property is as follows: given this set of signatures and a function $f : \mathcal{M}^k \to \mathcal{M}$ from a set of "admissible" functions $\mathcal{F}$, anyone can produce a signature on the pair $(f, f(m_1, \ldots, m_k)) \in \mathcal{F} \times \mathcal{M}$. Validation of the signature asserts that the claimed value is indeed the result of applying f to the underlying data; if the system is secure, then a malicious adversary cannot compute a valid signature on (f, m^*) for any $m^* \neq f(m_1, \ldots, m_k)$.

Homomorphic signatures were originally proposed by Johnson, Molnar, Song, and Wagner [19] and were adapted for the above application by Boneh, Freeman, Katz, and Waters [5], whose motivation was to authenticate packets in *network coding* protocols [1,23]. Other applications of homomorphic signatures include computing statistics, Fourier transforms, or least-squares fits on authenticated data, all of which can be done using "linearly homomorphic" signatures; i.e., those that authenticate linear functions.

The construction of Boneh et al. uses bilinear groups and authenticates linear functions on vectors over large prime fields. Follow-up work by Gennaro, Katz, Krawczyk, and Rabin [14] is based on RSA and authenticates linear functions on vectors over the integers, while the system of Boneh and Freeman [7] is based on lattice assumptions and authenticates linear functions on vectors over small fields. In a recent breakthrough, Boneh and Freeman [6] showed how to use "ideal lattices" to authenticate *polynomial* functions on data; this system is currently the only one that goes beyond linear functions.

In all of the above systems security is proven only in the random oracle model. At present there are only two homomorphic signature schemes proven secure in the standard model. The first is a scheme of Attrapadung and Libert [2], which is based on the Lewko-Waters identity-based encryption scheme [22] and uses bilinear groups of composite order. Signatures consist of three group elements of size at least 1024 bits, and security is proven using three nonstandard (fixed-size) assumptions, two of which are decisional and one of which is computational. The second is a scheme of Catalano, Fiore, and Warinschi [8], which is based on the general framework of "adaptive pseudo-free groups." In the instantiation based on the strong RSA assumption, signatures consist of two integers of size at least 1024 bits.

1.1 Our Contributions

A General Framework for Homomorphic Signatures. Motivated by a desire to construct efficient systems with stronger security, we propose a general framework that converts (ordinary) signature schemes having certain properties into linearly homomorphic signature schemes. The security of the homomorphic scheme follows from the same computational assumption as is used to prove security of the underlying signature scheme. We show that the schemes of Waters [24], Boneh and Boyen [4]; Gennaro,

Halevi, and Rabin [13]; and Hohenberger and Waters [18] all have the required properties and thus give rise to secure homomorphic signatures. The resulting homomorphic constructions are all secure under a computational (as opposed to a decisional) assumption in the standard model, and the pairing-based constructions offer shorter signatures than those of [2] or [8]. Our framework also leads to a variant of the construction of Attrapadung and Libert, as the signature scheme derived from Lewko-Waters IBE has the required properties; the security proof, however, still requires decisional assumptions.

A Stronger Security Model. Not only do our systems allow weaker security assumptions than were previously available for homomorphic signatures, but our schemes are proven secure in a model that allows a stronger adversary than in other proposed schemes. Specifically, in all previous schemes the adversary could adaptively query signatures on many data sets but was required to submit all messages belonging to a given data set at the same time, after which he would receive signatures on all of the messages at once. In our security model the adversary is allowed to adaptively query one *message* at a time, and even to intersperse queries from different data sets. It was not previously known how to construct a homomorphic signature scheme that is secure against such an adversary.

We also observe that certain of our constructions are also secure in the random oracle model under weak assumptions: the Waters-based scheme (actually the same as that of Gentry and Silverberg [15]) under (co-)CDH in bilinear groups, and the Gennaro-Halevi-Rabin scheme under RSA. While these random-oracle schemes are less efficient than current homomorphic schemes that use the same assumptions [5,14], they are secure against our stronger adversary. All previous proofs of security in the random oracle model break down completely when faced with our stronger adversary.

It is possible to modify the proofs of the standard-model schemes of Attrapadung-Libert [2] and Catalano-Fiore-Warinschi [8] to work against our stronger adversary; in the full version of this paper [12] we address a variant of the former.

Many Schemes. Our framework gives users a wide range of options when choosing a homomorphic signature scheme, including variability of the underlying vector space (vectors over $\mathbb{F}_p$ for pairing-based systems, vectors over $\mathbb{Z}$ for RSA-based ones) and tradeoffs between security and efficiency (the most efficient systems require stronger assumptions). We also expect our framework to be applicable to other signature schemes, both existing and not yet proposed.

1.2 Overview of Our Construction

We consider *linearly homomorphic* signature schemes, in which messages are vectors $\mathbf{v}$ with coordinates in some ring R and functions are R-linear combinations of messages. Using network coding terminology, we call a set of vectors that can be linearly combined with each other a "file."

The impetus for our framework comes from comparing the Attrapadung-Libert homomorphic signatures [2] to the Lewko-Waters signatures on which they are based [22]. The Lewko-Waters system uses a cyclic group $\mathbb{G}$ whose order $N = pqr$ is a product of three distinct primes, along with a nondegenerate, symmetric bilinear map $\hat{e}$ on $\mathbb{G}$. A signature on a message m consists of two group elements $(\sigma_1, \sigma_2) = \left(g^r h^s,\ g^\alpha H(m)^r h^{s'}\right)$,

where g, h are public group elements of prime order p, q, respectively; g^α is the secret key; H is a hash function; and r, s, s' are random in $\mathbb{Z}_N$. Verification can be carried out by testing whether $\hat{e}(\sigma_2, g)/\hat{e}(\sigma_1, H(m))$ is equal to $e(g,g)^\alpha$, where this last value is also public. (Here g and h are constructed so that $\hat{e}(g,h) = 1$.)

Attrapadung and Libert convert this scheme to a homomorphic scheme that signs n-dimensional vectors defined over $\mathbb{Z}_N$. The main idea is that to sign a vector $\mathbf{v} = (v_1, \ldots, v_n)$ belonging to a file F, we use the underlying scheme to sign the filename F (or more precisely, a "tag" chosen at random to identify F) and then add on a signed "homomorphic hash" of the vector $\mathbf{v}$ using *the same randomness* on the g part. Specifically, the signature has the form

$$(\sigma_1, \sigma_2, \sigma_3) = \left(g^r h^s,\ g^\alpha H(F)^r h^{s'},\ (h_1^{v_1} \cdots h_n^{v_n})^r h^{s''}\right)$$

where $h_1, \ldots, h_n$ are additional public group elements in $\langle g \rangle$ and s'' is random. To verify, we check whether the first two components form a valid signature on F, and whether $\hat{e}(\sigma_1, \prod h_i^{v_i}) = \hat{e}(\sigma_3, g)$.

To make signatures on different vectors within a file compatible, we need to use the same randomness r in the underlying signature each time, so the σ_1 and σ_2 components are the same for each vector in the file. Attrapadung and Libert achieve this property by applying a pseudorandom function to the filename F to produce r. Once the randomness is the same across all vectors within a file, the homomorphic property follows: given two vectors $\mathbf{v}, \mathbf{w}$ in the same file F and two signatures $\sigma_{\mathbf{v}} = (\sigma_1, \sigma_2, \sigma_3)$ and $\sigma_{\mathbf{w}} = (\sigma_1, \sigma_2, \sigma_3')$ produced with the same value of r, the triple $(\sigma_1, \sigma_2, \sigma_3\sigma_3')$ is a valid signature on the vector $\mathbf{v} + \mathbf{w}$. Specifically, we have

$$\hat{e}(\sigma_1, \textstyle\prod h_i^{v_i+w_i}) = \hat{e}(\sigma_1, \prod h_i^{v_i}) \cdot \hat{e}(\sigma_1, \prod h_i^{w_i}) = \hat{e}(\sigma_3, g) \cdot \hat{e}(\sigma_3', g) = \hat{e}(\sigma_3\sigma_3', g).$$

This property generalizes in the obvious way to authenticate $\mathbb{Z}_N$-linear combinations of arbitrary numbers of vectors in $(\mathbb{Z}_N)^n$.

Pre-homomorphic Signatures. The idea of using a homomorphic hash to authenticate linear combinations of vectors goes back to Krohn, Freedman, and Mazières [21], and the idea of signing such a hash is used in several previous constructions [5,14,6]. The key idea here — and the one that we can generalize to other systems — is signing the filename and the hash separately and tying them together with the signing function.

Specifically, the abstract properties of the Lewko-Waters scheme that make the homomorphic scheme work are as follows:

- The signature contains a component $\sigma_1 = g^{f(m,r)}$ for some fixed group element g and some function f of the message m and randomness r. (In Lewko-Waters we take $f(m,r) = r$, modulo h components.)
- Given σ_1, m, and two group elements x and y, there is an efficient algorithm to test whether $y = x^{f(m,r)}$. (In Lewko-Waters we use the pairing.)

In Section 3 we formalize these properties in the notion of a *pre-homomorphic signature*.

Our main construction is as follows: given a pre-homomorphic signature, we form a homomorphic signature on a vector $\mathbf{v}$ in a file F by generating signing randomness r using a PRF, signing the tag τ identifying F to produce the component $\sigma_1 = g^{f(m,r)}$

(and perhaps some other component σ_2), and then forming the component $\sigma_3 = (\prod h_i^{v_i})^{f(m,r)}$. The signature on $\mathbf{v}$ is $(\sigma_1, \sigma_2, \sigma_3)$. As in the Attrapadung-Libert scheme, homomorphic operations within the same file can be carried out by multiplying σ_3 components, and verification can be carried out using the testing algorithm. As stated this system is "weakly" secure, and we must add some kind of "chameleon hash" to obtain full security; details are in Section 3.

Examples. Surveying the literature, we see that many pairing-based schemes have the "pre-homomorphic" structure we define. These include the CDH-based schemes of Gentry-Silverberg [15], Boneh-Boyen [3], and Waters [24], where signatures have the same general form as in the Lewko-Waters system, as well as that of Boneh-Boyen [4], where signatures have the form $g^{1/(x+m+yr)}$ and security is based on the q-strong Diffie-Hellman problem. In all cases we can use the pairing to determine whether two pairs of elements have the same discrete log relationship.

Expanding into the RSA space, we see that the signatures of Gennaro, Halevi, and Rabin [13] also have our "pre-homomorphic" form: signatures are of the form $g^{1/H(m)} \bmod N$, and we can easily test whether $y = x^{1/H(m)}$ by raising both sides to the power $H(m)$. GHR signatures are secure under the strong RSA assumption; Hohenberger and Waters [18] demonstrate a hash function H that allows for a proof of security of the same construction under the (standard) RSA assumption.

Security. As formalized by Boneh et al. [5] for network coding and adapted to the more general homomorphic setting by Boneh and Freeman [6], an attacker tries to break a homomorphic signature scheme by adaptively submitting signature queries to a challenger and outputting a forgery. The forgery is a tuple $(\tau^*, \mathbf{w}^*, \sigma^*, f^*)$ consisting of a "tag" τ^* that identifies a file, a vector $\mathbf{w}^*$, a signature σ^*, and a function f^*. There are two winning conditions: either τ^* does not identify one of the files queried to the challenger (a *Type 1 forgery*), or τ^* does identify such a file F, but $\mathbf{w}^*$ is not equal to $f(\mathbf{v}_1, \ldots, \mathbf{v}_k)$, where $\mathbf{v}_1, \ldots, \mathbf{v}_k$ are the vectors in F (a *Type 2 forgery*).

For our general construction, we give a direct reduction that shows that a Type 1 forgery leads to a break of the underlying signature scheme. Furthermore we show that if the underlying signature scheme is *strongly* unforgeable, then certain Type 2 forgeries also break the underlying scheme. We also observe that since the identifying tags are chosen by the challenger, the underlying scheme need only be unforgeable against a *weak* adversary, i.e., one that submits all of its message queries before receiving the public key. This relaxation allows for improved efficiency in our construction.

For the remaining Type 2 forgeries we do not have a black-box reduction to the underlying signature scheme. However, we can do the next best thing: we can abstract out properties of the scheme's security proof that allow us to use a forgery in the homomorphic system to solve the computational problem used to prove the underlying scheme secure. Specifically, suppose we have a simulator that takes an instance of a computational problem and mimics the underlying signature scheme. Let f be the "pre-homomorphic" signing function discussed above, and suppose that the simulator can produce two group elements x, y with the following properties:

- The simulator can compute $x^{f(m,r)}$ for all message queries.
- The simulator can compute $y^{f(m,r)}$ for all but one message query m^*.

– If r^* is the randomness used to sign m^*, then the value of $y^{f(m^*,r^*)}$ can be used to solve the computational problem.

A typical example of such a simulator is the kind used in security proofs of (strong-)RSA signatures [13,11,16,17,18]: if $\{e_i\}$ is the set of integers that need to be inverted mod $\varphi(N)$ to answer signature queries, we compute $E = \prod e_i$ and $E^* = \prod_{i \neq \ell} e_i$ for a random ℓ and set $x = g^E \bmod N$, $y = g^{E^*} \bmod N$. Using Shamir's trick, given y^{1/e_ℓ} we can recover g^{1/e_ℓ} and in many cases solve the computational problem.

Given such a simulator, we "program" the homomorphic hash function so that for all vectors queried by the adversary, $H_{\mathsf{hom}}(\mathbf{v})$ consists of x factors only and therefore all signatures can be computed. However, if the adversary produces a linear function f^* described by coefficients $(c_1, \ldots, c_k)$ and a vector $\mathbf{w}^*$ such that $\mathbf{w}^* \neq \sum c_i \mathbf{v}_i$, then we can show that with noticeable probability the hash of $\mathbf{w}^*$ has a nontrivial y factor, and therefore a forged signature can be used to solve the computational problem.

Our general security theorem appears in Section 5. An example instantiation, based on Boneh-Boyen signatures, appears in Section 6. In the full version of this paper [12] we show how to modify our schemes in bilinear groups to achieve privacy; specifically, a derived signature on $m' = f(m_1, \ldots, m_k)$ reveals nothing about the values of the m_i that cannot be obtained from the value of m' and the knowledge of f. (We also show that our RSA schemes do not have this property.)

1.3 Concurrent Work

In concurrent and independent work, Catalano, Fiore, and Warinschi [9] have proposed two new linearly homomorphic signature schemes that are secure in the standard model: one based on Boneh-Boyen signatures and secure under the q-SDH assumption, and one based on Gennaro-Halevi-Rabin signatures and secure under the strong-RSA assumption. Signatures in these schemes consist only of the σ_3 component of our corresponding schemes, and thus signatures are shorter than those arising from our construction. The strong-RSA construction also has the feature that the length of integer vectors to be signed is unbounded. (Our RSA constructions as well as that of [14] require an upper bound on vector length.)

While the constructions in [9] are proved secure only against an adversary that queries entire files at once, it is possible to modify the proofs to work against our stronger adversary. We also expect that if the hash function from [18] is used in the strong-RSA scheme, the resulting scheme is secure under the (standard) RSA assumption. However, it does not appear that the techniques of [9] can be used to produce linearly homomorphic signatures based on Waters signatures and the co-CDH assumption.

2 Homomorphic Signatures

In a homomorphic signature scheme we can sign messages m in some message space $\mathcal{M}$ and apply functions f to signed messages for f in some set of "admissible" functions $\mathcal{F}$. Each set of messages is grouped together into a "data set" or "file," and each file is equipped with a "tag" τ that serves to bind together the messages in that file. Formally, we have the following.

Definition 2.1 ([6]). A *homomorphic signature scheme* is a tuple of probabilistic, polynomial-time algorithms $(\mathsf{Setup}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Eval})$ as follows:

- $\mathsf{Setup}(1^\lambda, k)$. Takes a security parameter λ and a maximum data set size k. Outputs a public key pk and a secret key sk. The public key pk defines a message space $\mathcal{M}$, a signature space Σ, and a set $\mathcal{F}$ of "admissible" functions $f\colon \mathcal{M}^k \to \mathcal{M}$.
- $\mathsf{Sign}(\mathsf{sk}, \tau, m, i)$. Takes a secret key sk, a tag $\tau \in \{0,1\}^\lambda$, a message $m \in \mathcal{M}$ and an index $i \in \{1, \ldots, k\}$, and outputs a signature $\sigma \in \Sigma$. (The index i indicates that this is the ith message in the file.)
- $\mathsf{Verify}(\mathsf{pk}, \tau, m, \sigma, f)$. Takes a public key pk, a tag $\tau \in \{0,1\}^\lambda$, a message $m \in \mathcal{M}$, a signature $\sigma \in \Sigma$, and a function $f \in \mathcal{F}$, and outputs either 0 (reject) or 1 (accept).
- $\mathsf{Eval}(\mathsf{pk}, \tau, f, \vec{\sigma})$. Takes a public key pk, a tag $\tau \in \{0,1\}^\lambda$, a function $f \in \mathcal{F}$, and a tuple of signatures $\vec{\sigma} \in \Sigma^k$, and outputs a signature $\sigma' \in \Sigma$.

Let $\pi_i\colon \mathcal{M}^k \to \mathcal{M}$ be the function $\pi_i(m_1, \ldots, m_k) = m_i$ that projects onto the ith component. We require that $\pi_1, \ldots, \pi_k \in \mathcal{F}$ for all pk output by $\mathsf{Setup}(1^\lambda, k)$.

Informally, the correctness conditions of our scheme are that (a) a signature produced by Sign on message m with index i verifies for the projection function π_i, and (b) if Eval is given a function g and signatures that verify for messages m_i and functions f_i, then the signature output by Eval verifies for the message $g(\vec{m})$ and the function obtained by composing g with the f_i.

Formally, we require that for each $(\mathsf{pk}, \mathsf{sk})$ output by $\mathsf{Setup}(1^\lambda, k)$, we have:

1. Let $\tau \in \{0,1\}^\lambda$ be any tag, let $m \in \mathcal{M}$ be any message, and let $i \in \{1, \ldots, k\}$ be any index. If $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \tau, m, i)$, then $\mathsf{Verify}(\mathsf{pk}, \tau, m, \sigma, \pi_i) = 1$.
2. Let $\tau \in \{0,1\}^\lambda$ be any tag, let $\vec{\mu} = (\mu_1, \ldots, \mu_k) \in \mathcal{M}^k$ be any tuple of messages, let $\vec{\sigma} = (\sigma_1, \ldots, \sigma_k) \in \Sigma^k$ be signatures produced by zero or more iterative applications of Eval on the outputs of $\mathsf{Sign}(\mathsf{sk}, \tau, \mu_i, i)$, and let $(f_1, \ldots, f_k, g) \in \mathcal{F}^{k+1}$ be any tuple of admissible functions. Let $g \circ \vec{f}$ denote the function that sends $\vec{x} = (x_1, \ldots, x_k)$ to $g(f_1(\vec{x}), \ldots, f_k(\vec{x}))$. If $\mathsf{Verify}(\mathsf{pk}, \tau, m_i, f_i) = 1$ for some $m_1, \ldots, m_k \in \mathcal{M}$, the message $g(m_1, \ldots, m_k)$ is in $\mathcal{M}$, and the function $g \circ \vec{f}$ is admissible, then $\mathsf{Verify}\left(\mathsf{pk},\ \tau,\ g(\vec{m}),\ \mathsf{Eval}\big(\mathsf{pk}, \tau, g, \vec{\sigma}\big),\ g \circ \vec{f}\right) = 1$.

Note that if $f_i = \pi_i$ is the ith projection function, then the function $g \circ \vec{f}$ in condition (2) is equal to g. Thus condition (2) says that if we apply Eval to the function g and signatures $\sigma_i = \mathsf{Sign}(\mathsf{pk}, \tau, \mu_i, i)$ for $i = 1, \ldots, k$, then the resulting signature verifies for the message $g(\vec{\mu})$ and the function g.

A *linearly homomorphic* signature scheme is a homomorphic signature scheme where the message space $\mathcal{M}$ consists of n-dimensional vectors over a ring R, and the set of admissible functions $\mathcal{F}$ consists of R-linear functions from $(R^n)^k$ to R. We identify $\mathcal{F}$ with a subset of R^k by representing the function $f(\mathbf{v}_1, \ldots, \mathbf{v}_k) = \sum c_k \mathbf{v}_i$ as the vector $(c_1, \ldots, c_k) \in R^k$.

Relationship to Network Coding. Definition 2.1 generalizes the definition of Boneh, Freeman, Katz and Waters for signatures in *network coding* systems [5, Definition 1].

In network coding, a file is parsed as a set of vectors $\mathbf{v}'_1, \ldots, \mathbf{v}'_k \in \mathbb{F}_p^n$. Each vector $\mathbf{v}'_i$ is then "augmented" by appending the ith unit vector $\mathbf{e}_i$, creating k "augmented vectors" $\mathbf{v}_1, \ldots, \mathbf{v}_k \in \mathbb{F}_p^{n+k}$. It is these augmented vectors that are transmitted through the network.

In the network coding protocol, each router in the network creates random linear combinations of its incoming vectors and passes the resulting vectors downstream. The vectors' augmentation carries information about the function that has been applied. Specifically, the ith unit vector that we append to the ith data vector represents the projection function π_i. If we apply the linear function $f\colon (\mathbb{F}_p^{n+k})^k \to \mathbb{F}_p^{n+k}$ given by $f(x_1, \ldots, x_k) = \sum_i c_i x_i$, then the "augmentation component" of $\mathbf{w} = f(\mathbf{v}_1, \ldots, \mathbf{v}_k)$ (i.e., the last k entries) is exactly $(c_1, \ldots, c_k)$. Thus there are two equivalent ways of viewing a signature on a derived vector $\mathbf{w}$: as a signature on the entire vector $\mathbf{w}$, or as a signature on the pair $(\mathbf{w}', f)$ where $\mathbf{w}' = \sum_i c_i \mathbf{v}'_i$ is the first n components of $\mathbf{w}$. Our definition takes the latter view, as it is the one that generalizes more readily to nonlinear functions (see e.g. [6]).

Security. The goal of an adversary attacking a homomorphic signature scheme is to produce a signature on a message-function pair that cannot be derived from previously seen data and signatures. This can be done in two ways: the adversary can produce a signature on a function-message pair that doesn't correspond to a previously seen data set (a *Type 1 forgery*), or the adversary can authenticate an *incorrect* value of a function on a previously seen data set (a *Type 2 forgery*).

In our model, the adversary is allowed to make adaptive queries on data sets of his choice. Our adversary is allowed to query one message at a time and proceed adaptively *within* each data set, or even to intersperse queries from different data sets. In contrast, in previous works the adversary was required to submit all messages in a given data set at once. This new flexibility implies a third type of forgery: the adversary might output a function-message pair that corresponds to a previously seen data set, but for which the adversary has not queried enough messages for the function's output to be well-defined on the input data set. We call this forgery a *Type 3 forgery*.

In our model (and in our constructions) we must avoid collisions between tags τ, so we have the challenger choose them uniformly from $\{0,1\}^\lambda$. Since the adversary can intersperse queries from different files, the signer must maintain a state to ensure that each query is signed with the correct tag and index.

Definition 2.2 (Adapted from [6]). A homomorphic signature scheme $\mathcal{S} = (\mathsf{Setup}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Eval})$ is *unforgeable against an adaptive per-message attack* (or simply *unforgeable*) if for all k the advantage of any probabilistic, polynomial-time adversary $\mathcal{A}$ in the following game is negligible in the security parameter n:

Setup: The challenger runs $\mathsf{Setup}(1^\lambda, k)$ to obtain $(\mathsf{pk}, \mathsf{sk})$ and gives pk to $\mathcal{A}$. The public key defines a message space $\mathcal{M}$, a signature space Σ, and a set $\mathcal{F}$ of admissible functions $f\colon \mathcal{M}^k \to \mathcal{M}$.

Queries: $\mathcal{A}$ specifies a filename $F \in \{0,1\}^*$ and a message $\mathbf{v} \in \mathcal{M}$. If $\mathbf{v}$ is the first query for F, the challenger chooses a tag τ_F uniformly from $\{0,1\}^\lambda$, gives it to $\mathcal{A}$, and sets a counter $i_F = 1$. Otherwise, the challenger looks up the value of τ_F previously

chosen and increments the counter i_F by 1. The challenger then gives to $\mathcal{A}$ the signature $\sigma^{(F,i_F)} \leftarrow \mathsf{Sign}(\mathsf{sk}, \tau_F, \mathbf{v}, i_F)$.
The above interaction is repeated a polynomial number of times, subject to the restriction that at most k messages can be queried for any given filename F. We let $\mathbf{V}_F$ denote the tuple of elements $\mathbf{v}$ queried for filename F, listed in the order they were queried.

Output: $\mathcal{A}$ outputs a tag $\tau^* \in \{0,1\}^\lambda$, a message $\mathbf{w}^* \in \mathcal{M}$, a signature $\sigma^* \in \Sigma$, and a function $f^* \in \mathcal{F}$.

We say a function f is *well-defined on* F if either $i_F = k$ or $i_F < k$ and $f(\mathbf{V}_F, \mathbf{v}_{i_F+1}, \ldots, \mathbf{v}_k)$ takes the same value for all possible choices of $(\mathbf{v}_{i_F+1}, \ldots, \mathbf{v}_k) \in \mathcal{M}^{k-i_F}$. The adversary *wins* if $\mathsf{Verify}(\mathsf{pk}, \tau^*, \mathbf{w}^*, \sigma^*, f^*) = 1$ and one of the following hold:

(1) $\tau^* \neq \tau_F$ for all filenames F queried by $\mathcal{A}$ (a *Type 1 forgery*),
(2) $\tau^* = \tau_F$ for filename F, f^* is well-defined on F, and $\mathbf{w}^* \neq f^*(\mathbf{V}_F)$ (a *Type 2 forgery*), or
(3) $\tau^* = \tau_F$ for filename F and f^* is not well-defined on F (a *Type 3 forgery*).

The *advantage* $\mathrm{HomSig\text{-}Adv}[\mathcal{A}, \mathcal{S}]$ of $\mathcal{A}$ is the probability that $\mathcal{A}$ wins the game.

For $t \in \{1,2,3\}$, we say that the scheme is *secure against type t forgeries* if the winning condition in Definition 2.2 is restricted to type t forgeries only. The proof of the following result can be found in the full version of this paper [12].

Proposition 2.3. *Let $\mathcal{H}$ be a linearly homomorphic signature scheme with message space $\mathcal{M} \subset R^n$ for some ring R. If $\mathcal{H}$ is secure against Type 2 forgeries, then $\mathcal{H}$ is secure against Type 3 forgeries.*

Privacy. In addition to the unforgeability property described above, one may wish homomorphic signatures to be *private*, in the sense that a derived signature on $m' = f(m_1, \ldots, m_k)$ reveals nothing about the values of the m_i beyond what can be ascertained from the values of m' and f. We discuss this property in the full version of this paper [12].

3 Building Blocks

Pre-homomorphic Signatures. Our generic conversion applies to "hash-and-sign" signatures with a specific form. Namely, a signature on a message m with randomness r must have a component $g^{f(m,r)}$, where g is some fixed generator of a cyclic group $\mathbb{G}$ and f is some function that may depend on the secret key. Furthermore, if we are given a valid signature on m with randomness r, then given x and y there is an efficient algorithm that tests whether $y = x^{f(m,r)}$.

Definition 3.1. Let $\mathcal{S} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ be a signature scheme. Let $\mathcal{M}$ be the space of messages and $\mathcal{R}$ be the space of randomness sampled by the signing algorithm. We say that $\mathcal{S}$ is *pre-homomorphic* if the following three conditions hold for each key pair $(\mathsf{pk}, \mathsf{sk})$ output by KeyGen:

1. There is a finite cyclic group $\mathbb{G}$ such that Sign defines a map $\mathsf{Sign}_{\mathsf{sk}}\colon \mathcal{M} \times \mathcal{R} \to \mathbb{G} \times \{0,1\}^*$, where $\mathcal{M}$ is the message space and $\mathcal{R}$ is the space of randomness used by Sign. We decompose a signature σ as (σ_1, σ_2) with $\sigma_1 \in \mathbb{G}$, and we allow the σ_2 component to be empty.
2. The public key pk contains a generator g of the group $\mathbb{G}$ in (1), and there is an efficiently computable function $f_{\mathsf{sk}}\colon \mathcal{M} \times \mathcal{R} \to \mathbb{Z}$ such that for each signature $(\sigma_1, \sigma_2) \leftarrow \mathsf{Sign}_{\mathsf{sk}}(m, r)$, we have $\sigma_1 = g^{f_{\mathsf{sk}}(m,r)}$
3. There is an efficient algorithm $\mathsf{Test}(\mathsf{pk}, m, \sigma, x, y)$ that takes input the public key pk, a message $m \in \mathcal{M}$, a signature $\sigma = (\sigma_1, \sigma_2)$, and group elements $x, y \in \mathbb{G}'$ for some group $\mathbb{G}'$ of the same order as $\mathbb{G}$. Suppose $\mathsf{Verify}(\mathsf{pk}, m, \sigma) = 1$. Then the algorithm outputs 1 if and only if $\log_g(\sigma_1) = \log_x(y)$; otherwise, the algorithm outputs 0. (If $\mathsf{Verify}(\mathsf{pk}, m, \sigma) \neq 1$ then the algorithm's output is unspecified.)

Homomorphic Hashing. A *homomorphic hash* is a linear function that maps vectors defined over some ring R to elements of some finite group $\mathbb{G}$. The ring R is interpreted as "exponents" of the group $\mathbb{G}$; the following definition makes this concept precise.

Definition 3.2. Let $\mathbb{G}$ be a finite cyclic group, R be a ring, and $\phi\colon R \to \mathbb{Z}$ be an injective function. We say (R, ϕ) is a *ring of exponents for* $\mathbb{G}$ if $\phi(r) \bmod |\mathbb{G}|$ defines a ring homomorphism from R to $\mathbb{Z}_{|\mathbb{G}|}$.

We shall assume from now on that the map ϕ is understood, in which case we say R itself is a ring of exponents for $\mathbb{G}$ and we identify R with its image under ϕ. In particular, for $g \in \mathbb{G}$ and $r \in R$, we interpret g^r to mean $g^{\phi(r)}$.

While Definition 3.2 is abstract, it is very concrete in our two principal examples:

- If $\mathbb{G}$ is a cyclic group of order p and ϕ is the map that lifts elements of $\mathbb{F}_p$ to integer representatives in $[0, p-1]$, then $(\mathbb{F}_p, \phi)$ is a ring of exponents for $\mathbb{G}$.
- If $\mathbb{G}$ is *any* finite cyclic group and ϕ is the identity map on $\mathbb{Z}$, then $(\mathbb{Z}, \phi)$ is a ring of exponents for $\mathbb{G}$. (In our constructions $\mathbb{G}$ will be a cyclic subgroup of $\mathbb{Z}_N^*$.)

In both cases our interpretation of g^r for $r \in R$ agrees with standard usage.

We now define the homomorphic hash used in our conversion. Our definition incorporates, in a single abstract framework, the homomorphic hash from previous linearly homomorphic signatures using discrete log groups [21, 10, 5, 2] as well as the RSA-based construction of Gennaro et al. [14].

Definition 3.3. Let $\mathbb{G}$ be a finite cyclic group and let R be a ring of exponents for $\mathbb{G}$. For any positive integer n, define the following algorithms:

$\mathsf{HomHash.Setup}(\mathbb{G}, n)$: Choose random elements $h_1, \ldots, h_n \xleftarrow{\mathrm{R}} \mathbb{G}$ and output $\mathsf{hk} = (h_1, \ldots, h_n)$.

$\mathsf{HomHash.Eval}(\mathsf{hk}, \mathbf{v})$**:** Given a key $\mathsf{hk} = (h_1, \ldots, h_n)$ and a vector $\mathbf{v} = (v_1, \ldots, v_n) \in R^n$, output $\prod_{j=1}^n h_j^{v_j}$.

For a fixed value of hk, we define $H_{\mathsf{hom}}\colon R^n \to \mathbb{G}$ by $H_{\mathsf{hom}}(\mathbf{v}) = \mathsf{HomHash.Eval}(\mathsf{hk}, \mathbf{v})$.

As the name implies, the key property of HomHash is that it is homomorphic: for $\mathbf{v}, \mathbf{w} \in R^n$ and $a, b \in R$,

$$H_{\mathsf{hom}}(\mathbf{v})^a \cdot H_{\mathsf{hom}}(\mathbf{w})^b = \left(\prod_{j=1}^n h_j^{v_j}\right)^a \cdot \left(\prod_{j=1}^n h_j^{w_j}\right)^b = \prod_{j=1}^n h_j^{av_j+bw_j} = H_{\mathsf{hom}}(a\mathbf{v}+b\mathbf{w}).$$

(In the middle equality we have used the homomorphic property of Definition 3.2.)

Uniform Sampling. To sample uniformly random elements of $\mathbb{G}$, we raise a generator to a random exponent. The following definition captures the property this exponent needs to have.

Definition 3.4. Let $\mathbb{G}$ be a finite cyclic group and (R, ϕ) be a ring of exponents for $\mathbb{G}$. We say a distribution χ on R is $\mathbb{G}$-*uniform* if:

1. For $x \xleftarrow{\mathrm{R}} \chi$, the distribution of $g^{\phi(x)}$ is statistically closeto the uniform distribution on $\mathbb{G}$; and
2. If the order of $\mathbb{G}$ is not efficiently computable, then for $x \xleftarrow{\mathrm{R}} \chi$, the distribution of $\phi(x) \bmod e$ is statistically close to the uniform distribution on $\mathbb{Z}_e$ for all $e \in [|\mathbb{G}|/16, |\mathbb{G}|]$.

If $R = \mathbb{Z}_p$ and $\mathbb{G}$ is a group of (known) order p, we can take χ to be the uniform distribution on R. If $R = \mathbb{Z}$ and $\mathbb{G}$ is the multiplicative group of nonzero squares mod $N = pq$, we can take χ to be the uniform distribution on $[0, a]$ for any $a \gg |\mathbb{G}|$. To obtain a statistical distance of at most 2^{-m}, it suffices to take $a = N \cdot 2^m$.

Chameleon Hashing. As defined by Krawczyk and Rabin [20], a *chameleon hash function* is a function C that takes two inputs: a message m and randomness s. It is collision-resistant and has the additional property that there is a "trapdoor" that allows collisions to be computed.

To show unforgeability of our homomorphic signature scheme (as opposed to weak unforgeability) we will embed a "homomorphic" chameleon hash function C.Since the underlying messages are vectors, the randomness will be an additional vector component s, and we define $C(\mathbf{v}, s) = H_{\mathsf{hom}}(\mathbf{v}) \cdot u^s$ for some fixed (public) $u \in \mathbb{G}$. Note that (up to relabeling) this is simply H_{hom} applied to the $(n+1)$-dimensional vector $(\mathbf{v}, s)$.

Let us a try a first attempt at embedding a "trapdoor" in the homomorphic hash. We can generate hk and u such that we know discrete logs of the h_i and u to some base g; e.g., $h_i = g^{\beta_i}$, $u = g^{\eta}$. When $\mathbb{G}$ has prime order p, to evaluate C we can choose a uniformly random $s \in \mathbb{Z}_p$, and to hit a fixed target $C(\mathbf{v}, s) = g^a$ we simply compute s in $\mathbb{Z}_p$ such that $\langle \vec{\beta}, \mathbf{v} \rangle + \eta s = a$. Since this s is unique, the distribution of s conditioned on $(\mathbf{v}, C(\mathbf{v}, s) = g^a)$ is the same in both cases.

However, if $\mathbb{G}$ is a group of unknown order then this attempt fails. To begin, we cannot sample s from the uniform distribution on $\mathbb{Z}_{|\mathbb{G}|}$; in addition, we can't invert in the exponent to compute s. To get around these obstacles, we choose s from the distribution that the simulator in our security proof needs to sample (see Section 5) and we set $\eta = 1$. Specifically, the trapdoor information is $\beta_1, \ldots, \beta_n$ and $\delta_1, \ldots, \delta_k$ sampled from a $\mathbb{G}$-uniform distribution χ (Definition 3.4). To produce a signature on the ith file vector $\mathbf{v}$, the simulator uses the trapdoor to set $s = \delta_i + \langle \vec{\beta}, \mathbf{v} \rangle$. Thus in the "forward" direction we compute a random s by sampling δ_i and β_j from the same distribution χ.

More precisely, s is chosen from the following distribution:

Definition 3.5. Let χ be a $\mathbb{G}$-uniform distribution on R and $\mathbf{v} \in R^n$ be a vector. Let $F : \mathcal{K} \times \{0,1\}^\lambda \times \mathbb{Z} \to R$ be a pseudorandom function whose outputs are indistinguishable from samples from χ. For a fixed key $\mu \in \mathcal{K}$, define the distribution $\Xi_{\tau,\mathbf{v}}$ on R as follows:

1. Compute $\beta_j \leftarrow F_\mu(\tau, j)$ for $j = 1, \ldots, n$.
2. Sample $\delta \leftarrow \chi$.
3. Output $\delta + \langle \beta, \mathbf{v} \rangle$.

(The distribution $\Xi_{\tau,\mathbf{v}}$ depends on μ, but we suppress this in the notation for readability.)

Since our simulator only needs to evaluate the chameleon hash for one file, it does not need to reuse the values of δ, so we can choose a new uniform δ each time. Note that if R is finite and χ is the uniform distribution on R, then $\Xi_{\tau,\mathbf{v}}$ is the uniform distribution on R. In particular, the distribution does not depend on the PRF F, so we have recovered our "first attempt" above.

4 A Generic Conversion

Let $\mathcal{S} = (\mathcal{S}.\mathsf{KeyGen}, \mathcal{S}.\mathsf{Sign}, \mathcal{S}.\mathsf{Verify})$ be a pre-homomorphic signature scheme. Define a homomorphic signature scheme $\mathsf{HomSig}(\mathcal{S})$ as follows:

$\mathsf{HomSig}(\mathcal{S}).\mathsf{Setup}(1^\lambda, k, n)$: On input a security parameter λ, a maximum data set size k, and a dimension n, do the following:

1. Compute $(\mathsf{pk}_\mathcal{S}, \mathsf{sk}_\mathcal{S}) \leftarrow \mathcal{S}.\mathsf{KeyGen}(1^\lambda)$. Let $\mathbb{G}, \mathbb{G}'$ be the groups in Definition 3.1 and let R be a ring of exponents for $\mathbb{G}$.
   (In our instantitaions, we use $R = \mathbb{F}_p$ if $\mathbb{G}$ has order p, and $R = \mathbb{Z}$ if $\mathbb{G} \subset \mathbb{Z}_N^*$.)
2. If the order of $\mathbb{G}$ is efficiently computable from pk_S, set $B_1 = B_2 = |\mathbb{G}|$. Otherwise, choose B_1, B_2 such that $kB_1B_2 < |\mathbb{G}|/32$. (We assume that a lower bound on $|\mathbb{G}|$ can be efficiently computed.)
3. Compute $\mathsf{hk} \leftarrow \mathsf{HomHash.Setup}(\mathbb{G}', n)$.
4. Choose random $t_1, \ldots, t_k, u \xleftarrow{R} \mathbb{G}'$.
5. Choose a pseudorandom function $\Psi : \mathcal{K} \times \{0,1\}^\lambda \to \mathcal{R}$, where $\mathcal{R}$ is the space of randomness sampled by $\mathcal{S}.\mathsf{Sign}$, and choose a random key $\kappa \xleftarrow{R} \mathcal{K}$.[1]
6. Choose a pseudorandom function $F : \mathcal{K}' \times \{0,1\}^\lambda \times \mathbb{Z} \to R$, and choose a random key $\mu \xleftarrow{R} \mathcal{K}'$.
7. Output the public key $\mathsf{pk} = (\mathsf{pk}_\mathcal{S}, \mathsf{hk}, \{t_i\}_{i=1}^k, u, R, B_1, B_2)$ and the secret key $\mathsf{sk} = (\mathsf{sk}_\mathcal{S}, \Psi, \kappa, F, \mu, \mathsf{pk})$.

– The message space is $\mathcal{M} = \{\mathbf{v} \in R^n : \|\mathbf{v}\| \le B_1\}$, where we define $\|\mathbf{v}\| = \max_j\{|v_j|\}$. (Recall that we are identifying R with a subset of $\mathbb{Z}$ as remarked after Definition 3.2. If $|\mathbb{G}|$ is efficiently computable, then $\mathcal{M}$ is all of R^n.)

[1] If $\mathcal{S}.\mathsf{Sign}$ is deterministic, then we do not need the PRF Ψ.

- We represent an R-linear function $f : R^n \to R$ as a k-tuple of elements of R; specifically, the function $f(\mathbf{v}_1, \ldots, \mathbf{v}_k) = \sum c_i \mathbf{v}_i$ is represented by the vector $(c_1, \ldots, c_k) \in R^k$. We define $\|f\| = \max_i\{|c_i|\}$.
- The set of admissible functions $\mathcal{F}$ is all R-linear functions on k-tuples of vectors in R^n with $\|f\| \leq B_2$. (Note that when $R = \mathbb{Z}_{|\mathbb{G}|}$ this is all R-linear functions from $(R^n)^k$ to R.)
- We use $H_{\mathsf{hom}}(\mathbf{v})$ to denote $\mathsf{HomHash.Eval}(\mathsf{hk}, \mathbf{v})$.

$\mathsf{HomSig}(\mathcal{S}).\mathsf{Sign}(\mathsf{sk}, \tau, \mathbf{v}, i)$: On input a secret key sk, a tag $\tau \in \{0,1\}^\lambda$, a vector $\mathbf{v} \in R^n$, and an index $i \in \{1, \ldots, k\}$, do the following:

1. Compute $r \leftarrow \Psi_\kappa(\tau)$.
2. Compute $(\sigma_1, \sigma_2) \leftarrow \mathcal{S}.\mathsf{Sign}(\mathsf{sk}_\mathcal{S}, \tau, r)$.
3. Using the PRF F, choose $s \leftarrow \Xi_{\tau,\mathbf{v}}$ (Definition 3.5).
   If $|\mathbb{G}|$ is known, this is equivalent to choosing $s \xleftarrow{R} \mathbb{Z}_{|\mathbb{G}|}$.
4. Compute $\sigma_3 \leftarrow \left(t_i \cdot H_{\mathsf{hom}}(\mathbf{v}) \cdot u^s\right)^{f_{\mathsf{sk}}(\tau,r)}$, where f_{sk} is the function in Definition 3.1 (2).
5. Output $\sigma = (\sigma_1, \sigma_2, \sigma_3, s)$.

$\mathsf{HomSig}(\mathcal{S}).\mathsf{Verify}(\mathsf{pk}, \tau, \mathbf{w}, \sigma, f)$: On input a public key pk, a tag $\tau \in \{0,1\}^\lambda$, a vector $\mathbf{w} \in R^n$, a signature $\sigma = (\sigma_1, \sigma_2, \sigma_3, s)$, and a function $f = (c_1, \ldots, c_k)$, do the following:

1. Compute $\zeta_1 \leftarrow \mathcal{S}.\mathsf{Verify}(\mathsf{pk}_\mathcal{S}, \tau, (\sigma_1, \sigma_2))$.
2. Let $x \leftarrow \left(\prod_{i=1}^k t_i^{c_i}\right) \cdot H_{\mathsf{hom}}(\mathbf{w}) \cdot u^s$ and compute $\zeta_2 \leftarrow \mathsf{Test}\left(\mathsf{pk}_\mathcal{S}, \tau, (\sigma_1, \sigma_2), x, \sigma_3\right)$, where Test is the algorithm from Definition 3.1 (3).
3. If $\|\mathbf{w}\| \leq kB_1B_2$, set $\zeta_3 = 1$; otherwise set $\zeta_3 = 0$.
4. If $\zeta_1 = \zeta_2 = \zeta_3 = 1$, output 1; otherwise output 0.

$\mathsf{HomSig}(\mathcal{S}).\mathsf{Eval}(\mathsf{pk}, \tau, f, \vec{\sigma})$: On input a public key pk, a tag $\tau \in \{0,1\}^\lambda$, a function $f = (c_1, \ldots, c_k)$, and a vector of signatures $\vec{\sigma} = (\sigma^{(1)}, \ldots, \sigma^{(k)})$ where $\sigma^{(i)} = (\sigma_1^{(i)}, \sigma_2^{(i)}, \sigma_3^{(i)}, s^{(i)})$, do the following:

1. Compute $\sigma_3' \leftarrow \prod_{i=1}^k \left(\sigma_3^{(i)}\right)^{c_i}$, $s' \leftarrow \sum_{i=1}^k c_i s^{(i)}$.
2. Output $\sigma' = (\sigma_1^{(1)}, \sigma_2^{(1)}, \sigma_3', s')$.

Lemma 4.1 (Proof in full version [12].). *The homomorphic signature scheme* $\mathsf{HomSig}(\mathcal{S})$ *satisfies the correctness properties of Definition 2.1.*

5 Security

Recall that an adversary can break a homomorphic signature scheme by computing any of three types of forgeries in Definition 2.2. By Proposition 2.3, a Type 3 forgery in a linearly homomorphic scheme implies a Type 2 forgery. In our security analysis we consider the remaining two types separately. We further split Type 2 into two subtypes. In a Type 2 forgery for $\mathsf{HomSig}(\mathcal{S})$, the adversary outputs a forged signature $(\sigma_1^*, \sigma_2^*, \sigma_3^*, s^*)$ and a tag τ^* equal to one of the tags τ_ℓ returned from a previous query.

By our construction of Eval, any signature derived from the queried signatures corresponding to τ_ℓ will have the same σ_1 and σ_2 components as in the queried signatures. This motivates the following definition:

- *Type 2a*: The pair (σ_1^*, σ_2^*) output by the adversary is *not* equal to $(\sigma_1, \sigma_2) \leftarrow \mathcal{S}.\mathsf{Sign}(\mathsf{sk}_\mathcal{S}, \tau^*, r^*)$ computed by the challenger. (Here $r^* = \Psi_\kappa(\tau^*)$.)
- *Type 2b*: The pair (σ_1^*, σ_2^*) output by the adversary *is* equal to $(\sigma_1, \sigma_2) \leftarrow \mathcal{S}.\mathsf{Sign}(\mathsf{sk}_\mathcal{S}, \tau^*, r^*)$ computed by the challenger.

Type 1, 2a Forgeries. We show that Type 1 forgery in our homomorphic scheme $\mathsf{HomSig}(\mathcal{S})$ leads to a forgery of the underlying signature scheme $\mathcal{S}$; i.e., a valid signature on a previously unseen message. In addition, a Type 2a forgery leads to a *strong* forgery of the underlying signature scheme, i.e., a *new* valid signature on a previously queried message. Since the underlying scheme $\mathcal{S}$ is used to sign random messages chosen by the challenger, we only require that $\mathcal{S}$ be unforgeable against a weak adversary.

Theorem 5.1. *If $\mathcal{S}$ is strongly unforgeable against a weak adversary and Ψ is a secure PRF, then $\mathsf{HomSig}(\mathcal{S})$ is secure against Type 1 and Type 2a forgeries.*

Sketch of Proof. We simulate the public key for $\mathsf{HomSig}(\mathcal{S})$ using the public key for $\mathcal{S}$ and elements $t_i = g^{\gamma_i}$, $h_j = g^{\alpha_j}$, and $u = g^\delta$ with known discrete logarithms. To sign vector $\mathbf{v}_i$, we query τ to the $\mathcal{S}$ challenger to obtain (σ_1, σ_2), and we compute $\sigma_3 = \sigma_1^{\gamma_i + \langle \vec{\alpha}, \mathbf{v}_i \rangle + \delta s}$ for $s \xleftarrow{\mathrm{R}} \Xi_{\tau, \mathbf{v}_i}$. Given a Type 1 or Type 2a forgery $(\tau^*, \mathbf{w}^*, f^*, \sigma^*$, the (σ_1, σ_2) component of σ^* is a valid forgery for $\mathcal{S}$ on the message τ^*. □

Type 2b Forgeries. In this case we do not have a black-box reduction to the underlying signature scheme. However, we do not have to prove each instance separately, as we can abstract out properties of the underlying scheme's security proof — or more specifically, of the simulator used in the reduction — that allow our reduction to go through. These properties are captured in the following definition:

Definition 5.2. Let $\mathcal{S}$ be a pre-homomorphic signature scheme and $\mathcal{P}$ be a computational problem. We say that $\mathcal{S}$ is *δ-simulatable and γ-extractable for $\mathcal{P}$* if there is a simulator Sim that takes an instance I of $\mathcal{P}$, interacts with a signature adversary $\mathcal{A}$ that makes at most q message queries, and has the following properties:

1. The probability that Sim aborts is at most $1 - \delta$.
2. Conditioned on Sim not aborting, the public key $\mathsf{pk}_\mathcal{S}$ produced by Sim is statistically indistinguishable from a real public key for $\mathcal{S}$.
3. Conditioned on Sim not aborting and for any public key $\mathsf{pk}_\mathcal{S}$, the signatures produced by Sim are statistically indistinguishable from real signatures produced by $\mathcal{S}$.
4. Let $(\sigma_1^{(\ell)}, \sigma_2^{(\ell)})$ be the signature produced by Sim on the ℓth message query, and let $\omega_\ell = \log_g(\sigma_1^{(\ell)})$. (If Sim simulates signatures perfectly, then $\omega_\ell = f_{\mathsf{sk}}(m_\ell, r_\ell)$ for implicit randomness r_ℓ.) Then Sim can efficiently compute generators x and y of $\mathbb{G}'$ such that

- Sim can efficiently compute x^{ω_ℓ} for all ℓ;
- Sim can efficiently compute y^{ω_ℓ} for all $\ell \neq \ell^*$, where ℓ^* is a value in $1, \ldots, q$ randomly chosen by Sim.

5. For y and ℓ^* as above, there is an efficient algorithm Extract that given an integer b, a value $z = y^{b \cdot \omega_{\ell^*}}$, and the internal state of Sim, outputs either $\perp$ or a solution to the instance I of $\mathcal{P}$. Furthermore, if the distribution of b is $\mathbb{G}$-uniform, then the probability (over the instances of $\mathcal{P}$ and the random coins of Sim) that Extract outputs $\perp$ is at most $1 - \gamma$.

Theorem 5.3. *Suppose $\mathcal{S}$ is a δ-simulatable, γ-extractable pre-homomorphic signature scheme for $\delta, \gamma \geq 1/\operatorname{poly}(\lambda)$. If there is no efficient algorithm to solve problem $\mathcal{P}$ in the group $\mathbb{G}$ and Ψ and F are secure PRFs, then $\mathsf{HomSig}(\mathcal{S})$ is secure against Type 2b forgeries.*

Proof. We describe an algorithm $\mathcal{B}$ that takes an instance I of problem $\mathcal{P}$ and interacts with an adversary $\mathcal{A}$ in the unforgeability game for $\mathsf{HomSig}(\mathcal{S})$. $\mathcal{B}$ runs as follows:

Setup: $\mathcal{B}$ does the following:

1. Run Sim on instance I to generate a (simulated) public key $\mathsf{pk}_{\mathcal{S}}$ and elements $x, y \in \mathbb{G}'$ and $\ell^* \in \{1, \ldots, q\}$ as in Definition 5.2; abort if Sim aborts.
2. Let R, B_1, B_2 be as in $\mathsf{HomSig}(\mathcal{S})$.Setup and let χ be a $\mathbb{G}'$-uniform distribution on R.
3. For $j = 1, \ldots, n$, choose $\alpha_j, \beta_j \xleftarrow{R} \chi$ and set $h_j \leftarrow x^{\alpha_j} y^{-\beta_j}$. Let $\vec{\alpha}, \vec{\beta}$ be the vectors of α_j and β_j, respectively, and let $\mathsf{hk} = (h_1, \ldots, h_n)$.
4. For $i = 1, \ldots, k$, choose $\gamma_i, \delta_i \xleftarrow{R} \chi$ and set $t_i \leftarrow x^{\gamma_i} y^{-\delta_i}$. Let $\vec{\gamma}, \vec{\delta}$ be the vectors of γ_i, δ_i, respectively.
5. Choose $\eta \xleftarrow{R} \chi$ and set $u = x^\eta y$.
6. Choose random tags $\tau_1, \ldots, \tau_\ell \xleftarrow{R} \{0,1\}^\lambda$, and abort if $\tau_i = \tau_j$ for $i \neq j$. Initialize an empty array A and counters $c_\ell = 1$ for $\ell = 1, \ldots, q$.
7. Send $\mathcal{A}$ the public key $\mathsf{pk} = (\mathsf{pk}_{\mathcal{S}}, \mathsf{hk}, \{t_i\}_{i=1}^k, u, R, B_1, B_2)$.

Queries: When $\mathcal{A}$ makes a query for filename $F \in \{0,1\}^*$ and a vector $\mathbf{v} \in R^n$, $\mathcal{B}$ does the following:

1. If F is not in the array A, append F to A. Let ℓ be the index of F in A and let $i = c_\ell$. If $c_\ell = 1$, send the tag τ_ℓ to the adversary.
2. Run Sim to produce (simulated) $\mathcal{S}$ signatures $(\sigma_1^{(\ell)}, \sigma_2^{(\ell)})$ on the message τ_ℓ, using (perhaps implicit) randomness r_ℓ; abort if Sim aborts.
3. If $\ell \neq \ell^*$, choose $s^{(\ell,i)} \xleftarrow{R} \Xi_{\ell,\mathbf{v}}$ (Definition 3.5).
   If $\ell = \ell^*$, set $s^{(\ell,i)} = \delta_i + \langle \vec{\beta}, \mathbf{v} \rangle$
4. Compute the third component of $\mathsf{Sign}(\mathsf{sk}, \tau_\ell, \mathbf{v}, i)$ as

$$\sigma_3^{(\ell,i)} \leftarrow (t_i \cdot H_{\mathsf{hom}}(\mathbf{v}_i) \cdot u^s)^{\omega_\ell} = \left(x^{\gamma_i + \langle \vec{\alpha}, \mathbf{v}_i \rangle + \eta s^{(\ell,i)}} y^{s^{(\ell,i)} - \delta_i - \langle \vec{\beta}, \mathbf{v}_i \rangle} \right)^{\omega_\ell}$$

   Property 4 of Definition 5.2 implies that we can efficiently compute this value for all ℓ. (Note that when $\ell = \ell^*$, there is no y term due to our choice of s.)
5. Send the signature $\sigma^{(\ell,i)} = (\sigma_1^{(\ell)}, \sigma_2^{(\ell)}, \sigma_3^{(\ell,i)}, s^{(\ell,i)})$ to the adversary.
6. Set $c_\ell \leftarrow c_\ell + 1$.

Forgery: When $\mathcal{A}$ outputs a Type 2b forgery $(\tau^*, \mathbf{w}^*, \sigma^*, f^*)$ with f^* represented by $\mathbf{c} = (c_1, \ldots, c_k)$ and $\sigma^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*, s^*)$, $\mathcal{B}$ does the following:

1. If $\tau^* \neq \tau_{\ell^*}$, abort.
2. Let $\mathbf{v}_1, \ldots, \mathbf{v}_k$ be the vectors queried with tag τ^*. Compute

$$a = \langle \vec{\gamma}, \mathbf{c} \rangle + \langle \vec{\alpha}, \mathbf{w}^* \rangle + \eta s^*, \quad b = -\langle \vec{\delta}, \mathbf{c} \rangle - \langle \vec{\beta}, \mathbf{w}^* \rangle + s^*, \quad z = \sigma_3^* / x^{a \cdot \omega_{\ell^*}}.$$

   Property 4 of Definition 5.2 implies that we can efficiently compute z.
3. Run $\mathsf{Extract}(b, z, \mathsf{Sim})$ and output the result.

In the full version of this paper [12], we analyze the simulation using a series of games; here we give a sketch of the analysis.

Let W_0 be the event that $\mathcal{A}$ wins the unforgeability game when interacting with a real challenger for $\mathsf{HomSig}(\mathcal{S})$ and W_1 be the event that $\mathcal{A}$ wins when interacting with our simulator. Then we can show that under the hypotheses of the theorem statement, $\Pr[W_1] \geq \frac{\delta}{q} \cdot \Pr[W_0] - \epsilon$ for some negligible ϵ. (The δ factor represents the simulator not aborting, and the $1/q$ factor reflects the simulator guessing the correct ℓ^*.)

If W_1 occurs, then the fact that the forgery is a valid signature for the tag τ_{ℓ^*} implies that the element z computed in the forgery is equal to $y^{b \cdot \omega_{\ell^*}}$. Under the assumption that b is $\mathbb{G}'$-uniform, property (5) of Definition 5.2 implies that $\mathcal{B}$ outputs a solution to the instance I of problem $\mathcal{P}$ with probability at least γ, which completes the proof.

It remains only to show that b is $\mathbb{G}'$-uniform. Let $\mathbf{y} = \sum c_i \mathbf{v}_i - \mathbf{w}^* \in R^n$ and let $\hat{s} = \sum_{i=1}^{k} c_i s^{(\ell^*, i)}$. It follows from our construction of the $s^{(\ell^*, i)}$ that $b = \langle \vec{\beta}, \mathbf{y} \rangle + s^* - \hat{s}$. Since the property of being $\mathbb{G}'$-uniform is invariant under translation by a scalar, it suffices to show that (1) $\mathbf{y} \neq 0 \bmod |\mathbb{G}|$, and (2) the vector $\vec{\beta}$ comes from a distribution statistically close to χ^n even when conditioned on the adversary's view. Property (1) follows from the fact that $\mathcal{A}$ outputs a Type 2b forgery, while property (2) can be verified by looking at the information about $\vec{\alpha}, \vec{\beta}, \vec{\gamma}, \vec{\delta}$ available to the adversary. □

6 Example Instantiation: Boneh-Boyen Signatures

We now describe how our construction can be instantiated using the signatures of Boneh and Boyen [4]; we describe additional instantiations in the full version of this paper [12].

Let $\mathbb{G}, \mathbb{G}_T$ be group of prime order p with an efficiently computable, nondegenerate bilinear map $\hat{e} \colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. (For simplicity we assume here that the pairing is symmetric; in the full version we consider a general pairing $\hat{e} \colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.) The signature scheme BB consists of the following algorithms:

BB.Setup: Choose random $\alpha \xleftarrow{R} \mathbb{Z}_p$ and $g \xleftarrow{R} \mathbb{G}$. The public key is $\mathsf{pk} = (g, g^\alpha)$, and the secret key is $\mathsf{sk} = \alpha$.

BB.Sign: Given a message $m \in \mathbb{Z}_p$, output $\sigma = g^{1/(\alpha + m)}$.

BB.Verify: Output 1 if $\hat{e}(\sigma, g^m \cdot g^\alpha) = \hat{e}(g, g)$; otherwise output 0.

An instance of the *q-strong Diffie-Hellman problem* is a tuple $(g, g^\alpha, g^{\alpha^2}, \ldots, g^{\alpha^q})$ for randomly chosen $g \xleftarrow{R} \mathbb{G}$ and $\alpha \xleftarrow{R} \mathbb{Z}_p$. A solution is a pair $(r, g^{1/(\alpha + r)}) \in \mathbb{Z}_p \times \mathbb{G}$.

Boneh and Boyen [4, Lemma 9] show that if the q-SDH assumption holds for $\mathbb{G}$, then BB is strongly unforgeable against a weak adversary making at most q signature queries.

The BB scheme is pre-homomorphic (Definition 3.1): the (deterministic) signing function is $f_{\mathsf{sk}}(m) = 1/(\alpha + m) \pmod p$, and we define $\mathsf{BB.Test}(\mathsf{pk}, m, \sigma, x, y)$ to output 1 if and only if $\hat{e}(\sigma, x) = \hat{e}(g_1, y)$ (regardless of the output of $\mathsf{Verify}(\mathsf{pk}, m, \sigma)$).

We now describe the Boneh-Boyen simulator $\mathsf{Sim}_{\mathsf{BB}}$ that takes an instance $(g, g^{\alpha}, g^{\alpha^2}, \ldots, g^{\alpha^q})$ of the q-SDH problem and interacts with a weak signature adversary.

Setup: Given distinct messages $m_1, \ldots, m_q \in \mathbb{Z}_p$ queried by the adversary, form the polynomial $P(t) = \prod_{i=1}^{q}(t + m_i) \in \mathbb{Z}_p[t]$. Since $P(t)$ has degree at most q, we can use the q-SDH instance to compute $x = g^{P(\alpha)}$. Output the public key $\mathsf{pk} = (x, g^{\alpha})$; the (implicit) secret key is α.

Signatures: Let $P_\ell(t) = \prod_{i \neq \ell}(t + m_i)$. The signature on m_ℓ is $\sigma^{(\ell)} = x^{1/(\alpha+m_\ell)} = g^{P_\ell(\alpha)}$, which can be computed from the q-SDH challenge.

Proposition 6.1. $\mathsf{Sim}_{\mathsf{BB}}$ *is* 1*-simulatable &* $(1 - \frac{1}{p})$*-extractable for the* q*-SDH problem.*

Proof. Conditions 1–3 of Definition 5.2 are obviously satisfied. To verify condition 4, let $P(t)$ and $P_\ell(t)$ be as above. Let $x = g^{P(\alpha)}$ and $y = g^{P_{\ell^*}(\alpha)}$. We have $\sigma^{(\ell)} = x^{1/\alpha+m_\ell}$, so $\omega_\ell = 1/(\alpha + m_\ell)$ and the simulator can compute x^{ω_ℓ} for all ℓ and y^{ω_ℓ} for all $\ell \neq \ell^*$.

To verify the last condition, write $P_{\ell^*}(t)/(t + m_{\ell^*}) = Q(t) + c/(t + m_{\ell^*})$ for some polynomial $Q \in \mathbb{Z}_p[t]$ of degree at most $q - 2$ and $c \in \mathbb{Z}_p$. Since the messages m_i are distinct, we have $c \neq 0$. Given an integer b and the element

$$z = y^{b \cdot \omega_{\ell^*}} = g^{b \cdot P^*(\alpha)/(\alpha + m_{\ell^*})} = g^{b \cdot (Q(\alpha) + c/(\alpha + m_{\ell^*}))},$$

we let Extract output $\left(m_{\ell^*}, (z^{1/b}/g^{Q(\alpha)})^{1/c}\right)$, or $\perp$ if $b = 0 \bmod p$. Thus for uniform b in $\mathbb{Z}_p$, Extract outputs a solution to the q-SDH problem with probability $1 - 1/p$. □

Corollary 6.2. *If the* q*-SDH assumption holds for* $\mathbb{G}$*, then* $\mathsf{HomSig}(\mathsf{BB})$ *is unforgeable.*

Acknowledgments. The author thanks Nuttapong Attrapadung, Dan Boneh, and Benoît Libert for helpful discussions, and the anonymous referees for their feedback.

References

1. Ahlswede, R., Cai, N., Li, S., Yeung, R.: Network information flow. IEEE Transactions on Information Theory 46(4), 1204–1216 (2000)
2. Attrapadung, N., Libert, B.: Homomorphic Network Coding Signatures in the Standard Model. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 17–34. Springer, Heidelberg (2011)
3. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
4. Boneh, D., Boyen, X.: Short signatures without random oracles and the SDH assumption in bilinear groups. J. Cryptology 21, 149–177 (2008); extended abstract in Advances in Cryptology — EUROCRYPT 2004

5. Boneh, D., Freeman, D., Katz, J., Waters, B.: Signing a Linear Subspace: Signature Schemes for Network Coding. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 68–87. Springer, Heidelberg (2009)
6. Boneh, D., Freeman, D.M.: Homomorphic Signatures for Polynomial Functions. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 149–168. Springer, Heidelberg (2011), full version available at http://eprint.iacr.org/2011/018
7. Boneh, D., Freeman, D.M.: Linearly Homomorphic Signatures over Binary Fields and New Tools for Lattice-Based Signatures. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 1–16. Springer, Heidelberg (2011), Full version available at http://eprint.iacr.org/2010/453
8. Catalano, D., Fiore, D., Warinschi, B.: Adaptive Pseudo-free Groups and Applications. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 207–223. Springer, Heidelberg (2011)
9. Catalano, D., Fiore, D., Warinschi, B.: Efficient Network Coding Signatures in the Standard Model. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 680–696. Springer, Heidelberg (2012), http://eprint.iacr.org/2011/696
10. Charles, D., Jain, K., Lauter, K.: Signatures for network coding. International Journal of Information and Coding Theory 1(1), 3–14 (2009)
11. Fischlin, M.: The Cramer-Shoup Strong-RSA Signature Scheme Revisited. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 116–129. Springer, Heidelberg (2002)
12. Freeman, D.M.: Improved security for linearly homomorphic signatures: A generic framework. Cryptology ePrint Archive, Report 2012/060 (2012), http://eprint.iacr.org/2012/060
13. Gennaro, R., Halevi, S., Rabin, T.: Secure Hash-and-Sign Signatures without the Random Oracle. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 123–139. Springer, Heidelberg (1999)
14. Gennaro, R., Katz, J., Krawczyk, H., Rabin, T.: Secure Network Coding over the Integers. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 142–160. Springer, Heidelberg (2010)
15. Gentry, C., Silverberg, A.: Hierarchical ID-Based Cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002)
16. Hofheinz, D., Kiltz, E.: Programmable Hash Functions and Their Applications. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 21–38. Springer, Heidelberg (2008)
17. Hohenberger, S., Waters, B.: Realizing Hash-and-Sign Signatures under Standard Assumptions. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 333–350. Springer, Heidelberg (2009)
18. Hohenberger, S., Waters, B.: Short and Stateless Signatures from the RSA Assumption. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 654–670. Springer, Heidelberg (2009)
19. Johnson, R., Molnar, D., Song, D., Wagner, D.: Homomorphic Signature Schemes. In: Preneel, B. (ed.) CT-RSA 2002. LNCS, vol. 2271, pp. 244–262. Springer, Heidelberg (2002)
20. Krawczyk, H., Rabin, T.: Chameleon signatures. In: Network and Distributed System Security Symposium (2000)
21. Krohn, M., Freedman, M., Mazières, D.: On-the-fly verification of rateless erasure codes for efficient content distribution. In: Proc. of IEEE Symposium on Security and Privacy, pp. 226–240 (2004)
22. Lewko, A.B., Waters, B.: New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (2010)
23. Li, S.-Y.R., Yeung, R.W., Cai, N.: Linear network coding. IEEE Trans. Info. Theory 49(2), 371–381 (2003)
24. Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 320–329. Springer, Heidelberg (2005)

On the Security of Dynamic Group Signatures: Preventing Signature Hijacking

Yusuke Sakai[1,⋆], Jacob C.N. Schuldt[2,⋆⋆], Keita Emura[3,⋆⋆⋆], Goichiro Hanaoka[2], and Kazuo Ohta[1]

[1] The University of Electro-Communications, Japan
yusuke.sakai@uec.ac.jp
[2] National Institute of Advanced Industrial Science and Technology, Japan
[3] National Institute of Information and Communications Technology, Japan

Abstract. We identify a potential weakness in the standard security model for dynamic group signatures which appears to have been overlooked previously. More specifically, we highlight that even if a scheme provably meets the security requirements of the model, a malicious group member can potentially claim ownership of a group signature produced by an honest group member by forging a proof of ownership. This property leads to a number of vulnerabilities in scenarios in which dynamic group signatures are likely to be used. We furthermore show that the currently most efficient dynamic group signature scheme does not provide protection against this type of malicious behavior.

To address this, we introduce the notion of *opening soundness* for group signatures which essentially requires that it is infeasible to produce a proof of ownership of a valid group signature for any user except the original signer. We then show a relatively simple modification of the scheme by Groth (ASIACRYPT 2007, full version) which allows us to prove opening soundness for the modified scheme without introducing any additional assumptions.

We believe that opening soundness is an important and natural security requirement for group signatures, and hope that future schemes will adopt this type of security.

1 Introduction

Group signatures, introduced by Chaum and van Heyst [11], allow a group member to anonymously sign a message on behalf of the group. More specifically, anyone will be able to verify that a signature originates from a group member, but the signature does not reveal the identity of the signer, not even to other members of the group. Group membership is controlled by an authority called the issuer, who handles enrollment of users through an interactive join protocol.

⋆ The first author is supported by a JSPS Fellowship for Young Scientists.
⋆⋆ The second author is supported by a JSPS Fellowship for Young Scientists.
⋆⋆⋆ This work was done when the third author was a postdoctoral researcher at Center for Highly Dependable Embedded Systems Technology, Japan Advanced Institute of Science and Technology (JAIST).

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 715–732, 2012.

To prevent misuse of the signing capabilities obtained by group members, another authority called the opener can revoke the anonymity of a signature and identify the signer of the message.

Following the introduction of group signatures, a series of different security requirements were proposed for this primitive, each of which aims at addressing a specific security concern by augmenting or refining previous notions, e.g. unforgeability, exculpability, traceability, coalition resistance, framing resistance, anonymity and unlinkability. These security notions were later consolidated in the security model proposed by Bellare, Micciancio, and Warinschi [2] who introduce two strong security requirements, full-anonymity and full-traceability, which imply all of the previously proposed notions of security.

However, a drawback of the model by Bellare, Micciancio, and Warinschi [2] is that only *static* group signature schemes are considered i.e. the set of group members is fixed, and the private key material of each group member is generated in the setup phase of the scheme. Furthermore, the authority controlling the group (which acts as both the issuer and opener) is considered to be fully trusted. To address this, Bellare, Shi, and Zhang [3] extended the model of [2] to capture *dynamic* group signature schemes in which a user can dynamically join the group by engaging in a join protocol with the issuer. Furthermore, to reduce trust in the opener, the model adopts the approach by Camenisch and Michels [10], and requires that the opener produces a non-interactive and publicly verifiable proof that a given signature was produced by a given signer. The model introduces three formal security notions: anonymity, traceability, and non-frameability. The former two notions are adaptations of the full-anonymity and full-traceability notions to the dynamic group signature setting. The latter notion, non-frameability, requires that even if a malicious opener and issuer collude, they cannot frame an honest user by producing a signature and corresponding opening which identify the honest user as the signer, when the honest user did not produce the signature in question.

Limitations of Non-frameability. While non-frameability is a strong security notion, it only partly covers the security properties one would intuitively expect to gain when the opener is required to produce a non-interactive and publicly verifiable proof of an opening. More specifically, the non-frameability notion only ensures that the opener cannot frame an *uncorrupted* user by constructing a proof that the user is the signer of a signature he did not produce. However, no guarantee is given regarding an opening involving a *corrupted* user. This leaves open the possibility that an opening showing that a malicious or corrupted user is the signer of a signature produced by an honest user, can be constructed. Furthermore, this might not require the opener to be corrupted or malicious, in which case a malicious user might be able to independently forge a proof showing that he is the signer of any signature of his choice.

Depending on the concrete scenario in which a dynamic group signature scheme is used, the ability to forge an opening proof might become a real security concern. We highlight several potential threats that this ability gives rise to:

- ***Signer impersonation.*** The most obvious threat is signer impersonation. This is a problem if a group signature scheme is used for an anonymous auction as suggested in [1]. In this scenario, the bidders correspond to group members, and when submitting a bid, a group member will attach a group signature on his bid. The opener serves as the auctioneer, and will make the opening of the signature on the highest bid public. This will enable anyone to verify who the winner of the auction is. However, a malicious bidder may forge a proof of ownership of the signature on the highest bid and may insist that he/she is the winner.
  A similar situation occurs if a dynamic group signature scheme is used to implement an authentication scheme with identity escrow [23]. In this case, a malicious group member can claim to be the user who authenticated himself to a server (and provide a proof thereof) when this is not the case.
- ***Proxy confession.*** The ability to open a group signature is introduced to keep the group members accountable of the messages signed on behalf of the group. However, assume that a signature on some message causes a dispute, but the real signer wants to avoid being blamed for this. Then the real signer asks (or intimidates) another group member to forge a proof of ownership of the signature and take the blame.
- ***Key exposure.*** Consider the case in which a group member's private key is exposed and falls into the hands of a malicious user. This will not only allow the malicious user to construct future signatures on any message of this choice, but will furthermore allow him to claim (and prove) that the original user is the signer of any *previously generated* signature.

Our Contribution. We highlight the above described potential weakness of the security guarantee provided by the formal model of Bellare, Shi, and Zhang [3]. Furthermore, we show that this is not only a property of the security model, but that the most efficient dynamic group signature schemes enable a malicious group member to forge a proof of ownership of a signature.

To address this, we propose a new security notion for dynamic group signatures which we denote *opening soundness*. We consider two variants of this notion, weak opening soundness and (ordinary) opening soundness. The former is intended to address the above highlighted security threats in an intuitive and straightforward manner, and will rule out the possibility that a malicious group member can produce a proof of ownership of a signature generated by an honest user. The latter considers a stronger adversary who has access to the private key of the opener, and who is only required to produce two different openings of a maliciously constructed signature. The notion of opening soundness implies the notion of weak opening soundness.

As a positive result, we prove that the generic construction of a dynamic group signature scheme by Bellare, Shi, and Zhang [3] achieves opening soundness. We furthermore propose a modification of the scheme by Groth [19] which allows us to prove opening soundness of the modified scheme. In contrast, we show that

the original scheme does not provide weak opening soundness. In addition, we briefly discuss opening soundness of the random oracle scheme [14,4]. A summary of our results regarding opening soundness of the above mentioned schemes can be seen in Table 1.

Table 1. Summary of the results. The mark "?" means it is an open question whether the scheme has the given property or not. The rightmost column denotes the section in which the security of the corresponding scheme is discussed.

	Opening Soundness	Weak Opening Soundness	
Our Variant of [19]	Yes	Yes	(§5.1)
Bellare-Shi-Zhang [3]	Yes	Yes	(§4)
Furukawa-Imai [14]	No	?	(§4)
Bichsel et al. [4]	No	?	(§4)
Groth (full version) [19]	No	No	(§4)

Related Work. Since the first proposal of group signature by Chaum and van Heyst, many efficient constructions have been proposed, most of which are relying on the random oracle model [1,6,9,22,14,12,4]. Many initial schemes were based on the strong-RSA assumption. The first group signature schemes based on assumptions of the discrete-logarithm type were achieved independently by Camenisch and Lysyanskaya [9], and Boneh, Boyen, and Shacham [6]. The former scheme is based on the LRSW assumption, while the latter is based on the q-strong Diffie-Hellman assumption. Kiayias, Tsiounis, and Yung proposed the notion of traceable signature [21], which can be seen as an extension of group signature with additional anonymity-revocation functionalities. One of these functionalities is that of allowing a group member to claim the authorship of a signature, however, its security requirement does not care about the possibility in which a malicious member falsely claims the authorship of an honestly generated signature by another.

Constructions which are provably secure without random oracles were only recently achieved. Besides the generic construction relying on NIZK proofs for general NP languages, Groth constructed the first concrete group signature scheme with constant signature size by exploiting the properties of bilinear groups [17], though signatures are extremely large. Boyen and Waters proposed group signature schemes [7,8] whose signature sizes are quite compact. In particular the latter scheme has signatures consisting only of six group elements of a composite order group. The drawback of these schemes is that they only achieve weaker security guarantees, that is, they only provide so called CPA-anonymity in the security model of Bellare, Micciancio, and Warinschi [2]. Groth proposed another group signature scheme [18,19] which has constant signature size (roughly one or two kilobytes) and which is provably secure in the dynamic group signature model of Bellare, Shi, and Zhang [3] without relying on random oracles.

2 Preliminaries

2.1 Group Signatures

In this section, we briefly review the model and the security notions of group signatures, presented by Bellare, Shi, and Zhang [3]. A group signature scheme consists of the following seven algorithms:

GKg: This is a group key generation algorithm which, on input 1^k, returns the keys (gpk, ik, ok), where gpk is a group public key, ik is an issuing key, and ok is an opening key.

UKg: This is a user key generation algorithm which, on input gpk, returns a personal public and private key pair (upk, usk). Each user i will generate a personal key pair (upk_i, usk_i) before engaging in the joining protocol which is described below.

Join/Issue: This is a pair of interactive algorithms which implement the joining protocol run by a user i and the issuer. The algorithm Join, which is run by the user, takes (gpk, upk, usk) as input, whereas Issue, which is run by the issuer, takes (gpk, upk, ik) as input. Upon successful completion of the protocol, Join outputs a private signing key gsk_i for user i, and Issue outputs the registration information of user i which is stored in $reg[i]$, where reg is a registration table maintained by the issuer.

GSig: This is the group signing algorithm run by a user i, which, on input gpk, a signing key gsk_i, and a message m, returns a group signature Σ.

GVf: This is the group signature verification algorithm which, on input (gpk, m, Σ), returns 1 to indicate that Σ is a valid signature on m, or 0 otherwise.

Open: This is the opening algorithm run by the opener, which, on input $(gpk, ok, reg, m, \Sigma)$, returns (i, τ), where i specifies that the originator of the signature Σ is the user i, and τ is a non-interactive proof of this. In case the algorithm fails to identify the originator of the signature, it outputs $i = 0$. Note that Open requires access to the registration table reg.

Judge: This is the judge algorithm which, on input $(gpk, i, upk_i, m, \Sigma, \tau)$, outputs either 1 or 0 indicating that the proof τ is accepted as valid or invalid, respectively.

The model in [3] introduces four requirements for a group signature scheme, namely, correctness, anonymity, non-frameability, and traceability. The correctness notion requires that honestly generated signatures will be accepted as valid by the verification algorithm, can be opened by the opening algorithm, and that the judging algorithm will accept the resulting proof as valid. The anonymity notion requires that no information about the identity of a signer is leaked from a group signature, even if the signing keys of all group members and the issuer are exposed. The non-frameability notion requires that no adversary corrupting both the opener and the issuer, can produce a signature and an opening proof that identify an uncorrupted group member as the signer, when the uncorrupted group member did not produce the signature in question. The traceability notion requires that an adversary corrupting the opener and controlling a group

of malicious group members, cannot produce a valid signature that cannot be opened correctly.

The formal definitions of the four notions are given as follows. We first define several oracles needed for security notions:

$\mathsf{AddU}(i)$**:** The add-user oracle runs $\mathsf{UKg}(gpk)$ and $\mathsf{Join}/\mathsf{Issue}$ protocol to add an honest user. It returns the user public key upk of the user. The oracle add i to the set HU.

$\mathsf{RReg}(i)$**:** The read-registration-table oracle reveals the content of the registration table $reg[i]$.

$\mathsf{SndToU}(i, M)$ The send-to-user oracle at first sets up a user public/secret key pair by $(upk_i, usk_i) \leftarrow \mathsf{UKg}(gpk)$ and add i to the set HU. The oracle then allows the adversary to engage a group-joining protocol of the user i on the behalf of the corrupted issuer. The message M is sent to the user i who follows the protocol $\mathsf{Join}(gpk, upk_i, usk_i)$. The response of the user is returned to the adversary.

$\mathsf{WReg}(i, M)$ The write-registration-table oracle updates $reg[i]$ to M.

$\mathsf{USK}(i)$**:** The user-secret-keys oracle reveals the secret keys (usk_i, gsk_i) of the user i to the adversary.

$\mathsf{CrptU}(i, M)$**:** The corrupt-user oracle sets the user public key of the user i to M and add i to the set CU.

$\mathsf{Open}(m, \Sigma)$**:** The open oracle returns the opening $(i, \tau) \leftarrow \mathsf{Open}(gpk, ok, m, \Sigma)$ of the signature Σ under the message m.

$\mathsf{Ch}_b(m, i_0, i_1)$**:** The challenge oracle returns a challenge $\Sigma^* \leftarrow \mathsf{GSig}(gpk, gsk_{i_b}, m)$. The users i_0 and i_1 needs to be in the set HU.

$\mathsf{GSig}(i, m)$**:** The signing oracle returns a signature $\Sigma \leftarrow \mathsf{GSig}(gpk, gsk_i, m)$ on the message m of the user i, who needs to be in the set HU.

$\mathsf{SndToI}(i, M)$**:** The send-to-issuer oracle allows the adversary to engage a group-joining protocol on behalf of the corrupted user i. The message M is sent to the issuer who follows the protocol $\mathsf{Issue}(gpk, upk_i, ik)$. The response of the issuer is returned to the adversary. The user i needs to be in the set CU.

The correctness and security requirements for a group signature scheme are as follows:

Definition 1. *A group signature scheme is said to have* correctness *if*

$$\Pr[(gpk, ik, ok) \leftarrow \mathsf{GKg}(1^k); (i, m) \leftarrow \mathcal{A}^{\mathsf{AddU},\mathsf{RReg}}(gpk); \\ \Sigma \leftarrow \mathsf{GSig}(gpk, gsk_i, m); (j, \tau) \leftarrow \mathsf{Open}(gpk, ok, reg, m, \Sigma) \\ : \mathsf{GVf}(gpk, m, \Sigma) = 0 \vee i \neq j \vee \mathsf{Judge}(gpk, i, upk_i, m, \Sigma, \tau) = 0]$$

is negligible for any probabilistic polynomial-time adversary $\mathcal{A}$*.*

Definition 2. *A group signature scheme is said to have* anonymity *if*

$$\Pr[b \leftarrow \{0,1\}; (gpk, ik, ok) \leftarrow \mathsf{GKg}(1^k); \\ b' \leftarrow \mathcal{A}^{\mathsf{SndToU},\mathsf{WReg},\mathsf{USK},\mathsf{CrptU},\mathsf{Open},\mathsf{Ch}_b}(gpk, ik) : b = b'] - \frac{1}{2}$$

is negligible for any probabilistic polynomial-time adversary $\mathcal{A}$*.*

Definition 3. *A group signature scheme is said to have* non-frameability *if*

$$\begin{aligned}\Pr[(gpk, ik, ok) \leftarrow \mathsf{GKg}(1^k);\\ (m, \Sigma, i, \tau) \leftarrow \mathcal{A}^{\mathsf{SndToU},\mathsf{WReg},\mathsf{USK},\mathsf{CrptU},\mathsf{GSig}}(gpk, ok, ik);\\ : \mathsf{GVf}(gpk, m, \Sigma) = 1 \wedge \mathsf{Judge}(gpk, i, upk_i, m, \Sigma, \tau) = 1\\ \wedge \mathcal{A} \textit{ queried neither } \mathsf{USK}(i) \textit{ nor } \mathsf{GSig}(i, m)]\end{aligned}$$

is negligible for any probabilistic polynomial-time adversary $\mathcal{A}$.

Definition 4. *A group signature scheme is said to have* traceability *if*

$$\begin{aligned}\Pr[(gpk, ik, ok) \leftarrow \mathsf{GKg}(1^k); (m, \Sigma) \leftarrow \mathcal{A}^{\mathsf{CrptU},\mathsf{SndToI},\mathsf{AddU},\mathsf{USK},\mathsf{RReg}}(gpk, ok);\\ (i, \tau) \leftarrow \mathsf{Open}(gpk, ok, reg, m, \Sigma)\\ : \mathsf{GVf}(gpk, m, \Sigma) = 1 \wedge (i = 0 \vee \mathsf{Judge}(gpk, i, upk_i, m, \Sigma, \tau) = 0)]\end{aligned}$$

is negligible for any probabilistic polynomial-time adversary $\mathcal{A}$.

2.2 Other Primitives

Public-Key Encryption. A public key encryption scheme consists of three algorithms $(\mathsf{EKg}, \mathsf{Enc}, \mathsf{Dec})$, which satisfy the following correctness condition: For any security parameter $\ell \in \mathbb{N}$, any plaintext $m \in \{0,1\}^*$, any random tape r for EKg, and any random tape s for Enc, the condition $\mathsf{Dec}(dk, \mathsf{Enc}(pk, m; s)) = m$ holds, where pk and dk are output from EKg as $(pk, dk) \leftarrow \mathsf{EKg}(1^\ell; r)$. In this paper we require a public key encryption scheme to satisfy the security notion of indistinguishability under chosen-ciphertext attack (IND-CCA) [25].

Digital Signature. A digital signature scheme consists of three algorithms $(\mathsf{SKg}, \mathsf{Sign}, \mathsf{Ver})$, which satisfy the following correctness condition: For any security parameter $\ell \in \mathbb{N}$, any message $m \in \{0,1\}^*$, any random tape r for SKg, and any random tape s for Sign, the condition $\mathsf{Ver}(vk, m, \mathsf{Sign}(sk, m; s)) = \top$ holds, where vk and sk are output from SKg as $(vk, sk) \leftarrow \mathsf{SKg}(1^\ell; r)$. In this paper we use two types of security for digital signature schemes. One is the standard security notion of unforgeability under adaptive chosen message attack (EUF-CMA), and the other is strong one-time signatures. See [16] for exact definitions.

Target Collision-Resistant Hash Functions. A family of functions is called target collision-resistant if no algorithms, which firstly chooses an input and then is given a description of a function in the family, can find another input that produces the same output to the first input. The formal definition we need is as follows: A function generator $\mathrm{HashGen}(1^\ell)$ takes as input a security parameter and outputs a function $\mathcal{H}$. The family of functions is said to be target collision-resistant when $\Pr[(x, s) \leftarrow \mathcal{A}; \mathcal{H} \leftarrow \mathrm{HashGen}(1^\ell); x' \leftarrow \mathcal{A}(\mathcal{H}, s) : \mathcal{H}(x) = \mathcal{H}(x') \wedge x \neq x']$ is negligible for any polynomial-time algorithm $\mathcal{A}$.

Non-interactive Proofs. A non-interactive proof system for an NP-relation $R \in \{0,1\}^* \times \{0,1\}^*$ defining $L = \{x | (x,w) \in R \text{ for some } w\}$ consists of three algorithms (K, P, V), which satisfy the following correctness and soundness conditions: For correctness, it is required that for any security parameter $\ell \in \mathbb{N}$, any common reference string $crs \leftarrow K(1^\ell)$, and any pair $(x,w) \in R$, it holds that $V(1^\ell, crs, x, P(1^\ell, crs, x, w)) = \top$; for soundness, it is required that for any $\ell \in \mathbb{N}$ and any probabilistic polynomial-time algorithm $\mathcal{A}$, the probability $\Pr[crs \leftarrow K(1^\ell); (x,\pi) \leftarrow \mathcal{A}(1^\ell, crs) : V(1^\ell, crs, x, \pi) = \top \wedge x \notin L]$ is negligible. In fact we will later use two types of proof systems, one which is zero-knowledge [5,13] and one which is simulation-sound [26] in addition to zero-knowledge.

Bilinear Maps and Groth-Sahai Proofs. Bilinear groups are groups $\mathbb{G}$ and $\mathbb{G}_T$ with the same order that have an efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Let $\mathcal{G}$ be a probabilistic polynomial-time algorithm that outputs a group parameter $gk = (p, \mathbb{G}, \mathbb{G}_T, e, g)$ where p is the order of $\mathbb{G}$ and $\mathbb{G}_T$, e is a non-degenerates bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, and g is a generator of $\mathbb{G}$.

Groth and Sahai [20] introduced a framework for very efficient non-interactive proof for the satisfiability of some algebraic equations they called quadratic equations. The proof system consists of algorithms $(K_{\mathrm{NI}}, P, V, X)$. The algorithm $K_{\mathrm{NI}}(gk)$ takes a group parameter gk as input and outputs (crs, xk), where crs is a common reference string and xk is a trapdoor extraction key for extracting a witness from a proof. The algorithm $P(crs, x, w)$ outputs a proof π for an equation described by x whose witness is w. A proof π is verified by running $V(crs, x, \pi)$. The algorithm $X_{xk}(x, \pi)$ extracts a witness from the proof π which passes the verification algorithm. In fact there are two types of proof systems $(K_{\mathrm{NI}}, P_{\mathrm{NIWI}}, V_{\mathrm{NIWI}}, X)$ and $(K_{\mathrm{NI}}, P_{\mathrm{NIZK}}, V_{\mathrm{NIZK}}, X)$, which respectively provide witness-indistinguishability and zero-knowledge properties. The two proof systems have the identical common reference string generation algorithm. Moreover they share single string for different sets of equations in the Groth group signature scheme.

The common reference string consists of eight group elements as $crs = (F, H, U, V, W, U', V', W')$. A notable property on this is that F and H essentially serve as a public key of the linear encryption [6]. This property is exploited in the Groth group signature scheme (and so in our modification of that scheme). For further details see [20].

Tag-Based Encryption. Tag-based encryption is an extension of public key encryption, which associates an additional "tag" with a ciphertext. The exact syntax is as follows: A key generation algorithm $G(1^\ell)$ generates a public key pk and a secret key dk; an encryption algorithm $E_{pk}(t, m)$ takes as input a public key pk, a tag t, and a plaintext m, and outputs a ciphertext c; a decryption algorithm $D_{dk}(t, c)$ takes as input a decryption key dk, a tag t, and a ciphertext c and outputs a plaintext m or a special symbol $\perp$ indicating the decryption failed. The correctness condition only ensures that the plaintext is recovered when the tags used in the encryption and the decryption are identical.

In this paper we use Kiltz's construction of tag-based encryption [24], which is explained below. The scheme can be built on bilinear groups. Let $gk = (p, \mathbb{G}, \mathbb{G}_T, e, g)$ be a group description. The key generation algorithm chooses random integers $\phi, \eta \leftarrow \mathbb{Z}_p$ and random elements $K, L \leftarrow \mathbb{G}$, and sets $pk = (F, H, K, L)$ where $F = g^{\phi}$ and $H = g^{\eta}$ and $dk = (\phi, \eta)$. A ciphertext of a plaintext m under a tag t is computed as $y = (y_1, y_2, y_3, y_4, y_5) = (F^r, H^s, mg^{r+s}, (g^t K)^r, (g^t L)^s)$. The decryption algorithm decrypts a ciphertext $(y_1, y_2, y_3, y_4, y_5)$ under a tag t by checking $e(F, y_4) = e(y_1, g^t K)$ and $e(H, y_5) = e(y_2, g^t L)$ and outputs $y_3 / y_1^{\phi} y_2^{\eta}$ if the two equations hold, otherwise outputs $\perp$. This encryption scheme is secure against selective-tag weak chosen-ciphertext attacks if the decisional linear assumption holds [24]. Another interesting property is that the scheme has public verifiability in the sense that it can be efficiently checked whether a given five-tuple $(y_1, y_2, y_3, y_4, y_5)$ lies in the range of the encryption algorithm under a given public key pk and a given tag t by checking the two equations $e(F, y_4) = e(y_1, g^t K)$ and $e(H, y_5) = e(y_2, g^t L)$.

3 Opening Soundness

In this section we give a formal definition of opening soundness. Specifically, we introduce two variants of opening soundness, weaker and stronger definitions.

The weaker definition, named weak opening soundness, is intended to address the security concerns discussed in the introduction in a straightforward manner, and will rule out the possibility that a malicious user can claim ownership of a signature produced by an honest user by forging an opening proof. The definition is as follows:

Definition 5. *A group signature scheme is said to have* weak opening soundness *if*

$$\Pr[(gpk, ik, ok) \leftarrow \mathsf{GKg}(1^k); (m, i, i^*, s) \leftarrow \mathcal{A}^{\mathsf{AddU}(\cdot)}(gpk); \\ \Sigma \leftarrow \mathsf{GSig}(gpk, gsk_i, m); \tau^* \leftarrow \mathcal{A}^{\mathsf{AddU}(\cdot)}(s, \Sigma, gsk_{i^*}) \\ : i \neq i^* \wedge i, i^* \in \mathsf{HU} \wedge \mathsf{Judge}(gpk, upk_{i^*}, m, \Sigma, \tau^*) = 1]$$

is negligible for all polynomial time adversaries $\mathcal{A}$, where the oracle AddU *is defined as follows:*

AddU: *On a query $i \in \mathbb{N}$, the oracle runs $(upk_i, usk_i) \leftarrow \mathsf{UKg}(gpk)$, then executes the protocol $(gsk_i, reg_i) \leftarrow \langle \mathsf{Join}(gpk, upk_i, usk_i), \mathsf{Issue}(gpk, ik) \rangle$, adds i to a set $\mathcal{HU}$, and lastly returns upk_i.*

Note that the adversary is only allowed to receive the secret signing key of a single user i^*. Hence, this definition will not rule out attacks involving a corrupted opener, and therefore cannot contribute towards reducing trust in this entity.

In contrast, the stronger definition, named opening soundness, is intended to rule out the possibility that an adversary can produce two different openings of a signature, even if he is allowed to corrupt the opener and all the users in the system, and furthermore generate the signature in question maliciously. The definition is as follows:

Definition 6. *A group signature scheme is said to have* opening soundness *if*

$$\Pr[(gpk, ik, ok) \leftarrow \mathsf{GKg}(1^k); (m, \Sigma, i_1, \tau_1, i_2, \tau_2) \leftarrow \mathcal{A}^{\mathsf{CrptU},\mathsf{WReg}}(gpk, ok, ik) : \mathsf{GVf}(gpk, m, \Sigma) = 1 \wedge i_1 \neq i_2 \wedge \mathsf{Judge}(gpk, upk_{i_1}, m, \Sigma, \tau_1) = 1 \wedge \mathsf{Judge}(gpk, upk_{i_2}, m, \Sigma, \tau_2) = 1]$$

is negligible for all polynomial time adversaries $\mathcal{A}$, where the oracle $\mathsf{CrptU}(i, M)$ sets the user public key of the user i to be M, and the oracle $\mathsf{WReg}(i, M)$ sets $reg[i]$ to M.

While the weaker definition provides a minimum level of protection against the type of attacks described in the introduction, we believe that, when applied to the scenarios mentioned in the introduction, any dynamic group signature scheme should provide (ordinary) opening soundness to prevent any type of attack which exploits ambiguity of openings, or involves a corrupted opener. Furthermore, we will show that this level of security can be achieved efficiently by showing that our modified version of the scheme by Groth provides opening soundness (See Sect. 5 for details).

4 Opening Soundness of Existing Schemes

We will now take a closer look at some of the existing dynamic group signature schemes, and highlight the level of opening soundness (ordinary, weak or none) achieved by these. Note that since the Bellare-Shi-Zhang security model for dynamic group signatures does not considers opening soundness, a security proof in this model will not allow us to make any conclusions regarding the opening soundness of existing schemes.

In this section, we will focus on the standard model scheme by Groth described in [19] (note that the updated scheme in [19] is slightly different from the scheme described in [18]) and the generic construction of a dynamic group signature scheme by Bellare, Shi, and Zhang [3]. More specifically, we will show that the scheme by Groth does not have weak opening soundness whereas the generic construction by Bellare, Shi and Zhang has opening soundness. We further show that the random oracle model schemes by Furukawa and Imai [14] and Bichsel et al. [4] do not have opening soundness. Interestingly, while these schemes do not provide opening soundness, there seems to be no obvious attack against the weak opening soundness of these.

The Groth Scheme. Figure 1 shows a description of the Groth scheme. Below, we will expand on the description given in the figure.

In the group key generation algorithm GKg, the elements f, h, T correspond to a verification key of the Zhou-Lin signature scheme [27], whereas z corresponds to a signing key. Furthermore, pk is a public key of Kiltz's tag-based encryption scheme. Note that the first two elements of pk and the common reference string crs for the non-interactive Groth-Sahai proofs are identical.

$\mathsf{GKg}(1^k)$:
$gk \leftarrow \mathcal{G}(1^k)$; $\mathcal{H} \leftarrow \mathrm{HashGen}(1^k)$
$(f, h, z) \leftarrow \mathbb{G}$; $T = e(f, z)$
$(crs, xk) \leftarrow K_{\mathrm{NI}}(gk)$;
$(F, H, U, V, W, U', V', W') \leftarrow crs$;
$K, L \leftarrow G$; $pk \leftarrow (F, H, K, L)$
(gpk, ik, ok)
$\leftarrow ((gk, \mathcal{H}, f, h, T, crs, pk), z, xk)$

$\mathsf{GSig}(gpk, gsk_i, m)$:
$(vk_{\mathrm{sots}}, sk_{\mathrm{sots}}) \leftarrow \mathrm{KeyGen}_{\mathrm{sots}}(1^k)$
(Repeat until $\mathcal{H}(vk_{\mathrm{sots}}) \neq -x_i$)
$\rho \leftarrow \mathbb{Z}_p$; $a \leftarrow a_i f^{-\rho}$; $b \leftarrow b_i (hv_i)^{\rho}$
$\sigma \leftarrow g^{1/(x_i + \mathcal{H}(vk_{\mathrm{sots}}))}$
$\pi \leftarrow P_{\mathrm{NIWI}}(crs, (gpk, a, \mathcal{H}(vk_{\mathrm{sots}})), (b, v_i, \sigma))$
$y \leftarrow E_{pk}(\mathcal{H}(vk_{\mathrm{sots}}), \sigma)$
$\psi \leftarrow P_{\mathrm{NIZK}}(crs, (gpk, y, \pi), (r, s, t))$
$\sigma_{\mathrm{sots}} \leftarrow \mathrm{Sign}_{sk_{\mathrm{sots}}}(vk_{\mathrm{sots}}, m, a, \pi, y, \psi)$
Return $\Sigma = (vk_{\mathrm{sots}}, a, \pi, y, \psi, \sigma_{\mathrm{sots}})$

$\mathsf{Join}/\mathsf{Issue}$(User i: gpk; Issuer: gpk, ik):
Run the coin-flipping protocol in [19]
The user obtains $v_i = g^{x_i}$ and x_i
and the issuer obtains v_i
Issuer: $r \leftarrow \mathbb{Z}_p$;
$(a_i, b_i) \leftarrow (f^{-r}, (v_i h)^r z)$;
set $reg[i] \leftarrow v_i$
send (a_i, b_i) to the user
User: If $e(a_i, hv_i) e(f, b_i) = T$
set $upk_i \leftarrow v_i$, $gsk_i \leftarrow (x_i, a_i, b_i)$

$\mathsf{GVf}(gpk, m, \Sigma)$:
Return 1 if the following holds:
$1 = \mathrm{Ver}_{vk_{\mathrm{sots}}}((vk_{\mathrm{sots}}, m, a, \pi, y, \psi), \sigma_{\mathrm{sots}})$,
$1 = V_{\mathrm{NIWI}}(crs, (gpk, a, \mathcal{H}(vk_{\mathrm{sots}})), \pi)$,
$1 = V_{\mathrm{NIZK}}(crs, (gpk, \pi, y), \psi)$, and
$1 = \mathrm{ValidCiphertext}(pk, \mathcal{H}(vk_{\mathrm{sots}}), y)$,
else return 0

$\mathsf{Open}(gpk, ok, reg, m, \Sigma)$:
(b, v, σ)
$\leftarrow X_{xk}(crs, (gpk, a, \mathcal{H}(vk_{\mathrm{sots}})), \pi)$
Return (i, σ) if there is i so $v = reg[i]$,
else return $(0, \sigma)$

$\mathsf{Judge}(gpk, i, upk_i, m, \Sigma, \sigma)$:
Return 1 if
$i \neq 0 \wedge e(\sigma, v_i g^{\mathcal{H}(vk_{\mathrm{sots}})}) = e(g, g)$,
else return 0

Fig. 1. The Groth group signature scheme [19]

In the group signing algorithm GSig, a group member constructs two non-interactive Groth-Sahai proofs. The first proof π, constructed via P_{NIWI}, shows knowledge of a signature σ, a verification key v and a part b of a (re-randomized) certificate (a, b) which satisfy $e(a, hv)e(f, b) = T \wedge e(\sigma, vg^{\mathcal{H}(vk_{\mathrm{sots}})}) = e(g, g)$. The first part a of the certificate can safely be revealed as part of the group signature since it does not leak any information about the identity of the member due to the re-randomization. The second proof ψ, constructed via P_{NIZK}, demonstrates that the plaintext of y is the same as the witness σ used in π. Let us explain in detailed. The tag-based encryption y has the form $(y_1, y_2, y_3, y_4, y_5) = (F^{r_y}, H^{s_y}, g^{r_y + s_y} \sigma, (g^{\mathcal{H}(vk_{\mathrm{sots}})} K)^{r_y}, (g^{\mathcal{H}(vk_{\mathrm{sots}})} L)^{s_y})$, while the Groth-Sahai proof π contains a commitment $c = (c_1, c_2, c_3) = (F^{r_c} U^t, H^{s_c} V^t, g^{r_c + s_c} W^t \sigma)$. The proof demonstrates that there exists (r, s, t) such that $(c_1 y_1^{-1}, c_2 y_2^{-1}, c_3 y_3^{-1}) = (F^r U^t, H^s V^t, g^{r+s} W^t)$. When y and c encrypt the same message, there exists (r, s, t) that satisfies above equation, but if y and c encrypt different messages, no such tuple (r, s, t) exists.

The verification algorithm GVf will, in addition to the verification of the two non-interactive proofs and the one-time signature, verify that the ciphertext y

is a valid ciphertext, using the algorithm ValidCiphertext. This algorithm is easily implemented for the tag-based encryption scheme by Kiltz (see the last paragraph of Sect. 2 for details).

We will now show how a malicious group member can forge a opening proof which shows that he is the signer of any signature Σ produced by user i. As shown above, an opening proof consists of a certified signature σ on vk_{sots} which is part of Σ. To verify the opening proof, it is only verified that σ is a valid signature on vk_{sots} under the verification key v_i of the user in question.

Hence, a malicious user i' who wants to impersonate the signer of the group signature Σ on m, simply uses his own private signing $x_{i'}$ key to construct a new signature σ' on vk_{sots}, and publicizes this as an opening proof together with his own identity i'. This proof will be accepted by the Judge algorithm since σ' is a valid signature in vk_{sots}.

We formally state this as a theorem:

Theorem 1. *The Groth group signature scheme does not provide weak opening soundness.*

Proof. We describe an algorithm for producing a forged proof: When the adversary receives the security parameter 1^ℓ and a group public key gpk, it firstly issues two queries $\mathsf{AddU}(1)$ and $\mathsf{AddU}(2)$ in order to add two members 1 and 2 the group. The adversary then requests the challenge by outputting $(i, i^*, m) = (1, 2, 0^\ell)$, and receives a tuple (Σ, gsk_2), where $\Sigma = (vk_{\mathrm{sots}}, a, \pi, y, \psi, \sigma_{\mathrm{sots}})$ and $gsk_2 = (x_2, a_2, b_2)$. The adversary forges a proof of ownership by computing $\sigma^* = g^{1/(x_2+\mathcal{H}(vk_{\mathrm{sots}}))}$ and outputs σ^* (Notice that vk_{sots} is taken from the group signature Σ).

One can easily verify that $\mathsf{Judge}(gpk, 2, reg[2], m, \Sigma, \sigma^*)$ actually outputs 1, which means that the algorithm successfully breaks the opening soundness. □

The Bellare-Shi-Zhang Scheme. Below, we will give an intuitive description of the generic construction of a dynamic group signature scheme by Bellare, Shi, and Zhang.

In the Bellare-Shi-Zhang construction, each group member i has a key pair (vk_i, sk_i) of an EUF-CMA secure signature scheme. The issuer also possesses his own key pair (ak, ck) of the signature scheme. The issuer signs the message $\langle i, vk_i \rangle$ to obtain the signature $cert_i$, and sends $cert_i$ to the user i. A group signature on a message m by the user i is a pair (C, π): here C is an encryption of $\langle i, vk_i, cert_i, s \rangle$, s is a signature on m under the key pair (vk_i, sk_i), and the NIZK proof π proves that the plaintext encrypted in C is of the form $\langle i, vk, cert, s \rangle$. The opener attributes a group signature $\Sigma = (C, \pi)$ to the user i by providing an NIZK proof τ for another statement (i.e. different from that of π), which claims the existence of a decryption key that corresponds to the opener's public key and that under that key C is decrypted to $\langle i, vk_i, cert_i, s \rangle$.

This simple scheme provides opening soundness. Intuitively, this is due to the correctness of the public key encryption used to encrypt the signature and the certificate, and the soundness of the NIZK proof system for τ. The correctness

condition of public key encryption ensures that given a public key pk and a ciphertext C, the decryption of C is determined uniquely. Now, let us assume that an adversary of the opening soundness game outputs a tuple $(m, \Sigma, i_1, \tau_1, i_2, \tau_2)$ where $\Sigma = (C, \pi)$ and wins the game. The proof τ_1 proves that C decrypts to $\langle i_1, vk, cert, s\rangle$ for some vk, $cert$, and s, whereas τ_2 proves that C decrypts to a different plaintext $\langle i_2, vk', cert', s'\rangle$ for some vk', $cert'$, and s'. However, this should not be possible since the decryption of C under a fixed public key is unique. Hence, the adversary breaks the soundness of the NIZK proof system.

A formal statement and its proof are deferred to the full version.

The Furukawa-Imai Scheme. The Furukawa-Imai group signature scheme does not have opening soundness, which we will show in the following.

The scheme exploits a group $\mathbb{G}$ on which the decisional Diffie-Hellman assumption holds, in addition to bilinear groups $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ with an asymmetric bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. In this scheme, each group member i has a public key $Q_i = g^{x_i}$ and its corresponding secret key x_i. The public key Q_i is encrypted in a group signature with (a kind of) ElGamal encryption. Let $(R, V) = (Q_i g^r, S^r)$ be the ciphertext that appears in a group signature, where $S = g^s$ is the public key of the ElGamal encryption. The opener possesses the decryption key s, and identifies the signer by decrypting the ciphertext. An opening contains a proof of knowledge of w such that $Q_i = R/V^{1/w}$, where Q_i is the public key of the specified member (The opener uses s as the witness for the above equation).

If the adversary corrupts the opener and two different members i and j, the adversary can construct two different openings of a single signature, each of which attributes the signature to the user i or the user j, respectively. The adversary proceeds as follows: At first the signature is honestly generated by the user i. Let $(R, V) = (g^{x_i+r}, S^r)$ be the ciphertext contained in this signature. The first opening is also honestly generated by the opener to attribute the signature to i. The second proof is generated by computing a proof of knowledge w that satisfies $Q_j = R/V^{1/w}$ with the witness $w = sr/(x_i + r - x_j)$. This proof attributes the signature to the user j. Note that the randomness r for the encryption is reused to forge the second proof. This is the reason why the adversary needs to corrupt the user i, not only the user j and the opener.

The Bichsel et al. Scheme. In the Bichsel et al. scheme, a group member receives a Camenisch-Lysyanskaya signature on a random message ξ from the issuer. To generate a group signature, the member rerandomizes the certificate and computes a "signature of knowledge" of ξ. This rerandomized certificate and the signature of knowledge constitute the group signature.

The issuer should not know the random message ξ, because otherwise non-frameability is compromised. For this reason, in a group-joining protocol, the ξ is jointly generated by the user and the issuer as follows: The user i chooses a random exponent τ_i and sends $\tilde{r} = \tilde{x}^{\tau_i}$ to the issuer, while the issuer also chooses a random κ_i and computes $\tilde{w} = \tilde{r} \cdot \tilde{x}^{\kappa_i} = \tilde{x}^{\tau_i + \kappa_i}$. This $\tau_i + \kappa_i$ will be used as the random message ξ mentioned above. To establish a publicly verifiable

connection between this ξ and the user i, the user i generates an (ordinary) signature on $k_i = e(g, \tilde{r})$ with a key pair which is previously registered in a public key infrastructure.

To open a signature, the opener uses $\tilde{w}$ to identify which ξ is the message of the Camenisch-Lysyanskaya signature. Since $\tilde{w}$ makes the Camenisch-Lysyanskaya signature publicly verifiable, it cannot be used as an opening. Instead, the opener produces a non-interactive zero-knowledge proof of $\tilde{w}$ and κ_i such that $k_i = e(g, \tilde{w})/e(g, \tilde{x})^{\kappa_i}$ and provides the signature on k_i. To verify this opening, a third party simply verifies the non-interactive zero-knowledge proof and the signature.

Unfortunately this scheme does not satisfy opening soundness. Assume a malicious signer obtains a group signature by an honest user, and further obtains an honestly generated opening of the signature. The proof of ownership contains k_i and a signature on this by the honest user. The malicious signer replaces the signature on k_i with his own signature on k_i. This forged opening passes the verification.

5 Achieving Opening Soundness

In this section we present a variant of the Groth scheme, which provides opening soundness (besides anonymity, non-frameability, and traceability).

5.1 The Modified Groth Scheme

The High-Level Idea. Let us first consider a general approach for achieving opening soundness.

The opener, who has the secret opening key, will always be able to determine the correct opening. To provide opening soundness, the opener needs to convince others that a given opening is correct. The easiest way to do that is to make the opening key public, but this will compromise the anonymity of the scheme. Instead, the opener can provide an NIZK proof of the correctness of an opening, to convince any third party. This is, in fact, the approach used in the Bellare-Shi-Zhang construction.

If the opening algorithm essentially corresponds to a "decryption" of a ciphertext contained in the group signature (this is the case for many existing schemes), we might be able to take a different and more efficient approach. If the encryption scheme provides randomness recovering, the opener can simply release the randomness used for the ciphertext in question instead of an expensive zero-knowledge proof. Any third party will then be able to verify the correctness of an opening by re-encrypting the relevant information with the randomness provided by the opener, and then confirm that the resulting ciphertext is the same as the one contained in the signature.

In the Groth scheme, an opening essentially corresponds to the decryption of a linear encryption scheme. While linear encryption is not randomness-recovering, the opener is able to release related values which, together with an algebraic trick using a bilinear map, allow a third party to confirm that the decryption

was done correctly. This property will allow us to add opening soundness to the original scheme. More specifically, in our variant of the Groth scheme, the opener, given a ciphertext $(c_1, c_2, c_3) = (F^r, H^s, vg^{r+s})$, reveals g^r and g^s as a part of an opening. Using the properties of the bilinear map, these values can replace the exact randomness r and s when checking the correspondence between a ciphertext and a decryption: If a third party, given g^r and g^s, wants to check the correspondence between a ciphertext (c_1, c_2, c_3) and a decryption v, he simply checks whether the equations $e(F, g^r) = e(c_1, g)$, $(H, g^s) = e(c_2, g)$, and $v = c_3/(g^r g^s)$ hold. If this is the case, he accepts the decryption as valid.

This idea is essentially the same as that used by Galindo et al. [15] in the context of public key encryption with non-interactive opening (PKENO). In [15], the application of PKENO schemes to group signature is briefly discussed as a mechanism for simplifying the construction of an opening. We will show that this technique is also able to ensure the opening soundness of group signature schemes.

Description of Our Variant. The Groth scheme can achieve opening soundness with the small modification shown in Fig. 2.

$\mathsf{Open}(gpk, ok, m, \Sigma)$:
If $\mathsf{GVf}(gpk, m, \Sigma) = 0$, return $(0, \perp)$
$(b, v, \sigma) \leftarrow X_{xk}(crs, (gpk, a, \mathcal{H}(vk_{\mathrm{sots}})), \pi)$
$(d_F, d_H) \leftarrow xk$; $(y_1, \ldots, y_5) \leftarrow y$
$\tau_F := y_1^{1/d_F}$; $\tau_H := y_2^{1/d_H}$
Return $(i, (\sigma, \tau_F, \tau_H))$
  if there is i so $v = reg[i]$,
else $(0, \perp)$

$\mathsf{Judge}(gpk, i, reg[i], m, \Sigma, (\sigma, \tau_F, \tau_H))$:
Return 1 if the following holds:
$\mathsf{GVf}(gpk, m, \Sigma) = 1$,
$i \neq 0$, $e(\sigma, v_i g^{\mathcal{H}(vk_{\mathrm{sots}})}) = e(g, g)$,
$e(F, \tau_F) = e(y_1, g)$, $e(H, \tau_H) = e(y_2, g)$,
and $\sigma\tau_F\tau_H = y_3$,
else return 0

Fig. 2. The proposed modification of the Groth group signature scheme. The algorithms that do not appear in the figure are exactly the same as in Fig. 1.

Theorem 2. *The modified Groth scheme shown in Fig. 2 provides opening soundness.*

Proof. Let us consider the game in Definition 6, and let gpk be the group public key in the game, where the key is parsed to $(F, H, \cdots)$, and $(m, \Sigma, i, \tau, i', \tau')$ be the output of the adversary. Let Σ, τ, and τ' be parsed as follows: $\Sigma = (vk_{\mathrm{sots}}, a, \pi, y, \psi, \sigma_{\mathrm{sots}})$ in which $y = (y_1, y_2, y_3, y_4, y_5)$, $\tau = (\sigma, \tau_F, \tau_H)$ and $\tau' = (\sigma', \tau'_F, \tau'_H)$.

We hereafter show that given a fixed Σ, it must hold that $i = i'$: Given a fixed Σ (in particular y_1, y_2, and y_3), the verification equations

$$e(F, \tau_F) \stackrel{?}{=} e(y_1, g) \wedge e(H, \tau_H) \stackrel{?}{=} e(y_2, g) \wedge \sigma\tau_F\tau_H \stackrel{?}{=} y_3$$

uniquely determine τ_F, τ_H, and σ. Since both $\tau = (\sigma, \tau_F, \tau_H)$ and $\tau' = (\sigma', \tau'_F, \tau'_H)$ pass the Judge verification, we must have that $(\sigma, \tau_F, \tau_H) = (\sigma', \tau'_F, \tau'_H)$. Now

v_i satisfies $e(\sigma, v_i g^{\mathcal{H}(vk_{\text{sots}})}) = e(g, g)$ and $v_{i'}$ satisfies $e(\sigma, v_{i'} g^{\mathcal{H}(vk_{\text{sots}})}) = e(g, g)$, but because $\sigma = \sigma'$, and the equation $e(\sigma, v g^{\mathcal{H}(vk_{\text{sots}})}) = e(g, g)$ uniquely determines v given fixed σ and $\mathcal{H}(vk_{\text{sots}})$, we have that $v_i = v_{i'}$, which implies that $i = i'$. □

The changes shown in Fig. 2 yields a scheme which is secure in the BSZ model i.e. the anonymity, the non-frameability, and the traceability of the original Groth scheme are maintained. This will be shown in the following.

Theorem 3. *The modified Groth scheme provides anonymity if the decision linear assumption holds on $\mathbb{G}$, the one-time signature scheme is strongly unforgeable, and the hash function is target collision-resistant.*

Proof (Sketch). The proof proceeds almost as in the original Groth scheme. The biggest difference from the original proof is that the simulator for the modified scheme needs to simulate two additional group elements $(\tau_F, \tau_H) = (g^r, g^s)$ when receiving an Open query. Note that in the simulation of Kiltz's tag-based encryption, when the simulator receives a decryption query $(y_1, y_2, y_3, y_4, y_5) = (F^r, H^s, mg^{r+s}, (g^t K)^r, (g^t L)^s)$, the simulator at first extracts g^r and g^s without the knowledge of the decryption key and then simulates the decryption by computing $y_3 / g^r g^s$. In a similar way, it is possible to simulate the two extra components required in our scheme. □

Non-frameability and traceability can be proven more easily since these security notions do not require simulation of the Open oracle. For non-frameability, once an opening of the modified scheme that compromises the non-frameability notion is produced, one can obtain an opening for the original scheme (by simply dropping the extra components of τ_F and τ_H) which will compromise the non-frameability of the original scheme.

Theorem 4. *The modified Groth scheme provides non-frameability.*

Theorem 5. *The modified Groth scheme provides traceability.*

6 Conclusion

We have identified an overlooked security concern for dynamic group signatures, namely, the possibility that a false opening proof can be produced by a corrupt user. To address this concern, we defined (two variants of) a new security notion denoted opening soundness, and furthermore discussed the opening soundness of several existing schemes. As a result, we have shown that the Bellare-Shi-Zhang construction [3] provides opening soundness as it is, and that small modifications to the Groth scheme (of the full version) [19] allow this scheme to provide opening soundness as well. We have also briefly discussed the opening soundness of some of the random oracle schemes [14,4], but leave further investigation of these schemes as future work.

Acknowledgment. The authors would like to thank the anonymous reviewers of PKC 2012. The authors are also grateful to Benoît Libert for his invaluable comments.

References

1. Ateniese, G., Camenisch, J., Joye, M., Tsudik, G.: A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 255–270. Springer, Heidelberg (2000)
2. Bellare, M., Micciancio, D., Warinschi, B.: Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 614–629. Springer, Heidelberg (2003)
3. Bellare, M., Shi, H., Zhang, C.: Foundations of Group Signatures: The Case of Dynamic Groups. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 136–153. Springer, Heidelberg (2005)
4. Bichsel, P., Camenisch, J., Neven, G., Smart, N.P., Warinschi, B.: Get Shorty via Group Signatures without Encryption. In: Garay, J.A., De Prisco, R. (eds.) SCN 2010. LNCS, vol. 6280, pp. 381–398. Springer, Heidelberg (2010)
5. Blum, M., De Santis, A., Micali, S., Persiano, G.: Noninteractive zero-knowledge. SIAM J. Comput. 20(6), 1084–1118 (1991)
6. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
7. Boyen, X., Waters, B.: Compact Group Signatures Without Random Oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 427–444. Springer, Heidelberg (2006)
8. Boyen, X., Waters, B.: Full-Domain Subgroup Hiding and Constant-Size Group Signatures. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 1–15. Springer, Heidelberg (2007)
9. Camenisch, J., Lysyanskaya, A.: Signature Schemes and Anonymous Credentials from Bilinear Maps. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 56–72. Springer, Heidelberg (2004)
10. Camenisch, J., Michels, M.: A Group Signature Scheme with Improved Efficiency (Extended Abstract). In: Ohta, K., Pei, D. (eds.) ASIACRYPT 1998. LNCS, vol. 1514, pp. 160–174. Springer, Heidelberg (1998)
11. Chaum, D., van Heyst, E.: Group Signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991)
12. Delerablée, C., Pointcheval, D.: Dynamic Fully Anonymous Short Group Signatures. In: Nguyên, P.Q. (ed.) VIETCRYPT 2006. LNCS, vol. 4341, pp. 193–210. Springer, Heidelberg (2006)
13. Feige, U., Lapidot, D., Shamir, A.: Multiple non-interactive zero knowledge proofs based on a single random string. In: 31st Annual Symposium on Foundations of Computer Science, pp. 308–317. IEEE Computer Society (1990)
14. Furukawa, J., Imai, H.: An Efficient Group Signature Scheme from Bilinear Maps. In: Boyd, C., González Nieto, J.M. (eds.) ACISP 2005. LNCS, vol. 3574, pp. 455–467. Springer, Heidelberg (2005)
15. Galindo, D., Libert, B., Fischlin, M., Fuchsbauer, G., Lehmann, A., Manulis, M., Schröder, D.: Public-Key Encryption with Non-Interactive Opening: New Constructions and Stronger Definitions. In: Bernstein, D.J., Lange, T. (eds.) AFRICACRYPT 2010. LNCS, vol. 6055, pp. 333–350. Springer, Heidelberg (2010)

16. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. 17(2), 281–308 (1988)
17. Groth, J.: Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 444–459. Springer, Heidelberg (2006)
18. Groth, J.: Fully Anonymous Group Signatures Without Random Oracles. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 164–180. Springer, Heidelberg (2007)
19. Groth, J.: Fully anonymous group signatures without random oracles, May 17 (2010) (manuscript), http://www.cs.ucl.ac.uk/staff/J.Groth/CertiSignFull.pdf
20. Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
21. Kiayias, A., Tsiounis, Y., Yung, M.: Traceable Signatures. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 571–589. Springer, Heidelberg (2004)
22. Kiayias, A., Yung, M.: Group Signatures with Efficient Concurrent Join. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 198–214. Springer, Heidelberg (2005)
23. Kilian, J., Petrank, E.: Identity Escrow. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 169–185. Springer, Heidelberg (1998)
24. Kiltz, E.: Chosen-Ciphertext Security from Tag-Based Encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006)
25. Rackoff, C., Simon, D.R.: Non-interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)
26. Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: 40th Annual Symposium on Foundations of Computer Science, pp. 543–553. IEEE Computer Society (1999)
27. Zhou, S., Lin, D.: Shorter Verifier-Local Revocation Group Signatures from Bilinear Maps. In: Pointcheval, D., Mu, Y., Chen, K. (eds.) CANS 2006. LNCS, vol. 4301, pp. 126–143. Springer, Heidelberg (2006)

Author Index