Henri Gilbert (Ed.)

LNCS 6110

Advances in Cryptology – EUROCRYPT 2010

29th Annual International Conference
on the Theory and Applications of Cryptographic Techniques
French Riviera, May/June 2010, Proceedings

INTERNATIONAL ASSOCIATION FOR CRYPTOLOGIC RESEARCH

c i a r

Springer

Lecture Notes in Computer Science 6110

Henri Gilbert (Ed.)

Advances in Cryptology – EUROCRYPT 2010

29th Annual International Conference
on the Theory and Applications of Cryptographic Techniques
French Riviera, May 30 – June 3, 2010
Proceedings

Springer

Volume Editor

Henri Gilbert
Orange Labs/MAPS/STT
38–40 rue du Général Leclerc, 92794 Issy les Moulineaux Cedex 9, France
E-mail: henri.gilbert@orange-ftgroup.com

Library of Congress Control Number: 2010926925

CR Subject Classification (1998): E.3, K.6.5, C.2, D.4.6, J.1, G.2.1

LNCS Sublibrary: SL 4 – Security and Cryptology

ISSN 0302-9743
ISBN-10 3-642-13189-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-13189-9 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

These are the proceedings of Eurocrypt 2010, the 29th in the series of European conferences on the Theory and Application of Cryptographic Techniques. The conference was sponsored by the International Association for Cryptologic Research and held on the French Riviera, May 30–June 3, 2010.

A total of 191 papers were received of which 188 were retained as valid submissions. These were each assigned to at least three Program Committee members and a total of 606 review reports were produced. The printed record of the reviews and extensive online discussions that followed would be almost as voluminous as these proceedings. In the end 35 submissions were accepted with two submission pairs being merged to give 33 papers presented at the conference. The final papers in these proceedings were not subject to a second review before publication and the authors are responsible for their contents.

The Program Committee, listed on the next page, deserves particular thanks for all their hard work, their outstanding expertise, and their constant commitment to all aspects of the evaluation process. These thanks are of course extended to the very many external reviewers who took the time to help out during the evaluation process. It was also a great pleasure to honor and welcome Moti Yung who gave the 2010 IACR Distinguished Lecture.

It might be recalled that Eurocrypt 2010 took place under exceptionally difficult circumstances. First, in the aftermath of the financial crisis, sponsorship was a low priority for many companies. We are therefore grateful to I3S, Ingenico, Microsoft, Nagravision, Oberthur, Orange Labs, Qualcomm, Sagem Sécurité, and Technicolor for their support of Eurocrypt 2010. We specifically acknowledge the kind efforts of Hervé Chabanne, Guillaume Dabosville, Jean-Bernard Fischer, Paul Friedel, Marc Joye, François Larbey, Kristin Lauter, Bruno Martin, David Naccache, Jim Ostrich, and Greg Rose for making it happen. Second, longstanding plans for Eurocrypt 2010 were disrupted by the sudden decision of the French Government to hold an international summit at the same time and at the same venue. For their help following this forced relocation, we would like to extend our gratitude to our friends and family members who helped with wise advice, good connections, and imaginative suggestions.

We would like to thank the IACR board for the honor of hosting Eurocrypt 2010. Particular thanks are due to Shai Halevi for all his unseen work on the submission, review, and registration sites, to Antoine Joux for sharing his experience as Program Chair of Eurocrypt 2009, and to Helena Handschuh and Bart Preneel for their constant advice, help, and support. Last, but not least, we are grateful for the help and input of our colleagues Ryad Benadjila, Gilles Macario-Rat, and Yannick Seurin, all at Orange Labs.

March 2010

Henri Gilbert (Program Chair)
Olivier Billet (General Chair)
Matthew Robshaw (General Chair)

Organization

General Chairs

Olivier Billet	
Matthew Robshaw	Orange Labs, France

Program Chair

Henri Gilbert	Orange Labs, France

Program Committee

Dan Boneh	Stanford University
Ran Canetti	Tel Aviv University
Anne Canteaut	INRIA
Carlos Cid	Royal Holloway, University of London
Jean-Sébastien Coron	Université du Luxembourg
Ivan Damgård	University of Aarhus
Steven Galbraith	Auckland University
Rosario Gennaro	IBM Research
Helena Handschuh	K.U.Leuven and Intrinsic-ID Inc.
Stanislaw Jarecki	University of California at Irvine
Antoine Joux	DGA and Université de Versailles
Marc Joye	Technicolor
Ari Juels	RSA Laboratories
Aggelos Kiayias	University of Connecticut
Lars Knudsen	Technical University of Denmark
Arjen Lenstra	EPFL and Alcatel-Lucent Bell Laboratories
Helger Lipmaa	Cybernetica AS
Mitsuru Matsui	Mitsubishi Electric
Alexander May	Ruhr-University Bochum
Tatsuaki Okamoto	NTT
Krzysztof Pietrzak	CWI Amsterdam
David Pointcheval	ENS/CNRS/INRIA
Bart Preneel	Katholieke Universiteit Leuven
Phillip Rogaway	University of California, Davis
Amit Sahai	UCLA
Berry Schoenmakers	Technische Universiteit Eindhoven
Ron Steinfeld	Macquarie University
Frederik Vercauteren	Katholieke Universiteit Leuven
Yiqun Lisa Yin	Independent Security Consultant

External Reviewers

Michel Abdalla
Masayuki Abe
Shweta Agrawal
Martin Albrecht
Davide Alessio
Elena Andreeva
Giuseppe Ateniese
Roberto Avanzi
Ali Bagherzandi
Paulo Barreto
Anja Becker
Mihir Bellare
Rikke Bendlin
Nir Bitansky
Bruno Blanchet
Julia Borghoff
Joppe Bos
Arnaud Boscher
Ahto Buldas
Sébastien Canard
Christophe De Cannière
David Cash
Wouter Castryck
Pascale Charpin
Céline Chevalier
Cécile Delerablée
Alex Dent
Léo Ducas
Thomas Dullien
Orr Dunkelman
Sebastian Faust
Marc Fischlin
Matthias Fitzi
Georg Fuchsbauer
Teddy Furon
Sebastian Gajek
David Galindo
Nicolas Gama
Praveen Gauravaram
Sharon Goldberg
Louis Goubin
Aline Gouget
Vipul Goyal
Jens Groth
Shai Halevi
Mike Hamburg
Carmit Hazay
Brett Hemenway
Jens Hermans
Mathias Herrmann
Dennis Hofheinz
Susan Hohenberger
Sebastiaan de Hoogh
Fumitaka Hoshino
Thomas Icart
Sorina Ionica
Yuval Ishai
Hongxia Jin
Ellen Jochemsz
Pascal Junod
Marcelo Kaihara
Alexandre Karlov
Marcel Keller
John Kelsey
Shahram Khazaei
Eike Kiltz
Thorsten Kleinjung
Hugo Krawczyk
Eyal Kushilevitz
Tanja Lange
Gregor Leander
Reynald Lercier
Gaëtan Leurent
Allison Lewko
Peter van Liesdonk
Xiaomin Liu
Carolin Lunemann
Hemanta Maji
Yoshifumi Manabe
Krystian Matusiewicz
Alfred Menezes
Alexander Meurer
Lorenz Minder
Marine Minier
Hart Montgomery
Sean Murphy
María Naya-Plasencia
Gregory Neven
Phong Q. Nguyen
Jesper Buus Nielsen
Svetla Nikova
Ryo Nishimaki
Karsten Nohl
Adam O'Neill
Josh Olsen
Alina Oprea
Rafi Ostrovsky
Dag Arne Osvik
Onur Ozen
Carles Padró
Pascal Paillier
Omkant Pandey
Omer Paneth
Jacques Patarin
Kenny Paterson
Serdar Pehlivanoglu
Duong Hieu Phan
Josef Pieprzyk
Benny Pinkas
Zeger Plug
Bart Preneel
Emmanuel Prouff
Xavier Pujol
Tal Rabin
Alfredo Rial
Thomas Ristenpart
Maike Ritzenhofen
Ben Riva
Sondre Rønjom
Rei Safavi-Naini
Juraj Sarinay
Christian Schaffner
Gil Segev
Yannick Seurin
Hakan Seyalioglu
Stefaan Seys
Hovav Shacham
Daniel Shahaf
Igor Shparlinski
Koen Simoens
Dave Singelée
Boris Škorić

Nigel Smart
Ben Smith
Martijn Stam
Till Stegers
Damien Stehlé
Mario Strefler
Xiaoming Sun
Daisuke Suzuki
Katsuyuki Takashima
Stefano Tessaro
Kobayashi Tetsutaro
Søren S. Thomsen
Mehdi Tibouchi
Jean-Pierre Tillich
Tomas Toft
Jacques Traoré
Joana Treger
Elena Trichina
Toyohiro Tsurumaru
Pim Tuyls
Berkant Ustaoğlu
Vinod Vaikuntanathan
Margarita Vald
Mayank Varia
Serge Vaudenay
Bastien Vayssière
Damien Vergnaud
José Villegas
Ivan Visconti
Martin Vuagnoux
Shabsi Walfish
Huaxiong Wang
Brent Waters
Steve Weis
Christopher Wolf
Brecht Wyseur
Keita Xagawa
Go Yamamoto
Santiago Zanella Béguelin
Erik Zenner
Haibin Zhang
Hong-Sheng Zhou

Table of Contents

Cryptosystems I

Obfuscation and Side Channel Security

2-Party Protocols

Cryptanalysis

2010 IACR Distinguished Lecture

Automated Tools and Formal Methods

Models and Proofs

Multiparty Protocols

Cryptosystems II

Hash and MAC

Foundational Primitives

On Ideal Lattices and Learning with Errors over Rings

Vadim Lyubashevsky[1,⋆], Chris Peikert[2,⋆⋆], and Oded Regev[1,⋆⋆⋆]

[1] Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel
[2] School of Computer Science, College of Computing, Georgia Institute of Technology
cpeikert@cc.gatech.edu

Abstract. The "learning with errors" (LWE) problem is to distinguish random linear equations, which have been perturbed by a small amount of noise, from truly uniform ones. The problem has been shown to be as hard as worst-case lattice problems, and in recent years it has served as the foundation for a plethora of cryptographic applications. Unfortunately, these applications are rather inefficient due to an inherent quadratic overhead in the use of LWE. A main open question was whether LWE and its applications could be made truly efficient by exploiting extra algebraic structure, as was done for lattice-based hash functions (and related primitives).

We resolve this question in the affirmative by introducing an algebraic variant of LWE called *ring-LWE*, and proving that it too enjoys very strong hardness guarantees. Specifically, we show that the ring-LWE distribution is pseudorandom, assuming that worst-case problems on ideal lattices are hard for polynomial-time quantum algorithms. Applications include the first truly practical lattice-based public-key cryptosystem with an efficient security reduction; moreover, many of the other applications of LWE can be made much more efficient through the use of ring-LWE. Finally, the algebraic structure of ring-LWE might lead to new cryptographic applications previously not known to be based on LWE.

1 Introduction

Over the last decade, *lattices* have emerged as a very attractive foundation for cryptography. The appeal of lattice-based primitives stems from the fact that

[⋆] Supported by a European Research Council (ERC) Starting Grant. Part of this work was performed while visiting Georgia Tech.

[⋆⋆] This material is based upon work supported by the National Science Foundation under Grant CNS-0716786. 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 National Science Foundation.

[⋆⋆⋆] Supported by the Binational Science Foundation, by the Israel Science Foundation, by the European Commission under the Integrated Project QAP funded by the IST directorate as Contract Number 015848, by the Wolfson Family Charitable Trust, and by a European Research Council (ERC) Starting Grant.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 1–23, 2010.

their security can often be based on *worst-case* hardness assumptions, and that they appear to remain secure even against *quantum* computers.

Many lattice-based cryptographic schemes are based directly upon two natural average-case problems that have been shown to enjoy worst-case hardness guarantees. The *short integer solution* (SIS) problem was first shown in Ajtai's groundbreaking work [2] to be at least as hard as approximating several lattice problems, such as the (gap) shortest vector problem, to within a polynomial factor in the lattice dimension. More recently, Regev [31] defined the *learning with errors* (LWE) problem and proved that it enjoys similar worst-case hardness properties, under a quantum reduction. (That is, an efficient algorithm for LWE would imply an efficient quantum algorithm for approximate lattice problems.) Peikert [26] subsequently proved the hardness of LWE under certain lattice assumptions, via a classical reduction.

The SIS problem may be seen as a variant of subset-sum over a particular additive group. In more detail, let $n \geq 1$ be an integer dimension and $q \geq 2$ be an integer modulus; the problem is, given polynomially many random and independent $\mathbf{a}_i \in \mathbb{Z}_q^n$, to find a 'small' integer combination of them that sums to $\mathbf{0} \in \mathbb{Z}_q^n$. The LWE problem is closely related to SIS, and can be stated succinctly as the task of distinguishing 'noisy linear equations' from truly random ones. More specifically, the goal is to distinguish polynomially many pairs of the form $(\mathbf{a}_i, b_i \approx \langle \mathbf{a}_i, \mathbf{s} \rangle) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ from *uniformly random* and independent ones, where $\mathbf{s} \in \mathbb{Z}_q^n$ is a uniformly random secret (which is kept the same for all pairs), each $\mathbf{a}_i \in \mathbb{Z}_q^n$ is uniformly random and independent, and each inner product $\langle \mathbf{a}_i, \mathbf{s} \rangle \in \mathbb{Z}_q$ is perturbed by a fresh random error term that is relatively concentrated around 0 (modulo q).

In recent years, a multitude of cryptographic schemes have been proposed around the SIS and LWE problems. As a search problem (without unique solution), SIS has been the foundation for one-way [2] and collision-resistant hash functions [15], identification schemes [25, 18, 17], and digital signatures [13, 8]. The LWE problem has proved to be amazingly versatile for encryption schemes, serving as the basis for secure public-key encryption under both chosen-plaintext [31, 29] and chosen-ciphertext [30, 26] attacks, oblivious transfer [29], identity-based encryption [13, 8, 1], various forms of leakage-resilient encryption (e.g., [4, 6]), and more.

One drawback of schemes based on the SIS and LWE problems, however, is that they tend not to be efficient enough for practical applications. Even the simplest primitives, such as one-way functions, have key sizes at least *quadratic* in the primary security parameter, which needs to be in the several hundreds for sufficient security against the best known attacks.

A promising approach for avoiding this intrinsic inefficiency is to use lattices that possess extra algebraic structure. Influenced by the heuristic design of the NTRU cryptosystem [16], Micciancio [23] proposed a "compact," efficient one-way function using a ring-based variant of SIS that he showed to be at least as hard as worst-case problems on *cyclic lattices*. Later, Peikert and Rosen [27] and Lyubashevsky and Micciancio [20] independently constructed collision-resistant hash functions based on *ideal lattices* (a generalization of cyclic lattices), and provided a fast and practical implementation [22]. These results paved the way for other efficient cryptographic constructions, including identification schemes [19] and signatures [21, 19]. (The recent fully homomorphic cryptosystem of Gentry [12] is also based on ideal lattices, but it relies on new assumptions that are not related to SIS or LWE.)

Despite its expected utility, a compact analogue of LWE with comparable security properties has not yet appeared in the literature (though see Section 1.5 for discussion of a recent related work). Indeed, the perspectives and techniques that have so far been employed for the ring-SIS problem appear insufficient for adapting the more involved hardness proofs for LWE to the ring setting. Our main contributions in this paper are to define an appropriate version of the learning with errors problem in a wide class of rings, and to prove its hardness under worst-case assumptions on ideal lattices in these rings.

1.1 Informal Description of Results

Here we give an informal overview of the ring-LWE problem and our hardness results for it. For concreteness, this summary deals with one particular 'nice' ring, and deliberately omits the exact error distribution for which we can prove hardness. Our results actually apply much more generally to *rings of algebraic integers* in number fields, and the error distribution is defined precisely using concepts from algebraic number theory.

Let $f(x) = x^n + 1 \in \mathbb{Z}[x]$, where the security parameter n is a power of 2, making $f(x)$ irreducible over the rationals. (This particular $f(x)$ comes from the family of *cyclotomic* polynomials, which play a special role in this work.) Let $R = \mathbb{Z}[x]/\langle f(x)\rangle$ be the ring of integer polynomials modulo $f(x)$. Elements of R (i.e., residues mod $f(x)$) are typically represented by integer polynomials of degree less than n. Let $q = 1 \bmod 2n$ be a sufficiently large public prime modulus (bounded by a polynomial in n), and let $R_q = R/\langle q\rangle = \mathbb{Z}_q[x]/\langle f(x)\rangle$ be the ring of integer polynomials modulo both $f(x)$ and q. Elements of R_q may be represented by polynomials of degree less than n -whose coefficients are from $\{0, \ldots, q-1\}$.

In the above-described ring, the R-LWE problem may be described as follows. Let $s = s(x) \in R_q$ be a uniformly random ring element, which is kept secret. Analogously to standard LWE, the goal of the attacker is to distinguish arbitrarily many (independent) 'random noisy ring equations' from truly uniform ones. More specifically, the noisy equations are of the form $(a, b \approx a \cdot s) \in R_q \times R_q$, where a is uniformly random and the product $a \cdot s$ is perturbed by some 'small' random error term, chosen from a certain distribution over R.

Main Theorem 1 (Informal). *Suppose that it is hard for polynomial-time* quantum *algorithms to approximate the shortest vector problem (*SVP*) in the* worst case *on* ideal lattices[1] *in R to within a fixed* $\mathrm{poly}(n)$ *factor. Then any* $\mathrm{poly}(n)$ *number of*

[1] Briefly, an ideal lattice in R is just an ideal under some appropriate geometric embedding. See Section 1.3 for a precise definition and discussion.

samples drawn from the R-LWE distribution are pseudorandom to any polynomial-time (even quantum) attacker.

Our main theorem follows from two component results, which are each of independent interest.

Worst-case hardness of the search problem. We give a quantum reduction from approximate SVP (in the worst case) on ideal lattices in R to the *search* version of ring-LWE, where the goal is to *recover* the secret $s \in R_q$ (with high probability, for any s) from arbitrarily many noisy products. This result follows the general outline of Regev's iterative quantum reduction for general lattices [31], but ideal lattices introduce several new technical roadblocks in both the 'algebraic' and 'geometric' components of the reduction. We overcome these obstacles using perspectives and tools from algebraic number theory, in particular, the canonical embedding of a number field and the Chinese remainder theorem. Our result is stated formally as Theorem 1, and is proved throughout Section 3.

We point out that in contrast with standard LWE, the precise error distribution for which we can prove worst-case hardness is somewhat subtle: the distribution has up to n independent parameters (one for each direction in a certain orthogonal basis) which themselves are chosen at random and kept secret. Most cryptographic applications only require (for correctness) that the error distribution be relatively concentrated, so this form of noise generally presents no problem. (It is also possible show hardness for a fixed spherical distribution, but for a slightly super-polynomial approximation factor, modulus q, and reduction runtime.) The non-spherical error distribution is an artifact of our proof technique, and can perhaps be avoided using additional ideas.

Search / decision equivalence. We then show that the R-LWE distribution is in fact *pseudorandom* if the search problem is hard (given arbitrarily many samples). This result is also inspired by analogous reductions for the standard LWE problem [7, 31], but again the ring context presents new obstacles, primarily related to proving that the *entire* n-dimensional quantity $b \approx a \cdot s$ is simultaneously pseudorandom. Here again, the solution seems to rely inherently on tools from algebraic number theory. The full result is stated as Theorem 2, and is proved throughout Section 4.

We stress that our search/decision equivalence works for a wide class of natural noise distributions, and is entirely classical (no quantum). Therefore, it is of value even without our worst-case reduction, and can be understood independently of it. For example, if one makes the plausible conjecture that the search version of R-LWE is hard for a fixed spherical error distribution and small modulus q, then our proof demonstrates that the same R-LWE distribution is also pseudorandom.

1.2 Discussion and Applications

For cryptographic applications, the R-LWE problem has many attractive features. First note the cryptographic strength of R-LWE versus standard LWE (or, for that matter, any other common number-theoretic function): each noisy product $b \approx a \cdot s$ is a pseudorandom *n-dimensional* vector over $\mathbb{Z}_q$, rather than just a scalar, and we can generate as many of these values as we like. Yet the cost of generating them is quite small: polynomial multiplication can be performed in $O(n \log n)$ scalar operations using the Fast Fourier Transform (FFT). Moreover, the specific choice of polynomial $f(x) = x^n + 1$ and modulus $q = 1 \bmod 2n$ (among others) admits an optimized implementation that works entirely over the field $\mathbb{Z}_q$, and is very fast on modern architectures (see [22]). Finally, in most applications each sample $(a, b) \in R_q \times R_q$ from the R-LWE distribution can replace n samples $(\mathbf{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ from the standard LWE distribution, thus reducing the size of the public key (and often the secret key as well) by a $\Theta(n)$ factor. This is especially beneficial because key size has probably been the main barrier to practical lattice-based cryptosystems with rigorous security analysis.

Sample cryptosystem. As an example application, we exploit the pseudorandomness of the R-LWE distribution (e.g., over the ring $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ described above) to construct a simple semantically secure public-key cryptosystem. This scheme and its security proof are a direct translation of the 'dual' scheme from [13] based on the standard LWE problem, and similarly direct adaptations are possible for most other LWE-based schemes, including Regev's original 'primal' cryptosystem [31], Peikert's CCA-secure cryptosystem [26], and at least the identity-based encryption schemes of [13, 8].[2]

In our example cryptosystem, the key generation algorithm chooses $m \approx \lg q = O(\log n)$ uniformly random and independent elements $a_i \in R_q$, along with m random 'small' ring elements $r_i \in R$ (e.g., having uniformly random and independent 0-1 coefficients when viewed as polynomials). The element $a_{m+1} \in R_q$ is computed as $a_{m+1} = \sum_{i \in [m]} r_i \cdot a_i$. The public and secret keys, respectively, are the tuples

$$(a_1, \ldots, a_{m+1}) \in R_q^{m+1} \quad \text{and} \quad (r_1, \ldots, r_m, r_{m+1} = -1) \in R^{m+1}.$$

This key generation procedure has two main properties: first, the public key is essentially uniform (statistically) over R_q^{m+1}, which can be shown by a variant of the leftover hash lemma for the ring R_q [23]. Second, the public and secret keys satisfy $\sum_i r_i \cdot a_i = 0 \in R_q$.

To encrypt an n-bit message $z \in \{0,1\}^n$, view it as an element of R by using its bits as the 0-1 coefficients of a polynomial. Choose a uniformly random $s \in R_q$, and for each $i \in [m+1]$ compute $b_i \approx a_i \cdot s \in R_q$, where each product is perturbed by an independent 'small' error term $e_i \in R$ from the prescribed LWE error distribution. Lastly, subtract (modulo q) from b_{m+1} the ring element $z \cdot \lfloor q/2 \rceil$. The ciphertext is the tuple $(b_1, \ldots, b_{m+1}) \in R_q^{m+1}$. Note that semantic security is straightforward to prove, because the adversary's view, i.e., the public key and ciphertext, simply consists of $m+1$ samples from the pseudorandom R-LWE distribution, which hide the message.

To decrypt the ciphertext, simply compute

$$\sum r_i \cdot b_i \approx z \cdot \lfloor q/2 \rceil + \left(\sum r_i \cdot a_i \right) \cdot s = z \cdot \lfloor q/2 \rceil + 0 \cdot s \in R_q,$$

[2] Some of these constructions also require an adaptation of the basis-generation procedure of [5] to the ring setting, which was done in [32].

where the $\approx$ symbol hides $\sum_i r_i \cdot e_i \in R$, the error terms accumulated by the short elements from the secret key. For appropriate choices of parameters, the coefficients of this sum have magnitudes much smaller than $q/2$, so the bits of z can be recovered by rounding each coefficient back to either 0 or $\lfloor q/2 \rceil$, whichever is closest (mod q).

Security. Given the utility, flexibility, and efficiency of the ring-LWE problem, a natural question is: how plausible is the underlying assumption? All of the algebraic and algorithmic tools (including quantum computation) that we employ in our hardness reductions can also be brought to bear against SVP and other problems on ideal lattices. Yet despite much effort in this vein, we have been unable to make any significant progress in attacking these problems. The best known algorithms for ideal lattices perform essentially no better than their generic counterparts, which require exponential time and space to achieve a poly(n) approximation factor [3].

We also gain some confidence in the inherent hardness of ideal lattices from the fact that they arise (under a suitable definition; see Section 1.3 below) from a deep and well-studied branch of mathematics, which has also been investigated reasonably thoroughly from a computational point of view (see, e.g., [9]). Due to their recent application in the design of cryptographic schemes, however, it is probably still too early to say anything about their security with great confidence. Further study is certainly a very important research direction.

1.3 Ideal Lattices

Here we give a brief description of *ideal lattices*, survey their use in previous work, and compare to our work. All of the definitions of ideal lattices from prior work are instances of the following general notion: let R be a ring whose additive group is isomorphic to $\mathbb{Z}^n$ (i.e., it is a free $\mathbb{Z}$-module of rank n), and let σ be an additive isomorphism mapping R to some lattice $\sigma(R)$ in an n-dimensional real vector space (e.g., $\mathbb{R}^n$). The family of ideal lattices for the ring R under the embedding σ is the set of all lattices $\sigma(\mathcal{I})$, where $\mathcal{I}$ is an ideal in R.[3] For instance, taking $R = \mathbb{Z}[x]/\langle x^n - 1\rangle$ and the naïve "coefficient embedding" σ, i.e., the one that views the coefficients of a polynomial residue (modulo $x^n - 1$) as an integer vector in $\mathbb{Z}^n$, leads exactly to the family of (integer) cyclic lattices. Note that under the coefficient embedding, addition of ring elements simply corresponds to (coordinate-wise) addition of their vectors in $\mathbb{Z}^n$, but multiplication does not have such a nice geometrical interpretation, due to the reduction modulo $x^n - 1$.

The main difference between this work and almost all previous work is in the choice of embedding σ. Prior works [23, 27, 20, 21, 12, 19, 32] used rings of the form $\mathbb{Z}[x]/\langle f(x)\rangle$ with the coefficient embedding described above. In this work, following Peikert and Rosen [28], we instead consider the so-called *canonical embedding* from algebraic number theory. Strictly speaking, the coefficient and canonical embeddings are equivalent up to a fixed linear transformation that

[3] An ideal $\mathcal{I}$ in a ring R is an additive subgroup of R that is closed under multiplication by R.

introduces some distortion. (In fact, this is true of *any* two fixed embeddings, under our definition above.) Moreover, in many cases the distortion is small; for example, in the ring $\mathbb{Z}[x]/\langle x^n+1\rangle$ for n a power of 2, the transformation is even an isometry (i.e., a scaled rotation). In such cases, lattice problems are essentially equivalent under either embedding. Yet due to its central role in the study of number fields and useful geometric properties (explained below), we contend that the canonical embedding is the 'right' notion to use in the study of ideal lattices.

First, unlike the coefficient embedding, under the canonical embedding both addition *and* multiplication of ring elements are simply coordinate-wise. As a result, both operations have simple geometric interpretations leading to tight bounds, and probability distributions such as Gaussians behave very nicely under multiplication by fixed elements. In contrast, understanding the behavior of multiplication under the coefficient embedding required previous work to introduce notions like the *expansion factor*, which implicitly measures the distortion involved in going between the coefficient and canonical embeddings, but is not of much help for analyzing probability distributions. Second, although for many rings the two embeddings are nearly isometric, in many other rings of interest the distortion can be quite large — even *super-polynomial* in the dimension for some cyclotomic polynomial rings [11]. This may explain why we can prove tight hardness results for *all* such cyclotomic rings (as explained below), whereas previous work was mostly restricted to $\mathbb{Z}[x]/\langle x^n+1\rangle$ for n a power of 2 (and a few others). A third point in favor of the canonical embedding is that it also behaves very nicely under the automorphisms that we use in our search-to-decision reductions for ring-LWE.

Moving now to the choice of ring R, in this work our main focus is on the rings of integer polynomials modulo a *cyclotomic* polynomial.[4] From an algebraic point of view, it is more natural to view these rings as the rings of (algebraic) integers in cyclotomic number fields, and this is indeed the perspective we adopt. Moreover, our main theorem's first component (hardness of the search version of ring-LWE) applies generically to the ring of integers in *any* number field. Almost all previous work applied to rings of the form $\mathbb{Z}[x]/\langle f(x)\rangle$ for a monic irreducible $f(x)$ having small "expansion" (under the coefficient embedding mentioned above). This set of rings is incomparable to the set used in our work, although for some important examples like cyclotomics, our set is larger.

Rings of integers in number fields have some nice algebraic properties that are useful for our results. For instance, they have unique factorization of ideals, and their fractional ideals form a multiplicative group; in general, neither property holds in $\mathbb{Z}[x]/\langle f(x)\rangle$, even for monic and irreducible $f(x)$ (as demonstrated by the ring $\mathbb{Z}[x]/\langle x^2+3\rangle$). Another useful property is that certain number fields, such as the cyclotomic number fields used in our search/decision reduction, have *automorphisms* that 'shuffle' groups of related prime ideals while still preserving the LWE error distribution (when appropriately defined using the canonical embedding).

[4] The mth cyclotomic polynomial in $\mathbb{Z}[x]$ is the polynomial of degree $n=\varphi(m)$ whose roots are the primitive mth roots of unity ζ_m^i for $i\in\mathbb{Z}_m^*$, where $\zeta_m=\exp(2\pi i/m)$.

To summarize, while the number-theoretic perspective on ideal lattices requires some investment in the mathematical background, we find that it delivers many nice geometric and algebraic properties that pay dividends in the ease of working with the objects, and in the strength and generality of results that can be obtained.

1.4 Techniques

We introduce several new techniques for working with rings of integers and their ideal lattices, which fall into two broad categories: first, those that work in *general* number fields for reducing worst-case problems on ideal lattices to ring-LWE (and related problems); second, those that work in *cyclotomic* number fields, where we demonstrate a search/decision equivalence for ring-LWE and construct cryptographic schemes. All of the new techniques are entirely classical, i.e., non-quantum. (Our main reduction uses existing quantum technology essentially as a black box.)

In the category of worst-case reductions for ideal lattices, we show how to use the Chinese remainder theorem (CRT) for 'clearing the ideal' $\mathcal{I}$ from an arbitrary ideal lattice instance. This involves mapping the quotient ring $\mathcal{I}/q\mathcal{I}$ to the fixed quotient ring R/qR in an 'algebraically consistent' way. Our CRT techniques are also compatible with the 'discrete Gaussian' style of worst-to-average-case reduction from [13], which implies simpler and slightly tighter hardness proofs for ring-SIS. We remark that prior reductions following [23] work by restricting to a *principal subideal* of $\mathcal{I}$ with known generator; however, this technique does not seem to be compatible with the approaches of [31, 13], where the reduction must deal with Gaussian samples from the full ideal $\mathcal{I}$.

In our search/decision equivalence for ring-LWE, we also develop new techniques that exploit special properties of cyclotomic number fields of degree n — namely, that they are *Galois* (i.e., have n automorphisms) — and our particular choice of modulus q — namely, that it 'splits completely' into n prime ideals $\mathfrak{q}_i$ each of norm $q = \text{poly}(n)$, which are permuted by the automorphisms. (Interestingly, this complete splitting of q is also useful for performing the ring operations very efficiently in practice; see [22].)

The basic layout of our pseudorandomness proof is as follows: first, a hybrid argument shows that any distinguisher between the ring-LWE distribution $A_{s,\psi}$ and the uniform distribution must have some noticeable advantage relative to *some* prime ideal factor $\mathfrak{q}_i$ of $\langle q \rangle$ (of the distinguisher's choice); this advantage can be amplified using standard self-reduction techniques. Next, an efficient search-to-decision reduction finds the value of s modulo $\mathfrak{q}_i$, using the fact that the ring modulo $\mathfrak{q}_i$ is a field of order $q = \text{poly}(n)$. Then, because the automorphisms of the number field permute the $\mathfrak{q}_i$s, we can find s modulo *each* $\mathfrak{q}_j$ by applying an appropriate automorphism to the distribution $A_{s,\psi}$. (Crucially, the error distribution ψ also remains legal under this transformation). Finally, we recover all of $s \bmod q$ using the Chinese remainder theorem.

1.5 Related Work

In a concurrent and independent work, Stehlé, Steinfeld, Tanaka, and Xagawa [32] also formulated a variant of LWE over certain polynomial rings and proved its hardness under a worst-case (quantum) assumption. Their main application is a public-key cryptosystem with $\log^{O(1)} n$ encryption and decryption time per message bit. Due to the close similarities between our works, we wish to give a detailed comparison of the approaches and final outcomes.

Stehlé *et al.* [32] give a quantum reduction from the (average-case) ring-SIS problem to (average-case) ring-LWE, by exploiting the duality between the two problems and making new observations about Regev's quantum machinery [31]. Then, invoking prior worst-case hardness results for ring-SIS [20], they conclude that ring-LWE is hard under a worst-case quantum assumption. More precisely, they show that the *search* version of ring-LWE is hard for an *a priori bounded* number of samples with *spherical* Gaussian noise; however, the approach does not seem to extend to the *decision* version (i.e., pseudorandomness), nor to an unbounded number of samples.

The lack of pseudorandomness has some important drawbacks. For example, a primary motivation for the use of ring-LWE is to encrypt and decrypt faster than the most efficient cryptosystems based on standard LWE. In [32], achieving this goal requires *many* simultaneous hard-core bits for the search variant of ring-LWE, which are obtained via the efficient Goldreich-Levin construction using Toeplitz matrices [14, Section 2.5]. This approach, however, induces a security reduction that runs in time *exponential* in the number of hard bits. Therefore, to encrypt in amortized $\tilde{O}(1)$ time per message bit induces the assumption that ideal-SVP is hard for $2^{o(n)}$-time quantum algorithms. In contrast, our scheme has the same (actually somewhat better) running times under a fully polynomial assumption.

It is also worth noting that the main proof technique from [32], while quite transparent and modular, requires an *a priori* bound on the number of LWE samples consumed, and the modulus q and underlying approximation factor for ideal-SVP grow with this bound. This is suboptimal for cryptographic schemes (such as those in [30, 26, 6, 8]) that use a large (or even unbounded) number of samples in their security proofs. Moreover, having an unbounded number of samples seems essential for proving a search/decision equivalence for any type of LWE problem, because at the very least, the reduction needs to amplify the adversary's success probability.

2 Preliminaries

For a vector $\mathbf{x}$ in $\mathbb{R}^n$ or $\mathbb{C}^n$ and $p \in [1, \infty]$, we define the ℓ_p norm as $\|\mathbf{x}\|_p = (\sum_{i\in[n]} |x_i|^p)^{1/p}$ when $p < \infty$, and $\|\mathbf{x}\|_\infty = \max_{i\in[n]} |x_i|$ when $p = \infty$.

When working with number fields and ideal lattices, it is convenient to work with the space $H \subseteq \mathbb{R}^{s_1} \times \mathbb{C}^{2s_2}$ for some numbers $s_1 + 2s_2 = n$, defined as

$$H = \{(x_1, \ldots, x_n) \in \mathbb{R}^{s_1} \times \mathbb{C}^{2s_2} \ : \ x_{s_1+s_2+j} = \overline{x_{s_1+j}}, \ \forall\, j \in [s_2]\} \subseteq \mathbb{C}^n.$$

It is not difficult to verify that H (with the inner product induced on it by $\mathbb{C}^n$) is isomorphic to $\mathbb{R}^n$ as an inner product space. This can seen via the orthonormal basis $\{\mathbf{h}_i\}_{i\in[n]}$, defined as follows: for $j \in [n]$, let $\mathbf{e}_j \in \mathbb{C}^n$ be the vector with 1 in its jth (complex) coordinate, and 0 elsewhere. Then for $j \in [s_1]$, the basis vector $\mathbf{h}_j = \mathbf{e}_j \in \mathbb{C}^n$; for $s_1 < j \leq s_1 + s_2$, the vector $\mathbf{h}_j = \frac{1}{\sqrt{2}}(\mathbf{e}_j + \mathbf{e}_{j+s_2})$ and $\mathbf{h}_{j+s_2} = \frac{\sqrt{-1}}{\sqrt{2}}(\mathbf{e}_j - \mathbf{e}_{j+s_2})$. Note that the complex conjugation operation (which maps H to itself) acts in the $\{\mathbf{h}_i\}_{i\in[n]}$ basis by flipping the sign of all coordinates in $\{s_1 + s_2 + 1, \ldots, n\}$.

We will also equip H with the ℓ_p norm induced on it from $\mathbb{C}^n$. We note that for any $p \in [1, \infty]$, this norm is equal within a factor of $\sqrt{2}$ to the ℓ_p norm induced on H from the isomorphism with $\mathbb{R}^n$ described above, and that for the ℓ_2 norm we in fact have an equality. This (near) equivalence between H and $\mathbb{R}^n$ will allow us to use known definitions and results on lattices in our setting, the only caveat being the $\sqrt{2}$ factor when dealing with ℓ_p norms for $p \neq 2$.

2.1 Lattice Background

We define a *lattice* as a discrete additive subgroup of H. We deal here exclusively with full-rank lattices, which are generated as the set of all integer linear combinations of some set of n linearly independent *basis* vectors $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset H$.

The *minimum distance* $\lambda_1(\Lambda)$ of a lattice Λ in some norm $\|\cdot\|$ is the length of a shortest nonzero lattice vector: $\lambda_1(\Lambda) = \min_{\mathbf{0}\neq\mathbf{x}\in\Lambda}\|\mathbf{x}\|$. When left unspecified, the norm is taken to be the Euclidean norm; for the minimum distance of Λ in the ℓ_p norm, we write $\lambda_1^{(p)}(\Lambda)$.

The *dual lattice* of $\Lambda \subset H$ is defined as $\Lambda^* = \{\mathbf{x} \in H \,:\, \forall\, \mathbf{v} \in \Lambda, \langle \mathbf{x}, \mathbf{v}\rangle \in \mathbb{Z}\}$. It is easy to see that $(\Lambda^*)^* = \Lambda$.

Gaussian Measures. For $r > 0$, define the Gaussian function $\rho_r : H \to (0, 1]$ as $\rho_r(\mathbf{x}) = \exp(-\pi\langle\mathbf{x}, \mathbf{x}\rangle/r^2) = \exp(-\pi\|\mathbf{x}\|_2^2/r^2)$. By normalizing this function we obtain the *continuous* Gaussian probability distribution D_r of width r, whose density is given by $r^{-n} \cdot \rho_r(\mathbf{x})$. We extend this to elliptical (non-spherical) Gaussian distributions (in the basis $\{\mathbf{h}_i\}_{i\in[n]}$) as follows. Let $\mathbf{r} = (r_1, \ldots, r_n) \in (\mathbb{R}^+)^n$ be a vector of positive real numbers, such that $r_{j+s_1+s_2} = r_{j+s_1}$ for each $j \in [s_2]$. Then a sample from $D_{\mathbf{r}}$ is given by $\sum_{i\in[n]} x_i \cdot \mathbf{h}_i$, where the x_i are chosen independently from the (one-dimensional) Gaussian distribution D_{r_i} over $\mathbb{R}$.

Micciancio and Regev [24] introduced a lattice quantity called the *smoothing parameter*, and related it to various lattice quantities.

Definition 1. *For a lattice Λ and positive real $\epsilon > 0$, the* smoothing parameter *$\eta_\epsilon(\Lambda)$ is defined to be the smallest r such that $\rho_{1/r}(\Lambda^*\backslash\{\mathbf{0}\}) \leq \epsilon$.*

Lemma 1 ([24, Lemma 4.1] and [31, Claim 3.8]). *For any lattice Λ, $\epsilon > 0$, $r \geq \eta_\epsilon(\Lambda)$, and $\mathbf{c} \in H$, we have $\rho_r(\Lambda + \mathbf{c}) \in [\frac{1-\epsilon}{1+\epsilon}, 1] \cdot \rho_r(\Lambda)$.*

For a lattice Λ, point $\mathbf{u} \in H$, and real $r > 0$, define the *discrete Gaussian probability distribution over $\Lambda + \mathbf{u}$* with parameter r as the distribution assigning probability proportional to $\rho_r(\mathbf{x})$ to each $\mathbf{x} \in \Lambda + \mathbf{u}$.

We also need the following property of the smoothing parameter, which says that continuous noise 'smooths' the discrete structure of a discrete Gaussian distribution into a continuous one.

Lemma 2 ([31]). *Let Λ be a lattice, let $\mathbf{u} \in H$ be any vector, and let $r, s > 0$ be reals. Assume that $1/\sqrt{1/r^2 + 1/s^2} \geq \eta_\epsilon(\Lambda)$ for some $\epsilon < \frac{1}{2}$. Consider the continuous distribution Y on H obtained by sampling from $D_{\Lambda+\mathbf{u},r}$ and then adding an element drawn independently from D_s. Then the statistical distance between Y and $D_{\sqrt{r^2+s^2}}$ is at most 4ϵ.*

2.2 Algebraic Number Theory Background

Due to space constraints, we assume familiarity with the standard concepts of a number field K, its field trace Tr and norm N, and its ring of integers $\mathcal{O}_K$, discriminant Δ_K, and group of (fractional) ideals. Details may be found in the full version or in any introductory book on the subject, e.g., [33].

Embeddings and Geometry. Here we describe the *embeddings* of a number field, which induce a natural 'canonical' geometry on it.

A number field $K = \mathbb{Q}(\zeta)$ of degree n has exactly n field homomorphisms $\sigma_i : K \to \mathbb{C}$ that fix every element of $\mathbb{Q}$. Concretely, each embedding takes ζ to a different one of its conjugates; it can be verified that these are the only such field homomorphisms because the conjugates are the only roots of ζ's minimal polynomial $f(x)$. An embedding whose image lies in $\mathbb{R}$ (corresponding to a real root of f) is called a *real* embedding; otherwise (for a complex root of f) it is called a complex embedding. Because complex roots of $f(x)$ come in conjugate pairs, so too do the complex embeddings. The number of real and complex *pairs* of embeddings are denoted s_1 and s_2 respectively, so we have $n = s_1 + 2s_2$. The pair (s_1, s_2) is called the *signature* of K. By convention, we let $\{\sigma_j\}_{j\in[s_1]}$ be the real embeddings, and we order the complex embeddings so that $\sigma_{s_1+s_2+j} = \overline{\sigma_{s_1+j}}$ for $j \in [s_2]$. The *canonical embedding* $\sigma : K \to \mathbb{R}^{s_1} \times \mathbb{C}^{2s_2}$ is defined as $\sigma(x) = (\sigma_1(x), \ldots, \sigma_n(x))$. The canonical embedding σ is a field homomorphism from K to $\mathbb{R}^{s_1} \times \mathbb{C}^{2s_2}$, where multiplication and addition in $\mathbb{R}^{s_1} \times \mathbb{C}^{2s_2}$ are both component-wise. Due to the pairing of the complex embeddings, we have that σ maps into H.

By identifying elements K with their canonical embeddings in H, we can speak of geometric norms (e.g., the Euclidean norm) on K. Recalling that we define norms on H as those induced from $\mathbb{C}^n$, we see that for any $x \in K$ and any $p \in [1, \infty]$, the ℓ_p norm of x is simply $\|x\|_p = \|\sigma(x)\|_p = (\sum_{i\in[n]} |\sigma_i(x)|^p)^{1/p}$ for $p < \infty$, and is $\max_{i\in[n]} |\sigma_i(x)|$ for $p = \infty$. (As always, we assume the ℓ_2 norm when p is omitted.) Because multiplication of embedded elements is component-wise (since σ is a ring homomorphism), we have $\|x \cdot y\|_p \leq \|x\|_\infty \cdot \|y\|_p$ for any $x, y \in K$ and any $p \in [1, \infty]$. Thus the ℓ_∞ norm acts as an 'absolute value' for K that bounds how much an element 'expands' any other by multiplication.

Using the canonical embedding also allows us to think of the distribution $D_\mathbf{r}$ (for $\mathbf{r} \in (\mathbb{R}^+)^n$) over H as a distribution over K. Strictly speaking, the distribution $D_\mathbf{r}$ is not quite over K, but rather over the field $K_\mathbb{R} = K \otimes_\mathbb{Q} \mathbb{R}$, which, roughly speaking, is to K as $\mathbb{R}$ is to $\mathbb{Q}$. Since multiplication of elements in the number field is mapped to coordinate-wise multiplication, we get that for any element $x \in K$, the distribution of $x \cdot D_{\mathbf{r}}$ is $D_{\mathbf{r}'}$, where $r'_i = r_i \cdot |\sigma_i(x)|$ (this uses the fact that our distributions have the same variance in the complex and real components of each embedding).

Ideal Lattices. Here we recall how (fractional) ideals in K yield lattices under the canonical embedding, and describe some of their properties. Recall that a fractional ideal $\mathcal{I}$ has a $\mathbb{Z}$-basis $U = \{u_1, \ldots, u_n\}$. Therefore, under the canonical embedding σ, the ideal yields a rank-n *ideal lattice* having basis $\{\sigma(u_1), \ldots, \sigma(u_n)\} \subset H$. The fundamental volume of the ideal lattice $\sigma(\mathcal{I})$ is $|\det(\sigma(U))| = \mathrm{N}(\mathcal{I}) \cdot \sqrt{\Delta_K}$; as expected, this quantity is basis-invariant. For convenience, we often identity an ideal with its embedded lattice, and speak of, e.g., the minimum distance $\lambda_1(\mathcal{I})$ of an ideal, etc.

We now recall the notion of a dual ideal and explain its close connection to both the inverse ideal and the dual lattice. For more details, see, e.g., [10].

For a (fractional) ideal $\mathcal{I}$, its (fractional) *dual ideal* is defined as $\mathcal{I}^\vee = \{x \in K : \mathrm{Tr}(x\mathcal{I}) \subset \mathbb{Z}\}$. It is not difficult to see that, under the canonical embedding into H, the dual ideal embeds exactly as the complex conjugate of the dual lattice, i.e., $\sigma(I^\vee) = \overline{\sigma(I)^*}$. This is due to the fact that $\mathrm{Tr}(xy) = \sum_i \sigma_i(x)\sigma_i(y) = \langle \sigma(x), \overline{\sigma(y)} \rangle$.

Except in the trivial number field $K = \mathbb{Q}$, the ring of integers $\mathcal{O}_K$ is not self-dual, nor are an ideal and its inverse dual to each other. Fortunately, a useful and important fact is that an ideal and its inverse *are* equivalent up to multiplication by the dual ideal of the entire ring. That is, for any fractional ideal $\mathcal{I}$, its dual ideal is $\mathcal{I}^\vee = \mathcal{I}^{-1} \cdot \mathcal{O}_K^\vee$. (Notice that for $\mathcal{I} = \mathcal{O}_K$ this holds by definition, since $\mathcal{O}_K^{-1} = \mathcal{O}_K$.) The dual ideal $\mathcal{O}_K^\vee$ is itself sometimes called the *codifferent ideal*.

Chinese Remainder Theorem. Here we recall the Chinese remainder theorem (CRT) for the ring of integers in a number field, and some of its important consequences for this work. Let K be an arbitrary fixed number field and let $R = \mathcal{O}_K$ be its ring of integers.

Lemma 3 (Chinese remainder theorem). *Let $\mathcal{I}_1, \ldots, \mathcal{I}_r$ be pairwise coprime ideals in R, and let $\mathcal{I} = \prod_{i \in [r]} \mathcal{I}_i$. The natural ring homomorphism $C : R \to \oplus_{i \in [r]}(R/\mathcal{I}_i)$ induces a ring isomorphism $R/\mathcal{I} \to \bigoplus_{i \in [r]}(R/\mathcal{I}_i)$.*

We state the following important consequences of the CRT; proofs are given in the full version.

Lemma 4. *Let $\mathcal{I}$ and $\mathcal{J}$ be ideals in R. Then there exists $t \in \mathcal{I}$ such that the ideal $t \cdot \mathcal{I}^{-1} \subseteq R$ is coprime to $\mathcal{J}$. Moreover, such t can be found efficiently given $\mathcal{I}$ and the prime ideal factorization of $\mathcal{J}$.*

Lemma 5. *Let $\mathcal{I}$ and $\mathcal{J}$ be ideals in R, let $t \in \mathcal{I}$ be such that $t \cdot \mathcal{I}^{-1}$ is coprime with $\mathcal{J}$, and let $\mathcal{M}$ be any fractional ideal in K. Then the function $\theta_t : K \to K$ defined as $\theta_t(u) = t \cdot u$ induces an isomorphism from $\mathcal{M}/\mathcal{J}\mathcal{M}$ to $\mathcal{I}\mathcal{M}/\mathcal{I}\mathcal{J}\mathcal{M}$,* *as R-modules. Moreover, this isomorphism may be inverted efficiently given $\mathcal{I}$, $\mathcal{J}$, $\mathcal{M}$, and t.*

Other Properties of Cyclotomic Number Fields. Here we state a few more useful facts about cyclotomic number fields, which are used only in our search-to-decision reductions of Section 4.

Letting $K = \mathbb{Q}(\zeta)$ for $\zeta = \zeta_m$ be the mth cyclotomic number field, recall that $\mathcal{O}_K = \mathbb{Z}[\zeta]$. For an integer prime $q \in \mathbb{Z}$, the factorization of the ideal $\langle q \rangle$ is as follows. Let the factorization of $\Phi_m(x)$ modulo q (i.e., in $\mathbb{Z}_q[x]$) into monic irreducible polynomials be $\Phi_m(x) = \prod_i (f_i(x))^{e_i}$. Then in $\mathcal{O}_K$, the prime ideal factorization of $\langle q \rangle$ is $\langle q \rangle = \prod_i \mathfrak{q}_i^{e_i}$, where each $\mathfrak{q}_i = \langle q, f_i(\zeta) \rangle$ is a prime ideal with norm $q^{\deg(f_i)}$.[5]

For an integer prime $q = 1 \bmod m$, the field $\mathbb{Z}_q$ has a primitive mth root of unity r, because the multiplicative group of $\mathbb{Z}_q$ is cyclic with order $q-1$. Indeed, there are $n = \varphi(m)$ distinct such roots of unity $r^i \in \mathbb{Z}_q$, for $i \in \mathbb{Z}_m^*$. Therefore, the cyclotomic polynomial $\Phi_m(x)$ factors in $\mathbb{Z}_q[x]$ as $\Phi(x) = \prod_{i \in \mathbb{Z}_m^*}(x - r^i)$. The ideal $\langle q \rangle \subset \mathcal{O}_K$ then "splits completely" into n distinct prime ideals, as $\langle q \rangle = \prod_{i \in \mathbb{Z}_m^*} \mathfrak{q}_i$ where $\mathfrak{q}_i = \left\langle q, \zeta - r^i \right\rangle$ is prime and has norm q. (The fact that the ideal $\langle q \rangle$ splits into *distinct* prime ideals with *small* norm will be crucial in our search-to-decision for ring-LWE.)

We also need the fact that K has $n = \varphi(m)$ automorphisms $\tau_k : K \to K$ indexed by $k \in \mathbb{Z}_m^*$, which are defined by $\tau_k(\zeta) = \zeta^k$. The automorphisms τ_j "act transitively" on the prime ideals $\mathfrak{q}_i$, i.e., $\tau_j(\mathfrak{q}_i) = \mathfrak{q}_{i/j}$. This fact follows directly from the fact that cyclotomic number fields are *Galois extensions* of $\mathbb{Q}$.

Computation in Number Fields. All of the operations required by our reductions can be performed in polynomial time given a suitable representation of the number field and its ring of integers. Due to space constraints, we defer the details to the full version.

We now define some seemingly hard computational problems on ideal lattices in number fields.

Definition 2. *Let K be a number field endowed with some geometric norm (e.g., the ℓ_2 norm), and let $\gamma \geq 1$. The K-SVP_γ problem (in the given norm) is: given a (fractional) ideal $\mathcal{I}$ in K, find some nonzero $x \in \mathcal{I}$ such that $\|x\| \leq \gamma \cdot \lambda_1(\mathcal{I})$.*

Definition 3. *Let K be a number field endowed with some geometric norm (e.g., the ℓ_∞ norm), let $\mathcal{I}$ be a (fractional) ideal in K, and let $d < \lambda_1(\mathcal{I})/2$. The K-$\mathsf{BDD}_{\mathcal{I},d}$ problem (in the given norm) is: given $\mathcal{I}$ and y of the form $y = x + e$ for some $x \in \mathcal{I}$ and $\|e\| \leq d$, find x.*

Without loss of generality, both of the above problems may be restricted to *integral* ideals $\mathcal{I} \subseteq \mathcal{O}_K$, by the following scaling argument: if $\mathcal{I}$ is a fractional ideal with denominator $d \in \mathcal{O}_K$ (such that $d\mathcal{I} \subseteq \mathcal{O}_K$ is an integral ideal), then the scaled ideal $\mathrm{N}(d) \cdot \mathcal{I} \subseteq \mathcal{O}_K$, because $\mathrm{N}(d) \in \langle d \rangle$.

[5] In fact, this factorization holds in any 'monogenic' ring of integers $\mathcal{O}_K = \mathbb{Z}[\zeta]$, with $\Phi_m(x)$ replaced by the minimal polynomial of ζ.

2.3 The Ring-LWE Problem

Let K be a number field, let $R = \mathcal{O}_K$ be its ring of integers, let $R^\vee$ be its dual (codifferent) ideal, let $q \geq 2$ be a (rational) integer modulus, and let $R_q = R/qR$ and $R_q^\vee = R^\vee/qR^\vee$. Let $\mathbb{T} = K_\mathbb{R}/R^\vee$.

For an $s \in R_q^\vee$ and a distribution ψ over $K_\mathbb{R}$, the distribution $A_{s,\psi}$ over $R_q \times \mathbb{T}$ is generated by choosing $a \leftarrow R_q$ uniformly at random, choosing $e \leftarrow \psi$, and outputting $(a, (a \cdot s)/q + e)$, where addition in the second component is in $\mathbb{T}$ (i.e., modulo $R^\vee$).

Definition 4 (Learning with Errors in a Ring of Integers). *Let $q \geq 2$ be a (rational) integer and let Ψ be a family of distributions over $K_\mathbb{R}$. The* ring-LWE *problem in $R = \mathcal{O}_K$, denoted R-$\mathsf{LWE}_{q,\Psi}$, is defined as follows: given access to arbitrarily many independent samples from $A_{s,\psi}$ for some arbitrary $s \in R_q^\vee$ and $\psi \in \Psi$, find s.*

For an asymptotic treatment of the ring-LWE problem, we let K come from an infinite sequence of number fields $\mathcal{K} = \{K_n\}$ of increasing dimension n.

A natural question at this point is, why is the secret s chosen from the domain $R_q^\vee$ rather than R_q, as the values a are? From a purely *algebraic* perspective, it is possible to transform the ring-LWE distribution to make s come from the quotient ring $\mathcal{I}/q\mathcal{I}$ for any desired fractional ideal $\mathcal{I}$, making the choice of domain appear arbitrary. However, from a *geometric* perspective, such a transformation can in general introduce some distortion in the noise distribution. Upon close inspection, there are several reasons why $R_q^\vee$ is the most natural "canonical" domain for s; due to space constraints, we defer an explanation to the full version.

We now define the exact LWE error distributions for which our results apply. Informally, they are elliptical Gaussians whose widths along each axis (in the canonical embedding) are bounded by some parameter α.

Definition 5. *For a positive real $\alpha > 0$, the family $\Psi_{\leq\alpha}$ is the set of all elliptical Gaussian distributions $D_\mathbf{r}$ (over $K_\mathbb{R}$) where each parameter $r_i \leq \alpha$.*

In Section 4.1, we exploit a particular closure property for the family $\Psi_{\leq\alpha}$ over the mth cyclotomic number field $K = \mathbb{Q}(\zeta)$, where $\zeta = \zeta_m$. Let $\tau_j : K \to K$ be an automorphism of K, which is of the form $\tau_j(\zeta) = \zeta^j$ for some $j \in \mathbb{Z}_m^*$. Then $\Psi_{\leq\alpha}$ is closed under τ_j, i.e., for any $\psi = D_\mathbf{r} \in \Psi_{\leq\alpha}$, we have $\tau_j(D_\mathbf{r}) = D_{\mathbf{r}'} \in \Psi_{\leq\alpha}$, where the entries of $\mathbf{r}'$ are merely a rearrangement of the entries of $\mathbf{r}$ and hence are at most α.

3 Main Reduction

Since the results in this section apply to arbitrary number fields, we choose to present them in their most general form. For concreteness, the reader may wish to keep in mind the particular case of a cyclotomic number field.

Throughout this section, let K denote an arbitrary number field of degree n. We prove that solving the search problem $\mathcal{O}_K$-LWE (for a certain family of error distributions) is at least as hard as quantumly solving K-SVP_γ, where the approximation factor γ depends on the parameters of the error distributions.

3.1 Main Theorem and Proof Overview

The following is the main theorem of this section. Here, K-DGS_γ denotes the *discrete Gaussian sampling* problem [31], which asks, given an ideal $\mathcal{I}$ and a number $r \geq \gamma$, to produce samples from the distribution $D_{\mathcal{I},r}$. It is easy to show reductions from other more standard lattice problems such as SVP to DGS (see [31] for some examples).

Theorem 1. *Let K be an arbitrary number field of degree n. Let $\alpha = \alpha(n) \in (0, 1)$ be arbitrary, and let the (rational) integer modulus $q = q(n) \geq 2$ be such that $\alpha \cdot q \geq \omega(\sqrt{\log n})$. There is a probabilistic polynomial-time* quantum *reduction from K-DGS_γ to $\mathcal{O}_K$-$\mathsf{LWE}_{q,\Psi_{\leq\alpha}}$, where $\gamma = \eta_\epsilon(\mathcal{I}) \cdot \omega(\sqrt{\log n})/\alpha$.*

We prove the theorem by taking Regev's iterative reduction for general lattices [31] and replacing its core component (namely, the reduction from the bounded-distance decoding (BDD) problem to LWE) with an analogous statement for the ideal case (Lemma 7). It is here that we crucially apply algebraic techniques such as the Chinese remainder theorem, and we view this as one of our main contributions.

For self-containment, we describe now the main steps of the iterative reduction of [31], making the necessary changes for our setting. The reduction works by repeated application of an *iterative step*, which consists of the following two components.

1. The first component, which forms the core of [31], is a reduction from BDD on the dual lattice to LWE that uses Gaussian samples over the primal lattice. In Section 3.2 we show how to perform an analogous reduction in the ring setting. Namely, we show that given an oracle that generates samples from the discrete Gaussian distribution $D_{\mathcal{I},r}$ for some (not too small) $r > 0$, using an $\mathcal{O}_K$-$\mathsf{LWE}_{q,\Psi_{\leq\alpha}}$ oracle we can solve the BDD problem on the dual ideal $\mathcal{I}^\vee$ to within distance $d = \alpha \cdot q/r$ in the ℓ_∞ norm. From this it follows that with probability negligibly close to 1, we can also solve BDD on $\mathcal{I}^\vee$ where the unknown offset vector e is drawn from the distribution $D_{d'}$ for $d' = \alpha \cdot q/(r \cdot \omega(\sqrt{\log n}))$. The reason is that a sample from D_s has ℓ_∞ norm at most $s \cdot \omega(\sqrt{\log n})$, except with negligible probability.
2. The second step is quantum, and is essentially identical to the corresponding step in Regev's reduction [31, Lemma 3.14]. This step uses an oracle that with all but negligible probability solves the BDD problem on $\mathcal{I}^\vee$, where the offset vector e is chosen from the distribution $D_{d'}$. Using a quantum procedure, it is shown in [31] how to use such an oracle to produce samples from the discrete Gaussian distribution $D_{\mathcal{I},r'}$ for $r' = 1/(2d')$. The exact statement of [31, Lemma 3.14] is more specialized; to get the above statement, one has to observe that the procedure used to prove the lemma calls the oracle with offset vectors e chosen from $D_{d'}$, and that correctness is maintained even if the oracle errs with negligible probability over the choice of e.

Notice that when $\alpha \cdot q \geq \omega(\sqrt{\log n})$, we can choose $r' \leq r/2$ so that the output distribution $D_{\mathcal{I},r'}$ of Step 2 is half as wide as the input distribution $D_{\mathcal{I},r}$ of

Step 1. The value of r starts out exponentially large so that the samples for the first execution of Step 1 can be generated classically (see [31, Lemma 3.2]), then in later phases of the iteration they are produced by the quantum part. By iterating back and forth between the two procedures, we can sample from a progressively tighter distribution until we obtain a sample from $D_{\mathcal{I},r}$ for the (typically small) r given as input to the DGS problem.

3.2 Core Step: The BDD to LWE Reduction

We first observe that to solve BDD on an ideal $\mathcal{J}$, it suffices to find the solution modulo $q\mathcal{J}$. This is actually a special case of a lemma from [31], which gives a *lattice-preserving* reduction for BDD in general lattices. Because the reduction is lattice-preserving, it also applies to ideal lattices.

Definition 6. *The q-$\mathsf{BDD}_{\mathcal{J},d}$ problem (in any norm) is: given an instance y of $\mathsf{BDD}_{\mathcal{J},d}$ that has solution $x \in \mathcal{J}$, find $x \bmod q\mathcal{J}$.*

Lemma 6 (Special case of [31, Lemma 3.5]). *For any $q \geq 2$, there is a deterministic polynomial-time reduction from $\mathsf{BDD}_{\mathcal{J},d}$ (in any ℓ_p norm) to q-$\mathsf{BDD}_{\mathcal{J},d}$ (in the same norm).*

Lemma 7. *Let $\alpha \in (0,1)$, let $q \geq 2$ be a (rational) integer with known factorization, let $\mathcal{I}$ be an ideal in $R = \mathcal{O}_K$, and let $r \geq \sqrt{2}q \cdot \eta_\epsilon(\mathcal{I})$ for some negligible $\epsilon = \epsilon(n)$. Given an oracle for the discrete Gaussian distribution $D_{\mathcal{I},r}$, there is a probabilistic polynomial-time (classical) reduction from q-$\mathsf{BDD}_{\mathcal{I}^\vee,d}$ (in the ℓ_∞ norm) to R-$\mathsf{LWE}_{q,\Psi_{\leq\alpha}}$, where $d = \alpha q/(\sqrt{2}r)$.*

Note that the hypothesis that $\mathcal{I}$ is an integral ideal (in $\mathcal{O}_K$) is without loss of generality, by the scaling argument at the end of Section 2.2.

Proof. The high-level description of the reduction is as follows. Its input is a q-$\mathsf{BDD}_{\mathcal{I}^\vee,d}$ instance $y = x + e$ (where $x \in \mathcal{I}^\vee$ and $\|e\|_\infty \leq d$), and it is given access to an oracle that generates independent samples from the discrete Gaussian distribution $D_{\mathcal{I},r}$, and an oracle $\mathcal{L}$ that solves R-LWE. The reduction produces samples from the LWE distribution $A_{s,\psi}$, where the secret s *and* the error distribution ψ are related to x and e, respectively. Finally, given the solution s output by $\mathcal{L}$, the reduction recovers $x \bmod q\mathcal{I}^\vee$ from s.

In detail, the reduction does the following, given a q-$\mathsf{BDD}_{\mathcal{I}^\vee,d}$ instance y:

1. Compute an element $t \in \mathcal{I}$ such that $t \cdot \mathcal{I}^{-1}$ and $\langle q \rangle$ are coprime.
   (By Lemma 4, such t exists and can be found efficiently using the factorization of $\langle q \rangle$.)
2. For each sample requested by $\mathcal{L}$, get a fresh $z \leftarrow D_{\mathcal{I},r}$ from the Gaussian oracle and provide to $\mathcal{L}$ the pair (a,b), computed as follows: let $e' \leftarrow D_{\alpha/\sqrt{2}}$, and

   $$a = \theta_t^{-1}(z \bmod q\mathcal{I}) \in R_q \quad \text{and} \quad b = (z \cdot y)/q + e' \bmod R^\vee.$$

   (Recall that by Lemma 5 with $\mathcal{J} = \langle q \rangle$ and $\mathcal{M} = R$, the function $\theta_t(u) = t \cdot u$ induces a bijection from $R_q = R/qR$ to $\mathcal{I}/q\mathcal{I}$ which may be inverted efficiently given $\mathcal{I}$, q, and t.)

3. When $\mathcal{L}$ produces a solution $s \in R_q^\vee$, output $\theta_t^{-1}(s) \in \mathcal{I}^\vee/q\mathcal{I}^\vee$.
   (Again, by Lemma 5 with $\mathcal{J} = \langle q \rangle$ and $\mathcal{M} = \mathcal{I}^\vee = \mathcal{I}^{-1} \cdot R^\vee$, the function θ_t induces a bijection from $\mathcal{I}^\vee/q\mathcal{I}^\vee$ to $R_q^\vee = R^\vee/qR^\vee$, which may be inverted efficiently).

The correctness of the reduction follows from Lemma 8 below, which says that the samples (a, b) are distributed according to $A_{s,\psi}$ for $s = \theta_t(x \bmod q\mathcal{I}^\vee) \in R_q^\vee$ and some $\psi \in \Psi_{\leq\alpha}$. By hypothesis, $\mathcal{L}$ returns s, so the reduction outputs $\theta_t^{-1}(s) = x \bmod q\mathcal{I}^\vee$, which is the correct solution to its q-$\mathsf{BDD}_{\mathcal{I}^\vee,d}$ input instance.

Lemma 8. *Let y be the $\mathsf{BDD}_{\mathcal{I}^\vee,d}$ instance given to the reduction above, where $y = x + e$ for some $x \in \mathcal{I}^\vee$ and $\|e\|_\infty \leq d$. Each pair (a, b) produced by the reduction has distribution $A_{s,\psi}$ (up to negligible statistical distance), for $s = \theta_t(x \bmod q\mathcal{I}^\vee) = t \cdot x \in R_q^\vee$ and some $\psi \in \Psi_{\leq\alpha}$.*

Proof. We first show that in each output pair (a, b), the component $a \in R_q$ is 2ϵ-uniform. Because $r \geq q \cdot \eta_\epsilon(\mathcal{I})$, each sample z from $D_{\mathcal{I},r}$ is 2ϵ-uniform in $\mathcal{I}/q\mathcal{I}$ by Lemma 1. Then because θ_t induces a bijection from $R_q = R/qR$ to $\mathcal{I}/q\mathcal{I}$ by Lemma 5, $a = \theta_t^{-1}(z \bmod q\mathcal{I})$ is 2ϵ-uniform over R_q.

Now condition on any fixed value of a. We next analyze the component

$$b = (z \cdot y)/q + e' = (z \cdot x)/q + (z/q) \cdot e + e' \bmod R^\vee,$$

starting with $(z \cdot x)/q$. By definition of a, we have $z = \theta_t(a) = a \cdot t \in \mathcal{I}/q\mathcal{I}$. Because $x \in \mathcal{I}^\vee = \mathcal{I}^{-1} \cdot R^\vee$, we have

$$z \cdot x = \theta_t(a) \cdot x = a \cdot (t \cdot x) \bmod R_q^\vee.$$

Then because $s = t \cdot x \bmod R_q^\vee$, we have $z \cdot x = a \cdot s \bmod R_q^\vee$, which implies $(z \cdot x)/q = (a \cdot s)/q \bmod R^\vee$.

To analyze the remaining $(z/q) \cdot e + e'$ term, note that conditioned on the value of a, the random variable z/q has distribution $D_{\mathcal{I}+u/q,r/q}$, where $\mathcal{I} + u/q$ is a coset of $\mathcal{I}$ (specifically, $u = \theta_t(a) \bmod q\mathcal{I}$) and $r/q \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{I})$. Note that

$$(r/q) \cdot \|e\|_\infty \leq (r/q) \cdot d = \alpha/\sqrt{2},$$

so we may apply Lemma 9 below, which implies that the distribution of $(z/q) \cdot e + e'$ is within negligible statistical distance of the elliptical Gaussian $D_{\mathbf{r}}$, where each

$$r_i^2 = (r/q)^2 \cdot |\sigma_i(e)|^2 + (\alpha/\sqrt{2})^2 \leq (r/q)^2 \cdot d^2 + \alpha^2/2 = \alpha^2.$$

We conclude that each (a, b) is distributed as $A_{s,\psi}$ for some $\psi \in \Psi_{\leq\alpha}$, as desired.

Lemma 9. *Let $\mathcal{I}$ be a (fractional) ideal in K, and let $r \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{I})$ for some $\epsilon = \mathrm{negl}(n)$. Let $e \in K$ be fixed, let z be distributed as $D_{\mathcal{I}+v,r}$ for arbitrary $v \in K$, and let e' be distributed as $D_{r'}$ for some $r' \geq r \cdot \|e\|_\infty$. Then the distribution of $z \cdot e + e'$ is within negligible statistical distance of the elliptical Gaussian distribution $D_{\mathbf{r}}$ over $K_{\mathbb{R}}$, where $r_i^2 = r^2 \cdot |\sigma_i(e)|^2 + (r')^2$.*

Proof. We can write $z \cdot e + e'$ as $(z + e'/e) \cdot e$. The distribution of e'/e is the elliptical Gaussian $D_{\mathbf{t}}$, where each $t_i = r'/|\sigma_i(e)| \geq r'/\|e\|_\infty \geq r$. Thus e'/e can be written as the sum $f + g$ of independent f and g, where f has distribution D_r, and g has distribution $D_{\mathbf{t}'}$ where $(t_i')^2 = t_i^2 - r^2$.

Now by Lemma 2, the distribution of $z + f$ is negligibly close to $D_{\sqrt{2}r}$, so $(z + e'/e) = (z + f + g)$ has distribution $D_{\mathbf{t}''}$, where

$$(t_i'')^2 = 2r^2 + t_i^2 - r^2 = r^2 + (r')^2/|\sigma_i(e)|^2.$$

We conclude that $(z + e'/e) \cdot e$ has distribution $D_{\mathbf{r}}$, as desired.

4 Pseudorandomness of Ring-LWE

In this section we show that for an appropriate choice of ring, modulus, and error distribution, the ring-LWE distribution is pseudorandom. For concreteness and simplicity, we specialize the discussion to cyclotomic fields (though our techniques generalize somewhat to others). So throughout this section we assume that $\zeta = \zeta_m \in \mathbb{C}$ is a primitive mth root of unity, $K = \mathbb{Q}(\zeta)$ is the mth cyclotomic number field, $R = \mathcal{O}_K$ is its ring of integers, $R^\vee = \mathcal{O}_K^\vee$ is its dual (codifferent) ideal, and $q = 1 \bmod m$ is a poly(n)-bounded prime.

The main goal of this section is to show that the following average-case problem is hard. (Recall that $R_q = R/qR$, $R_q^\vee = R^\vee/qR^\vee$, and $\mathbb{T} = K_\mathbb{R}/R^\vee$.)

Definition 7 (Distinguishing LWE). *For a distribution Υ over a family of noise distributions (each over $K_\mathbb{R}$), we say that an algorithm solves the $\mathsf{DLWE}_{q,\Upsilon}$ problem if its acceptance probability given samples from $A_{s,\psi}$, over the random choice of $(s, \psi) \leftarrow U(R_q^\vee) \times \Upsilon$ and all other randomness of the experiment, differs by a non-negligible amount from its acceptance probability on uniformly random samples from $R_q \times \mathbb{T}$.*

The following is the main theorem of this section. It shows a reduction from the worst-case search variant of LWE (which by Theorem 1 is as hard as a worst-case lattice problem) to the above average-case problem. This establishes the hardness of the average-case problem, which means that the LWE distribution $A_{s,\psi}$ is itself pseudorandom when both s and the error distribution ψ are both chosen at random from appropriate distributions (and kept secret).

Theorem 2. *Let R, m, q be as above, and let $\alpha \cdot q \geq 1 \geq \eta_{2^{-n}}(R^\vee)$. Then there is a randomized polynomial-time reduction from $\mathsf{LWE}_{q,\Psi_{\leq\alpha}}$ to $\mathsf{DLWE}_{q,\Upsilon_\alpha}$.*

The proof of Theorem 2 goes through a chain of reductions, summarized in the following diagram (the numbers refer to lemma numbers, and the definitions of all intermediate problems are given later).

$$\begin{tikzcd}
\mathsf{LWE}_{q,\Psi} \arrow[r, "\boxed{10}", "\text{Automorphisms}"'] & \mathfrak{q}_i\text{-}\mathsf{LWE}_{q,\Psi} \arrow[r, "\boxed{11}", "\text{Search to Decision}"'] & \mathsf{DecLWE}^i_{q,\Psi} & & \\
{} \arrow[r, "\text{Worst to Average}"'] & \mathsf{DecLWE}^i_{q,\Upsilon} \arrow[r, "\text{Amplification}"'] & \mathsf{DLWE}^i_{q,\Upsilon} \arrow[r, "\text{Hybrid}"'] & \mathsf{DLWE}_{q,\Upsilon}
\end{tikzcd}$$

This sequence of reductions is similar in spirit to the one given in previous work on the (non-ideal) LWE problem [31]. However, there are a few important differences, requiring the introduction of new tools. One fundamental issue arising in the ring setting is that an oracle for DLWE might only let us deduce the value of the secret s relative to *one* ideal factor $\mathfrak{q}_i$ of $\langle q \rangle$. In order to recover the entire secret, we 'shuffle' the ideal factors using the field's automorphisms (see Lemma 10) to recover s relative to *every* factor $\mathfrak{q}_j$.

Another issue arises from the fact that the reduction in Section 3 only establishes the hardness of $\mathsf{LWE}_{q,\Psi}$ for a family of distributions Ψ that contains *non-spherical* Gaussian distributions. As a result, the average-case problem requires a distribution Υ over Gaussian noise distributions that are both non-spherical and wider by a factor of $\sqrt{n}$. Although this is somewhat undesirable, we do not yet see any way to avoid it; luckily, this only has a minor effect on the resulting cryptographic applications, i.e., adding an extra step of choosing the noise parameters. Let us mention, though, that if one is willing to assume that (the search problem) $\mathsf{LWE}_{q,\Psi}$ is hard with *spherical* Gaussian noise distributions, then we can fix the noise distribution in all the average-case problems (so there is no need to use a distribution over noise distributions Υ) and we do not need to lose the factor $\sqrt{n}$.

Due to space constraints, here we present only the first two steps of the chain of reductions, which contain the bulk of the novel ideas. The rest can be found in the full version.

4.1 Worst-Case Search to Worst-Case Decision

In this subsection we reduce the search version of $\mathsf{LWE}_{q,\Psi}$ to a certain decision problem over just *one arbitrary* prime ideal $\mathfrak{q}_i$. All of the problems considered here are *worst-case* over the choice of $s \in R_q^\vee$ and error distribution $\psi \in \Psi$, where Ψ is a family of allowed error distributions (though the actual error terms drawn from ψ are still random), and require their solutions to be found with overwhelming probability (over all the randomness of the experiment).

We first define some intermediate computational problems and probability distributions, then present two reductions. For notational convenience, we identify the elements of $\mathbb{Z}_m^*$ with their integer representatives from the set $\{1, \ldots, m-1\}$, with the usual ordering. For $i \in \mathbb{Z}_m^*$ we let $i-$ denote the largest element in $\mathbb{Z}_m^*$ less than i, defining $1-$ to be 0.

We define the notation $R_{\mathfrak{q}_i}^\vee = R^\vee/\mathfrak{q}_i R^\vee$, and note that by Lemmas 3 and 5, there is an efficiently computable R-module isomorphism between $R_q^\vee$ and $\bigoplus_i R_{\mathfrak{q}_i}^\vee$.

Definition 8 (LWE over $\mathfrak{q}_i$). *The $\mathfrak{q}_i$-$\mathsf{LWE}_{q,\Psi}$ problem is: given access to $A_{s,\psi}$ for some arbitrary $s \in R_q^\vee$ and $\psi \in \Psi$, find $s \in R_{\mathfrak{q}_i}^\vee$.*

Definition 9 (Hybrid LWE distribution). *For $i \in \mathbb{Z}_m^*$, $s \in R_q^\vee$, and a distribution ψ over K, the distribution $A_{s,\psi}^i$ over $R_q \times \mathbb{T}$ is defined as follows: choose $(a, b) \leftarrow A_{s,\psi}$ and output $(a, b + r/q)$ where $r \in R_q^\vee$ is uniformly random and independent in $R_{\mathfrak{q}_j}^\vee$ for all $j \le i$, and is 0 in all the remaining $R_{\mathfrak{q}_j}^\vee$. Also define $A_{s,\psi}^0$ simply as $A_{s,\psi}$.*

Definition 10 (Decision LWE relative to $\mathfrak{q}_i$). *For $i \in \mathbb{Z}_m^*$ and a family of distributions Ψ, the $\mathsf{DecLWE}_{q,\Psi}^i$ problem is defined as follows. Given access to $A_{s,\psi}^j$ for arbitrary $s \in R_q^\vee$, $\psi \in \Psi$, and $j \in \{i-, i\}$, find j.*

Claim. For any $i \in \mathbb{Z}_m^*$ there exists an efficient procedure that transforms $A_{s,\psi}^k$ (for any unknown $k \in \mathbb{Z}_m^* \cup \{0\}$, s, and ψ) into $A_{s,\psi}^{\max\{i,k\}}$.

Lemma 10 (LWE to $\mathfrak{q}_i$-LWE). *Suppose that the family Ψ is closed under all the automorphisms of K, i.e., $\psi \in \Psi \Rightarrow \tau_k(\psi) \in \Psi$ for every $k \in \mathbb{Z}_m^*$. Then for every $i \in \mathbb{Z}_m^*$, there is a deterministic polynomial-time reduction from $\mathsf{LWE}_{q,\Psi}$ to $\mathfrak{q}_i$-$\mathsf{LWE}_{q,\Psi}$.*

Proof. To compute $s \in R_{\mathfrak{q}_j}^\vee$, we use the oracle for $\mathfrak{q}_i$-LWE along with the field automorphisms τ_k to recover the value $s \in R_{\mathfrak{q}_j}^\vee$ for *all* $j \in \mathbb{Z}_m^*$. We can then efficiently reconstruct $s \in R_q^\vee$.

The reduction that finds $s \in R_{\mathfrak{q}_j}^\vee$ works as follows: transform each sample $(a, b) \leftarrow A_{s,\psi}$ into the sample $(\tau_k(a), \tau_k(b))$, where $k = j/i \in \mathbb{Z}_m^*$ and hence $\tau_k(\mathfrak{q}_j) = \mathfrak{q}_i$. (Also note that because τ_k is an automorphism on K and R is the set of all its algebraic integers, $\tau_k(R) = R$ and $\tau_k(R^\vee) = R^\vee$.) Give the transformed samples to the $\mathfrak{q}_i$-$\mathsf{LWE}_{q,\Psi}$ oracle, and when the oracle returns some element $t \in R_{\mathfrak{q}_i}^\vee$, return $\tau_k^{-1}(t) \in R_{\mathfrak{q}_j}^\vee$.

We now prove that $\tau_k^{-1}(t) = s \in R_{\mathfrak{q}_j}^\vee$. For each sample (a, b) from $A_{s,\psi}$, notice that because $b = as/q + e$ and $\tau_k(q) = q$, we have

$$\tau_k(b) = \tau_k(a)\tau_k(s)/q + \tau_k(e).$$

Because τ_k is an automorphism on R, $\tau_k(a) \in R_q$ is uniformly random, and because $\psi' = \tau_k(\psi) \in \Psi$, the pairs $(\tau_k(a), \tau_k(b))$ are distributed according to $A_{\tau_k(s),\psi'}$. By hypothesis, the oracle returns $t = \tau_k(s) \in R_{\mathfrak{q}_i}^\vee$. Thus $\tau_k^{-1}(t) = s \in \tau_k^{-1}(R_{\mathfrak{q}_i}^\vee) = R_{\mathfrak{q}_j}^\vee$.

Lemma 11 (Search to Decision). *For any $i \in \mathbb{Z}_m^*$, there is a probabilistic polynomial-time reduction from $\mathfrak{q}_i$-$\mathsf{LWE}_{q,\Psi}$ to $\mathsf{DecLWE}_{q,\Psi}^i$.*

Proof. The idea for recovering $s \in R_{\mathfrak{q}_i}^\vee$ is to try each of its possible values, modifying the samples we receive from $A_{s,\psi}$ so that on the correct value the modified samples are distributed according to $A_{s,\psi}^{i-}$, whereas on all the other values the modified samples are distributed according to $A_{s,\psi}^i$. We can then use the $\mathsf{DecLWE}_{q,\Psi}^i$ oracle to tell us which distribution was generated. Because there are only $\mathrm{N}(\mathfrak{q}_i) = q = \mathrm{poly}(n)$ possible values for $s \in R_{\mathfrak{q}_i}^\vee$, we can enumerate over all of them efficiently and discover the correct value.

First note that by Claim 4.1, we can transform our input distribution $A_{s,\psi}$ to $A_{s,\psi}^{i-}$. We now give the transformation that takes some $g \in R_q^\vee$ and maps $A_{s,\psi}^{i-}$ to either $A_{s,\psi}^{i-}$ or $A_{s,\psi}^i$, depending on whether or not $g = s \in R_{\mathfrak{q}_i}^\vee$ (its values in the other $R_{\mathfrak{q}_j}^\vee$ are irrelevant). Given a sample $(a, b) \leftarrow A_{s,\psi}^{i-}$, the transformation produces a sample

$$(a', b') = (a + v, b + vg/q) \in R_q \times \mathbb{T},$$

where $v \in R_q$ is uniformly random modulo $\mathfrak{q}_i$ and is 0 modulo the other $\mathfrak{q}_j$. First, notice that since a is uniformly distributed in R_q, so is a'. Next, condition on any fixed value of a'. Then b' can be written as

$$\begin{aligned} b' = b + vg/q &= (as + r)/q + e + vg/q \\ &= (a's + v(g - s) + r)/q + e, \end{aligned}$$

where e is chosen from ψ, and r is distributed as in the definition of $A^{i-}_{s,\psi}$, i.e., it is uniformly random and independent modulo $\mathfrak{q}_j$ for all $j < i$, and is 0 modulo all the remaining $\mathfrak{q}_j$.

We consider two cases. First, assume that $g = s \in R^\vee_{\mathfrak{q}_i}$. Then by the Chinese remainder theorem (Lemma 3), $v(g - s) = 0 \in R^\vee_q$, and hence the distribution of (a', b') is exactly $A^{i-}_{s,\psi}$. Next, assume that $g \neq s \bmod \mathfrak{q}_i$. Then since $\mathfrak{q}_i$ is a maximal ideal (which in R is equivalent to being a prime ideal), $R^\vee_{\mathfrak{q}_i}$ is a field, and hence $v(g - s)$ is distributed uniformly in $R^\vee_{\mathfrak{q}_i}$ (and is zero in all other $R^\vee_{\mathfrak{q}_j}$). From this it follows that $v(g - s) + r$ is distributed uniformly random and independently in $R^\vee_{\mathfrak{q}_j}$ for all $j \leq i$, and is 0 in all the remaining $R^\vee_{\mathfrak{q}_j}$. Hence, the distribution of (a', b') is exactly $A^{i}_{s,\psi}$, as promised.

References

[1] Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: EUROCRYPT (to appear, 2010)

[2] Ajtai, M.: Generating hard instances of lattice problems. Quaderni di Matematica 13, 1–32 (2004); Preliminary version in STOC 1996

[3] Ajtai, M., Kumar, R., Sivakumar, D.: A sieve algorithm for the shortest lattice vector problem. In: STOC, pp. 601–610 (2001)

[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] Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. In: STACS, pp. 75–86 (2009)

[6] Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: Halevi, S. (ed.) Advances in Cryptology - CRYPTO 2009. LNCS, vol. 5677, pp. 595–618. Springer, Heidelberg (2009)

[7] Blum, A., Furst, M.L., Kearns, M.J., Lipton, R.J.: Cryptographic primitives based on hard learning problems. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 278–291. Springer, Heidelberg (1994)

[8] Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: EUROCRYPT (to appear, 2010)

[9] Cohen, H.: A Course in Computational Algebraic Number Theory. Springer, Heidelberg (1993)

[10] Conrad, K.: The different ideal (2009), http://www.math.uconn.edu/~kconrad/blurbs/ (last accessed October 12, 2009)

[11] Erdös, P.: On the coefficients of the cyclotomic polynomial. Bulletin of the American Mathematical Society 52(2), 179–184 (1946)

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

[13] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC, pp. 197–206 (2008)

[14] Goldreich, O.: Foundations of Cryptography, vol. II. Cambridge University Press, Cambridge (2004)

[15] Goldreich, O., Goldwasser, S., Halevi, S.: Collision-free hashing from lattice problems. Electronic Colloquium on Computational Complexity (ECCC) 3(42) (1996)

[16] 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)

[17] Kawachi, A., Tanaka, K., Xagawa, K.: Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 372–389. Springer, Heidelberg (2008)

[18] Lyubashevsky, V.: Lattice-based identification schemes secure under active attacks. In: Cramer, R. (ed.) PKC 2008. LNCS, vol. 4939, pp. 162–179. Springer, Heidelberg (2008)

[19] Lyubashevsky, V.: Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 598–616. Springer, Heidelberg (2009)

[20] 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)

[21] 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)

[22] Lyubashevsky, V., Micciancio, D., Peikert, C., Rosen, A.: SWIFFT: A modest proposal for FFT hashing. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 54–72. Springer, Heidelberg (2008)

[23] Micciancio, D.: Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. Computational Complexity 16(4), 365–411 (2007); Preliminary version in FOCS 2002

[24] Micciancio, D., Regev, O.: Worst-case to average-case reductions based on Gaussian measures. SIAM J. Comput. 37(1), 267–302 (2007); Preliminary version in FOCS 2004

[25] Micciancio, D., Vadhan, S.P.: Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 282–298. Springer, Heidelberg (2003)

[26] Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem. In: STOC, pp. 333–342 (2009)

[27] 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)

[28] Peikert, C., Rosen, A.: Lattices that admit logarithmic worst-case to average-case connection factors. In: STOC, pp. 478–487 (2007)

[29] 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)

[30] Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: STOC, pp. 187–196 (2008)

[31] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM 56(6) (2009); Preliminary version in STOC 2005

[32] 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)

[33] Stein, W.: A brief introduction to classical and adelic algebraic number theory (2004), http://modular.math.washington.edu/papers/ant/ (last accessed October 12, 2009)

Fully Homomorphic Encryption over the Integers

Marten van Dijk[1], Craig Gentry[2], Shai Halevi[2], and Vinod Vaikuntanathan[2]

[1] MIT CSAIL
[2] IBM Research

Abstract. We construct a simple fully homomorphic encryption scheme, using only elementary modular arithmetic. We use Gentry's technique to construct a fully homomorphic scheme from a "bootstrappable" somewhat homomorphic scheme. However, instead of using ideal lattices over a polynomial ring, our bootstrappable encryption scheme merely uses addition and multiplication over the integers. The main appeal of our scheme is the conceptual simplicity.

We reduce the security of our scheme to finding an approximate integer gcd – i.e., given a list of integers that are near-multiples of a hidden integer, output that hidden integer. We investigate the hardness of this task, building on earlier work of Howgrave-Graham.

1 Introduction

What is the simplest encryption scheme for which one can hope to achieve security? The Caesar cipher is simple, but not secure. We believe that conventional public-key encryption schemes with modular exponentiations are secure, but modular exponentiation is not a very simple operation. If we were to forget our current schemes and start from scratch, perhaps something like the following scheme would be a good candidate for a simple symmetric encryption scheme:

KeyGen**:** The key is an odd integer, chosen from some interval $p \in [2^{\eta-1}, 2^{\eta})$.
Encrypt(p, m)**:** To encrypt a bit $m \in \{0,1\}$, set the ciphertext as an integer whose residue mod p has the same parity as the plaintext. Namely, set $c = pq + 2r + m$, where the integers q, r are chosen at random in some other prescribed intervals, such that $2r$ is smaller than $p/2$ in absolute value.
Decrypt(p, c)**:** Output $(c \bmod p) \bmod 2$.

It is easy to see that when the noise r is sufficiently smaller than the secret key p, this simple scheme is both additively and multiplicatively homomorphic for shallow arithmetic circuits. Moreover, one can use Gentry's techniques [6] (i.e., "bootstrapping" and "squashing the decryption circuit") to morph this scheme into a fully homomorphic encryption scheme [20]. Amazingly, it seems that with judicious choice of parameters (say $r \approx 2^{\sqrt{\eta}}$ and $q \approx 2^{\eta^3}$), this simple scheme may even be secure!!

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 24–43, 2010.

So far we only described a symmetric scheme, but turning it into a public key scheme is easy: The public key consists of many "encryptions of zero", namely integers $x_i = q_i \cdot p + 2r_i$ where q_i, r_i are chosen from the same prescribed intervals as above. Then to encrypt a bit m, the ciphertext is essentially set as m plus a subset sum of the x_i's.

We reduce the security of this scheme to approximate integer gcd – roughly, that it is hard to recover p from the x_i's. This problem, for the case of two x_i's, was analyzed by Howgrave-Graham [9]. Our parameters – in particular, the large size of the q_i's – are designed to avoid a generalized version of his attack (as well as other attack avenues, such as solving the associated simultaneous Diophantine approximation problem).

We comment that our scheme is similar to Regev's first encryption scheme [19]. In fact, a slight variation of Regev's scheme can be described by exactly the same formula as ours, $\mathsf{Enc}(m, p) = qp + 2r + m$. The main difference between the schemes is that in order to get the homomorphic properties, our choice of parameters is much more aggressive than his. Another difference is that the secret key p in our scheme is an integer, whereas in Regev's scheme the secret key is chosen as an integral fraction of the domain size (i.e., $p = N/h$ for some integer h). Unfortunately, Regev's worst-to-average-case security reductions from [19] do not seem to apply to our scheme.

2 Preliminaries

Below we usually denote parameters by Greek letters (e.g., η, γ, τ, etc.), with λ always denoting the security parameter. Real numbers and integers are denoted by lowercase English letters (p, q, x, y, etc.). All logarithms in the text are base-2 unless stated otherwise.

For a real number z, we denote by $\lceil z \rceil$, $\lfloor z \rfloor$, $\lfloor z \rceil$ the rounding of a up, down, or to the nearest integer. Namely, these are the unique integers in the half open intervals $[z, z+1)$, $(z-1, z]$, and $(z - \frac{1}{2}, z + \frac{1}{2}]$, respectively.

For a real number z and an integer p, we use $q_p(z)$ and $r_p(z)$ to denote the quotient and remainder of z with respect to p, namely $q_p(z) \stackrel{\text{def}}{=} \lfloor z/p \rceil$ and $r_p(z) \stackrel{\text{def}}{=} z - q_p(z) \cdot p$. (Note that $r_p(z) \in (-p/2, p/2]$.) We also denote the remainder by $[z]_p$ or $(z \bmod p)$, we use these three notations interchangeably throughout the paper.

A family $\mathcal{H}$ of hash functions from X to Y, both finite sets, is said to be 2-universal if for all distinct $x, x' \in X$, $\Pr_{h \stackrel{\text{R}}{\leftarrow} \mathcal{H}}[h(x) = h(x')] = 1/|Y|$. A distribution D is ϵ-uniform if its statistical distance from the uniform distribution is at most ϵ, where the statistical difference between two distributions D_1, D_2 over a finite domain X is $\frac{1}{2}\sum_{x \in X} |D_1(x) - D_2(x)|$.

Lemma 1 (Simplified Leftover Hash Lemma [8]). *Let $\mathcal{H}$ be a family of 2-universal hash functions from X to Y. Suppose that $h \stackrel{\text{R}}{\leftarrow} \mathcal{H}$ and $x \stackrel{\text{R}}{\leftarrow} X$ are chosen uniformly and independently. Then, $(h, h(x))$ is $\frac{1}{2}\sqrt{|Y|/|X|}$-uniform over $\mathcal{H} \times Y$.*

2.1 Homomorphic Encryption

Our definitions are adapted from Gentry [6]. Below we only consider encryption schemes that are homomorphic with respect to boolean circuits consisting of gates for addition and multiplication mod 2. (Considering only bit operations also means that the plaintext space of the encryption schemes that we consider is limited to $\{0,1\}$.) See the works of Ishai and Paskin [10] for a more general definitional treatment of homomorphic encryption with respect to other forms of "programs."

A homomorphic public key encryption scheme $\mathcal{E}$ has four algorithms: the usual KeyGen, $\mathsf{Encrypt}$, and $\mathsf{Decrypt}$, and an additional algorithm $\mathsf{Evaluate}$. The algorithm $\mathsf{Evaluate}$ takes as input a public key pk, a circuit C, a tuple of ciphertexts $\boldsymbol{c} = \langle c_1, \ldots, c_t \rangle$ (one for every input bit of C), and outputs another ciphertext c.

Definition 1 (Correct Homomorphic Decryption). *The scheme* $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Evaluate})$ *is* correct *for a given t-input circuit C if, for any key-pair* (sk, pk) *output by* $\mathsf{KeyGen}(\lambda)$*, any t plaintext bits $m_1, \ldots, m_t$, and any ciphertexts $\boldsymbol{c} = \langle c_1, \ldots, c_t \rangle$ with* $c_i \leftarrow \mathsf{Encrypt}_{\mathcal{E}}(\mathrm{pk}, m_i)$*, it is the case that:*

$$\mathsf{Decrypt}\,(\mathrm{sk},\ \mathsf{Evaluate}(\mathrm{pk}, C, \boldsymbol{c})) = C(m_1, \ldots, m_t)$$

Definition 2 (Homomorphic Encryption). *The scheme* $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Evaluate})$ *is homomorphic for a class $\mathcal{C}$ of circuits*[1] *if it is correct for all circuits $C \in \mathcal{C}$. $\mathcal{E}$ is* fully homomorphic *if it is correct for all boolean circuits.*

The semantic security of a homomorphic encryption scheme is defined in the usual way [7], without reference to the $\mathsf{Evaluate}$ algorithm. (Indeed $\mathsf{Evaluate}$ is a public algorithm with no secrets.)

It is clear that as defined above, fully homomorphic encryption can be trivially realized from any secure encryption scheme, by an algorithm $\mathsf{Evaluate}$ that simply attaches a description of the circuit C to the ciphertext tuple, and a $\mathsf{Decrypt}$ procedure that first decrypts all the ciphertexts and then evaluates C on the corresponding plaintext bits. Two properties of homomorphic encryption that rule out this trivial solution are *circuit-privacy* and *compactness.*

Circuit privacy roughly means that the ciphertext generated by $\mathsf{Evaluate}$ does not reveal anything about the circuit that it evaluates beyond the output value of that circuit, even for someone who knows the secret key. We discuss circuit privacy in the full version [21]. It is folklore that circuit-private fully-homomorphic encryption can be realized using Yao's "garbled circuits" [22,15] and a two-flow oblivious transfer protocol. (This construction is similar to the trivial solution from above, essentially it replaces the plaintext circuit with a garbled circuit.) Hence the "real challenge" in constructing fully homomorphic encryption comes from the compactness property, which essentially means that the size of the ciphertext that $\mathsf{Evaluate}$ generates does not depend on the size of the circuit C.

Definition 3 (Compact Homomorphic Encryption). *The scheme* $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Evaluate})$ *is* compact *if there exists a fixed polynomial*

[1] Formally, $\mathcal{C}$ is an ensemble, parametrized by the security parameter.

bound $b(\lambda)$ *so that for any key-pair* $(\mathrm{sk}, \mathrm{pk})$ *output by* $\mathsf{KeyGen}(\lambda)$*, any circuit* C *and any sequence of ciphertext* $\boldsymbol{c} = \langle c_1, \ldots, c_t \rangle$ *that was generated with respect to* pk*, the size of the ciphertext* $\mathsf{Evaluate}(\mathrm{pk}, C, \boldsymbol{c})$ *is not more than* $b(\lambda)$ *bits (independently of the size of* C*).*

2.2 Bootstrappable Encryption

Following Gentry [6], we construct homomorphic encryption for circuits of any depth from one that is capable of evaluating just a little more than its own decryption circuit.

Definition 4 (Augmented Decryption Circuits). *Let* $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Evaluate})$ *be an encryption scheme, where decryption is implemented by a circuit that depends only on the security parameter.*[2]

For a given value of the security parameter λ*, the set of augmented decryption circuits consists of two circuits, both take as input a secret key and two ciphertexts: One circuit decrypts both ciphertexts and adds the resulting plaintext bits mod 2, the other decrypts both ciphertexts and multiplies the resulting plaintext bits mod 2. We denote this set by* $D_{\mathcal{E}}(\lambda)$*.*

Definition 5 (Bootstrappable Encryption). *Let* $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Evaluate})$ *be a homomorphic encryption scheme, and for every value of the security parameter* λ *let* $\mathcal{C}_{\mathcal{E}}(\lambda)$ *be a set of circuits with respect to which* $\mathcal{E}$ *is correct. We say that* $\mathcal{E}$ *is* bootstrappable *if* $D_{\mathcal{E}}(\lambda) \subseteq \mathcal{C}_{\mathcal{E}}(\lambda)$ *holds for every* λ*.*

Theorem 1 ([6]). *There is an (efficient, explicit) transformation that given a description of a bootstrappable scheme* $\mathcal{E}$ *and a parameter* $d = d(\lambda)$*, outputs a description of another encryption scheme* $\mathcal{E}^{(d)}$ *such that:*

1. $\mathcal{E}^{(d)}$ *is compact (in particular the* $\mathsf{Decrypt}$ *circuit in* $\mathcal{E}^{(d)}$ *is identical to that in* $\mathcal{E}$*), and*
2. $\mathcal{E}^{(d)}$ *is homomorphic for all circuits of depth up to* d*.*

Moreover, $\mathcal{E}^{(d)}$ *is semantically secure if* $\mathcal{E}$ *is: Any attack with advantage* ε *against* $\mathcal{E}^{(d)}$ *can be converted into an attack with similar complexity against* $\mathcal{E}$ *with advantage at least* $\varepsilon/\ell d$ *, where* ℓ *is the length of the secret key in* $\mathcal{E}$*.*

We also note that if the bootstrappable scheme $\mathcal{E}$ is "circular secure" then it can be converted into a single compact fully-homomorphic encryption scheme $\mathcal{E}'$. See [6] for details.

3 A Somewhat Homomorphic Encryption Scheme

Parameters. The construction below has many parameters, controlling the number of integers in the public key and the bit-length of the various integers.

[2] This in particular means that for a fixed value of the security parameter, the size of the secret key is always the same, and similarly all the ciphertexts that can be decrypted have the same size.

Specifically, we use the following four parameters (all polynomial in the security parameter λ):

γ is the bit-length of the integers in the public key,
η is the bit-length of the secret key (which is the hidden approximate-gcd of all the public-key integers),
ρ is the bit-length of the noise (i.e., the distance between the public key elements and the nearest multiples of the secret key), and
τ is the number of integers in the public key.

These parameters must be set under the following constraints:

- $\rho = \omega(\log \lambda)$, to protect against brute-force attacks on the noise;
- $\eta \geq \rho \cdot \Theta(\lambda \log^2 \lambda)$, in order to support homomorphism for deep enough circuits to evaluate the "squashed decryption circuit" (cf. Sections 3.2 and 6.2);
- $\gamma = \omega(\eta^2 \log \lambda)$, to thwart various lattice-based attacks on the underlying approximate-gcd problem (cf. Section 5);
- $\tau \geq \gamma + \omega(\log \lambda)$, in order to use the leftover hash lemma in the reduction to approximate gcd.

We also use a secondary noise parameter $\rho' = \rho + \omega(\log \lambda)$. A convenient parameter set to keep in mind is $\rho = \lambda$, $\rho' = 2\lambda$, $\eta = \tilde{O}(\lambda^2)$, $\gamma = \tilde{O}(\lambda^5)$ and $\tau = \gamma + \lambda$. (This setting results in a scheme with complexity $\tilde{O}(\lambda^{10})$.)

For a specific (η-bit) odd positive integer p, we use the following distribution over γ-bit integers:

$$\mathcal{D}_{\gamma,\rho}(p) = \left\{ \mathsf{choose}\ q \xleftarrow{\$} \mathbb{Z} \cap [0,\ 2^\gamma/p),\ r \xleftarrow{\$} \mathbb{Z} \cap (-2^\rho,\ 2^\rho)\ :\ \mathsf{output}\ x = pq + r \right\}$$

This distribution is clearly efficiently sampleable.

3.1 The Construction

$\mathsf{KeyGen}(\lambda)$**.** The secret key is an odd η-bit integer: $p \xleftarrow{\$} (2\mathbb{Z}+1) \cap [2^{\eta-1}, 2^\eta)$.
For the public key, sample $x_i \xleftarrow{\$} \mathcal{D}_{\gamma,\rho}(p)$ for $i = 0, \ldots, \tau$. Relabel so that x_0 is the largest. Restart unless x_0 is odd and $r_p(x_0)$ is even. The public key is $\mathrm{pk} = \langle x_0, x_1, \ldots, x_\tau \rangle$.

$\mathsf{Encrypt}(\mathbf{pk}, m \in \{0,1\})$**.** Choose a random subset $S \subseteq \{1, 2, \ldots, \tau\}$ and a random integer r in $(-2^{\rho'},\ 2^{\rho'})$, and output $c \leftarrow \left[m + 2r + 2\sum_{i \in S} x_i\right]_{x_0}$.

$\mathsf{Evaluate}(\mathbf{pk}, C, c_1, \ldots, c_t)$**.** Given the (binary) circuit $\mathcal{C}_\mathcal{E}$ with t inputs, and t ciphertexts c_i, apply the (integer) addition and multiplication gates of $\mathcal{C}_\mathcal{E}$ to the ciphertexts, performing all the operations over the integers, and return the resulting integer.

$\mathsf{Decrypt}(\mathbf{sk}, c)$**.** Output $m' \leftarrow (c \bmod p) \bmod 2$.

Remark 1. Recall that $(c \bmod p) = c - p \cdot \lfloor c/p \rceil$, and as p is odd we can instead decrypt using the formula $m' \leftarrow [c - \lfloor c/p \rceil]_2 \ = \ (c \bmod 2) \oplus (\lfloor c/p \rceil \bmod 2)$.

Remark 2. Originally, we described encryption as adding m to a random subset sum of "encryptions of zero". Indeed, the scheme can viewed this way. Let $w_i = [2x_i]_{x_0}$ for $i = 1, \ldots, \tau$. Each w_i, and also x_0, is essentially an encryption of zero; its noise is even. Moreover, $c = m + 2r + \sum_{i \in S} w_i - k \cdot x_0$ for some integer k.

3.2 Correctness

Permitted Circuits and Polynomials. For a mod-2 arithmetic circuit (composed of mod-2 Add and Mult gates), we consider its generalization to the integers, i.e., the same circuits with the Add and Mult gates applied to integers rather than to bits. Similar to Gentry [6], we define a *permitted circuit* as one where for any $\alpha \geq 1$ and any set of integer inputs all less than $2^{\alpha(\rho'+2)}$ in absolute value, it holds that the generalized circuit's output has absolute value at most $2^{\alpha(\eta-4)}$. Let $\mathcal{C}_{\mathcal{E}}$ denote the set of permitted circuits. Clearly, we have:

Lemma 2. *The scheme from above is correct for $\mathcal{C}_{\mathcal{E}}$.* □

Remark 3. Since "fresh" ciphertexts output by Encrypt have noise at most $2^{\rho'+2}$, the ciphertext output by Evaluate applied to a permitted circuit has noise at most $2^{\eta-4} < p/8$. The bound $2^{\eta-2} < p/2$ would suffice for correct decryption. But we will later use the fact that the noise remains below $p/8$ in Section 6 to perform the decryption operation using a very shallow arithmetic circuit.

The definition of the set $\mathcal{C}_{\mathcal{E}}$ from above is rather indirect. In particular this definition does not give a good picture of what $\mathcal{C}_{\mathcal{E}}$ "looks like". By the triangle inequality, a k-fan-in Add gate clearly increases the magnitude of the integers by at most a factor of k. However, a 2-fan-in Mult gate may *square* the magnitude of the integers – i.e., double their bit-lengths. So, clearly, the main bottleneck is the *multiplicative depth* of the circuit, or the *degree* of the multivariate polynomial computed by the circuit. We have the following lemma.

Lemma 3. *Let C be a boolean circuit with t inputs, and let $C^\dagger$ be the associated integer circuit (where boolean gates are replaced with integer operations). Let $f(x_1, \ldots, x_t)$ be the multivariate polynomial computed by $C^\dagger$; let d be its degree. If $|\boldsymbol{f}| \cdot (2^{\rho'+2})^d \leq 2^{\eta-4}$ (where $|\boldsymbol{f}|$ is the l_1 norm of the coefficient vector of f) then $C \in \mathcal{C}_{\mathcal{E}}$.* □

In particular, $\mathcal{E}$ can handle f as long as

$$d \;\leq\; \frac{\eta - 4 - \log|\boldsymbol{f}|}{\rho' + 2} \tag{1}$$

Below we refer to polynomials that satisfy Equation (1) as *permitted polynomials* and we denote by $\mathcal{P}_{\mathcal{E}}$ the set of permitted polynomials and by $C(\mathcal{P}_{\mathcal{E}})$ the set of circuits that compute them. The discussion above implies that $C(\mathcal{P}_{\mathcal{E}}) \subseteq \mathcal{C}_{\mathcal{E}}$.

Remark 4. For our purposes, we consider settings where $\log|\boldsymbol{f}|$ is small in relation to η, $\rho' = \omega(\log\lambda)$ and $t, \tau \leq \lambda^\beta$, and we need to support polynomials of degree up to $\alpha\lambda\log^2\lambda$ (for some constants α, β). Plugging these expressions in Equation (1), it is sufficient to set $\eta = \rho' \cdot \Theta(\lambda \log^2 \lambda)$.

3.3 Optimizations

Modular-reduction during Evaluate. Note that while Encrypt reduces the ciphertext modulo the public key element x_0, we cannot do the same in Evaluate. The reason is that after just one multiplication the ciphertext becomes much larger than x_0, so modular reduction will include a large multiple of x_0 hence introducing intolerable error.

To reduce the ciphertext size during Evaluate, we can add to the public key more elements of the form $x'_i = q'_i p + 2r'_i$ where the r'_i's are chosen as usual from the interval $(-2^\rho, 2^\rho)$ but the q_i's are chosen much larger than for the other public key elements. Specifically, for $i = 0, \dots \gamma$, we set:

$$q'_i \overset{\$}{\leftarrow} \mathbb{Z} \cap [2^{\gamma+i-1}/p,\ 2^{\gamma+i}/p), \quad r'_i \overset{\$}{\leftarrow} \mathbb{Z} \cap (-2^\rho, 2^\rho), \quad x'_i \leftarrow 2(q'_i \cdot p + r'_i),$$

thus getting $x'_i \in [2^{\gamma+i},\ 2^{\gamma+i+1}]$. During Evaluate, every time we have a ciphertext that grows beyond 2^γ, we reduce it first modulo x'_γ, then modulo $x'_{\gamma-1}$, and so on all the way down to x'_0, at which point we again have a ciphertext of bit-length no more than γ.

Recall that a single operation at most doubles the bit-length of the ciphertext. Hence after any one operation the ciphertext cannot be larger than $2x'_\gamma$, and therefore the sequence of modular reductions involves only small multiples of the x'_i's, which means that it only adds a small amount of noise. (We note that in addition to smaller ciphertexts, this optimization also reduces the public key size when we use the "decryption squashing" technique as described in Section 6.1.)

It is not clear to what extent adding these larger integers to the public key influences the security of the scheme. It does change the specifics of the approximate-GCD assumption that we need to make, but the same decision-to-search reduction from Section 4 still goes through.[3] Also, we note that having integers with these very large quotients does not seem to help in any of the attacks on approximate-GCD that we considered.

Remark 5. Note that when using the original scheme without the optimization, homomorphic evaluation of different circuits that compute the same polynomial would result in the exact same output ciphertext (i.e., the polynomial applied to the input ciphertexts over the integers). This is no longer true when using the size-reduction optimization, because of the additional modular reduction steps. For example, evaluating the circuit "$x_1(x_2 + x_3)$" is likely to yield a different ciphertext than the circuit "$x_1x_2 + x_1x_3$."

In principle, it is plausible that evaluating one circuit would yield a ciphertext with small enough noise to be decrypted, while evaluating another circuit for the same polynomial will produce a ciphertext with too much noise. Adapting the "bootstrappability analysis" from Section 6.2 to the optimized scheme, one would have to take into account not only the degree of the polynomial implementing the decryption process but also the particular circuit that implements this polynomial. It should not be hard to argue that the circuit in Section 6.2 does not introduce too much noise, but the analysis is quite tedious and is omitted here.

[3] Allowing this reduction to go through is the reason that the x'_i's are set as even integers.

Ciphertext compression. Even though the optimization from above keeps evaluated ciphertexts at the same length as original ciphertexts, the size of these ciphertexts is still very large – $\tilde{\theta}(\lambda^5)$ bits under our suggested parameters. We next show how to "compress", or post-process the ciphertexts, down to (asymptotically) the size of an RSA modulus, reducing the communication complexity of our scheme dramatically.

The price of this optimization, however, is that we cannot evaluate anything on these compressed ciphertexts. Hence we can only use this compression technique on the final output ciphertexts, after all applications of the Evaluate algorithm have been completed. (This technique also introduces another hardness assumption, similar to the ϕ-hiding assumption of Cachin et al. [3].)

Roughly, we supplement the public key with the description of a group G and an element $g \in G$ whose order is a multiple of the secret key p. Then, given the ciphertext c from our scheme, the compressed ciphertext is simply $c^* \leftarrow g^c$. Note that $\mathrm{DL}_g(c^*) = c \pmod{p}$, so decrypting is done by first computing $y \leftarrow \mathrm{DL}_g(c^*) \bmod p$, and then $m \leftarrow y \bmod 2$. Correctness follows immediately from the correctness of the original scheme.

To implement this idea, we need to choose the secret key p as a smooth number so that we can compute $(\mathrm{DL}_g(c^*) \bmod p)$ on decryption. It seems sufficient to choose the secret key as a product of random distinct $\lambda^2/\log\lambda$ small primes (say, all smaller than λ^3). Also, we need to ensure that publishing G, g does not violate the security of the scheme. This can be accomplished by publishing an RSA modulus N such that $p|\phi(N)$ (and $\log N$ sufficiently larger than $4\log p$),[4] along with a random element $g \in_R Z_N^*$, relying on a variant of the ϕ-hiding assumption [3]. Namely, we assume that given two smooth numbers p_1, p_2 as above and given N such that one of the p_i's divides $\phi(N)$, it is hard to determine which of the two p_i's divides $\phi(N)$. In the full version we describe this optimization in more details, and provide a proof of security for it under this ϕ-hiding variant.

4 Security of the Somewhat Homomorphic Scheme

We reduce the security of the scheme from Section 3 to the hardness of the approximate-gcd problem. Namely, given a set of integers $x_0, x_1, \ldots, x_\tau$, all randomly chosen close to multiples of a large integer p, find this "common near divisor" p.

On a high level, our reduction resembles classical hard-core-bit proofs in factoring-based cryptography (e.g., Alexi et al. [1]): Fixing a randomly-chosen public key, we roughly show that an adversary who can predict the encrypted bit in a random ciphertext under this public key can be used to find the secret key (for this fixed public key). As in [1], we describe a random-self-reduction and accuracy-amplification step that uses the promised adversary to get a reliable oracle for the least-significant bit, and then a binary-GCD algorithm that uses that reliable oracle to find p.

[4] The condition $\log N > 4\log p$ is needed, since otherwise we can use Coppersmith's method [4] to break the corresponding ϕ-hiding assumption.

The technical details, of course, are very different than in factoring-based cryptography. Perhaps the main difference is that our random self-reduction entails a loss in parameters. Specifically, we show that a noticeable advantage in guessing the encrypted bit in a random "high noise ciphertext" – where the noise is ρ' bits – can be converted into the ability to predict reliably the parity bit of the quotient in an arbitrary "low noise integer" – where the noise is ρ bits. (Roughly, the reason for this is that we need to add extra noise to "wipe out the traces" of the non-random noise in the arbitrary input integer.)

The implication is that we can only reduce the security of our cryptosystem in the "high-noise regime" to the hardness of approximate-gcd in the "low-noise regime." Note that the difference between "high noise" and "low noise" is rather small: only $\omega(\log \lambda)$ bits.

4.1 Reduction to Approximate-GCD

The approximate-gcd problem is defined as follows:

Definition 6 (Approximate GCD). *The (ρ, η, γ)-approximate-gcd problem is: given polynomially many samples from $\mathcal{D}_{\gamma,\rho}(p)$ for a randomly chosen η-bit odd integer p, output p.*

Theorem 2. *Fix the parameters $(\rho, \rho', \eta, \gamma, \tau)$ as in the Somewhat Homomorphic Scheme from Section 3 (all polynomial in the security parameter λ).*

Any attack $\mathcal{A}$ with advantage ε on the encryption scheme can be converted into an algorithm $\mathcal{B}$ for solving (ρ, η, γ)-approximate-gcd with success probability at least $\varepsilon/2$. The running time of $\mathcal{B}$ is polynomial in the running time of $\mathcal{A}$, and in λ and $1/\epsilon$.

Proof. Recall that we use $q_p(z)$ and $r_p(z)$ to denote the quotient and remainder of z with respect to p, hence $z = q_p(z) \cdot p + r_p(z)$. Let $\mathcal{A}$ be an attacker against the scheme. Namely, $\mathcal{A}$ takes as input a public key and a ciphertext (as produced by KeyGen and Encrypt of our scheme), and outputs the correct plaintext bit with probability $\frac{1}{2}+\epsilon$ for some noticeable ϵ. (The probability is over KeyGen and Encrypt, as well as the choice of the plaintext bit and the internal randomness of $\mathcal{A}$.)

We use $\mathcal{A}$ to construct a solver $\mathcal{B}$ for approximate-gcd with parameters ρ, η, γ. For a randomly chosen η-bit odd integer p, the solver $\mathcal{B}$ has access to as many samples from $\mathcal{D}_{\gamma,\rho}(p)$ as it needs, and the goal is to find p.

Step 1: Creating a public key. The solver $\mathcal{B}$ begins by constructing a public key for the scheme. $\mathcal{B}$ draws $\tau + 1$ samples $x_0, \ldots, x_\tau \xleftarrow{\$} \mathcal{D}_{\gamma,\rho}(p)$. It relabels so that x_0 is the largest. It restarts unless x_0 is odd. $\mathcal{B}$ then outputs a public key $\mathrm{pk} = \langle x_0, x_1, \ldots, x_\tau \rangle$. Clearly, if $r_p(x_0)$ *happens* to be even then the distribution induced on the public key is identical to that of the scheme.

Step 2: A subroutine for high-accuracy LSB predictor. Next, $\mathcal{B}$ produces a sequence of integers, and attempts to recover p by utilizing $\mathcal{A}$ to learn the least-significant bit of the quotients of these integers with respect to p. For this, $\mathcal{B}$ uses the following subroutine:

```
Subroutine Learn-LSB($z, \mathrm{pk}$):
Input: $z \in [0, 2^\gamma)$ with $|r_p(z)| < 2^\rho$, a public key $\mathrm{pk} = \langle x_0, x_1, \ldots, x_\tau \rangle$
Output: The least-significant-bit of $q_p(z)$
1. For $j = 1$ to $\mathrm{poly}(\lambda)/\epsilon$ do:          // $\epsilon$ is the overall advantage of $\mathcal{A}$
2.     Choose noise $r_j \xleftarrow{\$} (-2^{\rho'}, 2^{\rho'})$, a bit $m_j \xleftarrow{\$} \{0,1\}$,
           and a random subset $S_j \subseteq_R \{1, \ldots, \tau\}$
3.     Set $c_j \leftarrow \left[ z + m_j + 2r_j + 2\sum_{k \in S_j} x_k \right]_{x_0}$
4.     Call $\mathcal{A}$ to get a prediction $a_j \leftarrow \mathcal{A}(\mathrm{pk}, c_j)$
5.     Set $b_j \leftarrow a_j \oplus \mathsf{parity}(z) \oplus m_j$          // $b_j$ should be the parity of $q_p(z)$
6. Output the majority vote among the $b_j$'s.
```

In the full version [21] we show that for all but a negligible fraction of the public keys generated by the scheme, the "ciphertext" c_j in line 3 is distributed almost identically to a valid encryption of the bit $[r_p(z)]_2 \oplus m_j$. Note also that since p is odd, we always have $[q_p(z)]_2 = [r_p(z)]_2 \oplus \mathsf{parity}(z)$. It follows that if $\mathcal{A}$ has a noticeable advantage in guessing the encrypted bit under pk then Learn-LSB(z, pk) will return $[q_p(z)]_2$ with overwhelming probability.

Step 3: Binary GCD. Once we turned $\mathcal{A}$ into an oracle for the least-significant-bit of $q_p(z)$, recovering p is rather straightforward. Perhaps the simplest way of doing it is using the binary GCD algorithm [12]: Given any two integers $z_1 = q_p(z_1) \cdot p + r_p(z_1)$ and $z_2 = q_p(z_2) \cdot p + r_p(z_2)$ (with $r_p(z_i) \ll p$), repeatedly apply the following process to them:

1. If $z_2 > z_1$ then swap them, $z_1 \leftrightarrow z_2$.
2. Use the oracle to learn the parity bit of both $q_p(z_1)$ and $q_p(z_2)$, denote $b_i = [q_p(z_i)]_2$.
3. If both $q_p(z_i)$ are odd then replace z_1 by $z_1 \leftarrow z_1 - z_2$ and set $b_1 \leftarrow 0$.
4. For each z_i with $b_i = 0$, replace z_i by $z_i \leftarrow (z_i - \mathsf{parity}(z_i))/2$. (Note that $z_i - \mathsf{parity}(z_i)$ is even, so the new z_i is an integer.)

Observe that when $p \gg r_p(z_i)$, subtracting the parity bit does not change the quotient with respect to p, only the remainder. That is, $q_p(z_i - \mathsf{parity}(z_i)) = q_p(z_i)$. It follows that when we set $z_i' \leftarrow (z_i - \mathsf{parity}(z_i))/2$ in line 4 (where we know that $q_p(z_i)$ is even), we get

$$q_p(z_i') = q_p(z_i)/2 \quad \text{and} \quad r_p(z_i') = \big(r_p(z_i) - \mathsf{parity}(z_i)\big)/2.$$

We now show that the noise in z_1, z_2 never grows too large in this process. Clearly, setting $z_i' \leftarrow (z_i - \mathsf{parity}(z_i))/2$ in line 4 we have $|r_p(z_i')| \le (|r_p(z_i)| + 1)/2 \le |r_p(z_i)|$. Moreover, when we replace z_1 by $z_1' \leftarrow z_1 - z_2$ in line 3 and then by $z_1'' \leftarrow (z_1' - \mathsf{parity}(z_1'))/2$ in line 4, we have

$$|r_p(z_1'')| \;=\; \big(|r_p(z_1) - r_p(z_2) - \mathsf{parity}(z_1')|\big)/2 \;\le\; \max\big\{|r_p(z_1)|, |r_p(z_2)|\big\}$$

Hence the $r_p(z_i)$'s never grow beyond the largest of the initial two, so we always have $p \gg r_p(z_i)$.

This implies that the operations above correspond to the usual operations of the binary GCD algorithm, applied to the $q_p(z_i)$'s. Hence after $O(\gamma)$ iterations we will finally get two integers z'_1, z'_2 with $z'_2 = 0$ and $q_p(z'_1)$ being the odd part of $GCD(q_p(z_1), q_p(z_2))$ (for the two initial integers).

Step 4: Recovering **p**. To recover p, the solver $\mathcal{B}$ draws a pair of elements $z_1^*, z_2^* \overset{\$}{\leftarrow} \mathcal{D}_{\gamma,\rho}(p)$ and applies the binary-GCD algorithm to them. With probability at least $\pi^2/6 \approx 0.6$, the odd part of $GCD(q_p(z_1^*), q_p(z_2^*))$ is one, which means that the procedure will output an element $\tilde{z} = 1 \cdot p + r$ with $|r| \le 2^\rho$. (If this does not happen then $\mathcal{B}$ draws two new integers and tries again.)

Lastly $\mathcal{B}$ repeats the binary-GCD procedure from above using $z_1 = z_1^*$ and $z_2 = \tilde{z}$, and the sequence of parity bits of the $q_p(z_1)$'s in all the iterations spell out the binary representation of $q_p(z_1^*)$. Now $\mathcal{B}$ recovers $p = \lfloor z_1^*/q_p(z_1^*) \rceil$.

Summary. We have shown that $\mathcal{B}$ can recover p given access to a reliable oracle for computing $[q_p(z)]_2$ (for z's with noise much smaller than p). It is left to analyze the probability (over $\mathcal{B}$'s choice of public key) with which the procedure $\mathsf{Learn\text{-}LSB}(z, \mathrm{pk})$ from above is indeed such a reliable oracle.

The Success Probability of $\mathcal{B}$. In the full version we prove a simple technical lemma about the distribution of ciphertexts in our scheme. Recall that conditioned on some probability-$\frac{1}{2}$ event in our reduction (i.e., $q_p(x_0)$ is odd), the distribution of the public key that $\mathcal{B}$ generates is identical to the correct distribution from the scheme. Let us denote this probability-$\frac{1}{2}$ "good event" by $\mathcal{G}$. In the full version we prove that for every secret key p and for all but a negligible fraction of the public keys (as generated by KeyGen for the secret key p), the procedure that $\mathcal{B}$ uses to generate ciphertexts in line 3 of the subroutine $\mathsf{Learn\text{-}LSB}$ produces a distribution which is statistically close to the ciphertext distribution of the scheme. This lets us analyze the success probability of $\mathcal{B}$, as follows: Let $\mathcal{P}$ be the set of odd integers in $[2^{\eta-1}, 2^\eta)$ for which $\mathcal{A}$ has more than $\varepsilon/2$ advantage

$$\mathcal{P} \overset{\text{def}}{=} \left\{p \in [2^{\eta-1}, 2^\eta) \ : \ \mathsf{advantage}(\mathcal{A}) \text{ conditioned on sk} = p \text{ is at least } \varepsilon/2\right\}$$

A counting argument shows that the fraction of odd integers from $[2^{\eta-1}, 2^\eta)$ that are in $\mathcal{P}$ is at least $\varepsilon/2$. For a given $p \in \mathcal{P}$, we similarly denote by $\mathcal{PK}_p$ the set of public keys for which $\mathcal{A}$ has advantage at least $\varepsilon/4$:

$$\mathcal{PK}_p \overset{\text{def}}{=} \{\text{pk for } p \ : \ \mathsf{advantage}(\mathcal{A}) \text{ conditioned on pk is at least } \varepsilon/4\}$$

Again, for every $p \in \mathcal{P}$, the KeyGen algorithm (when using the secret key $\mathrm{sk} = p$) must output $\mathrm{pk} \in \mathcal{PK}_p$ with probability at least $\varepsilon/4$.

Consider now a single run of $\mathcal{B}$ when it is given access to $\mathcal{D}_{\gamma,\rho}(p)$ for some $p \in \mathcal{P}$. With probability 1/2 the "good event" $\mathcal{G}$ happens, in which case the public key that $\mathcal{B}$ produces is negligibly close to the right distribution. Hence conditioned on $\mathcal{G}$, $\mathcal{B}$ generates some $\mathrm{pk} \in \mathcal{PK}_p$ with probability $\varepsilon' \ge \varepsilon/4 - \mathsf{negl}$. Moreover, with probability $\varepsilon' - \mathsf{negl}$ not only is the public key in $\mathcal{PK}_p$, but also the ciphertext-generation that $\mathcal{B}$ uses in line 3 of $\mathsf{Learn\text{-}LSB}$ "works" for this public key (meaning that the ciphertexts that it generates are chosen from almost the right distribution). If that happens then $\mathcal{A}$ returns the right answer in line 4 of Learn-LSB with probability $\varepsilon/4 - \mathsf{negl}$. As that subroutine calls $\mathcal{A}$ for $\text{poly}(\lambda)/\varepsilon$ times and takes majority vote, it will return the right answer with overwhelming probability, and $\mathcal{B}$ will recover the approximate-gcd p.

Thus, when the hidden secret is $p \in \mathcal{P}$ then $\mathcal{B}$ has probability at least $1/2 \cdot (\varepsilon/4 - \mathsf{negl})$ of recovering it in a single run. Repeating the algorithm $\mathcal{B}$ for $(8/\epsilon) \cdot \omega(\log \lambda)$ times will therefore recover such p's with overwhelming probability. Hence we have a solver of complexity $\text{poly}(\lambda, 1/\varepsilon)$ that works with overwhelming probability for every $p \in \mathcal{P}$, so the overall success probability of this solver is at least the density of $\mathcal{P}$, which is at least $\varepsilon/2$. This completes the proof of Theorem 2. □

5 Known Attacks

Consider the approximate-gcd instance $\{x_0, \ldots, x_t\}$ where $x_i = pq_i + r_i$. In this section, we first review known attacks on the approximate-gcd problem for two numbers (i.e., when $t = 1$) – including brute-forcing the remainders, continued fractions, and Howgrave-Graham's approximate gcd algorithm [9]. Later, we consider attacks for arbitrarily large values of t – including lattice-based algorithms for simultaneous Diophantine approximation [13], Nguyen and Stern's orthogonal lattice [17], and extensions of Coppersmith's method to multivariate polynomials [4].

5.1 The Approximate GCD of Two Numbers

A simple brute-force attack is to try to guess r_1 and r_2, and verify the guess with a gcd computation. Specifically, for $r'_1, r'_2 \in (-2^\rho,\ 2^\rho)$, set

$$x'_1 \leftarrow x_1 - r'_1\ ,\quad x'_2 \leftarrow x_2 - r'_2\ ,\quad p' \leftarrow \gcd(x'_1, x'_2)$$

If p' has η bits, output p' as a possible solution. The solution p will definitely be found by this technique, and for our parameter choices, where ρ is much smaller than η, the solution is likely to be unique. The running time of the attack is approximately $2^{2\rho}$.

A variant of the brute-force attack is to set x'_1 as above, factor x'_1, and, if there is an η-bit factor p', see whether p' is an approximate divisor of x'_2. Since in our parameters γ is substantially greater than η, the attack should use a factoring algorithm whose performance depends primarily on the size of the target factor rather than the size of the entire number being factored. For example, Lenstra's elliptic curve factoring algorithm [14] runs in time roughly $\exp(O(\sqrt{\eta}))$ (with only polynomial dependence on γ), thus resulting in overall attack complexity $\approx 2^{\rho+\sqrt{\eta}}$. The attack time is less if the approximate gcd is known to be smooth, but still exponential in ρ.

Continued fractions seem like a natural way to recover p from x_1 and x_2. Using continued fractions, one obtains a sequence of integer pairs (a_i, b_i) such that

$|x_1/x_2 - a_i/b_i| < 1/b_i^2$. Moreover, every pair (s,t) such that $|x_1/x_2 - s/t| < 1/2t^2$ is in the sequence. Since q_1/q_2 is a good approximation of x_1/x_2, one may hope that it occurs as a pair in the sequence; if so, one recovers $p = \lfloor x_1/q_1 \rceil$. However, in our scheme, $|x_1/x_2 - q_1/q_2|$ is not small enough to be recovered using continued fractions. Specifically, we have

$$\left|\frac{x_1}{x_2} - \frac{q_1}{q_2}\right| = \left|\frac{q_2 r_1 - q_1 r_2}{q_2(pq_2 + r_2)}\right| \approx \left|\frac{q_2 r_1 - q_1 r_2}{p}\right| \cdot \frac{1}{q_2^2}$$

where $(q_2 r_1 - q_1 r_2)/p$ in the final term is likely to be *much* larger than 1. To describe the failure of continued fractions another way, the mere fact that an approximant a_i/b_i is close to x_1/x_2 does *not* mean that there exist $r_1', r_2' \ll p$ such that $x_1 = pa_i + r_1'$ and $x_2 = pb_i + r_2'$ – i.e., the continued fractions method is not constrained to output the kind of approximants that we need. See [9] for a more detailed exposition of the continued fractions approach to approximate-gcd.

Howgrave-Graham [9] also gives a lattice attack on the two-element approximate-gcd problem that is related to Coppersmith's celebrated algorithm for finding small solutions to univariate and bivariate modular equations [4]. For the case where x_1 is exactly divisible by p, where his algorithm performs slightly better, the attack recovers p when ρ/γ is smaller than $(\eta/\gamma)^2$. The algorithm does not degrade gracefully for ρ, η, γ that do not satisfy the constraint. Rather, in this case, the relevant lattice may contain exponentially vectors unrelated to the approximate-gcd solution, so that lattice reduction yields nothing useful.

5.2 The Approximate GCD of Many Numbers

Now, let us consider attacks – specifically, lattice attacks – for arbitrary t. First, note that the rational numbers $y_i = x_i/x_0$ are an instance of the simultaneous Diophantine approximation (SDA) problem: indeed for all i it holds that $\frac{x_i}{x_0} = \frac{q_i + s_i}{q_0}$, where $|s_i| \approx 2^{\rho - \eta}$. We can therefore try to use Lagarias' algorithm for SDA [13], namely apply LLL to the $(t+1)$-dimensional lattice L spanned by the rows of the following matrix:

$$M = \begin{pmatrix} 2^\rho & x_1 & x_2 & \dots & x_t \\ & -x_0 & & & \\ & & -x_0 & & \\ & & & \ddots & \\ & & & & -x_0 \end{pmatrix}$$

Our target solution corresponds to a vector of length roughly $2^{\gamma+\rho-\eta}\sqrt{t+1}$ – specifically,

$$\begin{aligned} \boldsymbol{v} = \langle q_0, q_1, \dots, q_t \rangle \cdot M &= \langle q_0 2^\rho,\ q_0 x_1 - q_1 x_0,\ \dots,\ q_0 x_t - q_t x_0 \rangle \\ &= \left\langle q_0 2^\rho,\ x_0 q_0 (\frac{x_1}{x_0} - \frac{q_1}{q_0}),\ \dots,\ x_0 q_0 (\frac{x_t}{x_0} - \frac{q_t}{q_0}) \right\rangle, \end{aligned}$$

where the first entry in $\boldsymbol{v}$ satisfies $|q_0 2^\rho| < 2^{\gamma-\eta+\rho}$ and all the other entries satisfy $|x_0 q_0(\frac{x_i}{x_0} - \frac{q_i}{q_0})| = |x_0 s_i| \approx 2^{\gamma+\rho-\eta}$.

However, the target solution is not necessarily the shortest nonzero vector in the lattice, and therefore is not necessarily discovered by lattice reduction. In particular, Minkowski tells us that L has a nonzero vector of length at most $\det(L)^{1/(t+1)}\sqrt{t+1} < 2^{(\rho+t\gamma)/(t+1)}\sqrt{t+1} = 2^{\gamma+(\rho-\gamma)/(t+1)}\sqrt{t+1}$. This is shorter than our target solution when $t+1 < \gamma/\eta$. In fact, heuristically, L will tend to have exponentially (in t) many vectors of length $\mathrm{poly}(t)\det(L)^{1/(t+1)}$, which obscure our target solution.[5]

On the other hand, when t is large, $\boldsymbol{v}$ likely is the shortest vector in L, but known lattice reductions algorithms will not be able to find it efficiently. Specifically, as a rule of thumb, they require time roughly $2^{t/k}$ to output a 2^k approximation of the shortest vector. Since clearly there are exponentially (in t) many vectors in L of length at most $\|x_0\|\sqrt{t+1} < 2^\gamma\sqrt{t+1}$, which is about $2^{\eta-\rho}$ times longer than $\boldsymbol{v}$, we need better than a $2^{\eta-\rho}$ approximation. For $t \geq \gamma/\eta$, the time needed to guarantee a 2^η approximation (which is not even good enough to recover $\boldsymbol{v}$) is roughly $2^{\gamma/\eta^2}$. Thus setting $\gamma/\eta^2 = \omega(\log\lambda)$ foils this attack.

Other known attacks are described in the full version. These attacks do not perform any better than the ones above, and our choice of parameters achieves at least 2^λ security against all of them.

6 Making the Scheme Fully Homomorphic

We follow Gentry's approach [6] for constructing a fully homomorphic encryption scheme from a somewhat homomorphic scheme $\mathcal{E}$ that is *bootstrappable* as per Definition 5. For reasons similar to those in Gentry's construction from [6], computing the decryption equation $m' \leftarrow [c - \lfloor c/p \rceil]_2$ seems to require boolean circuits that are deeper (by a constant factor) than what our somewhat homomorphic scheme can handle. Hence we use Gentry's transformation to "squash the decryption circuit." In this transformation, we add to the public key some extra information about the secret key, and use this extra information to "post process" the ciphertext. The post-processed ciphertext can be decrypted more efficiently than the original ciphertext, thus making the scheme bootstrappable. We pay for this saving by having a larger ciphertext, and also by introducing another hardness assumption (basically assuming that the extra information in the public key does not help an attacker break the scheme).

6.1 Squashing the Decryption Circuit

Let κ, θ, Θ be three more parameters, which are functions of λ. Concretely, below we use $\kappa = \gamma\eta/\rho'$, $\theta = \lambda$, and $\Theta = \omega(\kappa \cdot \log\lambda)$.[6] For a secret key $\mathrm{sk}^* = p$ and

[5] When t is very small – e.g., $t = 1$ – the information that one obtains from the two dimensional lattice is related to what one obtains from the continued fractions approach.

[6] When using the size-reduction optimization from Section 3.3 it is sufficient to use $\kappa = \gamma + 2$, which would also make Θ smaller.

public key pk* from the original somewhat homomorphic scheme $\mathcal{E}^*$, we add to the public key a set $\boldsymbol{y} = \{y_1, \ldots, y_\Theta\}$ of rational numbers in $[0, 2)$ with κ bits of precision, such that there is a sparse subset $S \subset \{1, \ldots, \Theta\}$ of size θ with $\sum_{i \in S} y_i \approx 1/p \pmod 2$. We also replace the secret key by the indicator vector of the subset S. In more details, we modify the encryption scheme from Section 3 as follows:

KeyGen. Generate sk$^* = p$ and pk* as before. Set $x_p \leftarrow \lfloor 2^\kappa/p \rceil$, choose at random a Θ-bit vector with Hamming weight θ, $\boldsymbol{s} = \langle s_1, \ldots, s_\Theta \rangle$, and let $S = \{i : s_i = 1\}$.

Choose at random integers $u_i \in \mathbb{Z} \cap [0, 2^{\kappa+1})$, $i = 1, \ldots, \Theta$, subject to the condition that $\sum_{i \in S} u_i = x_p \pmod{2^{\kappa+1}}$. Set $y_i = u_i/2^\kappa$ and $\boldsymbol{y} = \{y_1, \ldots, y_\Theta\}$. Hence each y_i is a positive number smaller than two, with κ bits of precision after the binary point. Also, $[\sum_{i \in S} y_i]_2 = (1/p) - \Delta_p$ for some $|\Delta_p| < 2^{-\kappa}$.

Output the secret key sk $= \boldsymbol{s}$ and public key pk $= (\text{pk}^*, \boldsymbol{y})$.

Encrypt and Evaluate. Generate a ciphertext c^* as before (i.e., an integer). Then for $i \in \{1, \ldots, \Theta\}$, set $z_i \leftarrow [c^* \cdot y_i]_2$, keeping only $n = \lceil \log \theta \rceil + 3$ bits of precision after the binary point for each z_i. Output both c^* and $\boldsymbol{z} = \langle z_1, \ldots, z_\Theta \rangle$.

Decrypt. Output $m' \leftarrow \left[c^* - \lfloor \sum_i s_i z_i \rceil \right]_2$.

Recall our definition of permitted polynomials from Section 3.2. We proved that our somewhat homomorphic scheme was correct for the set $C(\mathcal{P}_\mathcal{E})$ of circuit that compute permitted polynomials, and we now show that this is true also of the modified scheme.

Lemma 4. *The modified scheme from above is correct for $C(\mathcal{P}_\mathcal{E})$. Moreover, for every ciphertext $(c^*, \boldsymbol{z})$ that is generated by evaluating a permitted polynomial, it holds that $\sum s_i z_i$ is within 1/4 of an integer.*

Proof. Fix public and secret keys, generated with respect to security parameter λ, with $\{y_i\}_{i=1}^\Theta$ the rational numbers in the public key and $\{s_i\}_{i=1}^\Theta$ the secret-key bits. Recall that the y_i's were chosen so that $[\sum_i s_i y_i]_2 = (1/p) - \Delta_p$ with $|\Delta_p| \le 2^{-\kappa}$.

Fix a permitted polynomial $P(x_1, \ldots, x_t) \in \mathcal{P}_\mathcal{E}$, an arithmetic circuit C that computes P, and t ciphertexts $\{c_i\}_{i=1}^t$ that encrypt the inputs to C, and denote $c^* = \mathsf{Evaluate}(\text{pk}, C, c_1, \ldots, c_t)$. We need to establish that

$$\lfloor c^*/p \rceil = \left\lfloor \sum_i s_i z_i \right\rceil \pmod 2$$

where the z_i's are computed as $[c^* \cdot y_i]_2$ with only $\lceil \log \theta \rceil + 3$ bits of precision after the binary point, so $[c^* \cdot y_i]_2 = z_i - \Delta_i$ with $|\Delta_i| \le 1/16\theta$. We have

$$\begin{aligned}\left[(c^*/p) - \sum s_i z_i\right]_2 &= \left[(c^*/p) - \sum s_i [c^* \cdot y_i]_2 + \sum s_i \varDelta_i\right]_2 \\ &= \left[(c^*/p) - c^* \cdot \left[\sum s_i y_i\right]_2 + \sum s_i \varDelta_i\right]_2 \\ &= \left[(c^*/p) - c^* \cdot (1/p - \varDelta_p) + \sum s_i \varDelta_i\right]_2 \\ &= \left[c^* \cdot \varDelta_p + \sum s_i \varDelta_i\right]_2\end{aligned}$$

We claim that the final quantity inside the brackets has magnitude at most 1/8. By definition, since c^* is a valid ciphertext output by a permitted polynomial, the value c^*/p is within 1/8 of an integer. Together, these facts imply the lemma.

To establish the claim, observe that $|\sum s_i \varDelta_i| \leq \theta \cdot \frac{1}{16\theta} = 1/16$. Regarding $c^* \cdot \varDelta_p$, recall that the output ciphertext c^* is obtained by evaluating the polynomial P on the input ciphertexts c_i (as if P was an integer polynomial). By the definition of a permitted polynomial, for any $\alpha \geq 1$, if P's inputs have magnitude at most $2^{\alpha(\rho'+2)}$, its output has magnitude at most $2^{\alpha(\eta-4)}$ when its inputs have magnitude. In particular, when P's inputs are "fresh" ciphertexts, which have magnitude at most 2^γ, P's output ciphertext c^* has magnitude at most $2^{\gamma(\eta-4)/(\rho'+2)} < 2^{\kappa-4}$. Thus, $|c^* \cdot \varDelta_p| < 1/16$ and the claim follows.

6.2 Bootstrapping Achieved!

Theorem 3. *Let $\mathcal{E}$ be the scheme above, and let $D_\mathcal{E}$ be the set of augmented (squashed) decryption circuits. Then, $D_\mathcal{E} \subset C(\mathcal{P}_\mathcal{E})$.*

In other words, $\mathcal{E}$ is bootstrappable. The proof is similar to Gentry's [5,6]. By Theorem 1, we obtain homomorphic encryption schemes for circuits of any depth.

Proof. The goal is to express the modified decryption equation

$$m' \leftarrow c^* - \left\lfloor \sum s_i \cdot z_i \right\rceil \bmod 2$$

as a permitted polynomial (i.e., one satisfying Equation (1)), and show that there is a polynomial-size circuit that computes this polynomial. Recall that c^* is an integer, the s_i's are bits, and the z_i's are rational numbers in $[0, 2)$, in binary representation with $n = \lceil \log \theta \rceil + 3$ bits of precision after the binary point. Also, our parameter setting implies two promises – namely, that $\sum s_i \cdot z_i$ is within 1/4 of an integer, and that only θ of the bits $s_1, \ldots, s_\Theta$ are nonzero.

We split the computation up into three steps:

1. For $i \in \{1, \ldots, \Theta\}$, set $a_i \leftarrow s_i \cdot z_i$ (i.e., $a_i = z_i$ when $s_i = 1$ and $a_i = 0$ otherwise). The a_i's are still rational numbers in $[0, 2)$, given in binary representation with n bits of precision after the binary point.
2. From the Θ rational numbers $\{a_i\}_{i=1}^{\Theta}$, generate other $n+1$ rational numbers $\{w_j\}_{j=0}^{n}$, each with less than n bits of precision, such that $\sum_j w_j = \sum_i a_i$ (mod 2).
3. Output $c^* - (\sum_j w_j)$ mod 2.

The first step can be performed with a 1-level sub-circuit of multiplication gates. However, the second and third steps require more complicated sub-circuits.

The problem of using a shallow boolean circuit to compute the sum $\sum_{i=1}^{k} r_i$ of k rational numbers in binary representation is well-studied. A well-known technique uses the three-for-two trick (see [11]), whereby a constant-depth circuit is used to transform three numbers of arbitrary bit-length into two numbers that are at most 1 bit longer, such that the sum of the two output numbers is the same as the sum of the three input numbers. (The output bits of the constant-depth circuit are linear or quadratic expressions with 3 monomials in the input bits.) By applying this trick at most $\left\lceil \log_{3/2} k \right\rceil + 2$ times, one obtains two numbers s_1 and s_2 such that $s_1 + s_2 = \sum_{i=1}^{k} r_i$. Hence the total depth that it takes to reduce k numbers to two numbers is $d' \leq 2^{\left\lceil \log_{3/2} k \right\rceil + 2} < 8k^{1/\log(3/2)} < 8k^{1.71}$. The depth of the circuit needed to compute the final sum of two numbers is logarithmic in their bit-lengths, but if we are only interested in $\lfloor s_1 + s_2 \rceil \bmod 2$ and have the promise that $s_1 + s_2$ is within $1/4$ of an integer, this value can be computed by multivariate polynomial of degree 4 (and only nine terms). Overall, the circuit for computing $\left\lfloor \sum_{i=1}^{k} r_i \right\rceil \bmod 2$ corresponds to a polynomial of degree at most $d \leq 32k^{1/\log(3/2)}$. with coefficient vector having l_1-norm at most 27^d. Unfortunately, this degree (with $k = \Theta$) is still too large for our scheme to handle. Hence we use Gentry's technique from [5] that takes advantage of the fact that all but θ of the a_i's are zero.

Denote the bit representation of each number a_i by $a_{i,0\bullet}a_{i,-1}a_{i,-2}\dots a_{i,-n}$. That is, $a_i = \sum_{j=0}^{n} 2^{-j} a_{i,-j}$. The heart of this procedure is a subroutine for computing integers W_{-j}, $j = 0, 1, \dots, n$, where W_{-j} is the Hamming weight of the "column" of bits $(a_{1,-j}, a_{2,-j}, \dots, a_{\Theta,-j})$ (see an illustration in Figure 1). Since at most θ of the a_i's are nonzero, then the W_{-j}'s are no larger than θ, and hence can be represented by $\lceil \log(\theta + 1) \rceil < n$ bits. By Lemma 5 below, every

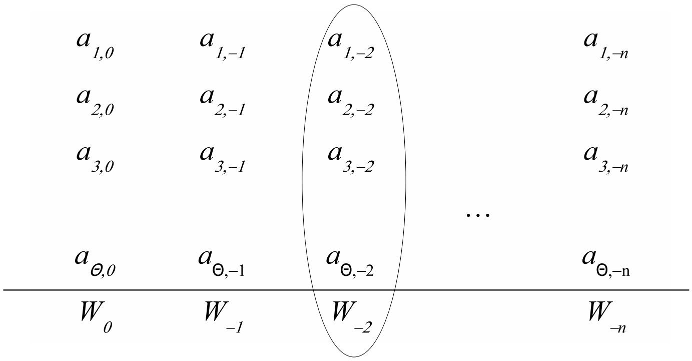


Fig. 1. The procedure for summing up the a_i's: The binary representation of the rational number a_i is $a_{i,0\bullet}a_{i,-1}a_{i,-2}\dots a_{i,-n}$. The integer W_{-j} is the Hamming weight of the column of bits $(a_{1,-j}, a_{2,-j}, \dots, a_{\Theta,-j})$.

bit in the binary representation of W_{-j} can be expressed as a polynomial of degree at most θ in the Θ variables $a_{i,-j}$, $i = 1, 2, \ldots, \Theta$. Moreover all of these polynomials can be computed simultaneously by an arithmetic circuit of size $O(\theta \cdot \Theta)$.

Once we have the W_{-j}'s, the sum of the a_i's can be obtained by $\sum_i a_i = \sum_j 2^{-j} W_{-j}$. For $j = 0, 1, \ldots, n$ we set $w_j = (2^{-j} \cdot W_{-j}) \bmod 2$, so the w_j's are rational numbers with $\lceil \log(\theta + 1) \rceil < n$ bits of precision. We can now sum-up the w_j's using the three-for-two trick as above, this time with $k = n + 1$, thus obtaining the sum of the a_i's mod 2.

We conclude that the degree of the polynomials in the first step is two, the degree of polynomials in the second step is at most θ, and the degree of the polynomial in the third step is at most

$$32(n+1)^{1/\log(3/2)} \ < \ 32 \left\lceil \log\theta + 4 \right\rceil^{1.71} \ < \ 32 \log^2 \theta$$

Therefore the total degree of the decryption circuit is bounded by $2 \cdot \theta \cdot 32 \log^2 \theta = 64\theta \log^2 \theta$, and since we are using $\theta = \lambda$ we have degree at most $64\lambda \log^2 \lambda$.

It follows that the augmented decryption circuits $D_{\mathcal{E}}$ (i.e., decryption followed by a single multiplication or addition, cf. Definition 4) can be expressed as polynomials of degree at most $128\lambda \log^2 \lambda$ in the Θ variables s_i. Since the logarithm of l_1-norm of this polynomial is small in relation to η, and since $\Theta = \frac{\eta\gamma}{\rho} \cdot \omega(\log \lambda) < \lambda^7$ (and also $\tau < \lambda^7$) the argument in Remark 4 at the end of Section 3.2 (with $\alpha = 128$ and $\beta = 7$) indicates that we can get $D_{\mathcal{E}} \subset C(\mathcal{P}_{\mathcal{E}})$, making the scheme bootstrappable, by setting $\eta = \rho \cdot \Theta(\lambda \log^2 \lambda)$.

It is left to show how to compute the W_j's using polynomials of degree no larger than θ.

Lemma 5. *Let $\boldsymbol{\sigma} = \langle \sigma_1, \sigma_2, \ldots, \sigma_t \rangle$ be a binary vector, let $W = W(\boldsymbol{\sigma})$ be the Hamming weight of $\boldsymbol{\sigma}$, and denote the binary representation of W by $W_n \ldots W_1 W_0$. (That is, $W = \sum_{i=0}^{n} 2^i W_i$ and all the W_i's are bits.)*

Then for every $i \leq n$, the bit $W_i(\boldsymbol{\sigma})$ can be expressed as a binary polynomial of degree exactly 2^i in the variables $\sigma_1, \ldots, \sigma_t$. Moreover, there is an arithmetic circuit of size $2^i \cdot t$ that simultaneously computes all the polynomials for $W_0, \ldots, W_i$.

Proof. It is well known that the i'th bit in the binary representation of the Hamming weight of bit-vector $\boldsymbol{\sigma}$ is equal to $e_{2^i}(\boldsymbol{\sigma})$ modulo 2, where $e_k(\cdot)$ is the k'th elementary symmetric polynomial, see Lemma 4 of [2]. That is,

$$W_i(\boldsymbol{\sigma}) \ = \ e_{2^i}(\boldsymbol{\sigma}) \bmod 2 \ = \ \left(\sum_{|S|=2^i} \ \prod_{j \in S} \sigma_j \right) \bmod 2$$

Clearly, the degree of e_{2^i} is exactly 2^i.

As for the "Moreover" part, we can compute the elementary symmetric polynomials in the σ_i's as the coefficients of the polynomial $P_{\boldsymbol{\sigma}}(z) = \prod_{i=1}^{t}(z - \sigma_i)$ in the auxiliary formal variable z, with $e_k(\boldsymbol{\sigma})$ being the coefficient of z^{t-k}. Conveniently, to compute only the first few bits $W_0, W_1, \ldots, W_i$, we can simply discard the lower-order terms in $P_{\boldsymbol{\sigma}}(z)$ – i.e., we do not need the coefficients of z^j for $j < t - 2^i$.

For example, one “dynamic programming” procedure for computing W_0, W_1, …, W_i (which can be trivially made into a circuit) would go as follows:

```
Input: bits $\sigma_1, \ldots, \sigma_t$
0. Initialization: Set $P_{0,0} \leftarrow 1$ and $P_{j,0} \leftarrow 0$ for $j = 1, 2, 3, \ldots, 2^i$
                        // $P_{j,k}$ is the $j$'th symmetric polynomial in $\sigma_1 \ldots \sigma_k$
1. For $k = 1, 2, \ldots, t$        // incorporate $\sigma_k$
2.      For $j = 2^i$ down to 1, set $P_{j,k} \leftarrow \sigma_k \times P_{j-1,k-1} + P_{j,k-1}$
3. Output $P_{1,t}, P_{2,t}, P_{4,t}, \ldots, P_{2^i,t}$
```

We can do a little better by using fast Fourier transform multiplication of polynomials. Using this technique, we can compute the entire polynomial $P_{\boldsymbol{\sigma}}(z)$ with complexity $t \cdot \text{polylog}(t)$.

Remark 6. Note that our first circuit implementation of the procedure from above is not “shallow”. Nonetheless, since it computes only “low degree polynomials” (i.e., up to degree 2^i), then by Lemma 3 it is a permitted circuit.

6.3 Security of the Squashed Scheme

Putting the hint $\boldsymbol{y}$ in the public key induces another computational assumption, related to the sparse subset sum problem (SSSP) used by Gentry [5], and studied previously (sometimes under the name “low-weight” knapsack) in the context of server-aided cryptography [16] and in connection to the Chor-Rivest cryptosystem [18]. We can easily avoid known attacks on the problem by choosing θ large enough to avoid brute-force attacks (and improvements using time-space trade-offs) and choosing Θ to be larger than $\omega(\log \lambda)$ times the bit-length of the rational numbers in the public key (which have length κ).[7]

7 Conclusion and Open Problems

We described a fully homomorphic encryption scheme that uses only simple integer arithmetic. The primary open problem is to improve the efficiency of the scheme, to the extent that it is possible while preserving the hardness of the approximate-gcd problem.

References

1. Alexi, W., Chor, B., Goldreich, O., Schnorr, C.-P.: Rsa and rabin functions: Certain parts are as hard as the whole. SIAM J. Comput. 17(2), 194–209 (1988)
2. Boyar, J., Peralta, R., Pochuev, D.: On the multiplicative complexity of boolean functions over the basis $(\wedge, \oplus, 1)$. Theor. Comput. Sci. 235(1), 43–57 (2000)

[7] Note that the SSSP instance and the approximate-GCD instance share the same integer p, but this is not a problem since SSSP is considered hard even if the attacker knows p.

3. Cachin, C., Micali, S., Stadler, M.: Computationally private information retrieval with polylogarithmic communication. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 402–414. Springer, Heidelberg (1999)
4. Coppersmith, D.: Small solutions to polynomial equations, and low exponent RSA vulnerabilities. J. Cryptology 10(4), 233–260 (1997)
5. Gentry, C.: A fully homomorphic encryption scheme. PhD thesis, Stanford University (2009), http://crypto.stanford.edu/craig
6. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC 2009, pp. 169–178. ACM, New York (2009)
7. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)
8. 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)
9. Howgrave-Graham, N.: Approximate integer common divisors. In: Silverman, J.H. (ed.) CaLC 2001. LNCS, vol. 2146, pp. 51–66. Springer, Heidelberg (2001)
10. Ishai, Y., Paskin, A.: Evaluating branching programs on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 575–594. Springer, Heidelberg (2007)
11. Karp, R.M., Ramachandran, V.: A Survey of Parallel Algorithms for Shared-Memory Machines. Technical Report CSD-88-408, UC Berkeley (1988)
12. Knuth, D.E.: *Seminumerical Algorithms*, 3rd edn. The Art of Computer Programming, vol. 2. Addison-Wesley, Reading (1997)
13. Lagarias, J.C.: The computational complexity of simultaneous diophantine approximation problems. SIAM J. Comput. 14(1), 196–209 (1985)
14. Lenstra, A.K.: Factoring multivariate polynomials over algebraic number fields. SIAM J. Comput. 16(3), 591–598 (1987)
15. Lindell, Y., Pinkas, B.: A proof of security of yao's protocol for two-party computation. J. Cryptology 22(2) (2009)
16. Nguyen, P.Q., Shparlinski, I.: On the insecurity of a server-aided RSA protocol. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 21–35. Springer, Heidelberg (2001)
17. Nguyen, P.Q., Stern, J.: The two faces of lattices in cryptology. In: Silverman, J.H. (ed.) CaLC 2001. LNCS, vol. 2146, pp. 146–180. Springer, Heidelberg (2001)
18. Nguyen, P.Q., Stern, J.: Adapting density attacks to low-weight knapsacks. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 41–58. Springer, Heidelberg (2005)
19. Regev, O.: New lattice-based cryptographic constructions. JACM 51(6), 899–942 (2004)
20. Rivest, R., Adleman, L., Dertouzos, M.: On data banks and privacy homomorphisms. In: Foundations of Secure Computation, pp. 169–177. Academic Press, London (1978)
21. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. Cryptology ePrint Archive, Report 2009/616 (2009), http://eprint.iacr.org/2009/616
22. Yao, A.C.: Protocols for secure computations. In: 23rd Annual Symposium on Foundations of Computer Science – FOCS 1982, pp. 160–164. IEEE, Los Alamitos (1982)

Converting Pairing-Based Cryptosystems from Composite-Order Groups to Prime-Order Groups

David Mandell Freeman*

Stanford University, USA
dfreeman@cs.stanford.edu

Abstract. We develop an abstract framework that encompasses the key properties of bilinear groups of composite order that are required to construct secure pairing-based cryptosystems, and we show how to use prime-order elliptic curve groups to construct bilinear groups with the same properties. In particular, we define a generalized version of the subgroup decision problem and give explicit constructions of bilinear groups in which the generalized subgroup decision assumption follows from the decision Diffie-Hellman assumption, the decision linear assumption, and/or related assumptions in prime-order groups.

We apply our framework and our prime-order group constructions to create more efficient versions of cryptosystems that originally required composite-order groups. Specifically, we consider the Boneh-Goh-Nissim encryption scheme, the Boneh-Sahai-Waters traitor tracing system, and the Katz-Sahai-Waters attribute-based encryption scheme. We give a security theorem for the prime-order group instantiation of each system, using assumptions of comparable complexity to those used in the composite-order setting. Our conversion of the last two systems to prime-order groups answers a problem posed by Groth and Sahai.

Keywords: pairing-based cryptography, composite-order groups, cryptographic hardness assumptions.

1 Introduction

Bilinear groups of composite order are a tool that has been used in the last few years to solve many problems in cryptography. The concept was introduced by Boneh, Goh, and Nissim [3], who applied the technique to the problems of private information retrieval, online voting, and universally verifiable computation. Subsequent authors have built on their work to create protocols such as non-interactive zero-knowledge proofs [13,14], ring and group signatures [6,20], attribute-based encryption [5,16], traitor tracing schemes [4], and hierarchical IBE [17,21].

* Research supported by a National Science Foundation International Research Fellowship, with additional support from the Office of Multidisciplinary Activities in the NSF Directorate for Mathematical and Physical Sciences.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 44–61, 2010.

Bilinear groups of composite order are pairs of abelian groups $(\mathbb{G}, \mathbb{G}_t)$, each of composite order $N = pq$, equipped with a nondegenerate bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$. Cryptosystems using bilinear groups of composite order usually base their security on variants of the *subgroup decision assumption.* Informally, this assumption says that given an element $g \in \mathbb{G}$, there is no efficient algorithm to determine whether g has order p. In particular, the assumption implies that it is infeasible to factor the group order N.

While the subgroup decision assumption is a useful tool for constructing secure protocols, it presents significant obstacles to implementing these protocols in practice. The only known instantiations of composite-order bilinear groups use elliptic curves (or more generally, abelian varieties) over finite fields. Since the elliptic curve group order n must be infeasible to factor, it must be at least (say) 1024 bits. On the other hand, the size of a prime-order elliptic curve group that provides an equivalent level of security is 160 bits [1]. As a result, group operations and especially pairing computations are prohibitively slow on composite-order curves: a Tate pairing on a 1024-bit composite-order elliptic curve is roughly 50 times slower than the same pairing on a comparable prime-order curve [18], and this performance gap will only get worse at higher security levels.

In short, requiring that the group order be infeasible to factor negates the principal advantage of elliptic curve cryptography over factoring-based systems, namely, that there is no known subexponential-time algorithm for computing discrete logarithms on an elliptic curve, while there is such an algorithm for factoring. Thus for efficient implementations we seek versions of protocols that use only prime-order elliptic curve groups. Developing these protocols is the main goal of this paper. In particular, we do the following:

- We **develop an abstract framework** that encompasses the key properties of bilinear groups of composite order, and we show how to use prime-order elliptic curves to construct bilinear groups with the same properties.
- We apply our framework and our prime-order construction to **create more efficient versions of cryptosystems** that originally used composite-order groups. Specifically, we consider:
    1. The Boneh-Goh-Nissim encryption scheme [3],
    2. The Boneh-Sahai-Waters traitor tracing system [4], and
    3. The Katz-Sahai-Waters attribute-based encryption scheme [16].

Our conversion of the last two systems to prime-order groups answers a problem posed by Groth and Sahai [14, Section 9], who themselves implicitly use our framework to construct non-interactive proof systems using either composite-order or prime-order groups.

Outline and Summary of Results. The starting point for our abstract framework is the fact that the subgroup decision assumption defined by Boneh, Goh, and Nissim depends only on the existence of a group G for which it is infeasible to determine if an element $g \in G$ lies in a given proper subgroup G_1 of G. This observation gives us a more general subgroup decision assumption in the language of abstract groups (see Section 2).

Our construction using prime-order groups is based on the observation, used implicitly by Cramer and Shoup [7] and articulated explicitly by Gjøsteen [12], that the decision Diffie-Hellman (DDH) assumption is a generalized subgroup decision assumption. Specifically, suppose we are given a cyclic group $\mathbb{G}$ and elements $g, g^a, g^b, g^c \in \mathbb{G}$. Then the DDH assumption for $\mathbb{G}$ says exactly that it is infeasible to determine whether (g^b, g^c) is in the cyclic subgroup of $\mathbb{G} \times \mathbb{G}$ generated by (g, g^a). Thus any protocol that requires two groups $G_1 \subset G$ in which the generalized subgroup decision assumption holds can be instantiated using $G = \mathbb{G} \times \mathbb{G}$ and $G_1 = \langle (g_1, g_2) \rangle$, where $\mathbb{G}$ is a cyclic group in which the DDH assumption holds and g_1, g_2 are random elements of $\mathbb{G}$.

More generally, we can use $G = \mathbb{G}^n$ for any $n \geq 2$ and let G_1 be a rank-k subgroup for any $1 \leq k < n$. In this case the subgroup decision assumption in G follows from the *k-Linear assumption* in $\mathbb{G}$, which generalizes the DDH assumption. In particular, the 1-Linear assumption is DDH, while the 2-Linear assumption is the *decision linear assumption.* This more general construction makes explicit a relationship noticed by several previous authors (e.g., [14,21]), namely, that functionality that can be achieved in composite-order groups under the subgroup decision assumption can also be achieved in prime-order groups under either the DDH or the decision linear assumption.

If the group $\mathbb{G}$ is equipped with a pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$, then applying $\hat{e}$ componentwise defines a pairing on $G = \mathbb{G}^n$. However, such a "symmetric" pairing (which only exists on supersingular elliptic curves) can be used to solve DDH in $\mathbb{G}$, so in this case our DDH-based construction is not secure. To get around this problem we use the fact that on ordinary (i.e., non-supersingular) elliptic curves there are two distinguished subgroups, denoted $\mathbb{G}_1$ and $\mathbb{G}_2$, in which DDH is believed to be infeasible for sufficiently large group orders. We can thus apply our construction twice to produce groups $G = \mathbb{G}_1^n$, $H = \mathbb{G}_2^n$, $G_t = \mathbb{G}_t^m$ (for some m), and an "asymmetric" pairing $e : G \times H \to G_t$. If the DDH assumption holds in $\mathbb{G}_1$ and $\mathbb{G}_2$, then the subgroup decision assumption holds in G and H. (If using the d-Linear assumption with $d \geq 2$, we can remain in the symmetric setting.)

While the security of composite-order group protocols depends on (variants of) the subgroup decision assumption, the correctness of these protocols depends on the groups having certain additional properties. In some cases, the groups G, H, G_t must be equipped with projection maps π_1, π_2, π_t that map them onto proper subgroups and commute with the pairing. In other cases, the groups must decompose into subgroups $G \cong \prod G_i$ and $H \cong \prod H_i$ such that the pairing restricted to $G_i \times H_j$ is trivial whenever $i \neq j$. In Section 3 we define these properties in our abstract framework and show how to instantiate them in the product groups $\mathbb{G}_1^n$, $\mathbb{G}_2^n$.

Sections 2 and 3 give us the framework and the tools necessary to convert composite-order group protocols to prime-order groups. Section 4 analyzes the efficiency gains realized in terms of the number of bits needed to represent group elements. For example, at a security level equivalent to 80-bit AES, ciphertexts in the Boneh-Goh-Nissim cryptosystem can be up to three times smaller when instantiated using our prime-order construction than in the original composite-order system. At the 256-bit security level the improvement can be as large as a factor of 12.

In Section 5 we describe in detail the conversion procedure for the Boneh-Goh-Nissim cryptosystem, and in Section 6 we sketch the same for the Boneh-Sahai-Waters traitor tracing system and the Katz-Sahai-Waters attribute-based encryption scheme. (Details are in the full version of this paper [10].) In each case we describe the scheme in our general framework and convert the assumptions used in the security proofs to our more general setting. We then consider the system instantiated with our prime-order group construction and give security theorems in this setting. If the original system is secure under a simple assumption (e.g., subgroup decision) then the converted scheme is also secure under a simple assumption (e.g., DDH); if the original system uses a complex assumption (as in the Katz-Sahai-Waters system) then the corresponding assumption in prime-order groups is also complex.

We note that our conversion process is not "black box": the security proof for each system must be analyzed to determine whether it carries over to our more general setting. For example, the recent IBE scheme of Lewko and Waters [17] uses explicitly in its security proof the fact that the group G has two subgroups of relatively prime order, and thus our techniques do not apply. However, we do expect that our framework can be used to convert to prime-order groups other cryptosystems originally built using composite-order groups.

2 Subgroup Decision Problems

The problem of determining whether a given element g of a finite group G lies in a specified proper subgroup G_1 was used as a hardness assumption for constructing cryptosystems long before Boneh, Goh, and Nissim defined their "subgroup decision problem." Gjøsteen [12] has undertaken an extensive survey of such problems, which he calls "subgroup membership problems." For example, the *quadratic residuosity problem* is a subgroup membership problem: if we let $N = pq$ be a product of two distinct primes and define the group G to be the group of elements of $\mathbb{Z}_N^*$ with Jacobi symbol 1, the problem is to determine whether a given element in G lies in the subgroup of squares in G.

Boneh, Goh, and Nissim [3] defined their problem for pairs of groups $(\mathbb{G}, \mathbb{G}_t)$ of composite order $N = pq$ for which there exists a nondegenerate bilinear map, or "pairing," $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$. The problem is to determine whether a given element $g \in \mathbb{G}$ is in the subgroup of order p. Note that if g' generates $\mathbb{G}$, then $e(g, g')$ is a challenge element for the same problem in $\mathbb{G}_t$; thus if the subgroup decision problem is infeasible in $\mathbb{G}$ then it is in $\mathbb{G}_t$ as well.

Our general notion of a subgroup decision problem extends Gjøsteen's work to the bilinear setting. We begin by defining an object that generates the groups we will work with. We assume that the two groups input to the pairing are not identical; this is known as an *asymmetric* pairing.

Definition 2.1. A *bilinear group generator* is an algorithm $\mathcal{G}$ that takes as input a security parameter λ and outputs a description of five abelian groups G, G_1, H, H_1, G_t, with $G_1 \subset G$ and $H_1 \subset H$. We assume that this description permits efficient (i.e., polynomial-time in λ) group operations and random sampling in each group. The algorithm also outputs an efficiently computable map (or "pairing") $e : G \times H \to G_t$ that is

- Bilinear: $e(g_1g_2, h_1h_2) = e(g_1, h_1)e(g_1, h_2)e(g_2, h_1)e(g_2, h_2)$ for all $g_1, g_2 \in G$, $h_1, h_2 \in H$; and
- Nondegenerate: for any $g \in G$, if $e(g, h) = 1$ for all $h \in H$, then $g = 1$ (and similarly with G, H reversed).

Our generalized subgroup decision assumption says that it is infeasible to distinguish an element in G_1 from a random element of G, and similarly for H. More precisely, we have the following definition. (The notation $x \overset{\text{R}}{\leftarrow} X$ means x is chosen uniformly at random from the set X.)

Definition 2.2. Let $\mathcal{G}$ be a bilinear group generator. We define the following distribution:

$$\mathbb{G} = (G, G_1, H, H_1, G_t, e) \overset{\text{R}}{\leftarrow} \mathcal{G}(\lambda), \ T_0 \overset{\text{R}}{\leftarrow} G, \ T_1 \overset{\text{R}}{\leftarrow} G_1.$$

We define the *advantage* of an algorithm $\mathcal{A}$ in solving the *subgroup decision problem on the left* to be

$$\text{SDP}_\text{L}\text{-Adv}[\mathcal{A}, \mathcal{G}] = \Big| \Pr[\mathcal{A}(\mathbb{G}, T_0) = 1] - \Pr[\mathcal{A}(\mathbb{G}, T_1) = 1] \Big|.$$

We say that $\mathcal{G}$ *satisfies the subgroup decision assumption on the left* if $\text{SDP}_\text{L}\text{-Adv}[\mathcal{A}, \mathcal{G}](\lambda)$ is a negligible function of λ for any polynomial-time algorithm $\mathcal{A}$. We define the *subgroup decision problem/assumption on the right* and $\text{SDP}_\text{R}\text{-Adv}[\mathcal{A}, \mathcal{G}]$ analogously, with $T_0 \overset{\text{R}}{\leftarrow} H$ and $T_1 \overset{\text{R}}{\leftarrow} H_1$. We say $\mathcal{G}$ *satisfies the subgroup decision assumption* if it satisfies both the left and right assumptions.

Example 2.3 ([3, Section 2.1]). Boneh, Goh, and Nissim construct a bilinear group generator $\mathcal{G}_{BGN}$ using supersingular elliptic curves of composite order. Let $\mathcal{E}(\lambda)$ be an algorithm that outputs a product $N = p_1p_2$ of two distinct primes greater than 2^λ, a prime $q \equiv -1 \pmod{N}$, and a supersingular elliptic curve E over the finite field $\mathbb{F}_q$. Then $\#E(\mathbb{F}_q)$ is divisible by N, and we can construct $\mathcal{G}_{BGN}(\lambda)$ by running $\mathcal{E}(\lambda)$ and setting the output as follows:

- $G = H$ is the order-N subgroup of $E(\mathbb{F}_q)$;
- $G_1 = H_1$ is the order-p_1 subgroup of $E(\mathbb{F}_q)$;
- G_t is the order-N subgroup of $\mathbb{F}_{q^2}^*$; and
- $e : G \times G \to G_t$ is the modified N-Tate pairing on E [8, Sect. 2.1].

Each group is described by giving a generator.

It is believed that $\mathcal{G}_{BGN}$ satisfies the subgroup decision assumption when N is infeasible to factor. The construction can be extended to produce a group G whose order is a product of three or more primes, and the subgroup decision assumption is believed to hold in any nontrivial proper subgroup G_1 of G. Using the generic group analysis of Katz, Sahai, and Waters [16, Theorem A.2], one can show that any efficient generic algorithm to solve the subgroup decision problem for $\mathcal{G}_{BGN}$ can be used construct an efficient algorithm to factor N.

2.1 Product Groups, DDH, and *d*-Linear Assumptions

The primary motivation for our abstraction of composite-order group protocols is the observation that the decision Diffie-Hellman problem is also a subgroup decision problem [12, Section 4.5].

Let $\mathbb{G}$ be a finite cyclic group, and let $T = (g, g^a, g^b, g^c)$ be a 4-tuple of elements in $\mathbb{G}$. The *decision Diffie-Hellman (DDH) problem* is to determine whether $c \equiv ab \pmod{|g|}$; if this is infeasible then we say that $\mathbb{G}$ satisfies the *decision Diffie-Hellman assumption*. Now suppose we are given a DDH challenge T. Define G to be $\mathbb{G} \times \mathbb{G}$ and G_1 to be the cyclic subgroup of G generated by (g, g^a). Then the element $(g^b, g^c) \in G$ is in G_1 if and only if $c \equiv ab \pmod{|g|}$ — so solving the subgroup decision problem for $G_1 \subset G$ is exactly equivalent to solving DDH in $\mathbb{G}$.

Now we consider the same construction in the bilinear setting: let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t$ be finite cyclic groups, and let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_t$ be a nondegenerate bilinear map. Then we can define $G = \mathbb{G}_1^2$, $H = \mathbb{G}_2^2$, and $G_t = \mathbb{G}_t^2$, and choose random elements of G and H to generate G_1 and H_1 respectively. We can define a nondegenerate pairing $e : G \times H \to \mathbb{G}_t$ by taking any invertible matrix $A = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{Mat}_2(\mathbb{F}_p)$ and setting

$$e((g_1, g_2), (h_1, h_2)) := e(g_1, h_1)^a e(g_1, h_2)^b e(g_2, h_1)^c e(g_2, h_2)^d.$$

We can define a pairing mapping to $G_t = \mathbb{G}_t^m$ by choosing different coefficients a, b, c, d to define each component of the output. With this setup, if the DDH assumption holds in $\mathbb{G}_1$ and $\mathbb{G}_2$, then the subgroup decision assumption holds for $G_1 \subset G$ and $H_1 \subset H$.

More generally, we consider a bilinear group generator $\mathcal{G}_k^n$ that produces two groups $G = \mathbb{G}_1^n$ and $H = \mathbb{G}_2^n$ and random rank-k subgroups $G_1 \subset G$ and $H_1 \subset H$. In this situation the natural analogue of the DDH problem is the *k-Linear problem*, introduced by Hofheinz and Kiltz [15] and Shacham [19]. The 1-Linear problem is simply DDH, while the 2-Linear problem is called the *decision linear problem* and was originally proposed by Boneh, Boyen, and Shacham [2] as a reasonable analogue for DDH in a group with a bilinear map.

The following definition and theorem formalize the relationship between subgroup decision problems and d-Linear problems. We will use the following notation: if we have a group $\mathbb{G}$ of order p, an element $g \in \mathbb{G}$, and a vector $\vec{x} = (x_1, \ldots, x_n) \in \mathbb{F}_p^n$, then we define $g^{\vec{x}} := (g^{x_1}, \ldots, g^{x_n}) \in \mathbb{G}^n$.

Definition 2.4. A bilinear group generator $\mathcal{P}$ is *prime-order* if the groups G, G_1, H, H_1, G_t all have prime order $p > 2^\lambda$. Then we have $G = G_1$ and $H = H_1$, and we denote the three distinct groups by $\mathbb{G}_1 = G$, $\mathbb{G}_2 = H$, and $\mathbb{G}_t = G_t$. We let $\hat{\mathbb{G}}$ denote the output $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e)$ of $\mathcal{P}(\lambda)$.

Let $d \geq 1$ be an integer. If $\mathcal{A}$ is an algorithm that takes as input $2d+2$ elements of $\mathbb{G}_1$, we define the *advantage* of $\mathcal{A}$ in solving the *d-Linear problem in* $\mathbb{G}_1$, denoted d-$\mathrm{Lin}_{\mathbb{G}_1}$-$\mathrm{Adv}[\mathcal{A}, \mathcal{P}]$, to be

$$\left| \Pr\left[\mathcal{A}(\hat{\mathbb{G}}, g_1, \dots, g_d, g_1^{r_1}, \dots, g_d^{r_d}, h, h^{r_1+\dots+r_d}) = 1 : \substack{\hat{\mathbb{G}} \xleftarrow{R} \mathcal{P},\ g_1,\dots,g_d \xleftarrow{R} \mathbb{G}_1, \\ r_1,\dots,r_d \xleftarrow{R} \mathbb{F}_p}\right] \right.$$
$$\left. - \Pr\left[\mathcal{A}(\hat{\mathbb{G}}, g_1, \dots, g_d, g_1^{r_1}, \dots, g_d^{r_d}, h, h^{s}) = 1 : \substack{\hat{\mathbb{G}} \xleftarrow{R} \mathcal{P},\ g_1,\dots,g_d \xleftarrow{R} \mathbb{G}_1, \\ r_1,\dots,r_d, s \xleftarrow{R} \mathbb{F}_p}\right] \right|,$$

and similarly for d-$\mathrm{Lin}_{\mathbb{G}_2}$-$\mathrm{Adv}[\mathcal{A}, \mathcal{P}]$. We say that $\mathcal{G}$ *satisfies the d-Linear assumption in* $\mathbb{G}_1$ if d-$\mathrm{Lin}_{\mathbb{G}_1}$-$\mathrm{Adv}[\mathcal{A}, \mathcal{G}](\lambda)$ is a negligible function of λ for any polynomial-time algorithm $\mathcal{A}$ (and similarly for $\mathbb{G}_2$). The *decision Diffie-Hellman (DDH) assumption* is the 1-Linear assumption. The *decision linear assumption* is the 2-Linear assumption.

Some previous authors (e.g., [14]) have called the assumption that DDH is infeasible in both $\mathbb{G}_1$ and $\mathbb{G}_2$ the *symmetric external Diffie-Hellman assumption*, or SXDH. For clarity in our arguments, we prefer to call the problems DDH in $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively.

Theorem 2.5. *Let $\mathcal{P}$ be a prime-order bilinear group generator. For integers n, k with $n \geq 2$ and $1 \leq k < n$, define $\mathcal{G}_k^n$ to be a bilinear group generator that on input λ does the following:*

1. *Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, \hat{e}) \xleftarrow{R} \mathcal{P}(\lambda)$.*
2. *Let $G = \mathbb{G}_1^n$, $H = \mathbb{G}_2^n$, $G_t = \mathbb{G}_t^m$ for some m.*
3. *Choose generators $g \xleftarrow{R} \mathbb{G}_1$, $h \xleftarrow{R} \mathbb{G}_2$.*
4. *Choose random $\vec{x}_i, \vec{y}_i \xleftarrow{R} \mathbb{F}_p^n$ for $i = 1, \dots, k$, such that the sets $\{\vec{x}_i\}$ and $\{\vec{y}_i\}$ are each linearly independent.*
5. *Let G_1 be the subgroup of G generated by $\{g^{\vec{x}_1}, \dots, g^{\vec{x}_k}\}$ and H_1 be the subgroup of H generated by $\{h^{\vec{y}_1}, \dots, h^{\vec{y}_k}\}$*
6. *Choose nonzero $n \times n$ matrices $A_\ell = (a_{ij}^{(\ell)})$ for $\ell = 1, \dots, m$.*
7. *Define $e : G \times H \to G_t$ by $e((g_1, \dots, g_n), (h_1, \dots, h_n))_\ell := \prod e(g_i, h_j)^{a_{ij}^{(\ell)}}$.*
8. *Output the tuple $\Gamma_k^n = (G, G_1, H, H_1, G_t, e)$.*

If $\mathcal{P}$ satisfies the k-Linear assumption in $\mathbb{G}_1$ and $\mathbb{G}_2$, then $\mathcal{G}_k^n$ satisfies the subgroup decision assumption. Specifically, for any adversary $\mathcal{A}$ that solves the subgroup decision problem on the left for $\mathcal{G}_k^n$, there exists an adversary $\mathcal{B}$ that solves the k-Linear problem in $\mathbb{G}_1$ for $\mathcal{P}$, with

$$\mathrm{SDP_L}\text{-}\mathrm{Adv}[\mathcal{A}, \mathcal{G}_k^n] \leq (n-k) \cdot k\text{-}\mathrm{Lin}_{\mathbb{G}_1}\text{-}\mathrm{Adv}[\mathcal{B}, \mathcal{P}].$$

An analogous statement holds for $\mathcal{A}$ solving the subgroup decision problem on the right for $\mathcal{G}_k^n$ and $\mathcal{B}$ solving the k-Linear problem in $\mathbb{G}_2$ for $\mathcal{P}$.

Proof sketch. We sketch the proof for $n = k+1$; the general case is proved in the full paper [10]. Let $(\hat{\mathbb{G}}, u_1, \ldots, u_k, v_1, \ldots, v_k, y, z)$ be a k-Linear challenge in $\mathbb{G}_1$. Let $\vec{x}_i = (x_{i,1}, \ldots, x_{i,n})$ be the vectors chosen in Step (4) above. Choose $\vec{b} \xleftarrow{R} \mathbb{F}_p^k$, and let G_1 be the subgroup generated by

$$\left\{(u_i^{x_{i,1}}, \ldots, u_i^{x_{i,k}}, v_i^{1/b_i})\right\}_{i=1}^{k}.$$

Now consider $T = (y^{\sum b_i x_{i,1}}, \ldots y^{\sum b_i x_{i,k}}, z) \in \mathbb{G}_1^n$, where each sum in the exponent runs over $i = 1$ to k. Write $v_i = u_i^{r_i}$, $z = y^c$. If $c = \sum_i r_i \pmod{p}$ then T is uniformly distributed in G_1, while if c is random then T is uniformly distributed in $\mathbb{G}_1^n$. It follows that any algorithm that has advantage ε in solving the subgroup decision problem for $\mathcal{G}_k^n$ can solve the k-Linear problem in $\mathbb{G}_1$ with advantage at least ε. □

Since the d-Linear assumption implies the $(d+1)$-Linear assumption for all $d \geq 1$ [15, Lemma 3], if $\mathcal{P}$ satisfies the DDH assumption in $\mathbb{G}_1$ and $\mathbb{G}_2$, then $\mathcal{G}_k^n$ satisfies the subgroup decision assumption for any $n \geq 1$ and $1 \leq k < n$. The converse holds when $k = 1$; the proof is in the full paper.

If we view all of the groups in the above construction as $\mathbb{F}_p$-vector spaces, then we see that the subgroup decision problem is a decisional version of the *vector decomposition problem* [22,11], in which the adversary is given a decomposition $G \cong G_1 \times G_2$ and an element $x \in G$ and asked to find $y \in G_1$ and $z \in G_2$ such that $x = yz$.

The nondegeneracy of the pairing e defined on $\mathcal{G}_k^n$ will depend on the matrices A_ℓ and must be checked in each case. However, if $m = 1$ then e is nondegenerate if and only if A_1 is invertible.

3 Pairings on Product Groups

In our construction of the bilinear group generator $\mathcal{G}_k^n$ from the prime-order bilinear group generator $\mathcal{P}$, we took the pairing e on the product groups to be any nontrivial linear combination of the componentwise pairings on the underlying prime-order group. However, the correctness proofs for protocols built in composite-order groups all use the fact that the pairings have some extra structure that arbitrary linear combinations are unlikely to have. We now investigate this structure further and determine how to replicate it in our product group context.

Projecting Pairings. The cryptosystem of Boneh, Goh, and Nissim works by taking elements $g \in G$ and $h \in G_1$ and encrypting a message m as $C = g^m h^r$, where r is random. The h term is a "blinding term" used to hide the part of the ciphertext that contains the message. Decryption is achieved by "projecting" the ciphertext away from the blinding term and taking a discrete logarithm to recover m. Specifically, when g has order $N = p_1 p_2$ and h has order p_1, the decryption can be achieved by first computing C^{p_1} to remove the h term, and then taking the discrete logarithm to the base g^{p_1} to recover m. The functionality of the cryptosystem requires that we can do this procedure either before or after the pairing; i.e., that we can construct and remove blinding terms in G_t. The following definition incorporates this concept into our abstract framework.

Definition 3.1. Let $\mathcal{G}$ be a bilinear group generator (Def. 2.1). We say $\mathcal{G}$ is *projecting* if it also outputs a group $G'_t \subset G_t$ and group homomor-phisms π_1, π_2, π_t mapping G, H, G_t to themselves, respectively, such that

1. G_1, H_1, G'_t are contained in the kernels of π_1, π_2, π_t, respectively, and
2. $e(\pi_1(g), \pi_2(h)) = \pi_t(e(g,h))$ for all $g \in G$, $h \in H$.

Example 3.2. The bilinear group generator $\mathcal{G}_{BGN}$ of Example 2.3 is projecting: we let G'_t be the order-p_1 subgroup of G_t, let $\pi_1 = \pi_2$ be exponentiation by p_1, and let π_t be exponentiation by $(p_1)^2$.

Given a prime-order bilinear group generator $\mathcal{P}$, we wish to modify the bilinear group generator $\mathcal{G}_k^n$ constructed in Theorem 2.5 so it is projecting. To do so, we interpret the generation of G_1 and H_1 in terms of matrix actions, and we define the pairing e using a *tensor product* of matrices.

We begin by defining the projection maps π_1 and π_2. Let $G = \mathbb{G}_1^n$ and let g be a generator of $\mathbb{G}_1$. For $i = 1, \dots, n$, let $\vec{e}_i$ be the unit vector with a 1 in the ith place and zeroes elsewhere. To construct the projection map π_1, we first observe that if G'_1 is the subgroup of G generated by $g^{\vec{e}_1}, \dots, g^{\vec{e}_k}$, then any element of G'_1 has 1's in the last $n-k$ coordinates, so we can define a projection map π'_1 whose kernel is G'_1 by

$$\pi'_1(g_1, \dots, g_n) := (1, \dots, 1, g_{n-k+1}, \dots, g_n).$$

Next we observe that the elements $g^{\vec{x}_1}, \dots, g^{\vec{x}_k}$ produced by $\mathcal{G}_k^n$ can be viewed as coming from a (right) action of an $n \times n$ matrix on the elements $g^{\vec{e}_1}, \dots, g^{\vec{e}_k}$. More precisely, for $\mathbf{g} = (g_1, \dots, g_n) \in G$ and a matrix $M = (a_{ij}) \in \mathrm{Mat}_n(\mathbb{F}_p)$, we define $\mathbf{g}^M$ by

$$\mathbf{g}^M := \left(\textstyle\prod_{i=1}^n g_i^{a_{i1}}, \dots, \prod_{i=1}^n g_i^{a_{in}}\right).$$

With this definition, we have $(g^{\vec{x}})^M = g^{(\vec{x}M)}$.

Now let M be an invertible matrix whose first k rows are the vectors $\vec{x}_i$. Then $g^{\vec{x}_i} = g^{\vec{e}_i M}$. If we define U_k to be the matrix with 1's in the last $n-k$ diagonal places and zeroes elsewhere, then the map π'_1 is given by $\pi'_1(\mathbf{g}) = \mathbf{g}^{U_k}$. Thus we can construct a projection map π_1 on G_1 by applying M^{-1} to map to G'_1, using π'_1 to project, and acting by M to map back to G_1; that is, $\pi_1(\mathbf{g}) = \mathbf{g}^{M^{-1}U_k M}$. We define π_2 analogously on H by computing an invertible matrix M' whose first k rows are the $\vec{y}_i$ produced by $\mathcal{G}_k^n$.

We now define the pairing e, the subgroup G'_t, and the projection map π_t. Recall that the *tensor product* of two n-dimensional vectors $\vec{x}, \vec{y}$ is

$$\vec{x} \otimes \vec{y} = (x_1\vec{y}, \dots, x_n\vec{y}) = (x_1y_1, \dots, x_1y_n, \dots, x_ny_1, \dots, x_ny_n).$$

We define $e : G \times H \to G_t := \mathbb{G}_t^{n^2}$ by $e(g^{\vec{x}}, h^{\vec{y}}) := \hat{e}(g,h)^{\vec{x}\otimes\vec{y}}$. That is, to pair $\mathbf{g} \in G$ and $\mathbf{h} \in H$, we take all the n^2 componentwise pairings $e(g_i, h_j)$ and write them in lexicographical order. In this case the A_ℓ of Theorem 2.5 are the n^2 matrices with a 1 in entry (i,j) and zeroes elsewhere; it is easy to see that these A_ℓ definite a nondegenerate pairing e.

Defining the pairing in this manner allows us to define the map π_t abstractly as the tensor product of the maps π_1 and π_2. In terms of the matrices we have defined, we have

$$\pi_t(\mathbf{g}_t) = \mathbf{g}_t^{(M^{-1}\otimes M'^{-1})(U_k \otimes U_k)(M\otimes M')},$$

where $\otimes$ indicates the tensor product (or *Kronecker product*) of matrices: if $A = (a_{ij})$ and $B = (b_{ij})$ are two $n \times n$ matrices, then $A \otimes B$ is the $n^2 \times n^2$ matrix which, when divided into $n \times n$ blocks, has the (i, j)th block equal to $a_{ij}B$.

Given this framework, we see that the constructions of Groth and Sahai [14, Section 5] are exactly the above with $(n, k) = (2, 1)$ and $(3, 2)$. We now give explicit details for the first case.

Example 3.3. Let $\mathcal{P}$ be a prime-order bilinear group generator. Define $\mathcal{G}_P$ to be a bilinear group generator that on input λ does the following:

1. Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, \hat{e}) \overset{\text{R}}{\leftarrow} \mathcal{P}(\lambda)$, and let $G = \mathbb{G}_1^2$, $H = \mathbb{G}_2^2$, $G_t = \mathbb{G}_t^4$.
2. Choose generators $g \overset{\text{R}}{\leftarrow} \mathbb{G}_1$, $h \overset{\text{R}}{\leftarrow} \mathbb{G}_2$, and let $\gamma = e(g, h)$.
3. Choose random $a_1, b_1, c_1, d_1, a_2, b_2, c_2, d_2 \overset{\text{R}}{\leftarrow} \mathbb{F}_p$, such that $a_1d_1 - b_1c_1 = a_2d_2 - b_2c_2 = 1$.
4. Let G_1 be the subgroup of G generated by (g^{a_1}, g^{b_1}), let H_1 be the subgroup of H generated by (h^{a_2}, h^{b_2}), and let G_t' be the subgroup of G_t generated by
$$\{\gamma^{(a_1a_2,a_1b_2,b_1a_2,b_1b_2)}, \gamma^{(a_1c_2,a_1d_2,b_1c_2,b_1d_2)}, \gamma^{(c_1a_2,d_1b_2,c_1a_2,d_1b_2)}\}.$$
5. Define $e : G \times H \to G_t$ by
$$e((g_1, g_2), (h_1, h_2)) := (\hat{e}(g_1, h_1), \hat{e}(g_1, h_2), \hat{e}(g_2, h_1), \hat{e}(g_2, h_2)).$$
6. Let $A = \left(\begin{smallmatrix} -b_1c_1 & -b_1d_1 \\ a_1c_1 & a_1d_1 \end{smallmatrix}\right)$, $B = \left(\begin{smallmatrix} -b_2c_2 & -b_2d_2 \\ a_2c_2 & a_2d_2 \end{smallmatrix}\right)$, and define
$$\begin{aligned} \pi_1((g_1, g_2)) &:= (g_1, g_2)^A = (g_1^{-b_1c_1} g_2^{a_1c_1}, g_1^{-b_1d_1} g_2^{a_1d_1}) \\ \pi_2((h_1, h_2)) &:= (h_1, h_2)^B = (h_1^{-b_2c_2} h_2^{a_2c_2}, h_1^{-b_2d_2} h_2^{a_2d_2}) \\ \pi_t((\gamma_1, \gamma_2, \gamma_3, \gamma_4)) &:= (\gamma_1, \gamma_2, \gamma_3, \gamma_4)^{A\otimes B} \end{aligned}$$
7. Output the tuple $(G, G_1, H, H_1, G_t, G_t', e, \pi_1, \pi_2, \pi_t)$.

It is easy (though tedious) to check that $\mathcal{G}_P$ is a projecting bilinear group generator. We note that the groups output by $\mathcal{G}_P$ can be described simply by giving $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t$ and the pairs (g^{a_1}, g^{b_1}), (h^{a_2}, h^{b_2}). In particular, the group G_t' is generated by elements of the form $e(\mathbf{g}, \mathbf{h}_1)$ and $e(\mathbf{g}_1, \mathbf{h})$ with $\mathbf{g} \in G$, $\mathbf{g}_1 \in G_1$, $\mathbf{h} \in H$, and $\mathbf{h}_1 \in H_1$. This is important since in applications the maps π_1, π_2, π_t will be "trapdoor" information used as the system's secret key.

Proposition 3.4. *If $\mathcal{P}$ satisfies the DDH assumption in $\mathbb{G}_1$ and $\mathbb{G}_2$, then $\mathcal{G}_P$ satisfies the subgroup decision assumption.*

Proof. Since g is uniform in $\mathbb{G}_1$ and a_1, b_1, c_1 are uniformly random in $\mathbb{F}_p$, imposing the condition $a_1d_1 - b_1c_1 = 1$ does not introduce any deviations from uniformity in the generation of G_1 (and similar for G_2). We can thus apply Theorem 2.5 with $n = 2$, $k = 1$. □

Cancelling Pairings. The traitor-tracing scheme of Boneh, Sahai, and Waters [4], the predicate encryption scheme of Katz, Sahai, and Waters [16], and many other schemes based on bilinear groups of composite order use in an essential manner the fact that if two group elements g, h have relatively prime orders, then $e(g, h) = 1$. This property implies, for example, that we can use the two subgroups generated by g and h to encode different types of information, and the two components will remain distinct after the pairing operation. The following definition incorporates this concept into our framework.

Definition 3.5. Let $\mathcal{G}$ be a bilinear group generator (Definition 2.1). We say that $\mathcal{G}$ is *r-cancelling* if it also outputs groups $G_2, \dots, G_r \subset G$ and $H_2, \dots, H_r \subset H$, such that

1. $G \cong G_1 \times \cdots \times G_r$ and $H \cong H_1 \times \cdots \times H_r$,
2. $e(g_i, h_j) = 1$ whenever $g_i \in G_i$, $h_j \in H_j$, and $i \neq j$.

Example 3.6. The bilinear group generator $\mathcal{G}_{BGN}$ of Example 2.3 is 2-cancelling: we set $G_2 = H_2$ to be the order-p_2 subgroup of $E(\mathbb{F}_p)$. An analogous r-cancelling generator can be built by making the group order N a product of r distinct primes.

Given a prime-order bilinear group generator $\mathcal{P}$, we now show how to modify the bilinear group generator $\mathcal{G}_1^n$ constructed in Theorem 2.5 so it is n-cancelling. We define the pairing $e : G \times H \to G_t := \mathbb{G}_t$ to be

$$e((g_1, \dots, g_n), (h_1, \dots, h_n)) := \textstyle\prod_{i=1}^n \hat{e}(g_i, h_i), \tag{3.1}$$

so we have $e(g^{\vec{x}}, h^{\vec{y}}) = e(g, h)^{\vec{x} \cdot \vec{y}}$, where $\cdot$ indicates the vector dot product; this pairing is necessarily nondegenerate.

If $\mathcal{G}_1^n$ is n-cancelling, then the subgroups G_i, H_i are all cyclic of order p. Thus we need to choose generators $g^{\vec{x}_i}$ of G_i and $h^{\vec{y}_i}$ of H_i such that $\vec{x}_i \cdot \vec{y}_j = 0$ if and only if $i = j$. This is straightforward: we first choose any set of n linearly independent $\vec{x}_i$; then the equation $\vec{x}_i \cdot \vec{y}_j = 0$ for all $i \neq j$ gives a linear system n variables of rank $n - 1$, so there is a one-dimensional solution space in $\mathbb{F}_p^n$. If we choose $\vec{y}_j$ in this space then with high probability we have $\vec{x}_j \cdot \vec{y}_j \neq 0$; if this is not the case then we can start again with a different set of $\vec{x}_i$. We illustrate with concrete examples for $n = 2$ and 3. We use the notation $\langle X \rangle$ to indicate the cyclic group generated by X.

Example 3.7. Let $\mathcal{P}$ be a prime-order bilinear group generator. Define $\mathcal{G}_{3C}$ to be a bilinear group generator that on input λ does the following:

1. Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, \hat{e}) \xleftarrow{\text{R}} \mathcal{P}(\lambda)$, and let $G = \mathbb{G}_1^3$, $H = \mathbb{G}_2^3$, $G_t = \mathbb{G}_t$.
2. Choose generators $g_1, g_2, g_3 \xleftarrow{\text{R}} \mathbb{G}_1$, $h_1, h_2, h_3 \xleftarrow{\text{R}} \mathbb{G}_2$.
3. Choose random $x, y, z, u, v, w \xleftarrow{\text{R}} \mathbb{F}_p$, with $\begin{cases} -xv - xw - yu + yw + zu + zv \neq 0, \\ xv - xw - yu + yw + zu - zv \neq 0. \end{cases}$
4. Define the subgroups

$$G_1 = \langle (g_1, g_1^x, g_1^u) \rangle,\ G_2 = \langle (g_2, g_2^y, g_2^v) \rangle,\ G_3 = \langle (g_3, g_3^z, g_3^w) \rangle,$$
$$H_1 = \langle (h_1^{zv-yw}, h_1^{w-v}, h_1^{y-z}) \rangle,\ H_2 = \langle (h_2^{zu-xw}, h_2^{w-u}, h_2^{z-x}) \rangle,$$
$$H_3 = \langle (h_3^{yu-xv}, h_3^{v-u}, h_3^{x-y}) \rangle.$$

5. Define $e : G \times H \to G_t$ by (3.1) (with $n = 3$).
6. Output the tuple $(G, G_1, G_2, G_3, H, H_1, H_2, H_3, G_t, e)$.

It is straightforward to show that $\mathcal{G}_{3C}$ is a 3-cancelling bilinear group generator. The inequalities in Step (3) guarantee non-degeneracy of the pairing e. Note that choosing the elements g_1, g_2, g_3 independently uniform allows us to scale the vectors $\vec{x}_1 = (1, x, u)$, $\vec{x}_2 = (1, y, v)$, $\vec{x}_3 = (1, z, w)$ so their first components are 1 without losing uniformity.

Example 3.8. We define a 2-cancelling bilinear group generator $\mathcal{G}_{2C}$ by restricting the construction in Example 3.7 to the first two components. Explicitly, we have $G = \mathbb{G}_1^2$, $H = \mathbb{G}_2^2$, $G_t = \mathbb{G}_t$ and we set $u = 0$, $v = 0$, $w = 1$ to obtain

$$G_1 = \langle (g_1, g_1^x) \rangle,\ G_2 = \langle (g_2, g_2^y) \rangle,\ H_1 = \langle (h_1^{-y}, h_1) \rangle,\ H_2 = \langle (h_2^{-x}, h_2) \rangle.$$

We define $e : G \times H \to G_t$ by (3.1) and output $(G, G_1, G_2, H, H_1, H_2, G_t, e)$.

Example 3.9. If we have a symmetric pairing (i.e. $\mathbb{G}_1 = \mathbb{G}_2$), then for any $n > k > 1$ we can obtain an $(n - k + 1)$-cancelling bilinear group generator $\mathcal{G}_L(n, k)$ by doing the following:

1. Let $(p, \mathbb{G}, \mathbb{G}_t, \hat{e}) \xleftarrow{R} \mathcal{P}(\lambda)$, and let $G = H = \mathbb{G}^n$, $G_t = \mathbb{G}_t$.
2. Choose $\vec{x}_1, \ldots, \vec{x}_n \xleftarrow{R} \mathbb{F}_p^n$, such that $\{\vec{x}_i\}$ is linearly independent and for all $i > k$ we have $\vec{x}_i \cdot \vec{x}_j = 0$ if $i \neq j$, and $\vec{x}_i \cdot \vec{x}_j \neq 0$ if $i = j$.
3. Choose a generator $g \xleftarrow{R} \mathbb{G}$, and let $\gamma_i = g^{\vec{x}_i} \in G$.
4. Let $G_1 = \langle \gamma_1, \ldots, \gamma_k \rangle$, and $G_i = \langle \gamma_{i+k-1} \rangle$ for $2 \leq i \leq n - k + 1$.
5. Define e by (3.1) and output $(G, G_1, \ldots, G_{n-k+1}, G_t, e)$.

Proposition 3.10. *If $\mathcal{P}$ satisfies the DDH assumption in $\mathbb{G}_1$ and $\mathbb{G}_2$, then $\mathcal{G}_{3C}$ and $\mathcal{G}_{2C}$ satisfy the subgroup decision assumption. If $\mathbb{G}_1 = \mathbb{G}_2$ and $\mathcal{P}$ satisfies the k-linear assumption in $\mathbb{G}_1$, then $\mathcal{G}_L(n, k)$ satisfies the subgroup decision assumption.*

Proof. Recall that an SDP adversary is given only G, G_1, H, H_1, and not a description of any G_i or H_i for $i \geq 2$. Since in each case the generators of G_1 and H_1 are independent and uniform, the outputs of $\mathcal{G}_{3C}$, $\mathcal{G}_{2C}$, and $\mathcal{G}_L(n, k)$ are distributed identically to the output of $\mathcal{G}_k^n$ (for the appropriate values of n, k) so we may apply Theorem 2.5. □

4 Performance Analysis

Our primary motivation for converting composite-order group protocols to prime-order groups is to improve efficiency in implementations. This improvement results from the fact that we can use smaller prime-order groups than composite-order groups at equivalent security levels. We now examine this improvement concretely. Specifically, we compare the sizes of the groups G, H, and G_t produced by the bilinear group generator $\mathcal{G}_{BGN}$ (Example 2.3) with the four examples from Section 3 of bilinear group generators built from prime-order generators.

For the generators $\mathcal{G}_P$ (Example 3.3), $\mathcal{G}_{3C}$ (Example 3.7), and $\mathcal{G}_{2C}$ (Example 3.8) we take the prime-order bilinear group generator $\mathcal{P}$ to be an algorithm that produces a "pairing-friendly" ordinary elliptic curve E over a finite field $\mathbb{F}_q$. On such curves there are two "distinguished" subgroups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order p in which the DDH problem is presumed to be infeasible, and such that the Tate pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_t \subset \mathbb{F}_{q^k}^*$ is nondegenerate. Here k is the *embedding degree*, defined to be the smallest integer such that p divides the order of $\mathbb{F}_{q^k}^*$.

The ordinary elliptic curves E that give the best performance while providing discrete log security comparable to three commonly proposed levels of AES security are as follows. The group sizes follow the 2007 NIST recommendations [1]. Further details can be found in the full version of this paper [10, Appendix A]; descriptions of the elliptic curves are in [9].

80-bit security: A 170-bit MNT curve with embedding degree $k = 6$.
128-bit security: A 256-bit Barreto-Naehrig curve with $k = 12$.
256-bit security: A 640-bit Brezing-Weng curve with $k = 24$.

The advantage of the generator $\mathcal{G}_L$ is that we can use a prime-order group with a symmetric pairing, which only exists on supersingular elliptic curves. Thus in this case we take $\mathcal{P}$ to produce a supersingular curve over $\mathbb{F}_{3^m}$ with embedding degree $k = 6$. The fields that provide the best "match" for group orders at our three security levels are $\mathbb{F}_{3^{111}}$, $\mathbb{F}_{3^{323}}$, and $\mathbb{F}_{3^{1615}}$. Since 6 is the maximum possible embedding degree for supersingular curves, at high security levels the group $\mathbb{G}_1$ will be much larger than the group $\mathbb{G}_1$ on an equivalent ordinary curve.

Table 1 compares the sizes of the groups produced by all of our bilinear group generators at each of the three security levels. In all cases the groups G and H built using products of prime-order groups are much smaller than the groups G and H built using composite-order groups. The group G_t for the projecting generator $\mathcal{G}_P$ is twice as large as the composite-order G_t, due to the fact that elements of G_t are four elements of $\mathbb{F}_{q^k}$. However, the groups G_t for the cancelling generators $\mathcal{G}_{2C}$, $\mathcal{G}_{3C}$, $\mathcal{G}_L$ are half as large as the composite-order G_t.

The last column indicates the number of elliptic curve pairings required to compute the pairing e for the specified generator. While the prime-order generators require more pairings than the composite-order generator $\mathcal{G}_{BGN}$, the sizes of the elliptic curve groups in this case are so much smaller that the pairings will

Table 1. Estimated bit sizes of group elements for bilinear group generators at three different security levels

	80-bit AES			128-bit AES			256-bit AES			#Pai-
Bilinear group generator	G	H	G_t	G	H	G_t	G	H	G_t	rings
$\mathcal{G}_{BGN}$ (Example 2.3)	1024	1024	2048	3072	3072	6144	15360	15360	30720	1
$\mathcal{G}_P$ (Example 3.3)	340	680	4080	512	1024	12288	1280	5120	61440	4
$\mathcal{G}_{3C}$ (Example 3.7)	510	1020	1020	768	1536	3072	1920	7680	15360	3
$\mathcal{G}_{2C}$ (Example 3.8)	340	680	1020	512	1024	3072	1280	5120	15360	2
$\mathcal{G}_L(3, 2)$ (Example 3.9)	528	528	1056	1536	1536	3072	7680	7680	15360	3
$\mathcal{G}_L(4, 2)$ (Example 3.9)	704	704	1056	2048	2048	3072	10240	10240	15360	4

be far more than four times faster. For example, the Tate pairing on a 1024-bit supersingular curve runs ≈ 50 times slower than the Tate pairing on a 170-bit MNT curve [18], so a pairing for $\mathbb{G}_P$ at the 80-bit security level will be roughly 12 times faster than a pairing for $\mathbb{G}_{BGN}$.

5 Application: The BGN Cryptosystem

Our first application of the framework developed above is to the public-key encryption scheme of Boneh, Goh, and Nissim [3]. This scheme has the feature that given two ciphertexts, anyone can create a new ciphertext that encrypts either the sum or the product of the corresponding plaintexts. The product operation can only be carried out once; the system is thus "partially doubly homomorphic."

Step 1 of the conversion process is to write the scheme in the abstract framework and transfer it to asymmetric groups. In the original BGN protocol any ciphertext may be paired with any other ciphertext, so in the asymmetric setting each computation in G must be duplicated in H. We must use a projecting pairing, as the decryption algorithm requires projection away from a certain subgroup.

$\mathsf{KeyGen}(\lambda)$**:** Let $\mathcal{G}$ be a projecting bilinear group generator (Definition 3.1). Compute $(G, G_1, H, H_1, G_t, G_t', e, \pi_1, \pi_2, \pi_t) \leftarrow \mathcal{G}(\lambda)$. Choose $g \xleftarrow{\text{R}} G$, $h \xleftarrow{\text{R}} H$, and output the public key $PK = (G, G_1, H, H_1, G_t, e,\ g, h)$ and the secret key $SK = (\pi_1, \pi_2, \pi_t)$.

$\mathsf{Encrypt}(PK, m)$**:** Choose $g_1 \xleftarrow{\text{R}} G_1$ and $h_1 \xleftarrow{\text{R}} H_1$. (Recall that the output of $\mathcal{G}$ allows random sampling from G_1 and H_1.) Output the ciphertext $(C_A, C_B) = (g^m \cdot g_1,\ h^m \cdot h_1) \in G \times H$.

$\mathsf{Multiply}(PK, C_A, C_B)$**:** This algorithm takes as input two ciphertexts $C_A \in G$ and $C_B \in H$. Choose $g_1 \xleftarrow{\text{R}} G_1$ and $h_1 \xleftarrow{\text{R}} H_1$, and output $C = e(C_A, C_B) \cdot e(g, h_1) \cdot e(g_1, h) \in G_t$.

$\mathsf{Add}(PK, C, C')$**:** This algorithm takes as input two ciphertexts C, C' in one of G, H, or G_t. Choose $g_1 \xleftarrow{\text{R}} G_1$, $h_1 \xleftarrow{\text{R}} H_1$, and do the following:
1. If $C, C' \in G$, output $C \cdot C' \cdot g_1 \in G$.
2. If $C, C' \in H$, output $C \cdot C' \cdot h_1 \in H$.
3. If $C, C' \in G_t$, output $C \cdot C' \cdot e(g, h_1) \cdot e(g_1, h) \in G_t$.

$\mathsf{Decrypt}(SK, C)$**:** The input ciphertext C is an element of G, H, or G_t.
1. If $C \in G$, output $m \leftarrow \log_{\pi_1(g)}(\pi_1(C))$.
2. If $C \in H$, output $m \leftarrow \log_{\pi_2(h)}(\pi_2(C))$.
3. If $C \in G_t$, output $m \leftarrow \log_{\pi_t(e(g,h))}(\pi_t(C))$.

It is clear that if C, C' are encryptions of m, m' respectively, then the Add algorithm gives a correctly distributed encryption of $m + m'$. Furthermore, it follows from the bilinear property of the pairing that if $C_A \in G$, $C_B \in H$ are the left and right halves of encryptions of m, m' respectively, then the $\mathsf{Multiply}$ algorithm gives a correctly distributed encryption of $m \cdot m'$. Since there is no pairing on G_t we can only perform the multiplication once.

Correctness of decryption of ciphertexts in G and H follows from the fact that G_1, H_1 are in the kernels of π_1, π_2, respectively. Correctness of decryption of ciphertexts in G_t follows from the "projecting" properties of $\mathcal{G}$; for example, we have $\pi_t(e(g, h_1)) = e(\pi_1(g), \pi_2(h_1)) = e(\pi_1(g), 1) = 1$.

Step 2 of the conversion process is to translate the security assumptions to asymmetric bilinear groups. In this case, semantic security of ciphertexts in G follows from the subgroup decision assumption on the left for $\mathcal{G}$. Intuitively, if $\mathcal{G}$ satisfies the subgroup decision assumption on the left, then an adversary cannot distinguish the real system from a "fake" system in which $g \in G_1$. Semantic security then follows from the fact that in the fake system the ciphertext element C_A will be a uniformly random element of G_1 and thus will contain no information about the message m. The same argument holds for ciphertexts in H, and semantic security of ciphertexts in G_t follows from semantic security in G and H. For further details see [3, Theorem 3.1].

Step 3 is to translate the assumption to prime-order groups. Since the security proof uses no intrinsic properties of the groups G and H, it carries over to our more general setting.

Theorem 5.1. *Let $\mathcal{P}$ be a prime-order bilinear group generator, and let $\mathcal{G}_P$ be the projecting bilinear group generator constructed from $\mathcal{P}$ in Example 3.3. If $\mathcal{P}$ satisfies the DDH assumption in $\mathbb{G}_1$ and $\mathbb{G}_2$, then the BGN cryptosystem instantiated with $\mathcal{G} = \mathcal{G}_P$ is semantically secure.*

When instantiated with either $\mathcal{G}_{BGN}$ or $\mathcal{G}_P$, decryption in the BGN system requires taking discrete logarithms in a group of large prime order. Thus to achieve efficient decryption the message space must be small (i.e., logarithmic in the group size). It is an open problem to find a projecting bilinear group generator $\mathcal{G}$ for which the subgroup decision assumption may hold and for which discrete logarithms can be computed in a subset of $\pi_1(G)$ whose size is a constant fraction of the full group order.

If we carry out the tensor product construction described in Section 3 for any k and $n \geq k+1$, we obtain an instantiation of the BGN cryptosystem whose security depends on the k-Linear assumption. Since ciphertexts will consist of n elements of $\mathbb{G}_1$ or $\mathbb{G}_2$ or n^2 elements of $\mathbb{G}_t$, these systems will be less efficient than the system constructed using $\mathcal{G}_P$, which has $(n, k) = (2, 1)$. We do note, however, that if $k \geq 2$ we can use a group with a symmetric pairing, in which case the $\mathsf{Encrypt}$ algorithm needs only to output the ciphertext C_A.

6 More Applications

We conclude by summarizing several further applications of our framework to cryptosystems constructed using composite-order groups. Details can be found in the full version of this paper [10].

Traitor Tracing. Boneh, Sahai, and Waters [4] construct a traitor tracing system that is fully collusion resistant and has short ciphertexts. After reducing the construction of their system to construction of a primitive called *private linear broadcast encryption* (or PLBE), Boneh et al. devise a PLBE scheme using bilinear groups of composite order and show it secure under three assumptions in bilinear groups: the subgroup decision assumption, the *3-party Diffie-Hellman assumption*, and the *bilinear subgroup decision assumption.*

To apply our general framework to the Boneh et al. PLBE scheme, we first write the original system using asymmetric pairings on abstract groups, and then convert the three assumptions to this more general context. We instantiate the system using the 2-cancelling bilinear group generator $\mathcal{G}_{2C}$ of Example 3.8 and obtain the following security theorem.

Theorem 6.1. *Let $\mathcal{P}$ be a prime-order bilinear group generator, and let $\mathcal{G}_{2C}$ be the 2-cancelling bilinear group generator constructed from $\mathcal{P}$ in Example 3.8. If $\mathcal{P}$ satisfies the DDH assumption in $\mathbb{G}_2$ and the 3-party DDH assumptions in $\mathbb{G}_1$ and $\mathbb{G}_2$ (i.e., given g, g^a, g^b, g^c, no efficient adversary can distinguish g^{abc} from a random group element), then the Boneh-Sahai-Waters PLBE system is secure when instantiated with $\mathcal{G} = \mathcal{G}_{2C}$.*

Predicate Encryption. Katz, Sahai, and Waters [16] construct a predicate encryption scheme using bilinear groups whose order is a product N of three distinct primes. The security of the system is based on two complex (yet constant-size) assumptions in composite-order bilinear groups, which we call Assumptions 1 and 2; both can be seen as variants of the subgroup decision problem.

To apply our general framework to this scheme, we write the scheme using a bilinear group generator with an asymmetric pairing and translate the security assumptions into this more general context. We then instantiate the system in two different ways, using the 3-cancelling bilinear group generators $\mathcal{G}_{3C}$ of Example 3.7 and $\mathcal{G}_L(4, 2)$ of Example 3.9. When using $\mathcal{G}_{3C}$, translating the asymmetric versions of Assumptions 1 and 2 explicitly to this setting produces two new (constant-size) assumptions in prime-order groups; we call these Assumptions 3 and 4. (We also show that these assumptions hold in the generic group model.) When using $\mathcal{G}_L(4, 2)$ we can use simpler assumptions at the expense of a less efficient system (cf. Table 1). We obtain the following security theorem for $\mathcal{G}_{3C}$; details for both cases appear in the full paper.

Theorem 6.2. *Let $\mathcal{P}$ be a prime-order bilinear group generator, and let $\mathcal{G}_{3C}$ be the* 3*-cancelling bilinear group generator constructed from $\mathcal{P}$ in Example 3.7. If $\mathcal{P}$ satisfies Assumptions 3 and 4, then the Katz-Sahai-Waters predicate encryption scheme is secure when instantiated with $\mathcal{G} = \mathcal{G}_{2C}$.*

Further Work. We expect that our framework can be used to create prime-order group instantiations of other cryptosystems that use composite-order bilinear groups. However, since our construction is not black box, the security proof of each cryptosystem must be checked to ensure that it is still valid in our more general framework. For example, the proof of the Lewko-Waters IBE system [17] uses in an essential way the fact that G has two subgroups with relatively prime order; thus our prime-order construction does not apply in this case. Lewko and Waters do give a version of their system in prime-order groups, with a different security proof under new assumptions. It remains an open problem to find a framework that incorporates both versions of the system.

Acknowledgments. The author thanks Dennis Hofheinz, Eike Kiltz, and Brent Waters for helpful discussions.

References

1. Barker, E., Barker, W., Burr, W., Polk, W., Smid, M.: Recommendation for key management — Part 1: General (revised). NIST Special Pub. 800-57 (2007)
2. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
3. 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)
4. 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)
5. 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)
6. 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)
7. 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 (2003)
8. Duquesne, S., Lange, T.: Pairing-based cryptography. In: Handbook of Elliptic and Hyperelliptic Curve Cryptography, pp. 573–590. Chapman & Hall/CRC, Boca Raton (2006)
9. Freeman, D., Scott, M., Teske, E.: A taxonomy of pairing-friendly elliptic curves. Journal of Cryptology 23, 224–280 (2010)
10. Freeman, D.M.: Converting pairing-based protocols from composite-order groups to prime-order groups. Cryptology ePrint Archive, Report 2009/540 (2009), http://eprint.iacr.org/2009/540
11. Galbraith, S., Verheul, E.: An analysis of the vector decomposition problem. In: Cramer, R. (ed.) PKC 2008. LNCS, vol. 4939, pp. 308–327. Springer, Heidelberg (2008)
12. Gjøsteen, K.: Subgroup membership problems and public key cryptosystems. Ph.D. dissertation, Norwegian University of Science and Technology (2004), http://ntnu.diva-portal.org/smash/get/diva2:121977/FULLTEXT01
13. Groth, J., Ostrovsky, R., Sahai, A.: Perfect non-interactive zero knowledge for NP. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 339–358. Springer, Heidelberg (2006)
14. 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)

15. 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)
16. 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), http://eprint.iacr.org/2007/404
17. 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)
18. Scott, M.: Personal communication (February 17, 2009)
19. Shacham, H.: A Cramer-Shoup encryption scheme from the Linear assumption and from progressively weaker Linear variants. Cryptology ePrint Archive, Report 2007/074 (2007), http://eprint.iacr.org/2007/074
20. Shacham, H., Waters, B.: Efficient ring signatures without random oracles. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 166–180. Springer, Heidelberg (2007)
21. 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)
22. Yoshida, M.: Inseparable multiplex transmission using the pairing on elliptic curves and its application to watermarking. In: Proc. 5th Conf. on Algebraic Geometry, Number Theory, Coding Theory and Cryptography, Univ. of Tokyo (2003), http://www.math.uiuc.edu/~duursma/pub/yoshida_paper.pdf

Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption

Allison Lewko[1,*], Tatsuaki Okamoto[2], Amit Sahai[3,**], Katsuyuki Takashima[4], and Brent Waters[5,***]

[1] University of Texas at Austin
alewko@cs.utexas.edu
[2] NTT
okamoto.tatsuaki@lab.ntt.co.jp
[3] UCLA
sahai@cs.ucla.edu
[4] Mitsubishi Electric
Takashima.Katsuyuki@aj.MitsubishiElectric.co.jp
[5] University of Texas at Austin
bwaters@cs.utexas.edu

Abstract. We present two fully secure functional encryption schemes: a fully secure attribute-based encryption (ABE) scheme and a fully secure (attribute-hiding) predicate encryption (PE) scheme for inner-product predicates. In both cases, previous constructions were only proven to be selectively secure. Both results use novel strategies to adapt the dual system encryption methodology introduced by Waters. We construct our ABE scheme in composite order bilinear groups, and prove its security from three static assumptions. Our ABE scheme supports arbitrary monotone access formulas. Our predicate encryption scheme is constructed via a new approach on bilinear pairings using the notion of dual pairing vector spaces proposed by Okamoto and Takashima.

1 Introduction

In a traditional public key encryption system, data is encrypted to be read by a particular individual who has already established a public key. Functional encryption is a new way of viewing encryption which opens up a much larger

[*] Supported by a National Defense Science and Engineering Graduate Fellowship.

[**] Research supported in part from NSF grants 0830803, 0627781, 0716389, 0456717, and 0205594, an equipment grant from Intel, and an Okawa Foundation Research Grant.

[***] Supported by NSF CNS-0716199, CNS-0915361, and CNS-0952692, Air Force Office of Scientific Research (AFO SR) under the MURI award for “Collaborative policies and assured information sharing” (Project PRESIDIO), Department of Homeland Security Grant 2006-CS-001-000001-02 (subaward 641), and the Alfred P. Sloan Foundation.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 62–91, 2010.

world of possibilities for sharing encrypted data. In a functional encryption system, there is a functionality $f(x, y)$ which determines what a user with secret key y can learn from a ciphertext encrypted under x (we can think of both x and y as binary strings, for example). This allows an encryptor to specify a policy describing what users can learn from the ciphertext, without needing to know the identities of these users or requiring them to have already set up public keys. The enhanced functionality and flexibility provided by such systems is very appealing for many practical applications.

Several previous works have pursued directions falling into this general framework, e.g. [34,25,17,5,32,24,39,27,12]. However, the same expressive power of these systems that makes them appealing also makes proving their security especially challenging. For this reason, all of the prior systems were only proven *selectively* secure, meaning that security was proven in a weaker model where part of the challenge ciphertext description must be revealed *before* the attacker receives the public parameters.

In this paper, we present fully secure systems for two cases of functional encryption, namely attribute-based encryption (ABE) and predicate encryption (PE) for inner products. Sahai and Waters [34] proposed Attribute-Based Encryption as a new concept of encryption algorithms that allow the encryptor to set a policy describing who should be able to read the data. In an attribute-based encryption system, private keys distributed by an authority are associated with sets of attributes and ciphertexts are associated with formulas over attributes. A user should be able to decrypt a ciphertext if and only if their private key attributes satisfy the formula. Predicate encryption for inner products was first presented by Katz, Sahai, and Waters [27]. In a predicate encryption scheme, secret keys are associated with predicates, and ciphertexts are associated with attributes. A user should be able to decrypt a ciphertext if and only if their private key predicate evaluates to 1 when applied to the ciphertext attribute.

Our Two Results. The ABE and PE schemes described in this paper have essential commonalities: both are functional encryption schemes that employ the dual system methodology of Waters [40] to prove full security. This is a powerful tool for achieving full security of systems with advanced functionalities, but realizing the dual system methodology in each new context presents unique challenges. In particular, the technical challenges for ABE and PE are distinct, and the two results now combined into this paper were obtained by separate research groups working independently. The ABE result was obtained by Lewko, Sahai, and Waters, while the PE result was obtained by Okamoto and Takashima.

1.1 Attribute-Based Encryption

We are particularly interested in attribute-based encryption as a special case of functional encryption because it provides a functionality that can be very useful in practice. For example, a police force could use an ABE system to encrypt documents under policies like “Internal Affairs OR (Undercover AND Central)” and give out secret keys to undercover officers in the central division corresponding to the attributes “Undercover” and “Central”. Given the many potential uses of ABE systems, constructing efficient systems with strong security guarantees is an important problem.

Previous Constructions and Selective Security. All previous constructions of ABE systems [34,25,18,5,32,24,39] have only been proven to be selectively secure. This is a limited model of security where the attacker is required to announce the target he intends to attack before seeing the public parameters of the system. This is an unnatural and undesirable restriction on the attacker, but it unfortunately appears to be necessary for the proof techniques used in prior works.

To see why this is the case, it is instructive to look into the way that previous security proofs have worked. In these security proofs, the simulator uses the attacker’s announced target to embed the challenge in the public parameters in such a way that the simulator can produce any keys the attacker can request but can also leverage the attacker’s output to break the underlying challenge. This is a *partitioning* strategy reminiscent of the strategies first used to prove security for IBE systems. The formation of the public parameters partitions the keys into two classes: those that the simulator can make, and those that are useful to the simulator in solving its challenge.

While this partitioning strategy was successfully employed by Boneh and Boyen [7], and Waters [38] to prove full security for an IBE system, any partitioning approach seems doomed to failure when one tries to achieve full security for ABE systems. Without selectivity, the simulator cannot anticipate which keys the attacker may ask for, so the attacker must make some type of a guess about what the partition should be. One natural direction is to partition the identity space in some random way and hope that the attacker’s queries respect the partition (which was the main idea behind the works in the IBE setting). For ABE systems, however, private keys and ciphertexts have much more structure; different keys can be related (they may share attributes), and this severely restricts allowable partitions. Thus, the power and expressiveness of ABE systems work directly against us when attempting to create partitioning proofs.

Our Approach. We are able to obtain full security by adapting the dual system encryption technique of [40,28] to the ABE case. Waters [40] introduced dual system encryption to overcome the limitations of partitioning. In a dual encryption system, keys and ciphertexts can take on one of two forms: normal and semi-functional. A normal key can decrypt both normal and semi-functional ciphertexts, while a semi-functional key can only decrypt normal ciphertexts. The semi-functional keys and ciphertexts are not used in the real system, only in the proof of security. The proof employs a hybrid argument over a sequence of security games. The first is the real security game, with normal keys and ciphertext. In the second game, the ciphertext is semi-functional and the keys remain normal. In subsequent games, the keys requested by the attacker are changed to be semi-functional one by one. By the final game, none of the keys given out are actually useful for decrypting a semi-functional ciphertext, and proving security becomes relatively easy.

There is one important subtlety inherent in the dual system technique. In the step where the k^{th} key becomes semi-functional, the simulator must be prepared to make any semi-functional challenge ciphertext and any key as the k^{th} key. At first, this appears to be a paradox, since it seems the simulator can just make a key that should decrypt the challenge ciphertext and decide for itself whether the key is semi-functional by attempting to decrypt the semi-functional challenge ciphertext. Waters addresses this issue by introducing tags: if a key and ciphertext in his IBE system have the same tag, decryption will fail *regardless* of semi-functionality. The simulator is constructed in such a way that if it attempts to check if key k is semi-functional by decrypting a semi-functional ciphertext, it will be thwarted because they will have equal tags. (This relationship between the tags will be hidden to an attacker who cannot request a key able to decrypt the challenge ciphertext.)

Lewko and Waters [28] provide a new realization of dual system encryption where tags are replaced by *nominally* semi-functional keys. Nominally semi-functional keys are structured like semi-functional keys except that they do also successfully decrypt semi-functional ciphertexts (the semi-functional contribution cancels out). When the k^{th} key turns semi-functional in the hybrid, the simulator is constructed so that it can only make a *nominally* semi-functional key k. It is then argued that this looks like a regular semi-functional key to the attacker.

Though they achieve fully secure HIBE with constant size ciphertext, it is not clear how to extend the techniques of [40,28] to obtain fully secure ABE systems. Both rely on the fact that the identities attached to keys and ciphertexts are the same. Waters relies on this to align tags, while Lewko and Waters use this symmetry in designing their system so that a nominally semi-functional key is identically distributed to a regular semi-functional key in the view of an attacker who cannot decrypt. This symmetry does not hold in an ABE system, where keys and ciphertexts are each associated with different objects: attributes and formulas. The additional flexibility and expressiveness of ABE systems leads to a much more complicated structure of relationships between keys and ciphertexts, which makes the potential paradox of the dual system encryption technique more challenging to address for ABE.

We overcome this by giving a new realization of *nominally* semi-functional keys in the ABE setting. We do this by designing the semi-functional components of our keys and ciphertexts to mirror the functionality of the ABE scheme. Intuitively, we want to argue that an attacker who cannot decrypt the message also cannot determine if the final contribution of the semi-functional components will be non-zero. We make this argument information-theoretically by showing that our nominally semi-functional keys are distributed identically to regular semi-functional keys from the attacker's perspective. This information-theoretic argument is more intricate than the HIBE analog executed in [28], due to the more complicated structure of ABE systems.

The ideas above allow us to construct an ABE system that is fully secure. We build our construction in two phases. First, we construct an ABE system with the restriction that each attribute can only be used once in an access formula.

We call this a *one-use* ABE system. Then, we provide a generic transformation from a one-use system to a system which is fully secure when attributes are used multiple times (up to a constant number of uses fixed at setup). While this transformation does incur some cost in key size, it does not increase the size of the ciphertext; we stress that ours is the first feasibility result for fully secure ABE. Our construction supports arbitrary monotone access formulas. We realize our ABE construction using bilinear groups of composite order and prove security under three assumptions used by Lewko and Waters [28].

1.2 Predicate Encryption for Inner Products

ABE systems have desirable functionality, but have one limitation in that the structure of the ciphertext is revealed to users who cannot reveal. For example, in a CP-ABE system, a user who cannot decrypt can still learn the formula associated with the ciphertext. For applications where the access policy must also be kept secret, this is unacceptable. In our second result we address a class of systems, called predicate encryption systems, that overcome this limitation. Our second result gives predicate encryption of inner products between the ciphertext and key vectors.

Predicate encryption (PE) for inner products was presented by Katz, Sahai and Waters [27] as a generalized (fine-grained) notion of encryption that covers identity-based encryption (IBE) [6,7,9,19,21,26], hidden-vector encryption (HVE) [12] and attribute-based encryption (ABE) [5,25,32,33,34]. Informally, secret keys in a PE scheme correspond to *predicates* in some class $\mathcal{F}$, and a sender associates a ciphertext with an *attribute* in set Σ; a ciphertext associated with attribute $I \in \Sigma$ can be decrypted using a secret key sk_f corresponding to predicate $f \in \mathcal{F}$ if and only if $f(I) = 1$.

The special case of inner product predicates is obtained by having each attribute correspond to a vector $\overrightarrow{x}$ and each predicate $f_{\overrightarrow{v}}$ correspond to a vector $\overrightarrow{v}$, where $f_{\overrightarrow{v}}(\overrightarrow{x}) = 1$ iff $\overrightarrow{x} \cdot \overrightarrow{v} = 0$. (Here, $\overrightarrow{x} \cdot \overrightarrow{v}$ denotes the standard inner-product). We note that these represent a wide class of predicates including equality tests (for IBE and HVE), disjunctions or conjunctions of equality tests, and, more generally, arbitrary CNF or DNF formulas (for ABE). However, we note that inner product predicates are less expressive than the LSSS access structures of ABE. To use inner product predicates for ABE, formulas must be written in CNF or DNF form, which can cause a superpolynomial blowup in size for arbitrary formulas.

Katz, Sahai, and Waters also introduced *attribute-hiding*, a security notion for PE that is stronger than the basic security requirement, *payload-hiding.* Roughly speaking, attribute-hiding requires that a ciphertext conceal the associated attribute as well as the plaintext, while payload-hiding only requires that a ciphertext conceal the plaintext. If attributes are identities, i.e., PE is IBE, attribute-hiding PE implies *anonymous* IBE. This notion of attribute-hiding addresses the limitation of ABE systems. Katz, Sahai, and Waters provided a scheme which is attribute-hiding PE for inner-product predicates, but it is only proven to be selectively secure and no delegation functionality is provided.

Our Results

- This paper proposes the first *adaptively secure* PE scheme for *inner-product* predicates in the *standard model.* The scheme is proven to be adaptively attribute-hiding (against CPA) under an assumption that is *non-interactive.* The number of terms of the assumption depends on a system parameter n, which is the vector length. (However, the number of terms does not depend on the number of adversarial private key queries.) We prove that the assumption is true in the generic model of bilinear pairing groups.

  The efficiency of the proposed PE scheme is comparable to that of the existing *selectively-secure* PE schemes [27,31].
- This paper also establishes a (hierarchical) delegation functionality on the proposed adaptively secure PE scheme. That is, we propose an *adaptively secure* (attribute-hiding) hierarchical PE (HPE) scheme for *inner-product* predicates (with polynomially many levels) in the *standard model* under the n-eDDH assumption.

  The proposed HPE scheme implies the first *anonymous* hierarchical IBE (HIBE) with polynomially many levels in the standard model as a special case (when the associated inner-product predicate is specialized as the equality test for HIBE).
- It is straightforward to convert the (CPA-secure) basic (H)PE scheme to a CCA-secure (H)PE scheme by employing an existing general conversion such as that by Canetti, Halevi and Katz [16] or that by Boneh and Katz [11] (using an additional level with two-dimensions for the basic (H)PE scheme, and a strongly unforgeable one-time signature scheme or message authentication code and encapsulation). That is, we can present a *fully secure* (adaptively attribute-hiding against CCA) (H)PE scheme for *inner-product* predicates in the *standard model* under the n-eDDH assumption as well as a strongly unforgeable one-time signature scheme or message authentication code and encapsulation.
- To achieve the result, this paper elaborately combines a new methodology, the dual system encryption, proposed by Waters [40] and a new approach based on a notion of higher dimensional vector spaces, *dual pairing vector spaces* (DPVS), proposed by Okamoto and Takashima [30,31]. The notion of DPVS is constructed on bilinear pairing groups, and they presented a selectively secure (H)PE scheme on DPVS [31]. We will explain this approach and our key technique in Section 3.1.

  Note that the n-eDDH assumption in this paper is defined over the basic primitive, bilinear pairing groups (not over the higher level concept, DPVS), although the proposed PE and HPE schemes are constructed over DPVS, and the assumptions in [31] are defined over DPVS.
- Since HPE is a generalized (fine-grained) version of anonymous HIBE (AHIBE) (or includes AHIBE as a special case), HPE covers (a generalized version of) applications described in [13], fully private communication and search on encrypted data. For example, we can use a two-level HPE scheme where the first level corresponds to the predicate/attribute of (single-layer)

PE and the second level corresponds to those of "attribute search by a predicate" (generalized "key-word search").

1.3 Related Work

Identity Based Encryption (IBE) was proposed by Shamir [35]. In an identity based encryption system, an authority distributes keys to users with associated identities, and messages are encrypted directly to identities. The first IBE schemes were constructed by Boneh and Franklin [9] and Cocks [19]. These schemes were proven secure in the random oracle model. Then selectively secure schemes in the standard model were constructed [15,6]. Boneh and Boyen [7] and Waters [38] constructed fully secure IBE schemes in the standard model. Gentry [21] gave an IBE system and security proof that moved beyond the confines of the partitioning strategy, but at the cost of a large and complicated complexity assumption.

Hierarchical Identity Based Encryption (HIBE) [23,26] expands the functionality of identity based encryption to include a hierarchical structure on identities, where identities can delegate secret keys to their subordinate identities. Boneh and Boyen [6] constructed a selectively secure HIBE scheme. Boneh, Boyen, and Goh [8] constructed a selectively secure HIBE scheme with constant size ciphertexts. Gentry and Halevi [22] extended Gentry's techniques to get a fully secure HIBE system, but under "q-type" assumptions. Waters [40] leveraged the dual system encryption methodology to obtain fully secure IBE and HIBE systems from simple assumptions. Lewko and Waters [28] extended the dual encryption technique to obtain a fully secure HIBE system with constant size ciphertexts.

Attribute-based encryption was introduced by Sahai and Waters [34]. Goyal, Pandey, Sahai, and Waters [25] formulated two complimentary forms of ABE: Ciphertext-Policy Attribute-Based Encryption (CP-ABE) and Key-Policy Attribute-Based Encryption (KP-ABE). In a CP-ABE system, keys are associated with sets of attributes and ciphertexts are associated with access policies. In a KP-ABE system, the situation is reversed: keys are associated with access policies and ciphertexts are associated with sets of attributes. Selectively secure CP-ABE and KP-ABE systems were constructed in [34,25,18,5,32,24,39].

Goyal, Jain, Pandey, and Sahai [24] provide a general way to transform a KP-ABE system into a CP-ABE system. Chase [17] considered the problem of ABE with multiple authorities.

Other works have discussed similar problems without addressing collusion resistance [1,2,3,14,29,37]. In these systems, the data encryptor specifies an access policy such that a set of users can decrypt the data only if the *union* of their credentials satisfies the access policy.

Predicate encryption was introduced by Katz, Sahai, and Waters [27], who also provided a scheme which is attribute-hiding PE for inner-product predicates; only the selective security (not adaptive security) is proven and no delegation functionality is provided.

Shi and Waters [36] presented a delegation mechanism for a class of PE, but the admissible predicates of the system, which is a class of equality tests for HVE, are more limited than inner-product predicates in [27]. Moreover, they proved only selective security.

Okamoto and Takashima [31] proposed a (hierarchical) delegation mechanism for a PE scheme, i.e., a hierarchical PE (HPE) scheme, for inner-product predicates, but only selective security is proven.

Dual pairing vector spaces were introduced by Okamoto and Takashima [30,31], who presented a selectively secure (H)PE scheme based on DPVS.

1.4 Organization

In Section 2, we present our result for ABE. In more detail, Subsection 2.1 provides the necessary background on linear secret-sharing schemes (LSSS), CP-ABE, and composite order bilinear groups, and states our complexity assumptions. Subsection 2.2, we describe our transformation from a one-use CP-ABE system to a system that is secure when attributes are used multiple times in a formula. In Subsection 2.3, we present our CP-ABE system and prove its security. In Subsection 2.4, we discuss extensions of our ABE result.

In Section 3, we present our result for PE for inner products. Subsection 3.1 describes the main ideas of the approach and establishes the necessary notations. In Subsection 3.2, we formally define DPVS. In Subsection 3.3, we state the complexity assumption. In Subsection 3.4, we formally define predicate encryption and inner product predicate encryption. In Subsection 3.5, we present our inner product predicate encryption scheme and its security. In Subsection 3.6, we present our HPE scheme.

2 Fully Secure Attribute-Based Encryption

2.1 Background

Linear Secret-Sharing Schemes. The formal definitions of access structures and linear secret-sharing schemes (LSSS) can be found in [4] and the full version of this paper. Informally, a LSSS is a share-generating matrix A whose rows are labeled by attributes. When we consider the column vector $v = (s, r_2, \ldots, r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared and $r_2, \ldots, r_n \in \mathbb{Z}_p$ are randomly chosen, then Av is the vector of ℓ shares of the secret s. A user's set of attributes S satisfies the LSSS access matrix if the rows labeled by the attributes in S have the *linear reconstruction* property, which means there exist constants $\{\omega_i\}$ such that, for any valid shares $\{\lambda_i\}$ of a secret s according to the LSSS matrix, we have: $\sum_i \omega_i \lambda_i = s$. Essentially, a user will be able to decrypt a ciphertext with access matrix A if and only if the rows of A labeled by the user's attributes include the vector $(1, 0, \ldots, 0)$ in their span.

Now, we formally define CP-ABE and give the full security definition. We also give the necessary background on composite order bilinear groups and state our complexity assumptions.

CP-ABE. A ciphertext-policy attribute-based encryption system consists of four algorithms: Setup, Encrypt, KeyGen, and Decrypt.

Setup$(\lambda, U) \rightarrow (PK, MSK)$. The setup algorithm takes in the security parameter λ and the attribute universe description U. It outputs the public parameters PK and a master secret key MSK.

Encrypt$(PK, M, \mathbb{A}) \rightarrow CT$. The encryption algorithm takes in the public parameters PK, the message M, and an access structure $\mathbb{A}$ over the universe of attributes. It will output a ciphertext CT such that only users whose private keys satisfy the access structure $\mathbb{A}$ should be able to extract M. We assume that $\mathbb{A}$ is implicitly included in CT.

KeyGen$(MSK, PK, S) \rightarrow SK$. The key generation algorithm takes in the master secret key MSK, the public parameters PK, and a set of attributes S. It outputs a private key SK.

Decrypt$(PK, CT, SK) \rightarrow M$. The decryption algorithm takes in the public parameters PK, a ciphertext CT, and a private key SK. If the set of attributes of the private key satisfies the access structure of the ciphertext, it outputs the message M.

Security Model for CP-ABE. We now give the full security definition for CP-ABE systems. This is described by a security game between a challenger and an attacker. The game proceeds as follows:

Setup. The challenger runs the Setup algorithm and gives the public parameters PK to the attacker.

Phase 1. The attacker queries the challenger for private keys corresponding to sets of attributes $S_1, \ldots, S_{q_1}$.

Challenge. The attacker declares two equal length messages M_0 and M_1 and an access structure $\mathbb{A}^*$. This access structure cannot be satisfied by any of the queried attribute sets $S_1, \ldots, S_{q_1}$. The challenger flips a random coin $\beta \in \{0, 1\}$, and encrypts M_b under $\mathbb{A}^*$, producing CT^*. It gives CT^* to the attacker.

Phase 2. The attacker queries the challenger for private keys corresponding to sets of attributes $S_{q_1+1}, \ldots, S_q$, with the added restriction that none of these satisfy $\mathbb{A}^*$.

Guess. The attacker outputs a guess β' for β.

The advantage of an attacker is this game is defined to be $Pr[\beta = \beta'] - \frac{1}{2}$. We note that the model can easily be extended to handle chosen-ciphertext attacks by allowing for decryption queries in Phase 1 and Phase 2.

Definition 1. *A ciphertext-policy attribute-based encryption system is fully secure if all polynomial time attackers have at most a negligible advantage in this security game.*

Selective security is defined by adding an initialization phase where the attacker must declare $\mathbb{A}^*$ before seeing PK. Unlike previous works [5,25,39], we do not impose this restriction on the attacker.

Composite Order Bilinear Groups. We will construct our systems in composite order bilinear groups. Composite order bilinear groups were first introduced in [10]. We define a group generator $\mathcal{G}$, an algorithm which takes a security parameter λ as input and outputs a description of a bilinear group G. For our purposes, we will have $\mathcal{G}$ output $(p_1, p_2, p_3, G, G_T, e)$ where p_1, p_2, p_3 are distinct primes, G and G_T are cyclic groups of order $N = p_1 p_2 p_3$, and $e : G^2 \to G_T$ is a non-degenerate bilinear map.

We now state the complexity assumptions that we will rely on to prove security of our systems. These same assumptions were used by Lewko and Waters to obtain full security of their IBE and HIBE constructions in composite order groups [28]. We note that all three assumptions are static (constant size) and the first assumption is just the subgroup decision problem in the case where the group order is a product of three primes. The assumptions were proven to be generically secure in [28].

In the assumptions below, we let $G_{p_1 p_2}$, e.g., denote the subgroup of order $p_1 p_2$ in G.

Assumption 1 (Subgroup decision problem for 3 primes). Given a group generator $\mathcal{G}$, we define the following distribution:

$$\mathbb{G} = (N = p_1 p_2 p_3, G, G_T, e) \xleftarrow{R} \mathcal{G},$$

$$g \xleftarrow{R} G_{p_1},\ X_3 \xleftarrow{R} G_{p_3},$$

$$D = (\mathbb{G}, g, X_3),$$

$$T_1 \xleftarrow{R} G_{p_1 p_2},\ T_2 \xleftarrow{R} G_{p_1}.$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 1 to be:

$$Adv1_{\mathcal{G},\mathcal{A}}(\lambda) := \Big|Pr[\mathcal{A}(D, T_1) = 1] - Pr[\mathcal{A}(D, T_2) = 1]\Big|.$$

We note that T_1 can be written (uniquely) as the product of an element of G_{p_1} and an element of G_{p_2}. We refer to these elements as the "G_{p_1} part of T_1" and the "G_{p_2} part of T_1" respectively. We will use this terminology in our proofs.

Definition 2. *We say that $\mathcal{G}$ satisfies Assumption 1 if $Adv1_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any polynomial time algorithm $\mathcal{A}$.*

Assumption 2. Given a group generator $\mathcal{G}$, we define the following distribution:

$$\mathbb{G} = (N = p_1 p_2 p_3, G, G_T, e) \xleftarrow{R} \mathcal{G},$$

$$g, X_1 \xleftarrow{R} G_{p_1},\ X_2, Y_2 \xleftarrow{R} G_{p_2},\ X_3, Y_3 \xleftarrow{R} G_{p_3},$$

$$D = (\mathbb{G}, g, X_1X_2, X_3, Y_2Y_3),$$

$$T_1 \xleftarrow{R} G,\ T_2 \xleftarrow{R} G_{p_1p_3}.$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 2 to be:

$$Adv2_{\mathcal{G},\mathcal{A}}(\lambda) := \big|Pr[\mathcal{A}(D, T_1) = 1] - Pr[\mathcal{A}(D, T_2) = 1]\big|.$$

We use $G_{p_1p_3}$ to denote the subgroup of order p_1p_3 in G. We note that T_1 can be (uniquely) written as the product of an element of G_{p_1}, an element of G_{p_2}, and an element of G_{p_3}. We refer to these as the "G_{p_1} part of T_1", the "G_{p_2} part of T_1", and the "G_{p_3} part of T_1", respectively. T_2 can similarly be written as the product of an element of G_{p_1} and an element of G_{p_3}.

Definition 3. *We say that $\mathcal{G}$ satisfies Assumption 2 if $Adv2_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any polynomial time algorithm $\mathcal{A}$.*

Assumption 3. Given a group generator $\mathcal{G}$, we define the following distribution:

$$\mathbb{G} = (N = p_1p_2p_3, G, G_T, e) \xleftarrow{R} \mathcal{G},\ \alpha, s \xleftarrow{R} \mathbb{Z}_N,$$

$$g \xleftarrow{R} G_{p_1},\ X_2, Y_2, Z_2 \xleftarrow{R} G_{p_2},\ X_3 \xleftarrow{R} G_{p_3},$$

$$D = (\mathbb{G}, g, g^\alpha X_2, X_3, g^s Y_2, Z_2),$$

$$T_1 = e(g, g)^{\alpha s},\ T_2 \xleftarrow{R} G_T.$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 3 to be:

$$Adv3_{\mathcal{G},\mathcal{A}}(\lambda) := \big|Pr[\mathcal{A}(D, T_1) = 1] - Pr[\mathcal{A}(D, T_2) = 1]\big|.$$

Definition 4. *We say that $\mathcal{G}$ satisfies Assumption 3 if $Adv3_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any polynomial time algorithm $\mathcal{A}$.*

2.2 Transformation from One-Use CP-ABE

Here we show how to obtain a fully secure CP-ABE system where attributes are used multiple times from a fully secure CP-ABE system where attributes are used only once. We do this with a simple encoding technique.

Suppose we have a CP-ABE system with a universe of n attributes with LSSS access structures that is secure when the function ρ is injective for each access structure associated to a ciphertext (i.e. attributes are only used once in the row labeling the of the share-generating matrix). Suppose we would like to have a system with n attributes where attributes can be used $\leq k$ times in the row labeling of a share-generating matrix. We can realize this by essentially taking k copies of each attribute in the system: instead of a single attribute B, we will have new "attributes" $B : 1, \ldots, B : k$. Each time we want to label a row of an access matrix A with B, we label it with $B : i$ for a new value of i. We let ρ denote the original row labeling of A and ρ' denote this new row labeling.

Each time we want to associate a subset S of attributes to a key, we instead use $S' := \{B : 1, \ldots, B : k | B \in S\}$. We can then employ the one use system on the new universe of kn attributes and retain its full security. We note that the set S' satisfies the access structure (A, ρ') if and only if the set S satisfies the access structure (A, ρ).

For our construction, the sizes of the public parameters and the secret keys grow linearly in the number of involved attributes, so these will expand by a factor of k under this transformation. Note that the size of the access matrix does not change, so ciphertexts in our construction will remain the same size.

2.3 Our Fully Secure CP-ABE System

We construct our fully secure CP-ABE system in composite order groups of order $N = p_1p_2p_3$ with LSSS access structures. We note the strong resemblance between our system and the selectively secure CP-ABE system of Waters [39]. The KP-ABE system we give in the full version of this paper also bears a strong resemblance to the selectively secure schemes in [25]. We thus provide additional examples of the phenomenon noted by [40,28]: dual system encryption is a powerful and versatile tool for transforming selectively secure schemes into fully secure ones.

The normal operation of our system essentially occurs in the subgroup G_{p_1}. Keys are additionally randomized in G_{p_3}, and the subgroup G_{p_2} is our semi-functional space, which is not used in the real system. Keys and ciphertexts will be semi-functional when they involve elements in the G_{p_2} subgroup. When normal keys are paired with semi-functional ciphertexts or semi-functional keys are paired with normal ciphertexts, the elements in G_{p_2} will not contribute to the pairings because they are orthogonal to elements in the G_{p_1} and G_{p_3} subgroups. When we pair a semi-functional key with a semi-functional ciphertext, we get an extra term arising from pairing the corresponding elements of G_{p_2} which will cause decryption to fail, unless this extra term happens to be zero. When this cancelation occurs and decryption still works, we say the key is *nominally* semi-functional. In other words, nominally semi-functional keys involve elements in G_{p_2}, but these cancel when paired with the G_{p_2} elements involved in the semi-functional ciphertext.

Our proof of security will rely on the restriction that each attribute can only be used once in the row labeling of an access matrix. This is because we will argue that a nominally semi-functional key is identically distributed to a regular semi-functional key in the attacker's view, since the attacker cannot ask for keys that can decrypt the challenge ciphertext. This information-theoretic argument fails when attributes can be used multiple times. Nonetheless, we can achieve full security for a system which uses attributes multiple times through the transformation given in the last section.

We believe that our fully secure system in composite order groups can be transformed to a fully secure system in prime order groups. This was accomplished for the previous applications of dual system encryption in [40,28].

Construction

Setup$(\lambda, U) \rightarrow PK, MSK$. The setup algorithm chooses a bilinear group G of order $N = p_1 p_2 p_3$ (3 distinct primes). We let G_{p_i} denote the subgroup of order p_i in G. It then chooses random exponents $\alpha, a \in \mathbb{Z}_N$, and a random group element $g \in G_{p_1}$. For each attribute $i \in U$, it chooses a random value $s_i \in \mathbb{Z}_N$. The public parameters PK are $N, g, g^a, e(g,g)^\alpha, T_i = g^{s_i} \forall i$. The master secret key MSK is α and a generator X_3 of G_{p_3}.

KeyGen(MSK, S, PK*)* $\rightarrow SK$. The key generation algorithm chooses a random $t \in \mathbb{Z}_N$, and random elements $R_0, R'_0, R_i \in G_{p_3}$. The secret key is:

$$S,\ K = g^\alpha g^{at} R_0,\ L = g^t R'_0,\ K_i = T_i^t R_i\ \forall i \in S.$$

Encrypt($(A, \rho), PK, M$*)* $\rightarrow CT$. A is an $\ell \times n$ matrix and ρ is map from each row A_x of A to an attribute $\rho(x)$. The encryption algorithm chooses a random vector $v \in \mathbb{Z}_N^n$, denoted $v = (s, v_2, \dots, v_n)$. For each row A_x of A, it chooses a random $r_x \in \mathbb{Z}_N$. The ciphertext is (we also include (A, ρ) in the ciphertext, though we do not write it below):

$$C = Me(g,g)^{\alpha s},\ C' = g^s,$$

$$C_x = g^{aA_x \cdot v} T_{\rho(x)}^{-r_x},\ D_x = g^{r_x}\ \forall x.$$

Decrypt(CT, PK, SK*)* $\rightarrow M$. The decryption algorithm computes constants $\omega_x \in \mathbb{Z}_N$ such that $\sum_{\rho(x) \in S} \omega_x A_x = (1, 0, \dots, 0)$. It then computes:

$$e(C', K) / \prod_{\rho(x) \in S} \left(e(C_x, L) e(D_x, K_{\rho(x)}) \right)^{\omega_x} = e(g,g)^{\alpha s}.$$

Then M can be recovered as $C / e(g,g)^{\alpha s}$.

Security. Before we give our proof of security, we need to define two additional structures: semi-functional ciphertexts and keys. These will not be used in the real system, but will be needed in our proof.

Semi-functional Ciphertext. A semi-functional ciphertext is formed as follows. We let g_2 denote a generator of G_{p_2} and c a random exponent modulo N. We also choose random values $z_i \in \mathbb{Z}_N$ associated to attributes, random values $\gamma_x \in \mathbb{Z}_N$ associated to matrix rows x, and a random vector $u \in \mathbb{Z}_N^n$. Then:

$$C' = g^s g_2^c,\ C_x = g^{aA_x \cdot v} T_{\rho(x)}^{-r_x} g_2^{A_x \cdot u + \gamma_x z_{\rho(x)}},\ D_x = g^{r_x} g_2^{-\gamma_x}\ \forall x.$$

Semi-functional Key. A semi-functional key will take on one of two forms. A semi-functional key of type 1 is formed as follows. Exponents $t, d, b \in \mathbb{Z}_N$ and elements $R_0, R'_0, R_i \in G_{p_3}$ are chosen randomly. The key is set as:

$$K = g^{\alpha} g^{at} R_0 g_2^d,\ L = g^t R'_0 g_2^b,\ K_i = T_i^t R_i g_2^{bz_i}\ \forall i \in S.$$

A semi-functional key of type 2 is formed without the terms g_2^b and $g_2^{bz_i}$ (one could also interpret this as setting $b = 0$):

$$K = g^{\alpha} g^{at} R_0 g_2^d,\ L = g^t R'_0,\ K_i = T_i^t R_i\ \forall i \in S.$$

We note that when we use a semi-functional key to decrypt a semi-functional ciphertext, we are left with an additional term:

$$e(g_2, g_2)^{cd-bu_1},$$

where u_1 denotes the first coordinate of u (i.e. $(1, 0, \ldots, 0) \cdot u$). We also note that these values z_i are common to semi-functional ciphertexts and semi-functional keys of type 1. These z_i terms always cancel when semi-functional keys are paired with semi-functional ciphertexts, so they do not hinder decryption. Instead, they are used as blinding factors to hide the value being shared in the G_{p_2} subgroup of a semi-functional ciphertext (the value u_1) from an attacker who cannot decrypt. This is where our one-use restriction is crucial: an attacker with a single semi-functional key of type 1 which cannot decrypt the challenge ciphertext should only be able to gain very limited information-theoretic knowledge of the z_i values. If attributes are used multiple times, too many z_i values may be exposed to the attacker. In each of the games we define below, at most one key is semi-functional of type 1 and all other semi-functional keys are type 2. This is to avoid information-theoretically leaking the z_i values by using them in multiple keys at once.

We call a semi-functional key of type 1 *nominally* semi-functional if $cd - bu_1 = 0$. Notice that when such a key is used to decrypt a corresponding semi-functional ciphertext, decryption will succeed.

We will prove the security of our system from Assumptions 1, 2, and 3 using a hybrid argument over a sequence of games. The first game, Game_{Real}, is the real security game (the ciphertext and all the keys are normal). In the next game, Game_0, all of the keys will be normal, but the challenge ciphertext will be semi-functional. We let q denote the number of key queries made by the attacker. For k from 1 to q, we define:

$Game_{k,1}$. In this game, the challenge ciphertext is semi-functional, the first $k-1$ keys are semi-functional of type 2, the k^{th} key is semi-functional of type 1, and the remaining keys are normal.

$Game_{k,2}$. In this game, the challenge ciphertext is semi-functional, the first k keys are semi-functional of type 2, and the remaining keys are normal.

We note that in $\text{Game}_{q,2}$, all of the keys are semi-functional of type 2. In the final game, Game_{Final}, all keys are semi-functional of type 2 and the ciphertext is a semi-functional encryption of a random message, independent of the two messages provided by the attacker. In Game_{Final}, the attacker's advantage is 0. We will prove these games are indistinguishable in the following four lemmas. We give the proof of the most interesting lemma below, and the rest of the proofs can be found in the full version of this paper. For notational purposes in the lemmas below, we think of $Game_{0,2}$ as another way of denoting Game 0.

Lemma 1. *Suppose there is an efficient algorithm $\mathcal{A}$ such that $Game_{Real}Adv_{\mathcal{A}} - Game_0Adv_{\mathcal{A}} = \epsilon$. Then we can construct an efficient algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 1.*

Lemma 2. *Suppose there is an efficient algorithm $\mathcal{A}$ such that $Game_{k-1,2}Adv_{\mathcal{A}} - Game_{k,1}Adv_{\mathcal{A}} = \epsilon$. Then we can construct an efficient algorithm $\mathcal{B}$ with advantage negligibly close to ϵ in breaking Assumption 2.*

Proof. $\mathcal{B}$ is given $g, X_1X_2, X_3, Y_2Y_3, T$. It will simulate Game$_{k-1,2}$ or Game$_{k,1}$ with $\mathcal{A}$. It chooses random exponents $a, \alpha \in \mathbb{Z}_N$ and a random exponent $s_i \in \mathbb{Z}_N$ for each attribute i in the system. It then sends $\mathcal{A}$ the public parameters:

$$PK = \{N,\ g,\ g^a,\ e(g,g)^{\alpha},\ T_i = g^{s_i}\ \forall i\}.$$

To make the first $k-1$ keys semi-functional of type 2, $\mathcal{B}$ responds to each key request by choosing a random $t \in \mathbb{Z}_N$, random elements R'_0, R_i of G_{p_3}, and setting:

$$K = g^{\alpha} g^{at} (Y_2Y_3)^t,\ L = g^t R'_0,\ K_i = T_i^t R_i\ \forall i \in S.$$

We note that K is properly distributed because the values of t modulo p_2 and p_3 are uncorrelated to its value modulo p_1. To make normal keys for requests $> k$, $\mathcal{B}$ can simply run the key generation algorithm since it knows the MSK.

To make key k, $\mathcal{B}$ will implicity set g^t equal to the G_{p_1} part of T. $\mathcal{B}$ chooses random elements R_0, R'_0, R_i in G_{p_3} and sets:

$$K = g^{\alpha} T^a R_0,\ L = TR'_0,\ K_i = T^{s_i} R_i\ \forall i \in S.$$

We note that if $T \in G_{p_1p_3}$, this is a properly distributed normal key. If $T \in G$, this is a semi-functional key of type 1. In this case, we have implicitly set $z_i = s_i$. If we let g_2^b denote the G_{p_2} part of T, we have that $d = ba$ modulo p_2 (i.e. the G_{p_2} part of K is $g_2^b a$, the G_{p_2} part of L is g_2^b, and the G_{p_2} part of K_i is $g_2^{bz_i}$. Note that the value of z_i modulo p_2 is uncorrelated from the value of s_i modulo p_1.

$\mathcal{A}$ sends $\mathcal{B}$ two messages M_0, M_1 and an access matrix (A^*, ρ). To make the semi-functional challenge ciphertext, $\mathcal{B}$ implicitly sets $g^s = X_1$ and $g_2^c = X_2$. It chooses random values $u_2, \ldots, u_n \in \mathbb{Z}_N$ and defines the vector u' as $u' = (a, u_2, \ldots, u_n)$. It also chooses a random exponent $r'_x \in \mathbb{Z}_N$. The ciphertext is formed as:

$$C = M_{\beta} e(g^{\alpha}, X_1X_2),\ C' = X_1X_2,$$

$$C_x = (X_1X_2)^{A_x^* \cdot u'} (X_1X_2)^{-r'_x s_{\rho(x)}},\ D_x = (X_1X_2)^{r'_x}.$$

We note that this sets $v = sa^{-1}u'$ and $u = cu'$, so s is being shared in the G_{p_1} subgroup and ca is being shared in the G_{p_2} subgroup. This also implicitly sets $r_x = r'_x s$, $\gamma_x = -cr'_x$. The values $z_{\rho(x)} = s_{\rho(x)}$ match those in the k^{th} key if it is semi-functional of type 1, as required.

The k^{th} key and ciphertext are *almost* properly distributed, except for the fact that the first coordinate of u (which equals ac) is correlated with the value of a modulo p_2 that also appears in key k if it is semi-functional. In fact, if the k^{th} key could decrypt the challenge ciphertext we would have $cd - bu_1 = cba - bca = 0$ modulo p_2, so our key is either normal or nominally semi-functional. We must argue that this is hidden to the attacker $\mathcal{A}$, who cannot request any keys that can decrypt the challenge ciphertext.

To argue that the value being shared in G_{p_2} in the challenge ciphertext is information-theoretically hidden, we appeal to our restriction that attributes are only used once in labeling the rows of the matrix. Since the k^{th} key cannot decrypt the challenge ciphertext, the rowspace R formed by the rows of the matrix whose attributes are in the key does not include the vector $(1, 0, \dots, 0)$. So for shares $\delta_x = A^*_x \cdot u$ in the G_{p_2} subgroup, we can write $u = u_R + u_W$, where u_R is in the space R and u_W is in its orthogonal complement, W. We note that $u_1 = u \cdot (1, 0, \dots, 0)$ cannot be determined from u_R alone - some information about u_W is needed.

The only places u_W appears are in equations of the form:

$$A^*_x \cdot u + \gamma_x z_{\rho(x)},$$

where the $\rho(x)$'s are each *unique* attributes not appearing the k^{th} key. As long as each γ_x is not congruent to 0 modulo p_2, each of these equations introduces a new unknown $z_{\rho(x)}$ that appears nowhere else, and so no information about u_W can be learned by the attacker. More precisely, for each potential value of u_1, there are an equal number of solutions to these equations, so each value is equally likely. Hence, the value being shared in the G_{p_2} subgroup in the semi-functional ciphertext is information-theoretically hidden, as long as each γ_x is non-zero modulo p_2. The probability that any of the γ_x values are congruent to 0 modulo p_2 is negligible. Thus, the ciphertext and key k are properly distributed in the attacker's view with probability negligibly close to 1.

Thus, if $T \in G_{p_1 p_3}$, then $\mathcal{B}$ has properly simulated Game$_{k-1,2}$, and if $T \in G$ and all the γ_x values are non-zero modulo p_2, then $\mathcal{B}$ has properly simulated Game$_{k,1}$. $\mathcal{B}$ can therefore use the output of $\mathcal{A}$ to gain advantage negligibly close to ϵ in breaking Assumption 2.

Lemma 3. *Suppose there is an efficient algorithm $\mathcal{A}$ such that $Game_{k,1}Adv_{\mathcal{A}} - Game_{k,2}Adv_{\mathcal{A}} = \epsilon$. Then we can construct an efficient algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2.*

Lemma 4. *Suppose there is an efficient algorithm $\mathcal{A}$ such that $Game_{q,2}Adv_{\mathcal{A}} - Game_{Final}Adv_{\mathcal{A}} = \epsilon$. Then we can construct an efficient algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 3.*

We have now proven the following theorem:

Theorem 1. *If Assumptions 1, 2, and 3 hold, then our CP-ABE system is secure.*

Proof. If Assumptions 1, 2, and 3 hold, then we have shown by the previous lemmas that the real security game is indistinguishable from Game_{Final}, in which the value of β is information-theoretically hidden from the attacker. Hence the attacker cannot attain a non-negligible advantage in breaking the CP-ABE system.

Expanding to Multi-Use. To build a fully secure CP-ABE system where each attribute can be used up to k times in the row labeling of an access matrix, we apply the encoding technique of Section 2.2. We note that the public parameters and key sizes will grow by a factor of k, but the encoding does not increase the size of the ciphertext.

2.4 Discussion

We have obtained the first fully secure CP-ABE system in the standard model. Our techniques also yield a fully secure KP-ABE system. Our KP-ABE system and the proof of its security can be found in the full version of this paper. Essentially, a KP-ABE system is like a CP-ABE system with the roles of keys and ciphertexts reversed: in a KP-ABE system, keys are associated with access structures and ciphertexts are associated with subsets of attributes. Our techniques readily adapt to KP-ABE, and the proof of security is very similar to the CP-ABE case.

It is also possible to adapt our techniques to obtain a large universe construction. In our current construction, the size of the public parameters is linear in the number of attributes in the universe. In a large universe construction, we could use all elements of $\mathbb{Z}_{p_1}^*$ as attributes, with the size of the public parameters linear in n, a parameter which denotes the maximum size of a set of attributes used in the system. This reduces the size of the public parameters and allows us to use arbitrary strings as attributes by applying a collision-resistant hash function $H : \{0,1\}^* \rightarrow \mathbb{Z}_{p_1}^*$. Note that these attributes no longer need to have been considered during setup. To obtain a large universe construction, we could replace the group elements T_i associated with attributes i with a function $T : \mathbb{Z}_{p_1} \rightarrow G_{p_1}$ based on a degree n polynomial. Goyal, Pandey, Sahai, and Waters [25] do this for their KP-ABE construction.

Though we build our ABE systems in composite order bilinear groups, we believe that similar systems can be constructed in prime order groups. Waters [40] first instantiated his fully secure IBE and HIBE systems in composite order groups and then transferred them into prime order groups, obtaining full security under the well-established $d - BDH$ and decisional Linear assumptions. Lewko and Waters [28] built upon these ideas to obtain an analog of their IBE system in asymmetric prime order groups. The introduction of asymmetry simplified their construction, at the expense of relying on non-standard (static) assumptions. Freeman [20] also discusses a general class of transformations from composite order groups to prime order groups, but this does not encompass our construction. In the future, these transformation techniques might be extended to obtain versions of our ABE schemes in prime order groups.

3 Fully Secure Predicate Encryption

3.1 Our Approach and Key Technique

Dual Pairing Vector Spaces (DPVS). We now briefly explain our approach, DPVS, constructed on symmetric pairing groups $(q, \mathbb{G}, \mathbb{G}_T, g, e)$, where q is a prime, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order q, g is a generator of $\mathbb{G}$, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a non-degenerate bilinear pairing operation, and $g_T := e(g, g) \neq 1$. Here we denote the group operation of $\mathbb{G}$ and $\mathbb{G}_T$ by multiplication. Note that this construction also works on *asymmetric* pairing groups (in this paper, we use symmetric pairing groups for simplicity of description). As for the definitions of some notations, see the last part of this subsection.

Vector space $\mathbb{V}$: $\mathbb{V} := \overbrace{\mathbb{G} \times \cdots \times \mathbb{G}}^{N}$, whose element is expressed by N-dimensional vector, $\boldsymbol{x} := (g^{x_1}, \ldots, g^{x_N})$ ($x_i \in \mathbb{F}_q$ for $i = 1, \ldots, N$).

Canonical base $\mathbb{A}$: $\mathbb{A} := (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N)$ of $\mathbb{V}$, where $\boldsymbol{a}_1 := (g, 1, \ldots, 1)$, $\boldsymbol{a}_2 := (1, g, 1, \ldots, 1), \ldots, \boldsymbol{a}_N := (1, \ldots, 1, g)$.

Pairing operation: $e(\boldsymbol{x}, \boldsymbol{y}) := \prod_{i=1}^{N} e(g^{x_i}, g^{y_i}) = e(g, g)^{\sum_{i=1}^{N} x_i y_i} = g_T^{\overrightarrow{x} \cdot \overrightarrow{y}} \in \mathbb{G}_T$, where $\boldsymbol{x} := (g^{x_1}, \ldots, g^{x_N}) = x_1\boldsymbol{a}_1 + \cdots + x_N\boldsymbol{a}_N \in \mathbb{V}$, $\boldsymbol{y} := (g^{y_1}, \ldots, g^{y_N}) = y_1\boldsymbol{a}_1 + \cdots + y_N\boldsymbol{a}_N \in \mathbb{V}$, $\overrightarrow{x} := (x_1, \ldots, x_N)$ and $\overrightarrow{y} := (y_1, \ldots, y_N)$. Here, $\boldsymbol{x}$ and $\boldsymbol{y}$ can be expressed by coefficient vector over basis $\mathbb{A}$ such that $(x_1, \ldots, x_N)_{\mathbb{A}} = (\overrightarrow{x})_{\mathbb{A}} := \boldsymbol{x}$ and $(y_1, \ldots, y_N)_{\mathbb{A}} = (\overrightarrow{y})_{\mathbb{A}} := \boldsymbol{y}$.

Base change: Canonical basis $\mathbb{A}$ is changed to basis $\mathbb{B} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_N)$ of $\mathbb{V}$ using a uniformly chosen (regular) linear transformation, $X := (\chi_{i,j}) \xleftarrow{\mathsf{U}} GL(N, \mathbb{F}_q)$, such that $\boldsymbol{b}_i = \sum_{j=1}^{N} \chi_{i,j}\boldsymbol{a}_j$, $(i = 1, \ldots, N)$. $\mathbb{A}$ is also changed to basis $\mathbb{B}^* := (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_N^*)$ of $\mathbb{V}$, such that $(\vartheta_{i,j}) := (X^T)^{-1}$, $\boldsymbol{b}_i^* = \sum_{j=1}^{N} \vartheta_{i,j}\boldsymbol{a}_j$, $(i = 1, \ldots, N)$. We see that $e(\boldsymbol{b}_i, \boldsymbol{b}_j^*) = g_T^{\delta_{i,j}}$, ($\delta_{i,j} = 1$ if $i = j$, and $\delta_{i,j} = 0$ if $i \neq j$) i.e., $\mathbb{B}$ and $\mathbb{B}^*$ are dual orthonormal bases of $\mathbb{V}$.

Here, $\boldsymbol{x} := x_1\boldsymbol{b}_1 + \cdots + x_N\boldsymbol{b}_N \in \mathbb{V}$ and $\boldsymbol{y} := y_1\boldsymbol{b}_1^* + \cdots + y_N\boldsymbol{b}_N^* \in \mathbb{V}$ can be expressed by coefficient vectors over $\mathbb{B}$ and $\mathbb{B}^*$ such that $(x_1, \ldots, x_N)_{\mathbb{B}} = (\overrightarrow{x})_{\mathbb{B}} := \boldsymbol{x}$ and $(y_1, \ldots, y_N)_{\mathbb{B}^*} = (\overrightarrow{y})_{\mathbb{B}^*} := \boldsymbol{y}$, and $e(\boldsymbol{x}, \boldsymbol{y}) = e(g, g)^{\sum_{i=1}^{N} x_i y_i} = g_T^{\overrightarrow{x} \cdot \overrightarrow{y}} \in \mathbb{G}_T$.

Intractable problem: One of the most natural decisional problems in this approach is the decisional subspace problem [30]. It is to distinguish $\boldsymbol{v} := v_{N_2+1}\boldsymbol{b}_{N_2+1} + \cdots + v_{N_1}\boldsymbol{b}_{N_1}$ $(= (0, \ldots, 0, v_{N_2+1}, \ldots, v_{N_1})_{\mathbb{B}})$, from $\boldsymbol{u} := v_1\boldsymbol{b}_1 + \cdots + v_{N_1}\boldsymbol{b}_{N_1}$ $(= (v_1, \ldots, v_{N_1})_{\mathbb{B}})$, where $(v_1, \ldots, v_{N_1}) \xleftarrow{\mathsf{U}} \mathbb{F}_q^{N_1}$ and $N_2 + 1 < N_1$.

Trapdoor: Although the decisional subspace problem is assumed to be intractable, it can be efficiently solved by using *trapdoor* $\boldsymbol{t}^* \in \mathsf{span}\langle \boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_{N_2}^* \rangle$. Given $\boldsymbol{v} := v_{N_2+1}\boldsymbol{b}_{N_2+1} + \cdots + v_{N_1}\boldsymbol{b}_{N_1}$ or $\boldsymbol{u} := v_1\boldsymbol{b}_1 + \cdots + v_{N_1}\boldsymbol{b}_{N_1}$, we can distinguish $\boldsymbol{v}$ from $\boldsymbol{u}$ using $\boldsymbol{t}^*$ since $e(\boldsymbol{v}, \boldsymbol{t}^*) = 1$ and $e(\boldsymbol{u}, \boldsymbol{t}^*) \neq 1$ with high probability.

Dual System Encryption Methodology. At the top level of strategy of the security proof, we follow the dual system encryption methodology proposed by Waters [40]. Security is proven using a sequence of games. Game 0 is the real security game. In Game 1, the target ciphertext is changed to semi-functional. When ν secret key queries are issued by an adversary, there are ν game changes from Game 1 (Game 2-0) through Game 2-ν. In Game 2-k, the first k keys are semi-functional while the remaining keys are normal. The final game with advantage 0 is changed from Game 2-ν. As usual, we prove that the advantage gaps between neighboring games are negligible.

The most difficult part in the security proof, *especially for inner-product predicate encryption*, is how to resolve a paradoxical problem to prove the negligible gap between Game 2-k and Game 2-$(k-1)$, where the simulator (for the security proof) itself may distinguish the simulated k-th key (semi-functional key) in Game 2-k and the k-th key (normal key) in Game 2-$(k-1)$ by using a simulated (semi-functional) ciphertext, since the simulator can make ciphertexts and keys for any legal attributes and predicates (especially, in the adaptive security game, the simulator should generate a target ciphertext associated with any attribute adaptively selected by the adversary).

For (H)IBE, this problem was resolved by introducing tricks such that the simulated k-th key and ciphertext have a special correlation regarding the equality of their identity values [28,40].

This problem is much harder for inner-product predicate encryption. Given a predicate vector $\overrightarrow{v}$ for secret key $\mathsf{sk}_{\overrightarrow{v}}$, there are exponentially many (orthogonal) attribute vectors $\overrightarrow{x}$ for ciphertext $c_{\overrightarrow{x}}$ such that $\mathsf{sk}_{\overrightarrow{v}}$ can decrypt $c_{\overrightarrow{x}}$, i.e., $\overrightarrow{v} \cdot \overrightarrow{x} = 0$. Therefore, in order to resolve the above-mentioned paradoxical problem, we should give some trick on the simulated k-th key $\mathsf{sk}_{\overrightarrow{v}}$ with $\overrightarrow{v}$ and all ciphertexts with $\overrightarrow{x}$ satisfying $\overrightarrow{v} \cdot \overrightarrow{x} = 0$, while a trick on the simulated k-th key sk_I with identity I and ciphertext with the same I is enough for (H)IBE.

We use *special form of semi-functional* keys and ciphertexts for simulating the k-th key and target ciphertext such that the simulated k-th key (a special form of semi-functional key) $\mathsf{sk}_{\overrightarrow{v}}$ in Game 2-k can decrypt *all* simulated ciphertexts (a special form of semi-functional ciphertexts) $c_{\overrightarrow{x}}$ with $\overrightarrow{x}$ satisfying $\overrightarrow{v} \cdot \overrightarrow{x} = 0$. Essentially, we adapt the notion of *nominally semi-functional* keys and ciphertexts that was introduced by Lewko and Waters [28] to the setting of inner product encryption.

In addition, the distribution of a pair comprising the simulated k-th key $\mathsf{sk}_{\overrightarrow{v}}$ and simulated ciphertext $c_{\overrightarrow{x}}$ (i.e., a *special semi-functional* key and ciphertext) is equivalent to that of an independent and random *semi-functional* key and ciphertext except with negligible probability, when $\overrightarrow{v} \cdot \overrightarrow{x} \neq 0$.

That is, the special forms of semi-functional keys and ciphertexts are correlated (for the case of $\overrightarrow{v} \cdot \overrightarrow{x} = 0$), but the adversary cannot notice the correlation since the adversary's queries should satisfy the condition $\overrightarrow{v} \cdot \overrightarrow{x} \neq 0$. In other words, nominal semi-functionality is information-theoretically hidden from the adversary. A more detailed explanation of how this is implemented on DPVS will be given in the proof outline in Section 3.5.

Notations. When A is a random variable or distribution, $y \overset{\mathsf{R}}{\leftarrow} A$ denotes that y is randomly selected from A according to its distribution. When A is a set, $y \overset{\mathsf{U}}{\leftarrow} A$ denotes that y is uniformly selected from A. $y := z$ denotes that y is set, defined or substituted by z. When a is a fixed value, $A(x) \to a$ (e.g., $A(x) \to 1$) denotes the event that machine (algorithm) A outputs a on input x. A function $f : \mathbb{N} \to \mathbb{R}$ is *negligible* in λ, if for every constant $c > 0$, there exists an integer n such that $f(\lambda) < \lambda^{-c}$ for all $\lambda > n$.

We denote the finite field of order q by $\mathbb{F}_q$. A vector symbol denotes a vector representation over $\mathbb{F}_q$, e.g., $\overrightarrow{x}$ denotes $(x_1, \ldots, x_n) \in \mathbb{F}_q^n$. For two vectors $\overrightarrow{x} = (x_1, \ldots, x_n)$ and $\overrightarrow{v} = (v_1, \ldots, v_n)$, $\overrightarrow{x} \cdot \overrightarrow{v}$ denotes the inner-product $\sum_{i=1}^{n} x_i v_i$. X^{T} denotes the transpose of matrix X. I_ℓ and 0_ℓ denote the $\ell \times \ell$ identity matrix and the $\ell \times \ell$ zero matrix, respectively. A bold face letter denotes an element of vector space $\mathbb{V}$, e.g., $\boldsymbol{x} \in \mathbb{V}$. When $\boldsymbol{b}_i \in \mathbb{V}$ $(i = 1, \ldots, n)$, $\mathsf{span}\langle \boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \rangle \subseteq \mathbb{V}$ (resp. $\mathsf{span}\langle \overrightarrow{x}_1, \ldots, \overrightarrow{x}_n \rangle$) denotes the subspace generated by $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ (resp. $\overrightarrow{x}_1, \ldots, \overrightarrow{x}_n$). For bases $\mathbb{B} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_N)$ and $\mathbb{B}^* := (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_N^*)$, $(x_1, \ldots, x_N)_{\mathbb{B}} := \sum_{i=1}^{N} x_i \boldsymbol{b}_i$ and $(y_1, \ldots, y_N)_{\mathbb{B}^*} := \sum_{i=1}^{N} y_i \boldsymbol{b}_i^*$.

3.2 Dual Pairing Vector Spaces by Direct Product of Symmetric Pairing Groups

Definition 5. *"Symmetric bilinear pairing groups" $(q, \mathbb{G}, \mathbb{G}_T, g, e)$ are a tuple of a prime q, cyclic (multiplicative) groups $\mathbb{G}$ and $\mathbb{G}_T$ of order q, $g \neq 1 \in \mathbb{G}$, and a polynomial-time computable nondegenerate bilinear pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ i.e., $e(g^s, g^t) = e(g, g)^{st}$ and $e(g, g) \neq 1$.*

Let $\mathcal{G}_{\mathsf{bpg}}$ be an algorithm that takes input 1^λ and outputs a description of bilinear pairing groups $(q, \mathbb{G}, \mathbb{G}_T, g, e)$ with security parameter λ.

In this paper, we concentrate on the symmetric version of dual pairing vector spaces [30,31] constructed by using symmetric bilinear pairing groups given in Definition 5.

Definition 6. *"Dual pairing vector spaces (DPVS)" $(q, \mathbb{V}, \mathbb{G}_T, \mathbb{A}, e)$ by a direct product of symmetric pairing groups $(q, \mathbb{G}, \mathbb{G}_T, g, e)$ are a tuple of prime q, N-dimensional vector space $\mathbb{V} := \overbrace{\mathbb{G} \times \cdots \times \mathbb{G}}^{N}$ over $\mathbb{F}_q$, cyclic group $\mathbb{G}_T$ of order q, canonical basis $\mathbb{A} := (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N)$ of $\mathbb{V}$, where $\boldsymbol{a}_i := (\overbrace{1, \ldots, 1}^{i-1}, g, \overbrace{1, \ldots, 1}^{N-i})$, and pairing $e : \mathbb{V} \times \mathbb{V} \to \mathbb{G}_T$.*

The pairing is defined by $e(\boldsymbol{x}, \boldsymbol{y}) := \prod_{i=1}^{N} e(g_i, h_i) \in \mathbb{G}_T$ where $\boldsymbol{x} := (g_1, \ldots, g_N) \in \mathbb{V}$ and $\boldsymbol{y} := (h_1, \ldots, h_N) \in \mathbb{V}$. This is nondegenerate bilinear i.e.,

$e(s\boldsymbol{x}, t\boldsymbol{y}) = e(\boldsymbol{x}, \boldsymbol{y})^{st}$ and if $e(\boldsymbol{x}, \boldsymbol{y}) = 1$ for all $\boldsymbol{y} \in \mathbb{V}$, then $\boldsymbol{x} = \boldsymbol{0}$. For all i and j, $e(\boldsymbol{a}_i, \boldsymbol{a}_j) = g_T^{\delta_{i,j}}$ where $\delta_{i,j} = 1$ if $i = j$, and 0 otherwise, and $g_T := e(g, g) \neq 1 \in \mathbb{G}_T$.

DPVS also has linear transformations $\phi_{i,j}$ on $\mathbb{V}$ s.t. $\phi_{i,j}(\boldsymbol{a}_j) = \boldsymbol{a}_i$ and $\phi_{i,j}(\boldsymbol{a}_k) = \boldsymbol{0}$ if $k \neq j$, which can be easily achieved by $\phi_{i,j}(\boldsymbol{x}) := (\overbrace{1, \ldots, 1}^{i-1}, g_j, \overbrace{1, \ldots, 1}^{N-i})$ where $\boldsymbol{x} := (g_1, \ldots, g_N)$. We call $\phi_{i,j}$ "distortion maps".

DPVS generation algorithm $\mathcal{G}_{\mathsf{dpvs}}$ takes input 1^λ ($\lambda \in \mathbb{N}$) and $N \in \mathbb{N}$, and outputs a description of $\mathsf{param}_{\mathbb{V}} := (q, \mathbb{V}, \mathbb{G}_T, \mathbb{A}, e)$ with security parameter λ and N-dimensional $\mathbb{V}$. It can be constructed by using $\mathcal{G}_{\mathsf{bpg}}$.

For the asymmetric version of DPVS, $(q, \mathbb{V}, \mathbb{V}^*, \mathbb{G}_T, \mathbb{A}, \mathbb{A}^*, e)$, see the full version of this paper. The above symmetric version is obtained by identifying $\mathbb{V} = \mathbb{V}^*$ and $\mathbb{A} = \mathbb{A}^*$ in the asymmetric version. (For the other realization using higher genus Jacobians, see [30].)

We describe random dual orthonormal bases generator $\mathcal{G}_{\mathsf{ob}}$ below, which is used as a subroutine in the proposed (H)PE scheme.

$$\begin{aligned}
&\mathcal{G}_{\mathsf{ob}}(1^\lambda, N): \ \mathsf{param}_{\mathbb{V}} := (q, \mathbb{V}, \mathbb{G}_T, \mathbb{A}, e) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{dpvs}}(1^\lambda, N), \\
&\quad X := (\chi_{i,j}) \xleftarrow{\mathsf{U}} GL(N, \mathbb{F}_q), \ (\vartheta_{i,j}) := (X^{\mathrm{T}})^{-1}, \\
&\quad \boldsymbol{b}_i := \textstyle\sum_{j=1}^N \chi_{i,j} \boldsymbol{a}_j, \ \mathbb{B} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_N), \ \boldsymbol{b}_i^* := \textstyle\sum_{j=1}^N \vartheta_{i,j} \boldsymbol{a}_j, \ \mathbb{B}^* := (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_N^*), \\
&\quad \text{return} \ \ (\mathsf{param}_{\mathbb{V}}, \mathbb{B}, \mathbb{B}^*).
\end{aligned}$$

3.3 Assumption

Definition 7 (n-eDDH: n-Extended Decisional Diffie-Hellman Assumption). *The n-eDDH problem is to guess $\beta \in \{0, 1\}$, given $(\mathsf{param}_{\mathbb{G}}, \ g, g^\kappa, \{g^{\omega+\gamma_i h_i}, g^{\gamma_i}, g^{h_i}\}_{1 \le i \le n}, \ \{g^{\gamma_i h_j}\}_{1 \le i \neq j \le n}, \ Y_\beta) \xleftarrow{\mathsf{R}} \mathcal{G}_\beta^{n\text{-}\mathsf{eDDH}}(1^\lambda)$, where*

$$\begin{aligned}
&\mathcal{G}_\beta^{n\text{-}\mathsf{eDDH}}(1^\lambda): \ \mathsf{param}_{\mathbb{G}} := (q, \mathbb{G}, \mathbb{G}_T, g, e) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{bpg}}(1^\lambda), \\
&\quad \kappa \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times, \quad \omega, h_i, \gamma_i \xleftarrow{\mathsf{U}} \mathbb{F}_q \ \text{ for } i = 1, \ldots, n, \\
&\quad Y_0 := g^{\kappa\omega}, \quad Y_1 \xleftarrow{\mathsf{U}} \mathbb{G}, \\
&\quad \text{return} \ \ (\mathsf{param}_{\mathbb{G}}, \ g, g^\kappa, \{g^{\omega+\gamma_i h_i}, g^{\gamma_i}, g^{h_i}\}_{1 \le i \le n}, \quad \{g^{\gamma_i h_j}\}_{1 \le i \neq j \le n}, \ Y_\beta),
\end{aligned}$$

for $\beta \xleftarrow{\mathsf{U}} \{0, 1\}$. For a probabilistic machine $\mathcal{C}$, we define the advantage of $\mathcal{C}$ for the n-eDDH problem as:

$$\begin{aligned}
\mathsf{Adv}_{\mathcal{C}}^{n\text{-}\mathsf{eDDH}}(\lambda) := \ &\Big| \mathsf{Pr}\left[\mathcal{C}(1^\lambda, \varrho) \to 1 \,\middle|\, \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_0^{n\text{-}\mathsf{eDDH}}(1^\lambda)\right] \\
&- \mathsf{Pr}\left[\mathcal{C}(1^\lambda, \varrho) \to 1 \,\middle|\, \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_1^{n\text{-}\mathsf{eDDH}}(1^\lambda)\right] \Big|.
\end{aligned}$$

The n-eDDH assumption is: For any polynomial-time adversary $\mathcal{C}$, the advantage $\mathsf{Adv}_{\mathcal{C}}^{n\text{-}\mathsf{eDDH}}(\lambda)$ is negligible.

The following lemma shows that the n-eDDH assumption is true in the generic bilinear pairing group model [8].

Lemma 5. *For any adversary $\mathcal{C}$ that makes a total of at most ν queries to the oracles computing the group operation in $\mathbb{G}$ and the bilinear pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, the advantage $\mathsf{Adv}_{\mathcal{C}}^{n\text{-}\mathsf{eDDH}}(\lambda)$ is $O((\nu+n^2)^2/2^\lambda)$ in the generic bilinear pairing group model.*

The proof of Lemma 5 is given in the full version of this paper.

3.4 Definition of Predicate Encryption

This section defines predicate encryption (PE) for the class of inner-product predicates and its security.

An attribute of inner-product predicates is expressed as a vector $\overrightarrow{x} \in \mathbb{F}_q^n \backslash \{\overrightarrow{0}\}$ and a predicate $f_{\overrightarrow{v}}$ is associated with a vector $\overrightarrow{v}$, where $f_{\overrightarrow{v}}(\overrightarrow{x}) = 1$ iff $\overrightarrow{v} \cdot \overrightarrow{x} = 0$. Let $\Sigma := \mathbb{F}_q^n \setminus \{\overrightarrow{0}\}$, i.e., the set of the attributes, and $\mathcal{F} := \{f_{\overrightarrow{v}} | \overrightarrow{v} \in \mathbb{F}_q^n \setminus \{\overrightarrow{0}\}\}$ i.e., the set of the predicates.

Definition 8. *A predicate encryption (PE) scheme for the class of inner-product predicates $\mathcal{F}$ and attributes Σ consists of probabilistic polynomial-time algorithms* $\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}$ *and* Dec. *They are given as follows:*

- Setup *takes as input security parameter 1^λ outputs (master) public key* pk *and (master) secret key* sk.
- KeyGen *takes as input the master public key* pk, *secret key* sk, *and predicate vector $\overrightarrow{v}$. It outputs a corresponding secret key* $\mathsf{sk}_{\overrightarrow{v}}$.
- Enc *takes as input the master public key* pk, *plaintext m in some associated plaintext space,* msg, *and attribute vector $\overrightarrow{x}$. It returns ciphertext c.*
- Dec *takes as input the master public key* pk, *secret key* $\mathsf{sk}_{\overrightarrow{v}}$ *and ciphertext c. It outputs either plaintext m or the distinguished symbol $\perp$.*

A PE scheme should have the following correctness property: for all $f_{\overrightarrow{v}} \in \mathcal{F}$ and $\overrightarrow{x} \in \Sigma$, for correctly generated pk, $\mathsf{sk}_{\overrightarrow{v}}$ and $c \xleftarrow{\mathsf{R}} \mathsf{Enc}(\mathsf{pk}, m, \overrightarrow{x})$, it holds that $m = \mathsf{Dec}(\mathsf{pk}, \mathsf{sk}_{\overrightarrow{v}}, c)$ if $f_{\overrightarrow{v}}(\overrightarrow{x}) = 1$. Otherwise, it holds with negligible probability.

Definition 9. *An inner-product predicate encryption scheme is* adaptively attribute-hiding (AH) against chosen plaintext attacks *if for all probabilistic polynomial-time adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following experiment is negligible in the security parameter.*

1. Setup *is run to generate keys* pk *and* sk, *and* pk *is given to $\mathcal{A}$.*
2. *$\mathcal{A}$ may adaptively make a polynomial number of key queries for predicate vectors, $\overrightarrow{v}$. In response, $\mathcal{A}$ is given the corresponding key* $\mathsf{sk}_{\overrightarrow{v}} \xleftarrow{\mathsf{R}} \mathsf{KeyGen}(\mathsf{sk}, \overrightarrow{v})$.
3. *$\mathcal{A}$ outputs challenge attribute vector $(\overrightarrow{x}^{(0)}, \overrightarrow{x}^{(1)})$ and challenge plaintexts $(m^{(0)}, m^{(1)})$, subject to the restriction that $\overrightarrow{v} \cdot \overrightarrow{x}^{(0)} \neq 0$ and $\overrightarrow{v} \cdot \overrightarrow{x}^{(1)} \neq 0$ for all the key queried predicate vectors, $\overrightarrow{v}$.*

4. *A random bit b is chosen. $\mathcal{A}$ is given $c^{(b)} \xleftarrow{\mathsf{R}} \mathsf{Enc}(\mathsf{pk}, m^{(b)}, \overrightarrow{x}^{(b)})$.*
5. *The adversary may continue to issue key queries for additional predicate vectors, $\overrightarrow{v}$, subject to the restriction that $\overrightarrow{v} \cdot \overrightarrow{x}^{(0)} \neq 0$ and $\overrightarrow{v} \cdot \overrightarrow{x}^{(1)} \neq 0$. $\mathcal{A}$ is given the corresponding key $\mathsf{sk}_{\overrightarrow{v}} \xleftarrow{\mathsf{R}} \mathsf{KeyGen}(\mathsf{sk}, \overrightarrow{v})$.*
6. *$\mathcal{A}$ outputs a bit b', and succeeds if $b' = b$.*

We define the advantage of $\mathcal{A}$ as the quantity $\mathsf{Adv}^{\mathsf{PE,AH}}_{\mathcal{A}}(\lambda) := \mathsf{Pr}[b' = b] - 1/2$.

Remark: In Definition 9, adversary $\mathcal{A}$ is not allowed to ask a key query for $\overrightarrow{v}$ such that $\overrightarrow{v} \cdot \overrightarrow{x}^{(b)} = 0$ for some $b \in \{0, 1\}$, while in the security definition in [27], such a key query is allowed provided that $m^{(0)} = m^{(1)}$ and $\overrightarrow{v} \cdot \overrightarrow{x}^{(b)} = 0$ for all $b \in \{0, 1\}$.

3.5 The Proposed PE Scheme

Construction

$$\begin{aligned}
&\mathsf{Setup}(1^\lambda, n): \ (\mathsf{param}_{\mathbb{V}}, \mathbb{B}, \mathbb{B}^*) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, 2n+3),\\
&\quad \widehat{\mathbb{B}} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n, \boldsymbol{b}_{2n+1}, \boldsymbol{b}_{2n+3}), \ \ \mathsf{sk} := \mathbb{B}^*, \ \ \mathsf{pk} := (1^\lambda, \mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}),\\
&\quad \text{return } \ \mathsf{sk}, \ \mathsf{pk}.\\
&\mathsf{KeyGen}(\mathsf{sk}, \overrightarrow{v} := (v_1, \ldots, v_n)): \ \sigma, \eta \xleftarrow{\mathsf{U}} \mathbb{F}_q,\\
&\quad \boldsymbol{k}^* := \sigma(\textstyle\sum_{i=1}^n v_i \boldsymbol{b}_i^*) + \boldsymbol{b}_{2n+1}^* + \eta \boldsymbol{b}_{2n+2}^*,\\
&\quad \text{return } \ \mathsf{sk}_{\overrightarrow{v}} := \boldsymbol{k}^*.\\
&\mathsf{Enc}(\mathsf{pk}, m \in \mathbb{G}_T, \overrightarrow{x} := (x_1, \ldots, x_n)): \ \delta_1, \delta_2, \zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q,\\
&\quad \boldsymbol{c}_1 := \delta_1(\textstyle\sum_{i=1}^n x_i \boldsymbol{b}_i) + \zeta \boldsymbol{b}_{2n+1} + \delta_2 \boldsymbol{b}_{2n+3}, \quad c_2 := g_T^\zeta m,\\
&\quad \text{return } (\boldsymbol{c}_1, c_2).\\
&\mathsf{Dec}(\mathsf{pk}, \boldsymbol{k}^*, (\boldsymbol{c}_1, c_2)): \ m' := c_2 / e(\boldsymbol{c}_1, \boldsymbol{k}^*),\\
&\quad \text{return } \ m'.
\end{aligned}$$

[**Correctness**] $\boldsymbol{k}^*$ and $\boldsymbol{c}_1$ can be expressed by $\boldsymbol{k}^* = (\sigma \overrightarrow{v}, 0, \ldots, 0, 1, \eta, 0)_{\mathbb{B}^*}$, and $\boldsymbol{c}_1 = (\delta_1 \overrightarrow{x}, 0, \ldots, 0, \zeta, 0, \delta_2)_{\mathbb{B}}$. Hence, $e(\boldsymbol{c}_1, \boldsymbol{k}^*) = g_T^{(\delta_1 \overrightarrow{x}, 0, \ldots, 0, \zeta, 0, \delta_2) \cdot (\sigma \overrightarrow{v}, 0, \ldots, 0, 1, \eta, 0)}$ $= g_T^{\delta_1 \sigma (\overrightarrow{x} \cdot \overrightarrow{v}) + \zeta}$, i.e., $e(\boldsymbol{c}_1, \boldsymbol{k}^*) = g_T^\zeta$ if $\overrightarrow{x} \cdot \overrightarrow{v} = 0$.

Security

Theorem 2. *The proposed PE scheme is adaptively attribute-hiding against chosen plaintext attacks under the n-eDDH assumption. For any adversary $\mathcal{A}$, there exist probabilistic machines $\mathcal{C}_k$ ($k = 0, \ldots, \nu$), whose running times are essentially the same as that of $\mathcal{A}$, such that for any security parameter λ,*

$$\mathsf{Adv}^{\mathsf{PE,AH}}_{\mathcal{A}}(\lambda) \leq \sum_{k=0}^{\nu} \mathsf{Adv}^{n\text{-}\mathsf{eDDH}}_{\mathcal{C}_k}(\lambda) + \frac{\nu}{q},$$

where ν is the maximum number of adversary $\mathcal{A}$'s key queries.

We will show Lemmas 6, 7, and 8 for the proof of Theorem 2. The proofs of these lemmas are given in the full version of this paper.

Definition 10. *Problem 1 is to guess* $\beta \in \{0,1\}$, *given* $(\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{e}_{\beta,i}\}_{i=1,..,n}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\beta}^{\mathsf{P1}}(1^{\lambda}, n)$, *where*

$$\begin{aligned}
&\mathcal{G}_{\beta}^{\mathsf{P1}}(1^{\lambda}, n): \ (\mathsf{param}_{\mathbb{V}}, \mathbb{B}, \mathbb{B}^*) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^{\lambda}, 2n+3),\\
&\quad \widehat{\mathbb{B}} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n, \boldsymbol{b}_{2n+1}, \boldsymbol{b}_{2n+3}), \quad \widehat{\mathbb{B}}^* := (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*, \boldsymbol{b}_{2n+1}^*, \boldsymbol{b}_{2n+2}^*),\\
&\quad \delta_1, \delta_{2,i} \xleftarrow{\mathsf{U}} \mathbb{F}_q, \quad \rho \xleftarrow{\mathsf{U}} \mathbb{F}_q^{\times}, \quad (u_{i,j}) \xleftarrow{\mathsf{U}} GL(n, \mathbb{F}_q) \ \text{ for } i,j = 1, \ldots, n,\\
&\quad \text{for } i = 1, \ldots, n,\\
&\qquad \boldsymbol{e}_{0,i} := \delta_1 \boldsymbol{b}_i + \delta_{2,i} \boldsymbol{b}_{2n+3},\\
&\qquad \boldsymbol{e}_{1,i} := \delta_1 \boldsymbol{b}_i + \rho \textstyle\sum_{j=1}^{n} u_{i,j} \boldsymbol{b}_{n+j} + \delta_{2,i} \boldsymbol{b}_{2n+3},\\
&\quad \text{return } \ (\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{e}_{\beta,i}\}_{i=1,\ldots,n}),
\end{aligned}$$

for $\beta \xleftarrow{\mathsf{U}} \{0,1\}$. *For a probabilistic machine* $\mathcal{B}$, *we define the advantage of* $\mathcal{B}$ *for Problem 1 as:*

$$\mathsf{Adv}_{\mathcal{B}}^{\mathsf{P1}}(\lambda) := \left| \mathsf{Pr}\left[\mathcal{B}(1^{\lambda}, \varrho) \to 1 \;\middle|\; \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_0^{\mathsf{P1}}(1^{\lambda}, n) \right] - \mathsf{Pr}\left[\mathcal{B}(1^{\lambda}, \varrho) \to 1 \;\middle|\; \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_1^{\mathsf{P1}}(1^{\lambda}, n) \right] \right|.$$

Lemma 6. *For any adversary* $\mathcal{B}$, *there is a probabilistic machine* $\mathcal{C}$, *whose running time is essentially the same as that of* $\mathcal{B}$, *such that for any security parameter* λ, $\mathsf{Adv}_{\mathcal{C}}^{n\text{-}\mathsf{eDDH}}(\lambda) = \mathsf{Adv}_{\mathcal{B}}^{\mathsf{P1}}(\lambda)$.

Definition 11. *Problem 2 is to guess* $\beta \in \{0,1\}$, *given* $(\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{h}_{\beta,i}^*, \boldsymbol{e}_i\}_{i=1,\ldots,n}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\beta}^{\mathsf{P2}}(1^{\lambda}, n)$, *where*

$$\begin{aligned}
&\mathcal{G}_{\beta}^{\mathsf{P2}}(1^{\lambda}, n): \ (\mathsf{param}_{\mathbb{V}}, \mathbb{B}, \mathbb{B}^*) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^{\lambda}, 2n+3),\\
&\quad \widehat{\mathbb{B}} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n, \boldsymbol{b}_{2n+1}, \boldsymbol{b}_{2n+3}), \quad \widehat{\mathbb{B}}^* := (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_{2n+2}^*),\\
&\quad \omega, \gamma_i, \delta \xleftarrow{\mathsf{U}} \mathbb{F}_q, \ \rho, \tau \xleftarrow{\mathsf{U}} \mathbb{F}_q^{\times},\\
&\quad (u_{i,j}) \xleftarrow{\mathsf{U}} GL(n, \mathbb{F}_q), \quad (z_{i,j}) := ((u_{i,j})^{-1})^{\mathrm{T}} \ \text{ for } i,j = 1, \ldots, n,\\
&\quad \text{for } i = 1, \ldots, n,\\
&\qquad \boldsymbol{h}_{0,i}^* := \omega \boldsymbol{b}_i^* + \gamma_i \boldsymbol{b}_{2n+2}^*,\\
&\qquad \boldsymbol{h}_{1,i}^* := \omega \boldsymbol{b}_i^* + \tau \textstyle\sum_{j=1}^{n} z_{i,j} \boldsymbol{b}_{n+j}^* + \gamma_i \boldsymbol{b}_{2n+2}^*,\\
&\qquad \boldsymbol{e}_i := \delta \boldsymbol{b}_i + \rho \textstyle\sum_{j=1}^{n} u_{i,j} \boldsymbol{b}_{n+j},\\
&\quad \text{return } \ (\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{h}_{\beta,i}^*, \boldsymbol{e}_i\}_{i=1,\ldots,n}),
\end{aligned}$$

for $\beta \xleftarrow{\mathsf{U}} \{0,1\}$. *For a probabilistic machine* $\mathcal{B}$, *the advantage of* $\mathcal{B}$ *for Problem 2,* $\mathsf{Adv}_{\mathcal{B}}^{\mathsf{P2}}(\lambda)$, *is similarly defined as in Definition 10.*

Lemma 7. *For any adversary $\mathcal{B}$, there is a probabilistic machine $\mathcal{C}$, whose running time is essentially the same as that of $\mathcal{B}$, such that for any security parameter λ, $\mathsf{Adv}^{n\text{-}\mathsf{eDDH}}_{\mathcal{C}}(\lambda) = \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}}(\lambda)$.*

Lemma 8. *Let $C := \{(\overrightarrow{x}, \overrightarrow{v}) | \overrightarrow{x} \cdot \overrightarrow{v} \neq 0\} \subset V \times V^*$ where V is n-dimensional vector space $\mathbb{F}_q^n$, and V^* its dual. For all $(\overrightarrow{x}, \overrightarrow{v}) \in C$, for all $(\overrightarrow{r}, \overrightarrow{w}) \in C$,*

$$\Pr_{\substack{Z \xleftarrow{\mathsf{U}} GL(n, \mathbb{F}_q), \\ \rho, \tau \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times}} [\overrightarrow{x}(\rho U) = \overrightarrow{r} \ \wedge \ \overrightarrow{v}(\tau Z) = \overrightarrow{w}] = \frac{1}{s},$$

where $U := (Z^{-1})^{\mathrm{T}}$ and $s := \sharp C \ (= (q^n - 1)(q^n - q^{n-1}))$.

Proof Outline of Theorem 2. To prove the security, we employ Game 0 (original adaptive-security game) through Game 3. Roughly speaking, the (normal) target ciphertext is changed to a *semi-functional* ciphertext in Game 1 (or Game 2-0), the k-th secret key replied to the adversary is changed to a *semi-functional* key in Game 2-k ($k = 1, \ldots, \nu$), and the (semi-functional) target ciphertext is changed to perfectly *randomized* key in Game 3, whose advantage is 0.

A *normal* secret key $\boldsymbol{k}^{*\,\mathsf{norm}}_{\overrightarrow{v}}$ (with predicate vector $\overrightarrow{v}$) is a correct form of the secret key of the proposed PE scheme, i.e., $\boldsymbol{k}^{*\,\mathsf{norm}}_{\overrightarrow{v}} := \sigma(\sum_{i=1}^n v_i \boldsymbol{b}^*_i) + \boldsymbol{b}^*_{2n+1} + \eta \boldsymbol{b}^*_{2n+2} = (\sigma\overrightarrow{v}, \overrightarrow{0}_n, 1, \eta, 0)_{\mathbb{B}^*}$, where $\overrightarrow{0}_n := (\overbrace{0, \cdots, 0}^{n})$. Similarly, a *normal* ciphertext (with attribute $\overrightarrow{x}$) is $(\boldsymbol{c}^{\mathsf{norm}}_{\overrightarrow{x}}, c_2)$ with $\boldsymbol{c}^{\mathsf{norm}}_{\overrightarrow{x}} := \delta_1(\sum_{i=1}^n x_i \boldsymbol{b}_i) + \zeta \boldsymbol{b}_{2n+1} + \delta_2 \boldsymbol{b}_{2n+3} = (\delta_1 \overrightarrow{x}, \overrightarrow{0}_n, \zeta, 0, \delta_2)_{\mathbb{B}}$. (Hereafter we will ignore c_2 since c_2 is always correctly generated.) A *semi-functional* secret key is $\boldsymbol{k}^{*\,\mathsf{semi}}_{\overrightarrow{v}} := (\sigma\overrightarrow{v}, \overrightarrow{r}, 1, \eta, 0)_{\mathbb{B}^*}$ and a *semi-functional* ciphertext is $\boldsymbol{c}^{\mathsf{semi}}_{\overrightarrow{x}} := (\delta_1 \overrightarrow{x}, \overrightarrow{s}, \zeta, 0, \delta_2)_{\mathbb{B}}$, where $\overrightarrow{r}, \overrightarrow{s} \xleftarrow{\mathsf{U}} \mathbb{F}_q^n$. If $\overrightarrow{x} \cdot \overrightarrow{v} = 0$, then $e(\boldsymbol{c}^{\mathsf{norm}}_{\overrightarrow{x}}, \boldsymbol{k}^{*\,\mathsf{norm}}_{\overrightarrow{v}}) = e(\boldsymbol{c}^{\mathsf{norm}}_{\overrightarrow{x}}, \boldsymbol{k}^{*\,\mathsf{semi}}_{\overrightarrow{v}}) = e(\boldsymbol{c}^{\mathsf{semi}}_{\overrightarrow{x}}, \boldsymbol{k}^{*\,\mathsf{norm}}_{\overrightarrow{v}}) = g_T^{\zeta}$, which leads to correct decryption. In contrast, $e(\boldsymbol{c}^{\mathsf{semi}}_{\overrightarrow{x}}, \boldsymbol{k}^{*\,\mathsf{semi}}_{\overrightarrow{v}}) = g_T^{\overrightarrow{s} \cdot \overrightarrow{r} + \zeta}$, which is uniformly and independently distributed over $\mathbb{F}_q$ since $\overrightarrow{r}, \overrightarrow{s} \xleftarrow{\mathsf{U}} \mathbb{F}_q^n$, (i.e., leads to random decryption).

To prove that the advantage gap between Games 0 and 1 is bounded by the advantage of Problem 1 (to guess $\beta \in \{0, 1\}$), we construct a simulator of the challenger of Game 0 (or 1) (against an adversary $\mathcal{A}$) by using an instance with $\beta \xleftarrow{\mathsf{U}} \{0, 1\}$ of Problem 1. We then show that the distribution of the secret keys and target ciphertext replied by the simulator is equivalent to those of Game 0 when $\beta = 0$ and Game 1 when $\beta = 1$. That is, the advantage of Problem 1 is equivalent to the advantage gap between Games 0 and 1 (Lemma 9). The advantage of Problem 1 is proven to be equivalent to that of the n-eDDH assumption (Lemma 6).

The advantage gap between Games 2-$(k-1)$ and 2-k is similarly shown to be bounded by the advantage of Problem 2 (i.e., of the n-eDDH assumption) $+1/q$ (Lemmas 7 and 10).

Problem 2 is based on our key trick (explained in Section 3.1). Here, we introduce *special form of semi-functional* keys and ciphertexts such that $\boldsymbol{k}^{*\ \mathsf{spec.semi}}_{\overrightarrow{v}} := (\sigma\overrightarrow{v}, (\tau\overrightarrow{v}Z), 1, \eta, 0)_{\mathbb{B}^*}$, and $\boldsymbol{c}^{\mathsf{spec.semi}}_{\overrightarrow{x}} := (\delta\overrightarrow{x}, (\rho\overrightarrow{x}U), \zeta, 0, \delta_2)_{\mathbb{B}}$, where Z is a random regular $(n \times n)$-matrix, $U := (Z^{-1})^{\mathrm{T}}$, and $\tau, \rho \xleftarrow{\mathsf{U}} \mathbb{F}_q$.

$\boldsymbol{k}^{*\ \mathsf{spec.semi}}_{\overrightarrow{v}}$ can decrypt $\boldsymbol{c}^{\mathsf{spec.semi}}_{\overrightarrow{x}}$ for *all* vectors $\overrightarrow{x}$ with $\overrightarrow{v}\cdot\overrightarrow{x} = 0$, since $(\tau\overrightarrow{v}Z)\cdot(\rho\overrightarrow{x}U) = \tau\rho(\overrightarrow{v}\cdot\overrightarrow{x})$, i.e., $e(\boldsymbol{c}^{\mathsf{spec.semi}}_{\overrightarrow{x}}, \boldsymbol{k}^{*\ \mathsf{spec.semi}}_{\overrightarrow{v}}) = g^{(\delta_1\sigma+\tau\rho)(\overrightarrow{v}\cdot\overrightarrow{x})+\zeta}$. In addition, $(\tau\overrightarrow{v}Z)$ and $(\rho\overrightarrow{x}U)$ are uniformly and pairwise-independently distributed (i.e., equivalently distributed to $(\overrightarrow{r}, \overrightarrow{s}) \xleftarrow{\mathsf{U}} (\mathbb{F}_q^n)^2 \backslash \{(\overrightarrow{r}, \overrightarrow{s}) \mid \overrightarrow{r}\cdot\overrightarrow{s} = 0\}$), when $\overrightarrow{v}\cdot\overrightarrow{x} \neq 0$ (Lemma 8). Therefore, the joint distribution of $\boldsymbol{k}^{*\ \mathsf{spec.semi}}_{\overrightarrow{v}}$ and $\boldsymbol{c}^{\mathsf{spec.semi}}_{\overrightarrow{x}}$ is equivalent to that of an independent pair of $\boldsymbol{k}^{*\ \mathsf{semi}}_{\overrightarrow{v}}$ and $\boldsymbol{c}^{\mathsf{semi}}_{\overrightarrow{x}}$ (except with probability $1/q$), when $\overrightarrow{v}\cdot\overrightarrow{x} \neq 0$.

Finally we show that Game 2-ν can be conceptually changed to Game 3 by using the fact that n elements of $\mathbb{B}$, $(\boldsymbol{b}_{n+1}, \ldots, \boldsymbol{b}_{2n})$, are secret to the adversary (Lemma 11).

Proof of Theorem 2: To prove Theorem 2, we consider the following $(\nu + 3)$ games.

Game 0. Original game.

Game 1. Same as Game 0 except that the target ciphertext $(\boldsymbol{c}_1, c_2)$ for challenge plaintexts $(m^{(0)}, m^{(1)})$ and challenge attributes $(\overrightarrow{x}^{(0)}, \overrightarrow{x}^{(1)})$ is

$$\boldsymbol{c}_1 := \delta_1(\textstyle\sum_{i=1}^n x_i^{(b)}\boldsymbol{b}_i) + \sum_{i=1}^n w_i\boldsymbol{b}_{n+i} + \zeta\boldsymbol{b}_{2n+1} + \delta_2\boldsymbol{b}_{2n+3}, \quad c_2 := g_T^\zeta m^{(b)},$$

where $\delta_1, \delta_2, \zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $b \xleftarrow{\mathsf{U}} \{0,1\}$, $(x_1^{(b)}, \ldots, x_n^{(b)}) := \overrightarrow{x}^{(b)}$, and $(w_1, \ldots, w_n) \xleftarrow{\mathsf{U}} \mathbb{F}_q^n \setminus \{\overrightarrow{0}\}$.

Game 2-k $(k = 1, \ldots, \nu)$. Game 2-0 is Game 1. Game 2-k is the same as Game 2-$(k-1)$ except the reply to the k-th key query for $\overrightarrow{v} := (v_1, \ldots, v_n)$ is:

$$\boldsymbol{k}^* := \sigma(\textstyle\sum_{i=1}^n v_i\boldsymbol{b}_i^*) + \sum_{i=1}^n r_i\boldsymbol{b}_{n+i}^* + \boldsymbol{b}_{2n+1}^* + \eta\boldsymbol{b}_{2n+2}^*,$$

where $\sigma, \eta \xleftarrow{\mathsf{U}} \mathbb{F}_q$ and $\overrightarrow{r} := (r_1, \ldots, r_n) \xleftarrow{\mathsf{U}} \mathbb{F}_q^n$.

Game 3. Same as Game 2-ν except that the target ciphertext $(\boldsymbol{c}_1, c_2)$ for challenge plaintexts $(m^{(0)}, m^{(1)})$ and challenge attributes $(\overrightarrow{x}^{(0)}, \overrightarrow{x}^{(1)})$ is

$$\boldsymbol{c}_1 := \textstyle\sum_{i=1}^n x_i'\boldsymbol{b}_i + \sum_{i=1}^n w_i\boldsymbol{b}_{n+i} + \zeta'\boldsymbol{b}_{2n+1} + \delta_2\boldsymbol{b}_{2n+3}, \quad c_2 := g_T^\zeta m^{(b)},$$

where $x_1', \ldots, x_n', \delta_2, \zeta, \zeta' \xleftarrow{\mathsf{U}} \mathbb{F}_q, b \xleftarrow{\mathsf{U}} \{0,1\}$, and $(w_1, \ldots, w_n) \xleftarrow{\mathsf{U}} \mathbb{F}_q^n \setminus \{\overrightarrow{0}\}$. In particular, we note that $(x_1', \ldots, x_n')$ and ζ' are chosen uniformly and independently from $\overrightarrow{x}^{(0)}, \overrightarrow{x}^{(1)}$ and ζ.

Let $\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda)$ be $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{PE,AH}}(\lambda)$ in Game 0, and $\mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda), \mathsf{Adv}_{\mathcal{A}}^{(2\text{-}k)}(\lambda), \mathsf{Adv}_{\mathcal{A}}^{(3)}(\lambda)$ be the advantage of $\mathcal{A}$ in Game $1, 2\text{-}k, 3$, respectively. It is clear that $\mathsf{Adv}_{\mathcal{A}}^{(3)}(\lambda) = 0$ by Lemma 12.

We will use three lemmas (Lemmas 9, 10, 11) that evaluate the gaps between pairs of $\mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda), \mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda), \mathsf{Adv}^{(2\text{-}k)}_{\mathcal{A}}(\lambda)$ $(k = 1, \ldots, \nu), \mathsf{Adv}^{(3)}_{\mathcal{A}}(\lambda)$. From these lemmas, we obtain $\mathsf{Adv}^{\mathsf{PE,AH}}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda) \leq \left|\mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda)\right| + \sum_{k=1}^{\nu} \left|\mathsf{Adv}^{(2\text{-}(k-1))}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(2\text{-}k)}_{\mathcal{A}}(\lambda)\right| + \left|\mathsf{Adv}^{(2\text{-}\nu)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(3)}_{\mathcal{A}}(\lambda)\right| + \mathsf{Adv}^{(3)}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}_0}(\lambda) + \sum_{k=1}^{\nu} \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}_k}(\lambda) + \frac{\nu}{q}$. From Lemmas 6 and 7, there exist probabilistic machines $\mathcal{C}_k$ $(k = 0, \ldots, \nu)$, whose running times are essentially the same as those of $\mathcal{B}_k$, respectively, such that $\mathsf{Adv}^{n\text{-}\mathsf{eDDH}}_{\mathcal{C}_0}(\lambda) = \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}_0}(\lambda)$ and $\mathsf{Adv}^{n\text{-}\mathsf{eDDH}}_{\mathcal{C}_k}(\lambda) = \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}_k}(\lambda)$ $(k = 1, \ldots, \nu)$. Hence, $\mathsf{Adv}^{\mathsf{PE,AH}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}_0}(\lambda) + \sum_{k=1}^{\nu} \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}_k}(\lambda) + \frac{\nu}{q} \leq \sum_{k=0}^{\nu} \mathsf{Adv}^{n\text{-}\mathsf{eDDH}}_{\mathcal{C}_k}(\lambda) + \frac{\nu}{q}$. This completes the proof of Theorem 2. □

The proofs of the following lemmas appear in the full version of this paper.

Lemma 9. *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}_0$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda)| = \mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}_0}(\lambda)$.*

Lemma 10. *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}_k$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathsf{Adv}^{(2\text{-}(k-1))}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(2\text{-}k)}_{\mathcal{A}}(\lambda)| \leq \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}_k}(\lambda) + \frac{1}{q}$.*

Lemma 11. *For any adversary $\mathcal{A}$, $\mathsf{Adv}^{(2\text{-}\nu)}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{(3)}_{\mathcal{A}}(\lambda)$.*

Lemma 12. *For any adversary $\mathcal{A}$, $\mathsf{Adv}^{(3)}_{\mathcal{A}}(\lambda) = 0$.*

3.6 The Proposed HPE Scheme

The definition of HPE and key idea for the proposed HPE (and the correctness of the HPE) are given in the full version of this paper.

Construction

$$\mathsf{Setup}(1^{\lambda}, \overrightarrow{\mu} := (n, d; \mu_1, \ldots, \mu_d)) : \ (\mathsf{param}_{\mathbb{V}}, \mathbb{B}, \mathbb{B}^*) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^{\lambda}, 2n+3),$$

$$\widehat{\mathbb{B}} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n, \boldsymbol{b}_{2n+1}, \boldsymbol{b}_{2n+3}), \quad \mathsf{sk} := \mathbb{B}^*, \quad \mathsf{pk} := (1^{\lambda}, \mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}),$$

return sk, pk.

$$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, (\overrightarrow{v}_1, \ldots, \overrightarrow{v}_{\ell}) := ((v_1, \ldots, v_{\mu_1}), \ldots, (v_{\mu_{\ell-1}+1}, \ldots, v_{\mu_{\ell}})) :$$

$$\sigma_{\mathsf{dec},t}, \eta_{\mathsf{dec}}, \ \sigma_{\mathsf{ran},j,t}, \eta_{\mathsf{ran},j} \ (j = 1, .., \ell+1), \ \sigma_{\mathsf{del},j,t}, \eta_{\mathsf{del},j} \ (j = 1, .., n), \ \psi \xleftarrow{\mathsf{U}} \mathbb{F}_q$$

for $t = 1, \ldots, \ell$,

$$\boldsymbol{k}^*_{\ell,\mathsf{dec}} := \textstyle\sum_{t=1}^{\ell} \sigma_{\mathsf{dec},t}(\sum_{i=\mu_{t-1}+1}^{\mu_t} v_i \boldsymbol{b}^*_i) + \boldsymbol{b}^*_{2n+1} + \eta_{\mathsf{dec}} \boldsymbol{b}^*_{2n+2},$$

$$\boldsymbol{k}^*_{\ell,\mathsf{ran},j} := \textstyle\sum_{t=1}^{\ell} \sigma_{\mathsf{ran},j,t}(\sum_{i=\mu_{t-1}+1}^{\mu_t} v_i \boldsymbol{b}^*_i) + \eta_{\mathsf{ran},j} \boldsymbol{b}^*_{2n+2} \quad \text{for } j = 1, \ldots, \ell+1,$$

$$\boldsymbol{k}^*_{\ell,\mathsf{del},j} := \textstyle\sum_{t=1}^{\ell} \sigma_{\mathsf{del},j,t}(\sum_{i=\mu_{t-1}+1}^{\mu_t} v_i \boldsymbol{b}^*_i) + \psi \boldsymbol{b}^*_j + \eta_{\mathsf{del},j} \boldsymbol{b}^*_{2n+2}$$

for $j = \mu_{\ell} + 1, \ldots, n$,

return $\overrightarrow{\boldsymbol{k}}^*_{\ell} := (\boldsymbol{k}^*_{\ell,\mathsf{dec}}, \boldsymbol{k}^*_{\ell,\mathsf{ran},1}, \ldots, \boldsymbol{k}^*_{\ell,\mathsf{ran},\ell+1}, \boldsymbol{k}^*_{\ell,\mathsf{del},\mu_{\ell}+1}, \ldots, \boldsymbol{k}^*_{\ell,\mathsf{del},n})$.

$\mathsf{Enc}(\mathsf{pk}, m \in \mathbb{G}_T, (\overrightarrow{x}_1, \ldots, \overrightarrow{x}_\ell) := ((x_1, \ldots, x_{\mu_1}), \ldots, (x_{\mu_{\ell-1}+1}, \ldots, x_{\mu_\ell})) :$

$(\overrightarrow{x}_{\ell+1}, \ldots, \overrightarrow{x}_d) \xleftarrow{\mathsf{U}} \mathbb{F}_q^{\mu_{\ell+1}-\mu_\ell} \times \cdots \times \mathbb{F}_q^{n-\mu_{d-1}}, \quad \delta_1, \ldots, \delta_\ell, \delta_{2n+3}, \zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q,$

$\boldsymbol{c}_1 := \sum_{t=1}^{\ell} \delta_t (\sum_{i=\mu_{t-1}+1}^{\mu_t} x_i \boldsymbol{b}_i) + \zeta \boldsymbol{b}_{2n+1} + \delta_{2n+3} \boldsymbol{b}_{2n+3}, \quad c_2 := g_T^{\zeta} m,$

return $(\boldsymbol{c}_1, c_2)$.

$\mathsf{Dec}(\mathsf{pk}, \boldsymbol{k}^*_{\ell,\mathsf{dec}}, \boldsymbol{c}_1, c_2) : \ m' := c_2 / e(\boldsymbol{c}_1, \boldsymbol{k}^*_{\ell,\mathsf{dec}}),$

return m'.

$\mathsf{Delegate}_\ell(\mathsf{pk}, \overrightarrow{\boldsymbol{k}}^*_\ell, \overrightarrow{v}_{\ell+1} := (v_{\mu_\ell+1}, \ldots, v_{\mu_{\ell+1}})) :$

$\alpha_{\mathsf{dec},t}, \sigma_{\mathsf{dec}}, \ \alpha_{\mathsf{ran},j,t}, \sigma_{\mathsf{ran},j} \ (j = 1, .., \ell+2), \ \alpha_{\mathsf{del},j,t}, \sigma_{\mathsf{del},j} \ (j = 1, .., n), \ \psi' \xleftarrow{\mathsf{U}} \mathbb{F}_q$ for $t = 1, \ldots, \ell+1$,

$\boldsymbol{k}^*_{\ell+1,\mathsf{dec}} := \boldsymbol{k}^*_{\ell,\mathsf{dec}} + \sum_{t=1}^{\ell+1} \alpha_{\mathsf{dec},t} \boldsymbol{k}^*_{\ell,\mathsf{ran},t} + \sigma_{\mathsf{dec}} (\sum_{i=\mu_\ell+1}^{\mu_{\ell+1}} v_i \boldsymbol{k}^*_{\ell,\mathsf{del},i}),$

$\boldsymbol{k}^*_{\ell+1,\mathsf{ran},j} := \sum_{t=1}^{\ell+1} \alpha_{\mathsf{ran},j,t} \boldsymbol{k}^*_{\ell,\mathsf{ran},t} + \sigma_{\mathsf{ran},j} (\sum_{i=\mu_\ell+1}^{\mu_{\ell+1}} v_i \boldsymbol{k}^*_{\ell,\mathsf{del},i})$ for $j = 1, .., \ell+2$,

$\boldsymbol{k}^*_{\ell+1,\mathsf{del},j} := \sum_{t=1}^{\ell+1} \alpha_{\mathsf{del},j,t} \boldsymbol{k}^*_{\ell,\mathsf{ran},t} + \sigma_{\mathsf{del},j} (\sum_{i=\mu_\ell+1}^{\mu_{\ell+1}} v_i \boldsymbol{k}^*_{\ell,\mathsf{del},i}) + \psi' \boldsymbol{k}^*_{\ell,\mathsf{del},j}$ for $j = \mu_{\ell+1}+1, \ldots, n$,

return $\overrightarrow{\boldsymbol{k}}^*_{\ell+1} := (\boldsymbol{k}^*_{\ell+1,\mathsf{dec}}, \boldsymbol{k}^*_{\ell+1,\mathsf{ran},1}, .., \boldsymbol{k}^*_{\ell+1,\mathsf{ran},\ell+2}, \boldsymbol{k}^*_{\ell+1,\mathsf{del},\mu_{\ell+1}+1}, .., \boldsymbol{k}^*_{\ell+1,\mathsf{del},n})$.

Remark: A PE scheme with general delegation is given in the full version of this paper.

Security

Theorem 3. *The proposed HPE scheme is adaptively attribute-hiding against chosen plaintext attacks under the n-eDDH assumption. For any adversary $\mathcal{A}$, there exist probabilistic machines, $\mathcal{C}_0$ and $\mathcal{C}_{(k,j)}$ $(k = 1, \ldots, \nu; \ j = 1, \ldots, n+1)$ whose running times are essentially the same as that of $\mathcal{A}$, such that for any security parameter λ,*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{HPE,AH}}(\lambda) < \mathsf{Adv}_{\mathcal{C}_0}^{n\text{-}\mathsf{eDDH}}(\lambda) + \sum_{k=1}^{\nu} \sum_{j=1}^{n+1} \mathsf{Adv}_{\mathcal{C}_{(k,j)}}^{n\text{-}\mathsf{eDDH}}(\lambda) + \frac{(n+4)\nu}{q},$$

where ν is the maximum number of adversary $\mathcal{A}$'s key queries.

The proof is given in the full version of this paper.

References

1. Al-Riyami, S., Malone-Lee, J., Smart, N.: Escrow-free encryption supporting cryptographic workflow. Int. J. Inf. Sec. 5, 217–229 (2006)
2. Bagga, W., Molva, R., Crosta, S.: Policy-based encryption schemes from bilinear pairings. In: ASIACCS, p. 368 (2006)
3. Barbosa, M., Farshim, P.: Secure cryptographic workflow in the standarad model. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 379–393. Springer, Heidelberg (2006)

4. Beimel, A.: Secure schemes for secret sharing and key distribution. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel (1996)
5. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: Proceedings of the IEEE Symposium on Security and Privacy (2007)
6. 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)
7. 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)
8. Boneh, D., Boyen, X., Goh, E.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
9. 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)
10. Boneh, D., Goh, E., Nissim, K.: Evaluating 2-dnf formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–342. Springer, Heidelberg (2005)
11. 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)
12. 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)
13. 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)
14. Bradshaw, R., Holt, J., Seamons, K.: Concealing complex policies with hidden credentials. In: CCS, pp. 146–157 (2004)
15. 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)
16. 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)
17. Chase, M.: Multi-authority attribute based encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 515–534. Springer, Heidelberg (2007)
18. Cheung, L., Newport, C.: Provably secure ciphertext policy abe. In: CCS, 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. 26–28. Springer, Heidelberg (2001)
20. Freeman, D.M.: Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In: EUROCRYPT (2010)
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. 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)

23. Gentry, C., Silverberg, A.: Hierarchical id-based cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002)
24. 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)
25. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute Based Encryption for Fine-Grained Access Conrol of Encrypted Data. In: CCS (2006)
26. Horwitz, J., Lynn, B.: Toward hierarchical identity-based encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 466–481. Springer, Heidelberg (2002)
27. 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)
28. 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)
29. Miklau, G., Suciu, D.: Controlling access to published data using cryptography. In: VLDB, pp. 898–909 (2003)
30. Okamoto, T., Takashima, K.: Homomorphic encryption and signatures from vector decomposition. In: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5209, pp. 57–74. Springer, Heidelberg (2008)
31. 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)
32. Ostrovksy, R., Sahai, A., Waters, B.: Attribute Based Encryption with Non-Monotonic Access Structures. In: CCS (2007)
33. Pirretti, M., Traynor, P., McDaniel, P., Waters, B.: Secure attribute-based systems. In: CCS, pp. 99–112 (2006)
34. Sahai, A., Waters, B.: Fuzzy Identity Based Encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (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. 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)
37. Smart, N.: Access control using pairing based cryptography. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 111–121. Springer, Heidelberg (2003)
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. Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. Cryptology ePrint Archive, Report 2008/290 (2008)
40. 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)

Secure Obfuscation for Encrypted Signatures

Satoshi Hada

IBM Research, Tokyo
satoshih@jp.ibm.com

Abstract. Obfuscation is one of the most intriguing open problems in cryptography and only a few positive results are known. In TCC'07, Hohenberger et al. proposed an obfuscator for a re-encryption functionality, which takes a ciphertext for a message encrypted under Alice's public key and transforms it into a ciphertext for the same message under Bob's public key [24]. It is the first complicated cryptographic functionality that can be securely obfuscated, but obfuscators for such cryptographic functionalities are still elusive. In this paper, we consider obfuscation for encrypted signature (ES) functionalities, which generate a signature on a given message under Alice's secret signing key and encrypt the signature under Bob's public encryption key. We propose a special ES functionality, which is the sequential composition of Waters's signature scheme [33] and the linear encryption scheme proposed by Boneh, Boyen, and Shacham [5], and construct a secure obfuscator for it. We follow the security argument by Hohenberger et al. to prove that our proposed obfuscator satisfies a virtual black-box property (VBP), which guarantees that the security of the signature scheme is preserved even when adversaries are given an obfuscated program. Our security argument is in the standard model.

Keywords: Obfuscation, encrypted signatures, signcryption, bilinear map.

1 Introduction

An obfuscator is a tool to convert a program into a new *unintelligible* program while preserving the functionality. Several formal definitions have been proposed so far [22,3,27,34,20,23,24,21,9,12]. Informally, obfuscators should satisfy the following two requirements: (1) functionality: the obfuscated program has the same functionality as the original one and (2) virtual black-box property (VBP): whatever one can efficiently compute given the obfuscated program can be computed given black-box access to the functionality. The functionality requirement is the syntactic requirement while the VBP is the security requirement to capture the unintelligibility of obfuscated programs.

As discussed in [3], obfuscators, if they exist, would have a wide variety of cryptographic applications including software protection, fully homomorphic encryption, removing random oracles, and transforming private-key encryption schemes into public-key encryption schemes. Unfortunately, the impossibility

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 92–112, 2010.

of generic obfuscation have been shown in [3,20] even under very weak VBP definitions, which require that any *predicate* that can be efficiently computed from an obfuscated program can also be efficiently computed given black-box access to the functionality. Specifically, they showed the existence of (contrived) functionalities for which no obfuscator can satisfy the weak VBPs. However, the negative results do not rule out the possibility that there exists an obfuscator for a *specific* functionality. Indeed, some positive results are known for point functions [8,11,27,34,20,16,23,9,12]. In spite of these positive results, obfuscators for traditional cryptographic functionalities have remained elusive.

In TCC'07, Hohenberger et al. proposed an obfuscator for a re-encryption functionality [24]. It is the first complicated cryptographic functionality that can be securely obfuscated. A re-encryption functionality for Alice and Bob takes a ciphertext for a message encrypted under Alice's public key and transforms it into a ciphertext for the same message under Bob's public key. The naive program contains both Alice's secret key and Bob's public key, and it simply decrypts the input ciphertext and encrypts the plain message. Clearly, this program reveals Alice's secret key to any party executing the program. If Alice can securely obfuscate the program, the VBP ensures that any party learns no more from the obfuscated program than it does from black-box access to the functionality. In particular, the obfuscated program does not reveal Alice's secret key and cannot be used for eavesdropping. Hohenberger et al. constructed a special encryption scheme and a secure obfuscator for the re-encryption functionality. Their security argument is based on a new VBP definition suitable for cryptographic functionalities. It ensures that the security of cryptographic functionalities can be preserved even when adversaries are given obfuscated programs (See the discussion in [24]). They showed that the security of their proposed encryption scheme is preserved even when adversaries are given an obfuscated re-encryption program.

From both the theoretical and practical perspectives, it is important to investigate the possibility of secure obfuscation for more cryptographic functionalities. In this paper, we consider obfuscation for encrypted signature (ES) functionalities. An ES functionality for Alice and Bob generates a signature on a given message under Alice's secret signing key and then encrypts the signature under Bob's public encryption key. As in re-encryption, the naive program contains Alice's secret key and Bob's public key, and reveals Alice's secret key. If Alice can securely obfuscate the program, the VBP ensures that any party executing the obfuscated program cannot forge Alice's signature. We propose a special pair of signature and encryption schemes and construct a secure obfuscator for the ES functionality. Also, we follow the security argument by Hohenberger et al. in [24] to show that the security of signature scheme is preserved even when adversaries are given an obfuscated ES program.

We believe that there are many useful applications of our proposed obfuscation. We informally describe an application to secure Webmail services. ES should not be confused with Signature-then-Encryption or Sign-then-Encrypt (StE). The StE functionality signs a given message and then encrypts both the message and signature. StE is the most widely-used approach to constructing a signcryption scheme [2,35][1]. On the other hand, the ES functionality does not encrypt the message itself and we can not necessarily use it as a signcryption scheme. However, as shown in Appendix A, we believe that one can use an ES functionality as a building block to construct a signcryption functionality and that an obfuscator for the ES functionality can be used to obfuscate the signcryption functionality. Potential target applications include Webmail services such as Yahoo! Mail and Google's Gmail where users use e-mail services via Web browsers. If users want to signcrypt their messages and their Web browsers do not have the required capability then their Webmail providers need to signcrypt their messages on behalf of them. This means that users need to securely delegate their signing capability to the Webmail providers. The combination of the proposed obfuscation for an ES functionality presented in this paper and the obfuscation for the signcryption scheme constructed in Appendix A would provide a solution. For example, even if a malicious operator inside Webmail providers is given an obfuscated signcryption program for Alice-to-Bob, he/she can neither forge Alice's signatures nor perform the signcryption operation for Alice-to-Carol. However, we must be careful. The obfuscation does not prevent the malicious operator from performing the signcryption for Alice-to-Bob. Also, if the malicious operator has access to Bob's secret decryption key, he/she can forge Alice's signatures (In the case of our proposed obfuscations, what is worse is that he/she can extract Alice's secret signing key from the obfuscated program). The formal security argument for the signcryption and obfuscation outlined in Appendix A is outside the scope of this proceedings version.

1.1 Basic Idea

Our obfuscation is based on the following basic idea: *We construct a special pair of signature and encryption schemes such that generating a signature on a message and then encrypting the signature is functionally equivalent to encrypting the signing key and then generating a signature on the message under the encrypted signing key. The former process is the ES functionality. In the latter process, "encrypting the signing key" can be viewed as an obfuscation of the ES functionality and "generating a signature on the message under the encrypted signing key" corresponds to executing the obfuscated program.*

We informally describe how to construct such a special pair using the BLS signature scheme proposed by Boneh et al. [7]. Let $(q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$ be a parameter for a bilinear map, where both $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of prime order q, $\mathbf{e}$ is an efficient bilinear mapping from $\mathbb{G} \times \mathbb{G}$ to $\mathbb{G}_T$, and g is a generator of $\mathbb{G}$. A public verification and secret signing key pair is (v, s) such that $v = g^s$, where s is a random number in Z_q^*. Given a message m, the signature σ is calculated as $H(m)^s$, where $H : \{0,1\}^* \to \mathbb{G}$ is a hash function. We can verify the signature by checking the equality $\mathbf{e}(g, \sigma) = \mathbf{e}(H(m), v)$. If the computational Diffie-Hellman

[1] Following [2], we use the term "signcryption" for any scheme achieving both privacy and authenticity in the public key setting.

problem is hard, the scheme is secure in the random oracle model [4], where H is modeled as a random oracle. Also, we use a secure key encapsulation mechanism (KEM) to encrypt the signature value $\sigma = H(m)^s$. Let $\mathsf{KEM.Enc}(pk)$ be the encryption algorithm of the secure KEM. Given a public encryption key pk, it generates a pair of a random key r and its ciphertext c. Given $\mathsf{KEM.Enc}$, we define an encryption algorithm Enc, which takes as input a plaintext $p(= H(m)^s) \in \mathbb{G}$ and a public key pk, generates $(r, c) \leftarrow \mathsf{KEM.Enc}(pk)$, and outputs (c, p^r) as the ciphertext. The key and message spaces of $\mathsf{KEM.Enc}$ and Enc are $\mathbb{Z}_q^*$ and $\mathbb{G}$, respectively. The decryption is straightforward (You can decrypt c to recover r and then p). Then we consider the ES functionality defined as the sequential composition of the BLS signing algorithm and the encryption algorithm Enc. That is, given a message m, it computes the signature $H(m)^s$, generates $(r, c) \leftarrow \mathsf{KEM.Enc}(pk)$, and outputs $(c, H(m)^{sr})$. The naive program implementing the ES functionality contains (s, pk) and reveals s. Our goal is to obfuscate it.

Approach 1. Given the naive program, we extract (s, pk) and encrypt s using $\mathsf{KEM.Enc}$. Specifically, we generate $(r, c) \leftarrow \mathsf{KEM.Enc}(pk)$ and compute $sr \bmod q$, where (c, sr) is an encryption of the secret signing key s. It reveals no information on s since $\mathsf{KEM.Enc}$ is secure. However, using it, we can still compute an encryption of the valid signature of a given message m. That is, we can construct an obfuscated program $C_{c,sr}$ containing (c, sr), which takes as input m and outputs $(c, H(m)^{sr})$. Note that the output is an encryption of the valid signature $H(m)^s$ by Enc. The problem here is that $C_{c,sr}$ does not preserve the *probabilistic* ES functionality since it is *deterministic*. If Enc is rerandomizable with pk[2], we can fix the problem simply by rerandomizing $(c, H(m)^{sr})$. That is, we can construct a new obfuscated program $C_{c,sr,pk}$ containing (c, sr, pk), which takes as input m, computes $(c, H(m)^{sr})$, rerandomizes it using pk, and outputs the rerandomized ciphertext. The contained information (c, sr, pk) reveals no information on s because it is a ciphertext. It is not difficult to see that the obfuscation satisfies a VBP under the assumption that $\mathsf{KEM.Enc}$ is secure. In other words, the VBP simply reduces to the security of $\mathsf{KEM.Enc}$.

Approach 1′ (A special case of Approach 1). We describe a sufficient condition under which Enc is rerandomizable with pk. If $\mathsf{KEM.Enc}$ satisfies a *scalar homomorphic* property (and is rerandomizable with pk), then Enc is rerandomizable with pk. By the scalar homomorphic property, we mean that, given a KEM ciphertext c, we can compute (r', c') such that r' is a new random key and c' is a ciphertext of $rr' \bmod q$. We denote the operation by $(r', c') \leftarrow \mathsf{multiply}_{pk}(c)$. In this approach, a modified obfuscated program $C'_{c,sr,pk}$ computes $(c, H(m)^{sr})$, generates $(r', c') \leftarrow \mathsf{multiply}_{pk}(c)$, and outputs $(c', H(m)^{srr'})$, which is a rerandomization of $(c, H(m)^{sr})$. Bob can decrypt c' to recover rr' and then $H(m)^s$.

[2] We mean that anybody having the public key can convert a ciphertext of a message into a different ciphertext that is distributed identically to a fresh encryption of the same message.

We are done if $C'_{c,sr,pk}$ preserves the probabilistic functionality. However, we still have a potential problem. The distribution of c' may be different from the original distribution produced by $\mathsf{KEM.Enc}$. If $\mathsf{KEM.Enc}$ is *rerandomizable* with pk, we can fix it simply by rerandomizing c'. That is, a modified program $C''_{c,sr,pk}$ computes $(c', H(m)^{srr'})$, rerandomizes c' as c'', and outputs $(c'', H(m)^{srr'})$. For example, we can use the Paillier encryption scheme as $\mathsf{KEM.Enc}$ [31]. However, since its message (key) space is $\mathbb{Z}_n$ such that n is the product of two large primes, we need to define the bilinear group for the BLS scheme as having order n.

Approach 2. When Enc cannot be rerandomizable and we can not take Approaches 1 and $1'$, we can consider a new ES functionality. The new ES functionality is the sequential composition of the BLS signing algorithm and a new encryption algorithm Enc'. Enc' takes as input a plaintext $p(= H(m)^s) \in \mathbb{G}$ and a public key pk, runs $\mathsf{KEM.Enc}$ twice $((r_1, c_1) \leftarrow \mathsf{KEM.Enc}(pk), (r_2, c_2) \leftarrow \mathsf{KEM.Enc}(pk))$, and outputs $(c_1, c_2, p^{r_1 r_2})$. Clearly, the use of two random keys (r_1, r_2) is redundant, but we can obfuscate the naive program as follows: Given the naive program, we extract (s, pk), generate $(r_1, c_1) \leftarrow \mathsf{KEM.Enc}(pk)$, and compute $sr_1 \bmod q$. Then, we construct an obfuscated program $C_{c_1,sr_1,pk}$ containing (c_1, sr_1, pk), which takes as input m, generates $(r_2, c_2) \leftarrow \mathsf{KEM.Enc}(pk)$, and outputs $(c_1, c_2, H(m)^{sr_1 r_2})$. The contained information (c_1, sr_1, pk) is the same as in the previous approaches and reveals no information on s. Note that the $C_{c_1,sr_1,pk}$ does not preserve the *probabilistic* ES functionality since the value of c_1 is always the same. However, if $\mathsf{KEM.Enc}$ is rerandomizable with pk, then it is easy to fix the problem by rerandomizing c_1. In this approach, we can use any rerandomizable encryption scheme as $\mathsf{KEM.Enc}$.

Comparison. Let us briefly compare the above three approaches. Approaches 1 and 2 require that Enc and $\mathsf{KEM.Enc}$ are rerandomizable, respectively. Approach $1'$ is a special case of Approach 1 and requires that a scalar homomorphic property (and rerandomizability) of $\mathsf{KEM.Enc}$, which is a strong requirement. Note that we may be able to take Approach 1 without using the scalar homomorphic property required by Approach $1'$ (although we don't have a concrete example). Approach 2 requires a redundancy of ciphertexts. Therefore, Approach 1 seems to be the best approach.

1.2 Our Contributions

In this paper, we will use the pair of Waters's signature scheme [33] and the linear encryption scheme proposed by Boneh, Boyen, and Shacham [5] to take Approach $1'$ and propose a secure obfuscator for the ES functionality. Following the security definition and argument by Hohenberger et al. in [24], we present a security analysis of our proposed obfuscation. Waters's signature scheme is more complicated than the BLS scheme, but it is provably secure in the standard model. All security arguments in this paper are in the standard model.

Our contributions are summarized as follows: In Section 3, we propose two security definitions of digital signature schemes in the context of ES. One requires that no adversary can existentially forge a signature even if it is given black-box access to the ES functionality. The other requires the same even if it is given an *obfuscated* program for the ES functionality. We expect that the former/weaker definition implies the latter/stronger definition if the obfuscator satisfies a VBP. In Section 4, we propose a natural generalization of the VBP definition proposed by Hohenberger et al. in [24] so that we can show that the weaker existential unforgeability implies the stronger one. As stated in [24], their proposed VBP provides a meaningful security for cryptographic schemes if they satisfy a special property called *distinguishable attack property*. Unfortunately, digital signature schemes do not have this property in the context of ES. This is the reason why we need to introduce the generalized VBP definition. In Section 5, we propose a special ES functionality, which is the sequential composition of Waters's signature scheme and the linear encryption scheme, and construct an obfuscator for it. We prove that the obfuscator is secure under the generalized VBP definition and that Waters's signature scheme satisfies the stronger existential unforgeability with the obfuscator.

1.3 Related Works

Some related works are already mentioned in the previous sections. In particular, our work is inspired by the secure obfuscation for re-encryption in [24]. Both re-encryption and ES functionalities output a ciphertext and this common property enables us to simulate *real* obfuscated programs by randomly generating *junk* programs to prove the VBPs.

Our proposed obfuscation can be viewed as *public-key obfuscation* for signing functionalities [30,1]. A generic construction of public key obfuscations with a fully homomorphic encryption scheme is discussed in [18].

There are some different definitional approaches than VBPs to capture the unintelligibility of obfuscation, e.g., indistinguishability of obfuscation [3], best-possible obfuscation [21], and non-malleable obfuscation [12].

2 Preliminaries

Given a positive integer n, we denote by $[n]$ the set $\{1, 2, \cdots, n\}$. We say that a function $\nu(\cdot) : \mathbb{N} \to \mathbb{R}^+$ is negligible in n if for every polynomial $p(\cdot)$ and all sufficiently large n's, it holds that $\nu(n) < 1/p(n)$. Given a probability distribution S, we denote by $x \leftarrow S$ the operation of selecting an element according to S. If A is a probabilistic machine then $A(x_1, x_2, \ldots, x_k)$ denotes the output distribution of A on inputs $(x_1, x_2, \ldots, x_k)$. Let $\Pr[x \leftarrow S_1; x_2 \leftarrow S_2; \ldots; x_k \leftarrow S_k : E]$ denote the probability of the event E after the processes $x_1 \leftarrow S_1, x_2 \leftarrow S_2, \ldots, x_k \leftarrow S_k$ are performed in order. PPT stands for "probabilistic polynomial time". All PPT machines in this paper run in probabilistic polynomial-time in the security parameter denoted by n. Also, some PPT machines (e.g., representing adversaries) are allowed to take non-uniform auxiliary input of polynomial length in n, which is denoted by z.

2.1 Circuit Obfuscators

A class of circuits is of the form $\mathcal{C} = \{\mathcal{C}_n\}_{n \in \mathbb{N}}$, where $\mathcal{C}_n$ is a set of polynomial-size circuits with input length $l_{in}(n)$ and output length $l_{out}(n)$, where $l_{in}(n)$ and $l_{out}(n)$ are polynomials. It has an associated PPT generation algorithm which takes as input 1^n and generates a random circuit C from $\mathcal{C}_n$. In this paper, it corresponds to the random selection of information such as cryptographic keys on security parameter 1^n. We denote the generation process by $C \leftarrow \mathcal{C}_n$. When a circuit is used as an input or an output argument of an algorithm, we assume that an encoding of circuits is used implicitly (e.g., obfuscators take as input a circuit and output a circuit). The results of this paper are independent of any particular encoding method.

Let $C(x, r)$ be a *probabilistic* circuit which takes the regular input x and the random input r. Given a regular input x, we can view $C(x, \cdot)$ as a sampling algorithm for the distribution obtained by evaluating $C(x, r)$ on random coins r. Given two probabilistic circuits (C_1, C_2) whose regular inputs are of the same length, we denote by $(C_1(x), C_2(x))$ the two distributions produced by $C_1(x, \cdot)$ and $C_2(x, \cdot)$ and by $\mathsf{StaDiff}(C_1(x), C_2(x))$ the statistical difference between them, i.e., $\mathsf{StaDiff}(C_1(x), C_2(x)) = \frac{1}{2} \sum_{y \in \{0,1\}^{l_{out}(n)}} |\Pr[o \leftarrow C_1(x) : o = y] - \Pr[o \leftarrow C_2(x) : o = y]|$.

When we say that a machine M has black-box access to a probabilistic circuit C, we have two different meanings: *oracle access* and *sampling access*. Oracle access is such that M is allowed to set both regular and random inputs. We denote it by M^C. Sampling access is such that M is allowed to set only the regular input, but not the random input. That is, when M makes an oracle query x, M obtains a uniform and independent sample from the distribution produced by $C(x, \cdot)$. We denote it by $M^{\lll C \ggg}$.

An obfuscator for a class of circuits $\mathcal{C} = \{\mathcal{C}_n\}_{n \in \mathbb{N}}$ is a PPT machine which takes as input a circuit $C \in \mathcal{C}_n$ and outputs an unintelligible circuit C' which preserve the functionality. In this paper, we require that the functionality should be perfectly preserved.

Definition 1. *A PPT machine* Obf *is a* circuit obfuscator *for a class of probabilistic circuits* $\mathcal{C} = \{\mathcal{C}_n\}_{n \in \mathbb{N}}$ *if, for every probabilistic circuit* $C \in \mathcal{C}_n$*, the following holds:* $\Pr[C' \leftarrow \mathsf{Obf}(C) : \forall x, \mathsf{StaDiff}(C(x), C'(x)) = 0] = 1$.

Remark 1. We can relax the functionality requirement by allowing a negligible statistical difference and a negligible error probability as in [23,24]. In this paper, we use this stronger definition because our proposed obfuscator can satisfy it.

Definition 1 says nothing about the security requirement and we will formulate it based on VBPs in Section 4.

2.2 Public-Key Encryption and Digital Signatures

We review the security notions of public-key encryption (PKE) and digital signature (DS) schemes (Our definitions are based on [19]). Let Setup be an algorithm which, on security parameter 1^n, generates a parameter to be used commonly by multiple users in a pair of PKE and DS schemes.

A PKE scheme consists of three algorithms $(\mathsf{EKG}, \mathsf{E}, \mathsf{D})$. The key generation algorithm EKG is a probabilistic algorithm which takes as input a common parameter p and returns a public-secret key pair (pk, sk). The encryption algorithm E is a probabilistic algorithm which takes a common parameter p, a public key pk, and a plaintext $m \in MS(p, pk)$ to return a ciphertext c, where $MS(p, pk)$ is the message space defined by (p, pk). The encryption process is denoted by $c \leftarrow \mathsf{E}(p, pk, m)$. The decryption algorithm D is a deterministic algorithm which takes a common parameter p, a secret key sk, and a ciphertext c to return the plaintext m, and the decryption process is denoted by $m = \mathsf{D}(p, sk, c)$. When the given ciphertext is invalid, the decryption algorithm produces a special symbol $\perp$ to indicate that the ciphertext was invalid. It is required that, for every key information (p, pk, sk) and every message $m \in MS(p, pk)$, the decryption always succeeds, i.e., $\Pr[c \leftarrow \mathsf{E}(p, pk, m) : \mathsf{D}(p, sk, c) = m] = 1$. The following is the standard indistinguishability requirement against chosen plaintext attacks (CPAs). The definition is for a single message, but implies the indistinguishability requirement for polynomially multiple messages [19].

Definition 2 (Indistinguishability of Encryptions against CPAs). *A PKE scheme (*$\mathsf{EKG}, \mathsf{E}, \mathsf{D}$*) satisfies the indistinguishability if the following condition holds: For every PPT machine pair* (A_1, A_2) *(adversary), every polynomial* $p(\cdot)$*, all sufficiently large* $n \in \mathbb{N}$*, and every* $z \in \{0,1\}^{\mathsf{poly}(n)}$*,*

$$2 \cdot \Pr\left[\begin{array}{l} p \leftarrow \mathsf{Setup}(1^n); (pk, sk) \leftarrow \mathsf{EKG}(p); \\ (m_1, m_2, h) \leftarrow A_1(p, pk, z); b \leftarrow \{0,1\}; c \leftarrow \mathsf{E}(p, pk, m_b); \\ d \leftarrow A_2(p, pk, (m_1, m_2, h), c, z) : \\ b = d \end{array}\right] - 1 < \frac{1}{p(n)},$$

where we assume that A_1 produces a valid message pair m_1 and $m_2 \in MS(p, pk)$.

A DS scheme consists of three algorithms $(\mathsf{SKG}, \mathsf{S}, \mathsf{V})$. The key generation algorithm SKG is a probabilistic algorithm which takes as input a common parameter p and returns a public-secret key pair (pk, sk). The signing algorithm S is a probabilistic algorithm which takes a common parameter p, a secret key sk, and a plaintext $m \in MS(p, pk)$ to return a signature σ, where $MS(p, pk)$ is the message space defined by (p, pk). The signing process is denoted by $\sigma \leftarrow \mathsf{S}(p, sk, m)$. The verification algorithm V is a deterministic algorithm which takes a common parameter p, a public key pk, a message m, and a signature σ to return Accept if σ is a valid signature of m, and the verification process is denoted by $d = \mathsf{V}(p, pk, m, \sigma)$. It is required that, for every key information (p, pk, sk) and every message $m \in MS(p, pk)$, the verification of valid signatures always succeeds, i.e., $\Pr[\sigma \leftarrow \mathsf{S}(p, sk, m) : \mathsf{V}(p, pk, m, \sigma) = \mathsf{Accept}] = 1$. The following is the standard existential unforgeability (EU) requirement against chosen-message attacks (CMAs).

Definition 3 (Existential Unforgeability against CMAs). *A DS scheme (*$\mathsf{SKG}, \mathsf{S}, \mathsf{V}$*) is existentially unforgeable if the following condition holds: For every*

PPT oracle machine A (adversary), every polynomial $p(\cdot)$, all sufficiently large $n \in \mathbb{N}$, and every $z \in \{0,1\}^{\mathsf{poly}(n)}$,

$$\Pr\left[\begin{array}{l} p \leftarrow \mathsf{Setup}(1^n); (pk, sk) \leftarrow \mathsf{SKG}(p); \\ (m, \sigma, Q) \leftarrow A^{\ll \mathsf{S}_{p,sk} \gg}(p, pk, z) : \\ \mathsf{V}(p, pk, m, \sigma) = \mathsf{Accept} \text{ and } m \notin Q \end{array}\right] < \frac{1}{p(n)},$$

where $\mathsf{S}_{p,sk}$ is the signing oracle (circuit) and Q is the set of messages queried by A adaptively.

3 Security of Digital Signatures in the Context of ES

In this section, we re-define the EU requirement on DS schemes in the context of ES. Let $(\mathsf{EKG}, \mathsf{E}, \mathsf{D})$ and $(\mathsf{SKG}, \mathsf{S}, \mathsf{V})$ be a pair of PKE and DS schemes. We consider the ES functionality $F_{ES} = \{F_n\}_{n \in \mathbb{N}}$ for the two schemes. Given a common parameter p, a secret signing key sk, and a public encryption key pk_e generated with the security parameter 1^n, the ES functionality $F_{p,sk,pk_e} \in F_n$ is defined as follows:

1. When F_{p,sk,pk_e} is run on a message m, it generates a signature on m under sk ($\sigma \leftarrow \mathsf{S}(p, sk, m)$), encrypts σ under pk_e ($c \leftarrow \mathsf{E}(p, pk_e, \sigma)$), and outputs c.
2. When F_{p,sk,pk_e} is run on the special input keys, it outputs (p, pk, pk_e), where pk is the public verification key corresponding to sk.

Also, we define a corresponding class of circuits $\mathcal{C}_{ES} = \{\mathcal{C}_n\}_{n \in \mathbb{N}}$ which implements F_{ES}. $\mathcal{C}_n$ is a set of circuits C_{p,sk,pk_e} implementing F_{p,sk,pk_e}. The associated generation algorithm takes as input 1^n, generates $p \leftarrow \mathsf{Setup}(1^n)$, $(sk, pk) \leftarrow \mathsf{SKG}(p)$ and $(sk_e, pk_e) \leftarrow \mathsf{EKG}(p)$, and outputs C_{p,sk,pk_e}.

The following is the EU requirement re-defined in the context of ES. The difference from Definition 3 is that A is given the public encryption key pk_e. However, it is still equivalent to Definition 3.

Definition 4 (EU w.r.t. ES Functionality). *Let ($\mathsf{EKG}, \mathsf{E}, \mathsf{D}$) and ($\mathsf{SKG}, \mathsf{S}, \mathsf{V}$) be a pair of PKE and DS schemes. The DS scheme is existentially unforgeable w.r.t. the ES functionality if the following condition holds: For every PPT machine A (adversary), every polynomial $p(\cdot)$, all sufficiently large $n \in \mathbb{N}$, and every $z \in \{0,1\}^{\mathsf{poly}(n)}$,*

$$\Pr\left[\begin{array}{l} p \leftarrow \mathsf{Setup}(1^n); (pk, sk) \leftarrow \mathsf{SKG}(p); (pk_e, sk_e) \leftarrow \mathsf{EKG}(p); \\ (m, \sigma, Q) \leftarrow A^{\ll \mathsf{S}_{p,sk} \gg}(p, pk, pk_e, z) : \\ \mathsf{V}(pk, m, \sigma) = \mathsf{Accept} \text{ and } m \notin Q \end{array}\right] < \frac{1}{p(n)}.$$

Note that the adversary A implicitly has sampling access to F_{p,sk,pk_e} because it has sampling access to the signing oracle $\mathsf{S}_{p,sk}$ and takes as input the public encryption key (p, pk_e). In this sense, Definition 4 requires that the signature scheme is still existentially unforgeable *even when A is given sampling access to F_{p,sk,pk_e}*.

Next, we consider a stronger EU, which requires that the signature scheme is still existentially unforgeable *even when A is given an obfuscated circuit for* F_{p,sk,pk_e}. The following is the strengthened definition and the difference from Definition 4 is that A is given an obfuscated circuit for F_{p,sk,pk_e}.

Definition 5 (EU w.r.t. ES Obfuscator). *Let (*$\mathsf{EKG}, \mathsf{E}, \mathsf{D}$*) and (*$\mathsf{SKG}, \mathsf{S}, \mathsf{V}$*) be a pair of PKE and DS schemes. Also, let* Obf *be a circuit obfuscator for* $\mathcal{C}_{ES}$*. The DS scheme is existentially unforgeable w.r.t.* Obf *if the following condition holds: For every PPT machine A (adversary), every polynomial $p(\cdot)$, all sufficiently large $n \in \mathbb{N}$, and every $z \in \{0,1\}^{\mathsf{poly}(n)}$,*

$$\Pr\left[\begin{array}{l} p \leftarrow \mathsf{Setup}(1^n); (pk, sk) \leftarrow \mathsf{SKG}(p); (pk_e, sk_e) \leftarrow \mathsf{EKG}(p); \\ C' \leftarrow \mathsf{Obf}(C_{p,sk,pk_e}); \\ (m, \sigma, Q) \leftarrow A^{\ll \mathsf{S}_{p,sk} \gg}(p, pk, pk_e, C', z) : \\ \mathsf{V}(pk, m, \sigma) = \mathsf{Accept} \text{ and } m \notin Q \end{array}\right] < \frac{1}{p(n)}.$$

We expect that if Obf satisfies a strong VBP, then the EU w.r.t. the ES functionality implies the (stronger) EU w.r.t. Obf. The question is what VBP Obf should satisfy for the implication to hold. We will answer it in the next section.

Remark 2. We can re-define the indistinguishability of encryptions in the context of ES in a similar way. However, we omit it because the main purpose of obfuscators is to hide Alice's secret signing key but not Bob's secret decryption key. Note that the indistinguishability is preserved even when the distinguisher D in Definition 2 is given the naive program for ES that reveals the signing key.

4 Virtual Black-Box Properties

In this section, we review average-case VBP (ACVBP) proposed by Hohenberger et al. in [24], under which the VBP of the re-encryption obfuscator is proved. As stated in [24], their ACVBP provides a meaningful security for cryptographic schemes if they satisfy a special property called *distinguishable attack property.* Unfortunately, DS schemes do not have this property in the context of ES. That is, even if Obf satisfies the ACVBP under their ACVBP definition, the EU w.r.t. the ES functionality does not necessarily imply the stronger EU w.r.t. Obf. Therefore, we propose a natural generalization of their ACVBP definition under which we can claim that, if Obf satisfies the (stronger) ACVBP, then the implication holds.

First of all, we review the definition of ACVBP proposed in [24].

Definition 6 (ACVBP [24]). *A circuit obfuscator* Obf *for $\mathcal{C}$ satisfies the* ACVBP *if the following condition holds: There exists a PPT oracle machine S (simulator) such that, for every PPT oracle machine D (distinguisher), every polynomial $p(\cdot)$, all sufficiently large $n \in \mathbb{N}$, and every $z \in \{0,1\}^{\mathsf{poly}(n)}$,*

$$\left|\Pr\left[\begin{array}{l} C \leftarrow \mathcal{C}_n; \\ C' \leftarrow \mathsf{Obf}(C); \\ b \leftarrow D^{\ll C \gg}(C', z) \end{array} : b = 1\right] - \Pr\left[\begin{array}{l} C \leftarrow \mathcal{C}_n; \\ C'' \leftarrow S^{\ll C \gg}(1^n, z); \\ b \leftarrow D^{\ll C \gg}(C'', z) \end{array} : b = 1\right]\right| < \frac{1}{p(n)}.$$

It was proposed as a general definition in the sense that it is not specific to re-encryption. The authors gave an informal discussion that their proposed ACVBP provides a meaningful security in cryptographic settings [24, Section 2.1]. We briefly review it below. In general, VBPs should guarantee that *if a cryptographic scheme is secure when the adversary is given black-box access to a program, then it remains secure when the adversary is given the obfuscated program.* The authors claim that for a large class of applications (including re-encryption), obfuscators satisfying Definition 6 indeed give this guarantee. More specifically, the authors propose to use the following informal argument: **If** a cryptographic scheme has the following three properties:

1. The scheme is secure against black-box adversaries with sampling access to functionality X selected randomly from a family F;
2. A distinguisher D with sampling access to X can test whether an adversary A can break the security guarantee of the scheme (*distinguishable attack property*);
3. There exists a circuit obfuscator satisfying ACVBP for a class C_F of circuits implementing F;

Then the cryptographic scheme is also secure against adversaries who are given an obfuscation of a circuit selected at random from the class C_F.

The argument works for re-encryption functionalities as discussed in [24], where F is a re-encryption functionality and the cryptographic scheme is the underlying encryption scheme. However, it does NOT work for ES functionalities, where the cryptographic scheme is a pair of PKE and DS schemes, F is the ES functionality F_{ES}, and X is F_{p,sk,pk_e}. Let us check whether the argument goes through for the DS scheme. We have no problem with the first and third conditions. The first condition requires that the DS scheme satisfies the standard EU requirement according to Definition 4. The third condition requires that there exists a circuit obfuscator satisfying ACVBP for $\mathcal{C}_{ES}$. The problem is that the second condition is not satisfied in this case. The reason is that A has sampling access to the signing oracle, but D does not have.

Remark 3. Readers might ask if Definition 6 still provides a meaningful security for cryptographic schemes even when they do NOT satisfy the *distinguishable attack property*. However, it is not the case. We can show that there exists a cryptographic functionality using a secret information such that (1) the secret operation does not satisfy the distinguishable attack property and (2) it has an obfuscator satisfying the ACVBP under Definition 6, but any obfuscated circuit reveals the secret information.

As discussed above, Definition 6 is not strong enough for our purpose. In order to make it stronger, we propose a natural generalization. The generalization allows distinguishers to have sampling access not only to $\ll C \gg$ but also to a set of oracles dependent on C.

Definition 7 (ACVBP w.r.t. Dependent Oracles). *Let $T(C)$ be a set of oracles dependent on the circuit C. A circuit obfuscator* Obf *for $\mathcal{C}$ satisfies the*

ACVBP w.r.t. dependent oracle set T *if the following condition holds: There exists a PPT oracle machine S (simulator) such that, for every PPT oracle machine D (distinguisher), every polynomial $p(\cdot)$, all sufficiently large $n \in \mathbb{N}$, and every $z \in \{0,1\}^{\mathsf{poly}(n)}$,*

$$\left| \Pr\left[\begin{array}{ll} C \leftarrow \mathcal{C}_n; & \\ C' \leftarrow \mathsf{Obf}(C); & : b = 1 \\ b \leftarrow D^{\ll C, T(C) \gg}(C', z) & \end{array}\right] - \Pr\left[\begin{array}{ll} C \leftarrow \mathcal{C}_n; & \\ C'' \leftarrow S^{\ll C \gg}(1^n, z); & : b = 1 \\ b \leftarrow D^{\ll C, T(C) \gg}(C'', z) & \end{array}\right] \right| < \frac{1}{p(n)},$$

where $D^{\ll C, T(C) \gg}$ means that D has sampling access to all oracles contained in $T(C)$ in addition to C.

Clearly, for every T, this generalized ACVBP implies the ACVBP in Definition 6.

Remark 4. Since $T(C)$ can be viewed as *dependent* auxiliary-input to adversaries, it is natural to allow the simulator S to have access to $T(C)$. However, we did not allow it because the security proof of our obfuscator does not need it.

Now we can clarify the condition on Obf under which the EU w.r.t. the ES functionality implies the EU w.r.t. Obf.

Theorem 1. *Let $T(C_{p,sk,pk_e})$ be $\{\mathsf{S}_{p,sk}\}$. If an obfuscator Obf for $\mathcal{C}_{ES}$ satisfies ACVBP w.r.t. dependent oracle set T, then the EU w.r.t the ES functionality implies the EU w.r.t. Obf.*

Proof. We show that, if the EU w.r.t. the ES functionality is satisfied, but the stronger EU w.r.t. Obf is NOT satisfied, then it contradicts the ACVBP w.r.t. dependent oracle set T. Let A be the adversary that breaks the stronger EU. Consider the following distinguisher D that uses sampling access to $T(C_{p,sk,pk_e})$ to check whether the adversary A succeeds in breaking the stronger EU.

1. Take as input a circuit C and an auxiliary-input z. (C is either an obfuscated circuit or a simulated circuit).
2. Use the sampling access to C_{p,sk,pk_e} to get (p, pk, pk_e).
3. Use the sampling access to $\mathsf{S}_{p,sk}$ to simulate $(m, \sigma, Q) \leftarrow A^{\ll \mathsf{S}_{p,sk} \gg}(p, pk, pk_e, C, z)$.
4. Output 1 if and only if $\mathsf{V}(pk, m, \sigma) = \mathsf{Accept}$ and $m \notin Q$.

If C is an obfuscated circuit, then the probability D outputs 1 is equal to the probability that A breaks the stronger EU, which is not negligible by the assumption. On the other hand, if C is a simulated circuit, then the probability D outputs 1 is negligible, otherwise, A can be used to break the standard EU w.r.t. the ES functionality. Therefore, it contradicts the ACVBP w.r.t. dependent oracle set T. □

Remark 5. Note that the proof argument does not work under the ACVBP definition (Definition 6), where distinguishers are not allowed to use dependent oracle set T.

5 Secure Obfuscator for a Special ES Functionality

In this section, we propose an obfuscator for a special ES functionality and prove the security based on the generalized ACVBP definition. Our proposed ES functionality is the sequential composition of Waters's signature scheme and the linear encryption scheme.

5.1 Algebraic Setting and Complexity Assumptions

First of all, we review the required algebraic setting and complexity assumptions. Let Setup be an algorithm which, on input the security parameter 1^n, randomly generates the parameters for a bilinear map $(q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$, where q is a prime of length n, both $\mathbb{G}$ and $\mathbb{G}_T$ are groups of order q, $\mathbf{e}$ is an efficient bilinear mapping from $\mathbb{G} \times \mathbb{G}$ to $\mathbb{G}_T$, g is a generator of $\mathbb{G}$ (e.g., refer to [6, Section 5]). The mapping $\mathbf{e}$ satisfies the following two properties: (i) Bilinear: for all $g \in \mathbb{G}$ and $a, b \in \mathbb{Z}_q$, $\mathbf{e}(g^a, g^b) = \mathbf{e}(g, g)^{ab}$. (ii) Non-degenerate: if g generates $\mathbb{G}$, then $\mathbf{e}(g^a, g^b) \neq 1$.

In this paper, we use the following two Diffie-Hellman assumptions. All assumptions are standard ones which have been used in the literature. The first one is so-called the Decisional Bilinear Diffie-Hellman (DBDH) assumption (e.g., see [25,33]), which assumes that, given $g, g^a, g^b, g^c, \mathbf{e}(g, g)^d$, it is hard to check whether $abc = d$. The second one is the Decisional Linear (DL) assumption (e.g., see [5,24]), which assumes that, given $g, g^a, g^b, g^t, (g^a)^r, (g^b)^s$, it is hard to check whether $r + s = t$.

Definition 8 (DBDH Assumption). *For every PPT machine D, every polynomial $p(\cdot)$, all sufficiently large $n \in \mathbb{N}$, and every $z \in \{0,1\}^{\mathsf{poly}(n)}$,*

$$\left| \Pr\begin{bmatrix} p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g) \leftarrow \mathsf{Setup}(1^n); & \\ a \leftarrow \mathbb{Z}_q; b \leftarrow \mathbb{Z}_q; c \leftarrow \mathbb{Z}_q; & : decision = 1 \\ decision \leftarrow D(p, g^a, g^b, g^c, \mathbf{e}(g,g)^{abc}, z) & \end{bmatrix} - \right.$$
$$\left. \Pr\begin{bmatrix} p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, h) \leftarrow \mathsf{Setup}(1^n); & \\ a \leftarrow \mathbb{Z}_q; b \leftarrow \mathbb{Z}_q; c \leftarrow \mathbb{Z}_q; d \leftarrow \mathbb{Z}_q; & : decision = 1 \\ decision \leftarrow D(p, g^a, g^b, g^c, \mathbf{e}(g,g)^{d}, z) & \end{bmatrix} \right| < \frac{1}{p(n)}.$$

Definition 9 (DL Assumption). *For every PPT machine D, every polynomial $p(\cdot)$, all sufficiently large $n \in \mathbb{N}$, and every $z \in \{0,1\}^{\mathsf{poly}(n)}$,*

$$\left| \Pr\begin{bmatrix} p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g) \leftarrow \mathsf{Setup}(1^n); & \\ a \leftarrow \mathbb{Z}_q; b \leftarrow \mathbb{Z}_q; r \leftarrow \mathbb{Z}_q; s \leftarrow \mathbb{Z}_q; & : decision = 1 \\ decision \leftarrow D(p, (g^a, g^b), (g^{r+s}, (g^a)^r, (g^b)^s), z) & \end{bmatrix} - \right.$$
$$\left. \Pr\begin{bmatrix} p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g) \leftarrow \mathsf{Setup}(1^n); & \\ a \leftarrow \mathbb{Z}_q; b \leftarrow \mathbb{Z}_q; r \leftarrow \mathbb{Z}_q; s \leftarrow \mathbb{Z}_q; t \leftarrow \mathbb{Z}_q; & : decision = 1 \\ decision \leftarrow D(p, (g^a, g^b), (g^t, (g^a)^r, (g^b)^s), z) & \end{bmatrix} \right| < \frac{1}{p(n)}.$$

5.2 Waters's Signature Scheme

We recall Waters's signature scheme [33]. The message space is $\{0,1\}^n$.

$\mathsf{SKG}(p)$:

1. Parse $p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$.
2. Randomly select $\alpha \leftarrow \mathbb{Z}_q$ and compute $g_1 = g^{\alpha}$.
3. Randomly select $g_2 \leftarrow \mathbb{G}$ and $u' \leftarrow \mathbb{G}$.
4. Randomly select $u_i \leftarrow \mathbb{G}$ for every $i \in [n]$ and set $U = \{u_i\}_{i \in [n]}$.
5. Output $pk = (g_1, g_2, u', U)$ and $sk = (g_2^{\alpha}, u', U)$ as public and secret keys, respectively.

$\mathsf{Sign}(p, sk, m)$:

1. Parse $p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$, $sk = (g_2^{\alpha}, u', \{u_i\}_{i \in [n]})$, and $m = (m_1, m_2, \cdots, m_n)$, where m_i denotes the i'th bit of m.
2. Randomly select $x \leftarrow \mathbb{Z}_q$.
3. Compute $(\sigma_1, \sigma_2) = (g_2^{\alpha}(u' \prod_{i \in \mathcal{M}} u_i)^x, g^x)$, where $\mathcal{M}$ is the set of all i such that $m_i = 1$.
4. Output $\sigma = (\sigma_1, \sigma_2)$.

$\mathsf{Verify}(p, pk, m, \sigma)$:

1. Parse $p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$, $pk = (g_1, g_2, u', \{u_i\}_{i \in [n]})$, $m = (m_1, m_2, \cdots, m_n)$, and $\sigma = (\sigma_1, \sigma_2)$.
2. Output Accept if $\mathbf{e}(\sigma_1, g)/\mathbf{e}(\sigma_2, u' \prod_{i \in \mathcal{M}} u_i) = \mathbf{e}(g_1, g_2)$. Output Reject otherwise.

The security is proved under the DBDH assumption.

Theorem 2 ([33]). *Under the DBDH assumption, Waters's signature scheme is existentially unforgeable.*

5.3 Linear Encryption Scheme

We recall the linear encryption scheme [5]. The message space is $\mathbb{G}$.

$\mathsf{EKG}(p)$:

1. Parse $p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$.
2. Randomly select $a \leftarrow \mathbb{Z}_q$ and $b \leftarrow \mathbb{Z}_q$.
3. Output $pk_e = (g^a, g^b)$ and $sk_e = (a, b)$ as public and secret keys, respectively.

$\mathsf{Enc}(p, pk_e, m)$:

1. Parse $p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$ and $pk_e = (g^a, g^b)$.
2. Randomly select $r \leftarrow \mathbb{Z}_q$ and $s \leftarrow \mathbb{Z}_q$.
3. Compute $(c_1, c_2, c_3) = ((g^a)^r, (g^b)^s, g^{r+s}m)$.
4. Output $c = (c_1, c_2, c_3)$.

$\mathsf{Dec}(p, sk_e, c)$:

1. Parse $p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$, $sk_e = (a, b)$, and $c = (c_1, c_2, c_3)$.
2. Output $m = c_3/(c_1^{1/a} \cdot c_2^{1/b})$.

The security is proved under the DL assumption.

Theorem 3 ([5]). *Under the DL assumption, the linear encryption scheme satisfies the indistinguishability.*

We can view g^{r+s} as a random key generated by a KEM and (c_1, c_2) as its ciphertext. Note that the KEM encryption algorithm has the scalar homomorphic property described in Section 1.1 and Enc is rerandomizable. Specifically, given a ciphertext $c = (c_1, c_2, c_3)$ and the public key $pk_e = (g^a, g^b)$, we can rerandomize the ciphertext by computing $(c_1(g^a)^{r'}, c_2(g^b)^{s'}, c_3 g^{r'+s'})$, where r' and s' are random numbers in $\mathbb{Z}_q$. We denote by $\mathsf{ReRand}(p, pk_e, (c_1, c_2, c_3))$ the rerandomization algorithm.

5.4 The Obfuscator for the ES Functionality

Our special ES functionality is the sequential composition of Waters's signature scheme and the linear encryption scheme. Given a common parameter p, a secret signing key sk, and a public encryption key pk_e, the ES functionality F_{p,sk,pk_e} provides the following two functions:

- $\mathsf{ES}_{p,sk,pk_e}(m)$:
  1. Run $(\sigma_1, \sigma_2) \leftarrow \mathsf{Sign}(p, sk, m)$.
  2. Run $C_1 \leftarrow \mathsf{Enc}(p, pk_e, \sigma_1)$.
  3. Run $C_2 \leftarrow \mathsf{Enc}(p, pk_e, \sigma_2)$.
  4. Output (C_1, C_2).
- $\mathsf{Keys}_{p,sk,pk_e}(\mathsf{keys})$:
  1. Output (p, pk, pk_e), where pk is the public key corresponding to sk.

We define a (naive) class of circuits $\mathcal{C}_{ES} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ for the ES functionality, which we want to obfuscate. $\mathcal{C}_n$ is a set of circuits C_{p,sk,pk_e} and each C_{p,sk,pk_e} is a naive implementation of F_{p,sk,pk_e}. Without loss of generality, we assume that we can extract (p, sk, pk_e) from C_{p,sk,pk_e}. The associated generation algorithm takes as input 1^n, generates a common parameter $p \leftarrow \mathsf{Setup}(1^n)$, runs $(pk, sk) \leftarrow \mathsf{SKG}(p)$, runs $(pk_e, sk_e) \leftarrow \mathsf{EKG}(p)$, and outputs C_{p,sk,pk_e}.

Now, we describe our proposed obfuscator Obf_{ES} for $\mathcal{C}_{ES}$ below. According to the basic idea in Section 1.1, the obfuscation is done by encrypting the signing key g_2^α and the obfuscated circuit generates a signature using the encrypted signing key.

Given a circuit C_{p,sk,pk_e}, the obfuscator Obf_{ES}

1. Extracts (p, sk, pk_e).
2. Gets pk using the Keys function.
3. Parses $p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$ and $sk = (g_2^\alpha, u', U)$.
4. Runs $(c_1, c_2, c_3) \leftarrow \mathsf{Enc}(p, pk_e, g_2^\alpha)$ to encrypt g_2^α. (NOTE: We have (c_1, c_2, c_3) $= ((g^a)^r, (g^b)^s, g^{r+s} g_2^\alpha))$.
5. Sets $sk' = (c_3, u', U)$, which is an encrypted form of the signing key sk.
6. Constructs and outputs an obfuscated circuit that contains the values (p, pk_e, pk, sk', (c_1, c_2)) and does the following: (1) On input keys, outputs (p, pk, pk_e) and (2) On input a message $m \in \{0,1\}^n$,
   (a) Runs $(\sigma_1, \sigma_2) \leftarrow \mathsf{Sign}(p, sk', m)$. Note that (c_1, c_2, σ_1) is an encryption of the first part of a valid signature by Enc. (NOTE: we have $(c_1, c_2, \sigma_1) =$ $((g^a)^r, (g^b)^s, g^{r+s} g_2^\alpha (u' \prod_{i \in \mathcal{M}} u_i)^x))$.
   (b) Computes $C_1 = (c'_1, c'_2, c'_3) \leftarrow \mathsf{ReRand}(p, pk_e, (c_1, c_2, \sigma_1))$.
   (c) Runs $C_2 \leftarrow \mathsf{Enc}(p, pk_e, \sigma_2)$.
   (d) Outputs (C_1, C_2).

Remark 6. For simplicity, we used Enc to encrypt σ_2 in the definition of F_{p,sk,pk_e}. However, we can use an arbitrary encryption algorithm instead of Enc and it is easy to modify the obfuscator Obf_{ES}. Furthermore, we may want to omit the encryption of σ_2 since it is just a random number and leaks no meaningful information (as long as σ_1 is encrypted). In this case, we have $C_2 = \sigma_2$ in the both definitions of F_{p,sk,pk_e} and Obf_{ES}.

Clearly, it satisfies the functionality requirement according to Definition 1. We prove that it satisfies ACVBP even though distinguishers are given sampling access to the signing oracle according to Definition 7.

Theorem 4. *Let $T(C_{p,sk,pk_e})$ be $\{\mathsf{S}_{p,sk}\}$. Under the DL assumption,* Obf_{ES} *satisfies ACVBP w.r.t. dependent oracle set T.*

Proof. Since we can identify an obfuscated ES circuit with the values (p, pk_e, pk, sk', (c_1, c_2)) contained in the circuit, it is sufficient to construct a simulator which simulates the values by the help of sampling access to the original circuit C_{p,sk,pk_e}. The first three values (p, pk_e, pk) can be obtained from the sampling access to C_{p,sk,pk_e} using the $\mathsf{Keys}_{p,sk,pk_e}$ function and so the question is how to simulate the last three values $(sk', (c_1, c_2))$. We show that it is sufficient to generate *junk* values for them because it is essentially an encryption of the signing key g_2^α.

Consider the following simulator S having sampling access to C_{p,sk,pk_e}.

1. Take as input the security parameter 1^n and an auxiliary-input z.
2. Use the sampling access to C_{p,sk,pk_e} to get (p, pk, pk_e).
3. Parse $p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$ and $pk = (g_1, g_2, u', U)$.
4. Randomly select $Junk \leftarrow \mathbb{G}$.

5. Run $(c_1, c_2, c_3) \leftarrow \mathsf{Enc}(p, pk_e, Junk)$.
6. Set $sk' = (c_3, u', U)$.
7. Output $(p, pk_e, pk, sk', (c_1, c_2))$.

We need to show that the output distribution of S is indistinguishable from the real distribution of $(p, pk_e, pk, sk', (c_1, c_2))$ for any PPT distinguisher *even when it is allowed to have sampling access to* $CS = \{C_{p,sk,pk_e}, \mathsf{S}_{p,sk}\}$. For contradiction, assume that the probability that a distinguisher $D^{\ll CS \gg}$ can distinguish between them is not negligible. That is, the difference between the following two probabilities is not negligible. They are the probabilities that D outputs 1 given the real and simulated distributions, respectively. g_2^α is encrypted in the real distribution while $Junk$ is encrypted in the simulated distribution. It is the only difference.

$$\Pr\left[\begin{array}{l} p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g) \leftarrow \mathsf{Setup}(1^n); \\ (pk_e, sk_e) \leftarrow \mathsf{EKG}(p); \\ (pk, sk) = ((g_1, g_2, u', U), (g_2^\alpha, u', U)) \leftarrow \mathsf{SKG}(p); \\ (c_1, c_2, c_3) \leftarrow \mathsf{Enc}(p, pk_e, g_2^\alpha); \\ sk' = (c_3, u', U); \\ b \leftarrow D^{\ll CS \gg}((p, pk_e, pk, sk', (c_1, c_2)), z) : \\ b = 1 \end{array}\right]$$

$$\Pr\left[\begin{array}{l} p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g) \leftarrow \mathsf{Setup}(1^n); \\ (pk_e, sk_e) \leftarrow \mathsf{EKG}(p); \\ (pk, sk) = ((g_1, g_2, u', U), (g_2^\alpha, u', U)) \leftarrow \mathsf{SKG}(p); \\ Junk \leftarrow \mathbb{G}; \\ (c_1, c_2, c_3) \leftarrow \mathsf{Enc}(p, pk_e, Junk); \\ sk' = (c_3, u', U); \\ b \leftarrow D^{\ll CS \gg}((p, pk_e, pk, sk', (c_1, c_2)), z) : \\ b = 1 \end{array}\right]$$

Then we can construct an adversary pair (A_1, A_2) which breaks the indistinguishability of the linear encryption scheme. A_1 produces a message pair (m_1, m_2) and an associated hint h as follows:

1. Take as input a common parameter p, a public key pk_e, and auxiliary input z.
2. Parse $p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$.
3. Randomly generate $(pk, sk) = ((g_1, g_2, u', U), (g_2^\alpha, u', U)) \leftarrow \mathsf{SKG}(p)$.
4. Randomly generate $Junk \leftarrow \mathbb{G}$.
5. Set $m_1 = g_2^\alpha$, $m_2 = Junk$, and $h = pk$.
6. Output (m_1, m_2, h).

Given a ciphertext c (of either m_1 or m_2), A_2 can use the distinguisher D to distinguish between m_1 and m_2 as follows:

1. Take as input a common parameter p, a public key pk_e, A_1's output (m_1, m_2, h), a ciphertext c, and auxiliary input z.
2. Parse $p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$, $h = pk = (g_1, g_2, u', U)$, and $c = (c_1, c_2, c_3)$.

3. Compute $sk' = (c_3, u', U = \{u_i\}_{i \in [n]})$.
4. Simulate $D^{\ll CS \gg}((p, pk_e, pk, sk', (c_1, c_2)), z)$, where the oracle queries can be perfectly simulated using p, $sk = (m_1, u', U)$, and pk_e.
5. Output the output of D.

If the target ciphertext c is a ciphertext of m_1, the probability that A_2 outputs 1 is equal to the former probability, otherwise, it is equal to the latter probability. Since the difference is not negligible, it contradicts Theorem 3. □

As a corollary, we can conclude that Waters's signature scheme is existentially unforgeable even when adversaries are given an obfuscated circuit for F_{ES}. It immediately follows from Theorems 1, 2, and 4.

Corollary 1. *Under the DL and DBDH assumptions, Waters's signature scheme is existentially unforgeable w.r.t.* Obf_{ES}.

6 Concluding Remarks

In this paper, we have constructed an obfuscator for a special ES functionality and presented a security analysis. We can generalize our construction to clarify the properties that a pair of PKE and DS schemes should satisfy so that the ES functionality can be securely obfuscated. We omit it due to the space limitation. Here, we list several DS schemes satisfying all the required properties: Lysyanskaya's unique signature scheme [28], Dodis's verifiable random function (signature scheme) [15], the undeniable signature scheme by Chaum and Antwerpen [13], the DDH-based pseudo-random function (MAC) proposed by Naor and Reingold [29], and Schnorr's signature scheme [32].

We have proposed generic approaches to obfuscating ES functionalities (Approaches 1,1′, and 2). We took Approach 1′ using the scalar homomorphic property of the linear encryption scheme. If we take Approaches 1 and 2 where the only requirement on Enc and $\mathsf{KEM.Enc}$ is rerandomizability, then we can use an encryption scheme with a relaxed version of chosen-ciphertext attack (CCA) security [10]. It is an interesting research issue to investigate what kind of CCA security we can achieve in the context of ES obfuscation.

Finally, we believe that our proposed obfuscation can be used to securely obfuscate a signcryption scheme as described in Appendix A. The formal security argument is a future work item.

Acknowledgments. I would like to thank valuable comments from anonymous reviewers of TCC'2009, ACM CCS'2009, and Eurocrypt'2010. In particular, I would like to thank an Eurocrypt reviewer for suggesting the use of the Paillier encryption scheme as mentioned in Section 1.1, another Eurocrypt reviewer for suggesting the ES functionality and obfuscator presented in Section 5.4, which are much simpler than my original ones, and Henri Gilbert for helping me to revise the submitted version.

References

1. Adida, B., Wikstrom, D.: How to shuffle in public. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 555–574. Springer, Heidelberg (2007)
2. An, J.H., Dodis, Y., Rabin, T.: On the Security of Joint Signature and Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 83–107. Springer, Heidelberg (2002)
3. Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S., Yang, K.: On the (Im)possibility of Obfuscating Programs. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 1–18. Springer, Heidelberg (2001); ECCC, Report No. 57
4. Bellare, M., Rogaway, P.: Random Oracles are Practical: a paradigm for designing efficient protocols. In: Proceedings of the 1st ACM Conference on Computer and Communications Security, pp. 62–73 (1993)
5. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
6. Boneh, D., Franklin, M.: Identity based encryption from the Weil pairing. SIAM J. of Computing 32(3), 586–615 (2003)
7. 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)
8. Canetti, R.: Towards Realizing Random Oracles: Hash Functions that Hide All Partial Information. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 455–469. Springer, Heidelberg (2007)
9. Canetti, R., Dakdouk, R.R.: Obfuscating Point Functions with Multibit Output. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 489–508. Springer, Heidelberg (2008)
10. Canetti, R., Krawczyk, H., Nielsen, J.B.: Relaxing Chosen-Ciphertext Security. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 565–582. Springer, Heidelberg (2003)
11. Canetti, R., Micciancio, D., Reingold, O.: Perfectly One-way Probabilistic Hash Functions. In: Proceedings of 30th STOC (1998)
12. Canetti, R., Varia, M.: Non-malleable Obfuscation. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 73–90. Springer, Heidelberg (2009)
13. Chaum, D., van Antwerpen, H.: Undeniable Signatures. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 212–216. Springer, Heidelberg (1990)
14. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing 33(11), 167–226 (2003)
15. Dodis, Y.: Efficient Construction of (Distributed) Verifiable Random Functions. In: Naccache, D., Paillier, P. (eds.) PKC 2002. LNCS, vol. 2274, pp. 1–17. Springer, Heidelberg (2002)
16. Dodis, Y., Smith, A.: Correcting Errors without Leaking Partial Information. In: Proceedings of 37th STOC (2005)
17. ElGamal, T.: A public key cryptosystem and signature scheme based on discrete logarithms. IEEE Trans. Inform. Theory 31, 469–472 (1985)
18. Gentry, C.: A fully homomorphic encryption scheme, PhD Thesis (2009)
19. Goldreich, O.: Foundations of Cryptography. Basic Applications, vol. II. Cambridge University Press, Cambridge (2004)
20. Goldwasser, S., Kalai, Y.T.: On the Impossibility of Obfuscation with Auxiliary Input. In: Proceedings of FOCS 2005, pp. 553–562 (2005)
21. Goldwasser, S., Rothblum, G.N.: On Best-Possible Obfuscation. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 194–213. Springer, Heidelberg (2007)

22. Hada, S.: Zero-Knowledge and Code Obfuscation. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 443–457. Springer, Heidelberg (2000)
23. Hofheinz, D., Malone-Lee, J., Stam, M.: Obfuscation for Cryptographic Purposes. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 214–232. Springer, Heidelberg (2007)
24. Hohenberger, S., Rothblum, G.N., Shelat, A., Vaikuntanathan, V.: Securely Obfuscating Re-Encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 233–252. Springer, Heidelberg (2007)
25. Joux, A.: The Weil and Tate pairings as building blocks for public key cryptosystems. In: Fieker, C., Kohel, D.R. (eds.) ANTS 2002. LNCS, vol. 2369, pp. 20–32. Springer, Heidelberg (2002)
26. Joux, A., Nguyen, K.: Separating Decision Diffie-Hellman from Diffie-Hellman in Cryptographic Groups. Journal of Cryptology 16(4), 239–247 (2003)
27. Lynn, B., Prabhakaran, M., Sahai, A.: Positive Results and Techniques for Obfuscation. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 20–39. Springer, Heidelberg (2004)
28. Lysyanskaya, A.: Unique Signatures and Verifiable Random Functions from the DH-DDH Separation. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, p. 597. Springer, Heidelberg (2002)
29. Naor, M., Reingold, O.: Number-Theoretic Constructions of Efficient Pseudo-Random Functions. Journal of the ACM 51(2), 231–262 (2004)
30. Ostrovsky, R., Skeith III, W.E.: Private searching on streaming data. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 223–240. Springer, Heidelberg (2005)
31. 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)
32. Schnorr, C.P.: Efficient Signature Generation by Smart Cards. Journal of Cryptology 4(3), 161–174 (1991)
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. Wee, H.: On Obfuscating Point Functions. In: Proceedings of STOC 2005, pp. 523–532 (2005)
35. Zheng, Y.: Digital Signcryption or How to Achieve Cost (Signature & Encryption) << Cost (Signature) + Cost (Encryption). In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 165–179. Springer, Heidelberg (1997)

Appendix

A A Relation to Signcryption

In this appendix, we informally describe (1) how to use an ES functionality as a building block to construct a secure signcryption scheme and (2) how to obfuscate the resulting signcryption scheme using an obfuscator for the ES functionality.

A.1 EncryptedSignature-then-Encryption

We propose a new composition method which we call EncryptedSignature-then-Encryption (EStE) as a new approach to constructing a secure signcryption scheme: To signcrypt a message, we generate an encrypted signature and encrypt both the message and encrypted signature. More specifically, given a message m, we compute $\sigma \leftarrow \mathsf{S}(p, sk, m)$, $c_1 \leftarrow \mathsf{E}_1(p, pk, \sigma)$, and $c_2 \leftarrow \mathsf{E}_2(p, pk, (m, c_1))$, where the sequential composition of S and E_1 is the ES functionality, c_1 is the encrypted signature, and c_2 is the resulting signcryption of m. The difference from the standard StE composition is that the signature σ is doubly encrypted. The first encryption E_1 is by the ES functionality and the second encryption E_2 can be done by a standard hybrid encryption as in StE (using the same public encryption key pk).

We follow the security argument of [2] to show that the EStE-based signcryption scheme satisfies a meaningful security requirement (privacy and authenticity properties). In [2], two security formalizations are considered: Outsider security and insider security. In outsider security, adversaries are outsiders who only know the public keys (p, pk, pk_e). On the other hand, in insider security, adversaries are insiders who know either the signing key sk or the decryption key sk_e in addition to the public keys (p, pk, pk_e). We focus on insider security since it is stronger. The insider security is defined in terms of *induced* PKE and DS schemes (See [2] for the meaning of *induced*). That is, we say that a signcryption scheme is *insider-secure* if the induced PKE and DS schemes are secure. More specifically, we say that a signcryption scheme is *insider-secure* against CPAs and CMAs if the induced PKE and DS schemes satisfy the indistinguishability against CPAs and the existential unforgeability against CMAs, respectively (For simplicity, we don't consider the indistinguishability against chosen-ciphertext attacks). Following the security argument in [2], we can show that (1) if the PKE scheme of E_2 satisfies the indistinguishability against CPAs then the induced PKE scheme does so (The indistinguishability of E_1 does not matter) and (2) if the DS scheme of S satisfies the existential unforgeability against CMAs then the induced DS scheme does so. Therefore, we can say that the EStE-based signcryption scheme provides a meaningful security if E_2 and S are secure as in the two statements. A next question is how to obfuscate the EStE-based signcryption functionality.

A.2 Obfuscation for EStE

A secure obfuscator for an ES functionality can be used to obfuscate the EStE-based signcryption functionality since the second encryption E_2 is just a public operation. In other words, given an obfuscated ES program, we can append a program for performing the second encryption to it so that the resulting program computes the EStE composition, where we don't need any extra secret information. Therefore, by an argument similar to Section 5.4, we can show that the resulting obfuscator satisfies the ACVBP against distinguishers having sampling access to the signcryption and signing oracles and that the security of the DS scheme is preserved even when adversaries are given an obfuscated signcryption program. A question here is what kind of security we can achieve for the signcryption scheme (rather than the DS scheme) when adversaries are given an obfuscated signcryption program. This is not a trivial question. For example, the insider security of the signcryption scheme is violated when adversaries have access to the obfuscated program and the secret decryption key. The formal security argument is outside the scope of this proceedings version.

Public-Key Encryption in the Bounded-Retrieval Model

Joël Alwen[1], Yevgeniy Dodis[1,⋆], Moni Naor[2,⋆⋆], Gil Segev[2,⋆⋆⋆], Shabsi Walfish[3], and Daniel Wichs[1]

[1] New York University (NYU), New York, USA
{jalwen,dodis,wichs}@cs.nyu.edu
[2] Weizmann Institute of Science, Rehovot, Israel
{moni.naor,gil.segev}@weizmann.ac.il
[3] Google Inc. Mountain View, USA
shabsi@google.com

Abstract. We construct the *first* public-key encryption scheme in the *Bounded-Retrieval Model* (BRM), providing security against various forms of adversarial "key leakage" attacks. In this model, the adversary is allowed to learn arbitrary information about the decryption key, subject only to the constraint that the overall amount of "leakage" is bounded by at most ℓ bits. The goal of the BRM is to design cryptographic schemes that can flexibly tolerate arbitrarily leakage bounds ℓ (few bits or many Gigabytes), by *only* increasing the size of secret key proportionally, but keeping *all the other parameters* — including the size of the public key, ciphertext, encryption/decryption time, and the number of secret-key bits accessed during decryption — *small and independent of* ℓ.

As our main technical tool, we introduce the concept of an *Identity-Based Hash Proof System* (IB-HPS), which generalizes the notion of hash proof systems of Cramer and Shoup [CS02] to the identity-based setting. We give three different constructions of this primitive based on: (1) bilinear groups, (2) lattices, and (3) quadratic residuosity. As a result of independent interest, we show that an IB-HPS almost immediately yields an Identity-Based Encryption (IBE) scheme which is secure against (small) partial leakage of the target identity's decryption key. As our main result, we use IB-HPS to construct public-key encryption (and IBE) schemes in the Bounded-Retrieval Model.

1 Introduction

Traditionally, the security of cryptographic schemes has been analyzed in an idealized setting, where an adversary only sees the specified "input/output behavior" of a scheme, but has no other access to its internal secret state. Unfortunately, in the real world, an adversary may often learn some partial information

⋆ Research supported by NSF grants CNS-0831299 and CNS-0716690.

⋆⋆ Research supported in part by a grant from the Israel Science Foundation.

⋆⋆⋆ Research supported by the Adams Fellowship Program of the Israel Academy of Sciences and Humanities, and by a grant from the Israel Science Foundation.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 113–134, 2010.

about secret state via various *key leakage* attacks. Such attacks come in a large variety and include *side-channel attacks*, where the physical realization of a cryptographic primitive can leak additional information, such as the computation-time, power-consumption, radiation/noise/heat emission etc. The cold-boot attack of Halderman et al. [HSH$^+$08] is another example of a key-leakage attack, where an adversary can learn (imperfect) information about memory contents of a machine, even after the machine is powered down. Lastly, and especially relevant to this work, we will also consider key-leakage attacks where a remote adversary hacks into a target computer, or infects it with some malware, allowing her to download large amounts of secret-key information from the system. Schemes that are proven secure in an idealized setting, without key leakage, may become completely insecure if the adversary learns even a small amount of information about the secret key. Indeed, even very limited leakage attacks have been shown to have devastating consequences for the security of many natural schemes.

In this work, we study the design of leakage-resilient public-key encryption schemes, which are provably secure even in the presence of some limited key-leakage attacks. In particular, we will assume that the attacker can learn *any efficiently computable function of the secret key*, subject only to the constraint that the total amount of information learned (i.e. the output size of the leakage function) is bounded by ℓ bits, where ℓ is some arbitrary "leakage parameter" of the system. Clearly, at this level of generality, the secret-key size s must be strictly greater than the leakage-parameter ℓ. In the literature, there seems to be a distinction between two related models of leakage, which differ in how they treat the leakage-parameter ℓ in relation to the secret-key size s.

Relative-Leakage Model. In the model of *relative leakage*, firs studied by Akavia Goldwasser and Vaikuntanathan, [AGV09], the key-size s is chosen in the same way as in standard (non leakage-resilient) cryptographic schemes: it is based on a security parameter, and is usually made as *small* as possible (e.g. 1024 bits) to give the system some sufficient level of security. Once the key-size s is determined, the allowed leakage ℓ should be *relatively large in proportion to* s so that e.g. up to 50% of the key can be leaked without compromising security. Therefore, the relative-leakage model implicitly assumes that, no matter what the key-size is, a leakage attack can reveal at most some *relatively small fraction* of the key. This assumption is very reasonable for some attacks, such as the cold-boot attack, where all memory contents decay uniformly over time.

Bounded-Retrieval Model (BRM). The *Bounded-Retrieval Model (BRM)* [Dzi06, CLW06, ADW09] is a generalization of the relative-leakage model. In this model, the leakage-parameter ℓ is an arbitrary and independent parameter of the system, which is based on practical considerations about how much leakage the system needs to tolerate on an *absolute scale*. The secret-key size s is then chosen flexibly, depending on the security parameter *and* the leakage parameter ℓ, so as to simultaneously provide a sufficient level of security while allowing up to ℓ bits of leakage. Therefore, we can tolerate settings where the leakage ℓ might be small (several bits) or huge (several Gigabytes) by flexibly increasing the secret-key size s depending on (and necessarily exceeding) the leakage parameter ℓ.[1] Of course, the key-size s should be as small as possible otherwise, so that the allowed leakage ℓ is a large *relative portion* of s as well.

With the additional flexibility in secret-key size, the BRM imposes an added efficiency requirement: the *public-key size, ciphertext size, encryption-time and decryption-time* must remain small, only depending on the security parameter, *and essentially independent of the leakage-parameter* ℓ. In other words, ℓ could potentially grow to the order of Gigabytes, and still result in a usable system, where the secret key is huge, but the public-key size, ciphertext size and encryption/decryption times are not much different from those of standard cryptosystems. This also means that the number of secret-key bits accessed during decryption (called *locality* from now on) must remain small and essentially independent of the flexibly growing secret-key size.

The flexibility of the BRM seems necessary to protect against large classes of key-leakage attacks. For example, if the key size is (only) proportional to the security parameter, several consecutive side-channel readings of a handful of bits might already leak the entire secret key. Therefore, for natural side-channel attacks (such as radiation/heat/noise emission) it might already make sense to make ℓ moderately large (say on the order of Megabytes) to get security. The main intention of the BRM in prior works, which we also focus on here, is to offer a novel method for protecting systems against hacking/malware attacks, where an adversary can download large amounts of information from an attacked system. It is clear that no security can be achieved using standard-sized (e.g. 1,024 bit) secret keys, as the adversary can download such keys in their entirety. However, it may be conceivable that the adversary still cannot download *too much* (e.g. many Gigabytes) worth of information because: (1) the bandwidth between the attacker and the system may be too slow to allows this, (2) the operating-system security may detect such large levels of leakage, or (3) such attacks would simply not be cost-effective. Therefore we can conceivably protect against such attacks by just making the leakage-parameter ℓ large enough (e.g. potentially many Gigabytes), and using a proportionally larger secret-key-size s. Having a large secret key may, by itself, not be a major concern due to the increasing size and affordability of local storage. On the other hand, it is crucial that the other efficiency measures of the system – ciphertext and public-key sizes, encryption and decryption times – must not degrade with the growth of ℓ.

1.1 Our Results

As our main contribution, we construct the first leakage-resilient Public-Key Encryption (PKE) scheme in the BRM. Along the way, we develop new notions and get results of independent interest. In particular, we:

- Develop a new notion of an Identity-Based Hash Proof System (IB-HPS), which naturally yields Identity-Based Encryption (IBE) schemes.

[1] Historically, the BRM setting envisioned ℓ as being necessarily huge. Here we take a more general view of the BRM, insisting only that the key size can be set flexibly based on the leakage ℓ.

- Give three constructions of $\mathsf{IB\text{-}HPS}$ based on the ideas behind three prior IBE schemes: [Gen06, BGH07, GPV08]. In particular, we show that the notion of $\mathsf{IB\text{-}HPS}$ unifies these seemingly unrelated constructions under a single framework. As a result, we get constructions of $\mathsf{IB\text{-}HPS}$ under (1) a bilinear Diffie-Hellman type assumption (2) the quadratic-residuosity assumption (3) the Learning With Errors (LWE) assumption. The first scheme is secure in the standard model, while the latter two rely on Random Oracles or, alternatively, non-standard interactive assumptions.
- Show that an IBE based on $\mathsf{IB\text{-}HPS}$ can easily be made leakage-resilient, in the relative-leakage model.
- Show how to use $\mathsf{IB\text{-}HPS}$ to construct public-key encryption (PKE) schemes in the BRM, allowing for arbitrary large leakage-bounds, while preserving efficiency. Our techniques also naturally extend to allow for the construction of IBE schemes in the BRM.
- Develop new information-theoretic tools to analyze our construction of PKE in the BRM. Namely, we define a new notion of *approximate* hash functions (where only elements that are far in Hamming distance are unlikely to collide) and generalize the Leftover-Hash Lemma to approximate hashing.
- Show how to achieve CCA security for our leakage-resilient IBE and PKE in BRM constructions.

Before describing our construction of PKE in the BRM, it is instructive to understand why this problem is non-trivial, and therefore we begin with some naïve approaches, which we improve in several steps.

Naïve Approach: Inflating the Security Parameter. As the first step of getting a PKE in the BRM, we would like to simply design a leakage-resilient PKE scheme that allows for arbitrarily large leakage-bounds ℓ, without necessarily meeting the additional efficiency requirements of the BRM. Luckily, there are several recent PKE schemes in the *relative-leakage model* [AGV09, NS09] where the leakage-bound $\ell(\lambda)$ is a large portion of the key-size $s(\lambda)$ which, in turn, depends on a security parameter λ. Therefore, one simple solution is to simply artificially inflate the security parameter λ sufficiently, until $s(\lambda)$ and, correspondingly, $\ell(\lambda)$ reach the desired level of leakage we would like to tolerate. Unfortunately, it is clear that this approach gets extremely inefficient very fast – e.g. to allow for Gigabytes worth of leakage, we may need to perform exponentiations on group elements with Gigabyte-long description sizes.

Better Approach: Leakage-Amplification via Parallel Repetition. As an improvement over the previous suggestion, we propose an alternative which we call *parallel-repetition.* Assume we have a leakage-resilient PKE scheme in the relative-leakage model, tolerating ℓ-bits of leakage, for some small ℓ. We can create a new "parallel-repetition scheme", by taking n independent copies of the above PKE with key-pairs $(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_n, \mathsf{sk}_n)$ and setting the secret-key of the new scheme to be $\overline{\mathsf{sk}} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$ and the public key to be $\overline{\mathsf{pk}} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$. To encrypt under the repetition scheme, a user would n-out-of-n secret-share the message m, and, encrypt each share m_i under the public key

pk_i. One may hope to argue that, if an adversary learns fewer than $n\ell$ bits about the secret-key $\overline{\mathsf{sk}}$ of the repetition scheme, then there is at least one secret key sk_i about which the adversary learns fewer than ℓ bits, thus maintaining security. Therefore, the hope is that parallel-repetition *amplifies leakage-resilience* from ℓ bits to $n\ell$ bits, and thus lets us meet any leakage-bound just by increasing n sufficiently. In terms of efficiency, the parallel-repetition approach will usually be more efficient than artificially inflating the security parameter, but it is still far from the requirements of the BRM: the public-key size, ciphertext size, and encryption/decryption times are all proportional to n, and therefore must grow as we strive to tolerate more and more leakage.

Security of Parallel-Repetition? Surprisingly, we do not know how to formalize the hope that parallel-repetition amplifies leakage-resilience generically via a reduction. Such a reduction would need to use an attacker that expects a public key and $n\ell$ bits of leakage on its secret key in the repetition scheme, to break the original scheme with ℓ bits of leakage. Unfortunately, it does not seem like there is any way to embed a challenge public key pk_i into $\overline{\mathsf{pk}}$, and faithfully simulate the output of an arbitrary leakage-function $f(\overline{\mathsf{sk}})$ with $n\ell$-bit output, by only learning $g(\mathsf{sk}_i)$ for some $g(\cdot)$ with ℓ bit output. In fact, as a subject of future work, we believe that there is a black-box separation showing that no such reduction can succeed *in general.* Luckily, we show that (a variant of) parallel-repetition amplifies leakage for schemes of a special form, which we will discuss later. For now, let us get back to the issue of efficiency, which we still need to resolve.

Improvement I: Improved Efficiency via Random Selection. To decrease ciphertext size and encryption/decryption times, the encryptor selects some random subset $\{r_1, \ldots, r_t\} \subseteq \{1 \ldots n\}$ of t indices, and targets the ciphertext to the corresponding public keys $\mathsf{pk}_{r_1}, \ldots, \mathsf{pk}_{r_t}$ (e.g. t-out-of-t secret-shares the message m and encrypts each share m_i under the public key pk_{r_i}). Intuitively, if an adversary learns much less than $n\ell$ bits of leakage about $\overline{\mathsf{sk}}$, then there should be *many* component-keys sk_i for which the adversary learns less than ℓ bits. Therefore the encryptor should select at least one index corresponding to such a key with large probability, when t is made proportional to the security parameter, and potentially much smaller than n. Although the ciphertext size and encryption/decryption times (and locality) are now only proportional to the security parameter, the size of the public key still grows with n, and so this scheme is still not appropriate for the BRM in terms of efficiency.

Improvement II: Small Public-Key Size via IBE. A natural solution to having a short public key is to use *identity-based encryption* (IBE) instead of standard PKE. This way, the public key of the repetition scheme is simply a short *master public key* of an IBE scheme, while the secret key $\overline{\mathsf{sk}} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$ consists of secret-keys for some fixed “identities” $\mathsf{ID}_1, \ldots, \mathsf{ID}_n$. Together, the above two improvements yield a scheme which meets the efficiency requirements of the BRM: the public-key size, ciphertext size, encryption/decryption times are now only proportional to the security parameter and independent of n, which can grow flexibly.

Security of the IBE-Based PKE in BRM Construction? In order to show that the resulting scheme, utilizing the two proposed improvements, is a PKE in the BRM we need to show the following. If we start with a leakage-resilient IBE that allows for ℓ-bits of leakage, then the construction amplifies this to any desired amount ℓ' just by increasing the number of secret keys n sufficiently. Unfortunately, it turns out that this is not the case in general and, in the full version of this work [ADN$^+$09], we construct a counterexaple. That is, we can construct an artificial IBE scheme which is leakage-resilient in the relative leakage model, with leakage ℓ, but the above construction does not amplify leakage-resilience beyond $\ell' = \ell$, no matter how large n is. The problem is that, conceivably, after observing *all* n secret keys for n identities, it might be possible to come up with a very short "compressed" key (e.g. whose size is independent of n) which allows one to decrypt ciphertexts for *each one* of the given n identities. Our main result is to show that (a variant of) the construction is secure, if the leakage-resilient IBE has some additional underlying structure, which we call an Identity-Based Hash Proof System ($\mathsf{IB\text{-}HPS}$).

Hash Proof Systems and Identity-Based Hash Proof Systems. Recently, Naor and Segev [NS09] showed how to use a *hash proof system (HPS)* to construct leakage-resilient PKE in the relative-leakage model. Following, [KPSY09, NS09], we view an HPS as a *key-encapsulation mechanism (KEM)* with special structure.[2] A KEM consists of a key-generation procedure $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, an encapsulation procedure $(c, k) \leftarrow \mathsf{Encap}(\mathsf{pk})$ which produces ciphertext/randomness pairs (c, k), and a decapsulation procedure $k{=}\mathsf{Decap}(c, \mathsf{sk})$, which uses the secret key sk to recover the randomness k from a ciphertext c. A KEM allows a sender that knows pk, to securely agree on randomness k with a receiver that possesses sk, by sending an encapsulation-ciphertext c. A *hash proof system* is a KEM with the following two properties:

- There exists an *invalid-encapsulation procedure* $c \leftarrow \mathsf{Encap}^*(\mathsf{pk})$, so that ciphertexts generated by $\mathsf{Encap}^*(\mathsf{pk})$ are computationally indistinguishable from those generated by $\mathsf{Encap}(\mathsf{pk})$, *even given the secret key* sk.
- For a fixed pk and *invalid ciphertext* c generated by $\mathsf{Encap}^*(\mathsf{pk})$, the output of $\mathsf{Decap}(c, \mathsf{sk})$ is *statistically* uniform, over the randomness of sk. This property can only hold if a fixed pk leaves statistical entropy in sk.

Notice the difference between valid and invalid ciphertexts. For a fixed pk, a *valid* c, produced by $(c, k) \leftarrow \mathsf{Encap}(\mathsf{pk})$, always decapsulated to the same value k, no matter which secret key sk is used to decapsulate it. On other hand, an invalid c produced by $c \leftarrow \mathsf{Encap}^*(\mathsf{pk})$, decapsulated to a statistically random value based on the randomness of sk.

[2] Our informal description and definition of HPS here, which will also be a basis of our formal definition of $\mathsf{IB\text{-}HPS}$ in Section 3.1, is a simplified version of the standard one. Although the two are *not* technically equivalent, the standard definition implies ours, which is in-turn sufficient for leakage-resilience and captures the main essence of HPS.

The above two properties are sufficient to prove KEM security, showing that for $(c, k) \leftarrow \mathsf{Encap}(\mathsf{pk})$, an attacker given c cannot distinguish k from uniform. The proof proceeds in two steps:

1. We replace the honestly generated $(c, k) \leftarrow \mathsf{Encap}(\mathsf{pk})$ with $c' \leftarrow \mathsf{Encap}^*(\mathsf{pk})$ and $k' \leftarrow \mathsf{Decap}(c', \mathsf{sk})$.
2. The value $k' = \mathsf{Decap}(c', \mathsf{sk})$ is statistically uniform over the choice of sk, which is unknown to the adversary.

As Naor and Segev noticed in [NS09], this proof also works in the presence of leakage since step (1) holds even if the adversary saw *all of* sk, and step (2) is information-theoretic, so we can argue that ℓ bits of leakage about sk will only reduce the statistical entropy of k' by at most ℓ bits. To agree on a uniform value k in the presence of leakage, we just compose the KEM with a randomness extractor.

The main benefit of this proof strategy is that, after switching valid/invalid ciphertexts in the first step, we can argue about leakage using a purely information-theoretic analysis. We observe that it is therefore relatively easy to show that (a variant of) parallel repetition amplifies leakage-resilience, since it amplifies the statistical entropy of the secret key $\overline{\mathsf{sk}} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$. In this work, we generalize the notion of HPS to the identity-based setting by defining Identity-Based Hash Proof System (IB-HPS) in a natural way. First of all, this gives us a general framework for constructing leakage-resilient IBE schemes in the relative-leakage model. Second of all, it also allows us to prove that a variant of the previously proposed leakage-amplification technique (using an IB-HPS rather than just any IBE) can indeed be used to get PKE (and IBE) schemes in the BRM.

1.2 Related Work

Restricted Models of Leakage-Resilience. Several other models of leakage-resilience have appeared in the literature. They differ from the model we described in the that they restrict the *type*, as well as *amount*, of information that the adversary can learn. For example, the work on *exposure resilient cryptography* [CDH+00, DSS01, KZ03] studies the case where an adversary can only learn some small *subset of the physical bits of the secret key*. Similarly, [ISW03] studies how to implement arbitrary computation in the setting where an adversary can observe a small *subset of the physical wires of a circuity*. Unfortunately, these models fail to capture many meaningful side-channel attacks, such as learning the hamming-weight of the bits or their parity.

In their seminal work, Micali and Reyzin [MR04] initiated the formal modeling of side-channel attacks under the axiom that *"only computation leaks information"*, where each invocation of a cryptographic primitive leaks a function of *only* the bits accessed during that invocation. Several primitives have been constructed in this setting including stream ciphers [DP08, Pie09] and signatures [FKPR10]. On the positive side, this model only imposes a bound on the amount of information learned during each invocation of a primitive, but not on the overall amount of information that the attacker can get throughout the lifetime of the system. On the negative side, this model fails to capture many leakage-attacks, such as the cold-boot attack of [HSH$^+$08], where *all* memory contents leak information, even if they were never accessed.

Certainly, all of the restricted models fail to capture hacking/malware attacks, where it is very conceivable that an attacker can compute *even complicated functions* of *all* information stored on the system.

Relative-Leakage Model. Several constructions of primitives in the relative-leakage model have appeared recently. The works of [AGV09, NS09] construct public-key encryption schemes in this model, and [KV09] constructs signatures. The works of [DKL09, DGK$^+$10] considers a yet-stronger model of leakage-resilience, called the *auxiliary input model*, where the leakage-function need only be one-way (and not necessarily length-bounded), and constructs symmetric-key and public-key encryption in this model.

BRM. The Bounded-Retrieval Model was (concurrently) proposed by Di Crescenzo et al. [CLW06] and Dziembowski [Dzi06]. The name serves as an analogy to the Bounded Storage Model (BSM) of [Mau92], which restricts the amount of data that an adversary can *store after observing a huge public random string*, rather than the amount of data an adversary can *retrieve from a huge secret key.* With the exception of [ADW09], all of the work on the BRM is in the symmetric-key setting, where two parties share a huge secret key. The recent work of Alwen et al. [ADW09] gave the first public-key results in the BRM, by constructing identification schemes, (variants of) signatures, and authenticated-key-agreement protocols. However, these primitives cannot be used to encrypt a message non-interactively, as is done in the current work. Moreover, the authenticated-key agreement protocols of [ADW09] required the use of Random Oracles, while we offer (some) constructions in the standard model. We note that many of the prior schemes in the BRM and BSM employ ideas similar to the "parallel repetition" and "random-subset selection" that we described in the introduction. However, the proof-techniques in this paper differ significantly from previous works.

2 Preliminaries

Notation. For an integer n, we use the notation $[n]$ to denote the set $[n] \stackrel{\text{def}}{=} \{1, \ldots, n\}$. For a randomized function f, we write $f(x; r)$ to denote the unique output of f on input x with random coins r. We write $f(x)$ to denote a random variable for the output of $f(x; r)$, over the random coins r. For a set S, we let U_S denote the uniform distribution over S. For an integer $v \in \mathbb{N}$, we let U_v denote the uniform distribution over $\{0,1\}^v$, the bit-strings of length v. For a distribution or random variable X we write $x \leftarrow X$ to denote the operation of sampling a random x according to X. For a set S, we write $s \leftarrow S$ as shorthand for $s \leftarrow U_S$.

Entropy. The *min-entropy* of a r.v. X is $\mathbf{H}_\infty(X) \stackrel{\text{def}}{=} -\log(\max_x \Pr[X = x])$. This is a standard notion of entropy used in cryptography, since it measures the worst-case predictability of X. The *average conditional min-entropy* [DORS08] of X given Z is defined by $\widetilde{\mathbf{H}}_\infty(X|Z) \stackrel{\text{def}}{=} -\log\left(\mathbb{E}_{z\leftarrow Z}\left[2^{-\mathbf{H}_\infty(X|Z=z)}\right]\right)$. This measures the worst-case predictability of X by an adversary that may observe a correlated variable Z.

Statistical Distance and Extractors. The *statistical distance* between X, Y is defined by $\mathbf{SD}(X,Y) = \frac{1}{2}\sum_x |\Pr[X=x] - \Pr[Y=x]|$. We write $X \approx_\varepsilon Y$ to denote $\mathbf{SD}(X,Y) \le \varepsilon$, and $X \approx Y$ to denote that the statistical distance is negligible. An extractor [NZ96] can be used to extract uniform randomness out of a weakly-random value which is only assumed to have sufficient min-entropy. Our definition follows that of [DORS08], which is defined in terms of conditional min-entropy.

Definition 1 (Extractors). *We say that an efficient randomized function* $\mathsf{Ext} : \{0,1\}^u \to \{0,1\}^v$ *is an* (m,ε)-extractor *if for all* X, Z *such that* X *is distributed over* $\{0,1\}^u$ *and* $\widetilde{\mathbf{H}}_\infty(X|Z) \ge m$, *we get* $(Z, R, \mathsf{Ext}(X;R)) \approx_\varepsilon (Z, R, U_v)$ *where* R *is a random variable for the coins of* Ext.

Due to space constraints, almost all the proofs are omitted from the conference version of this paper. Please see the full version [ADN$^+$09] for proofs and additional details.

3 Identity-Based Hash Proof System (IB-HPS)

3.1 Definition

An *Identity-Based Hash Proof System* (IB-HPS) consists of PPT algorithms: Setup, KeyGen, Encap, Encap*, Decap. The algorithms have the following syntax.

$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ **:** The setup algorithm takes as input a security parameter λ and produces the *master public key* mpk and the *master secret key* msk. The master public key defines an *identity set* $\mathcal{ID}$, and an *encapsulated-key set* $\mathcal{K}$. All other algorithms $\mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Decap}, \mathsf{Encap}^*$ implicitly include mpk as an input.

$\mathsf{sk}_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk})$ **:** For any identity $\mathsf{ID} \in \mathcal{ID}$, the KeyGen algorithm uses the master secret key msk to sample an identity secret key $\mathsf{sk}_{\mathsf{ID}}$.

$(c,k) \leftarrow \mathsf{Encap}(\mathsf{ID})$ **:** The *valid* encapsulation algorithm creates pairs (c,k) where c is a valid ciphertext, and $k \in \mathcal{K}$ is the encapsulated-key.

$c \leftarrow \mathsf{Encap}^*(\mathsf{ID})$ **:** The alternative *invalid* encapsulation algorithm which samples an invalid ciphertext c.

$k \leftarrow \mathsf{Decap}(c, \mathsf{sk}_{\mathsf{ID}})$ **:** The decapsulation algorithm is deterministic, and takes an identity secret key $\mathsf{sk}_{\mathsf{ID}}$ and a ciphertext c and outputs the encapsulated key k.

We require that an Identity-Based Hash Proof System satisfies the following properties.

I. Correctness of Decapsulation. For any values of $\mathsf{mpk}, \mathsf{msk}$ produced by $\mathsf{Setup}(1^\lambda)$, any $\mathsf{ID} \in \mathcal{ID}$ we have:

$$\Pr\left[k \neq k' \;\middle|\; \begin{array}{c} \mathsf{sk}_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk}) \\ (c,k) \leftarrow \mathsf{Encap}(\mathsf{ID}) \quad , \quad k' = \mathsf{Decap}(c, \mathsf{sk}_{\mathsf{ID}}) \end{array}\right] \le \mathrm{negl}(\lambda)$$

II. Valid/Invalid Ciphertext Indistinguishability. The valid ciphertexts generated by Encap and the invalid ciphertexts generated by Encap^* should be indistinguishable *even given the identity secret key.* In particular, we define the following distinguishability game between an adversary $\mathcal{A}$ and a challenger.

$\mathsf{VI\text{-}IND}(\lambda)$

Setup: The challenger computes $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and gives mpk to the adversary $\mathcal{A}$.

Test Stage 1: The adversary $\mathcal{A}$ adaptively queries the challenger with $\mathsf{ID} \in \mathcal{ID}$ and the challenger responds with $\mathsf{sk}_{\mathsf{ID}}$.

Challenge Stage: The adversary selects an *arbitrary* challenge identity $\mathsf{ID}^* \in \mathcal{ID}$.
The challenger chooses $b \leftarrow \{0,1\}$.
If $b = 0$ the challenger computes $(c, k) \leftarrow \mathsf{Encap}(\mathsf{ID}^*)$.
If $b = 1$ the challenger computes $c \leftarrow \mathsf{Encap}^*(\mathsf{ID}^*)$.
The challenger gives c to the adversary $\mathcal{A}$.

Test Stage 2: The adversary $\mathcal{A}$ adaptively queries the challenger with $\mathsf{ID} \in \mathcal{ID}$ and the challenger responds with $\mathsf{sk}_{\mathsf{ID}}$.

Output: The adversary $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ which is the output of the game. We say that $\mathcal{A}$ *wins* the game if $b' = b$.

Note: In test stages 1,2 the challenger computes $\mathsf{sk}_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk})$ the first time that ID is queried and responds to all future queries on the same ID with the same $\mathsf{sk}_{\mathsf{ID}}$.

Note that, during the challenge phase, the adversary can choose *any* identity ID^*, and possibly even one for which it has seen the secret key $\mathsf{sk}_{\mathsf{ID}^*}$ in Test Stage 1 (or the adversary can simply get $\mathsf{sk}_{\mathsf{ID}^*}$ in Test Stage 2). We define the advantage of $\mathcal{A}$ in distinguishing valid/invalid ciphertexts to be $\mathsf{Adv}^{\mathsf{VI\text{-}IND}}_{\mathsf{IB\text{-}HPS},\mathcal{A}}(\lambda) \stackrel{\text{def}}{=} |\Pr[\mathcal{A} \text{ wins }] - \frac{1}{2}|$. We require that $\mathsf{Adv}^{\mathsf{VI\text{-}IND}}_{\mathsf{IB\text{-}HPS},\mathcal{A}}(\lambda) = \mathrm{negl}(\lambda)$.

III. Universality/Smoothness/Leakage-Smoothness. Other than properties I and II, we will need one additional information theoretic property. Essentially, we want to ensure that there are many possibilities for the decapsulation of an *invalid* ciphertext, which are left undetermined by the public parameters of the system. We define three flavors of this property as follows.

Definition 2 (Universal $\mathsf{IB\text{-}HPS}$). *We say that an* $\mathsf{IB\text{-}HPS}$ *is* (m, ρ)**-universal** *if, for any fixed values of* $\mathsf{mpk}, \mathsf{msk}$ *produced by* $\mathsf{Setup}(1^\lambda)$*, and any fixed* $\mathsf{ID} \in \mathcal{ID}$ *the following two properties hold:*

1. *Let* $\mathsf{SK} \leftarrow \mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk})$ *be a random variable. Then* $\mathbf{H}_\infty(\mathsf{SK}) \geq m$*.*
2. *For any fixed distinct values* $\mathsf{sk}_{\mathsf{ID}} \neq \mathsf{sk}'_{\mathsf{ID}}$ *in the support of* SK*, we have*

$$\Pr_{c \leftarrow \mathsf{Encap}^*(\mathsf{ID})}[\mathsf{Decap}(c, \mathsf{sk}_{\mathsf{ID}}) = \mathsf{Decap}(c, \mathsf{sk}'_{\mathsf{ID}})] \leq \rho.$$

Notice the significant difference between valid and invalid ciphertexts. For valid ciphertexts c, the correctness of decapsulation ensures that there is a single value

$k \in \mathcal{K}$ such that $\mathsf{Decap}(c, \mathsf{sk}_{\mathsf{ID}}) = k$ for (virtually) all choices of $\mathsf{sk}_{\mathsf{ID}}$ (of which there are many by (1)). On the other hand, for invalid ciphertexts c, (2) ensures that it is highly unlikely that any two distinct secret-keys $\mathsf{sk}_{\mathsf{ID}}$ will decapsulate c to the same value k.

Definition 3 (Smooth/Leakage-Smooth IB-HPS). *We say that an* IB-HPS *is* **smooth** *if, for any* fixed values *of* $\mathsf{mpk}, \mathsf{msk}$ *produced by* $\mathsf{Setup}(1^\lambda)$*, any* $\mathsf{ID} \in \mathcal{ID}$*, we have:*

$$\mathbf{SD}(\ (c, k)\ ,\ (c, k')\ ) \le \mathrm{negl}(\lambda)$$

where $c \leftarrow \mathsf{Encap}^*(\mathsf{ID})$*,* $k' \leftarrow U_{\mathcal{K}}$ *and* k *is sampled by choosing* $\mathsf{sk}_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk})$ *and computing* $k = \mathsf{Decap}(c, \mathsf{sk}_{\mathsf{ID}})$*. We say that an* IB-HPS *is* **ℓ-leakage-smooth** *if, for any (possibly randomized) function* $f(\cdot)$ *with* ℓ*-bit output, we have:*

$$\mathbf{SD}(\ (c, f(\mathsf{sk}_{\mathsf{ID}}), k)\ ,\ (c, f(\mathsf{sk}_{\mathsf{ID}}), k')\ ) \le \mathrm{negl}(\lambda)$$

where $c, k, \mathsf{sk}_{\mathsf{ID}}, k'$ *are sampled as above. Note, for this property,* f *need not be efficient.*

3.2 Relations between Universality, Smoothness and Leakage-Smoothness

The following theorem is a simple consequence of the leftover hash lemma.

Theorem 1. *Assume that an* IB-HPS*, with key set* $\mathcal{K} = \{0,1\}^v$*, is* (m, ρ)*-universal. Then it is also* ℓ*-leakage smooth as long as* $\ell \le m - v - \omega(\log(\lambda))$ *and* $\rho \le \frac{1}{2^v}(1 + \mathrm{negl}(\lambda))$*.*

We also show how to convert a *smooth* IB-HPS ($\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Encap}^*, \mathsf{Decap}$) into a *leakage-smooth* IB-HPS using an extractor $\mathsf{Ext} : \mathcal{K} \to \{0,1\}^v$. We define:

- $\mathsf{Encap}_2(\mathsf{ID})$: Choose $(c, k) \leftarrow \mathsf{Encap}(\mathsf{ID}), k' \leftarrow \mathsf{Ext}(k; r)$ where r is a random seed. Output $c' = (c, r), k'$.
- $\mathsf{Encap}^*_2(\mathsf{ID})$: Choose a random seed r and $c \leftarrow \mathsf{Encap}^*(\mathsf{ID})$. Output $c' = (c, r)$.
- $\mathsf{Decap}_2(c', \mathsf{msk})$: Parse $c' = (c, r)$. Compute $k = \mathsf{Decap}(c, \mathsf{msk}), k' = \mathsf{Ext}(k; r)$. Output k'.

Theorem 2. *Assume that an* IB-HPS *is* smooth *and that* $|\mathcal{K}| = 2^m$*. Let* $\mathsf{Ext} : \mathcal{K} \to \{0,1\}^v$ *be an* $(m - \ell, \varepsilon)$*-extractor for some* $\varepsilon = \mathrm{negl}(\lambda)$*. Then the above transformation produces an* ℓ-leakage-smooth IB-HPS.

4 Constructions of IB-HPS

4.1 A Construction of IB-HPS Based on Bilinear Groups

Background: Let $\mathbb{G}, \mathbb{G}_T$ be two (multiplicative) groups of prime order p and let g be a generator of $\mathbb{G}$. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a map from $\mathbb{G}$ to the *target group* $\mathbb{G}_T$. We say that the group $\mathbb{G}$ is *bilinear* if we have

1. **Bilinearity:** For all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$ we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. **Non-degeneracy:** For the generator g of $\mathbb{G}$, we get $e(g, g) \neq 1$.
3. **Efficiency:** Operations (multiplication, exponentiation) in $\mathbb{G}, \mathbb{G}_T$ and the map e can be computed efficiently.

We assume the existence of a group-generation algorithm $\mathcal{G}(1^\lambda)$ which outputs a tuple $(\mathbb{G}, \mathbb{G}_T, g, e(\cdot,\cdot), p)$ where $\mathbb{G}$ is a bilinear group of prime order p.

We will rely on the *truncated augmented bilinear Diffie-Hellman exponent assumption* (q-TABDHE) from [Gen06]. We define the two distributions

$$D^{(0)}_{\lambda,q} = \left(g, g^\alpha, g^{(\alpha^2)}, \ldots, g^{(\alpha^q)}, g', g'^{(\alpha^{q+2})}, e\left(g^{(q+1)}, g'\right)\right)$$

and

$$D^{(1)}_{\lambda,q} = \left(g, g^\alpha, g^{(\alpha^2)}, \ldots, g^{(\alpha^q)}, g', g'^{(\alpha^{q+2})}, Z\right)$$

where $(\mathbb{G}, \mathbb{G}_T, g, e(\cdot,\cdot), p) \leftarrow \mathcal{G}(1^\lambda)$, $g' \leftarrow \mathbb{G}$, $\alpha \leftarrow \mathbb{Z}_p$, and $Z \leftarrow \mathbb{G}_T$. For any algorithm $\mathcal{B}$, the *distinguishing advantage of* $\mathcal{B}$ *in the* q-TABDHE *problem* is $\mathsf{Adv}^{\mathsf{TABDHE}}_{\mathcal{B}}(\lambda, q) \overset{\text{def}}{=} \left|\Pr\left[\mathcal{B}\left(D^{(0)}_{\lambda,q}\right) = 0\right] - \Pr\left[\mathcal{B}\left(D^{(1)}_{\lambda,q}\right) = 0\right]\right|$.

Definition 4. *We say that the* q-TABDHE *assumption holds if, for any PPT* $\mathcal{B}$, $\mathsf{Adv}^{\mathsf{TABDHE}}_{\mathcal{B}}(\lambda, q) = \text{negl}(\lambda)$. *We say that the* TABDHE *assumption holds if* q-TABDHE *holds for all polynomial* q.

CONSTRUCTION: We now present the construction of $\mathsf{IB\text{-}HPS}$ which is based directly on Gentry's IBE [Gen06].

$\mathsf{Setup}(1^\lambda)$: Let $(\mathbb{G}, \mathbb{G}_T, g, e, p) \leftarrow \mathcal{G}(1^\lambda)$. Let $h \leftarrow \mathbb{G}$, $\alpha \leftarrow \mathbb{Z}_p$ and $g_1 := g^\alpha$.
Set $\mathsf{mpk} = (\mathbb{G}, \mathbb{G}_T, g, e, p, g_1, h)$ and set $\mathsf{msk} = \alpha$.
The identity set is $\mathcal{ID} = \mathbb{Z}_p \setminus \{\alpha\}$ and the encapsulated-key set is $\mathcal{K} = \mathbb{G}_T$. [a]

$\mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk})$: For $\mathsf{ID} \in \mathcal{ID}$, choose $r_{\mathsf{ID}} \leftarrow \mathbb{Z}_p$ and compute $h_{\mathsf{ID}} = (hg^{-r_{\mathsf{ID}}})^{1/(\alpha - \mathsf{ID})}$. Output $\mathsf{sk}_{\mathsf{ID}} = (r_{\mathsf{ID}}, h_{\mathsf{ID}})$.

$\mathsf{Encap}(\mathsf{ID})$: Choose random $s \in \mathbb{Z}_p$ and compute $u = g_1^s g^{-s\mathsf{ID}}$, $v = e(g, g)^s$ and output $c = (u, v), k = e(g, h)^s$.

$\mathsf{Encap}^*(\mathsf{ID})$: Choose a random pair $(s, s') \in \mathbb{Z}_p$ subject to the constraint $s \neq s'$.
Let $u = g_1^s g^{-s\mathsf{ID}}$, $v = e(g, g)^{s'}$ and output $c = (u, v)$.

$\mathsf{Decap}(c, \mathsf{sk}_{\mathsf{ID}})$: Parse $c = (u, v)$ and output $k = e(u, h_{\mathsf{ID}}) v^{r_{\mathsf{ID}}}$.

[a] The set $\mathcal{ID}$ is defined in terms of the secret α. Given $\mathsf{ID} \in \mathbb{Z}_p$, one can efficiently check if $\mathsf{ID} \in \mathcal{ID}$ by checking if $g^{\mathsf{ID}} \overset{?}{=} g_1$.

Essentially, various parts of Gentry's proof already show that the scheme satisfies the properties of $\mathsf{IB\text{-}HPS}$. We provide a moularized proof of the following theorem in the full version [ADN+09].

Theorem 3. *Under the* TABDHE *assumption, the above construction is an* $\mathsf{IB\text{-}HPS}$ *which is simultaneously* smooth *and* (m, ρ)-universal *for* $\rho = 0$ *and* $m = \log(p)$. *More precisely, the valid/invalid ciphertext indistinguishability property holds under the* q-TABDHE *assumption for any adversary making at most* q *secret-key and leakage queries.*

4.2 Parameters of Three IB-HPS Constructions

In the full version of this work [ADN$^+$09], we give two additional constructions of IB-HPS based on the recent IBE schemes of [BGH07, GPV08]. Here we, just give a short note on each construction and explain its parameters. We will be interested in the following:

1. The *actual identity-key size* $\hat{m}$: the number of bits needed to efficiently represent an identity secret key $\mathsf{sk}_{\mathsf{ID}}$.
2. The *encapsulated-key size* $v = \log(|\mathcal{K}|)$: the size of the encapsulated key.
3. The min-entropy m and the universality ρ: the values for which the scheme is (m, ρ)-universal.

An important parameter is the ratio $\frac{m}{\hat{m}}$, which determines the amount of *relative leakage* that our IBE and PKE in BRM constructions can handle. We note that *all* of the schemes satisfy the definition of *smoothness.*

A Scheme Based on Bilinear Groups. The parameters of our construction from the previous section, based on Gentry's IBE, are:

$$\hat{m} = 2\log(p) + O(1) \quad , \quad m = \log(p) \quad , \quad \frac{m}{\hat{m}} \approx \frac{1}{2} \quad , \quad v = \log(p) \quad , \quad \rho = 0.$$

where p is the (prime) order of an appropriate bilinear-group $\mathbb{G}$.

A Scheme Based on Quadratic Residuosity. We show that the IBE scheme of Boneh, Gentry and Hamburg [BGH07] contains a IB-HPS. The construction and proof essentially follow [BGH07] (with a minor modification in how identity secret keys are chosen, to get universality). The scheme is secure under the Quadratic Residuosity assumption in the Random Oracle model, or under a non-standard *interactive quadratic residuosity assumption* in the standard model. The parameters of interest are:

$$\hat{m} = \log(N) \quad , \quad m = 1 \quad , \quad \frac{m}{\hat{m}} = \frac{1}{\log(N)} \quad , \quad v = 1 \quad , \quad \rho = 0.$$

where N is an appropriately sized RSA modulus. Unfortunately, it is not clear how to make the scheme leakage-smooth for any $\ell > 0$, since the secret-key entropy m is too small to extract even a single bit. This problem can be fixed, as will be done in the BRM, by using parallel-repetition to amplify the entropy. Still, the relative leakage of the scheme will be poor because of the poor ratio of the entropy m to actual-key-size $\hat{m}$.

A Scheme Based on Lattices. We show how to get a construction of IB-HPS using the IBE scheme of Gentry, Peikert and Vaikuntanathan [GPV08]. Note that this IBE construction was already observed to be leakage-resilient by [AGV09], but this does not imply that it is an IB-HPS. In fact, we need to make some simple modifications so that the scheme satisfies our definition. The security of the scheme is based on a (decisional) Learning With Errors (LWE) assumption, in the random oracle model. Note that this assumption can be reduced to the GapSVP problem for lattices, using the techniques of [Reg05, Pei09].[3] We show

[3] We note that our construction requires that we use some (slightly) super-polynomial modulus q in the LWE problem, which means that we need to assume GapSVP is hard against some (slightly) super-polynomial time adversaries.

that, for any constant $\varepsilon > 0$, there exists some setting of the actual-key-size $\hat{m}$ so that:

$$m = (1-\varepsilon)\hat{m} \quad , \quad \frac{m}{\hat{m}} = (1-\varepsilon) \quad , \quad v = 1 \quad , \quad \rho = \frac{1}{2}(1+\mathrm{negl}(\lambda)).$$

Note that, by Theorem 2, this construction is therefore *already* ℓ-leakage smooth, for any $\ell \leq m - \omega(\log(\lambda))$, without any need to apply an extractor.

5 Leakage-Resilient IBE from IB-HPS

We define what it means for an Identity-Based Encryption (IBE) scheme to be resistant to key leakage attacks and show how to use an IB-HPS to construct such an IBE scheme. Our notion of leakage-resilience only allows leakage-attacks against the secret keys of the various identities, but *not* the master secret key. Also, we only allow the adversary to perform leakage attacks before seing the challenge ciphertext. As noted by [AGV09, NS09, ADW09], this limitation is inherent to (non-interactive) encryption schemes since otherwise the leakage function can simply decrypt the challenge ciphertext and output its first bit.

Definition. Recall an IBE scheme consists of four PPT algorithms Setup, KeyGen, Encrypt, and Decrypt. We omit discussion of the usual correctness requirements. We define the *semantic security game*, parameterized by a security parameter λ and a leakage parameter ℓ as the following game between an adversary $\mathcal{A}$ and a challenger.

$\mathsf{IBE\text{-}SS}(\lambda, \ell)$

Setup: Challenger computes $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, gives mpk to the adv. $\mathcal{A}$.

Test Stage 1: The adv. $\mathcal{A}$ adaptively makes the following queries:

Secret-Key Queries: On input $\mathsf{ID} \in \mathcal{ID}$, the challenger replies with $\mathsf{sk}_{\mathsf{ID}}$.

Leakage Queries: On input $\mathsf{ID} \in \mathcal{ID}$, a PPT function $f : \{0,1\}^* \to \{0,1\}$, the challenger replies with $f(\mathsf{sk}_{\mathsf{ID}})$.

Challenge Stage: The adversary selects two messages $\mathsf{m}_0, \mathsf{m}_1 \in \mathcal{M}$ and a challenge identity $\mathsf{ID}^* \in \mathcal{ID}$ which *never appeared* in a secret-key query and appeared in *at most* ℓ leakage queries. The challenger chooses $b \leftarrow \{0,1\}$ uniformly at random and computes $c \leftarrow \mathsf{Encrypt}(\mathsf{ID}^*, \mathsf{m}_b)$ and gives c to the adversary $\mathcal{A}$.

Test Stage 2: The adversary gets to make *secret-key queries* for arbitrary $\mathsf{ID} \neq \mathsf{ID}^*$. The challenger replies with $\mathsf{sk}_{\mathsf{ID}}$.

Output: The adversary $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$. We say that the adversary *wins* the game if $b' = b$.

Note: In test stages 1,2 the challenger computes $\mathsf{sk}_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}(\mathsf{ID}, \mathsf{msk})$ the first time that ID is queried (in a secret-key or leakage query) and responds to all future queries on the same ID with the same $\mathsf{sk}_{\mathsf{ID}}$.

The *advantage* of an adversary $\mathcal{A}$ in the *semantic security game with leakage* ℓ is $\mathsf{Adv}^{\mathsf{IBE\text{-}SS}}_{\mathsf{IBE},\mathcal{A}}(\lambda, \ell) \overset{\mathrm{def}}{=} \left|\Pr[\mathcal{A} \text{ wins }] - \frac{1}{2}\right|$.

Definition 5 (Leakage-Resilient IBE). *An* IBE *scheme is* ℓ-leakage-resilient*, if the advantage of any any PPT adversary* $\mathcal{A}$ *in the semantic security game* *with leakage* ℓ*, is* $\mathsf{Adv}^{\mathsf{IBE\text{-}SS}}_{\mathsf{IBE},\mathcal{A}}(\lambda, \ell) = \mathrm{negl}(\lambda)$*. We define the* relative leakage *of the scheme to be* $\alpha \overset{def}{=} \ell/\hat{m}$*, where* $\hat{m}$ *is the number of bits needed to efficiently store identity secret keys* $\mathsf{sk}_{\mathsf{ID}}$*.*

Construction: The construction of a leakage-resilient IBE from a leakage-smooth IB-HPS is almost immediate, by simply using the encapsulated key as a one-time-pad to encrypt a message. In particular, given an IB-HPS where the encapsulated key set $\mathcal{K}$ has some group structure $(\mathcal{K}, +)$ (e.g. bit-strings with $\oplus$), we construct an IBE scheme with the same identity set $\mathcal{ID}$ and message set $\mathcal{M} = \mathcal{K}$. The $\mathsf{Setup}, \mathsf{KeyGen}$ algorithms are the same for both primitives and $\mathsf{Encrypt}, \mathsf{Decrypt}$ are defined by:

$\mathsf{Encrypt}(\mathsf{ID}, \mathsf{m})$**:** Choose $(c_1, k) \leftarrow \mathsf{Encap}(\mathsf{ID})$ and let $c_2 = k + \mathsf{m}$.
Output $c = (c_1, c_2)$.
$\mathsf{Decrypt}(c, \mathsf{sk}_{\mathsf{ID}})$**:** For $c = (c_1, c_2)$, compute $k = \mathsf{Decap}(c_1, \mathsf{sk}_{\mathsf{ID}})$.
Output $\mathsf{m} = c_2 - k$.

Note that the Encap^* algorithm of the IB-HPS is not used in the construction, but will be used to argue security.

Theorem 4. *Assume that we start with an* ℓ*-leakage-smooth* IB-HPS*. Then the above construction yields an* ℓ*-leakage-resilient* IBE*.*

6 Leakage Amplification of IB-HPS

We now show how to construct an ℓ-leakage-smooth IB-HPS, for arbitrarily large values of ℓ, meeting the efficiency requirements of the BRM. This will be the main step towards building PKE (and IBE) schemes in the BRM. We start with a IB-HPS scheme $\Pi_1 = (\mathsf{Setup}, \mathsf{KeyGen}_1, \mathsf{Encap}_1, \mathsf{Encap}^*_1, \mathsf{Decap}_1)$ and compile it into a new IB-HPS scheme $\Pi_2 = (\mathsf{Setup}, \mathsf{KeyGen}_2, \mathsf{Encap}_2, \mathsf{Encap}^*_2, \mathsf{Decap}_2)$, where the identity secret keys can be made arbitrarily large, so as to achieve ℓ-leakage-smoothness for a large ℓ. We will assume there is a one-to-one function $H : \mathcal{ID}_2 \times [n] \to \mathcal{ID}_1$ where $\mathcal{ID}_1, \mathcal{ID}_2$ are the identity sets of Π_1, Π_2 respectively. In the constructed scheme, the identity secret key of each $\mathsf{ID} \in \mathcal{ID}_2$ consists of n components $\mathsf{sk}_{\mathsf{ID}} = (\mathsf{sk}_{\mathsf{ID}}[1], \ldots, \mathsf{sk}_{\mathsf{ID}}[n])$, where each component $\mathsf{sk}_{\mathsf{ID}}[i]$ is an independently sampled identity secret key for an identity $H(\mathsf{ID}, i) \in \mathcal{ID}_1$ of the original scheme. Here, n will be a key-size parameter, which gives us flexibility in the size of the identity secret key in the constructed scheme, and will depend on the desired leakage-parameter ℓ. The encapsulation procedure $\mathsf{Encap}_2(\mathsf{ID})$ will target only a small subset of t-out-of-n of the identities $H(\mathsf{ID}, i)$, and decapsulation Decap_2 will only need to read the values $\mathsf{sk}_{\mathsf{ID}}[i]$ associated with these t identities. Here t will be a *locality-parameter* which can be much smaller than (and independent of) n. A formal description of the construction appears in Figure 1. It is described abstractly in terms of arbitrary parameters n, t, v. In the theorem that follows, we show how to instantiate these appropriately based on the setting of ℓ, λ.

Let $\Pi_1 = (\mathsf{Setup}, \mathsf{KeyGen}_1, \mathsf{Encap}_1, \mathsf{Encap}^*_1, \mathsf{Decap}_1)$ be a IB-HPS with encapsulated-key-set $\mathcal{K}$ and identity-set $\mathcal{ID}_1$.
Let $n, t, v \in \mathbb{Z}^+$. We call n a *key-size parameter*, t a *locality parameter* and v a *output-size* parameter.
Let $H : \mathcal{ID}_2 \times [n] \to \mathcal{ID}_1$ be a one-to-one function for some set $\mathcal{ID}_2$.[a]
Let $\mathcal{G}$ be a $\frac{1}{2^v}$-universal hash function family of functions $g : \mathcal{K}^t \to \{0,1\}^v$.

Define $\Pi_2 = (\mathsf{Setup}, \mathsf{KeyGen}_2, \mathsf{Encap}_2, \mathsf{Encap}^*_2, \mathsf{Decap}_2)$ as follows:

$\mathsf{Setup}(1^\lambda)$**:** The setup procedure is the same as that of Π_1.
$\mathsf{KeyGen}_2(\mathsf{ID}, \mathsf{msk})$**:** For $i \in [n]$, sample $\mathsf{sk}_{\mathsf{ID}}[i] \leftarrow \mathsf{KeyGen}_1(H(\mathsf{ID}, i), \mathsf{msk})$. Output $\mathsf{sk}_{\mathsf{ID}} = (\mathsf{sk}_{\mathsf{ID}}[1], \ldots, \mathsf{sk}_{\mathsf{ID}}[n])$.
$\mathsf{Encap}_2(\mathsf{ID})$**:** Choose t random indices $\overline{r} = (r_1, \ldots, r_t) \leftarrow [n]^t$. Choose $g \leftarrow \mathcal{G}$. For $i \in \{1, \ldots, t\}$, compute: $(c_i, k_i) \leftarrow \mathsf{Encap}_1(H(\mathsf{ID}, r_i))$. Let $\overline{c} = (c_1, \ldots, c_t)$. Output: $C = (\overline{r}, \overline{c}, g)$, $k = g(k_1, \ldots, k_t)$.
$\mathsf{Encap}^*_2(\mathsf{ID})$**:** Choose t random indices $\overline{r} = (r_1, \ldots, r_t) \leftarrow [n]^t$. Choose $g \leftarrow \mathcal{G}$. For $i \in \{1, \ldots, t\}$, compute: $c_i \leftarrow \mathsf{Encap}^*_1(H(\mathsf{ID}, r_i))$. Let $\overline{c} = (c_1, \ldots, c_t)$. Output: $C = (\overline{r}, \overline{c}, g)$.
$\mathsf{Decap}_2(C, \mathsf{sk}_{\mathsf{ID}})$**:** Parse $C = (\overline{r}, \overline{c}, g)$. Compute $k_i = \mathsf{Decap}_1(c_i, \mathsf{sk}_{\mathsf{ID}}[r_i])$ for $i \in \{1, \ldots, t\}$. Output $k = g(k_1, \ldots, k_t)$.

[a] A collision-resistant hash function (CRHF) would suffice here as well.

Fig. 1. Leakage-Amplification of an IB-HPS: Construction of Π_2 from Π_1

For the analysis of the construction, we need to define a new parameter called the *effective key size* m'. This is the minimal value such that, for any fixed $\mathsf{mpk}, \mathsf{msk}, \mathsf{ID}$, the number of values that $\mathsf{sk}_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}(\mathsf{ID})$ can take on is bounded by $2^{m'}$. If the actual key size is $\hat{m}$ and the key entropy is m, then $\hat{m} \geq m' \geq m$. Note that in all of our constructions, m/m' is a constant (even when $m/\hat{m}$ is not, as is the case for our QR-based construction).

Theorem 5. *Assume Π_1 is an (m, ρ)-universal* IB-HPS *with effective key size m', where $\rho < 1$ and $m/m' > 0$ are constants. Then, for any constant $\varepsilon > 0$ and any polynomial $v(\lambda)$, there exists some $t = O(v + \lambda)$ so that, for any polynomial $n(\lambda)$, the above construction of Π_2 with parameters (n, t, v) is an ℓ-leakage-smooth* IB-HPS *where $\ell(\lambda) = (1-\varepsilon)nm - v - \lambda$. The encapsulated-key-set of Π_2 is $\mathcal{K} = \{0,1\}^v$.*

The full proof of the above theorem appears in [ADN+09]. We give some intuition here. It is easy to see that Π_2 satisfies correctness. Also, the valid/invalid ciphertext indistinguishability property of Π_2 follows by a simple hybrid argument. Therefore, we only need to show ℓ-leakage smoothness, for the ℓ given by the theorem statement. For a fixed $\mathsf{mpk}, \mathsf{msk}, \mathsf{ID}$ in Π_2, the entropy of the random variable $\mathsf{SK}_{\mathsf{ID}} \sim \mathsf{KeyGen}_2(\mathsf{ID}, \mathsf{msk})$, is amplified to $\mathbf{H}_\infty(\mathsf{SK}_{\mathsf{ID}}) \geq nm$, since it consists of n independently sampled secret keys of Π_1. If we could show that the scheme is also ρ'-universal, for some small $\rho' \leq (\frac{1}{2^v} + \text{negl}(\lambda))$, then we could rely on Theorem 1 to show leakage-smoothness. Unfortunately, this is not the case. The problem is that, if two values $\mathsf{sk}_{\mathsf{ID}} \neq \mathsf{sk}'_{\mathsf{ID}}$ in the constructed scheme differ in only one position j, then $\mathsf{Decap}_2(C, \mathsf{sk}_{\mathsf{ID}}) = \mathsf{Decap}(C, \mathsf{sk}'_{\mathsf{ID}})$ as long as the ciphertext C does not "select" j, which happens with large probability. Therefore, to analyze the leakage smoothness of the construction, we define a new notion called *approximately universal hashing*, where we only insist that values which are far from each other in Hamming distance (over some alphabet) are unlikely to collide. We then show a variant of the leftover-hash lemma, called the *approximate leftover-hash lemma* holds for approximate hashing. Lastly, we show that the decapsulation procedure $\mathsf{Decap}_2(C, \mathsf{sk}_{\mathsf{ID}})$ of the amplified scheme Π_2 is approximately universal, for appropriate parameters, when $C \leftarrow \mathsf{Encap}^*(\mathsf{ID})$.[4] Combining these results, we get the parameters of the theorem.

7 Public-Key Encryption and IBE in the BRM

A public-key encryption (PKE) scheme in the BRM consists of the algorithms (KeyGen, Encrypt, Decrypt), which are all parameterized by a security parameter λ *and a leakage parameter* ℓ. The syntax and the correctness property of an encryption scheme follow the standard notion of public-key encryption. We define the following *semantic-security game with leakage* ℓ between an adversary $\mathcal{A}$ and a challenger.

$\mathsf{SemS}(\lambda, \ell)$

Key Generation: The challenger computes $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda, 1^\ell)$ and gives pk to the adversary $\mathcal{A}$.

Leakage: The adversary $\mathcal{A}$ selects a PPT function $f : \{0,1\}^* \to \{0,1\}^\ell$ and gets $f(\mathsf{sk})$ from the challenger.

Challenge: The adversary $\mathcal{A}$ selects two messages $\mathsf{m}_0, \mathsf{m}_1$. The challenger chooses $b \leftarrow \{0,1\}$ uniformly at random and gives $c \leftarrow \mathsf{Encrypt}(\mathsf{m}_b, \mathsf{pk})$ to the adversary $\mathcal{A}$.

Output: The adversary $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$. We say that $\mathcal{A}$ wins the game if $b' = b$.

For any adversary $\mathcal{A}$, the *advantage of* $\mathcal{A}$ in the above game is defined as $\mathsf{Adv}^{\mathsf{SemS}}_{\mathsf{PKE},\mathcal{A}}(\lambda, \ell) \overset{\text{def}}{=} \left|\Pr[\mathcal{A} \text{ wins }] - \frac{1}{2}\right|$.

Definition 6 (Leakage-Resilient PKE). *A public-key encryption scheme* PKE *is leakage-resilient, if for any polynomial* $\ell(\lambda)$ *and any PPT adversary* $\mathcal{A}$*, we have* $\mathsf{Adv}^{\mathsf{SemS}}_{\mathsf{PKE},\mathcal{A}}(\lambda, \ell(\lambda)) = \mathrm{negl}(\lambda)$.

Definition 7 (PKE in the BRM). *We say that a leakage-resilient PKE scheme is a* PKE in the BRM*, if the public-key size, ciphertext size, encryption-time and decryption-time (and the number of secret-key bits read by decryption) are independent of the leakage-bound* ℓ*. More formally,* **there exist** *polynomials* $\mathsf{pksize}, \mathsf{ctsize}, \mathsf{encT}, \mathsf{decT}$*, such that,* **for any** *polynomial* ℓ *and any* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda, 1^{\ell(\lambda)})$*,* $\mathsf{m} \in \mathcal{M}$*,* $c \leftarrow \mathsf{Encrypt}(\mathsf{m}, \mathsf{pk})$*, the scheme satisfies:*

[4] For approximate universality, we think of the "big key" $\mathsf{sk}_{\mathsf{ID}}$ as consisting of n alphabet symbols, with one symbol for each component key $\mathsf{sk}_{\mathsf{ID}}[i]$.

1. *Public-key size is* $|\mathsf{pk}| \leq O(\mathsf{pksize}(\lambda))$*, ciphertext size is* $|c| \leq O(\mathsf{ctsize}(\lambda, |\mathsf{m}|))$.
2. *Run-time of* $\mathsf{Encrypt}(\mathsf{m}, \mathsf{pk})$ *is* $\leq O(\mathsf{encT}(\lambda, |\mathsf{m}|))$.
3. *Run-time of* $\mathsf{Decrypt}(c, \mathsf{sk})$*, and the number of bits of* sk *accessed, is* $\leq O(\mathsf{decT}(\lambda, |\mathsf{m}|))$.

The relative-leakage *of the scheme is* $\alpha \overset{def}{=} \ell/|\mathsf{sk}|$.

We can generalize the above definition to IBE schemes. A leakage-resilient IBE is an *IBE in the BRM* if the master-public-key size, master-secret-key size, ciphertext size and encryption/decryption times are bounded by polynomials independent of ℓ.

Theorem 6 (PKE and IBE in BRM). *Assume that we have an* (m, ρ)*-universal* IB-HPS *satisfying the conditions of Theorem 5 and having actual key size* $\hat{m}$*. Then, for any constant* $\varepsilon > 0$ *and any polynomial* v*, we get PKE (resp. IBE) schemes in the BRM with message space* $\mathcal{M} = \{0, 1\}^v$ *and:*

1. *Public-key size (resp. master public/secret key size) is the same as that of the underlying* IB-HPS.
2. *The locality-parameter is* $t = O(v + \lambda)$*. The # of secret-key bits accessed during decryption is* $t\hat{m}$.
3. *Ciphertext-size/encryption-time/decryption-time differ by a factor of* t *from those of the underlying* IB-HPS.
4. *Relative leakage is* $\alpha \geq \frac{m}{\hat{m}}(1 - \varepsilon)$*, for sufficiently large values of the leakage-parameter* ℓ*. In particular, for large enough* ℓ*, the secret-key size (resp. identity-secret-key size) is* $\leq \frac{\hat{m}}{m}(1 + \varepsilon)\ell$.

Proof. Follows directly from leakage-amplification (Theorem 5). For any leakage-parameter ℓ, the key-size parameter n in the construction of Π_2 in Figure 1 is made just large enough so that $\ell \leq (1 - \varepsilon)nm - v - \lambda$. Therefore, Π_2 is ℓ-leakage smooth. By Theorem 4, this yields an ℓ-leakage resilient IBE. The efficiency parameters are obvious from the construction, so it is easy to see that we get an IBE in the BRM. By ignoring all identities except for a single one, we naturally get a PKE in the BRM. The relative leakage is $\alpha = \frac{\ell}{\hat{m}n} \approx \frac{m}{\hat{m}}(1 - \varepsilon)$, for ℓ large enough in relation to v, λ. □

8 Extensions

In the full version [ADN+09] of this work, we show several extensions to the results from the previous section. We describe them briefly here.

CCA Security. We show that the main ideas underlying our approach can be extended to deal with chosen-ciphertext attacks. We present constructions of encryption schemes that are resilient to leakage even under chosen-ciphertext attacks. That is, these schemes are semantically secure even against an adversary that is allowed to submit both leakage queries and decryption queries. We first consider identity-based encryption, and show that the CCA-secure variant of Gentry's scheme [Gen06] can be generalized to deal with leakage. We then consider public-key encryption in the BRM, and observe that the generic transformation from chosen-plaintext security to chosen-ciphertext security, using the Naor-Yung paradigm [NY90], also applies in the BRM.

Shorter Ciphertexts via Anonymous Encapsulation. We notice that two of our IB-HPS constructions, based on lattices and quadratic residuosity, have additional structure, which allows for a more efficient version of our leakage-amplification construction. In the construction shown in Figure 1, the ciphertext C of the constructed scheme Π_2 contains t ciphertexts $c_1, \ldots, c_t$ of the underlying scheme Π_1, where $t = O(\lambda + v)$. We show how to reduce this to a single ciphertext if we start with an IB-HPS construction Π_1 that has an additional property, which we call *anonymous encapsulation*. Such a scheme has two additional procedures:

- $(c, s) \leftarrow \mathsf{EncapC}()$, which samples a ciphertext c together with a trapdoor s *without* knowing the target ID.
- $k = \mathsf{EcnapK}(c, s, \mathsf{ID})$, which (deterministically) computes k for any ID, given c and a trapdoor s.

Note that the procedures $\mathsf{EncapC}, \mathsf{EcnapK}$ (like Encap) are implicitly parameterized by the master public key mpk.

Definition 8 (Anonymous Encapsulation). *An* IB-HPS *has* anonymous encapsulation *if there exist efficient procedures* $\mathsf{EncapC}, \mathsf{EcnapK}$ *as above, such that, for any fixed* $\mathsf{mpk}, \mathsf{msk}, \mathsf{ID}$*, sampling* $(c, k) \leftarrow \mathsf{Encap}(\mathsf{ID})$ *is equivalent to sampling* $(c, s) \leftarrow \mathsf{EncapC}()$ *and computing* $k = \mathsf{EcnapK}(c, s, \mathsf{ID})$*.*

For the lattice and quadratic-residuosity based constructions, the procedures EncapC, EcnapK are already implicitly defined by Encap, which first samples c anonymously (independently of ID) and then computes k for a given ID using the randomness s that was used to generate c.

There are several advantages to IB-HPS schemes that have the anonymous-encapsulation property. Firstly, it's easy to see that the IBE constructed from such schemes has *anonymity*, in that the ciphertext does not reveal the target identity. Perhaps more importantly, anonymous encapsulation can be used to get an improved leakage-amplification scheme with shorter ciphertexts.[5] In particular, we modify the procedure $\mathsf{Encap}_2(\mathsf{ID})$ of the constructed Π_2 scheme, so that it samples a *single* ciphertext/trapdoor pair $(c, s) \leftarrow \mathsf{EncapC}_1()$ of the underlying scheme Π_1, and computes $k_i = \mathsf{EcnapK}_1(c, s, H(\mathsf{ID}, r_i))$ for each of of the t random indices $r_i \in [n]$. The ciphertexts of the constructed scheme therefore consist of $C = (\overline{r}, c, g)$, and contain only a single ciphertext c of the underlying scheme. To reduce the ciphertext size still further, we can employ the following optimizations:

1. Instead of sampling the indices $\overline{r} \leftarrow [n]^t$ uniformly at random, and communicating this choice in the ciphertext, we use use a *hitting sampler* to

[5] A similar technique is implicitly used to get shorter ciphertexts relative to the message length in the IBE constructions of [BGH07, GPV08].

sample $\overline{r} \in [n]^t$ efficiently. This choice can then be communicated using a seed of description size $\log(n) + O(\lambda + v)$, rather than the previous size $t \log(n) = O((\lambda + v)\log(n))$ needed to communicate $\overline{r}$ explicitly.

2. Use a γ-universal, instead of fully universal, hash function g, where $\gamma = \frac{1}{2^v}(1 + \text{negl}(\lambda))$. As observed in [SZ99], such hash functions can have description sizes $O(v + \lambda)$, only proportional to the output size, and not the somewhat larger input size.

We show that leakage-amplification still holds for the modified constructions, by showing that $\mathsf{Decap}_2(C, \cdot)$ is an approximately-universal hash function with appropriate parameters, when $C \leftarrow \mathsf{Encap}^*(\mathsf{ID})$. Unfortunately, the setting of the parameters requires that $\rho \leq \frac{1}{2^v}$ in the original scheme, which is only the case for our QR-based scheme but *not* the lattice-based scheme.

9 Comparison of PKE (and IBE) in BRM Constructions

In Table 1, we compare the efficiency and relative-leakage of our various IBE and PKE in BRM constructions. We assume that the plaintext size is $v = O(\lambda)$.[6] In all of the schemes, the leakage-parameter ℓ can be arbitrarily large and the relative leakage column indicates the ratio of leakage to secret-key size. The public-key size of all schemes is the same as the master-public-key size of the corresponding IB-HPS and the encryption/decryption times (and the number of bits accessed) differ by a multiplicative factor of $t = O(\lambda)$ from those of the underlying IB-HPS. The "CT expansion" column indicates the ratio of the ciphertext size in the BRM to that of the underlying IB-HPS. The "CT size in BRM" column measures the size of the ciphertext in the BRM on an absolute scale.[7] The value $\varepsilon > 0$ can be an arbitrary constant.

Table 1. Comparison of Our PKE in BRM Constructions

Scheme	Assumption	Relative Leakage	CT Size in BRM	CT Expansion
Bilinear-Groups [Gen06]	TABDHE	$(\frac{1}{2} - \varepsilon)$	$O(\lambda^2)$	$O(\lambda)$
Quadratic Residuosity [BGH07]	QR †	$\frac{1}{O(\lambda)}$	$O(\lambda)$	$O(1)$
Lattices [GPV08]	LWE/GapSVP †	$(1 - \varepsilon)$	$O(\lambda^4)$	$O(\lambda)$

† = Random Oracle Model/Interactive Assumption

[6] To encrypt larger messages, it is sufficient to encrypt a short $O(\lambda)$ sized key for a symmetric-key encryption scheme.

[7] Note that, to make a fair comparison, we assume that RSA moduli and bilinear-group elements have description sizes $O(\lambda)$. For our LWE based construction, the modulus q needs to be (slightly) super-polynomial, and we are pessimistic by just bounding its description size by $O(\lambda)$.

Acknowledgements

We would like to thank Vinod Vaikuntanathan for many enlightening discussions, and especially for his invaluable help in answering our technical questions about his recent lattice-related results. We would also like to thank Craig Gentry for his helpful discussion and for pointing us to the IBE scheme of [BGH07].

References

[ADN$^+$09] Alwen, J., Dodis, Y., Naor, M., Segev, G., Walfish, S., Wichs, D.: Public-key encryption in the bounded-retrieval model. Cryptology ePrint Archive, Report 2009/512 (2009), http://eprint.iacr.org/

[ADW09] 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)

[AGV09] 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)

[BGH07] Boneh, D., Gentry, C., Hamburg, M.: Space-efficient identity based encryption without pairings. In: FOCS, pp. 647–657 (2007)

[CDH$^+$00] Canetti, R., Dodis, Y., Halevi, S., Kushilevitz, E., Sahai, A.: Exposure-resilient functions and all-or-nothing transforms. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 453–469. Springer, Heidelberg (2000)

[CLW06] Di Crescenzo, G., Lipton, R.J., Walfish, S.: Perfectly secure password protocols in the bounded retrieval model. In: Halevi and Rabin [HR06], pp. 225–244 (2006)

[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)

[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)

[DKL09] Dodis, Y., Kalai, Y.T., Lovett, S.: On cryptography with auxiliary input. In: STOC (2009)

[DORS08] 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)

[DP08] Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: FOCS, pp. 293–302. IEEE Computer Society, Los Alamitos (2008)

[DSS01] Dodis, Y., Sahai, A., Smith, A.: On perfect and adaptive security in exposure-resilient cryptography. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 301–324. Springer, Heidelberg (2001)

[Dzi06] Dziembowski, S.: Intrusion-resilience via the bounded-storage model. In: Halevi and Rabin [HR06], pp. 207–224 (2006)

[FKPR10] 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)

[Gen06] Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)

[GPV08] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC 2008: Proceedings of the 40th annual ACM symposium on Theory of computing, pp. 197–206. ACM, New York (2008)

[HR06] Halevi, S., Rabin, T. (eds.): TCC 2006. LNCS, vol. 3876. Springer, Heidelberg (2006)

[HSH$^+$08] Alex Halderman, J., 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. USENIX Association, pp. 45–60 (2008)

[ISW03] 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)

[KPSY09] 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)

[KV09] 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)

[KZ03] Kamp, J., Zuckerman, D.: Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. In: FOCS, pp. 92–101 (2003)

[Mau92] Maurer, U.M.: Conditionally-perfect secrecy and a provably-secure randomized cipher. J. Cryptology 5(1), 53–66 (1992)

[MR04] Micali, S., Reyzin, L.: Physically observable cryptography (extended abstract). In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 278–296. Springer, Heidelberg (2004)

[NS09] 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)

[NY90] Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, pp. 427–437 (1990)

[NZ96] Nisan, N., Zuckerman, D.: Randomness is linear in space. J. Comput. Syst. Sci. 52(1), 43–52 (1996)

[Pei09] Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: STOC, pp. 333–342 (2009)

[Pie09] Pietrzak, K.: A leakage-resilient mode of operation. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 462–482. Springer, Heidelberg (2009)

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

[SZ99] Srinivasan, A., Zuckerman, D.: Computing with very weak random sources. SIAM J. Comput. 28(4), 1433–1459 (1999)

Protecting Circuits from Leakage: The Computationally-Bounded and Noisy Cases

Sebastian Faust[1,*], Tal Rabin[2], Leonid Reyzin[3,**],
Eran Tromer[4,***], and Vinod Vaikuntanathan[2]

[1] K.U. Leuven ESAT-COSIC/IBBT
[2] IBM Research
[3] Boston University
[4] MIT

Abstract. Physical computational devices leak side-channel information that may, and often does, reveal secret internal states. We present a *general* transformation that compiles any circuit into a new, functionally equivalent circuit which is resilient against well-defined classes of leakage. Our construction requires a *small*, *stateless* and *computation-independent* leak-proof component that draws random elements from a fixed distribution. In essence, we reduce the problem of shielding arbitrarily complex circuits to the problem of shielding a single, simple component.

Our approach is based on modeling the adversary as a powerful observer that inspects the device via a limited measurement apparatus. We allow the apparatus to access all the bits of the computation (except those inside the leak-proof component) and the amount of leaked information to grow unbounded over time. However, we assume that the apparatus is limited either in its computational ability (namely, it lacks the ability to decode certain linear encodings and outputs a limited number of bits per iteration), or its precision (each observed bit is flipped with some probability). While our results apply in general to such leakage classes, in particular, we obtain security against:

- *Constant depth circuits leakage*, where the measurement apparatus can be implemented by an AC^0 circuit (namely, a constant depth circuit composed of NOT gates and unbounded fan-in AND and OR gates), or an $\mathsf{ACC}^0[p]$ circuit (which is the same as AC^0, except that it also uses $\mathtt{MOD}_p$ gates) which outputs a limited number of bits.
- *Noisy leakage*, where the measurement apparatus reveals all the bits of the state of the circuit, perturbed by independent binomial noise. Namely, each bit of the computation is perturbed with probability p, and remains unchanged with probability $1 - p$.

[*] Supported in part by Microsoft Research through its PhD Scholarship Programme, by the IAP Programme P6/26 BCRYPT of the Belgian State (Belgian Science Policy), and FWO grant G.0225.07.

[**] Supported in part by NSF grants CNS-0831281 and CNS-0546614.

[***] Supported by NSF CyberTrust grant CNS-0808907 and AFRL grant FA8750-08-1-0088.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 135–156, 2010.

1 Introduction

The best of cryptographic algorithms are insecure when their implementations inadvertently reveal secrets to an eavesdropping adversary. Even when the software is flawless, practical computational devices leak information via numerous side channels, including electromagnetic radiation (visible and otherwise) [30,23], timing [7], power consumption [22], acoustic emanations [33], and numerous effects at the system architecture level (e.g., cache attacks [5,26,27]). Leaked information is even more easily accessible when the computational device is at the hands of an adversary, as is often the case for many modern devices such as smart-cards, TPM chips and (potentially stolen) mobile phones and laptops. Reducing such information leakage has proven excruciatingly difficult and costly, and its complete elimination is nowhere in sight.

There has lately been a growing amount of interest in coming up with precise definitions of security against side-channel attacks and in designing cryptographic algorithms that withstand these attacks (e.g., [24,19,28,17,11,8,29,3,25,9] and others). Micali and Reyzin [24] were the first to propose a general model of side-channel attacks. They model a side-channel attacker as a two part entity – the first is the *measurement apparatus* that performs measurements on the physical state of the device. This is done on behalf of the second entity which is the *adversarial observer*. The observer is assumed to be computationally powerful (e.g., polynomial-time or even unbounded), and takes as input the measurements of the apparatus. Thus, the power of the adversarial observer is primarily constrained by the quality of the information provided by the measurement apparatus.

It is interesting to note that even though computational devices leak abundantly, many side channel attacks are hard to carry out and some devices remain unbroken. This is due to the fact that *useful* measurements can often be difficult to realize in practice. Physical measurement apparatuses typically produce a "shallow" or "noisy" measurement of the state of the object, by combining some of its salient physical properties in a simple way. The measurement consists of a limited amount of information, obtained as a simple leakage function applied to the physical state of the device; any in-depth analysis happens only in the form of post-processing by the observer (rather than in the measurement apparatus).

In this work, we follow the paradigm of Ishai, Sahai, and Wagner [19] who construct a *general transformation* from any cryptographic algorithm into one that is functionally equivalent, but also leakage-resilient. The particular class of leakage functions they consider is the class of spatially local measurement functions, namely functions that read and output at most t bits of information. In particular, the leakage functions are completely oblivious of a large portion of the circuit's state.

In contrast, we are interested in security against global measurements, which are often easier to carry out than localized measurements that require a focus on specific wires or memory cells; in many side-channel attacks, the main practical difficulty for the attacker lies precisely in obtaining high spatial resolution and accuracy. Furthermore, global measurements are typically also more informative than local measurements. The question that motivates our work is whether, analogously to [19], we can construct a general circuit transformation that tolerates global side-channel measurements.

1.1 Our Results

Similar to Ishai et al. [19], we present a general transformation for arbitrary circuits that makes them resilient against certain classes of leakage. We now explain what these classes of leakage are and describe our techniques.

Measurement Apparatus. As in most prior work, the measurement apparatus in our model is not allowed to access some (very limited) portions of the computation. It can observe the rest of the computation, and return either a "computationally bounded" or a "noisy" function of the entire state.[1] Specifically, the measurement apparatus is modeled as computing either of the following types of *leakage functions*:

– a *computationally-bounded leakage function* f applied to the state of the device and all intermediate results that occur during the computation. The class of functions $\mathcal{L}$ from which f can be chosen models the practical limitations of the physical experimental setting available to the attacker. For example, $\mathcal{L}$ may consist of all functions computable by circuits of small depth.

  For the computational limitation to be meaningful, the function must also be limited in its output length (otherwise, the measurement apparatus could simply leak the entire state by "computing" the identity function).

– a *noisy leakage function*, where the measurement apparatus returns the accessed bit with probability $1 - p$ and flips it with probability p. The measurement apparatus can potentially access *all the bits of the computation* this way.

There are specific components of the circuit that we consider to be leak-free. We diverge from previous solutions by requiring that these components be *simple, stateless and computation-independent*. By this, we mean that the complexity of implementing the leak-free component is independent of the complexity of the computed function, and that it neither holds secrets nor maintains state. In particular, the leak-free component cannot hold the secret data used in the computation.

Specifically, our leak-free components, which we call *opaque gates*, are defined as follows. The opaque gate has *no inputs* and it outputs an element sampled according to a fixed distribution which is independent of the computation being carried out. For example, an opaque gate that we consider is one that samples t uniformly random bits subject to the condition that they have even parity.

[1] When we refer to the state of a computation, we mean all the intermediate values produced during the computation on a particular input. Once this computation is done, the intermediate state is erased to make room for new computations. Thus, the leakage function can access all the bits of the current computation, but *not the past computations*. In fact, this is necessary to achieve security.

The leakage function cannot observe the innards of the opaque gate, but it can observe the wires going into and coming out of it. Although the requirement of a leak-free component is a strong one, the leak-free components we require are minimal in many senses:

1. It is a fixed standardized functionality which can be designed and validated once and added to one's VLSI "cell library" — which is far better than having to devise separate protection mechanisms for every circuit of interest.
2. It has no secret keys, no inputs and no internal state, i.e., it is independent of the computation in the circuit and merely samples from a distribution.
3. Alternatively, because we only need samples from a distribution, we can have the opaque "gate" simply read them one by one from a precomputed list. Thus, it suffices to have leak-proof one-time storage (a consumable "tape roll") instead of leak-proof computation. This is a viable option if the computation is performed only a bounded number of times.

Many variations of the leak-proof component assumption have been made in the literature. We highlight some of these works below.

- The "Oblivious RAM" model of Goldreich and Ostrovsky [15,16] considered memory to be leaky and the computation to be on a leak-free secure processor which stores a long-term secret key.
- The model of Micali and Reyzin [24] (and subsequent works [11,29,12]) reversed these roles: they assume that the memory cells that are not accessed during a computation step do not affect the observable leakage from that stage and cannot be measured by the apparatus. They called it the "only computations leaks" assumption.[2]
- The model of Goldwasser et al. [17] (which, although presented in the one-time programs setting, can be transformed into the leakage-resilient setting) relaxes the assumption of Micali and Reyzin, assuming only that some *read-only* memory (which holds secrets correlated to the computation) is leak-free if it is not "touched". The circuit, however, can only be executed a single time (or more generally, a bounded number of times).

The adversarial observer is all-powerful, and in each invocation of the circuit, it comes up with an input to the circuit as well as a leakage function, and obtains the output of the computation (on the given input), together with the leakage. The adversary decides which leakage function to use in a particular invocation *adaptively*, depending on all the information it received so far. We design circuit transformations that withstand such adversaries, and obtain the following main results.

Theorem 1 (Informal). *Let t be a (statistical) security parameter. There are circuit transformations that convert any (possibly stateful) circuit C into a circuit $\widehat{C}$ that is resilient against the following leakage functions:*

[2] [11,29] point out that this requirement can be somewhat relaxed – it suffices that leakage of memory that is not used is independent of the leakage from computation.

- *Constant-depth* AC^0 *circuits whose output length in each invocation is bounded by* $t^{1-\delta}$*, for any* $\delta > 0$*, and whose output length over the course of time is unbounded.*
- *Noisy measurements that leak the entire state of the circuit in each invocation, where each bit flipped independently with probability* p*, for any constant* $p \in (0, 1/2]$*.*

In both cases, the size of the transformed circuit $\widehat{C}$ *is larger than the size of the original circuit* C *by a factor of* $O(t^2)$*.*

Both results follow from a more general transformation that protects against *any* leakage class, provided that it has an associated encoding scheme (See Theorem 2 for details). We should note that although AC^0 is not a particularly strong class of functions, it is strong enough to allow for measuring approximate Hamming weight of the values on the wires [2]: something routinely measured by side-channel attacks in practice.

1.2 Overview of the Techniques

To protect against the kinds of information leakage described above, we encode the computation in a way that prevents the powerful computing observer from gaining additional information about the computation. We show that, indeed, for certain classes of leakage, *any* computation can be so encoded: namely, we give a method for transforming arbitrary circuits into new circuits, which are still leaky but whose leakage is useless to the attacker (in the sense of offering no advantage over black-box access to the original circuit's functionality).

More precisely, given any linear secret sharing scheme Π and a leakage class $\mathcal{L}$ which cannot decode Π[3], we show an explicit construction that transforms any circuit C into a circuit $\widehat{C}$ that is resilient against leakage in $\mathcal{L}$.

The gist of the construction is to encode every wire of C into a bundle of wires in $\widehat{C}$ using Π, where each wire in the bundle carries a single share. Similarly to Ishai et al. [19], we transform each gate in C into a gadget in $\widehat{C}$ which operates on encoded bundles. The gadgets are carefully constructed to use Π internally in a way that looks "essentially random" to leakage functions in $\mathcal{L}$, and we show that this implies that the whole content of the transformed circuit remains "essentially random" to a leakage in $\mathcal{L}$. Hence, the adversary gets no advantage from his observation of the leakage; formally, this is captured by a simulation-based definition.

An important contribution of this work is a general technique for proving security of leakage-resilient circuit transformations. Namely, we capture a strong notion of leakage-resilience for circuits or parts thereof, by saying that they are *reconstructible* if there exist certain efficient simulators for their internal wires that fool the leakage class. We then show a *composition lemma*: if all parts of a circuit are reconstructible then so is the whole circuit. This implies security

[3] Technically, the requirement that we make for the class $\mathcal{L}$ is a little bit stronger then not being able to decode.

of the transformation. Thus, security of the overall transformation is reduced to the reconstructibility of the individual gadgets. Our specific results using linear secret-sharing schemes follow this route, and other transformations can be built by devising different gate gadgets and merely showing that each is reconstructible by itself.

Other Related Approaches. Recently, starting from the work of Akavia et al. [3], several results have appeared that show security against adversaries that learn arbitrary functions of the secret state of a device *without requiring leak-free components* (see [3,4,9,21,25] and the references therein). All these constructions assume that the total leakage does not exceed the size of the secret key; in contrast, the total leakage in our case can be unbounded (subject only to the condition that in every time period, it is bounded). Furthermore, these works design specific cryptographic primitives such as encryption and signatures, whereas we focus on a general leakage-resilient transformation.

Standaert et al. [35] consider security against particular attacks such as Hamming weight attacks and analyze in [28] the security of a block-cipher based construction of a pseudorandom number generator.

2 Preliminaries and Definitions

Notation. Throughout the paper, we let t denote the security parameter. For $n \in \mathbb{N}$, let $[1, n]$ denote the set of integers $\{1, \ldots, n\}$. We denote function composition by $f \circ g : x \mapsto f(g(x))$. If $\mathcal{L}_1$ and $\mathcal{L}_2$ are two sets of functions, then $\mathcal{L}_2 \circ \mathcal{L}_1$ is a set of functions $\{f \circ g \mid f \in \mathcal{L}_2, g \in \mathcal{L}_1\}$. Vectors, denoted $\boldsymbol{v} = (v_1, \ldots, v_n)$, will be treated as column vectors.

If $\mathcal{D}$ is a probability distribution, then the notation $d \leftarrow \mathcal{D}$ means that the random variable d is drawn from $\mathcal{D}$. (If D is a set with no distribution specified, then by default we assume the uniform distribution.) If $\mathcal{D}$ is a randomized algorithm, then $d \leftarrow \mathcal{D}(x)$ denotes the output of $\mathcal{D}$ on input x. The notation $\mathcal{D} \equiv \mathcal{D}'$ means the distributions $\mathcal{D}$ and $\mathcal{D}'$ are identical.

Circuits. We consider circuits whose wires carry elements of an arbitrary finite field $\mathcal{K}$; in particular, we may set $\mathcal{K} = GF(2)$ to speak of a Boolean circuit. We consider circuits composed of the following gates operating on elements of $\mathcal{K}$ (in addition to the input, output, and memory gates): $\oplus, \ominus$, and $\odot$ (which compute, respectively, the sum, difference, and product in $\mathcal{K}$, of their two inputs), the "coin flip" gate \$ (which has no inputs and produces a random independently chosen element of $\mathcal{K}$), and for every $\alpha \in \mathcal{K}$, the constant gate $\mathtt{const}_\alpha$ (which has no inputs and simply outputs α). Fanout is handled by a special copy gate that takes as input a single value and outputs two copies. Notice that copy gates compute the identity function (pass-through wires) and are present mainly for notational convenience.

For a circuit C containing w wires, a *wire assignment to* C is a string in $\mathcal{K}^w$, where each element represents a value on a wire of C. By $\mathcal{W}_C(X)$, we denote a distribution of wire assignments that is induced when a circuit C is being evaluated on an input X (in particular, if C is deterministic, then $\mathcal{W}_C(X)$ has only one element in its support). By $\mathcal{W}_C(X|Y)$, we denote the same distribution conditioned on the fact that the output of $C(X)$ was Y.

Two classes of circuits figure prominently in this paper.

- The first class of circuits is $\mathsf{SHALLOW}(d, s)$, the class of all deterministic circuits (i.e., ones without \$ gates) that have at most s $\oplus, \ominus$, and $\odot$ gates that are arranged at most d deep (i.e., the longest path in the circuit has at most d such gates on it).[4]
- The second is a class that contains a single *probabilistic* circuit $\mathcal{N}_p$ that gets as input a string $\boldsymbol{v}$, and outputs $\boldsymbol{w} = \boldsymbol{v} \oplus \boldsymbol{r}$, where each bit of $\boldsymbol{r}$ is *independently* 1 with probability p, and 0 with probability $1 - p$.

Stateful Circuits. A *stateful* circuit additionally contains memory gates, which have a single incoming edge and any number of outgoing edges.[5] Memory gates maintain state: at any clock cycle, a memory gate sends its current state down its outgoing edges and updates it according to the value of its incoming edge. Any cycle in the circuit must contain at least one memory gate.

The state of all memory gates at clock cycle i is denoted by M_i, with M_0 denoting the initial state. Inputs to and outputs from clock cycle i are denoted, respectively, by x_i and y_i. When a circuit is run in state M_{i-1} on input x_i, the computation will result in a wire assignment $\mathcal{W}_i$; the circuit will output y_i and the memory gates will be in a new state M_i. We will denote this by $(y_i, M_i, \mathcal{W}_i) \Leftarrow C[M_{i-1}](x_i)$.

2.1 Leakage-Resilient Circuit Transformation

In this work, we construct a circuit transformation that takes as input a circuit and outputs a functionally equivalent, and yet, leakage-resilient circuit. Our definition generalizes the notion of a private transformation from Ishai, Sahai and Wagner [19]. For readers familiar with the model of Ishai et al., we note that the main difference is that whereas they speak of a "t-private transformation" that is secure against observers who can access at most t wires, we consider the general notion of a "$\mathcal{L}$-secure transformation" that is secure against observers who can evaluate any leakage function f within a class $\mathcal{L}$. One can recover the definition of Ishai et al. from our definition by simply letting $\mathcal{L}$ be the class of functions that output a subset of their input bits.

In order to understand our definition, it helps to keep the following scenario in mind. Imagine a circuit that has a secret stored within it (possibly in an encoded form) and it uses the secret together with a (public) input to come up with an output; the encoding of the secret itself may get modified during the computation. For example, the circuit may implement a block cipher or the RSA signing algorithm, where the keys are secret. An adversarial observer

[4] Note that `copy` and $\texttt{const}_\alpha$ gates do not count towards the depth d or the size s.

[5] Formally, our notion of a stateful circuit is essentially the same as the one in [19].

(who we denote OBS) gets to interact with the circuit and the measurement apparatus by iterating the following process polynomially many times, in an adaptive manner: choosing an input for the circuit and a leakage function for the measurement apparatus, and receiving the output of the circuit on the chosen input and the physical leakage from the measurement apparatus. We would like to make sure that the ability to observe physical leakage does not help the observer: that is, the observer learns nothing more about the state of the circuit from the leakage than it could have learnt from input-output access.

Circuit Transformer. A circuit transformer TR takes as input a security parameter t, a circuit C, and an initial state M_0 and produces a new circuit $\widehat{C}$ and new initial state $\widehat{M_0}$.[6] We require the transformer to be *sound*: for all C and M_0, $C[M_0]$ should behave identically to $\widehat{C}[\widehat{M_0}]$. By "behave identically" we mean that for any number of clock cycles q and any set of inputs $x_1, x_2, \ldots, x_q$ (one for each clock cycle) the distribution of the outputs $y_1, y_2, \ldots, y_q$ is the same for C starting at state M_0 and $\widehat{C}$ starting at state $\widehat{M_0}$.

Security. We want to ensure that the transformed circuit leaks no useful information to an observer other than what the observer could have obtained by input-output access to the circuit's functionality. We define an $(\mathcal{L}, \tau, q)$-*observer* OBS to be an algorithm that: [7]

- Queries the circuit q times with inputs x_i, and receives the outputs y_i.
- For each execution of the circuit (say, with input x_i), chooses a leakage function $f \in \mathcal{L}$, and obtains $f(\mathcal{W}_C(x_i))$. That is, the leakage function f takes as input the circuit's wire assignment on input x_i, and outputs the resulting leakage.
- Runs for at most τ steps (not including the computation by the leakage function itself).

The observer makes the choice of which leakage function to use in a particular execution *adaptively*, depending on all the information it has received so far. To formalize that such an observer learns nothing useful, we show the existence of a simulator SIM, and prove that anything the observer learns can also be learned by SIM which only sees inputs and outputs of the circuit.

Consider the following two experiments that start with some circuit C in state M_0, and allow it to run for q iterations. In both experiments, we assume that OBS and SIM are stateful, namely, they remember their state from one invocation to the next.

[6] Throughout this paper, we use the hat notation $\hat{\Box}$ (reminiscent of the proverbial "tinfoil hat") to designate circuit or components that are transformed for leakage-resilience.

[7] The number of observations q, the observer's running time τ, and various other running times and success probabilities are all parameterized by a security parameter t, which is given as input to the transformation TR. For readability, we will omit t from most of our discussion.

```
$\mathsf{Exp}^{\mathrm{real}}_{\mathsf{TR}}(\mathsf{OBS}, \mathcal{L}, q, C, M_0)$:
$(\widehat{C}, \widehat{M}_0) \leftarrow \mathsf{TR}(C, M_0)$
$(x_1, f_1) \leftarrow \mathsf{OBS}(\widehat{C})$, with $f_1 \in \mathcal{L}$
For $i = 1$ to $q - 1$
    $(y_i, \widehat{M}_i, \mathcal{W}_i) \Leftarrow \widehat{C}[\widehat{M}_{i-1}](x_i)$;
    $(x_{i+1}, f_{i+1}) \leftarrow \mathsf{OBS}(y_i, f_i(\mathcal{W}_i))$
$(y_q, M_q, \mathcal{W}_q) \Leftarrow \widehat{C}[\widehat{M}_{q-1}](x_q)$;
Return output of $\mathsf{OBS}(y_q, f_q(\mathcal{W}_q))$.
```

```
$\mathsf{Exp}^{\mathrm{sim}}_{\mathsf{TR}}(\mathsf{SIM}, \mathsf{OBS}, q, C, M_0)$:
$(\widehat{C}, \widehat{M}_0) \leftarrow \mathsf{TR}(C, M_0)$
$(x_1, f_1) \leftarrow \mathsf{OBS}(\widehat{C})$, with $f_1 \in \mathcal{L}$
For $i = 1$ to $q - 1$
    $(y_i, M_i) \leftarrow C[M_{i-1}](x_i)$
    $\Lambda_i \leftarrow \mathsf{SIM}(x_i, y_i, f_i)$, with $\Lambda_i$ being the leakage
    $(x_{i+1}, f_{i+1}) \leftarrow \mathsf{OBS}(y_i, \Lambda_i)$
$(y_q, M_q) \leftarrow C[M_{q-1}](x_q)$;
$\Lambda_q \leftarrow \mathsf{SIM}(x_q, y_q, f_q)$
Return output of $\mathsf{OBS}(y_q, \Lambda_q)$.
```

The definition below says that the transformed circuit is leakage-resilient if the outputs of the two experiments above are indistinguishable.

Definition 1. *Let $\mathcal{L}$ be a class of circuits, and let $\tau = \tau(t)$, $\tau' = \tau'(t)$, $q = q(t)$ and $\epsilon = \epsilon(t)$ be functions of the security parameter t. A circuit transformer* TR *is said to be $(\mathcal{L}, \tau, \tau', q, \epsilon)$-secure if for every $(\mathcal{L}, \tau, q)$-observer* OBS*, there is a simulator* SIM *that runs in time τ' such that for all circuits C and all initial states M_0,*

$$\left| \Pr[\mathsf{Exp}^{\mathrm{real}}_{\mathsf{TR}}(\mathsf{OBS}, \mathcal{L}, q, C, M_0) = 1] - \Pr[\mathsf{Exp}^{\mathrm{sim}}_{\mathsf{TR}}(\mathsf{SIM}, \mathsf{OBS}, q, C, M_0) = 1] \right| \leq \epsilon,$$

where the probabilities are taken over all the coin tosses involved in the experiments. We refer to a circuit transformer being $\mathcal{L}$-secure, as a shorthand for saying that it is $(\mathcal{L}, \mathsf{poly}(t), \mathsf{poly}(t), \mathsf{poly}(t), \mathsf{negl}(t))$-secure in the above sense.

Remark. We note that a stronger result is obtained when $\mathcal{L}$, τ and q are as large as possible (as it allows for more leakage functions, and stronger observers), when τ' is as close as possible to τ, and when the distinguishing advantage ϵ is as small as possible (because either of these indicate a tighter simulation).

3 Circuit Transformation from Linear Secret-Sharing

Our main result states that if there exists a linear encoding scheme for elements of any field $\mathcal{K}$ (taking a single element to t elements) for which encodings of any two values are indistinguishable by functions in a class $\mathcal{L}$, then there exists a circuit transformation that is secure against a slightly less powerful leakage class $\mathcal{L}_{\mathsf{TR}}$. (Jumping ahead, we remark that the leakage class $\mathcal{L}$ is essentially the same as the class $\mathcal{L}_{\mathsf{TR}}$ "augmented with" a depth-3 circuit of size $O(t^2)$).

We now describe the main elements in the circuit transformation.

Encoding for the wires. Our transformation can be based on any *linear encoding scheme* $\Pi = (\mathsf{Enc}, \mathsf{Dec})$, which maps a single element of $\mathcal{K}$ to a vector in $\mathcal{K}^t$ and back. In the simplest case of $\mathcal{K} = \mathsf{GF}(2)$, an encoding of a bit x is a random string of t bits whose exclusive-or is x. More generally, for security parameter t, a linear encoding scheme Π is defined by a *decoding vector* $\boldsymbol{r} = (r_1, \ldots, r_t) \in \mathcal{K}^t$ and the decoding function $\mathsf{Dec} : (y_1, \ldots, y_t) \mapsto \sum_i y_i r_i = \boldsymbol{r}^{\top}\boldsymbol{y}$.

Enc is a (probabilistic) algorithm that, on input x, chooses uniformly at random an element of $\mathsf{Dec}^{-1}(x)$.

Linear encoding schemes include the aforementioned parity encoding, as well as any threshold or non-threshold linear secret sharing scheme, e.g., [32,6,20].

We need the notion of leakage-indistinguishability of an encoding scheme which, roughly speaking, formalizes what it means for an encoding of two values to be indistinguishable in the presence of leakage. In conjunction with formalizing this notion, let us first introduce a more general definition that speaks about leakage-indistinguishability of two distributions.

Definition 2. *Two distributions X and Y are said to be* $(\mathcal{L}, p, \tau, \epsilon)$-leakage-indistinguishable*, if for any observer* OBS*, running in time τ and making at most p queries to its oracle where each query f is a function in $\mathcal{L}$,*

$$|\Pr[x \leftarrow X; \mathsf{OBS}^{\mathsf{Eval}(x,\cdot)}(1^t) = 1] - \Pr[y \leftarrow Y; \mathsf{OBS}^{\mathsf{Eval}(y,\cdot)}(1^t) = 1| \leq \epsilon,$$

where $\mathsf{Eval}(x, \cdot)$ *takes as input a leakage function f and outputs $f(x)$.*

We say that an encoding scheme *Π is* $(\mathcal{L}, p, \tau, \epsilon)$-leakage-indistinguishable *if for any $a, b \in \mathcal{K}$ the two distributions* $\mathsf{Enc}(a)$ *and* $\mathsf{Enc}(b)$ *are $(\mathcal{L}, p, \tau, \epsilon)$-leakage-indistinguishable. If τ =* $\mathsf{poly}(t)$ *and ϵ =* $\mathsf{negl}(t)$*, then we abbreviate this to $(\mathcal{L}, p)$-leakage-indistinguishable.*

Opaque gates. In our scheme, the transformed circuit $\widehat{C}$ is built of the same gate types as the original circuit, with the addition of a new *opaque* gate denoted $\mathcal{O}$. As mentioned in the introduction, the $\mathcal{O}$ gate has *no inputs*, and outputs an encoding sampled from the distribution $\mathsf{Enc}(0)$. Crucially, while the wires coming out of this gate can be observed by the leakage function, we assume that its internals do not leak (we show how to somewhat relax this condition in the full version). For the case of $\mathcal{K} = \mathsf{GF}(2)$ our leak-free component can be implemented by a circuit that works as follows: generate t random bits $b_0, \ldots, b_{t-1}$ and output the bits $c_i := b_i \oplus b_{i+1 \bmod t}$ for $0 \leq i \leq t-1$.

As mentioned in the introduction, our leak-free component is minimal in many senses; the only sense in which it is not minimal is that its size is proportional to the security parameter t. Improving on this is left as an important open problem.

We now state our main theorem. The rest of this section describes the transformation, and the next section contains an overview of the proof of security.[8]

Theorem 2. *Let t be the security parameter, and let $\mathcal{L}_{\mathsf{TR}}$ be some class of leakage functions. If there exists a linear encoding scheme Π that is $(\mathcal{L}_\Pi, 2)$-leakage-indistinguishable, then there exists a circuit transformation* TR *that is $\mathcal{L}_{\mathsf{TR}}$-secure provided that:*

$$\mathcal{L}_\Pi \supseteq \mathcal{L}_{\mathsf{TR}} \circ \mathsf{SHALLOW}(3, O(t^2))$$

The transformation increases the size of each multiplication gate by a factor of $O(t^2)$ and the rest of the circuit by a factor of $O(t)$, where the constants hidden in $O(\cdot)$ are small.

[8] A complete statement of the theorem keeps track of other parameters such as the running-time of the observer as well as the simulator, and the distinguishing advantage. We postpone the more detailed theorem statement to the full version.

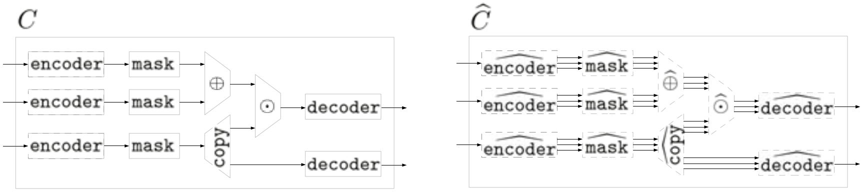


Fig. 1. Example of a circuit C for the function $(a, b, c) \mapsto ((a \oplus b) \odot c, c)$, and the corresponding transformed circuit $\widehat{C}$. Three parallel lines denote encoding (t wires). Dashed borders indicate a gadgets, whose internal wires leak. Note that in C, the special gates `encoder`, `decoder`, `mask` and `copy` are just the identity and are present for notational convenience.

3.1 The Transformation for Stateless Circuits

We will first describe our transformation for circuits without any memory gates, which we call, like in [19], *stateless circuits*. We then show how to extend the transformation to general (i.e., stateful) circuits.

Given a stateless circuit C, our transformation TR produces the transformed circuit $\widehat{C}$ as follows (see Figure 1 for an example). Each wire w in C is replaced by a *wire bundle* in $\widehat{C}$, consisting of t wires $\boldsymbol{w} = (w_1, \dots, w_t)$, that carry an encoding of w. Each gate is transformed into a *gadget*, built out of gates, which takes encodings and outputs encodings. Crucially, note that the internals of these gadgets may leak. The gadgets themselves are described in Figure 2.

```
Transformation $c \leftarrow a \odot b \;\Rightarrow\; \boldsymbol{c} \leftarrow \boldsymbol{a}\widehat{\odot}\boldsymbol{b}$:
    Compute the $t \times t$ matrix
        $B \leftarrow \boldsymbol{a}\boldsymbol{b}^{\mathsf{T}} = (a_i b_j)_{1 \le i,j \le t}$ using $t^2$ $\odot$ gates
    Compute the $t \times t$ matrix $S$
        where each column of $S$ is output by $\mathcal{O}$
    $U \leftarrow B + S$ (using $t^2$ $\oplus$ gates)
    Decode each row of $U$ using $t - 1$ $\oplus$ gates,
        $t$ $\odot$ gates, and $t$ $\mathtt{const}_\alpha$ gates
        to obtain $\boldsymbol{q} \leftarrow U\boldsymbol{r}$,
        where $\boldsymbol{r}$ is the decoding vector
        (it does not matter how this decoding is
        performed as long as there are $O(t)$ wires
        in the decoding subcircuit and each one
        carries some linear combination of the
        wires being decoded, plus possibly a
        constant)
    $\boldsymbol{o} \leftarrow \mathcal{O}$
    $\boldsymbol{c} \leftarrow \boldsymbol{q} + \boldsymbol{o}$ (using $t$ $\oplus$ gates)
```

```
Transformation $c \leftarrow \$ \;\Rightarrow\; \boldsymbol{c} \leftarrow \widehat{\$}$:
    $c_i \leftarrow \$$     for $i \in [1, t]$
    Output $\boldsymbol{c}$
```

```
Transformation $c \leftarrow a \oplus b \;\Rightarrow\; \boldsymbol{c} \leftarrow \boldsymbol{a}\widehat{\oplus}\boldsymbol{b}$
            (or $c \leftarrow a \ominus b \;\Rightarrow\; \boldsymbol{c} \leftarrow \boldsymbol{a}\widehat{\ominus}\boldsymbol{b}$):
    $\boldsymbol{q} \leftarrow \boldsymbol{a} + \boldsymbol{b}$ (or $\boldsymbol{q} \leftarrow \boldsymbol{a} - \boldsymbol{b}$)
        using $t$ $\oplus$ (or $\ominus$) gates
    $\boldsymbol{o} \leftarrow \mathcal{O}$
    $\boldsymbol{c} \leftarrow \boldsymbol{q} + \boldsymbol{o}$ (using $t$ $\oplus$ gates)
```

```
Transformation $b \leftarrow \mathtt{mask}(a) \;\Rightarrow\; \boldsymbol{b} \leftarrow \widehat{\mathtt{mask}}(\boldsymbol{a})$
    $\boldsymbol{o} \leftarrow \mathcal{O}$
    $\boldsymbol{b} \leftarrow \boldsymbol{a} + \boldsymbol{o}$ (using $t$ $\oplus$ gates)
```

```
Transformation $a \leftarrow \mathtt{const}_\alpha \;\Rightarrow\; \boldsymbol{a} \leftarrow \widehat{\mathtt{const}}_\alpha$,
        for any $\alpha \in \mathcal{K}$
    Let $\boldsymbol{\alpha}$ be a fixed arbitrary encoding of $\alpha$.
    $\boldsymbol{o} \leftarrow \mathcal{O}$
    $\boldsymbol{a} \leftarrow \boldsymbol{\alpha} + \boldsymbol{o}$ (using $t$ $\oplus$ gates)
```

```
Gadget $(\boldsymbol{b}, \boldsymbol{c}) \leftarrow \widehat{\mathtt{copy}}(\boldsymbol{a})$
    $\boldsymbol{o_1} \leftarrow \mathcal{O}$, $\boldsymbol{o_2} \leftarrow \mathcal{O}$
    $\boldsymbol{b} \leftarrow \boldsymbol{a} + \boldsymbol{o_1}$ (using $t$ $\oplus$ gates)
    $\boldsymbol{c} \leftarrow \boldsymbol{a} + \boldsymbol{o_2}$ (using $t$ $\oplus$ gates)
```

Fig. 2. Gadgets used in the stateless circuit transformation TR

Since our gadgets operate on encoded values, $\widehat{C}$ needs to have a subcircuit at the beginning that encodes the inputs and another subcircuit at the end that decodes the outputs. However, in our proofs, we want to be able to also reason about transformed circuits without encoding and decoding. Thus, we do not require that every transformed circuit $\widehat{C}$ should have such encoding and decoding. Instead, we introduce artificial input and output gates that can be part of C for syntactic purposes. If such gates are present (as they would be on any "complete" circuit that one would actually wish to transform), then $\widehat{C}$ will include input encoding and output decoding. If they are not, then $\widehat{C}$ will operate on already encoded inputs and produce encoded outputs.

More precisely, if we wish for $\widehat{C}$ to include input encoding and output decoding, then the circuit C given to TR must have two special gates in sequence on every input wire: an $\texttt{encoder}$ gate followed by a $\texttt{mask}$ gate, both of which are simply the identity. Also, on every output wire there must be a special $\texttt{decoder}$ gate, which is also the identity. These special gates must not appear anywhere else in C. In $\widehat{C}$ each $\texttt{encoder}$ gate is replaced by an $\widehat{\texttt{encoder}}$ gadget which performs encoding (see below), each $\texttt{decoder}$ gate is replaced by a $\widehat{\texttt{decoder}}$ gadget that performs decoding (see below), and each $\texttt{mask}$ gate is replaced by a $\widehat{\texttt{mask}}$ gadget (that is needed for security and is described in Figure 2).

The $\widehat{\texttt{encoder}}$ gadget takes an input $a \in \mathcal{K}$ and outputs an encoding (i.e., a wire bundle) $\boldsymbol{a} \in \mathcal{K}^t$ of a. The encoding can be chosen arbitrarily from the support of $\mathsf{Enc}(a)$: $\boldsymbol{a} = (r_1^{-1}a, 0, \ldots, 0)$. The $\widehat{\texttt{decoder}}$ gadget takes an encoding (i.e., a wire bundle) $\boldsymbol{a} \in \mathcal{K}^t$ of a and outputs $a \leftarrow \mathsf{Dec}(\boldsymbol{a})$. This is computed by a decoding circuit with just $\texttt{const}_\alpha$, $\oplus$, and $\odot$ gates. The operation of all the gadgets is described in 2. For the soundness of our transformation, we refer the reader to the full version.

Incidentally, observe that because every gadget other than $\widehat{\texttt{encoder}}$ or $\widehat{\texttt{decoder}}$ ends with a masking by an output of $\mathcal{O}$,[9] and wire bundles do not fan-out (instead, they go through the $\widehat{\texttt{copy}}$ gadget), each connecting wire bundle carries an encoding of its value that is chosen *uniformly and independently of all the wires in the transformed circuit.* This fact, together with the construction of the gadgets, is what enables the simulation.

Handling Stateful Circuits. To augment the above stateless circuit transformation to a full circuit transformation, we have to explain how to transform the initial state M_0 and what to do with each memory gate. The initial state is replaced by a randomly chosen encoding $\mathsf{Enc}(M_0)$. Each memory gate is replaced by a gadget that consists of t memory gates to store the encoding followed by a $\widehat{\texttt{mask}}$ gadget to guarantee re-randomization of the state.[10]

[9] One can instead define the basic gadgets as not including this masking with $\mathcal{O}$, and instead place a $\texttt{mask}$ gate on every wire. The resulting transformation is similar.

[10] Masking the output of the memory gadget has two reasons: first, we want to allow the total leakage to be much larger than the size of the state, and second, we want to allow the adversary to choose leakage functions adaptively.

4 Proof of Security

Conceptually, the proof of security for the circuit transformation in Section 3 proceeds in two steps. First, consider a mental experiment where each gadget in the transformed circuit $\widehat{C}$ is *perfectly opaque*. Namely, the only wires that the observer OBS can "see" are the *external wires* of the gadgets that connect the output of a gadget to the input of another gadget (these are exactly the wires that carry encodings of the values in the circuit C). The wires internal to the gadgets are off-limits to OBS. Once in this (imaginary) world, we use the first key property of our gadgets, namely

Re-randomizing: The output of each gadget in $\widehat{C}$ is a *uniformly random* encoding of the output of the corresponding gate in C.[11]

Letting $w_1, \ldots, w_m$ denote the values of the wires in C, the re-randomizing property says that the wire-bundles in $\widehat{C}$ that are external to the gadgets are distributed like $(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m)$ where the $\boldsymbol{w}_i \leftarrow \mathsf{Enc}(w_i)$ are *random and independent* encodings of the bit w_i.

The simulator does not know the value w_i (because it does not know the secret state in the circuit), but will simulate it with a random encoding of a random value w'_i. Now, the leakage indistinguishability of the encoding scheme tells us that given the leakage from any of these encodings (individually), it is hard to tell if the underlying value is w_i or w'_i. By a hybrid argument, the same holds for a vector of *independent* encodings of m values as well, which is what the simulator uses.

Before we declare victory (in this imaginary world), let us look a little more closely at the hybrid argument. At each hybrid step, we will prove indistinguishability by a reduction to the security of the encoding scheme. In other words, we will show by reduction that if OBS equipped with functions from $\mathcal{L}_{\mathsf{TR}}$ can distinguish two hybrid wire distributions, then some adversary OBS_Π, equipped with functions from a slightly larger class $\mathcal{L}_\Pi$, can distinguish two encodings. Given an encoding, our reduction will need to fake the remaining wires of the circuit and give them as input to the function from $\mathcal{L}_{\mathsf{TR}}$.

Efficiency of such a reduction is particularly important. If OBS specifies a leakage function $f \in \mathcal{L}_{\mathsf{TR}}$ for $\widehat{C}$, then OBS_Π will specify its own leakage function f_Π for the encoding and return its result to OBS. This leakage function f_Π has to fake (in a way that will look real to f and OBS) all the wires of $\widehat{C}$ before it can invoke f. At the same time, f_Π should not be much more complex than f, because our result is more meaningful when difference between the power of $\mathcal{L}_\Pi$ and the power of $\mathcal{L}_{\mathsf{TR}}$ is smaller. The main trick is for OBS_Π to hardwire as much as possible into f_Π, so that when f_Π observes the encoding, it has to do very little work before it can invoke f. In fact, in this imaginary situation,

[11] Of course, given the values of the internal wires of the gadgets as well, the outputs of the gadgets are not independent encodings any more. But, note that we are still in the mental experiment where the observer does not get to see the internals of the gadgets.

all the remaining wires can be hardwired into f_Π because of independence of encodings, so f_Π has to simply invoke f on its input wires and hardwired values.

The second step in the proof is to move from the mental experiment to the real world, where the internals of the gadgets also leak. Unlike in the mental experiment, where the values of all wire bundles were independent, values of wires inside a gadget are correlated to its input and output wire bundles. Thus, they cannot be hardwired into f_Π. Nor can they be computed by f_Π, because the complexity of the gadgets is too high.

Handling this problem requires invoking the second key property of the gadgets, namely:

> *Reconstructibility:* We say that a pair of strings (X, Y) is *plausible for* $\widehat{G}$ if $\widehat{G}$ might output Y on input X. For every gadget $\widehat{G}$, there exists a distribution $\mathsf{REC}_{\widehat{G}}$ over low-complexity functions $\mathcal{R}$, which takes as input X, Y and produces a *simulated distribution* of the internal wires of $\widehat{G}$. If for any plausible X, Y this distribution is $(\mathcal{L}, \tau, \epsilon)$-leakage-indistinguishable from the actual distribution of the internal wires of $\widehat{G}$ (conditioned on X and Y), then we say that $\widehat{G}$ is $(\mathcal{L}, \tau, \epsilon)$-reconstructible by $\mathcal{R}$, and call $\mathsf{REC}_{\widehat{G}}$ a $(\mathcal{L}, \tau, \epsilon)$-reconstructor.

In the following we will often omit the parameters τ and ϵ.

We use this property to handle leakage from gadgets. Given reconstructors for each single gadget we can show that a transformed circuit that is encoding-based (i.e. the gadgets operate on encodings) and composed of reconstructible gadgets is secure according to Definition 1. On a high-level we will replace each gadget with its reconstructor in addition to replacing connecting wire bundles with random encodings. The proof that the simulation is indistinguishable requires first doing a hybrid argument over gadgets as they are replaced by reconstructors one-by-one, and then modifying the hybrid argument over the wires described above. In the hybrid argument over the wires, f_Π can have hardwired values for every wire in the circuit except the gadgets connected to the challenge encoding, which will be computed by f_Π using the low-complexity function given by the reconstructor. This allows for a very efficient reduction. The formal statement of the composition lemma is given in Lemma 3.

Let us now move on to building reconstructors for two simple gadgets.

4.1 Reconstructors for Single Gadgets

We present proof sketches for the reconstructibility of the $\widehat{\oplus}$ and $\widehat{\odot}$ gadget.

Lemma 1 ($\widehat{\oplus}$ and $\widehat{\ominus}$ gadgets are reconstructible). *For any class of circuits $\mathcal{L}$, the $\widehat{\oplus}$ and $\widehat{\ominus}$ gadgets are $(\mathcal{L}, \infty, 0)$-reconstructible, where the reconstructor can be computed by $\mathsf{SHALLOW}(2, O(t))$.*

Proof. In this sketch we will do the proof only for $\widehat{\oplus}$. The reconstructor $\mathsf{REC}_{\widehat{\oplus}}$ is the distribution whose only support is the following circuit $R_{\widehat{\oplus}}$. On inputs (X, Y) where $X = (\boldsymbol{a}, \boldsymbol{b})$ (i.e., the desired input of the $\widehat{\oplus}$ gate), and $Y = (\boldsymbol{c})$ (i.e., its desired output), $R_{\widehat{\oplus}}$ assigns the wires of $\widehat{\oplus}$ to $\boldsymbol{q} \leftarrow \boldsymbol{a} \oplus \boldsymbol{b}$ and $\boldsymbol{o} \leftarrow \boldsymbol{c} \ominus \boldsymbol{q}$.

If X, Y are chosen as in the definition of a reconstructor, then the resulting output of $R_{\widehat{\oplus}}$ is identically distributed to the wire distribution $\mathcal{W}_{\widehat{\oplus}}(X|Y)$, since in both cases $\boldsymbol{o}$ takes the only possible consistent value $\boldsymbol{o} \leftarrow \boldsymbol{c} \ominus \boldsymbol{q}$. Notice that $R_{\widehat{\oplus}}$ can be computed by a circuit of depth 2 because on inputs X, Y it first computes $\boldsymbol{q} \leftarrow \boldsymbol{a} \oplus \boldsymbol{b}$ and based on that $\boldsymbol{o} \leftarrow \boldsymbol{c} \ominus \boldsymbol{q}$. The $\ominus$ and $\oplus$ gates above operate only on single field elements, so $R_{\widehat{\oplus}}$ requires $O(t)$ size. □

Let us now give a proof sketch for the $\widehat{\odot}$ reconstructor. Notice that the main technical difficulty is the fact that our simulation has to be shallow whereas the real $\widehat{\odot}$ gadget is already a deep circuit. In the following, let $\mathcal{K} = \mathsf{GF}(2)$.

Lemma 2 ($\widehat{\odot}$ is reconstructible). *Let $\mathcal{L}_{\widehat{\odot}}$ be a class of functions, and assume that the encoding Π is $\mathcal{L}_{\Pi}$-leakage-indistinguishable, where $\mathcal{L}_{\Pi} \supseteq \mathcal{L}_{\widehat{\odot}} \circ \mathsf{SHALLOW}(2, O(t^2))$. Then, the $\widehat{\odot}$ gadget is $\mathcal{L}_{\widehat{\odot}}$-reconstructible, where the reconstructor can be computed by $\mathsf{SHALLOW}(2, O(t^2))$.*

Proof (sketch). The reconstructor $\mathsf{REC}_{\widehat{\odot}}$ takes as inputs (X, Y), where $X = (\boldsymbol{a}, \boldsymbol{b})$, and $Y = (\boldsymbol{c})$ and is defined as follows:

1. Sample U uniformly from $\mathcal{K}^{t \times t}$ and compute the values on the wires in the subcircuits for the computation of $\boldsymbol{q}$. Hard-wire the results as $R_{\widehat{\odot}}$'s outputs.
2. On input X, $R_{\widehat{\odot}}$ computes the matrix $B \leftarrow (a_i \odot b_j)_{i,j}, i, j \in [1, t]$ and outputs it as part of the wire assignment.
3. $R_{\widehat{\odot}}$ computes $S \leftarrow B - U$ and $\boldsymbol{o} \leftarrow \boldsymbol{c} - \boldsymbol{q}$.

$\mathsf{REC}_{\widehat{\odot}}$ has size $O(t^2)$ (because it needs to compute matrices B and S) and depth 2, because S is computed from B, that in turn has been computed from the inputs.

It remains to show that the distribution $R_{\widehat{\odot}}(X, Y)$ produced by the reconstructor and the actual wire distribution $\mathcal{W}_{\widehat{\odot}}(X|Y)$ are leakage-indistinguishable by leakage functions in $\mathcal{L}_{\widehat{\odot}}$. Since U is computed as $B + S$ it suffices to show that S can be replaced by a matrix sampled uniformly at random from $\mathcal{K}^{t \times t}$.

We prove it by a hybrid argument and define hybrids $\mathcal{W}^{\ell}_{\widehat{\odot}}(X|Y)$ ($\ell \in [0, t]$) as $\mathcal{W}_{\widehat{\odot}}(X|Y)$, except that for the first ℓ columns of S the elements are drawn uniformly from $\mathcal{K}$. We show the leakage-indistinguishability between two consecutive hybrids by a reduction to the encoding leakage-indistinguishability. As part of this reduction we build the observer OBS_{Π} that runs $\mathsf{OBS}_{\widehat{\odot}}$ and has to answer its leakage queries $f_{\widehat{\odot}} \in \mathcal{L}_{\widehat{\odot}}$. OBS_{Π} runs $f_{\widehat{\odot}}$ as part of its own leakage function $f_{\Pi} \in \mathcal{L}_{\Pi}$. However, f_{Π} only expects a single target encoding $\boldsymbol{e}$ as input, whereas functions from $\mathcal{L}_{\widehat{\odot}}$ expect a full wire assignment for $\widehat{\odot}$. Thus, before f_{Π} runs $f_{\widehat{\odot}}$, a wire simulator f_S, computes a wire assignment for $\widehat{\odot}$ given only the target encoding $\boldsymbol{e}$. To keep the reduction tight (and our result meaningful), f_S has to be very simple; i.e. we use the input $\boldsymbol{e}$ as little as possible and hard-wire most of the values of the wires of $\widehat{\odot}$ into f_S. For any X, Y:

1. From X compute $B = (a_i b_j)_{i,j \in [1,t]}$ and hard-wire $\boldsymbol{a}, \boldsymbol{b}, B$ into f_S.
2. Hard-wire the columns $1 \ldots \ell - 1$ to random encodings and $\ell + 1 \ldots t$ to $\mathsf{Enc}(0)$. The ℓth column is filled with the challenge encoding $\boldsymbol{e}$.

3. Hard-wire all elements of $U = B + S$ into f_S except for the ℓth column. For the ℓth column, compute for each $i \in [1, t]$, the value $U_{i,\ell} \leftarrow B_{i,\ell} + e_i$.
4. The wires in the decoding sub-circuits to compute $\boldsymbol{q}$ from U carry the $\oplus$ of some row $\{U_{i,j}\}_j$. If a wire in the sub-circuit does not depend on $U_{i,\ell}$ (i.e., the input to f_S), then pre-compute its value and hard-wire the intermediate result. On the other hand, if it depends on $U_{i,\ell} = B_{i,\ell} + e_i$, then pre-compute a partial sum except the term that depends on e_i and hard-wire the result. On input $\boldsymbol{e}$, f_S computes the missing outputs by $\oplus$-ing the relevant parts of $\boldsymbol{e}$.
5. With fixed Y and $\boldsymbol{q}$ from (3) compute $\boldsymbol{o} \leftarrow Y - \boldsymbol{q}$ and output it.

It is not difficult to check that f_S outputs a valid wire assignment for $\widehat{\odot}$ that is either distributed as $\mathcal{W}^{\ell-1}_{\widehat{\odot}}(X|Y)$ or $\mathcal{W}^{\ell}_{\widehat{\odot}}(X|Y)$. If $\boldsymbol{e}$ is drawn from $\mathsf{Enc}(0)$, then the ℓth column of S is assigned an encoding drawn from $\mathsf{Enc}(0)$. Since all the other wires are computed honestly using either hard-wired values or the input $\boldsymbol{e}$, $f_S(\mathsf{Enc}(0))$ and $\mathcal{W}^{\ell-1}_{\widehat{\odot}}(X|Y)$ are distributed identically. If $\boldsymbol{e} \leftarrow \mathsf{Enc}(x)$, for $x \in \mathcal{K}$, then the ℓth column of S is assigned an encoding drawn from $\mathsf{Enc}(x)$, hence, we get that $f_S(\mathsf{Enc}(x))$ and $\mathcal{W}^{\ell}_{\widehat{\odot}}(X|Y)$ are distributed identically. Since f_S needs to compute the ℓth column of U, the values in the decoding sub-circuits, and from $\boldsymbol{q}$ the value of $\boldsymbol{o}$, $f_S \in \mathsf{SHALLOW}(2, O(t^2))$. Together with the t hybrids, we get that $\mathcal{W}_{\widehat{\odot}}(X|Y)$ and $R_{\widehat{\odot}}(X,Y)$ are $(\mathcal{L}_{\widehat{\odot}}, t\epsilon)$-leakage-indistinguishable, if Π is $(\mathcal{L}_\Pi, \epsilon)$-leakage-indistinguishable (where $\mathcal{L}_\Pi \supseteq \mathcal{L}_{\widehat{\odot}} \circ \mathsf{SHALLOW}(2, O(t^2))$). □

The rerandomizing property of the simple gadgets follows immediately from the fact that every gadget's output is masked by the output of $\mathcal{O}$.

4.2 Security of Full Circuit Transformation

Until now we showed that individual gadgets are re-randomizing, and reconstructible. The following central lemma, that is proved in the full version, states how to compose reconstructors for single gadgets to yield a reconstructor for the entire circuit.

Lemma 3 (Composition Lemma). *Let $\mathcal{L}_{\widehat{C}}$ be some set of leakage functions and $\epsilon_\Pi > 0, \tau_\Pi > 0, t > 0$. Let Π be $(\mathcal{L}_\Pi, \tau_\Pi, \epsilon_\Pi)$-leakage-indistinguishable. Let C be a stateless circuit of size s, without* `encoder` *or* `decoder` *gates with k_{I} inputs and k_{O} outputs. Then the transformed circuit $\widehat{C}$ is rerandomizing and $(\mathcal{L}_{\widehat{C}}, \tau_{\widehat{C}}, \epsilon_{\widehat{C}})$-reconstructible by* $\mathsf{SHALLOW}(2, (k_{\mathrm{I}} + k_{\mathrm{O}})O(t^2))$ *where $\mathcal{L}_\Pi = \mathcal{L}_{\widehat{C}} \circ \mathsf{SHALLOW}(3, O(t^2))$, $\epsilon_{\widehat{C}} = \epsilon_\Pi s(t+2)$, and $\tau_{\widehat{C}} = \tau_\Pi - O(st^2)$.*

There is one caveat that remains in proving security according to Definition 1: the $\widehat{\texttt{encoder}}$ and $\widehat{\texttt{decoder}}$ gadget are not reconstructible, however, the simulator can easily include them into his simulation since the inputs and outputs of these gadgets are known.

We would like to make a final remark: the circuit transformation that we discussed so far are based on any linear encoding scheme, however, the proof techniques that we introduced along the way are more general. Note that Lemma 3 relies essentially on the fact that the gate gadgets are rerandomizing and reconstructible. One can obtain an analogously result using *any* (not necessarily linear) encoding scheme and a corresponding set of sound gate gadgets that are rerandomizing and reconstructible. We refer the interested reader to the full version.

5 Security against Constant Depth Leakage

In this section, we show how to use the general circuit transformation from Section 3 to achieve security against leakage functions that can be computed by constant-depth circuits.

5.1 AC^0 Leakage

The first leakage class we consider is AC^0, the class of constant-depth, polynomial-size circuits formed out of NOT gates and unbounded fan-in AND and OR gates. Let $\mathcal{C}(d, s, \lambda)$ denote the class of AND-OR-NOT Boolean circuits with depth d, size s and λ bits of output.

The Encoding. The encoding we use in this case is the parity encoding. The (randomized) parity encoding of a bit b is a sequence of bits $(b_1, \ldots, b_t)$ which are uniformly random subject to the condition that their parity is the bit b. This encoding can be computed in many different ways, for example, as:

ENC(b): Generate bits $b_1, \ldots, b_{t-1}$ uniformly at random, and set $b_t := b \oplus \bigoplus_{i=1}^{t-1} b_i$.

Obviously, the decoding function for the parity encoding is simply the parity function, namely the function that outputs the exclusive-or of the t bits in the encoding.

The parity encoding is hard to decode for AC^0 circuits. The classical result of Håstad [18] (which builds on [1,14]), translated to our definition, states that the parity encoding of the bits 0 and 1 are indistinguishable by circuits in the class $\mathcal{C}(d, 2^{t^{1/d}}, 1)$ for any constant d. This protects against AC^0 circuits that output 1 bit. Using a recent result of Dubrov and Ishai [10, Theorem 3.4], we can protect against the circuit class $\mathcal{C}(d, e^{O(t^{\delta}/d)}, t^{1-\delta})$ for any $0 < \delta < 1$, namely AC^0 circuits that output up to $t^{1-\delta}$ bits.

We obtain the following theorem by instantiating Theorem 2 with the parity encoding, and using the above observations about the leakage-indistinguishability of the parity encoding against AC^0 circuits. The reader is referred to the full version for a tight statement and a formal proof of security.

Theorem 3. *Let t be the security parameter, and $0 < \delta < 1$, and $d \in \mathbb{N}$ be constants. Then, there exists a circuit transformation that is $\mathcal{L}_{\mathsf{AC}^0,d,\delta}$-secure where $\mathcal{L}_{\mathsf{AC}^0,d,\delta} = \mathcal{C}(d-4, e^{O(t^{\delta})/d}, t^{1-\delta})$ is the class of all Boolean AND-OR-NOT circuits of depth at most $d-4$, size at most $e^{O(t^{\delta})/d}$ and output length at most $t^{1-\delta}$.*

In particular, the theorem states that the transformation is secure against AC^0 circuits (constant depth, polynomial-size circuits) that output at most $t^{1-\delta}$ bits, *for any constant* $\delta > 0$.

5.2 $\mathsf{ACC}^0[q]$ Leakage

A natural way to extend the class of leakage functions from AC^0 to something more general is to allow the leakage function to have parity gates. Clearly, such circuits can decode the parity encoding, but are there still other linear encoding schemes that cannot be decoded by even such circuits? It turns out that such encodings indeed exist. For any integer q, let $\mathtt{MOD}_q$ be the gate that outputs 0 if the sum of its inputs is 0 modulo q, and 1 otherwise. The class $\mathcal{C}_{\mathtt{MOD}\text{-}q}(d, s, \lambda)$ is defined to be the functions computable by circuits made of $\mathtt{NOT}$ gates and unbounded fan-in $\mathtt{AND}, \mathtt{OR}$ and $\mathtt{MOD}_q$ gates, with depth d, size s and output length λ. For example, letting $q = 2$, we get the class of depth d circuits that include parity gates as well.

The encoding scheme we use in this case is the mod-q' encoding scheme, for some q' that is co-prime to q, defined analogously to the parity encoding scheme in Section 5.1. By a result of Razborov and Smolensky [31,34], for any distinct primes q' and q, the mod-q' encoding is leakage-indistinguishable for functions in the class $\mathcal{C}_{\mathtt{MOD}\text{-}q}(O(1), \mathsf{poly}(t), 1)$, i.e., $\mathsf{ACC}^0[q]$ circuits with output length 1. Since the mod-q' encoding is linear, we can apply Theorem 2 to get a secure circuit transformation.

6 Security against Noisy Leakage

So far, we considered leakage classes that are constrained in terms of their computational power and output length. In this section, we consider the noisy leakage model, where the leakage consists of the values of *all the wires* in the circuit, except that each bit is flipped with some probability $p \in [0, 1/2]$. More precisely, the class of noisy leakage functions is represented by the circuit class $\mathcal{L} = \{\mathcal{N}_p\}_{p \in [0,1/2]}$, where each circuit $\mathcal{N}_p$ is probabilistic, and is defined as follows: Let B_p be the binomial distribution with parameter p which outputs 1 with probability p and 0 otherwise. Then, $\mathcal{N}_p(\boldsymbol{x}) = \boldsymbol{x} \oplus \boldsymbol{b}$, where each bit b_i is drawn from the distribution B_p and the different b_i are independent.

Ideally, we would hope that the circuit transformation in Section 3 provides security against noisy leakage as well. However, this turns out to be false, and in fact, there is an explicit attack against the transformation in Section 3 (as well as the circuit transformation of Ishai et al. [19]) in the presence of noisy leakage, even when the noise is very small.

We outline the basic idea of the attack here. Specifically, the attack is against the construction of the multiplication gadget $\widehat{\odot}$ in Figure 2. The gadget takes as input two encodings $\boldsymbol{a}$ and $\boldsymbol{b}$ and first computes the t^2 bits $\{a_i \wedge b_j : i, j \in [t]\}$. Consider the first t bits $(a_1 \wedge b_1, \ldots, a_1 \wedge b_t)$. If $a_1 = 0$, then all these bits are 0, whereas if $a_1 = 1$, then roughly half of them are 1. Given such disparity, the observer can determine whether a_1 is 0 or 1, even if he is given a noisy version of these t bits (for any noise parameter $p < 1/2$). Proceeding in a similar way, he can reconstruct all the bits a_i, and thus the input bit a itself. The fundamental reason why this attack works is that the construction of the $\widehat{\odot}$ gadget in Figure 2 has *high input locality*, namely it accesses the input bits a large number of times.

6.1 A New Circuit Transformation against Noisy Leakage

We construct a new circuit transformation against noisy leakage. The transformation proceeds in the same way as in Section 3, except for the construction of the multiplication gadget $\widehat{\odot}$. The new construction of the multiplication gadget avoids the attack outlined below, and is constructed using a new opaque gate that we call $\mathcal{M}$ (in addition to the opaque gate $\mathcal{O}$). We stress that the opaque gate $\mathcal{M}$ that we design and use, inherits the main characteristics of the opaque gate $\mathcal{O}$ in that it is *stateless*, and *independent of the computation*. In other words, $\mathcal{M}$ simply produces samples from a fixed distribution.

In what follows, we describe the specification of the opaque gate $\mathcal{M}$ as well as the construction of the $\widehat{\odot}$ gadget.

The Opaque Gate $\mathcal{M}$. The opaque gate $\mathcal{M}$ is probabilistic, takes no inputs and operates in the following way: Sample $2t$ uniformly random 0-sharings $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_t \leftarrow \mathcal{O}$ and $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_t \leftarrow \mathcal{O}$. Let $\mathbf{R}$ and $\mathbf{S}$ be the following two $t \times t$ matrices:

$$\mathbf{R} = \begin{pmatrix} \boldsymbol{r}_1 \\ \vdots \\ \bigoplus_{j=1}^{i} \boldsymbol{r}_j \\ \vdots \\ \bigoplus_{j=1}^{t} \boldsymbol{r}_j \end{pmatrix} \text{ and } \mathbf{S} = \begin{pmatrix} \boldsymbol{s}_1 \\ \vdots \\ \bigoplus_{j=1}^{i} \boldsymbol{s}_j \\ \vdots \\ \bigoplus_{j=1}^{t} \boldsymbol{s}_j \end{pmatrix}$$

Let $R_{i,j}$ (resp. $S_{i,j}$) denote the $(i,j)^{th}$ entry of the matrix $\mathbf{R}$ (resp. $\mathbf{S}$). Define $\mathbf{R} \otimes \mathbf{S}$ to be the "inner product of the matrices $\mathbf{R}$ and $\mathbf{S}$", when written out as bit-strings. That is,

$$\mathbf{R} \otimes \mathbf{S} = \bigoplus_{i,j} R_{i,j} S_{i,j}$$

The output of the opaque gate $\mathcal{M}$ is the tuple $(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_t, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_t, u)$ where $u = \mathbf{R} \otimes \mathbf{S}^t$, the inner product of the matrices $\mathbf{R}$ and *the transpose of* $\mathbf{S}$.

The new Multiplication Gadget $\widehat{\odot}$. The operation of the multiplication gadget $\widehat{\odot}$ proceeds in two stages.

- The first stage uses a gadget $\widehat{\mathsf{mult}}$ that takes as input two encodings $\boldsymbol{a} = (a^{(1)}, \ldots, a^{(t)})$ and $\boldsymbol{b} = (b^{(1)}, \ldots, b^{(t)})$, and outputs a *longer encoding* $\boldsymbol{q} = (q^{(1,1)}, \ldots, q^{(t,t)})$ of size t^2.
- The second stage "compresses" this longer encoding into an encoding $\boldsymbol{c} = (c^{(1)}, \ldots, c^{(t)})$, using a gadget $\widehat{\mathsf{compress}}$.

We first describe how the (sub-)gadget $\widehat{\mathsf{mult}}$ works.

1. First, generate $(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_t, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_t, u) \leftarrow \mathcal{M}$.
2. Define $\boldsymbol{a}_0 := \boldsymbol{a}$ and $\boldsymbol{b}_0 := \boldsymbol{b}$. Compute the encodings $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ iteratively as follows. For $1 \le i \le t$, set

$$\boldsymbol{a}_i = \boldsymbol{a}_{i-1} \oplus \boldsymbol{r}_i, \text{ and } \boldsymbol{b}_i = \boldsymbol{b}_{i-1} \oplus \boldsymbol{s}_i$$

3. Let $a_i^{(j)}$ (resp. $b_i^{(j)}$) denote the j^{th} bit of the vector $\boldsymbol{a}_i$ (resp. $\boldsymbol{b}_i$). Output $\boldsymbol{q} = (q^{(1,1)}, \ldots, q^{(t,t)})$ defined as follows:

$$q^{(i,j)} = \begin{cases} a_1^{(1)} \wedge b_1^{(1)} \oplus u & \text{if } (i,j) = (1,1) \\ a_i^{(j)} \wedge b_j^{(i)} & \text{otherwise} \end{cases}$$

   (Note the asymmetry in the evaluation, namely the bit $a_i^{(j)}$ is multiplied with the bit $b_j^{(i)}$, where the subscript and the superscript are switched; this asymmetry is intentional, and indeed, crucial to the correctness).
4. Generate $\mathbf{z} \leftarrow \mathcal{O}_{t^2}$ (thus, $\mathbf{z}$ is a uniformly random t^2-bit string whose entries xor to 0). Output $\boldsymbol{w} := \boldsymbol{q} \oplus \mathbf{z}$.

Now, we invoke the $\widehat{\mathsf{compress}}$ gadget on the output of the $\widehat{\mathsf{mult}}$ gadget. The $\widehat{\mathsf{compress}}$ gadget takes t^2 bits $(q^{(1,1)}, \ldots, q^{(t,t)})$ and outputs t bits $(c^{(1)}, \ldots, c^{(t)})$ such that $\bigoplus_{i,j} q^{(i,j)} = \bigoplus_i c^{(i)}$. The construction of the $\widehat{\mathsf{compress}}$ gadget proceeds in the following way.

1. Split the bits $q^{(i,j)}$ into t blocks of t bits each.
2. Construct a tree of $\widehat{\oplus}$ gadgets that takes as input t *blocks* of t bits each, and outputs *one block* of t bits. (The structure of the tree can be arbitrary.) Apply the tree to the bits $q^{(i,j)}$ and call $\boldsymbol{c} = (c^{(1)}, \ldots, c^{(t)})$ the output.

The correctness of the $\widehat{\odot}$ gadget can be verified by a simple computation, and is omitted. The efficiency of implementation is practically the same as that in [3]. Namely, the transformation converts a circuit of size s into another circuit of size $O(s \cdot t^2)$, where t is the security parameter. We now outline the main ideas behind the proof of security of the new transformation against noisy leakage.

Outline of the Security Proof. As in Section 3, the proof proceeds in two steps. First, we show that the gadgets are re-randomizing and reconstructible. In other words, this says that the internals of a gadget reveal no more useful information than its inputs and output. Secondly, we apply a general version of the Composition Lemma (Lemma 3) to conclude that since each individual gadget is re-randomizing and reconstructible, the entire circuit transformation is leakage-resilient. We describe these two steps in a little more detail below.

It is easy to see that the gadgets are re-randomizing. The key difference from Section 3 is that in the proof of reconstructibility, we are not concerned about the *computational efficiency* of the reconstructor, but rather the *number of times* *the reconstructor accesses its input.* This is a consequence of the fact that the larger the number of noisy copies of an encoding $\boldsymbol{e}$ (with independent binomial noise) the observer sees, the easier it is for him to tell if $\boldsymbol{e}$ is an encoding of 0 or 1. Thus, the bulk of the effort in the design of the circuit transformation as well as the reconstructor is in ensuring that the inputs and the intermediate values are "touched" as few times as possible. The technical heart of the proof (similar to the theorems of [13,18,10] for the AC^0 case) is a lemma which states that for any constant c and any fixed vectors $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_c$, the distribution of $(\mathcal{N}_p(\boldsymbol{e} \oplus \boldsymbol{f}_1), \ldots, \mathcal{N}_p(\boldsymbol{e} \oplus \boldsymbol{f}_c))$ when $\boldsymbol{e}$ is an encoding of 0 or 1 are statistically close. We refer the reader to the full version for the design of the reconstructors and the formal proof.

Acknowledgments. We thank Debajyoti Bera, Shafi Goldwasser, Yuval Ishai, Moni Naor, Ran Raz and Ronen Shaltiel for helpful discussions, and the anonymous Eurocrypt reviewers for their detailed comments.

References

1. Ajtai, M.: $\sum_1^1$-formulae on finite structures. Annals of Pure and Applied Logic 24(1), 48 (1983)
2. Ajtai, M.: Approximate counting with uniform constant-depth circuits (1993)
3. 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)
4. 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)
5. Bernstein, D.J.: Cache-timing attacks on AES (2005), `http://cr.yp.to/papers.html#cachetiming`
6. Blakley, G.R.: Safeguarding cryptographic keys 48, 313–317 (1979)
7. Brumley, D., Boneh, D.: Remote timing attacks are practical. Comput. Netw. 48(5), 701–716 (2005)
8. Davì, F., Dziembowski, S.: Leakage-resilient storage. Cryptology ePrint Archive, Report 2009/399 (2009), `http://eprint.iacr.org/`
9. Dodis, Y., Kalai, Y.T., Lovett, S.: On cryptography with auxiliary input. In: STOC 2009, pp. 621–630. ACM, New York (2009)
10. Dubrov, B., Ishai, Y.: On the randomness complexity of efficient sampling. In: STOC 2006, pp. 711–720. ACM, New York (2006)
11. Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: FOCS 2008, pp. 293–302. IEEE Computer Society, Los Alamitos (2008)
12. 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)
13. Furst, M.L., Saxe, J.B., Sipser, M.: Parity, circuits, and the polynomial-time hierarchy. In: SFCS 1981: Proceedings of the 22nd Annual Symposium on Foundations of Computer Science, Washington, DC, USA, pp. 260–270. IEEE Computer Society, Los Alamitos (1981)
14. Furst, M.L., Saxe, J.B., Sipser, M.: Parity, circuits, and the polynomial-time hierarchy. Mathematical Systems Theory 17(1), 13–27 (1984)

15. Goldreich, O.: Towards a theory of software protection and simulation by oblivious rams. In: STOC, pp. 182–194 (1987)
16. Goldreich, O., Ostrovsky, R.: Software protection and simulation on oblivious rams. J. ACM 43(3), 431–473 (1996)
17. Goldwasser, S., Kalai, Y.T., Rothblum, G.N.: One-time programs. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 39–56. Springer, Heidelberg (2008)
18. Håstad, J.: Almost optimal lower bounds for small depth circuits. In: STOC, pp. 6–20 (1986)
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. Karchmer, M., Wigderson, A.: On span programs. In: Structure in Complexity Theory Conference, pp. 102–111 (1993)
21. 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)
22. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
23. Kuhn, M.G.: Compromising emanations: eavesdropping risks of computer displays. PhD thesis, University of Cambridge, Technical Report UCAM-CL-TR-577 (2003)
24. Micali, S., Reyzin, L.: Physically observable cryptography (extended abstract). In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 278–296. Springer, Heidelberg (2004)
25. 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)
26. Osvik, D.A., Shamir, A., Tromer, E.: Cache attacks and countermeasures: The case of AES. In: Pointcheval, D. (ed.) CT-RSA 2006. LNCS, vol. 3860, pp. 1–20. Springer, Heidelberg (2006)
27. Percival, C.: Cache missing for fun and profit. Presented at BSDCan 2005, Ottawa (2005); http://www.daemonology.net/hyperthreading-considered-harmful
28. Petit, C., Standaert, F.-X., Pereira, O., Malkin, T., Yung, M.: A block cipher based pseudo random number generator secure against side-channel key recovery. In: ASIACCS, pp. 56–65 (2008)
29. Pietrzak, K.: A leakage-resilient mode of operation. In: Joux, A. (ed.) EUROCRYPT 2009, vol. 5479, pp. 462–482. Springer, Heidelburg (2009)
30. Quisquater, J.-J., Samyde, D.: Electromagnetic analysis (EMA): Measures and counter-measures for smart cards. In: Attali, S., Jensen, T. (eds.) E-smart 2001. LNCS, vol. 2140, pp. 200–210. Springer, Heidelberg (2001)
31. Razborov, A.: Lower bounds for the size of circuits of bounded depth with basis and xor. Math. Notes of the Academy of Science of the USSR 41 (1987)
32. Shamir, A.: How to share a secret. Communications of the ACM 22(11), 612–613 (1979)
33. Shamir, A., Tromer, E.: Acoustic cryptanalysis: on nosy people and noisy machines. Presented at the Eurocrypt 2004 rump session (2004), http://tromer.org/acoustic
34. Smolensky, R.: Algebraic methods in the theory of lower bounds for boolean circuit complexity. In: STOC, pp. 77–82 (1987)
35. Standaert, F.-X., Malkin, T., 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)

Partial Fairness in Secure Two-Party Computation

S. Dov Gordon and Jonathan Katz*

Department of Computer Science, University of Maryland
{gordon,jkatz}@cs.umd.edu

Abstract. A seminal result of Cleve (STOC '86) is that *complete* fairness is impossible to achieve in two-party computation. In light of this, various techniques for obtaining *partial* fairness have been suggested in the literature. We propose a definition of partial fairness within the standard real-/ideal-world paradigm that addresses deficiencies of prior definitions. We also show broad feasibility results with respect to our definition: partial fairness is possible for any (randomized) functionality $f : X \times Y \to Z_1 \times Z_2$ at least one of whose domains or ranges is polynomial in size. Our protocols are always private, and when one of the domains has polynomial size our protocols also simultaneously achieve the usual notion of security with abort. In contrast to some prior work, we rely on standard assumptions only.

We also show that, as far as general feasibility is concerned, our results are *optimal* (with respect to our definition).

1 Introduction

In the setting of secure two-party computation, two parties run a protocol that enables each of them to learn a (possibly different) function of their inputs while preserving security properties such as privacy, correctness, input independence, etc. These requirements, and more, are traditionally formalized by comparing a real-world execution of the protocol to an *ideal world* where there is a trusted entity who performs the computation on behalf of the parties. Informally, a protocol is "secure" if for any real-world adversary $\mathcal{A}$ there is a corresponding ideal-world adversary $\mathcal{S}$ (corrupting the same party) such that an execution of the protocol in the real world with $\mathcal{A}$ is computationally indistinguishable from computing the function in the ideal world with $\mathcal{S}$.

One desirable security property is *fairness* which, intuitively, ensures that either *both parties* learn the output or else *neither party* does. In a "true" ideal world — this is the ideal world used in the multi-party setting when a majority of parties are honest — fairness is ensured since the trusted party evaluating the function provides output to both parties. Unfortunately, Cleve [10] shows that complete fairness is impossible to achieve, in general, in the two-party setting.

* Research supported by NSF CAREER award #0447075 and NSF-CCF #0830464.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 157–176, 2010.

For this reason, the usual treatment of secure two-party computation (see [18]) *weakens* the ideal world to one in which fairness is not guaranteed at all. A protocol is said to be "secure-with-abort" if it can be simulated (as described above) with respect to this less-satisfying ideal world.

Various methods for achieving *partial* fairness have been suggested; we provide an extensive discussion in Section 1.1. With the exception of [17], however, all previous work has departed from the traditional real-/ideal- world paradigm in defining partial fairness. (Indeed, addressing this deficiency is explicitly mentioned as an open problem by Goldreich [18, Section 7.7.1.1].) Furthermore, many previously suggested approaches to partial fairness only apply in specific settings (e.g., fair exchange of signatures) or under certain assumptions on the parties' inputs and auxiliary information (e.g., that inputs are chosen uniformly at random) but do not give a "general-purpose" solution that can be used for arbitrary functions computed on arbitrary inputs. Finally, much previous work on partial fairness requires strong cryptographic assumptions, e.g., regarding the precise amount of time needed to solve some problem (even using parallelism).

As noted earlier, the most desirable (but, in the two-party setting, unachievable) definition of security requires computational indistinguishability between the real world and a "true" ideal world where both parties receive output. The usual relaxation of security-with-abort [18] leaves unchanged the requirement of computational indistinguishability, but weakens the ideal world to one in which fairness is no longer guaranteed at all. Katz [23] suggested an alternate relaxation: keep the ideal world unchanged, *but relax the notion of simulation* and require instead that the real and ideal worlds be distinguishable with probability at most $\frac{1}{p} + \mathsf{negl}$, where p is some specified polynomial[1] (see Definition 1). We refer to a protocol satisfying this definition as being "$\frac{1}{p}$-secure". Cleve [10] and Moran et al. [27] show $\frac{1}{p}$-secure protocols for two-party coin tossing (where parties have no inputs), but we are not aware of any other results satisfying our definition. In particular, none of the prior approaches for achieving partial fairness yield protocols that are $\frac{1}{p}$-secure.

We propose the notion of $\frac{1}{p}$-security as a new way to approach the problem of partial fairness, and view this as an independent contribution. We also demonstrate protocols that achieve this definition for a broad class of functions. Specifically, let $f_n : X_n \times Y_n \rightarrow Z_n^1 \times Z_n^2$ be a (randomized) functionality where player 1 (resp., player 2) provides input $x \in X_n$ (resp., $y \in Y_n$) and receives output $z^1 \in Z_n^1$ (resp., $z^2 \in Z_n^2$). (Throughout this paper, n denotes the security parameter.) For arbitrary polynomial p, we show $\frac{1}{p}$-secure protocols for computing f_n as long as at least one of X_n, Y_n, Z_n^1, Z_n^2 is polynomial size (in n). Our protocols are always *private*, and when either X_n or Y_n is polynomial-size we also achieve the usual notion of security-with-abort. (Relevant definitions are

[1] This definition is similar in spirit to (but weaker than) the notion of ϵ-zero knowledge [13] and is analogous to the definition used in [19] for password-based key exchange (although there p is fixed by the size of the password dictionary). A similar idea, formalized differently and with different motivation, is also used in [1].

standard and appear in the full version of this paper.) We assume only the existence of enhanced trapdoor permutations or, more generally, oblivious transfer.

We also prove that our feasibility results are, in general, *optimal.* First, we demonstrate a deterministic, boolean function $f_n : X_n \times Y_n \to \{0,1\}$, where X_n and Y_n both have super-polynomial size, for which no protocol computing f_n can simultaneously achieve both security-with-abort and $\frac{1}{p}$-security (for $p > 4$). We also show a deterministic function $f_n : X_n \times Y_n \to Z_n$, with each of X_n, Y_n, Z_n super-polynomial in size, such that f_n cannot be $\frac{1}{p}$-securely computed for $p > 2$.

1.1 Prior Work

There is an extensive literature devoted to the problem of achieving partial fairness when an honest majority is not present, both for the case of specific functionalities like coin tossing [10,11,27] and contract signing/exchanging secrets [5,25,14,4,12,6], as well as for the case of general functionalities [29,16,3,20,15,28,17]. Prior work (with the exception of [17]; see below), however, does not consider a *simulation-based* definition within the standard real/ideal world paradigm as we do here. Moreover, to the best of our knowledge none of the previous approaches (with the exception of [10,27], that deal only with coin tossing) can be proven $\frac{1}{p}$-secure. Beyond the theoretical advantages of achieving a simulation-based notion of security, our protocols offer several concrete benefits with respect to prior solutions; these are explained in what follows.

One approach that has been suggested for achieving partial fairness is to construct a protocol where, roughly speaking, at every round both parties can recover their output using a “similar” amount of work (except in early rounds, where one party can recover their output only by investing exponential work). This idea was used in [16,12,6,28], and was formalized by Garay et al. [17] within the framework of universal composability [9]. An unsatisfying feature of this approach, no matter how it is implemented, is that the decision of whether an honest party should invest the necessary work and recover the output is not determined by the protocol, but is somehow decided “externally”; if the adversary knows how this decision is made, then it can abort at “exactly the right time” and violate fairness completely. In this approach there may also be no *a priori* polynomial bound on the honest party’s running time. This approach also seems problematic in defending against an adversary who runs in polynomial time, but has more computational power than honest parties are able to invest. Finally, this technique appears to inherently require strong assumptions regarding the precise time required to solve some specific computational problem.

A second approach, used in, e.g., [25] for exchanging secrets and in [3,20] for computation of general functions, gradually increases each party’s confidence in their output by, roughly speaking, masking the correct output with “noise” that tends to 0 as the protocol progresses. Protocols of this sort are inapplicable when the adversary has auxiliary information about the output of the function, since in that case the adversary’s “confidence” at any point in the protocol is impossible to estimate. More problematic is that an adversary can *bias* the output of the honest party beyond what is possible in the ideal world. As a simple illustration, consider a computation of the equality function where each party holds a value chosen uniformly from some domain D. In the ideal world, the probability that an adversary can cause the honest player to output 1 is exactly $1/|D|$. Using the approach of [3, 20], however, the adversary can cause the honest player to output 1 with probability essentially 1/2 by aborting in the first round (when the true answer is masked by an almost uniform random bit). Besides indicating a weakness of previous protocols, this example also demonstrates the importance of defining partial fairness within the simulation paradigm.

Gordon et al. [22] recently showed that *complete* fairness is possible in the two-party setting for certain specific functions. Work continuing that direction is complementary to our work here: while we do not yet have a complete characterization of what *can* be computed with complete fairness, we know that there certainly do exist some functions that *cannot* be computed with complete fairness [10] and so some relaxation must be considered (at least for some functions). Our feasibility results here apply to a much richer class of functions.

Other work has looked at achieving complete fairness with off-line trusted third parties (e.g., [7]) or in non-standard communication models (e.g., [24]). We work in the standard communication model, and without any trusted parties.

1.2 Overview of Our Approach

We now give an informal description of our feasibility results (details are in Section 3). Let x denote the input of P_1, let y denote the input of P_2, and let $f : X \times Y \to Z$ denote the function they are trying to compute. (For simplicity, here we omit the dependence of X, Y, and Z on n, and focus on the case where each party receives the same output.) As in [23, 22, 27], our protocols will be composed of two stages, where the first stage can be viewed as a "pre-processing" step and the second stage takes place in a sequence of $r = r(n)$ iterations. The stages have the following form:

First stage. This consists of the following steps:

1. A value $i^* \in \{1, \ldots, r\}$ is chosen according to some distribution (see below). This represents the iteration in which the parties will learn the "true output".
2. Values $a_1, \ldots, a_r$ and $b_1, \ldots, b_r$ are generated. For $i < i^*$, the $\{a_i\}$ (resp., $\{b_i\}$) are chosen (independently) according to some distribution that is independent of y (resp., x). For $i \geq i^*$, however, it holds that $a_i = b_i = f(x, y)$.
3. Each a_i is randomly shared as $a_i^{(1)}, a_i^{(2)}$ with $a_i^{(1)} \oplus a_i^{(2)} = a_i$ (and similarly for each b_i). The stage concludes with P_1 being given $a_1^{(1)}, b_1^{(1)}, \ldots, a_r^{(1)}, b_r^{(1)}$, and P_2 being given $a_1^{(2)}$, $b_1^{(2)}$, …, $a_r^{(2)}, b_r^{(2)}$. (Shares are also authenticated with an information-theoretic MAC.)

After this stage, each party has a set of random shares that reveal nothing about the other party's input. This stage can thus be carried out by any protocol that is secure-with-abort.

Second stage. In each iteration i, for $i = 1, \ldots, r$, the parties do the following: First, P_2 sends $a_i^{(2)}$ to P_1 who reconstructs a_i; then P_1 sends $b_i^{(1)}$ to P_2 who reconstructs b_i. (Parties also verify validity of the MAC but we omit this here.) If a party (say, P_1) aborts in some iteration i, then the other party (here, P_2) outputs the value reconstructed in the previous iteration (i.e., b_{i-1}). Otherwise, after reaching iteration r the parties output a_r and b_r, respectively.

To fully specify the protocol we must specify the distribution of i^* as well as the distribution of the a_i, b_i for $i < i^*$. As in [23,27], we choose i^* uniformly from $\{1, \ldots, r\}$. (In [22] a geometric distribution was used. That would work here, but with slightly worse round complexity.) When X and Y (the domains of f) are polynomial size, we follow [22] and set $a_i = f(x, \hat{y})$ for $\hat{y}$ chosen uniformly from Y, and set $b_i = f(\hat{x}, y)$ for $\hat{x}$ chosen uniformly (and independently) from X. Note that a_i (resp., b_i) is independent of y (resp., x), as desired.

Intuitively, this is partially fair because fairness is only violated if P_1 aborts *exactly* in iteration i^*. (If P_1 aborts before iteration i^* then neither party learns the "correct" value $z = f(x, y)$, while if it aborts subsequently then both parties learn the correct value. An abort by P_2 in iteration i^* does not violate fairness, since by then P_1 has already learned the output.) We show that *even if P_1 knows the value of z* (which it may, depending on partial information P_1 has about y), it cannot determine with certainty when iteration i^* occurs. Specifically, we prove a general result (see Lemma 1) implying (roughly) that as long as $\Pr[a_i = z] \geq \alpha$ for all $i < i^*$, then P_1 cannot abort in iteration i^* except with probability at most $1/\alpha r$ (recall that r is the number of iterations in the second phase). Since $\Pr[a_i = f(x, y)] = \Pr_{\hat{y} \in Y}[f(x, \hat{y}) = f(x, y)] \geq \Pr_{\hat{y} \in Y}[\hat{y} = y] = 1/|Y|$ for any x, y, we conclude that setting $r = p \cdot |Y|$, so that $1/\alpha r = 1/p$, suffices to achieve $\frac{1}{p}$-security. We thus get a protocol with polynomially many rounds as long as Y is polynomial size.

The above does not work directly when Y has super-polynomial size. To fix this, we must ensure that for every possible $z \in Z$ (the range of f) we have that $\Pr[a_i = z]$ is noticeable. We do this by changing the distribution of a_i (for $i < i^*$) as follows: with probability $1 - 1/q$ choose a_i as above, but with probability $1/q$ choose a_i uniformly from Z. Now, for any f, x, y, we have $\Pr[a_i = f(x, y)] \geq \frac{1}{q} \cdot \Pr_{a_i \in Z}[a_i = f(x, y)] \geq 1/q|Z|$ and so setting $r = pq|Z|$ ensures that P_1 cannot abort in iteration i^* except with probability at most $1/p$.

Changing the distribution of a_i, however, introduces a new problem: if P_2 aborts prior to iteration i^*, the output of the honest P_1 in the real world cannot necessarily be simulated in the ideal world. We show, however, that it *can* be simulated to within statistical difference $O(1/q)$. Taking $q = p$ (along with $r = pq|Z|$) thus gives a $\frac{1}{p}$-secure protocol with polynomially many rounds.

2 Definitions

Preliminaries. A function $\mu(\cdot)$ is *negligible* if for every positive polynomial $p(\cdot)$ and all sufficiently large n it holds that $\mu(n) < 1/p(n)$. A *distribution ensemble* $X = \{X(a, n)\}_{a \in \mathcal{D}_n,\, n \in \mathbb{N}}$ is an infinite sequence of random variables indexed by $a \in \mathcal{D}_n$ and $n \in \mathbb{N}$, where $\mathcal{D}_n$ may depend on n.

For a fixed function p, the distribution ensembles $X = \{X(a,n)\}_{a\in\mathcal{D}_n,\, n\in\mathbb{N}}$ and $Y = \{Y(a,n)\}_{a\in\mathcal{D}_n,\, n\in\mathbb{N}}$ are *computationally* $\frac{1}{p}$*-indistinguishable*, denoted $X \overset{1/p}{\approx} Y$, if for every non-uniform polynomial-time algorithm D there exists a negligible function $\mu(\cdot)$ such that for every n and every $a \in \mathcal{D}_n$

$$\Big|\Pr[D(X(a,n)) = 1] - \Pr[D(Y(a,n)) = 1]\Big| \leq \frac{1}{p(n)} + \mu(n).$$

Two distribution ensembles are *computationally indistinguishable*, denoted $X \overset{c}{\equiv} Y$, if for every $c \in \mathbb{N}$ they are computationally $\frac{1}{n^c}$-indistinguishable.

Functionalities. A *functionality* $\mathcal{F} = \{f_n\}_{n\in\mathbb{N}}$ is a sequence of poly-time computable, randomized mappings $f_n : X_n \times Y_n \to Z_n^1 \times Z_n^2$, where X_n and Z_n^1 (resp., Y_n and Z_n^2) denote the input and output of the first (resp., second) party. We write $f_n = (f_n^1, f_n^2)$ if we wish to emphasize the two outputs of f_n, but stress that if f_n^1 and f_n^2 are randomized then the outputs of f_n^1 and f_n^2 are correlated random variables. If $\Pr[f_n^1(x,y) = f_n^2(x,y)] = 1$ for all x, y, then we call f_n a *single-output functionality* and write it as $f_n : X_n \times Y_n \to Z_n$. If $\mathcal{F}$ is deterministic, we sometimes call it a *function*. For notational convenience, we sometimes drop the explicit dependence on n.

Two-party computation. A two-party protocol for computing a functionality $\mathcal{F} = \{(f^1, f^2)\}$ is a protocol running in polynomial time and satisfying the following correctness requirement: if party P_1 begins by holding 1^n and input $x \in X$, and party P_2 holds 1^n and input $y \in Y$, then the joint distribution of the outputs of the parties is statistically close to $(f^1(x,y), f^2(x,y))$.

Security of protocols. We consider *active* adversaries, who may deviate from the protocol in an arbitrary manner, and static corruptions. We use the standard real/ideal paradigm [18] (based on [26,2,8]). Define $\text{IDEAL}_{\mathcal{F},\mathcal{A}(\mathsf{aux})}(x,y,n)$ as the random variable consisting of the output of the adversary $\mathcal{A}$ and the output of the honest party following a computation of $\mathcal{F}$ in the ideal model (where complete fairness is guaranteed), with security parameter n and parties holding initial inputs x and y, respectively, and auxiliary input aux. We also define $\text{REAL}_{\Pi,\mathcal{A}(\mathsf{aux})}(x,y,n)$ as the analogous random variable for the real-world execution of protocol Π.

Having defined the ideal and real models, we now state our new notion of security. Loosely speaking, our definition asserts that a secure protocol (in the real model) emulates the ideal model (in which a trusted party exists) *to within a difference of* $\frac{1}{p}$. This is formulated as follows:

Definition 1. *Let $\mathcal{F}, \Pi$ be as above, and fix a function p. Protocol Π is said to* $\frac{1}{p}$-securely compute *$\mathcal{F}$ if for every non-uniform probabilistic polynomial-time adversary $\mathcal{A}$ in the real model, there exists a non-uniform probabilistic polynomial-time adversary $\mathcal{S}$ in the ideal model such that*

$$\left\{\text{IDEAL}_{\mathcal{F},\mathcal{S}(\mathsf{aux})}(x,y,n)\right\} \overset{1/p}{\approx} \left\{\text{REAL}_{\Pi,\mathcal{A}(\mathsf{aux})}(x,y,n)\right\}\ .$$

Although our definition of $\frac{1}{p}$-security allows privacy to be violated with probability $\frac{1}{p}$, in fact all our protocols are fully private. We remark further that $\frac{1}{p}$-security (even with privacy) and security-with-abort are incomparable.

3 $\frac{1}{\mathbf{p}}$-Secure Computation of General Functionalities

We begin in Section 3.1 by stating a lemma that forms an essential piece of our analysis in the two sections that follow. In Section 3.2 we demonstrate a private and $\frac{1}{p}$-secure protocol for functionalities defined on polynomial-size domains. A slight modification of this protocol is also simultaneously secure-with-abort. To keep the exposition as simple as possible, we restrict our attention there to single-output functionalities (though the techniques extend easily to the general case). In Section 3.3 we show how to adapt the protocol for functionalities defined over domains of super-polynomial size (but polynomial range), and also generalize to functionalities generating different outputs for each party.

3.1 A Useful Lemma

We analyze an abstract game Γ between a challenger and an (unbounded) adversary $\mathcal{A}$. The game is parameterized by a value $\alpha \in (0,1]$ and an integer $r \geq 1$. Fix arbitrary distributions D_1, D_2 such that for every z it holds that

$$\Pr_{a \leftarrow D_1}[a = z] \geq \alpha \cdot \Pr_{a \leftarrow D_2}[a = z]. \tag{1}$$

The game $\Gamma(\alpha, r)$ proceeds as follows:

1. The challenger chooses i^* uniformly from $\{1, \ldots, r\}$, and then chooses $a_1, \ldots, a_r$ as follows:
   - For $i < i^*$, it chooses $a_i \leftarrow D_1$.
   - For $i \geq i^*$, it chooses $a_i \leftarrow D_2$.
2. The challenger and $\mathcal{A}$ then interact in a sequence of at most r iterations. In iteration i:
   - The challenger gives a_i to the adversary.
   - The adversary can either abort or continue. In the former case, the game stops. In the latter case, the game continues to the next iteration.
3. $\mathcal{A}$ *wins* if it aborts the game in iteration i^*.

Let $\mathsf{Win}(\alpha, r)$ denote the maximum probability with which $\mathcal{A}$ wins the game.

Lemma 1. *For any D_1, D_2 satisfying (1), it holds that $\mathsf{Win}(\alpha, r) \leq 1/\alpha r$.*

Proof. Fix D_1, D_2 satisfying (1). We prove the lemma by induction on r. When $r = 1$ the lemma is trivially true; for completeness, we also directly analyze the case $r = 2$. Since $\mathcal{A}$ is unbounded we may assume it is deterministic. So without loss of generality, we may assume the adversary's strategy is determined by a set S in the support of D_2 such that $\mathcal{A}$ aborts in the first iteration iff $a_1 \in S$, and otherwise aborts in the second iteration (no matter what). We have

$$\begin{aligned}
\Pr[\mathcal{A} \text{ wins}] &= \Pr[\mathcal{A} \text{ wins and } i^* = 1] + \Pr[\mathcal{A} \text{ wins and } i^* = 2] \\
&= \frac{1}{2} \cdot \Pr_{a \leftarrow D_2}[a \in S] + \frac{1}{2} \cdot \Big(1 - \Pr_{a \leftarrow D_1}[a \in S]\Big) \\
&\leq \frac{1}{2} \cdot \Pr_{a \leftarrow D_2}[a \in S] + \frac{1}{2} \cdot \Big(1 - \alpha \cdot \Pr_{a \leftarrow D_2}[a \in S]\Big) \\
&= \frac{1}{2} + \frac{1}{2} \cdot \Big((1-\alpha) \cdot \Pr_{a \leftarrow D_2}[a \in S]\Big) \;\; \leq \;\; 1 - \alpha/2,
\end{aligned}$$

where the first inequality is due to Equation (1). One can easily verify that $1 - \alpha/2 \leq 1/2\alpha$ when $\alpha > 0$. We have thus proved $\mathsf{Win}(\alpha, 2) \leq 1/2\alpha$.

Assume $\mathsf{Win}(\alpha, r) \leq 1/\alpha r$, and we now bound $\mathsf{Win}(\alpha, r+1)$. As above, let S denote a set in the support of D_2 such that $\mathcal{A}$ aborts in the first iteration iff $a_1 \in S$. If $\mathcal{A}$ does *not* abort in the first iteration, and the game does not end, then the conditional distribution of i^* is uniform in $\{2, \ldots, r+1\}$ and the game $\Gamma(\alpha, r+1)$ from this point forward is exactly equivalent to the game $\Gamma(\alpha, r)$. In particular, conditioned on the game $\Gamma(\alpha, r+1)$ not ending after the first iteration, the best strategy for $\mathcal{A}$ is to play whatever is the best strategy in game $\Gamma(\alpha, r)$. We thus have

$$\begin{aligned}
\mathsf{Win}(\alpha, r+1) &= \Pr[\mathcal{A} \text{ wins and } i^* = 1] + \Pr[\mathcal{A} \text{ wins and } i^* > 1] \\
&= \frac{1}{r+1} \cdot \Pr_{a \leftarrow D_2}[a \in S] + \frac{r}{r+1} \cdot \Big(1 - \Pr_{a \leftarrow D_1}[a \in S]\Big) \cdot \mathsf{Win}(\alpha, r) \\
&\leq \frac{1}{r+1} \cdot \Pr_{a \leftarrow D_2}[a \in S] + \frac{1}{\alpha(r+1)} \cdot \Big(1 - \alpha \cdot \Pr_{a \leftarrow D_2}[a \in S]\Big) \cdot \\
&= \frac{1}{\alpha(r+1)}.
\end{aligned}$$

This completes the proof.

3.2 $\frac{1}{p}$-Security for Functionalities with Polynomial-Size Domain

In this section, we describe a protocol that works for functionalities where at least one of the domains is polynomial-size. (We stress that the protocol works *directly* for randomized functionalities; the standard reduction from randomized to deterministic functionalities [18] would not apply here since, in general, it makes the domain too large.) Although a small modification of the protocol works even when the parties receive different outputs, for simplicity we assume here that the parties compute a single-output function. We return to the more general setting in the following section.

Theorem 1. *Let $\mathcal{F} = \{f_n : X_n \times Y_n \to Z_n\}$ be a (randomized) functionality where $|Y_n| = \mathsf{poly}(n)$. Assuming the existence of enhanced trapdoor permutations, for any polynomial p there is an $\mathcal{O}(p \cdot |Y_n|)$-round protocol computing $\mathcal{F}$ that is private and $\frac{1}{p}$-secure.*

$\mathsf{ShareGen}_r$

Inputs: The security parameter is n. Let the inputs to $\mathsf{ShareGen}_r$ be $x \in X_n$ and $y \in Y_n$. (If one of the received inputs is not in the correct domain, a default input is substituted.)

Computation:

1. Define values $a_1, \ldots, a_r$ and $b_1, \ldots, b_r$ in the following way:
   - Choose i^* uniformly at random from $\{1, \ldots, r\}$.
   - For $i = 1$ to $i^* - 1$ do:
     - Choose $\hat{y} \leftarrow Y_n$ and set $a_i = f_n(x, \hat{y})$.
     - Choose $\hat{x} \leftarrow X_n$ and set $b_i = f_n(\hat{x}, y)$.
   - Compute $z = f_n(x, y)$. For $i = i^*$ to r, set $a_i = b_i = z$.
2. For $1 \leq i \leq r$, choose $(a_i^{(1)}, a_i^{(2)})$ and $(b_i^{(1)}, b_i^{(2)})$ as random secret sharings of a_i and b_i, respectively. (I.e., $a_i^{(1)}$ is random and $a_i^{(1)} \oplus a_i^{(2)} = a_i$.)
3. Compute $k_a, k_b \leftarrow \mathsf{Gen}(1^n)$. For $1 \leq i \leq r$, let $t_i^a = \mathsf{Mac}_{k_a}(i \| a_i^{(2)})$ and $t_i^b = \mathsf{Mac}_{k_b}(i \| b_i^{(1)})$.

Output:

1. Send to P_1 the values $a_1^{(1)}, \ldots, a_r^{(1)}$ and $(b_1^{(1)}, t_1^b), \ldots, (b_r^{(1)}, t_r^b)$, and the MAC-key k_a.
2. Send to P_2 the values $(a_1^{(2)}, t_1^a), \ldots, (a_r^{(2)}, t_r^a)$ and $b_1^{(2)}, \ldots, b_r^{(2)}$, and the MAC-key k_b.

Fig. 1. Functionality $\mathsf{ShareGen}_r$

Proof. As described in Section 1.2, our protocol Π consists of two stages. Let p be an arbitrary polynomial, and set $r = p \cdot |Y_n|$. We will implement the first stage of Π using a sub-protocol π for computing a randomized functionality $\mathsf{ShareGen}_r$ defined in Figure 1. ($\mathsf{ShareGen}_r$ is parameterized by a polynomial r.) This functionality returns shares to each party, authenticated using an information-theoretically secure r-time MAC $(\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$. In the second stage of Π the parties exchange these shares in a sequence of r iterations as described in Figure 2.

We analyze our protocol in a hybrid model where there is a trusted party computing $\mathsf{ShareGen}_r$ according to the second ideal model where a malicious P_1 can abort the trusted party before it sends output to the honest party. We prove privacy and $\frac{1}{p}$-security of Π in this hybrid model; it follows as in [8] that if we use a sub-protocol for computing $\mathsf{ShareGen}_r$ that is secure-with-abort, then the real-world protocol Π is private and $\frac{1}{p}$-secure.

We first consider the case of a malicious P_1. Intuition for the following claim was given in Section 1.2. The formal statement and proof follow.

Claim 1. *Let Π^{hy} denote an execution of Π in a hybrid model with access to an ideal functionality computing $\mathsf{ShareGen}_r$ (with abort). For every non-uniform, polynomial-time adversary $\mathcal{A}$ corrupting P_1 and running Π^{hy}, there exists a non-uniform, polynomial-time adversary $\mathcal{S}$ corrupting P_1 and running in the ideal*

Protocol 1

Inputs: Party P_1 has input x and party P_2 has input y. The security parameter is n. Let $r = p \cdot |Y_n|$.

The protocol:

1. **Preliminary phase:**
   - (a) P_1 chooses $\hat{y} \in Y_n$ uniformly at random, and sets $a_0 = f_n(x, \hat{y})$. Similarly, P_2 chooses $\hat{x} \in X_n$ uniformly at random, and sets $b_0 = f_n(\hat{x}, y)$.
   - (b) Parties P_1 and P_2 run a protocol π to compute $\mathsf{ShareGen}_r$, using their inputs x and y.
   - (c) If P_2 receives $\perp$ from the above computation, it outputs b_0 and halts. Otherwise, the parties proceed to the next step.
   - (d) Denote the output of P_1 from π by $a_1^{(1)}, \ldots, a_r^{(1)}$, $(b_1^{(1)}, t_1^b), \ldots, (b_r^{(1)}, t_r^b)$, and k_a.
   - (e) Denote the output of P_2 from π by $(a_1^{(2)}, t_1^a), \ldots, (a_r^{(2)}, t_r^a)$, $b_1^{(2)}, \ldots, b_r^{(2)}$, and k_b.
2. **For $\boldsymbol{i = 1, \ldots, r}$ do:**

   $\boldsymbol{P_2}$ sends the next share to $\boldsymbol{P_1}$:
   - (a) P_2 sends $(a_i^{(2)}, t_i^a)$ to P_1.
   - (b) P_1 receives $(a_i^{(2)}, t_i^a)$ from P_2. If $\mathsf{Vrfy}_{k_a}(i \| a_i^{(2)}, t_i^a) = 0$ (or if P_1 received an invalid message, or no message), then P_1 outputs a_{i-1} and halts.
   - (c) If $\mathsf{Vrfy}_{k_a}(i \| a_i^{(2)}, t_i^a) = 1$, then P_1 sets $a_i = a_i^{(1)} \oplus a_i^{(2)}$ (and continues running the protocol).

   $\boldsymbol{P_1}$ sends the next share to $\boldsymbol{P_2}$:
   - (a) P_1 sends $(b_i^{(1)}, t_i^b)$ to P_2.
   - (b) P_2 receives $(b_i^{(1)}, t_i^b)$ from P_1. If $\mathsf{Vrfy}_{k_b}(i \| b_i^{(1)}, t_i^b) = 0$ (or if P_2 received an invalid message, or no message), then P_2 outputs b_{i-1} and halts.
   - (c) If $\mathsf{Vrfy}_{k_b}(i \| b_i^{(1)}, t_i^b) = 1$, then P_2 sets $b_i = b_i^{(1)} \oplus b_i^{(2)}$ (and continues running the protocol).
3. If all r iterations have been run, party P_1 outputs a_r and party P_2 outputs b_r.

Fig. 2. Generic protocol for computing a functionality f_n

world with access to an ideal functionality computing $\mathcal{F}$ (with complete fairness), such that $\frac{1}{p}$-security and privacy hold.

Proof. We construct a simulator $\mathcal{S}$ given black-box access to $\mathcal{A}$. *For readability in what follows, we ignore the MAC-tags and keys.* When we say that $\mathcal{A}$ "aborts", we include in this the event that $\mathcal{A}$ sends an invalid message, or a message whose tag does not pass verification. We also drop the subscript n from our notation and write X, Y in place of X_n, Y_n.

1. $\mathcal{S}$ invokes $\mathcal{A}$ on the input[2] x', the auxiliary input, and the security parameter n. The simulator also chooses $\hat{x} \in X$ uniformly at random (it will send $\hat{x}$ to the trusted party, if needed).
2. $\mathcal{S}$ receives the input x of $\mathcal{A}$ to the computation of the functionality $\mathsf{ShareGen}_r$. (If $x \notin X$ a default input is substituted.)

[2] We reserve x for the value input by $\mathcal{A}$ to the computation of $\mathsf{ShareGen}_r$.

3. $\mathcal{S}$ sets $r = p \cdot |Y|$, and chooses uniformly-distributed shares $a_1^{(1)}, \ldots, a_r^{(1)}$ and $b_1^{(1)}, \ldots, b_r^{(1)}$. Then, $\mathcal{S}$ gives these shares to $\mathcal{A}$ as its output from the computation of $\mathsf{ShareGen}_r$.
4. If $\mathcal{A}$ sends abort to the trusted party computing $\mathsf{ShareGen}_r$, then $\mathcal{S}$ sends $\hat{x}$ to the trusted party computing f, outputs whatever $\mathcal{A}$ outputs, and halts. Otherwise (i.e., if $\mathcal{A}$ sends $\mathsf{continue}$), $\mathcal{S}$ proceeds as below.
5. Choose i^* uniformly from $\{1, \ldots, r\}$
6. For $i = 1$ to $i^* - 1$:
   (a) $\mathcal{S}$ chooses $\hat{y} \in Y$ uniformly at random, computes $a_i = f(x, \hat{y})$, and sets $a_i^{(2)} = a_i^{(1)} \oplus a_i$. It gives $a_i^{(2)}$ to $\mathcal{A}$. (A fresh $\hat{y}$ is chosen in every iteration.)
   (b) If $\mathcal{A}$ aborts, then $\mathcal{S}$ sends $\hat{x}$ to the trusted party, outputs whatever $\mathcal{A}$ outputs, and halts.
7. For $i = i^*$ to r:
   (a) If $i = i^*$ then $\mathcal{S}$ sends x to the trusted party computing f and receives $z = f(x, y)$.
   (b) $\mathcal{S}$ sets $a_i^{(2)} = a_i^{(1)} \oplus z$ and gives $a_i^{(2)}$ to $\mathcal{A}$.
   (c) If $\mathcal{A}$ aborts, then $\mathcal{S}$ then outputs whatever $\mathcal{A}$ outputs, and halts. If $\mathcal{A}$ does not abort, then $\mathcal{S}$ proceeds.
8. If $\mathcal{A}$ never aborted (and all r iterations are done), $\mathcal{S}$ outputs what $\mathcal{A}$ outputs and halts.

It is immediate that the view of $\mathcal{A}$ in the simulation above is distributed identically to its view in Π^{hy}; privacy follows. We now prove $\frac{1}{p}$-security.

Ignoring the possibility of a MAC forgery, we claim that the statistical difference between an execution of $\mathcal{A}$, running Π in a hybrid world with access to an ideal functionality computing $\mathsf{ShareGen}_r$, and an execution of $\mathcal{S}$, running in an ideal world with access to an ideal functionality computing f, is at most $1/p$. (Thus, taking into account the possibility of a MAC forgery makes the statistical difference at most $1/p + \mu(n)$ for some negligible function μ.) To see this, let y denote the input of the honest P_2 and consider three cases depending on when the adversary aborts:

1. $\mathcal{A}$ aborts in round $i < i^*$. Conditioned on this event, the view of $\mathcal{A}$ is identically distributed in the two worlds (and is independent of y), and the output of the honest party is $f(\hat{x}, y)$ for $\hat{x}$ chosen uniformly in X.
2. $\mathcal{A}$ aborts in round $i > i^*$ (or never). Conditioned on this, the view of $\mathcal{A}$ is again distributed identically in the two worlds, and in both worlds the output of the honest party is $f(x, y)$.
3. $\mathcal{A}$ aborts in round $i = i^*$: here, although the view of $\mathcal{A}$ is still identical in both worlds, the output of the honest party is not: in the hybrid world the honest party will output $f(\hat{x}, y)$, for $\hat{x}$ chosen uniformly in X, while in the ideal world the honest party will output $f(x, y)$.

   However, Lemma 1 implies that this event occurs with probability at most $1/p$. To see this, let D_1 denote the distribution of a_i for $i < i^*$ (i.e., this is the distribution defined by the output of $f(x, \hat{y})$, for $\hat{y}$ chosen uniformly from Y), and let D_2 denote the distribution of a_{i^*} (i.e., the distribution defined by the output of $f(x,y)$). For any $z \in Z$ we have

$$\begin{aligned}\Pr_{a \leftarrow D_1}[a = z] &\overset{\text{def}}{=} \Pr_{\hat{y} \leftarrow Y}[f(x,\hat{y}) = z] \\ &\geq \frac{1}{|Y|} \cdot \Pr[f(x,y) = z] \;=\; \frac{1}{|Y|} \cdot \Pr_{a \leftarrow D_2}[a = z].\end{aligned}$$

Taking $\alpha = 1/|Y|$ and applying Lemma 1, we see that $\mathcal{A}$ aborts in iteration i^* with probability at most $1/\alpha r = |Y|/|Y|p = 1/p$.

This completes the proof of the claim.

Next we consider the case of a malicious P_2. A proof of the following is almost identical to that of Claim 1; in fact, the proof is simpler and we can prove a stronger notion of security since P_1 always "gets the output first" in every iteration of Π. For these reasons, a proof is omitted.

Claim 2. *(Informal.) Let Π^{hy} denote an execution of Π in a hybrid model where the parties have access to an ideal functionality computing $\mathsf{ShareGen}_r$ (with abort). Then for any adversary corrupting P_2, protocol Π^{hy} securely computes $\mathcal{F}$ (which in particular implies privacy).*

The results of [8], along with the fact that a secure-with-abort protocol for $\mathsf{ShareGen}_r$ is implied by the existence of enhanced trapdoor permutations, complete the proof of Theorem 1.

Achieving security-with-abort. As written, the protocol is not secure-with-abort. However, the protocol can be modified easily so that it is (without affecting $\frac{1}{p}$-security): simply have $\mathsf{ShareGen}_r$ choose i^* uniformly from $\{2,\ldots,r+1\}$ and set $b_{i^*-1} = \perp$, where $\perp$ is some distinguished value outside the range of f. Although this allows a malicious P_2 to identify exactly when iteration i^* occurs, this does not affect security since by that time P_1 has already received the correct output.

3.3 $\frac{1}{\mathbf{p}}$-Security for Functionalities with Polynomial-Size Range

The protocol from the previous section does not apply to functions on domains of super-polynomial size, since the round complexity is linear in the size of the smaller domain. Here we show how to extend the protocol to handle arbitrary domains if the range of the function (for at least one of the parties) is polynomial size. We now also explicitly take into account the case when parties obtain different outputs. Intuition for the changes we introduce is given in Section 1.2.

Theorem 2. *Let $\mathcal{F} = \{f_n : X_n \times Y_n \to Z_n^1 \times Z_n^2\}$ be a (randomized) functionality, where $|Z_n^1| = \mathsf{poly}(n)$. Assuming the existence of enhanced trapdoor permutations, for any polynomial p there is an $\mathcal{O}\left(p^2 \cdot |Z_n^1|\right)$-round protocol computing $\mathcal{F}$ that is private and $\frac{1}{p}$-secure.*

$\mathsf{ShareGen}'_{p,r}$

Inputs: The security parameter is n. Let the inputs to $\mathsf{ShareGen}'_{p,r}$ be $x \in X_n$ and $y \in Y_n$. (If one of the received inputs is not in the correct domain, a default input is substituted.)

Computation:

1. Define values $a_1, \ldots, a_r$ and $b_1, \ldots, b_r$ in the following way:
   - Choose i^* uniformly at random from $\{1, \ldots, r\}$.
   - For $i = 1$ to $i^* - 1$ do:
     - Choose $\hat{x} \leftarrow X_n$ and set $b_i = f_n^2(\hat{x}, y)$.
     - With probability $\frac{1}{p}$, choose $z \leftarrow Z_n^1$ and set $a_i = z$. With the remaining probability $1 - \frac{1}{p}$, choose $\hat{y} \leftarrow Y$ and set $a_i = f_n^1(x, \hat{y})$.
   - Compute $z_1 = f_n^1(x, y)$ and $z_2 = f_n^2(x, y)$ (if $f_n = (f_n^1, f_n^2)$ is randomized, these values are computed using the same random tape). For $i = i^*$ to r, set $a_i = z_1$ and $b_i = z_2$.
2. For $1 \leq i \leq r$, choose $(a_i^{(1)}, a_i^{(2)})$ and $(b_i^{(1)}, b_i^{(2)})$ as random secret sharings of a_i and b_i, respectively. (E.g., $a_i^{(1)}$ is random and $a_i^{(1)} \oplus a_i^{(2)} = a_i$.)
3. Compute $k_a, k_b \leftarrow \mathsf{Gen}(1^n)$. For $1 \leq i \leq r$, let $t_i^a = \mathsf{Mac}_{k_a}(i \| a_i^{(2)})$ and $t_i^b = \mathsf{Mac}_{k_b}(i \| b_i^{(1)})$.

Output:

1. Send to P_1 the values $a_1^{(1)}, \ldots, a_r^{(1)}$ and $(b_1^{(1)}, t_1^b), \ldots, (b_r^{(1)}, t_r^b)$, and the MAC-key k_a.
2. Send to P_2 the values $(a_1^{(2)}, t_1^a), \ldots, (a_r^{(2)}, t_r^a)$ and $b_1^{(2)}, \ldots, b_r^{(2)}$, and the MAC-key k_b.

Fig. 3. Functionality $\mathsf{ShareGen}'_{p,r}$

Proof. Our protocol Π is, once again, composed of two stages. The second stage is identical to the second stage of the previous protocol (see Figure 2), except that the number of iterations r is now set to $r = p^2 \cdot |Z_n^1|$. The first stage generates shares using a sub-routine π computing a different functionality $\mathsf{ShareGen}'_{p,r}$, parameterized by both p and r and described in Figure 3.

We again analyze our protocol in a hybrid model, where there is now a trusted party computing $\mathsf{ShareGen}'_{p,r}$. (Once again, P_1 can abort the computation of $\mathsf{ShareGen}'_{p,r}$ in the ideal world.) We prove privacy and $\frac{1}{p}$-security of Π in this hybrid model, implying [8] that if the parties use a secure-with-abort protocol for computing $\mathsf{ShareGen}'_{p,r}$, then the real-world protocol Π is private and $\frac{1}{p}$-secure. We first consider the case of a malicious P_1.

Claim 3. *(Informal.) Let Π^{hy} denote an execution of Π in a hybrid model where the parties have access to an ideal functionality computing $\mathsf{ShareGen}'_{p,r}$ (with abort). Then for any adversary corrupting P_1, protocol Π^{hy} privately and $\frac{1}{p}$-securely computes $\mathcal{F}$.*

Proof. The simulator used to prove this claim is essentially the same as the simulator used in the proof of Claim 1, except that in step 6(a) the distribution on

a_i (for $i < i^*$) is changed to the one used by $\mathsf{ShareGen}'_{p,r}$. The analysis is similar, too, except for bounding the probability that $\mathcal{A}$ aborts in iteration i^*. To bound this probability we will again rely on Lemma 1, but now distribution D_1 (i.e., the distribution of a_i for $i < i^*$) is different. Let y denote the input of P_2. Note that, by construction of $\mathsf{ShareGen}'_{p,r}$, for any $z \in Z_n^1$ we have $\Pr_{a \leftarrow D_1}[a = z] \geq \frac{1}{p} \cdot \frac{1}{|Z_n^1|}$. Regardless of f^1 and y, it therefore holds for all $z \in Z_n^1$ that

$$\Pr_{a \leftarrow D_1}[a = z] \geq \frac{1}{p \cdot |Z_n^1|} \cdot \Pr_{a \leftarrow D_2}[a = z].$$

Setting $\alpha = 1/p \cdot |Z_n^1|$ and applying Lemma 1, we see that $\mathcal{A}$ aborts in iteration i^* with probability at most

$$\frac{1}{\alpha r} = \frac{p \cdot |Z_n^1|}{p^2 \cdot |Z_n^1|} = \frac{1}{p}.$$

This completes the proof of the claim.

We next consider the case of a malicious P_2. Note that, in contrast to Claim 2, here we claim only $\frac{1}{p}$-security.

Claim 4. *(Informal.) Let Π^{hy} denote an execution of Π in a hybrid model where the parties have access to an ideal functionality computing $\mathsf{ShareGen}'_{p,r}$ (with abort). Then for any adversary corrupting P_2, protocol Π^{hy} privately and $\frac{1}{p}$-securely computes $\mathcal{F}$.*

Proof. A proof appears in the full version of this work, and is omitted here due to space constraints.

The results of [8], along with the fact that a secure-with-abort protocol for $\mathsf{ShareGen}'_{p,r}$ is implied by the existence of enhanced trapdoor permutations, complete the proof of Theorem 2.

4 Optimality of Our Results

We show that the results of the previous section are optimal as far as generic feasibility is concerned.

4.1 Impossibility of $\frac{1}{p}$-Security and Security-with-Abort Simultaneously

In Section 3.2 (cf. the remark at the end of that section) we showed a protocol achieving $\frac{1}{p}$-security and security-with-abort *simultaneously* for functionalities where at least one of the domains is polynomial-size. We show that if both domains are super-polynomial in size then, in general, it is impossible to achieve both these criteria at once.

Theorem 3. *Let $\mathcal{F} = \left\{\mathsf{EQ}_n : \{0,1\}^{\ell(n)} \times \{0,1\}^{\ell(n)} \to \{0,1\}\right\}$, where EQ_n denotes the equality function on strings and $\ell(n) = \omega(\log n)$. Let Π be any protocol computing $\mathcal{F}$. If Π is secure-with-abort, then Π does not $\frac{1}{p}$-securely compute $\mathcal{F}$ for any $p \geq 4 + \frac{1}{\mathsf{poly}(n)}$.*

Proof. Let Π be a protocol that computes $\mathcal{F}$ and is secure-with-abort. Assume without loss of generality that P_2 sends the first message in Π and that P_1 sends the last message. Say Π has $r = r(n)$ iterations for some polynomial r, where an iteration consists of a message sent by P_2 followed by a message sent by P_1. Let a_0 denote the value that P_1 outputs if P_2 sends nothing, and let a_i, for $1 \leq i \leq r$, denote the value that P_1 outputs if P_2 aborts after sending its iteration-i message. Similarly, let b_0 denote the value that P_2 outputs if P_1 sends nothing, and let b_i, for $1 \leq i \leq r$, denote the value that P_2 outputs if P_1 aborts after sending its iteration-i message. We may assume without loss of generality that, for all i, we have $a_i \in \{0,1\}$ and $b_i \in \{0,1,\perp\}$.

We will consider two experiments involving an execution of Π. In the first, x and y are chosen uniformly and independently from $\{0,1\}^{\ell(n)}$; the parties are given inputs x and y, respectively; and the parties then run protocol Π honestly. We denote the probability of events in this experiment by $\Pr_{\mathrm{rand}}[\cdot]$. In the second experiment, x is chosen uniformly from $\{0,1\}^{\ell(n)}$ and y is set equal to x; these inputs are given to the parties and they run the protocol honestly as before. We denote the probability of events in this probability space by $\Pr_{\mathrm{eq}}[\cdot]$.

Lemma 2. $\Pr_{\mathrm{rand}}[a_0 = 1 \vee \cdots \vee a_r = 1]$ *and* $\Pr_{\mathrm{rand}}[b_0 = 1 \vee \cdots \vee b_r = 1]$ *are negligible.*

Proof. This follows from the fact that Π is secure-with-abort. If, say, it were the case that $\Pr_{\mathrm{rand}}[a_0 = 1 \vee \cdots \vee a_r = 1]$ is not negligible, then we could consider an adversarial P_2 that runs the protocol honestly but aborts at a random round. This would cause the honest P_1 to output 1 with non-negligible probability in the real world, whereas P_1 outputs 1 with only negligible probability in the ideal world (since the parties are given independent, random inputs).

Assume for simplicity that Π has perfect correctness, i.e., that $a_r = b_r = \mathsf{EQ}(x,y)$ when the two parties run the protocol honestly holding initial inputs x and y. (This assumption is not necessary, but allows us to avoid having to deal with annoying technicalities.) Then

$$\Pr_{\mathrm{eq}}[a_0 = 1 \vee \cdots \vee a_r = 1] = \Pr_{\mathrm{eq}}[b_0 = 1 \vee \cdots \vee b_r = 1] = 1$$

since, in particular, $\Pr_{\mathrm{eq}}[a_r = 1] = \Pr_{\mathrm{eq}}[b_r = 1] = 1$. In a given execution, let i^* denote the lowest index for which $a_{i^*} = 1$, and let j^* denote the lowest index for which $b_{j^*} = 1$. Since

$$\Pr_{\mathrm{eq}}[i^* \leq j^*] + \Pr_{\mathrm{eq}}[i^* > j^*] = 1,$$

at least one of the terms on the left-hand side is at least 1/2. We assume that $\Pr_{\mathrm{eq}}[i^* \leq j^*] \geq 1/2$ in what follows, but the same argument (swapping the roles of the parties) applies if $\Pr_{\mathrm{eq}}[i^* > j^*] \geq 1/2$.

Consider now a third experiment that is a mixture of the previous two. Specifically, in this experiment a random bit b is chosen; if $b = 0$ then the parties are given inputs x and y as in the first experiment (i.e., chosen uniformly and independently at random), while if $b = 1$ then the parties are given (random) $x = y$ as in the second experiment. The parties then run protocol Π honestly. We denote the probability of events in this probability space by $\Pr_3^{\mathsf{real}}[\cdot]$. We use the superscript real to distinguish this from an ideal-world version of this experiment where the bit b is chosen uniformly and the parties are given x and y generated accordingly, but now the parties interact with an ideal party computing EQ *without abort* (i.e., in the first ideal model). We denote the probability of events in this experiment by $\Pr_3^{\mathsf{ideal}}[\cdot]$.

Consider an execution of the third experiment (in either the real or ideal worlds), in the case when P_1 is malicious. Let guess denote the event that P_1 correctly guesses the value of the bit b, and let out_2 denote the output of P_2. It is not hard to show that

$$\Pr_3^{\mathsf{ideal}}[\mathsf{guess} \wedge \mathsf{out}_2 \neq 1] = \frac{1}{2}. \tag{2}$$

(Note that $\mathsf{out}_2 \in \{0,1\}$ in the first ideal world.) Now take the following real-world adversary $\mathcal{A}$ corrupting P_1: upon receiving input x, adversary $\mathcal{A}$ runs Π honestly but computes a_i after receiving each iteration-i message from P_2. Then:

- If, at some point, $a_i = 1$ then $\mathcal{A}$ aborts the protocol (before sending the iteration-i message on behalf of P_1) and outputs the guess "$b = 1$".
- If $a_i = 0$ for all i, then $\mathcal{A}$ simply runs the protocol to the end (including the final message of the protocol) and outputs the guess "$b = 0$".

We have:

$$\begin{aligned}
&\Pr_3^{\mathsf{real}}[\mathsf{guess} \wedge \mathsf{out}_2 \neq 1] \\
&= \frac{1}{2} \cdot \Pr_{\mathrm{rand}}[\mathsf{guess} \wedge \mathsf{out}_2 \neq 1] + \frac{1}{2} \cdot \Pr_{\mathrm{eq}}[\mathsf{guess} \wedge \mathsf{out}_2 \neq 1] \\
&\geq \frac{1}{2} \cdot \Pr_{\mathrm{rand}}[a_1 = 0 \wedge \cdots \wedge a_r = 0 \wedge b_r = 0] + \frac{1}{2} \cdot \Pr_{\mathrm{eq}}[i^* \leq j^*] \\
&\geq \frac{1}{2} \cdot (1 - \mathsf{negl}(n)) + \frac{1}{4} \;\; = \;\; \frac{3}{4} - \mathsf{negl}(n),
\end{aligned} \tag{3}$$

using Lemma 2 for the second inequality. Equations (2) and (3) show that Π cannot also be $\frac{1}{p}$-secure for any $p \geq 4 + \frac{1}{\mathsf{poly}(n)}$.

4.2 Impossibility of $\frac{1}{p}$-Security for General Functions

Our results show that $\frac{1}{p}$-security is achievable for any functionality $f : X_n \times Y_n \to Z_n^1 \times Z_n^2$ if at least one of X_n, Y_n, Z_n^1, Z_n^2 has polynomial size. Here, we demonstrate that this limitation is inherent.

Define a deterministic, single-output function $\mathcal{F} = \{\mathsf{Swap}_n\}$ with

$$\mathsf{Swap}_n : \{0,1\}^{\omega(\log n)} \times \{0,1\}^{\omega(\log n)} \to \{0,1\}^{\omega(\log n)}$$

as follows: Fix some $\ell(n) = \omega(\log n)$. Let $(\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ denote an information-theoretic, one-time MAC for messages of length $2 \cdot \ell(n)$ with key length $O(\ell(n))$ and tag length $\ell(n)$. Then

$$\begin{aligned}&\mathsf{Swap}_n\left((x_1, t_1, k_2), (x_2, t_2, k_1)\right)\\ &\overset{\text{def}}{=} \begin{cases} (x_1, x_2) & \text{if } \mathsf{Vrfy}_{k_1}(x_1, t_1) = \mathsf{Vrfy}_{k_2}(x_2, t_2) = 1 \\ \perp & \text{otherwise} \end{cases}.\end{aligned}$$

(Note that both parties receive the same output (x_1, x_2) in the first case.)

Theorem 4. *Function $\mathcal{F}$ cannot be $\frac{1}{p}$-securely computed for any $p \geq 2 + \frac{1}{\mathsf{poly}(n)}$.*

Proof. Consider an ideal-world computation of Swap where:

- x_1, x_2 are chosen uniformly at random from $\{0,1\}^{2\ell(n)}$.
- k_1, k_1', k_2, k_2' are output by $\mathsf{Gen}(1^n)$ (i.e., they are random MAC-keys).
- $t_1 = \mathsf{Mac}_{k_1}(x_1)$, $t_1' = \mathsf{Mac}_{k_1'}(x_1)$, $t_2 = \mathsf{Mac}_{k_2}(x_2)$, and $t_2' = \mathsf{Mac}_{k_2'}(x_2)$.
- P_1 is given input (x_1, t_1, k_2) and auxiliary information (k_2', t_2')
- P_2 is given input (x_2, t_2, k_1) and auxiliary information (k_1', t_1').

Define a *win* for P_1 as the event that P_1 outputs x_2 while P_2 fails to output x_1. (A win for P_2 is defined analogously.) It is easy to see that, e.g., a malicious P_1 cannot win in the ideal world, where complete fairness is guaranteed, except with negligible probability. This is because x_2 is a uniform $2\ell(n)$-bit value, while the only information P_1 has about x_2 initially is the $\ell(n)$-bit tag t_2'. Thus, the only way for P_1 to learn x_2 is to submit to the trusted party some input $(\hat{x}_1, \hat{t}_1, \hat{k}_2)$ for which $\mathsf{Vrfy}_{k_1}(\hat{x}_1, \hat{t}_1) = 1$; unless $\hat{x}_1 = x_1$, however, this condition holds with negligible probability.

In any real-world computation of Swap, however, there must be one party who "gets its output first" with probability at least 1/2, and can identify exactly when this occurs using its auxiliary information. More formally, say we have an r-iteration protocol Π computing Swap where P_2 sends the first message and P_1 sends the last message. Let a_i, for $i = 0, \ldots, r$, denote the second component of the value P_1 would output if P_2 aborts the protocol after sending its iteration-i message, and let b_i denote the first component of the value that P_2 would output if P_1 aborts the protocol after sending its iteration-i message. Each value a_i and b_i can be computed in polynomial time after receiving the other party's iteration-i message. We can therefore define an adversary P_1^* that acts as follows:

> Run the protocol honestly until the first round where $\mathsf{Vrfy}_{k_2'}(a_i, t_2') = 1$; then output a_i and abort.

An adversary P_2^* can be defined analogously. Note that if, e.g., $\mathsf{Vrfy}_{k_2'}(a_i, t_2') = 1$ then $a_i = x_2$ except with negligible probability; this follows from the information-theoretic security of the MAC along with the fact that the execution of Π is independent of k_2', t_2'.

Let i denote the first round in which $\mathsf{Vrfy}_{k'_2}(a_i, t'_2) = 1$, and let j denote the first round in which $\mathsf{Vrfy}_{k'_1}(b_j, t'_1) = 1$. Assuming for simplicity that Π has perfect correctness, we have

$$\Pr[i \leq j] + \Pr[j > i] = 1.$$

Further, since $\left|\Pr[P_1^* \text{ wins}] - \Pr[i \leq j]\right|$ and $\left|\Pr[P_2^* \text{ wins}] - \Pr[i > j]\right|$ are both negligible, we see that either P_1^* or P_2^* wins in the real world with probability at least $1/2 - \mathsf{negl}(n)$. Since an adversary wins in the ideal world with negligible probability, this rules out $\frac{1}{p}$-security for $p > 2$.

Theorem 4 does not contradict the results of [12], or any previous work on fair exchange of signatures. One reason is that prior work on fair exchange typically assumes that each party has no auxiliary information about the other party's secret, whereas our definition (as is standard for definitions of secure computation) accounts for this possibility.[3] Also, in some previous work on fair exchange the running time of the honest party is not bounded by a fixed polynomial, whereas in our setting we require this to be the case.

5 Conclusions and Open Questions

Our work offers a clean definition of partial fairness within the standard real/ideal world paradigm, and settles the question of the general feasibility of achieving this notion in the two-party setting. Several compelling questions remain:

- An easy modification of our second impossibility result (cf. Theorem 4) rules out our definition of partial fairness for the interesting special case of exchanging digital signatures. What is the appropriate (simulation-based?) notion of partial fairness for that setting?
- We can show a function $\mathcal{F} = \{f_n : X_n \times Y_n \to Z_n\}$ for which any protocol computing $\mathcal{F}$ with $\frac{1}{p}$-security requires $\min\{p, |X_n|, |Y_n|\}$ rounds. This leaves a gap as compared to Theorem 1.
- The question of partial fairness in the multi-party setting (with dishonest majority) is wide open. We are not aware of any results in this direction except for the case of coin tossing [10,27], or functions where complete fairness is possible [21].

References

1. Aumann, Y., Lindell, Y.: Security against covert adversaries: Efficient protocols for realistic adversaries. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 137–156. Springer, Heidelberg (2007)

[3] Our proof of Theorem 4 exploits this. We can prove an analogue of Theorem 4, based on the assumption that one-way functions exist, that rules out partial fairness even if the adversary has no auxiliary information.

2. Beaver, D.: Foundations of secure interactive computing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 377–391. Springer, Heidelberg (1992)
3. Beaver, D., Goldwasser, S.: Multiparty computation with faulty majority. In: 30th Annual Symposium on Foundations of Computer Science (FOCS), pp. 468–473. IEEE, Los Alamitos (1989)
4. Ben-Or, M., Goldreich, O., Micali, S., Rivest, R.: A fair protocol for signing contracts. IEEE Trans. Information Theory 36(1), 40–46 (1990)
5. Blum, M.: How to exchange (secret) keys. ACM Transactions on Computer Systems 1, 175–193 (1984)
6. Boneh, D., Naor, M.: Timed commitments. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 236–254. Springer, Heidelberg (2000)
7. Cachin, C., Camenisch, J.: Optimistic fair secure computation. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 93–111. Springer, Heidelberg (2000)
8. Canetti, R.: Security and composition of multiparty cryptographic protocols. Journal of Cryptology 13(1), 143–202 (2000)
9. 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, Los Alamitos (2001)
10. Cleve, R.: Limits on the security of coin flips when half the processors are faulty. In: 18th Annual ACM Symposium on Theory of Computing (STOC), pp. 364–369. ACM Press, New York (1986)
11. Cleve, R.: Controlled gradual disclosure schemes for random bits and their applications. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 573–588. Springer, Heidelberg (1990)
12. Damgård, I.: Practical and provably secure release of a secret and exchange of signatures. Journal of Cryptology 8(4), 201–222 (1995)
13. Dwork, C., Naor, M., Sahai, A.: Concurrent zero-knowledge. Journal of the ACM 51(6), 851–898 (2004)
14. Even, S., Goldreich, O., Lempel, A.: A randomized protocol for signing contracts. Comm. ACM 28(6), 637–647 (1985)
15. Franklin, M.: Complexity and Security of Distributed Protocols. PhD thesis, Columbia University (1993)
16. Galil, Z., Haber, S., Yung, M.: Cryptographic computation: Secure faut-tolerant protocols and the public-key model. In: Pomerance, C. (ed.) CRYPTO 1987. LNCS, vol. 293, pp. 135–155. Springer, Heidelberg (1988)
17. Garay, J.A., MacKenzie, P.D., Prabhakaran, M., Yang, K.: Resource fairness and composability of cryptographic protocols. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 404–428. Springer, Heidelberg (2006)
18. Goldreich, O.: Foundations of Cryptography. Basic Applications, vol. 2. Cambridge University Press, Cambridge (2004)
19. Goldreich, O., Lindell, Y.: Session-key generation using human passwords only. Journal of Cryptology 19(3), 241–340 (2006)
20. Goldwasser, S., Levin, L.A.: Fair computation of general functions in presence of immoral majority. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 77–93. Springer, Heidelberg (1991)
21. Gordon, S.D., Katz, J.: Complete fairness in multi-party computation without an honest majority. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 19–35. Springer, Heidelberg (2009)
22. Gordon, S.D., Hazay, C., Katz, J., Lindell, Y.: Complete fairness in secure two-party computation. In: 40th Annual ACM Symposium on Theory of Computing (STOC), pp. 413–422. ACM Press, New York (2008)

23. Katz, J.: On achieving the "best of both worlds" in secure multiparty computation. In: 39th Annual ACM Symposium on Theory of Computing (STOC), pp. 11–20. ACM Press, New York (2007)
24. Lepinski, M., Micali, S., Peikert, C., Shelat, A.: Completely fair SFE and coalition-safe cheap talk. In: 23rd ACM Symposium Annual on Principles of Distributed Computing, pp. 1–10. ACM Press, New York (2004)
25. Luby, M., Micali, S., Rackoff, C.: How to simultaneously exchange a secret bit by flipping a symmetrically-biased coin. In: 24th Annual Symposium on Foundations of Computer Science (FOCS), pp. 23–30. IEEE, Los Alamitos (1983)
26. Micali, S., Rogaway, P.: Secure computation. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 392–404. Springer, Heidelberg (1992)
27. Moran, T., Naor, M., Segev, G.: An optimally fair coin toss. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 1–18. Springer, Heidelberg (2009)
28. Pinkas, B.: Fair secure two-party computation. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 87–105. Springer, Heidelberg (2003)
29. Yao, A.C.-C.: How to generate and exchange secrets. In: 27th Annual Symposium on Foundations of Computer Science (FOCS), pp. 162–167. IEEE, Los Alamitos (1986)

Secure Message Transmission with Small Public Discussion

Juan Garay[1], Clint Givens[2], and Rafail Ostrovsky[3,⋆]

[1] AT&T Labs – Research
garay@research.bell-labs.com
[2] Department of Mathematics, UCLA
cgivens@math.ucla.edu
[3] Departments of Computer Science and Mathematics, UCLA
rafail@cs.ucla.edu

Abstract. In the problem of Secure Message Transmission in the public discussion model (SMT-PD), a Sender wants to send a message to a Receiver privately and reliably. Sender and Receiver are connected by n channels, up to $t < n$ of which may be maliciously controlled by a computationally unbounded adversary, as well as one public channel, which is reliable but not private.

The SMT-PD abstraction has been shown instrumental in achieving secure multi-party computation on sparse networks, where a subset of the nodes are able to realize a broadcast functionality, which plays the role of the public channel. However, the *implementation* of such public channel in point-to-point networks is highly costly and non-trivial, which makes minimizing the use of this resource an intrinsically compelling issue.

In this paper, we present the first SMT-PD protocol with *sublinear* (i.e., logarithmic in m, the message size) communication on the public channel. In addition, the protocol incurs a private communication complexity of $O(\frac{mn}{n-t})$, which, as we also show, is *optimal*. By contrast, the best known bounds in both public and private channels were linear. Furthermore, our protocol has an optimal round complexity of $(3, 2)$, meaning three rounds, two of which must invoke the public channel.

Finally, we ask the question whether some of the lower bounds on resource use for a single execution of SMT-PD can be beaten *on average* through amortization. In other words, if Sender and Receiver must send several messages back and forth (where later messages depend on earlier ones), can they do better than the naïve solution of repeating an SMT-PD protocol each time? We show that amortization can indeed drastically reduce the use of the public channel: it is possible to limit the total number of uses of the public channel to *two*, no matter how many messages are ultimately sent between two nodes. (Since two uses of the public channel are required to send any reliable communication whatsoever, this is best possible.)

1 Introduction

Dolev, Dwork, Waarts and Yung [DDWY93] introduced the model of *Secure Message Transmission* (SMT) in an effort to understand the connectivity requirements for secure

[⋆] Supported in part by IBM Faculty Award, Xerox Innovation Group Award, the Okawa Foundation Award, Intel, Teradata, NSF grants 0716835, 0716389, 0830803, 0916574, BSF grant 2008411 and U.C. MICRO grant.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 177–196, 2010.

communication in the information-theoretic setting. Generally speaking, an SMT protocol involves a sender, $\mathcal{S}$, who wishes to transmit a message M to a receiver, $\mathcal{R}$, using a number n of channels ("wires"), some of which are controlled by a malicious adversary $\mathcal{A}$. The goal is to send the message both *privately* and *reliably*. Since its introduction, SMT has been widely studied and optimized with respect to several different settings of parameters (for example, see [SA96, SNP04, ACH06, FFGV07, KS08]).

Garay and Ostrovsky [GO08] studied a model they called Secure Message Transmission *by Public Discussion* (SMT-PD) as an important building block for achieving secure multi-party computation [BGW88, CCD88] on sparse (i.e., not fully connected) networks. (An equivalent setup was studied earlier in a different context by Franklin and Wright [FW98].) In this model, in addition to the wires in the standard SMT formulation, called "common" or "private" wires from now on, $\mathcal{S}$ and $\mathcal{R}$ gain access to a *public* channel which the adversary can read but not alter. In this new setting, secure message transmission is achievable even if the adversary corrupts up to $t < n$ of the private wires—i.e., up to all but one.

The motivation for this abstraction comes from the feasibility in partially connected settings for a subset of the nodes in the network to realize a broadcast functionality despite the limited connectivity [DPPU86, Upf92, BG93][1], which plays the role of the public channel. (The private wires would be the multiple paths between them.) As such, the *implementation* of the public channel in point-to-point networks is costly and highly non-trivial in terms of rounds of computation and communication, as already the sending of a single message to a node that is not directly connected is simulated by sending the message over multiple paths, not just blowing up the communication but also incurring a slowdown factor proportional to the diameter of the network, and this is a process that must be repeated many times—linear in the number of corruptions for deterministic, error-free broadcast protocols (e.g., [GM98]), or expected (but high) constant for randomized protocols [FM97, KK06].

A main goal of this work is to minimize the use of this expensive resource, both in terms of communication as well as in the number of times it must be used when sender and receiver must send many messages back and forth, as it is the case in secure multi-party computation. We first present an SMT-PD protocol with a logarithmic (in m, the message size) communication complexity on the public channel; the best known bound, due to Shi, Jiang, Safavi-Naini, and Tuhin [SJST09], was linear (see related work below). In addition, our protocol incurs a private communication complexity of $O(\frac{mn}{n-t})$, which, as we also show, is *optimal*, thus providing an affirmative answer to the question posed in [SJST09] of whether their $O(mn)$ private communication could be improved. Furthermore, our protocol has an optimal round complexity of $(3, 2)$, meaning 3 rounds, 2 of which must invoke the public channel [SJST09].

Regarding the number of times the public channel must be used when considering SMT-PD as a subroutine in a larger protocol, we ask the question whether some of the lower bounds on resource use for a single execution of SMT-PD can be beaten *on average* through amortization. In other words, if a sender and receiver must send several messages back and forth (where later messages depend on earlier ones), can they do better than the naïve solution of repeating an SMT-PD protocol each time, incurring a cost of three

[1] Called "almost-everywhere" agreement, or broadcast, in this setting.

rounds and two public channel transmissions per message? We show that amortization can in fact drastically reduce the use of the public channel: indeed, it is possible to limit the total number of uses of the public channel to *two*, no matter how many messages are ultimately sent between two nodes. (Since two uses of the public channel are required to send any reliable communication whatsoever, this is best possible.)

Prior work. The first variant of SMT considered in the literature is *perfectly secure message transmission* (PSMT), in which both privacy and reliability are perfect [DDWY93]. It is shown in the original paper that PSMT is possible if and only if $n \geq 2t+1$. For such n, 2 rounds are necessary and sufficient for PSMT, while one-round PSMT is possible if and only if $n \geq 3t+1$.

The communication complexity of PSMT depends on the number of rounds. For 1-round PSMT, Fitzi *et al.* [FFGV07] show that transmission rate $\geq \frac{n}{n-3t}$ is necessary and sufficient. (Recall that $n > 3t$ is required in this case.) For 2-round PSMT, Srinathan *et al.* [SNP04] show that a transmission rate $\geq \frac{n}{n-2t}$ is required[2]; this was extended in [SPR07], which showed that increasing the number of rounds does not help. Kurosawa and Suzuki [KS08] construct the first efficient (i.e., polynomial-time) 2-round PSMT protocol which matches this optimal transmission rate.

A number of relaxations of the perfectness requirements of PSMT are considered in the literature to achieve various tradeoffs (see for example [CPRS08] for a detailed discussion of variants of SMT). The most general version of SMT (or SMT-PD) is perhaps (ϵ, δ)-SMT. We call a protocol for SMT(-PD) an (ϵ, δ)-SMT(-PD) protocol provided that the adversary's advantage in distinguishing any two messages is at most ϵ, and the receiver correctly outputs the message with probability $1-\delta$. The lower bound $n \geq 2t+1$ holds even in this general setting (at least for non-trivial protocols, such as those satisfying $\epsilon + \delta < 1/2$); hence the most interesting case for SMT-PD is the case when the public channel is required: $t < n \leq 2t$. As noted above, this requires round complexity (3,2) [SJST09]. Franklin and Wright [FW98] show that perfectly reliable ($\delta = 0$) SMT-PD protocols are impossible when $n \leq 2t$. On the other hand, perfect privacy ($\epsilon = 0$) is possible, and is achieved by previous SMT-PD constructions (see below).

The communication complexity lower bounds noted above all apply to PSMT; for more general SMT bounds, we are aware only of [KS07]. They consider the problem of *almost-secure message transmission*, which is only slightly less restrictive than PSMT. Namely, the problem requires perfect privacy, and that the Receiver *never* output an incorrect message, though he may output "failure" with probability δ. The authors show that in this model, there is a communication complexity lower bound of $n(m + \log(1/\delta))$ (up to an additive constant).

A number of protocols for SMT-PD appear in previous work. The first such comes in [FW98] as a consequence of the equivalence shown there between networks with multicast and those with simple lines and broadcast (i.e., the public discussion model). Their solution has optimal round complexity $(3, 2)$[3]; however, when $t < n < \lceil \frac{3t}{2} \rceil$

[2] The authors claim a matching upper bound as well, but this was shown to be flawed [ACH06].

[3] The round complexity is not apparent from the text, for two reasons: (1) The protocol is described in terms of the multicast model, not SMT-PD directly; and (2) the authors consider synchronous "rounds" not in the abstract SMT-PD model, but in the more concrete setting of nodes relaying messages in the underlying network.

(including the worst case $t = n + 1$), their protocol has (pick your poison) either positive privacy error $\epsilon > 0$, or *exponential* communication complexity. Garay and Ostrovsky [GO08] first describe a (4,3)-round $(0, \delta)$ protocol which was subsequently improved to (3,2) rounds. The protocol has linear transmission rate (in terms of message size) on the public and private channels. Shi *et al.* [SJST09] give the first protocol with constant transmission rate on the public channel (for messages of sufficient, modest size) with linear transmission rate on the private channels as well; however, the *communication complexity* of their protocol is linear.

Our contributions. By contrast, we obtain the first round-optimal SMT-PD protocol with *sublinear* (logarithmic) communication complexity on the public channel. More specifically (and assuming for simplicity $\delta = O(1)$), our protocol has public channel communication complexity $O(n \log n \log m)$ for messages of sufficient size, as compared with $O(m)$ in the protocol of [SJST09]. (The message size required by either protocol—namely, $m/ \log m = \Omega(n \log n)$ for ours, or $m = \Omega(n^2)$ for that of [SJST09]—ensures that $O(n \log n \log m)$ improves over $O(m)$ for relevant values of n, m.) The protocol also enjoys a private communication complexity of $O(\frac{nm}{n-t})$, which (just by itself) improves on previous constructions and, as we also show, is optimal. At a high level, the protocol has the same structure as previous 3-round SMT-PD protocols, with the following important differences: (1) our use of randomness extractors allows us to reduce the amount of transmitted randomness, which is reflected in the gain in private communication, and (2) typically in previous protocols the message is transmitted in the last round over the public channel, blinded by the private randomness thought not to have been tampered with; our improvement to public communication comes from the transmission of the (blinded) message on the *private* wires, provided that the sender *authenticates* the transmission making use of the public channel, which in turn requires smaller communication. Additionally, we achieve these improved communication bounds even for messages of smaller required size than Shi *et al.* [SJST09].[4] Finally, the protocol achieves perfect privacy.

We arrive at this result through a series a transformations. First, we design a generic SMT-PD protocol with linear public communication and $O(\frac{nm}{n-t})$ private communication (note that this already improves on existing results); second, we consider instantiations of the generic protocol's "black boxes" with different randomness extractors, each providing its own benefits (perfect privacy *vis-à-vis* smaller message size); and last, we obtain the final protocol by essentially running two perfect-privacy instantiations of the generic protocol in parallel, one for the message itself and a "smaller" version for the authentication key. These results are presented in Section 3.

As noted above, we also show (Section 4) an $\Omega(\frac{nm}{n-t})$ lower bound on private communication. The lower bound holds for SMT without public discussion as well. The bound itself is weaker than previous, but it holds for a more general class of SMT protocols. In particular, it is the first communication complexity lower bound to consider non-perfect privacy, as well as the first to allow for the Receiver outputting an incorrect message.

[4] Specifically, [SJST09] require message size $m = \Omega(n^2(\log(1/\delta))^2)$, where we require only $m = \Omega(n(\log n + \log(1/\delta)) \log q)$, with $q \approx mn/(n - t)$.

Finally, we show in Section 5 how amortization can drastically reduce the use of the public channel, allowing sender and receiver to communicate *indefinitely* after using the public channel twice and a limited initial message. Our approach is to separate Sender and Receiver's interaction following the first execution of SMT-PD into two modes: a *Normal Mode* and a *Fault-Recovery Mode*. At a high level, in the Normal Mode, secure communication is successful provided the adversary does not interfere; this is implemented by a one-round protocol satisfying a relaxed version of the problem that we call *Weak* SMT-PD. Fault-Recovery Mode is entered if corruption is detected.[5]

Preliminaries and definitions are given in Section 2. Due to space limitations, most of the proofs, as well as additional background material, are given in the full version of the paper [GGO09].

2 Model and Preliminaries

Definition 1. *If X and Y are random variables over a discrete space S, the* statistical distance between X and Y *is defined to be*

$$\Delta(X,Y) \stackrel{\text{def}}{=} \frac{1}{2}\sum_{s\in S} \left|\Pr[X=s]-\Pr[Y=s]\right| .$$

We say that X and Y are ϵ-close *if $\Delta(X,Y)\leq\epsilon$.*

The public discussion model. The *public discussion model* for secure message transmission [GO08] consists of a Sender $\mathcal{S}$ and Receiver $\mathcal{R}$ (PPTMs) connected by n communication channels, or *wires*, and one *public channel*. $\mathcal{S}$ wishes to send a message $M_{\mathcal{S}}$ from message space $\mathcal{M}$ to $\mathcal{R}$, and to this end $\mathcal{S}$ and $\mathcal{R}$ communicate with each other in synchronous rounds in which one player sends information across the wires and/or public channel. Communication on the public channel is reliable but public; the common wires may be corrupted and so are not necessarily reliable or private.

$\mathcal{A}$ is a computationally unbounded adversary who seeks to disrupt the communication and/or gain information on the message. $\mathcal{A}$ may *adaptively* corrupt up to $t<n$ of the common wires (potentially all but one!). Corrupted wires are actively controlled by $\mathcal{A}$: he can eavesdrop, block communication, or place forged messages on them. Further, we assume $\mathcal{A}$ is *rushing*—in each round, he observes what is sent on the public channel and all corrupted wires before deciding what to place on corrupted wires, or whether to corrupt additional wires (which he then sees immediately).

An *execution* E of an SMT-PD protocol is determined by the random coins of $\mathcal{S}$, $\mathcal{R}$, and $\mathcal{A}$ (which we denote $C_{\mathcal{S}}$, $C_{\mathcal{R}}$, $C_{\mathcal{A}}$ respectively), and the message $M_{\mathcal{S}}\in\mathcal{M}$. The *view of a player* $\mathcal{P}\in\{\mathcal{S},\mathcal{R},\mathcal{A}\}$ *in an execution* E, denoted $\text{View}_{\mathcal{P}}$, is a random variable consisting of $\mathcal{P}$'s random coins and all messages received (or overheard) by $\mathcal{P}$. ($\mathcal{S}$'s view also includes $M_{\mathcal{S}}$). Additionally, let $\text{View}_{\mathcal{P}}(M_0)$ denote the distribution on $\text{View}_{\mathcal{P}}$ induced by fixing $M_{\mathcal{S}}=M_0$. In each execution, $\mathcal{R}$ outputs a received message $M_{\mathcal{R}}$, a function of $\text{View}_{\mathcal{R}}$.

[5] Effectively, this is an instantiation in the SMT context of the "fast-track" approach (e.g., [Lam87, GRR98]), where if things are "smooth" then the algorithm or protocol performs very efficiently, reverting to a more punctilious mode otherwise.

We can now define an (ϵ, δ)-SMT-PD protocol (cf. [FW98, GO08, SJST09]):

Definition 2. *A protocol Π in the model above, in which $\mathcal{S}$ attempts to send a message $M_{\mathcal{S}}$ to $\mathcal{R}$, is* (ϵ, δ)-secure *(or simply, is an* (ϵ, δ)-SMT-PD protocol*) if it satisfies:*

PRIVACY*: For any two messages $M_0, M_1 \in \mathcal{M}$,* $\mathrm{View}_{\mathcal{A}}(M_0)$ *and* $\mathrm{View}_{\mathcal{A}}(M_1)$ *are ϵ-close.*

RELIABILITY*: For all $M_{\mathcal{S}} \in \mathcal{M}$ and all adversaries $\mathcal{A}$, $\mathcal{R}$ should correctly receive the message with probability at least $1 - \delta$; i.e.,* $\Pr[M_{\mathcal{R}} = M_{\mathcal{S}}] \geq 1 - \delta$*. (The probability is taken over all players' random coins.)*

Error-correcting codes and consistency checks for codewords. For our purposes, the following definition of error-correcting codes is sufficient:

Definition 3. *Given a finite alphabet Σ, an* error-correcting code $\mathcal{E}$ of minimum distance *d is a pair of mappings $Enc : \Sigma^K \to \Sigma^N$, where $K < N$ and $Dec : \Sigma^N \to \Sigma^K$, such that (1) any two distinct elements x, y in the image of Enc (the* codewords*) have $dist(x, y) \geq d$ in the Hamming metric; (2) $Dec(Enc(x)) = x$ for all $x \in \mathbb{F}_q^K$.*[6] *We say $\mathcal{E}$ has* rate K/N *and* relative minimum distance d/N.

We require a family of codes of increasing input length which is *asymptotically good*, that is, $\mathcal{E}$ should have *constant* rate and *constant* relative minimum distance D. See, e.g., [MS83] for a standard reference.

Of particular interest for us are the well-known Reed-Solomon codes over F_q, obtained by oversampling polynomials in $\mathbb{F}_q[X]$. Given an input in $\mathbb{F}_q^K$, we interpret it as a polynomial f of degree $\leq K - 1$; to obtain a codeword from f, we simply evaluate it at N distinct points in $\mathbb{F}_q$, for any $N > K$. Indeed, any two such polynomials agree on at most $K - 1$ points, therefore the Reed-Solomon code has minimum distance $N - K + 1$.

Our protocols make use of a simple method to probabilistically detect when codewords sent on the private wires are altered by $\mathcal{A}$. Simply put, the sender of the codeword reveals a small subset of the codeword symbols. Formally, suppose $\mathcal{S}$ sends a codeword $\mathcal{C} \in \Sigma^N$ to $\mathcal{R}$ over one of the private wires, and $\mathcal{R}$ receives the (possibly altered) codeword $\mathcal{C}^*$. (If $\mathcal{R}$ receives a non-codeword, he immediately rejects it.) Then to perform the consistency check, $\mathcal{S}$ chooses a random set $J = \{j_1, j_2, \ldots, j_\ell\} \subset [N]$ and sends $(J, \mathcal{C}|_J)$ to $\mathcal{R}$, where $\mathcal{C}|_J$ represents the codeword $\mathcal{C}$ restricted to the indices in J. If the revealed symbols match, then the consistency check succeeds; otherwise the check fails and $\mathcal{R}$ rejects $\mathcal{C}^*$ as tampered.

Suppose $\mathcal{A}$ alters $\mathcal{C}$ to a different codeword, $\mathcal{C}^* \neq \mathcal{C}$. Since $\mathcal{C}$ and $\mathcal{C}^*$ are distinct valid codewords, they differ in at least, say, 1/3 of their symbols. Therefore, the probability that they agree on a randomly chosen index is $\leq 2/3$, and so

$$\Pr[\mathcal{R} \text{ accepts } \mathcal{C}^*] = \Pr[\mathcal{C}|_J = \mathcal{C}^*|_J] \leq (2/3)^\ell.$$

Thus, with probability $\geq 1 - (2/3)^\ell$, $\mathcal{R}$ will reject a tampered codeword. Of course, the validity of the check depends upon $\mathcal{A}$ not knowing J at the time of potential corruption of $\mathcal{C}$.

[6] Note in particular that this allows us to test for membership in the image $Enc(\Sigma^K)$ by first decoding and then re-encoding.

Average min-entropy and average-case randomness extractors. Recall that the *min-entropy* of a distribution $X = (X_1, \ldots, X_N)$ over $\{0,1\}^N$ is defined as

$$H_\infty(X) = \min_x \left(-\log\left(\Pr[X = x]\right)\right),$$

and gives a measure of the amount of randomness "contained" in a weakly random source. We say a distribution X is a k_{min}*-source* if $H_\infty(X) \geq k_{min}$.

A *(seeded)* $(N, M, k_{min}, \epsilon)$*-strong extractor* is a (deterministic) function

$$\mathrm{Ext} : \{0,1\}^N \times \{0,1\}^D \to \{0,1\}^M$$

such that for *any* k_{min}-source X, the distribution $U_D \circ \mathrm{Ext}(X, U_D)$ is ϵ-close to $U_D \circ U_M$ (where U_k represents the uniform distribution on $\{0,1\}^k$). The input to the extractor is the N-bit k_{min}-source, X, together with a truly random seed s, which is uniformly distributed over $\{0,1\}^D$. Its output is an M-bit string which is statistically close to uniform, *even conditioned on the seed s used to generate it.*

This notion of min-entropy, and of a general randomness extractor, may be an awkward fit when considering an adversary with side information Y as above. In these cases, a more appropriate measure may be found in the *average min-entropy* of X given Y, defined in [DORS08] by

$$\tilde{H}_\infty(X \mid Y) = -\log\left(\mathbb{E}_{y \leftarrow Y}\left[\max_x \Pr\left[X = x \mid Y = y\right]\right]\right).$$

Note that this definition is based on the *worst-case* probability for X, conditioned on the *average distribution* (as opposed to worst-case probability) of Y. The rationale is that Y is assumed to be outside of the adversary's control; however, once Y is known, the adversary then predicts the *most likely* X, given that particular Y.

[DORS08] use average min-entropy to define an object closely related to extractors: A *(seeded) average-case* $(N, M, k_{min}, \epsilon)$*-strong extractor* is a (deterministic) function

$$\mathrm{Ext} : \{0,1\}^N \times \{0,1\}^D \to \{0,1\}^M$$

such that the distribution of $(U_D \circ \mathrm{Ext}(X, U_D), I)$ is ϵ-close to $(U_D \circ U_M, I)$, whenever (X, I) is a jointly distributed pair satisfying $\tilde{H}_\infty(X \mid I) \geq k_{min}$. The similarity to an ordinary extractor is clear. [DORS08] prove the following fact about average min-entropy:

Fact 1. *If Y has at most 2^ℓ possible values, then $\tilde{H}_\infty(X \mid (Y, Z)) \geq \tilde{H}_\infty(X \mid Z) - \ell$.*

Extracting randomness from $\mathbb{F}_q$. We will make use of a special-purpose *deterministic* (seedless) extractor Ext_q which operates at the level of field elements in $\mathbb{F}_q$ as opposed to bits. Ext_q works not on general min-entropy sources, but on the restricted class of *symbol-fixing sources*, which are strings in $\mathbb{F}_q^N$ such that some subset of K symbols is distributed independently and uniformly over $\mathbb{F}_q$, while the remaining $N - K$ symbols are fixed. Given a sample from any such source, Ext_q outputs K field elements which are uniformly distributed over $\mathbb{F}_q^K$.

Ext_q works as follows: Given $\alpha \in \mathbb{F}_q^N$, construct $f \in \mathbb{F}_q[X]$ of degree $\leq N - 1$, such that $f(i) = \alpha_i$ for $i = 0, \ldots, N-1$. Then $\mathrm{Ext}_q(\alpha) = (f(N), f(N+1), \ldots, f(N+K-1))$. (Of course we require $N + K \leq q$.) This extractor has proven useful in previous SMT protocols as well (see, e.g., [ACH06, KS08]).

3 SMT-PD with Small Public Discussion

In this section we present our main positive results. First, we construct a basic (ϵ, δ)-SMT-PD protocol, Π_{Gen} (for "generic"), with optimal private communication and linear public communication. We then consider possible instantiations of Π_{Gen}; using, in particular, Reed-Solomon codes and the extractor Ext_q, improves it to a 0-private protocol. Finally, we use Π_{Gen} (instantiated with Reed-Solomon codes) as a building block to construct our main protocol Π_{SPD}, which achieves *logarithmic* public communication while maintaining optimal private communication (and other desirable properties).

3.1 A Generic Protocol with Optimal Private Communication

Protocol Π_{Gen} achieves essentially optimal communication complexity on the private wires of $O(\frac{mn}{n-t})$, where m is the length of the message, while maintaining linear communication complexity on the public channel. (See Section 4 for a precise statement of the lower bound.) This is the first SMT-PD protocol to achieve sublinear transmission rate on the private wires, and as such provides an affirmative answer to the question posed in [SJST09] of whether $O(n)$ private-wire transmission rate can be improved.

Π_{Gen} relies on two primitives as black boxes: an error-correcting code $\mathcal{E}$ and an average-case strong extractor, Ext_A. The efficiency of the protocol depends on the interaction between the basic parameters of the protocol—ϵ, δ, m, n, and t—and the parameters of $\mathcal{E}$ and Ext_A. After presenting the protocol and proving its security, we will examine its complexity in terms of these parameters.

At a high level, the protocol has the same structure as previous 3-round SMT-PD protocols: (1) in the first round, one of the parties (in our case $\mathcal{R}$) sends lots of randomness on each private wire; (2) using the public channel, $\mathcal{R}$ then sends checks to verify the randomness sent in (1) was not tampered with; (3) $\mathcal{S}$ discards any tampered wires, combines each remaining wire's randomness to get a one-time pad R, and sends $C = M \oplus R$ on the public channel. However, our use of extractors allows us to reduce the amount of transmitted randomness, which is reflected in the gain in private communication.

We remark that one may modify Π_{Gen} to have interaction order $\mathcal{S}$-$\mathcal{R}$-$\mathcal{S}$, instead of $\mathcal{R}$-$\mathcal{R}$-$\mathcal{S}$ as we present it. One advantage of $\mathcal{R}$-$\mathcal{R}$-$\mathcal{S}$ is that when instantiated with deterministic extractors (see below), it does not require any random coins for $\mathcal{S}$ (in contrast to $\mathcal{S}$-$\mathcal{R}$-$\mathcal{S}$, where both parties use randomness crucially).

Now we turn to the details of protocol Π_{Gen}. Let error-correcting code $\mathcal{E}$ have encoding and decoding functions $Enc : \{0,1\}^K \to \{0,1\}^N$ and $Dec : \{0,1\}^N \to \{0,1\}^K$, respectively, and relative minimum distance D. (We will specify K below.) While $N > K$ may be arbitrarily large for the purpose of correctness, we will want K/N and D both to be constant for our complexity analysis—that is, we want $\mathcal{E}$ to be asymptotically good.

Second, let Ext_A be an average-case $(nK, m, k_{min}, \epsilon/2)$-strong extractor. Here K is, as above, the source length of the error-correcting code $\mathcal{E}$, and m and ϵ are the message-length and privacy parameters of Π_{Gen}. k_{min} is the min-entropy threshold. Now clearly $m \leq k_{min} \leq nK$. On the other hand, we require $k_{min} = O(m)$ for our complexity claim to hold—that is, Ext_A should extract a constant fraction of the min-entropy. Further, the extractor's seed length s should be $O(n+m)$.

Finally, let $b = \frac{1}{1-D}$, and then set $\ell = \lceil \log_b(t/\delta) \rceil$. Now with foresight, we set $K = \lceil k_{min}/(n-t) \rceil + \ell$.[7] Note that if $k_{min} = O(m)$, then $K = O(m)/(n-t) + \ell$. The protocol, Π_{Gen}, is presented in Fig. 1.

Protocol $\Pi_{\mathrm{Gen}}(\epsilon, \delta, m, n, t, \mathcal{E}, \mathrm{Ext}_A)$

1. $(\mathcal{R} \overset{PRI}{\to} \mathcal{S})$. For each wire i, $\mathcal{R}$ chooses a random $r_i \in \{0,1\}^K$ and sends the codeword $\mathcal{C}_i = Enc(r_i)$ along wire i. Let $\mathcal{C}_i^*$ be the codeword received by $\mathcal{S}$, and $r_i^* = Dec(\mathcal{C}_i^*)$.
2. $(\mathcal{R} \overset{PUB}{\to} \mathcal{S})$. $\mathcal{R}$ chooses a random subset $J = \{j_1, j_2, \ldots, j_\ell\} \subset [N]$ of codeword indices, $|J| = \ell$. Let
$$\mathcal{C}_i|_J = (\mathcal{C}_{i,j_1}, \mathcal{C}_{i,j_2}, \ldots, \mathcal{C}_{i,j_\ell}) \in \{0,1\}^\ell$$
be the codeword $\mathcal{C}_i$ restricted to the indices of J. $\mathcal{R}$ sends $(J, \{\mathcal{C}_i|_J\}_{i\in[n]})$ to $\mathcal{S}$ over the public channel.
3. $(\mathcal{S} \overset{PUB}{\to} \mathcal{R})$. $\mathcal{S}$ rejects any wire i which is syntactically incorrect (including the case that $\mathcal{C}_i^*$ is not a valid codeword), or for which $\mathcal{C}_i|_J$ conflicts with $\mathcal{C}_i^*$. Call the set of remaining, accepted wires ACC, and let $B \in \{0,1\}^n$, where $b_i = 1 \iff i \in \mathsf{ACC}$. Let α^* denote the concatenation of r_i^* for all $i \in \mathsf{ACC}$, padded with zeroes so that $|\alpha^*| = nK$. $\mathcal{S}$ chooses $seed \in \{0,1\}^s$ uniformly at random. He applies $\mathrm{Ext}_A : \{0,1\}^{nK} \times \{0,1\}^s \to \{0,1\}^m$ to obtain $R^* = \mathrm{Ext}_A(\alpha^*, seed)$, where $|R^*| = m$. $\mathcal{S}$ puts $C = M_{\mathcal{S}} \oplus R^*$, and sends $(B, C, seed)$ on the public channel.
Receiver: $\mathcal{R}$ uses B to reconstruct ACC. He forms α by concatenating r_i for each $i \in \mathsf{ACC}$, and padding with zeroes to size nK. He applies $\mathrm{Ext}_A : \{0,1\}^{nK} \to \{0,1\}^m$, obtaining $R = \mathrm{Ext}_A(\alpha, seed)$. He then recovers $M_{\mathcal{R}} = C \oplus R$.

Fig. 1. A generic SMT-PD protocol with optimal communication complexity on the private wires and linear communication complexity on the public channel

Theorem 2. *Let $t < n$. Protocol Π_{Gen} is a $(3,2)$-round (ϵ,δ)-SMT-PD protocol with communication complexity $O(\frac{mn}{n-t})$ on the private wires provided that $m/(n-t) = \Omega(\log(t/\delta))$, and communication complexity $\max(O(\log(t/\delta)(n+\log m)), O(m+n))$ on the public channel, provided only that $m = \Omega(\log(t/\delta))$.*

Proof. **Privacy.** We first claim that if we omit C, then $\mathcal{A}$ has essentially no information (up to ϵ) on $\mathcal{S}$'s output of the average-case extractor, $R* = \mathrm{Ext}_A(\alpha^*, seed)$. Formally:

Claim. The distribution $(U_s, R^*, \mathrm{View}_{\mathcal{A}} \setminus C)$ is $\epsilon/2$-close to $(U_s, U_m, \mathrm{View}_{\mathcal{A}} \setminus C)$.

The remainder of the proof of ϵ-privacy is by contradiction: We show that, if there exists an adversary $\mathcal{A}$ and messages M_0, M_1 such that $\Delta(\mathrm{View}_{\mathcal{A}}(M_0), \mathrm{View}_{\mathcal{A}}(M_1)) > \epsilon$,

[7] As a sanity check, observe that $k_{min} \leq nK = n(k_{min}/(n-t) + \ell)$, so the extractor we define can exist.

then there exists a distinguisher $\mathcal{D}$ which can distinguish $(U_s, R^*, \mathrm{View}_{\mathcal{A}} \setminus C)$ from $(U_s, U_m, \mathrm{View}_{\mathcal{A}} \setminus C)$, in contradiction to the above claim.

So suppose such an $\mathcal{A}$, M_0, M_1 exist. Then there exists a distinguisher $\mathcal{D}_0$ which satisfies

$$|\Pr[D_0(\mathrm{View}_{\mathcal{A}}(M_0)) = 1] - \Pr[D_0(\mathrm{View}_{\mathcal{A}}(M_1)) = 1]| > \epsilon$$

In particular it follows that either

$$(1) \qquad \big|\Pr[D_0(\mathrm{View}_{\mathcal{A}}(M_0)) = 1] - \Pr[D_0(\mathrm{View}_{\mathcal{A}}(M_\$)) = 1]\big| > \epsilon/2$$

or

$$(2) \qquad \big|\Pr[D_0(\mathrm{View}_{\mathcal{A}}(M_\$)) = 1] - \Pr[D_0(\mathrm{View}_{\mathcal{A}}(M_1)) = 1]\big| > \epsilon/2.$$

Here $\mathrm{View}_{\mathcal{A}}(M_\$)$ denotes the random variable obtained by first sampling $M_\$$ uniformly from $\{0,1\}^m$, and then sampling from $\mathrm{View}_{\mathcal{A}}$ conditioned on $M_{\mathcal{S}} = M_\$$. (If the probability distribution on $\mathcal{M}$ is uniform, then the distribution of $\mathrm{View}_{\mathcal{A}}(M_\$)$ is identically that of $\mathrm{View}_{\mathcal{A}}$, but we do not assume this here.)

Without loss of generality, we assume case (1) above holds. Now we describe $\mathcal{D}$, which uses $\mathcal{D}_0$ as a black box in order to distinguish $(U_s, R^*, \mathrm{View}_{\mathcal{A}} \setminus C)$ and $(U_s, U_m, \mathrm{View}_{\mathcal{A}} \setminus C)$. First, the challenger flips a coin. On heads, he samples $u \leftarrow (U_s, R^*, \mathrm{View}_{\mathcal{A}} \setminus C)$, and on tails, $u \leftarrow (U_s, U_m, \mathrm{View}_{\mathcal{A}} \setminus C)$. In either case he obtains $u = (u_s, u_{test}, u_{view})$ which he passes on to $\mathcal{D}$. $\mathcal{D}$ forms $C_{\mathcal{D}} = M_0 \oplus u_{test}$, which plays the role of C in the protocol. He passes $u_{view} \cup C_{\mathcal{D}}$ to $\mathcal{D}_0$, which returns a bit b representing its guess that $u_{view} \cup C_{\mathcal{D}}$ was sampled from $\mathrm{View}_{\mathcal{A}}(M_b)$. If $b = 0$, then $\mathcal{D}$ outputs a guess of "heads" (i.e., guesses u_{test} was sampled from R^*), otherwise $\mathcal{D}$ guesses "tails" (u_{test} was sampled from U_m).

Now consider the success probability of $\mathcal{D}$ when the challenger flips heads, so that $u_{test} \sim R^*$. In this case, $C_{\mathcal{D}} = M_0 \oplus R^*$ is obtained exactly as in Π_{Gen}, and therefore $u_{view} \cup C_{\mathcal{D}}$ is distributed identically with $\mathrm{View}_{\mathcal{A}}(M_0)$. Thus $\Pr[\mathcal{D}(u) = 1 \mid \text{heads}] = \Pr[D_0(\mathrm{View}_{\mathcal{A}}(M_0)) = 1]$. Alternatively, suppose the challenger flips tails, and u_{test} is uniform. Then $C_{\mathcal{D}} = M_0 \oplus u_{test}$ is uniform, which is also the distribution of C if we choose $M = M_{\mathcal{S}}$ uniformly at random. Thus $\Pr[\mathcal{D}(u) = 1 \mid \text{tails}] = \Pr[D_0(\mathrm{View}_{\mathcal{A}}(M_\$)) = 1]$. Putting these together, we discover

$$\begin{aligned}\big|\Pr[\mathcal{D}(U_s, R^*, \mathrm{View}_{\mathcal{A}} \setminus C) = 1] - \Pr[\mathcal{D}(U_s, U_m, \mathrm{View}_{\mathcal{A}} \setminus C) = 1]\big| \\ = \big|\Pr[\mathcal{D}_0(\mathrm{View}_{\mathcal{A}}(M_0)) = 1] - \Pr[\mathcal{D}_0(\mathrm{View}_{\mathcal{A}}(M_\$)) = 1]\big| \\ > \epsilon/2,\end{aligned}$$

which contradicts the above claim. This completes the verification of ϵ-privacy.

Reliability. Observe that $M_{\mathcal{R}} = C \oplus R$ and $M_{\mathcal{S}} = C \oplus R^*$. Therefore,

$$\begin{aligned}\mathcal{R} \text{ fails to decode correctly } (M_{\mathcal{R}} \neq M_{\mathcal{S}}) &\Longleftrightarrow \mathrm{Ext}(\alpha, \mathit{seed}) = R \neq R^* = \mathrm{Ext}(\alpha^*, \mathit{seed}) \\ &\Longrightarrow \alpha \neq \alpha^* \\ &\Longrightarrow \exists i \in \mathsf{ACC} \text{ s.t. } r_i \neq r_i^* \\ &\Longrightarrow \exists i \in \mathsf{ACC} \text{ s.t. } \mathcal{C}_i \neq \mathcal{C}_i^*.\end{aligned}$$

The latter event only happens if $\mathcal{A}$ succeeds in altering $\mathcal{C}_i$ without $\mathcal{S}$ detecting it. By construction, our consistency check (Section 2) guarantees that this happens with probability at most $(1-D)^\ell = \delta/t$ for a single wire, hence (taking a union bound over corrupt wires) probability at most δ overall. Consequently, $\Pr[M_\mathcal{R} = M_\mathcal{S}] \geq 1-\delta$.

Complexity. The private wires are used only in round 1, to send $Enc(r_i)$ on each wire. The total complexity is therefore $nN = O(nK)$ (for $\mathcal{E}$ of constant rate). As noted above, our assumptions on $\mathcal{E}$ and Ext_A imply that $K = O(m/(n-t)+\ell)$, and therefore the total private wire complexity is $O(mn/(n-t)+n\ell)$, which is $O(mn/(n-t))$ provided $m/(n-t) = \Omega(\ell)$.

The public channel is used in Rounds 2 and 3. In Round 2, $\mathcal{R}$ transmits $J \subset [N]$ of size ℓ, and the restricted codewords $\mathcal{C}_i|_J$, at total cost $\ell n + \ell \log N = \ell n + \ell(\log K + O(1)) = \ell n + O(\ell(\log(m/(n-t)+\ell)))$. Provided that $m = \Omega(\ell)$, this is $O(\ell(n + \log m))$.

In Round 3, $\mathcal{S}$ uses the public channel to send $(B, C, seed)$ where B indicates accepted wires, C hides the message $M_\mathcal{S}$, and $seed$ is a seed for Ext_A. Thus the Round 3 public communication is $n+m+s$, which is $O(n+m)$ for any extractor with reasonable seed length. □

3.2 Instantiating the Generic Protocol

Here we consider possible instantiations of Π_{Gen}. Since our main interest is in 0-private protocols, the most important instantiation will be that with Reed-Solomon codes and the extractor Ext_q of Section 2. Nevertheless, other choices of (explicit) extractor, such as Kamp and Zuckerman's deterministic symbol-fixing extractor [KZ06], are possible; refer to [GGO09] for more details.

Statistical error is a feature of all general-purpose randomness extractors. To get around it, we can exploit the fact that the sources arising from Π_{Gen} are not general min-entropy sources. Rather, conditioning on the adversary's view, each good wire carries independent, uniform randomness, and the corrupt wires carry fixed values. Thus the source we are interested in actually carries quite a great deal of structure. In particular, we may view it as a symbol-fixing source as described in Section 2, since we may group bits into symbols, and the adversary has no information on the symbols carried by good wires.

Consider an instantiation of Π_{Gen} using the extractor $\mathrm{Ext}_q : \mathbb{F}_q^{kN} \rightarrow \mathbb{F}_q^r$ of Section 2, which is indeed errorless. (Here $r = \lceil m/\log q \rceil$ is the size of $M_\mathcal{S}$ in field elements.) Ext_q is, according to our notation, a $(kN, r, r, 0)$ extractor for sources over $\mathbb{F}_q$: It extracts 100% of the randomness from its input with no statistical error. (It is also deterministic, hence trivially strong.) Since Ext_q operates at the level of field elements, Reed-Solomon codes are a natural choice for the error-correcting code $\mathcal{E}$ of Π_{Gen}. We choose $\mathcal{E}$ to be $\mathrm{Ext}_q : \mathbb{F}_q^K \rightarrow \mathbb{F}_q^{2K}$, with relative minimum distance $1/2$.

We now describe two requirements imposed by this instantiation. First, the description of Π_{Gen} assumes an extractor which operates on bits rather than field elements. This presents no real problem, as all statements can be recast in a straightforward way to this new setting. However, as mentioned above, the move from $\{0,1\}$ to $\mathbb{F}_q$ does have the effect of adding a $\log q$ term to the message size required for optimal communication complexity (see statement of and complexity analysis for Theorem 3).

Second, we must specify the appropriate field size q in terms of the basic parameters m, n, t, δ. Recall $\ell = \lceil \log(t/\delta) \rceil$. We require (with foresight):

$$q \log q = \Omega(mn/(n-t)) \qquad \text{and} \qquad (q-2\ell)\log q > \frac{2m}{n-t}.$$

Thus $M_{\mathcal{S}} \in \mathbb{F}_q^r$, where $r = \lceil m/\log q \rceil$.

For the proof of privacy, we require $\mathrm{Ext}_q : \mathbb{F}_q^{nK} \rightarrow \mathbb{F}_q^r$ is in fact a perfect randomness extractor—so we need $q \geq nK + r$. Since $K = r/(n-t) + \ell$, we have (using $m = \Omega(n\ell)$):

$$\begin{aligned} nK + r &= n \cdot (\frac{r}{n-t} + \ell) + r = r(\frac{n}{n-t} + 1) + n\ell \\ &= \frac{m}{(\log q)} \cdot \frac{n}{n-t} + O(m) = O(\frac{m}{(\log q)} \cdot \frac{n}{n-t}). \end{aligned}$$

Thus, for $q \geq nK + r$ it suffices that $q \log q = \Omega(mn/(n-t))$, which is our first assumption on q.

Now observe that in order for our codeword authentication to be valid, we need $q \geq 2K = 2r/(n-t) + 2\ell$. Thus we require:

$$\begin{aligned} q \geq 2r/(n-t) + 2\ell &\iff q \geq \frac{2m}{(\log q)(n-t)} + 2\ell \\ &\iff q \log q \geq \frac{2m}{n-t} + 2\ell \log q \\ &\iff (q-2\ell)\log q \geq \frac{2m}{n-t}, \end{aligned}$$

which gives our second condition on q.

3.3 A Protocol with Logarithmic Public Communication

In this section we present a protocol for SMT-PD which is the first to achieve logarithmic communication complexity (in m) *on the public channel*. The protocol is perfectly private, achieves the optimal communication complexity of $O(\frac{mn}{n-t})$ on the private wires, and has optimal round complexity of $(3, 2)$.

In its Round 3 communication, Π_{Gen} incurs a cost of size m on the public channel, which we wish to reduce to $O(\log m)$. Our improvement comes from the insight that $\mathcal{S}$ can send the third-round message (C, in the notation of Π_{Gen}) on the *common* wires, provided that $\mathcal{S}$ *authenticates* the transmission (making use of the public channel).

$\mathcal{S}$ could simply send C on every common wire and authenticate C publicly. The downside of this approach is that the private wire complexity would be $\Omega(mn)$ rather than $O(\frac{mn}{n-t})$—no longer optimal. Our solution is to take C and encode it *once again* using Reed-Solomon into shares $C_1, \ldots, C_n$, each of size $\approx \frac{m}{n-t}$, such that any $n-t$ correct C_i's will reconstruct C. $\mathcal{S}$ then sends C_i on wire i, and authenticates each C_i publicly.

This authentication uses a short secret key, s^*, of size $\ell(n + \log(\frac{cm}{n-t}))$ (which is the cost of authenticating n messages of size $cm/(n-t)$, using the consistency check

Protocol Π_{SPD}

1. $(\mathcal{R} \overset{PRI}{\to} \mathcal{S})$. **(small)** For each wire i, $\mathcal{R}$ chooses a random $\hat{f}_i \in \mathbb{F}_{\hat{q}}[X]$ such that $\deg(\hat{f}_i) \leq \hat{K}$. $\mathcal{R}$ sends the Reed-Solomon (RS) codeword $\hat{\mathcal{C}}_i = \big(\hat{f}_i(1), \hat{f}_i(2), \ldots, \hat{f}_i(\hat{N})\big)$ along wire i. Let $\hat{\mathcal{C}}_i^*$ be the codeword received by $\mathcal{S}$, and $\hat{f}_i^* = Dec_{RS}(\hat{\mathcal{C}}_i^*)$.
   (big) For each wire i, $\mathcal{R}$ chooses a random $f_i \in \mathbb{F}_q[X]$ such that $\deg(f_i) \leq K$. $\mathcal{R}$ sends the RS codeword $\mathcal{C}_i = \big(f_i(1), f_i(2), \ldots, f_i(N)\big)$ along wire i. Let $\mathcal{C}_i^*$ be the codeword received by $\mathcal{S}$, and $f_i^* = Dec_{RS}(\mathcal{C}_i^*)$.
2. $(\mathcal{R} \overset{PUB}{\to} \mathcal{S})$. **(small)** $\mathcal{R}$ chooses a random subset $\hat{J} = \{\hat{j}_1, \ldots, \hat{j}_\ell\} \subset [\hat{N}]$ of codeword indices, $|\hat{J}| = \ell$. $\mathcal{R}$ performs codeword verification as in Section 2 by sending $\hat{J}$, as well as $\{\hat{\mathcal{C}}_i|_{\hat{J}}\}$ for each wire i, over the public channel.
   (big) $\mathcal{R}$ chooses a random subset $J = \{j_1, \ldots, j_\ell\} \subset [N]$ of codeword indices, $|J| = \ell$. $\mathcal{R}$ performs codeword verification as in Section 2 by sending J, as well as $\{\mathcal{C}_i|_J\}$ for each wire i, over the public channel.
3. $(\mathcal{S} \overset{PUB+PRI}{\longrightarrow} \mathcal{R})$. $\mathcal{S}$ rejects any wire i which is syntactically incorrect or which fails one of the consistency checks in Round 2. Call the set of remaining, accepted wires ACC.
   (small) Let $\hat{\alpha}^*$ denote the concatenation of $\hat{f}_i^*$ for each $i \in \mathsf{ACC}$, padded with $0 \in \mathbb{F}_q$ so its length is $\hat{K}n$. Applying $\text{Ext}_{\hat{q}} : \mathbb{F}_{\hat{q}}^{\hat{K}n} \to \mathbb{F}_{\hat{q}}^{\hat{r}}$ of Section 2, $\mathcal{S}$ obtains $s^* = \text{Ext}_{\hat{q}}(\hat{\alpha}^*)$.
   (big) Let α^* denote the concatenation of f_i^* for each $i \in \mathsf{ACC}$, padded with $0 \in \mathbb{F}_q$ so its length is Kn. Applying the randomness extractor $\text{Ext}_q : \mathbb{F}_q^{Kn} \to \mathbb{F}_q^r$, $\mathcal{S}$ obtains $R^* = \text{Ext}_q(\alpha^*)$.
   Now $M_{\mathcal{S}}$ and R^* are both vectors in $\mathbb{F}_q^r$; $\mathcal{S}$ puts $C = R^* + M_{\mathcal{S}}$. Now $\mathcal{S}$ applies the Reed Solomon code $\mathbb{F}_q^r \to \mathbb{F}_q^{Kn}$ to C, obtaining a codeword $D \in \mathbb{F}_q^{Kn}$. Let $D = (D_1, \ldots, D_n)$ where each $D_i \in \mathbb{F}_q^K$. View D_i as a bit-string of length $K \log q$, and let $E_i = Enc(D_i)$, so that $|E_i| = cK \log q$ (in bits). $\mathcal{S}$ sends E_i on wire $i \in \mathsf{ACC}$; let E_i^* denote the message received by $\mathcal{R}$ on wire i.
   To authenticate each E_i, $\mathcal{S}$ chooses a random subset $J' \subseteq [cK \log q]$, $|J'| = \ell_{3/2}$. Put $auth_{\mathcal{S}} = (J', \{E_i|_{J'}\}_{i \in \mathsf{ACC}})$; we have $|auth_{\mathcal{S}}| \leq \hat{m}$ (with equality if every wire is in ACC). Padding as necessary, view $auth_{\mathcal{S}}$ as an element of $\mathbb{F}_{\hat{q}}^{\hat{r}}$. $\mathcal{S}$ sets $V = s^* + auth_{\mathcal{S}}$ and sends (V, B) over the public channel, where B is an n-bit string representing the set ACC.
   Receiver: $\mathcal{R}$ learns ACC from B. For $i \in \mathsf{ACC}$, he forms α, the concatenation of f_i for each $i \in \mathsf{ACC}$ (padded with $0 \in \mathbb{F}_q$ to length Kn). He applies Ext_q to obtain $R = \text{Ext}_q(\alpha) \in \mathbb{F}_q^r$.
   Similarly, for $i \in \mathsf{ACC}$, he forms $\hat{\alpha}$, the concatenation of $\hat{f}_i$ for each $i \in \mathsf{ACC}$ (padded with $0 \in \mathbb{F}_q$ to length $\hat{K}n$). He applies $\text{Ext}_{\hat{q}}$ to obtain $s = \text{Ext}_{\hat{q}}(\hat{\alpha}) \in \mathbb{F}_{\hat{q}}^{\hat{r}}$.
   Next $\mathcal{R}$ forms $V - s$, which he parses as $auth_{\mathcal{R}} = (J'^*, \{check_i\}_{i \in \mathsf{ACC}})$. For each (correctly formed) E_i^*, $\mathcal{R}$ verifies its authenticity by checking that $E_i^*|^*_{J'} = check_i$. For those which pass, he recovers $D_i^* = Dec(E_i^*)$, $D_i^* \in \mathbb{F}_q^K$. Once $\mathcal{R}$ has recovered at least $n - t$ valid D_i^*'s, he has $K(n - t) = r$ symbols in $\mathbb{F}_q$, which he uses to decode the RS code used by $\mathcal{S}$ to encode C. (This is simply interpolation.) Call the result $C^* \in \mathbb{F}_q^r$. Finally, $\mathcal{R}$ obtains $M_{\mathcal{R}} = C^* - R$.
   (On failure to authenticate at least $n - t$ E_i^*'s, or to parse $auth_{\mathcal{R}}$ correctly, $\mathcal{R}$ outputs $\perp$.)

Fig. 2. SMT-PD protocol with small (logarithmic) public communication and optimal private communication

of Section 2; c is an absolute constant defined below). Thus, $\mathcal{S}$ and $\mathcal{R}$ will run two processes in parallel: a "small" strand, in which $\mathcal{S}$ privately sends the short key to $\mathcal{R}$; and a "big" strand, in which $\mathcal{S}$ sends $M_{\mathcal{S}}$ to $\mathcal{R}$, making use of the shared key in the third round. The small protocol sends the short key using any reasonably efficient SMT-PD protocol; for ease of exposition, we use Π_{Gen}, instantiated with Reed-Solomon codes. We also use Π_{Gen} with Reed-Solomon codes for the big strand of the protocol in order to achieve perfect privacy and optimal private wire complexity.

We now describe the protocol in detail. Many of the parameters are the same as in (the Reed-Solomon instantiation of) Π_{Gen}: We set $\ell = \lceil \log(t/\delta) \rceil$, and fix a prime q such that

$$q \log q = \Omega(mn/(n-t)) \qquad \text{and} \qquad (q - 2\ell)\log q \geq \frac{2m}{n-t}.$$

The message space is $\mathcal{M} = \mathbb{F}_q^r$, that is, an m-bit message is considered as a sequence of $r = \lceil m/\log q \rceil$ field elements in $\mathbb{F}_q$. (However, we also assume, for the purpose of the Round 3 authentication, that the field elements are actually *represented* as bit-strings of length $r \log q$.) Set $K = \lceil r/(n-t) \rceil + \ell$ and $N = 2K$.

In addition to the above parameters, we will also define their small-strand counterparts, which we notate using variables with hats. Set $\hat{m} = \ell(n + \log(cK \log q))$—as noted above, this is the size of the shared secret which will be used to authenticate the C_i's. Here the constant $c > 1$ is the expansion factor of an efficiently computable, constant-rate error-correcting code $\mathcal{E}'$ of relative minimum distance (say) $1/3$. (We caution that $\mathcal{E}'$ plays a different role in Π_{SPD} than $\mathcal{E}$ did in Π_{Gen}, hence the different name.) We will use Enc and Dec to denote the encoding and decoding functions of $\mathcal{E}'$; we use Enc_{RS} and Dec_{RS} for the encoding and decoding functions of the Reed-Solomon code which functions as $\mathcal{E}$ for Π_{SPD}.

Fix $\hat{q}$ to be a prime such that

$$\hat{q} \log \hat{q} = \Omega(\frac{\hat{m}n}{n-t}) \qquad \text{and} \qquad (\hat{q} - 2\ell)\log \hat{q} > \frac{2\hat{m}}{n-t},$$

Set $\hat{r} = \lceil \hat{m}/\log \hat{q} \rceil$, $\hat{K} = \lceil \hat{r}/(n-t) \rceil + \ell$, and $\hat{N} = 2\hat{K}$. Finally, set $\ell_{3/2} = \log_{3/2}(t/\delta)$.

The protocol, Π_{SPD} (for "small public discussion"), is shown in Figure 2. Keep in mind the high-level understanding of the protocol: The first two rounds are simply parallel versions of Rounds 1 and 2 of Π_{Gen}, run with different (big and small) parameters. In Round 3, we complete the small instance of Π_{Gen} as usual, and use the resulting shared secret to blind the (public-channel) authentication of the C_i's which encode C. The latter have been sent on the unreliable private wires, unlike in Π_{Gen}, where no authentication was required in Round 3 since C itself was sent on the public channel.

Theorem 3. *Protocol* Π_{SPD} *(Fig. 2) is a valid* $(3, 2)$*-round* $(0, 3\delta)$*-SMT-PD protocol. It has communication complexity* $O(\frac{mn}{n-t})$ *on the private wires and* $O(n \log(t/\delta) \log m)$ *on the public channel, provided* $m = \Omega(n \log(t/\delta) \log q)$.

4 Private Communication Lower Bound

In this section we prove a lower bound of $\Omega(\frac{nm}{n-t})$ for the expected communication complexity on the private wires, for *any* (ϵ, δ)-SMT-PD protocol (where ϵ and δ are considered constants). Since protocol Π_{Gen} of the previous section meets this bound, we provide a complete answer to the question raised in [SJST09] of determining the optimal transmission rate on private wires for an (ϵ, δ)-SMT-PD protocol.

Our communication lower bound holds even for a weakened adversary who is *passive* and *non-adaptive*—that is, $\mathcal{A}$ chooses which wires to corrupt at the start of the protocol and only eavesdrops thereafter. It also holds even if we modify δ-reliability so that the probability that $M_{\mathcal{R}} = M_{\mathcal{S}}$ is taken over the the choice of $M_{\mathcal{S}}$ as well (and not just the players' coins). Further, as noted in the Introduction, it also holds in the case of SMT with no public channel, *mutatis mutandis*.

For the lower bound, we assume that $M_{\mathcal{S}}$ is chosen uniformly at random from $\mathcal{M}$; in this case $H(M_{\mathcal{S}}) = \log|\mathcal{M}|$. In the following lemmas we assume Π is a valid (ϵ, δ)-SMT-PD protocol, and probabilities are over all players' coins as well as the random selection of $M_{\mathcal{S}} \in \mathcal{M}$.

The first two lemmas are complementary, establishing entropy versions of ϵ-privacy and δ-reliability, respectively. Namely, in Lemma 1, we show that in any ϵ-private protocol, the entropy of $M_{\mathcal{S}}$ remains high given the adversary's view. Then in Lemma 2, we show that for any δ-reliable protocol (with passive adversary), the entropy of $M_{\mathcal{S}}$ given the entire transcript of communications is low. Though these statements are quite intuitive, their proofs are relatively delicate.

Lemma 1. *For all adversaries $\mathcal{A}$ and all ϵ-private protocols, $H\left(M_{\mathcal{S}} \mid \text{View}_{\mathcal{A}}\right) \geq -\log(1/|\mathcal{M}| + 2\epsilon)$.*[8]

The *transcript* T of an (ϵ, δ)-SMT-PD protocol execution is the random variable consisting of the list of messages the players send on public and private channels over the course of the protocol. Thus in the case of a passive adversary, T is completely determined by $M_{\mathcal{S}}$, $C_{\mathcal{S}}$, and $C_{\mathcal{R}}$. For a given set of wires S, we will let T_S denote the transcript restricted to communications on the wires in S. In the sequel we use PUB, PRIV, CORR, and SEC to denote respectively the public channel, private wires, corrupted wires, and secure (uncorrupted and private) wires.

We use $H_2(\cdot)$ to denote the binary entropy function, $H_2(p) = -p\log p - (1 - p)\log(1-p)$.

Lemma 2. *For all δ-reliable protocols, $H(M_{\mathcal{S}} \mid T) \leq H_2(\sqrt{\delta}) + 2\sqrt{\delta}H(M_{\mathcal{S}})$.*

Given Lemmas 1 (a proof of "high" entropy) and 2 (a proof of "low" entropy), we take the difference of the two inequalities (leaving still a "high" amount of entropy), and show that this bounds from below $H(T_{\mathsf{SEC}} \mid \mathsf{SEC})$. This is intuitive: the adversary knows which wires are secure, and yet it is only from these wires that $\mathcal{S}$ and $\mathcal{R}$ can leverage any privacy at all. Therefore the entropy of the messages on them should be high.

Lemma 3. $-\log(1/|\mathcal{M}| + 2\epsilon) - H_2(\sqrt{\delta}) - 2\sqrt{\delta}\log|\mathcal{M}| \leq H\left(T_{\mathsf{SEC}} \mid \mathsf{SEC}\right)$.

Our main lower bound theorem follows. The idea is straightforward. Since the set of secure wires is unknown to $\mathcal{S}$ and $\mathcal{R}$ (for a passive adversary, say), it must be that, in

[8] This entropy lemma is not directly equivalent to a seemingly related probability version (as in [SJST09], Lemma 2).

an average sense, *every* set of $n - t$ private wires carries the requisite entropy. Then we use Han's inequality (see proof in [GGO09]) to "average" the entropy over all subsets of $n - t$ wires and obtain an estimate for the total entropy on private wires, completing the proof.

Theorem 4. *Let Π be any (ϵ, δ)-SMT-PD protocol with $n \leq 2t$, in the presence of a passive, non-adaptive adversary $\mathcal{A}$. Let C denote the expected communication (in bits) over the private wires (the expectation is taken over all players' coins and the choice of $M_S \in \mathcal{M}$). Then*

$$C \geq \frac{n}{n-t} \cdot (-\log(1/|\mathcal{M}| + 2\epsilon) - H_2(\sqrt{\delta}) - 2\sqrt{\delta}\log|\mathcal{M}|)$$

In particular, if $\epsilon = O(1/|\mathcal{M}|)$ and $\delta = O(1)$, then $C = \Omega(mn/(n-t))$.

Corollary 1. *Provided that $\epsilon = O(1/|\mathcal{M}|)$, and $\delta = O(1)$, protocols Π_{Gen} and Π_{SPD} have optimal private communication complexity $O(\frac{nm}{n-t})$ for messages of size $m = \Omega(n\ell)$ and $m = \Omega(n\ell\log q)$, respectively.*

5 Amortized Use of the Public Channel

A natural question when considering SMT-PD as a subroutine in a larger protocol is whether some of the lower bounds on resource use for a single execution of SMT-PD can be beaten *on average* through amortization. For instance, an almost-everywhere secure computation protocol may invoke an SMT-PD subroutine every time any two nodes in the underlying network need to communicate. Must they use the public channel twice every single time, or can the nodes involved, say, save some state information which allows them to reduce their use of the public channel in later invocations?

Our next result shows that amortization can in fact drastically reduce the use of the public channel: indeed, it is possible to limit the total number of uses of the public channel to *two*, no matter how many messages are ultimately sent between two nodes. (Since two uses of the public channel are required to send any reliable communication whatsoever, this is best possible.)

Of course, $\mathcal{S}$ and $\mathcal{R}$ may use the first execution of SMT-PD to establish a shared secret key, which can be used for message encryption and authentication on the common wires. The Sender computes a ciphertext and sends it (with authentication) on every common wire. With overwhelming probability, no forged message is accepted as authentic, and the Receiver accepts the unique, authentic message which arrives on any good wire. However, since we are considering the information-theoretic setting, each use of the shared key reduces its entropy with respect to the adversary's view. If the parties know in advance an upper bound on the total communication they will require, and can afford to send a proportionally large shared key in the first execution of SMT-PD, then this approach is tenable by itself.

In some situations, however, the players may not know a strict upper bound on the number of messages they will send. And even when they do, it may happen that the protocol terminates early with some probability, so that an initial message with large entropy is mostly wasted. With these considerations in mind, we now explore strategies which allow $\mathcal{S}$ and $\mathcal{R}$ to communicate *indefinitely* after using only two broadcast rounds and a limited initial message. Our approach is to separate Sender and Receiver's interaction following the first execution of SMT-PD into two modes: a *Normal Mode* and a *Fault-Recovery Mode*.

In the Normal Mode, $\mathcal{S}$ and $\mathcal{R}$ communicate over the common wires without making use of their shared key; they are successful provided the adversary does not actively interfere. If the adversary does interfere, one of the players (say $\mathcal{R}$) will detect this and enter Fault-Recovery Mode, in which he uses the shared key to broadcast information about the messages he received on each common wire, allowing $\mathcal{S}$ to determine at least one corrupted wire (which he then informs $\mathcal{R}$ about, authentically).

In this way, $\mathcal{S}$ and $\mathcal{R}$ communicate reliably and privately so long as the adversary is passive; and any time he is active, they are able to eliminate at least one corrupted wire.[9] (Of course, once they have eliminated all t corrupt wires, communication becomes *very* efficient.) In the sequel, we describe implementations of Normal Mode and Fault-Recovery Mode, as well as how the two modes interact with each other.

Normal Mode. Let us first define a weaker version of SMT by public discussion in which reliability is only guaranteed for a passive adversary. Let Π be a protocol which attempts to send a message from $\mathcal{S}$ to $\mathcal{R}$ using *only the common wires* (and not relying on any shared secret key). Then we say Π is a *Weak* (ϵ, δ) *SMT-PD* protocol if it satisfies Definition 2 where we (1) add to the adversary's view a bit indicating whether $\mathcal{R}$ accepted a message or not (see next point), and (2) replace RELIABILITY with:

WEAK RELIABILITY:

- *(Correctness with passive adversary)* If the adversary only eavesdrops, then $\mathcal{R}$ receives the message correctly.
- *(Detection of active adversary)* If the adversary actively corrupts any wire, then with probability $\geq 1 - \delta$, *either* $\mathcal{R}$ receives the message correctly ($M_{\mathcal{R}} = M_{\mathcal{S}}$), *or* $\mathcal{R}$ outputs "Corruption detected."

The first change above affects ϵ-privacy since it alters the definition of $\text{View}_{\mathcal{A}}$; this is necessary because in the compiled, amortized protocol using Weak SMT-PD as a subroutine, the adversary will learn whether $\mathcal{R}$ accepted a message based on whether $\mathcal{R}$ does or does not enter Fault-Recovery Mode.

We remark in passing that Weak SMT-PD is similar in spirit to *almost* SMT from the standard (non-public discussion) model [KS07], in that both are relaxations which allow one-round transmission (for Weak SMT-PD, only with a passive adversary). The difference is that in the ordinary model, definitions for almost SMT require that the message be correctly received with overwhelming probability regardless of the adversary's actions; in the public discussion model, when the adversary controls a majority of wires, this is impossible, so we only require that corruptions be detected. Indeed, we cannot guarantee reliability in a single round even when the adversary simply *blocks* transmission on corrupted wires (otherwise a minority of wires would carry enough information to recover the message, thus violating privacy).

[9] This is akin to the "slow" PSMT original protocol in [DDWY93].

If we do not require the Weak SMT-PD protocol to finish in *one round*, then there is a simple solution: use the common wires to *simulate* the public channel wire in an ordinary SMT-PD protocol. Any time a party would use the public channel, they instead send the public-channel message over *every* common wire. Two possibilities arise: (1) The adversary never tampers with any such "virtual" public channel invocation. In this case, the virtual public channel functions like an actual public channel, and the protocol succeeds with the same probability as the underlying SMT-PD protocol. (2) The adversary at some point tampers with a virtual public channel invocation. If he does, then the receiving party in that round will detect tampering, and can notify the other player by sending a flag on every channel (or, if the receiving player is $\mathcal{R}$ and it is the final round, he just outputs "Corruption Detected").[10]

The above Weak SMT-PD protocol is conceptually simple (given a pre-existing SMT-PD protocol!), but we might hope to do Weak SMT-PD in a single round, as opposed to the three rounds required for ordinary SMT-PD. The following simple scheme shows one way this can be done.

Assume the Sender wants to send a single field element $M_{\mathcal{S}} = \alpha \in \mathbb{F}_q$. The one-round protocol, $\Pi_{\mathrm{W-SMT-PD}}$, is shown in Figure 3. Essentially, the sender performs a $3t+2$-out-of-$3n$ Shamir secret sharing of the message; however, rather than sending externally specified shares on each wire i (such as $f(1), f(2), f(3)$ on wire 1), he chooses a set of *random* points on which to evaluate f.

Lemma 4. *The protocol of Figure 3 is a Weak (δ,δ)-SMT-PD protocol for q sufficiently large ($\Omega(t/\delta)$).*

We are now ready to describe Normal Mode for $\mathcal{S}$ and $\mathcal{R}$: it is simply the repeated execution of the Weak SMT-PD protocol, with the two players alternating the role of Sender and Receiver, until one of them as Receiver outputs "Corruption detected." At that time, that player's next message to the other party will alert them to enter Fault-Recovery Mode.

Protocol $\Pi_{\mathrm{W-SMT-PD}}$

1. ($\mathcal{S} \overset{PRI}{\rightarrow} \mathcal{R}$). $\mathcal{S}$ chooses a random polynomial $f \in \mathbb{F}_q[x]$ with $\deg(f) \leq 3t+1$ and $f(0) = \alpha$, and a random sequence $x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}, \ldots, x_{n1}, x_{n2}, x_{n3}$ of $3n$ distinct elements of $\mathbb{F}_q \setminus \{0\}$. On wire i $\mathcal{S}$ sends to $\mathcal{R}$ the three pairs $(x_{i1}, f(x_{i1})), (x_{i2}, f(x_{i2})), (x_{i3}, f(x_{i3}))$.
   Receiver: On wire i, $\mathcal{R}$ receives (x^*_{ij}, y^*_{ij}) for $j = 1,2,3$. He verifies that all $3n$ x^*_{ij}'s are distinct, and that the $3n$ points (x^*_{ij}, y^*_{ij}) lie on a polynomial f^* of degree $\leq 3t+1$. If so, he outputs $M_{\mathcal{R}} = f^*(0)$; otherwise (or in case some wire is syntactically incorrect) he outputs "Corruption detected."

Fig. 3. A one-round Weak SMT-PD protocol

[10] We do not consider here whether such a protocol preserves (ϵ-)privacy when the adversary knows whether $\mathcal{R}$ detects corruption; obviously this depends on the details of the protocol. Therefore this is not quite a black-box reduction.

Fault-Recovery Mode. Specifically, suppose $\mathcal{R}$ detects corruption in a message sent by $\mathcal{S}$. He will then use the shared secret established in the initial execution of (ordinary) SMT-PD to secretly and authentically send the following on all wires: (1) a flag signalling Fault-Recovery Mode; (2) a list of specific wires known to be corrupted (if any); (3) the received transmission on all wires not known to be corrupt.

Since at least one of the wires is not corrupted, $\mathcal{S}$ will receive this communication on it and (verifying its authenticity) enter Fault-Recovery Mode also. $\mathcal{S}$ recovers the set of received transmissions and determines which ones were tampered with. He then sends the following to $\mathcal{R}$, again using the shared secret for privacy and authentication: (1) the message $M_{\mathcal{S}}$ on which $\mathcal{R}$ detected corruption; (2) an updated list of specific wires known to be corrupted. At this time, $\mathcal{R}$ has received the intended message and Normal Mode resumes with $\mathcal{R}$ now playing the role of Sender.

Each time Fault-Recovery Mode occurs, $\mathcal{S}$ and $\mathcal{R}$ are able to detect at least one previously unknown corrupt wire. If at any point $\mathcal{S}$ and R have jointly detected t wires as corrupt, they will simply send all future transmissions on the remaining, good wires, guaranteeing perfect privacy and reliability.

Theorem 5. *Given an initial shared secret consisting of $O(n^2)$ field elements, $\mathcal{S}$ and $\mathcal{R}$ can communicate* indefinitely *using only the private wires. The probability that one of them will* ever *accept an incorrect message is $\leq t\delta$. Moreover, with probability $\geq 1-t\delta$, $\mathcal{A}$ gains at most δ information on each of t different messages, and no information on any other message.*

References

[ACH06] Agarwal, S., Cramer, R., de Haan, R.: Asymptotically optimal two-round perfectly secure message transmission. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 394–408. Springer, Heidelberg (2006)

[BG93] Berman, P., Garay, J.: Fast consensus in networks of bounded degree. Distributed Computing 2(7), 62–73 (1993); Preliminary version in WDAG 1990

[BGW88] Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: STOC, pp. 1–10 (1988)

[CCD88] Chaum, D., Crepeau, C., Damgard, I.: Multiparty unconditionally secure protocols. In: STOC, pp. 11–19 (1988)

[CPRS08] Choudhary, A., Patra, A., Pandu Rangan, C., Srinathan, K.: Unconditionally reliable and secure message transmission in undirected synchronous networks: Possibility, feasibility and optimality. Cryptology ePrint Archive, Report 2008/141 (2008)

[DDWY93] Dolev, D., Dwork, C., Waarts, O., Yung, M.: Perfectly secure message transmission. Journal of ACM 1(40), 17–47 (1993)

[DORS08] Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM J. Comput. (2008)

[DPPU86] Dwork, C., Peleg, D., Pippinger, N., Upfal, E.: Fault tolerance in networks of bounded degree. In: STOC, pp. 370–379 (1986)

[FFGV07] Fitzi, M., Franklin, M., Garay, J., Harsha Vardhan, S.: Towards optimal and efficient perfectly secure message transmission. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 311–322. Springer, Heidelberg (2007)

[FM97] Feldman, P., Micali, S.: An optimal probabilistic protocol for synchronous Byzantine agreement. SIAM J. Comput. 26(4), 873–933 (1997)

[FW98] Franklin, M., Wright, R.: Secure communications in minimal connectivity models. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 346–360. Springer, Heidelberg (1998)

[GGO09] Garay, J., Givens, C., Ostrovsky, R.: Secure message transmission with small public discussion. Cryptology ePrint Archive, Report 2009/519 (2009)

[GM98] Garay, J., Moses, Y.: Fully polynomial Byzantine agreement for $n > 3t$ processors in $t + 1$ rounds. SIAM J. Comput. 27(1), 247–290 (1998); Prelim. in STOC 1992

[GO08] Garay, J., Ostrovsky, R.: Almost-everywhere secure computation. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 307–323. Springer, Heidelberg (2008)

[GRR98] Gennaro, R., Rabin, M., Rabin, T.: Simplified VSS and fast-track multiparty computations with applications to threshold cryptography. In: Proc. 17th Annual ACM Symp. on Principles of Distributed Computing, PODC, pp. 101–111. ACM, New York (1998)

[KK06] Katz, J., Koo, C.: On expected constant-round protocols for Byzantine agreement. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 445–462. Springer, Heidelberg (2006)

[KS07] Kurosawa, K., Suzuki, K.: Almost secure (1-round, n-channel) message transmission scheme. Cryptology ePrint Archive, Report 2007/076 (2007)

[KS08] Kurosawa, K., Suzuki, K.: Truly efficient 2-round perfectly secure message transmission scheme. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 324–340. Springer, Heidelberg (2008)

[KZ06] Kamp, J., Zuckerman, D.: Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. SIAM J. Comput. 36(5), 1231–1247 (2006)

[Lam87] Lamport, L.: A fast mutual exclusion algorithm. ACM Transactions on Computer Systems 5(1), 1–11 (1987)

[MS83] Macwilliams, F., Sloane, N.: The Theory of Error-Correcting Codes. North-Holland, Amsterdam (January 1983)

[SA96] Sayeed, H., Abu-Amara, H.: Efficient perfectly secure message transmission in synchronous networks. Information and Computation 1(126), 53–61 (1996)

[SJST09] Shi, H., Jiang, S., Safavi-Naini, R., Tuhin, M.: Optimal secure message transmission by public discussion. In: IEEE Symposium on Information Theory (2009)

[SNP04] Srinathan, K., Narayanan, A., Pandu Rangan, C.: Optimal perfectly secure message transmission. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 545–561. Springer, Heidelberg (2004)

[SPR07] Srinathan, K., Prasad, N.R., Pandu Rangan, C.: On the optimal communication complexity of multiphase protocols for perfect communication. In: IEEE Symposium on Security and Privacy, vol. 0, pp. 311–320 (2007)

[Upf92] Upfal, E.: Tolerating linear number of faults in networks of bounded degree. In: PODC, pp. 83–89 (1992)

On the Impossibility of Three-Move Blind Signature Schemes

Marc Fischlin and Dominique Schröder

Darmstadt University of Technology, Germany
www.minicrypt.de

Abstract. We investigate the possibility to prove security of the well-known blind signature schemes by Chaum, and by Pointcheval and Stern in the standard model, i.e., without random oracles. We subsume these schemes under a more general class of blind signature schemes and show that finding security proofs for these schemes via black-box reductions in the standard model is hard. Technically, our result deploys meta-reduction techniques showing that black-box reductions for such schemes could be turned into efficient solvers for hard non-interactive cryptographic problems like RSA or discrete-log. Our approach yields significantly stronger impossibility results than previous meta-reductions in other settings by playing off the two security requirements of the blind signatures (unforgeability and blindness).

Keywords: Blind signature scheme, black-box reduction, meta-reduction, random oracle, round complexity.

1 Introduction

Blind signatures [11] implement a carbon copy envelope allowing a signer to issue signatures for messages such that the signer's signature on the envelope is imprinted onto the message in the sealed envelope. In particular, the signer remains oblivious about the message (blindness), but at the same time no additional signatures without the help of the signer can be created (unforgeability).

Many blind signature schemes have been proposed in the literature, e.g., [1, 2, 6, 11, 12, 16, 17, 19, 20, 22, 23, 24, 26, 27, 29], with varying security and efficiency characteristics. The arguably most prominent examples are the schemes by Chaum [11] based on RSA and the ones by Pointcheval and Stern [27] based on the discrete logarithm problem, RSA and factoring. Both approaches admit a security proof in the random oracle model, in the case of Chaum's scheme the "best" known security proofs currently even requires the one-more RSA assumption [5].

Here we investigate the possibility of instantiating the random oracles in the schemes by Chaum and by Pointcheval and Stern, and of giving a security proof based on standard assumptions like RSA or discrete logarithm. Although both schemes are different in nature we can subsume them under a more general pattern of blind signature schemes where

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 197–215, 2010.

- blindness holds in a statistical sense, i.e., where even an unbounded malicious signer cannot link executions of the issuing protocol to message-signature pairs,
- the interactive signature issuing has three (or less) moves, and
- one can verify from the communication between a possibly malicious signer and an honest user if the user is eventually able to derive a valid signature from the interaction.

We note that the construction by Boldyreva [6] based on the one-more Gap Diffie-Hellman problem in the random oracle model also obeys these three properties such that any impossibility result immediately transfers to this scheme as well. The third property, which we coin signature derivation check, basically guarantees that blindness still holds if the user fails to produce a signature in the postprocessing step, after the actual interaction with the signer has been completed. Common notions of blindness do not provide any security guarantee in this case (see [13,17] for further discussions).

1.1 The Idea Behind Our Result

Given a blind signature scheme with the properties above we can show that for such schemes finding black-box reductions from successful forgers to an *arbitrary* non-interactive cryptographic problem (like RSA, discrete-log, or general one-wayness or collision-resistance) is infeasible. The key idea to our result is as follows. Assume that we are given a three-move blind signature scheme as above and a reduction $\mathcal{R}$ reducing unforgeability to a presumably hard problem (given only black-box access to an alleged forger). Vice versa, if the problem is indeed infeasbile, then the reduction therefore shows that the scheme is unforgeable.

Our approach is to show that the existence of a reduction $\mathcal{R}$ as above already violates the assumption about the hardness of the underlying problem. Our starting point is to design an oracle Σ with unlimited power and a "magic" adversary $\mathcal{A}^{\Sigma}$ breaking the unforgeability of the blind signature scheme with the help of Σ. By assumption, the reduction $\mathcal{R}$ with access to $\mathcal{A}^{\Sigma}$ is then able to break the underlying cryptographic problem (see the left part of Figure 1). Note that, at this point, we are still in a setting with an all-powerful oracle Σ and the non-interactive problem may indeed be easy relative to this oracle, without contradicting the presumed hardness in the standard model.

Now we apply meta-reduction techniques, as put forward for example in [7,9,14,28], to remove the oracle Σ from the scenario. Given $\mathcal{R}$ we show how to build a meta-reduction $\mathcal{M}$ (a "reduction for the reduction") to derive an efficient solver for the problem, but now without any reference to the magic adversary and Σ (right part of Figure 1). To this end, the meta-reduction $\mathcal{M}$ fills in for adversary $\mathcal{A}^{\Sigma}$ and simulates the adversary's actions without Σ, mainly by resetting the reduction $\mathcal{R}$ appropriately. We have then eventually derived an algorithm $\mathcal{M}^{\mathcal{R}}$ solving the underlying non-interactive problem in the standard model, meaning

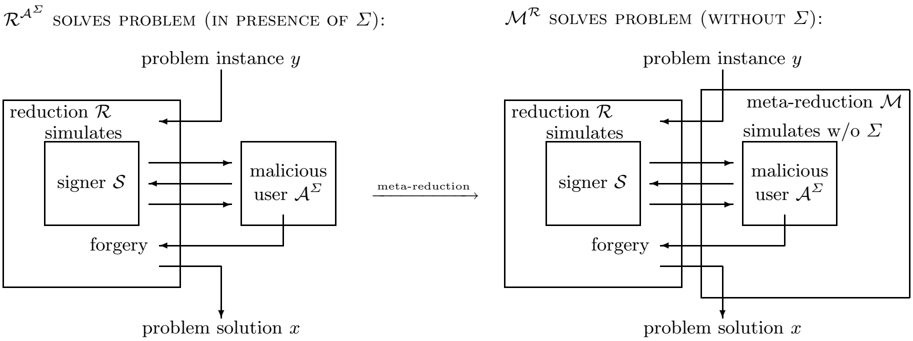

$\mathcal{R}^{\mathcal{A}^\Sigma}$ SOLVES PROBLEM (IN PRESENCE OF Σ):
problem instance y
reduction $\mathcal{R}$ simulates
signer $\mathcal{S}$
malicious user $\mathcal{A}^\Sigma$
forgery
problem solution x
meta-reduction
$\mathcal{M}^{\mathcal{R}}$ SOLVES PROBLEM (WITHOUT Σ):
problem instance y
reduction $\mathcal{R}$ simulates
meta-reduction $\mathcal{M}$ simulates w/o Σ
signer $\mathcal{S}$
malicious user $\mathcal{A}^\Sigma$
forgery
problem solution x

Fig. 1. Meta-reduction technique: The black-box reduction $\mathcal{R}$ on the left hand side uses the adversary $\mathcal{A}^\Sigma$ against unforgeability to solve an instance y of the non-interactive problem. The meta-reduction $\mathcal{M}$ on the right hand side then uses $\mathcal{R}$ to solve the problem from scratch, i.e., by simulating $\mathcal{A}^\Sigma$ without Σ. For this, the meta-reduction $\mathcal{M}$ exploits the blindness property of the scheme.

that the problem cannot be hard. In other words, there cannot exist such a reduction $\mathcal{R}$ to a hard problem.[1]

At this point it seems as if we have not used the blindness property of the scheme and that the idea would paradoxically also apply to regular signature schemes (for which we know secure constructions based on any one-way function). This is not the case. The blindness subtly guarantees that the meta-reduction's simulation of the adversary is indistinguishable from the actual behavior of $\mathcal{A}^\Sigma$, such that the success probabilities of $\mathcal{R}^{\mathcal{A}^\Sigma}$ and of $\mathcal{M}^{\mathcal{R}}$ are close. For these two cases to be indistinguishable, namely $\mathcal{R}$ communicating with $\mathcal{A}^\Sigma$ or with $\mathcal{M}$, we particularly rely on the fact that blindness holds relative to the all-powerful oracle Σ used by $\mathcal{A}$, as in case of statistically-blind signature schemes.

The reason that our approach only applies to blind signature schemes with at most three moves originates from the resetting strategy of our meta-reduction. In a three-move scheme the user sends a single message only, such that resetting the reduction in such an execution allows our meta-reduction to choose independent user messages in each run. This is essential for our proof. In schemes with four or more moves the user sends at least two messages and the second message may then depend on the first one, e.g., the scheme may implement a commit-and-prove strategy with four moves.

[1] We consider very general reductions running *multiple* instances of the adversary in a concurrent and *resetting* manner, covering all known reductions for blind signatures in the literature. Yet, since the meta-reduction itself uses rewinding techniques, we somewhat need to restrict the reduction in regard of the order of starting and finishing resetted executions of different adversarial instances (called resetting with restricted cross-resets). This saves us from an exponential running time for $\mathcal{M}$. For example, any resetting reduction running only a single adversarial instance at a time obeys our restriction.

1.2 The Essence of Our Meta-reduction and Impossibility of Random Oracle Instantiations

There are essentially two approaches in the literature to derive black-box separations like ours. One class of black-box separation results (e.g., [21,30,31]) basically starts with an oracle Σ breaking any cryptographic primitive of type A, like a collision-resistant hash function, but adds an oracle Π implementing another primitive of type B like a one-way function (and which cannot be broken by Σ). Here, the cryptographic primitives in question are usually treated as black boxes.

The other approach uses meta-reductions [4,7,8,9,14,28] and usually treats the adversary as a black box. In our case, we show that no black-box reduction to *arbitrary* (non-interactive) cryptographic problems can exist. This includes common assumptions like the RSA and discrete logarithm problem, but also more general notions of one-way functions and collision-resistant hash functions. Compared to oracle-based separations and previous meta-reduction techniques our result gives the following two advantages:

- Oracle separations involving a "positive" oracle Π implementing a primitive often do not allow to make statements about the possibility of deriving schemes based on concrete primitives such as RSA or discrete-log. The latter primitives have other properties which could potentially be exploited for a security proof, like homomorphic properties. This limitation does not hold for our results.
- Meta-reduction separations such as [4,8,28] consider the impossibility of reductions from secure encryption or signatures to a given RSA instance. Yet, they often fall short of providing any meaningful claim if other assumptions enter the security proof, e.g., the result in [28] does not hold anymore if two RSA instances are given or an additional collision-resistant hash function is used in the design. In comparison, our general approach covers such cases as we can easily combine non-interactive problems P_1, P_2 into more complex problems like $P_1 \vee P_2$ and $P_1 \wedge P_2$, requiring to break one of the two problems and both of them, respectively.

The latter advantage emerges because our meta-reduction plays off unforgeability against blindness. This idea may be useful in similar settings where two or more security properties are involved, to provide stronger separation results for meta-reductions.

The broader class of problems ruled out by our meta-reduction also allows to make meaningful claims when it comes to the possibility instantiating the random oracle in the blind signature schemes. Namely, our separation indicates the limitations of hash function options (assuming some restriction on the resets of the reductions, mentioned in the previous section):

> *Any hash function whose security can be proven by black-box reduction to hard non-interactive problems does not allow a black-box reduction from the unforgeability of the blind signature scheme to hard non-interactive problems, such as RSA or discrete-logarithm.*

This can be seen as follows. Any reduction from the unforgeability either breaks the underyling non-interactive problem like RSA or discrete-log, or breaks some security property of the hash function. The latter, in turn, yields a nested reduction from the unforgeability of the blind signature scheme to the non-interactive problem on which the hash function is based. One only needs to ensure that this nested reduction falls within our admissible reset strategy. This is clearly true if the security property of the hash function is given by a hard non-interactive problem itself, like one-wayness or collision-resistance, or allows a suitable reduction to these problems or RSA, discrete-log etc.

1.3 Extension to Computational Blindness

In principle our result extends to computationally-blind signature schemes but the conditions are arguably more restrictive than in the statistical case. First, recall that blindness needs to hold relative to the forgery oracle Σ, i.e., the powerful forgery oracle must not facilitate the task of breaking blindness. While this comes "for free" in the statistical case, in the computational case one must assume that unforgeability and blindness of the scheme are somewhat independent. This is true for instance for Fischlin's scheme [16], but there are also examples where blindness and unforgeability are correlated, as in Abe's scheme [1] where unforgeability is based on the discrete-log problem and blindness on the DDH problem.

Second, given that the scheme is computationally-blind relative to Σ we still rely on the signature derivation check. One can easily design computationally-blind schemes infringing this property, say, by letting the user sent a public key and having the signer encrypt each reply (we are not aware of any counter example in the statistical case). On the other hand, these signature derivation checks are very common, e.g., besides the schemes above the ones by Okamoto [26] and by Fischlin [16] too have this property.

Third, since we have to change the forgery oracle Σ for the computational case, we also need a key-validity check which allows to verify if a public key has a matching secret key (i.e., if there is a key pair with this public key in the range of the key generating algorithm). For schemes based on discrete-logarithm this usually boils down to check that the values are group elements. Given that these three conditions are met we show that our techniques carry over to the computational case.

1.4 Related Work

In a sense, our results match the current knowledge about the round complexity of blind signature schemes. Nowadays, the best upper bound to build (non-concurrently) secure blind signatures are four moves for the standard model, i.e., neither using random oracles nor set-up assumptions like a common reference string. This is achieved by a protocol of Okamoto [26] based on the 2SDH bilinear Diffie-Hellman assumption. Any schemes with three moves or less either use the random oracle model [6,11,27] or a commom reference string [2,16,19].

We note that Lindell [25] rules out any concurrently secure blind signature scheme in the standard model, independently of any cryptographic assumption. Hence, it seems that two-move schemes —which are concurrently secure by nature— are impossible in the standard model. However, Lindell's impossibility result only refers to the stronger (black-box) *simulation-based* definition of blind schemes and can indeed be circumvented by switching to the common *game-based* definition, as shown by [20]. In contrast, our result holds with respect to game-based definitions and also covers three-move schemes, thus showing that such blind signature schemes may be hard to build even under this relaxed notion.

The recent results by Brown [8] and Bresson et al. [4] show meta-reduction based separations of the one-more RSA and one-more discrete-logarithm problem from their regular counterparts. The conclusion in [4] is that it should be hard to find a security proof for Chaum's scheme and the Pointcheval-Stern schemes using only these regular assumptions. As mentioned before, the meta-reductions in [8,4] are limited in the sense that they either cannot rewind (as in [8]) or can only forward the input RSA or discrete log problem (as in [4]). Our approach, however, considers arbitrary hard non-interactive problems and is robust with respect to the combination of several underlying assumptions.

We also remark that the well-known three-move lower bound for non-trivial zero-knowledge [18] is not known to provide a lower bound for blind signature schemes. The intuitively appealing idea of using the blind signature scheme as a commitment scheme in such zero-knowledge proofs unfortunately results in proofs which require more than three moves. This is even true if we start with a two-move blind signature scheme where a "hidden" third move is required for the initial transmission of the signer's public key. In addition, the game-based notion of blind signatures is not known to yield appropriate zero-knowledge simulators.

Organization. We start with the definition of blind signature schemes in Section 2. In Section 3 we discuss our notion of black-box reductions to hard problems. Before presenting our main result in Section 5 where we show the hardness of finding black-box reductions from unforgeability to non-interactive problems we first discuss a simpler case for restricted reductions in Section 4 to provide some intuition about the general result. Due to the space restrictions, we have delegated the case of computational blindness, as well as most of the proofs, to the full version.

2 Blind Signatures

To define blind signatures formally we introduce the following notation for interactive execution between algorithms $\mathcal{X}$ and $\mathcal{Y}$. By $(a, b) \leftarrow \langle \mathcal{X}(x), \mathcal{Y}(y) \rangle$ we denote the joint execution, where x is the private input of $\mathcal{X}$, y defines the private input for $\mathcal{Y}$, 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 sequential order. Accordingly, $\mathcal{X}^{\langle \cdot, \mathcal{Y}(y_0) \rangle^1, \langle \cdot, \mathcal{Y}(y_1) \rangle^1}(x)$ can invoke sequentially ordered executions with $\mathcal{Y}(y_0)$ and $\mathcal{Y}(y_1)$, but interact with each algorithm only once.

Definition 1 (Blind Signature Scheme). *A blind signature scheme consists of a tuple of efficient algorithms* $\mathsf{BS} = (\mathsf{KG}, \langle \mathcal{S}, \mathcal{U} \rangle, \mathsf{Vf})$ *where*

Key Generation. $\mathsf{KG}(1^n)$ *generates a key pair* (sk, pk)*.*

Signature Issuing. *The joint execution of the algorithms* $\mathcal{S}(sk)$ *and* $\mathcal{U}(pk, m)$ *for message* $m \in \{0,1\}^n$ *generates an output* σ *of the user,* $(\perp, \sigma) \leftarrow \langle \mathcal{S}(sk), \mathcal{U}(pk, m) \rangle$*, where possibly* $\sigma = \perp$*.*

Verification. $\mathsf{Vf}(pk, m, \sigma)$ *outputs a bit.*

It is assumed that the scheme is complete*, i.e., for any* $(sk, pk) \leftarrow \mathsf{KG}(1^k)$*, any message* $m \in \{0,1\}^n$ *and any* σ *output by* $\mathcal{U}$ *in the joint execution of* $\mathcal{S}(sk)$ *and* $\mathcal{U}(pk, m)$ *we have* $\mathsf{Vf}(pk, m, \sigma) = 1$*.*

Security of blind signature schemes requires two properties, namely unforgeability and blindness [22,27]. A malicious user $\mathcal{U}^*$ against unforgeability tries to generate $k+1$ valid message-signatures pairs after at most k completed interactions with the signer, where the number of interactions is adaptively determined by the user during the attack. The blindness condition says that it should be infeasible for a malicious signer $\mathcal{S}^*$ to decide upon the order in which two messages m_0 and m_1 have been signed in two executions with an honest user $\mathcal{U}$.

Definition 2 (Secure Blind Signature Scheme). *A blind signature scheme* $\mathsf{BS} = (\mathsf{KG}, \langle \mathcal{S}, \mathcal{U} \rangle, \mathsf{Vf})$ *is called secure if the following holds:*

Unforgeability. *For any efficient algorithm* $\mathcal{U}^*$ *the probability that experiment* $\mathsf{Forge}^{\mathsf{BS}}_{\mathcal{U}^*}$ *evaluates to* 1 *is negligible (as a function of* n*) where*

```
***Experiment*** $\mathsf{Forge}^{\mathsf{BS}}_{\mathcal{U}^*}$
    $(sk, pk) \leftarrow \mathsf{KG}(1^n)$
    $((m_1, \sigma_1), \ldots, (m_{k+1}, \sigma_{k+1})) \leftarrow \mathcal{U}^{*\langle \mathcal{S}(sk), \cdot \rangle^\infty}(pk)$
    *Return* 1 *iff*
        $m_i \neq m_j$ *for* $1 \leq i < j \leq k+1$, *and*
        $\mathsf{Vf}(pk, m_i, \sigma_i) = 1$ *for all* $i = 1, 2, \ldots, k+1$, *and*
        *at most* $k$ *interactions with* $\langle \mathcal{S}(sk), \cdot \rangle^\infty$ *were completed.*
```

Computational resp. Statistical Blindness. *For any (efficient resp. computationally unbounded) algorithm* $\mathcal{S}^*$ *working in modes* find*,* issue *and* guess*, the probability that the following experiment* $\mathsf{Blind}^{\mathsf{BS}}_{\mathcal{S}^*}$ *evaluates to* 1 *is negligibly close to* 1/2*, where*

```
***Experiment*** $\mathsf{Blind}^{\mathsf{BS}}_{\mathcal{S}^*}$
    $(pk, m_0, m_1, \mathsf{st}_{\mathsf{find}}) \leftarrow \mathcal{S}^*(\mathsf{find}, 1^n)$
    $b \leftarrow \{0,1\}$
    $\mathsf{st}_{\mathsf{issue}} \leftarrow \mathcal{S}^{*\langle \cdot, \mathcal{U}(pk, m_b) \rangle^1, \langle \cdot, \mathcal{U}(pk, m_{1-b}) \rangle^1}(\mathsf{issue}, \mathsf{st}_{\mathsf{find}})$
        *and let* $\sigma_b, \sigma_{1-b}$ *denote the (possibly undefined) local outputs*
        *of* $\mathcal{U}(pk, m_b)$ *resp.* $\mathcal{U}(pk, m_{1-b})$*.*
    *set* $(\sigma_0, \sigma_1) = (\perp, \perp)$ *if* $\sigma_0 = \perp$ *or* $\sigma_1 = \perp$
    $b^* \leftarrow \mathcal{S}^*(\mathsf{guess}, \sigma_0, \sigma_1, \mathsf{st}_{\mathsf{issue}})$
    *return* 1 *iff* $b = b^*$*.*
```

We remark that, even if occassionally not mentioned, all algorithms in this paper receive the security parameter 1^n as additional input.

3 Hard Problems and Black-Box Reductions

In order to prove the security of a cryptographic protocol, usually reduction techniques are used. A reduction from a cryptographic protocol to an underlying problem shows that breaking the protocol implies breaking the underlying problem. A reduction is *black-box* if it treats the adversary and/or the underlying primitive as an oracle. Reingold et al. [30] call reductions which use both the adversary and the primitive merely as an oracle *fully-black-box*, whereas *semi-black-box* reductions work for any efficient adversaries (whose code the reduction may access) as long as the primitive is black-box.

In our case we only need the orthogonal requirement to semi-black-box reductions, namely that the reduction treats the adversary as an oracle but we do not make any assumption about the representation of the underlying primitive. The reduction we consider works for any kind of non-interactive primitive (i.e., in which one gets an instance as input and outputs a solution without further interaction):

Definition 3 (Hard Non-Interactive Problem). *A non-interactive (cryptographic) problem $P = (I, V)$ consists of two efficient algorithms:*

Instance generation $I(1^n)$**.** *The instance generation algorithm takes as input the security parameter 1^n and outputs an instance y.*

Instance Verification $V(x, y)$**.** *The instance verification algorithm takes as input a value x as well as an instance y of a cryptographic problem, and outputs a decision bit.*

We call a cryptographic problem P hard *if the following condition is fulfilled:*

Hardness. *We say that an algorithm $\mathcal{A}$* solves the cryptographic problem *P if the probability that $\mathcal{A}$ on input $y \leftarrow I(1^n)$ outputs x' such that $V(x', y) = 1$, is non-negligible. We say that the problem P is hard if no efficient algorithm solves it.*

Note that in the definition above we do not impose any completeness requirement on the cryptographic problem. The reason is that reductions from the security of blind signatures must work for arbitrary problems, and in particular to the ones with non-trivial completeness conditions.

The notion of a non-interactive cryptographic problem clearly covers such popular cases like the RSA problem, the discrete logarithm problem, or finding collisions for hash functions. It also comprises more elaborate combination of such problems, e.g., if P_0, P_1 are two non-interactive problems then so are $P_0 \wedge P_1$ and $P_0 \vee P_1$ (with the straightforward meaning requiring to solve both problems or at least one of them).

4 Warm Up: Impossibility Result for Vanilla Reductions

To give some intuition about our technique we first consider the simpler case of *vanilla* reductions. This type of reduction only runs a single execution with the adversary (without rewinding) and, if communicating with an honest user, makes the user output a valid signature with probability 1. This means that a vanilla reduction takes advantage of the magic adversary and its output, instead of solving the problem on its own. We then augment our result in the next section to deal with resetting reductions running multiple adversarial instances.

4.1 Preliminaries

For our impossibility result we need another requirement on the blind signature scheme, besides statistically blindness. This property says that one can tell from the public data and communication between a malicious signer and an honest user whether the user is able to compute a valid signature or not.

For instance, in Chaum's scheme the honest user sends a value y and receives z from the signer, and the user is able to compute a signature σ for an arbitrary message m if and only if $z^e = y \bmod N$. This is easily verifiable with the help of the public key and the communication. The scheme of Pointcheval and Stern implements the signature derivation check already in the user algorithm.[2] Analogous derivation checks occur in the schemes by Okamoto and by Fischlin. More formally:

Definition 4 (Signature-Derivation Check). *A blind signature scheme* BS *allows (computational resp. statistical) signature-derivation checks if there exists an efficient algorithm* SDCh *such that for any (efficient resp. unbounded) algorithm* $\mathcal{S}^*$ *working in modes* find *and* issue *the probability that the experiment* $\mathsf{SigDerCheck}^{\mathsf{BS}}_{\mathcal{S}^*,\mathsf{SDCh}}$ *evaluates to* 1 *is negligible, where*

```
***Experiment*** $\mathsf{SigDerCheck}^{\mathsf{BS}}_{\mathcal{S}^*,\mathsf{SDCh}}$
    $(pk, m, \mathsf{st}) \leftarrow \mathcal{S}^*(\mathsf{find}, 1^n)$
    $(\perp, \sigma) \leftarrow \langle \mathcal{S}^*(\mathsf{issue}, \mathsf{st}), \mathcal{U}(pk, m)\rangle$
        *where* $\mathsf{trans}$ *denotes the communication between* $\mathcal{S}^*, \mathcal{U}$
    $c \leftarrow \mathsf{SDCh}(pk, \mathsf{trans})$
    *return* $1$ *if* $\sigma \neq \perp$ *and* $c = 0$*, or if* $\sigma = \perp$ *but* $c = 1$*.*
```

In the computational case, if the above holds even if $\mathcal{S}^*$ *gets access to an oracle* Σ *then we say that the scheme has computational signature-derivation checks relative to* Σ*. (In the statistical case* $\mathcal{S}^*$ *could simulate* Σ *internally, such that granting access to* Σ *is redundant.)*

The notion in some sense augments the blindness property of blind signature schemes to the case that the user algorithm fails to produce a valid signature

[2] The signature derivation check is given by the user's local verification $a = g^R h^S y^e$, where the values a, r, R, S are exchanged during the signature issuing protocol and the values g, h, y are part of the public key.

in the final local step. The common notion of blindness does not provide any security in this case (because the malicious signer does not receive any of the signatures if the user fails only then). See [17] for more discussions and solutions. Here, the signature derivation check provides something stronger, as it can be efficiently performed by anyone and holds independently of the user's message.

Next we introduce a weaker notion than blindness which is geared towards our separation result. Informally, a blind signature scheme has so-called *transcript-independent signatures* if one cannot associate a transcript to a signature. This is formalized by comparing signatures generated via an execution with a malicious signer and signatures generated "magically" via an oracle Σ producing the signature for a message from the public key and the transcript of the first execution. The intuition behind the following experiment is that the malicious signer has to distinguish whether the second signature σ_b results from the signature issuing protocol, or if the oracle Σ derived the signature σ_b from the transcript of the signature issuing protocol where the honest user gets as input the message m_0.

Definition 5 (Transcript-Independent Signatures). *A blind signature scheme* BS *has (computationally resp. statistically) transcript-independent signatures with respect to* Σ *if for any (efficient resp. unbounded) algorithm* $\mathcal{S}^*_{\mathit{trans}}$ *the probability that the experiment* $\mathsf{trans\text{-}ind}^{\mathsf{BS}}_{\mathcal{S}^*_{\mathit{trans}},\Sigma}(n)$ *evaluates to* 1 *is negligibly close to* $1/2$*, where*

```
***Experiment*** $\mathsf{trans\text{-}ind}^{\mathsf{BS}}_{\mathcal{S}^*_{\mathsf{trans}},\Sigma}(n)$:
    $b \leftarrow \{0,1\}$
    $(pk, \mathit{st}_1, m_{-1}, m_0) \leftarrow \mathcal{S}^{*,\Sigma}_{\mathit{trans}}(\mathit{init}, 1^n)$
    $\mathit{st}_2 \leftarrow \mathcal{S}^{*,\Sigma,\langle\cdot,\mathcal{U}(pk,m_{-1})\rangle^1,\langle\cdot,\mathcal{U}(pk,m_0)\rangle^1}_{\mathit{trans}}(\mathit{issue}, \mathit{st}_1)$
        let $\sigma_{-1}$ and $\sigma_0$ be the local outputs of the users in the two
        executions (possibly $\sigma_{-1} = \bot$ and/or $\sigma_0 = \bot$)
        and let $\mathit{trans}_{-1}$ be the transcript of the left execution
    set $m_1 = m_0$ and compute $\sigma_1 \leftarrow \Sigma(pk, \mathit{trans}_{-1}, m_1)$
    set $(\sigma_{-1}, \sigma_0, \sigma_1) = (\bot, \bot, \bot)$ if $\sigma_{-1} = \bot$ or $\sigma_0 = \bot$ or $\sigma_1 = \bot$
    $b^* \leftarrow \mathcal{S}^{*,\Sigma}_{\mathit{trans}}(\mathit{guess}, \mathit{st}_2, m_{-1}, \sigma_{-1}, m_b, \sigma_b)$
    return $1$ iff $b = b^*$.
```

To define our generic forgery oracle Σ allowing $\mathcal{A}$ to break unforgeability we first outline the idea for the case of Chaum's blind signature scheme. Assume that the adversary has already obtained a valid signature for some message m' by communicating with the signer. Let $\mathsf{trans} = (y, z)$ denote the transcript of this communication. Algorithm $\Sigma(pk, \mathsf{trans}, m)$ for $m \neq m'$ then searches some randomness r such that the user's first message for m and r matches y in the transcript, i.e., $H(m)r^e \bmod N = y$. Such an r exists by the perfect blindness and the signature derivation check.[3]

[3] Note that blindness for Chaum's scheme is only guaranteed if the user can verify that the exponent e is relatively prime to $\varphi(N)$, say, if e is a prime larger than N; only then is guaranteed that the function $(\cdot)^e \bmod N$ really is a permutation.

The above example can be generalized to any blind signature scheme and the following generic forgery oracle (which only depends on the blind signature scheme in question):

Definition 6 (Generic Forgery Oracle). *For a statistically-blind signature scheme* BS *the generic forgery oracle* $\Sigma(pk, \mathsf{trans}, m)$ *performs the following steps:*

```
enumerate all values $r$ such that
    the user algorithm $\mathcal{U}(pk, m)$ for randomness $r$ generates the same
    transcript $\mathsf{trans}$ when fed with the same signer messages as in $\mathsf{trans}$;
    also store all signatures $\sigma$ the user's algorithm generates in these executions.
select a value $r$ of the set at random and return the corresponding signature $\sigma$
(or return $\perp$ if there is no such $r$).
```

Proposition 1. *Every statistically blind signature scheme, which has statistical signature-derivation checks, also has statistical transcript-independent signatures with respect to the generic forgery oracle* Σ.

The proof appears in the full version. The idea is that we can safely exchange the order of messages m_{-1}, m_0 in the transcript-independence experiment because of the blindness property. Then oracle Σ in this experiment simply computes another signature for $m_1 = m_0$ from the transcript for a run with the same message m_0 (instead of m_{-1}). By construction of Σ this is perfectly indistinguishable from the original signature derived from this transcript. We note that the signature derivation check and the statistical blindness ensure that failures of Σ do not interfere with the blindness definition (where there are only two executions with the user instances).

Given the generic forgery oracle Σ we can now define the "magic" adversary which first plays an honest users communicating with the signer once. If this single execution yields a valid signature (which is certainly the case when interacting with the genuine signer, but possibly not when interacting with the reduction), then the adversary generates another valid message-signature pair without interaction but using Σ as a subroutine instead.

Definition 7 (Magic Adversary). *The magic adversary* $\mathcal{A}$ *for input* pk *and with oracle access to the generic forgery oracle* Σ *and communicating with an oracle* $\langle \mathcal{S}(sk), \cdot \rangle^1$ *is described by the following steps:*

```
pick random messages $m'_0, m'_1 \leftarrow \{0,1\}^n$
run an execution $\langle \mathcal{S}(sk), \mathcal{U}(pk, m'_0) \rangle$ in the role of an honest user
    to obtain $\sigma'_0$ and let $\mathsf{trans}'_0$ be the corresponding transcript
if $\mathsf{Vf}(pk, m'_0, \sigma'_0) = 1$ then let $\sigma'_1 \leftarrow \Sigma(pk, \mathsf{trans}'_0, m'_1)$ else set $\sigma'_1 \leftarrow \perp$
return $(m'_0, \sigma'_0, m'_1, \sigma'_1)$
```

By the completeness of the blind signature scheme the magic adversary, when attacking the honest signer, returns two valid message-signature pairs, with probability negligibly close to 1 (there is a probability of at most 2^{-n} that the adversary outputs identical pairs for $m'_0 = m'_1$). We also remark that the magic adversary, when attacking the actual scheme, applies the forgery oracle to derive a signature for the second message using the transcript of the first signature issuing protocol.

4.2 Impossibility Result

The following theorem states that vanilla black-box reductions to (non-interactive) cryptographic problems do not provide a meaningful security statement. That is, if there was such a reduction then the underlying problem would already be easy. Since we only deal with non-resetting reductions the claim even holds for schemes with arbitrary round complexity (instead of three-move schemes):

Theorem 1. *Let* BS *be a statistically blind signature scheme that allows statistical signature-derivation checks. Then there is no vanilla black-box reduction from unforgeability of* BS *to a hard non-interactive problem.*

Proof. For sake of readability we divide the reduction $\mathcal{R}$ into steps, according to the black-box simulation of the magic adversary in which $\mathcal{R}$ takes over the role of the signer: in mode init the reduction outputs the public key pk and in mode $\mathsf{msg}i$ the reduction creates the i-th protocol message msg$i$ of the signer. After getting the adversary's signatures σ_0, σ_1 in the post-processing step final the reduction outputs a putative solution x' for its input y. In each step the reduction also outputs some state information which is passed on to the next stage.

Analogously to the reduction $\mathcal{R}$ we denote by $\mathsf{msg}j$ the step of the honest user $\mathcal{U}$ which on input a public key pk, a message m and the previous message msg$i$ of the signer, outputs message msg$j$ sent to the signer. Likewise, in mode finish the user creates the signature from its state and the final message sent by the signer.

Description of the Meta-Reduction. The meta-reduction $\mathcal{M}$ works as follows (see Figure 2 for the case of three moves). It gets as input an instance y of the problem. It start to simulate the reduction $\mathcal{R}$ on y to derive a public key pk as well as the first message msg1 on behalf of the signer and a state $\mathsf{st}_{\text{msg1}}$. Algorithm $\mathcal{M}$ first completes an instance of the signature issuing protocol with $\mathcal{R}$ using the program of the honest user on input a random message m_0 from $\{0,1\}^n$ and some randomness r. Afterwards, it selects another message m' from $\{0,1\}^n$ at random together with some independent randomness r' and resets the reduction to the point where $\mathcal{R}$ has returned the first message of the signature issuing protocol. As before, $\mathcal{M}$ executes the honest user algorithm on m' using the randomness r'.

Now, if the meta-reduction obtains two valid signatures σ_0, σ_1 from both executions, then it hands the pairs (m_0, σ_0), (m_1, σ_1) to the reduction which then outputs some x'. The meta-reduction returns x' and stops. For brevity we often write $\mathcal{R}^{\mathcal{M}}(y)$ for this interaction.

Analysis of the Meta-Reduction. The final step is to show that the reduction $\mathcal{R}$ successfully outputs a solution x', even if given the pairs from $\mathcal{M}$ instead of receiving them from the magic adversary. For this it suffices to show that

```
Meta-reduction $\mathcal{M}(y)$
let $(pk, \mathsf{st}_{\mathsf{init}}) \leftarrow \mathcal{R}(\mathsf{init}, y)$
let $(\mathrm{msg}_1, \mathsf{st}_{\mathsf{msg1}}) \leftarrow \mathcal{R}(\mathsf{msg1}, \mathsf{st}_{\mathsf{init}})$

    choose $m_0 \leftarrow \{0,1\}^n$
    let $(\mathrm{msg2}_0, \mathsf{st}^0_{\mathrm{msg2}}) \leftarrow \mathcal{U}(\mathsf{msg2}, pk, m_0, \mathrm{msg1})$
    let $(\mathrm{msg3}_0, \mathsf{st}^0_{\mathrm{msg3}}) \leftarrow \mathcal{R}(\mathsf{msg3}, \mathsf{st}_{\mathrm{msg1}}, \mathrm{msg2}_0)$
    let $\sigma_0 \leftarrow \mathcal{U}(\mathsf{finish}, \mathsf{st}^0_{\mathrm{msg2}}, \mathrm{msg3}_0)$

    choose $m_1 \leftarrow \{0,1\}^n$
    let $(\mathrm{msg2}_1, \mathsf{st}^1_{\mathrm{msg2}}) \leftarrow \mathcal{U}(\mathsf{msg2}, pk, m_1, \mathrm{msg1})$
    let $(\mathrm{msg3}_1, \mathsf{st}^1_{\mathrm{msg3}}) \leftarrow \mathcal{R}(\mathsf{msg3}, \mathsf{st}_{\mathrm{msg1}}, \mathrm{msg2}_1)$
    let $\sigma_1 \leftarrow \mathcal{U}(\mathsf{finish}, \mathsf{st}^1_{\mathrm{msg2}}, \mathrm{msg3}_1)$

output $x' \leftarrow \mathcal{R}(\mathsf{final}, \mathsf{st}^0_{\mathrm{msg3}}, m_0, \sigma_0, m_1, \sigma_1)$
```

Fig. 2. Meta-Reduction for Vanilla Reduction (three moves), where $\mathsf{trans}_0 = (\mathrm{msg1}, \mathrm{msg2}, \mathrm{msg3})$ denotes the transcript of the first execution

$$\mathrm{Prob}\left[y \leftarrow I(1^n), x' \leftarrow \mathcal{R}^{\mathcal{M}}(y) : V(x', y) = 1 \,\middle|\, \mathcal{M}\right]$$

is non-negligible. As outlined above, for this we exploit the transcript-independence of signatures.

Assume to the contrary that the reduction $\mathcal{R}$ outputs a valid solution x' with non-negligible probability if $\mathcal{R}$ receives two message-signature pairs (m_0, σ_0), (m_1, σ_1) from the magic adversary,

$$\mathrm{Prob}\left[y \leftarrow I(1^n), x' \leftarrow \mathcal{R}^{\mathcal{A}}(y) : V(x', y) = 1 \,\middle|\, \mathcal{A} \text{ magic}\right] \not\approx 0,$$

but succeeds only with negligible probability if the message-signature pairs are generated by $\mathcal{M}$:

$$\mathrm{Prob}\left[y \leftarrow I(1^n), x' \leftarrow \mathcal{R}^{\mathcal{M}}(y) : V(x', y) = 1 \,\middle|\, \mathcal{M}\right] \approx 0.$$

Then we construct an adversary $\mathcal{S}^*_{\mathsf{trans}}$ who breaks the transcript independence of signatures in experiment $\mathsf{trans\text{-}ind}^{\mathsf{BS}}_{\mathcal{S}^*,\Sigma}(n)$.

Description of Adversary $\mathcal{S}^*_{trans}$. Informally, the adversary relays the first execution between the reduction and the external user instance and resets to reduction afterwards to answer the second execution. Afterwards $\mathcal{S}^*_{\mathsf{trans}}$ receives two message-signature pairs without knowing whether the second signature σ_0 has been derived from the signature issuing protocol or with the help of Σ. We then use the result of the reduction to distinguish this case.

More formally, the adversary $\mathcal{S}^*_{\mathsf{trans}}$ generates an instance $y \leftarrow I(n)$ of a cryptographic problem P. It simulates $\mathcal{R}$ in a black-box way, which for input y initially outputs a public key pk as well as the first message msg1 and some state information $\mathsf{st}_{\mathsf{msg1}}$. The algorithm $\mathcal{S}^*_{\mathsf{trans}}$ selects two random message $m_{-1}, m_0 \in \{0,1\}^n$ and outputs pk, m_{-1}, m_0 according to the transcript-independence experiment. It stores the first message (from $\mathcal{R}$ to $\mathcal{U}$) and relays the communication between the reduction $\mathcal{R}$ and the first external user instance $\mathcal{U}(pk, m_{-1})$. Then the adversary resets $\mathcal{R}$ to the point where $\mathcal{R}$ has returned msg1 and forwards the communication between $\mathcal{R}$ and $\mathcal{U}$.

After having finished both executions $\mathcal{S}^*_{\mathsf{trans}}$ receives two (valid) signatures (σ_{-1}, σ_0) and runs the reduction $\mathcal{R}$ in mode final on input $(\mathsf{st}^0_{\mathrm{msg3}}, m_{-1}, \sigma_{-1}, m_0, \sigma_0)$ to obtain a putative solution x' of the cryptographic problem P. The final output of the adversary is $b^* \leftarrow V(x', y)$.

Analysis of $\mathcal{S}^*_{\mathsf{trans}}$. For the analysis recall that the magic adversary, after a single interaction, outputs two message-signature pairs (with the help of Σ). In fact, taking the message-signature pairs (m_{-1}, σ_{-1}) of the first execution together with the message-signature pair (m_0, σ_0) derived from Σ in experiment $\mathsf{trans\text{-}ind}^{\mathsf{BS}}_{\mathcal{S}^*,\Sigma}(n)$ corresponds exactly to the behavior of the magic adversary $(b = 0)$. Here we take advantage of the fact that the second execution with the user cannot fail (and force the signatures to be undefined) by our assumption about the vanilla reduction always making the honest user derive a signature.

On the other hand, during the issuing protocol with the honest user $\mathcal{U}$, the adversary $\mathcal{S}^*_{\mathsf{trans}}$ resets $\mathcal{R}$ and uses in the second execution the prefix msg1 (obtained during the signature generation of (m_{-1}, σ_{-1})) in experiment $\mathsf{trans\text{-}ind}^{\mathsf{BS}}_{\mathcal{S}^*,\Sigma}(n)$. Therefore the message-signature pairs $(m_{-1}, \sigma_{-1}), (m_b, \sigma_b)$ are computed in the same way as the meta-reduction $\mathcal{M}$ does $(b = 1)$. Note that the additional run of Σ in the transcript-independence experiment cannot make the three signatures invalid (except with negligible probability), because of the statistical blindness and the signature derivation checks. More specifically, the statistical blindness guarantees that the transcript generated with $\mathcal{U}$ for message m_{-1} is (almost surely) also a potential transcript for $m_0 = m_1$ used by Σ. Furthermore, the signature derivation check tells us that, independently of the message, the transcript allows the user to derive a signature (such that Σ, too, will find a valid random string r for the simulated user with a valid signature). This fact is stated more formally in the full version. For simplicity we neglect the small error for Σ returning an invalid signature in the analysis below.

We obtain for the probability that $\mathcal{S}^*_{\mathsf{trans}}$ outputs the right bit $b^* = b$:

$$\mathrm{Prob}[\, b^* = b] = \tfrac{1}{2} + \tfrac{1}{2} \cdot (\mathrm{Prob}[\, b^* = 1 \mid b = 1] - \mathrm{Prob}[\, b^* = 1 \mid b = 0])$$

According to our construction, $b = 0$ corresponds to the case where the simulation mimics the behavior of the magic adversary, and $b = 1$ the setting involving the meta-reduction. Furthermore, the adversary $\mathcal{S}^*_{\mathsf{trans}}$ returns $b^* = 1$ in the case that the reduction $\mathcal{R}$ returns a valid solution x' of y. Hence,

$$\begin{aligned}\mathrm{Prob}[\, b^* = 1 \mid b = 1] &- \mathrm{Prob}[\, b^* = 1 \mid b = 0] \\ &= \mathrm{Prob}\left[y \leftarrow I(1^n), x' \leftarrow \mathcal{R}^{\mathcal{A}}(y) : V(x', y) = 1 \;\middle|\; \mathcal{A} \text{ magic} \right] \\ &\qquad - \mathrm{Prob}\left[y \leftarrow I(1^n), x' \leftarrow \mathcal{R}^{\mathcal{M}}(y) : V(x', y) = 1 \;\middle|\; \mathcal{M} \right].\end{aligned}$$

By assumption the difference is non-negligible (because the first probability is non-negligible and we have assumed that the second probability is negligible). This, however, contradicts the transcript independence of signatures. □

5 Impossibility Result for Statistically Blind Signature Schemes

Here we discuss more general reductions which may reset the adversary and run several nested executions with multiple copies of the adversary.

5.1 Preliminaries

To build our meta-reduction we will reset the reduction continuously. That is, whenever the reduction expects a forgery from an instance of the magic adversary, we freeze the scenario and branch into a loop in which the meta-reduction seeks a second valid message-signature pair. In order to avoid an exponential blow-up in the running time of such rewinding executions [15], we consider slightly restricted reductions.

Resetting Reductions with Restricted Cross-Resets. Any reduction in our case is allowed to run $q = q(n)$ concurrent executions with the copies of the adversary, each copy resetting at most q times, except that the reduction has to finish the interaction in the order according to the arrival of the second messages of the signature issue protocol. That is, consider a three-move signature issuing run of the reduction with a copy of the adversary playing the honest user. Assume that the reduction receives the second message in this execution (which has been sent by the adversary resp. user), and call this execution pending from then on. We say that the reduction successfully finishes this pending execution if it sends the third message of the protocol such that the user is able to derive a valid signature.

The cross-reset restriction now demands that, if the reduction ever finishes a pending execution successfully, then there is no other execution which has become pending and has been finished successfully meanwhile. In other words, between the pending state of an execution and its completion the reduction may not receive the second message and complete any other execution (for which the user can compute a signature). We remark that the reduction may decide to entirely abort a pending execution and is still allowed to finish other pending executions, as long as the user is unable to produce a signature from that interaction. A formal definition appears in the full version.

Note that the scheduling of reductions with restricted cross-resets is related to so-called bounded concurrent (and resettable) executions [3]. In m-bounded concurrent executions the number of instances running simultaneously is bounded by some fixed function $m = m(n)$ where the bound itself is known by the protocol. We do not put any a-priori bound on the number of concurrently running executions, because the number q of such instances depends on the reduction and is not bounded by a fixed polynomial. We merely restrict the way successful executions are finished. We also note that we can extend our proof below to allow a constant number of successfully finished executions between pending runs, but state and prove the simpler version for sake of readability.

q-wise Independent functions. An adequate measure to thwart reset attacks are usually pseudorandom functions (e.g., as in [10]). The idea is to make the randomness of the adversary depend on the communication by computing it as the output of the pseudorandom function for the communication. In this case, resetting the adversary essentially yields runs with independent random choices. Here, we use the same idea but can fall back to the weaker requirement of q-wise independent functions in order to avoid the additional assumption that pseudorandom functions exist.

We note that using q-wise independent functions instead of pseudorandom functions makes the adversary now depend on the reduction. Namely, below we use q as the number of maximal resets per row. However, since we deal with black-box reductions this is admissible. We also remark that we can overcome this dependency by using pseudorandom functions instead of q-wise independent function.

The New Magic Adversary. We use again the generic forgery oracle from the vanilla case. But here we augment our "new" magic adversary through a q-wise independent function (i.e., the random hash function h is given by parts of the adversary's randomness). Informally, the adversary again runs the issuing protocol with the signer in the role of the honest user once. However, it now generates the message (and the user's randomness) as the result of the q-wise independent function applied to the public key and the first message of the signer. Again, in the case that the single execution yields a valid signature, then the magic adversary here also creates another valid signature.

As we will later view Σ to be an integral part of the magic adversary and thus let the adversary provide the randomness $s \in \{0,1\}^{\psi(n)}$ required by oracle Σ. We denote this augmented (deterministic) oracle with Σ^{aug} which now takes pk, trans, m and randomness s as input and returns σ. This randomness is also derived through the q-wise independent function, ensuring consistent answers for the same data $(pk, \text{msg1})$. We note that the length $\psi(n)$ of this randomness is only polynomial by construction of the generic forgery oracle:

Definition 8 (Magic Adversary). *The magic adversary $\mathcal{A} = \mathcal{A}_q$ (with parameter q) for input pk and access to the generic forgery oracle Σ^{aug} and communicating with an oracle $\langle \mathcal{S}(sk), \cdot \rangle^1$ works as described in the following steps:*

```
select a hash function h from the family of q-wise independent functions $\mathcal{H}$
run an execution $\langle \mathcal{S}(sk), \mathcal{U}(pk, m_0'; r_0') \rangle$ in the role of an honest user, where
    $(m_0', m_1', r_0', s_0') \leftarrow h(pk, msg1)$ is generated as the result of the
    q-wise independent function applied to the public key pk and
    the first message msg1 of $\mathcal{S}$; let $\sigma_0'$ denote the resulting signature and
    $\mathsf{trans}_0'$ the corresponding transcript.
if $\mathsf{Vf}(pk, m_0', \sigma_0') = 1$ then let $\sigma_1' \leftarrow \Sigma^{aug}(pk, \mathsf{trans}_0', m_1'; s_0')$ else set $\sigma_0', \sigma_1' \leftarrow \perp$
return $(m_0', \sigma_0', m_1', \sigma_1')$
```

It follows again from the completeness of BS together with the construction of the generic forgery oracle that the magic adversary succeeds in the unforgeability experiment with probability negligibly close to 1.

5.2 Impossibility Result

In the following we extend our result to restricted-cross resets.

Theorem 2. *Let* BS *be a three-move blind signature scheme, which is statistically blind and has statistical signature-derivation checks. Then there is no resetting (with restricted cross-resets) black-box reduction from unforgeability of the blind signature scheme* BS *to a hard non-interactive problem.*

The proof appears in the full version. The idea is similar to the vanilla case. Only here we use the q-wise independent hash function to ensure independent randomness for runs with the adversary, and we need to take care of the fact that the meta-reduction now loops to find the second message-signature pair. The latter can be done in (expected) polynomial time by the assumption about the restricted resets. Appropriate truncation then yields a meta-reduction running in fixed polynomial time.

Transcript-independence again guarantees that the redcution cannot distinguish answers from the magic adversary from the ones of the meta-reduction. Formally, one first reduces the case of at most q instances, each with at most q resets, to a single run by a standard hybrid argument. Then one injects the data from the transcript-independence experiment into this single run. The signature derivation check allows to verify (without the help of Σ) if one has successfully inserted the data in a “good” execution (and not in a run in which the magic adversary would have failed to produce a forgery).

6 Conclusion

We have shown that for the blind signature schemes of Chaum [11] and of Pointcheval-Stern [27] finding security reductions to any non-interactive cryptographic problem in the standard model is hard. This class of cryptographic problems is very broad in the sense that it contains candidates like RSA and collision-resistant hash functions, and also any combination thereof. This also allows us to make stronger infeasibility claims compared to previous results using meta-reductions in other areas.

Concerning optimality of our results we remark that:

- Our result can be transfered to the computational blindness case (under additional stipulations), thus also ruling out many approaches to revert to computationally blindness to circumvent the results for the statistical schemes.
- Enlarging the class of cryptographic problems to interactive ones is too demanding: unforgeability of any blind signature scheme can indeed be securely reduced to an interactive problem in the standard model by simply assuming that the scheme is unforgeable. It is, however, unclear if and how decisional problems can be subsumed under our class of non-interactive (computational) problems.
- Extending the result to protocols with more moves is impossible in light of Okamoto’s scheme [26] with four moves in the standard model, based on a non-interactive assumption.

Hence, our result fits well into our current knowledge about constructing blind signatures and shows close boundaries for potential improvements on the efficiency or assumptions.

Acknowledgments

We thank the participants of the Dagstuhl Workshop on Cryptography, in particular Oded Goldreich for bringing up the connection to m-bounded concurrency and Salil Vadhan for pointing out the usage of q-wise independent functions instead of pseudorandom functions. We also thank the anonymous reviewers for valuable comments. Both authors are supported by the Emmy Noether Grant Fi 940/2-1 of the German Research Foundation (DFG), and by CASED (www.cased.de).

References

1. 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)
2. 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)
3. Barak, B.: How to Go Beyond the Black-Box Simulation Barrier. In: Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS) 2001, pp. 106–115. IEEE Computer Society Press, Los Alamitos (2001)
4. Bresson, E., Monnerat, J., Vergnaud, D.: Separation Results on the "One-More" Computational Problems. In: Malkin, T.G. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 71–87. Springer, Heidelberg (2008)
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. Boldyreva, A.: Efficient 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 (2003)
7. Brown, D.: What Hashes Make RSA-OAEP Secure? Number 2006/223 in Cryptology eprint archive (2006), eprint.iacr.org
8. D. Brown. Irreducibility to the One-More Evaluation Problems: More May Be Less. Number 2007/435 in Cryptology eprint archive (2007), eprint.iacr.org
9. Boneh, D., Venkatesan, R.: Breaking RSA May Be Easier Than Factoring. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 59–71. Springer, Heidelberg (1998)
10. Canetti, R., Goldreich, O., Goldwasser, S., Micali, S.: Resettable Zero-Knowledge. In: Proceedings of the Annual Symposium on the Theory of Computing (STOC) 2000, pp. 235–244. ACM Press, New York (2000)
11. Chaum, D.: Blind Signatures for Untraceable Payments. In: Advances in Cryptology — Crypto 1982, pp. 199–203. Plemum, New York (1983)
12. 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 (2004)
13. Camenisch, J., Neven, G., Shelat, A.: Simulatable Adaptive Oblivious Transfer. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 573–590. Springer, Heidelberg (2007)

14. 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)
15. Dwork, C., Naor, M., Sahai, A.: Concurrent zero-knowledge. J. ACM 51(6), 851–898 (2004)
16. 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)
17. 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)
18. Goldreich, O., Krawczyk, H.: On the composition of Zero-Knowledge Proof Systems. SIAM Journal on Computing 25(1), 169–192 (1996)
19. 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)
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. Impagliazzo, R., Rudich, S.: Limits on the Provable Consequences of One-Way Permutations. In: Proceedings of the Annual Symposium on the Theory of Computing, STOC 1989, pp. 44–61. ACM Press, New York (1989)
22. Juels, A., Luby, M., Ostrovsky, R.: Security of Blind Digital Signatures. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 150–164. Springer, Heidelberg (1997)
23. Kiayias, A., Zhou, H.-S.: Concurrent Blind Signatures Without Random Oracles. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 49–62. Springer, Heidelberg (2006)
24. 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)
25. Lindell, Y.: Bounded-Concurrent Secure Two-Party Computation Without Setup Assumptions. In: Proceedings of the Annual Symposium on the Theory of Computing, STOC 2003, pp. 683–692. ACM Press, New York (2003)
26. 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)
27. Pointcheval, D., Stern, J.: Security Arguments for Digital Signatures and Blind Signatures. Journal of Cryptology 13(3), 361–396 (2000)
28. Paillier, P., Villar, J.L.: Trading One-Wayness Against Chosen-Ciphertext Security in Factoring-Based Encryption. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 252–266. Springer, Heidelberg (2006)
29. Rückert, M.: Lattice-based Blind Signatures. Cryptology ePrint Archive, Report 2008/322 (2008) http://eprint.iacr.org/
30. Reingold, O., Trevisan, L., Vadhan, S.P.: Notions of Reducibility between Cryptographic Primitives. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 1–20. Springer, Heidelberg (2004)
31. Simon, D.: Finding 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)

Efficient Device-Independent Quantum Key Distribution*

Esther Hänggi[1], Renato Renner[2], and Stefan Wolf[1]

[1] Computer Science Department, ETH Zurich, CH-8092 Zürich, Switzerland
[2] Institute for Theoretical Physics, ETH Zurich, CH-8093 Zürich, Switzerland

Abstract. An efficient protocol for quantum key distribution is proposed the security of which is entirely device-independent and not even based on the accuracy of quantum physics. A scheme of that type relies on the quantum-physical phenomenon of *non-local correlations* and on the assumption that no illegitimate information flows within and between Alice's and Bob's laboratories. The latter can be enforced via the non-signaling postulate of relativity if all measurements are carried out simultaneously enough.

1 Non-locality, General Non-signaling Adversaries, and Device-Independent Secrecy

1.1 Minimizing Assumptions for Secure Key Agreement

It is well-established that secrecy must be based on certain premises such as a *limitation* on the adversary's computing power [2], [3] or memory [4], [5], *noise* in communication channels [6], [7], [8], the uncertainty principle of quantum physics [9], or entanglement [10]. In traditional quantum key distribution, the security proof is based on

1. the postulates of quantum physics,
2. the assumptions that the used devices work according to their specification, and
3. that Eve does not get information about the generated key out of the legitimate partners' laboratories.

This article is concerned with a variant of quantum key distribution which allows the first two assumptions to be dropped, if at the same time, the third is augmented by the assumption that no unauthorized information is exchanged within and between the legitimate laboratories. One possibility to guarantee this is via the non-signaling postulate of relativity if certain actions are carried out in a space-like separated[1] way. Of particular importance is *device independence*

[*] Because of space limitations, technical proofs are omitted in this extended abstract. The full proofs are given in [1].

[1] Two events, i.e., points in space-time, are called *space-like separated* if no signal at the speed of light, or smaller, can get from one to the other.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 216–234, 2010.

(i.e., dropping Condition 2), for two reasons. First, the necessity to trust the manufacturer is never satisfactory. Second, the security of traditional protocols for quantum key distribution relies *crucially* on the fact that the devices exactly match the theoretical model used in the security analysis, e.g., that a single photon source only emits always exactly one photon. For instance, the BB84 protocol [9] becomes *completely insecure* if larger systems, such as *pairs of photons*, are transmitted. With present technology, this is a significant issue. The fact that practical deviations from the theoretical model open the possibility of attacks has been demonstrated experimentally, see [11], [12], [13], [14], [15], [16], and references therein.

The question of *device-independent* security has been raised by Mayers and Yao in [17]. It was shown in [18] that such security is possible in principle. However, no non-zero secret-key rate has been achieved, and the classical-communication cost is exponential in the security parameter. Later schemes, robust against noise and achieving a positive key rate, have been proven secure against certain restricted types of attacks [19], [20], [21], [22]. The current state of the art is that security can hold against all attacks for which no (quantum) correlation is introduced between subsequent measurements, see, e.g., [23].

1.2 The Basic Idea: Systems, Correlations, and Non-locality

We explain the basic idea of achieving device-independent security by Barrett, Hardy, and Kent [18]. The resulting confidentiality is based on certain correlations — called *non-local* — between Alice and Bob.[2]

Non-locality is a property of the joint input-output behavior of two (or more) remote objects. Surprisingly, certain quantum states show such a behavior: The two parts of some *entangled states* display, under measurements, correlations unexplainable by shared classical information. This fact was observed by Bell [25] in 1964 and terminated attempts to completely describe quantum physics by local classical parameters, so-called *hidden variables*, as claimed by Einstein, Podolsky, and Rosen in 1935 [26]. It is, roughly speaking, exactly the non-existence of such hidden variables which can be exploited cryptographically: Information that does not *exist* can, in particular, not be *known* to an adversary (see Sect. 1.5).

In order to explain non-local correlations, we introduce the notion of a *two-party system*, defined by its joint input-output behavior $P_{XY|UV}$ (see Fig. 1).

Definition 1. A bipartite *system* is a conditional distribution $P_{XY|UV}$. It is *local* if $P_{XY|UV} = \sum_{i=1}^{n} w_i P^i_{X|U} P^i_{Y|V}$ for some $w_i \geq 0$ and distributions $P^i_{X|U}$ and $P^i_{Y|V}$, $i = 1, \ldots, n$. It is *non-signaling* if it does not allow for message transmission, i.e., if $\sum_x P_{XY|UV}(x, y, u, v) = \sum_x P_{XY|UV}(x, y, u', v)$ for all y, v, u, u', and similarly for the converse direction. A bipartite system that is non-signaling is also called a *non-signaling box.*

[2] Note that although classically the possibility to derive secrecy from correlations alone appears unusual, this is not so in quantum physics, since entanglement is monogamous to some extent [24]: If Alice and Bob are maximally entangled, then Eve factors out and must be independent.

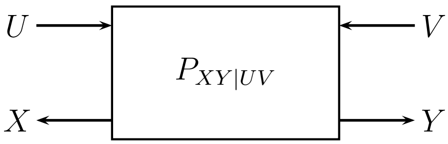


Fig. 1. A two-party *system*. If it does not allow for message transmission, it is called a *non-signaling box*.

Local systems are exactly what can be achieved with shared randomness: The randomness is equal to the i in the weighted sum. We will concentrate on systems that are *non-local and at the same time non-signaling*. It may be somewhat surprising that such systems exist, and we describe an example in Sect. 1.3. Note that throughout this paper, all systems are non-signaling boxes.

1.3 Non-locality Exists in Nature

In this section, we discuss a type of non-locality that exists in nature, named *CHSH* after [27]. For simplicity, we first discuss an idealization of that behavior, introduced by Popescu and Rohrlich [28] and called the *PR box* (see Fig. 2).

	U	0		1	
V	$Y \backslash X$	0	1	0	1
0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0
	1	0	$\frac{1}{2}$	0	$\frac{1}{2}$
1	0	$\frac{1}{2}$	0	0	$\frac{1}{2}$
	1	0	$\frac{1}{2}$	$\frac{1}{2}$	0

Fig. 2. The PR box

Definition 2. [**28**] A *Popescu-Rohrlich box* (or *PR box* for short) is the following bipartite system $P_{XY|UV}$: For each input pair (u, v), X is a random bit and $\text{Prob}\ [X \oplus Y = U \cdot V] = 1$.

Bell [25] showed this system to be non-local. More precisely, any system that behaves like a PR box with probability greater than 75% is. This can be seen as follows: Locality is equivalent to the possibility that the outputs to the two alternative inputs are pre-determined on each side. Let us call these bits X_0 (Alice's output if $U = 0$), X_1, and Y_0, Y_1, respectively. Now, $X \oplus Y = U \cdot V$ translates to the four contradictory conditions $X_0 = Y_0$, $X_1 = Y_0$, $X_0 = Y_1$, and $X_1 \neq Y_1$: Only three out of the four can be satisfied at a time!

The concept of a non-signaling box can now be used to investigate the properties of entangled quantum states. For this one considers a setting where Alice and Bob can choose local measurements, U and V respectively, and obtain outputs X and Y. Interestingly, in this model a PR box can be approximated by roughly 85%! In order to see this, note first that when the two Qbits of a system in the singlet state $|\psi^-\rangle := (|01\rangle - |10\rangle)/\sqrt{2}$ are measured in bases that enclose an angle of φ, then the probability of observing opposite measurement results is $\cos^2 \varphi$. The behavior of a PR box can be approximated with probability $\cos^2 22.5° \approx 85\%$ if the bases as shown in Fig. 3 are used, and if Bob flips his output bit. (Here, U_0 determines the measurement basis Alice uses upon getting input 0, etc.) This is optimal for all quantum states [29]. We have seen above that with shared (classical) information, at most 75% can be achieved; hence, nature *is* non-local!

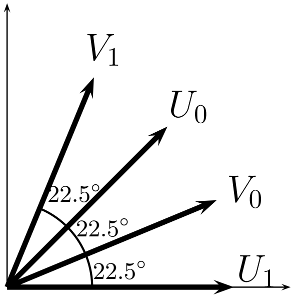


Fig. 3. Alice's and Bob's measurement bases for obtaining a 85%-PR box

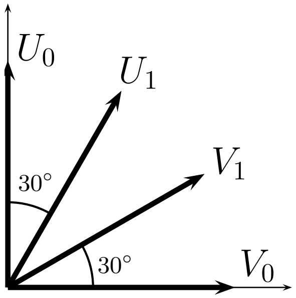


Fig. 4. The measurement bases used in Protocol 1

1.4 The General Non-signaling Adversary

We model an adversary as an additional interface to the non-signaling box, with the only restriction that the tripartite box is still non-signaling. In our security analysis, we will show that the key, generated by Alice and Bob by interacting with their respective parts of the non-signaling box, is *secure* in the sense that it is uniform and independent of all information accessible at this third interface. This model obviously puts minimal assumptions on the adversary: As usual in quantum key distribution, *Eve* may be in control of the entire environment, i.e., the complement of the two laboratories. Moreover, the information she has about what happens in these laboratories is *only* restricted by the non-signaling postulate: From the adversary's viewpoint (i.e., given all her information), no signaling can occur between space-like separated events, and no information is leaked out of the legitimate laboratories to the adversary. Note, in particular, that Eve is *not* assumed to be limited by quantum physics, *neither* is she assumed not to be the manufacturer of the devices used by Alice and Bob.

The non-signaling condition may be enforced by relativity, i.e., by carrying out the corresponding measurements in a space-like separated way. An alternative is to place every partial system into a shielded laboratory. Non-signaling is also a direct consequence of the assumption usually made in quantum key distribution that the Hilbert space is the tensor product of the Hilbert spaces associated with the local measurement processes of the parties and the dynamics factorizes.

We will see in Sect. 1.5 that in a non-local system, the non-signaling condition leads to a limitation on the bias of the system's outputs. When this fact is interpreted as being from an adversary's viewpoint, it represents a limitation on her information about these outputs: Bits that are *unbiased* for an adversary are *secret.*

1.5 Non-locality + Non-signaling = Limited Bias = Secrecy

The PR box is non-signaling: X and Y separately are perfectly random bits and independent of the input pair. On the other hand, as we show below, a system $P_{XY|UV}$ (where all variables are bits) satisfying $X \oplus Y = U \cdot V$ is non-signaling *only* if the outputs are completely unbiased, given the input pair, i.e., $P_{X|U=u,V=v}(0) = P_{Y|U=u,V=v}(0) = 1/2$. In other words, the output bit can not even be slightly biased, let alone pre-determined. Assume that Alice and Bob share some kind of physical system, carry out space-like separated measurements—hereby excluding message transmission—, and measure data having the statistics of a PR box. The outputs must then be perfectly secret bits because *even when conditioned on an adversary's complete information*, the correlation between Alice and Bob must still be non-signaling and fulfill $X \oplus Y = U \cdot V$.

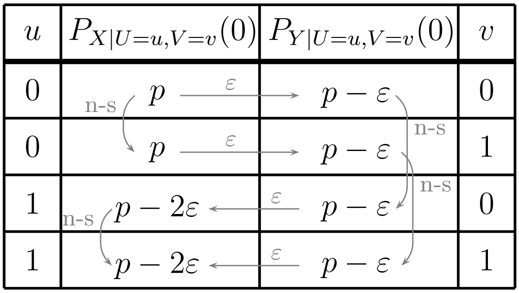
Unfortunately, the behavior of perfect PR boxes does not occur in nature: Quantum physics is non-local, but not maximally so. Can we also obtain secret bits from weaker, quantum-physical, non-locality? Barrett, Hardy, and Kent [18] have shown that the answer is *yes.* But their protocol is inefficient: In order to force the probability that the adversary learns a generated bit shared by Alice and Bob below ε, they have to communicate $\Theta(1/\varepsilon)$ Qbits.

If we measure maximally entangled quantum states, we can get at most 85%-approximations to the PR-box' behavior. Fortunately, *any* CHSH non-locality implies *some* secrecy. In order to illustrate this, consider a system approximating a PR box with probability $1-\varepsilon$ for all inputs. More precisely, we have

$$\text{Prob}\ [X \oplus Y = U \cdot V | U = u, V = v] = 1 - \varepsilon \qquad (1)$$

for all $(u,v) \in \{0,1\}^2$. Then, what is the maximal possible bias $p := \text{Prob}\ [X = 0|U = 0, V = 0]$ such that the system is non-signaling?

We explain the table (Fig. 5): Because of the $(1-\varepsilon)$-CHSH condition (1), the bias of Y, given $U = V = 0$, must be at least $p - \varepsilon$. Because of non-signaling, X's bias must be p as well when $V = 1$, and so on. Finally, the $(1-\varepsilon)$-CHSH condition for $U = V = 1$ implies $p - \varepsilon - (1 - (p - 2\varepsilon)) \leq \varepsilon$, hence, $p \leq 1/2 + 2\varepsilon$. For any $\varepsilon < 1/4$, this is a non-trivial bound. (This reflects the fact that $\varepsilon = 1/4$ is the "local limit.") In the special case of $\varepsilon = 0$ the bit is perfectly secret.



u	$P_{X\|U=u,V=v}(0)$	$P_{Y\|U=u,V=v}(0)$	v
0	p	$p-\varepsilon$	0
0	p	$p-\varepsilon$	1
1	$p-2\varepsilon$	$p-\varepsilon$	0
1	$p-2\varepsilon$	$p-\varepsilon$	1

n-s
ε
ε
n-s
n-s
ε
n-s
ε

Fig. 5. The maximal bias of the output of a $(1-\varepsilon)$-approximation of the PR box

1.6 Strong from Weak Secrecy

Conditioned on Eve's entire information, this reads: Weak non-locality means weak secrecy. Can it be amplified? *Privacy amplification* is a concept well-known from classical [30], [31], [32] and quantum [33], [34] cryptography, and means transforming a weakly secret string into a highly secret key by hashing. These results are, however, not applicable with respect to general non-signaling adversaries which may be strictly stronger than any quantum adversary. In [35], it has been pessimistically argued that privacy amplification of non-signaling secrecy is impossible, the problem being that certain collective attacks exist which leave the adversary with significant information about the final key, however the latter is obtained from the raw key.

Fortunately, the situation changes when one assumes an additional non-signaling condition between the individual measurements performed *within* Alice's as well as Bob's laboratories (see Fig. 8). This assumption could, for instance, be enforced by a space-like separation of the individual measurement events. In [36], Masanes has shown that in this case, privacy amplification is possible in principle — by hashing with a function chosen at random from the set of *all functions*.[3] Later, he has shown that it is sufficient to consider a two-universal set of functions (see [37], IV.C).

Our result differs from Masanes' in the sense that we show a *single explicit function*, namely the XOR, to be a good privacy-amplification function. More precisely, we prove that the adversary's probability of correctly predicting the XOR of the outcomes of n non-signaling boxes is exponentially (in n) close to 1/2 (Lemma 5). This can be seen as a generalization of the well-known fact that the XOR of many partially uniform bits is almost uniform, and may be of independent interest.

1.7 Our Protocol and Results

Protocol 1

1. Alice prepares $n+k$ Qbit pairs in the state $|\Psi^-\rangle = (|01\rangle - |10\rangle)/\sqrt{2}$, for suitable $k = \Theta(n)$, and sends one Qbit of every state to Bob.

[3] Masanes' result is a non-constructive proof of the fact that there exists a *fixed* function for privacy amplification.

2. Alice and Bob randomly measure the ith system in either the basis U_0 or U_1 (for Alice) and V_0 or V_1 (Bob);[4] the four bases are shown in Fig. 4. All $2(n+k)$ measurement events are *pairwise space-like separated.*
3. They randomly choose n of the measurement results from the instances where Alice has measured in U_0 and Bob in V_0. This forms the raw key.
4. For the remaining k measurements, they announce the results over the public channel and estimate the correlations. More precisely, they determine the parameter ε, where ε is the probability of violating the CHSH condition (i.e., $X \oplus Y \neq U \cdot V$) for uniform inputs, and δ, where δ is the probability of different outputs bits when U_0 and V_0 were measured. They also check whether they have obtained roughly the same number of 1's and 0's. If the parameters are such that key agreement is possible (Fig. 6), they continue; otherwise they abort.
5. Information reconciliation and privacy amplification: Alice randomly chooses an $(m+s) \times n$-matrix A such that $p(0) = p(1) = 1/2$ for all entries and $m := \lceil n \cdot h(\delta) \rceil$. She calculates $A \odot \mathbf{x}$ (where $\mathbf{x}$ is Alice's raw key) and sends the first m bits and the matrix A to Bob over the public authenticated channel. The remaining bits form the key. Bob uses the information received from Alice to reconstruct the key.

Theorem 1. *Protocol 1 achieves a positive secret-key-generation rate as soon as the parameter estimation shows an approximation of PR boxes with an accuracy*

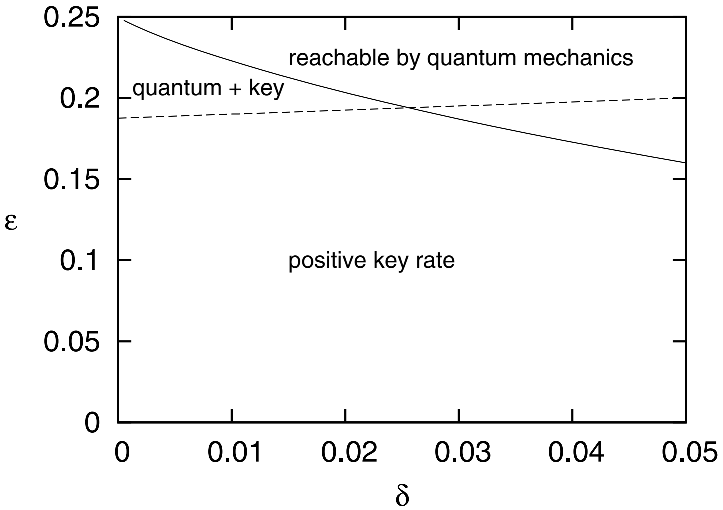


Fig. 6. The parameter regions for which key agreement is possible (below the solid line) and reachable by quantum mechanics (above the dashed line). ε is the probability of violating the CHSH condition (i.e., $X \oplus Y \neq U \cdot V$) for uniform inputs, and δ the probability of different output bits on input $(0,0)$.

[4] To increase the efficiency, the bases U_0 and V_0 may be choosen with very high probability, such that there are at least n positions where both Alice and Bob have measured in this basis.

exceeding 80% *and a correlation of the outputs on input* (0, 0) *higher than* 98%*, i.e., if* $\varepsilon \leq 0.2$ *and* $\delta \leq 0.02$*. The security of the protocol is based solely on the non-signaling condition; in particular, it is independent of quantum physics and of the devices used.*

Protocol 1 also allows for "traditional" entanglement-based quantum key agreement [10]. Therefore, we have the following.

Corollary 1. *Protocol 1 allows for efficient information-theoretic key agreement if quantum* or *relativity theory is correct.*

2 Model and Security Definition

2.1 Modeling the Attacks

When Alice, Bob, and Eve carry out measurements on a (joint) physical system, they can choose their measurement settings (the inputs) and receive their respective outcomes (the outputs). It is, therefore, natural to model the situation by a tripartite system, characterized by $P_{XYZ|UVW}$ as depicted in Fig. 7. Our security analysis will be based on the *non-signaling* condition, i.e., the input/output behavior of one side tells nothing about the input on the other side(s) (the same must also hold with respect to a separation of all interfaces in two groups).

Condition 1. *[18] The system* $P_{XYZ|UVW}$ *must not allow for signaling:*

$$\sum_x P_{XYZ|UVW}(x,y,z,u,v,w) = \sum_x P_{XYZ|UVW}(x,y,z,u',v,w)$$

for all u, u', y, z, v, w *and similarly for signaling in all other directions.*

If a system is non-signaling between its interfaces, this also means that its marginal systems are well-defined: What happens at one of the interfaces does not depend on any other input. This implies that at all the interfaces, an output can always be provided immediately after the input has been given.

This tripartite scenario can be reduced to a bipartite one: Because Eve cannot signal to Alice and Bob (even together) by her choice of input, we must have

$$\sum_z P_{XYZ|UVW}(x,y,z,u,v,w) = P_{XY|UV}(x,y,u,v) \text{ for all } w \text{ ,}$$

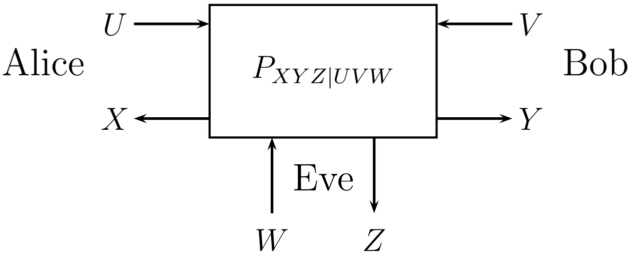


Fig. 7. The tripartite scenario including the eavesdropper

and the right-hand side is exactly the marginal box as seen by Alice and Bob. We can, therefore, see Eve's input as a choice of convex decomposition of Alice's and Bob's box, and her output as indicating one part of the decomposition. Furthermore, the condition that even Alice and Eve together must not be able to signal to Bob and *vice versa* means that the distribution conditioned on Eve's outcome, $P^z_{XY|UV}$, must also be non-signaling between Alice and Bob. Informally, we can write

$$\boxed{A\ \ B} = p(z_0|w)\cdot \boxed{A\ \ B\ {}_{z_0}} + p(z_1|w)\cdot \boxed{A\ \ B\ {}_{z_1}} + \cdots$$

and this also covers all possibilities available to Eve. Formally, we define:

Definition 3. A *box partition* of a given bipartite non-signaling box $P_{XY|UV}$ is a family of pairs $(p^z,P^z_{XY|UV})$, where p^z is a weight and $P^z_{XY|UV}$ is a non-signaling box, such that $P_{XY|UV} = \sum_z p^z \cdot P^z_{XY|UV}$.

This definition allows us to switch between the scenario of a bipartite non-signaling box plus box partition and the scenario of a tripartite non-signaling box, as stated in Lemmas 1 and 2.

Lemma 1. *For any given tripartite non-signaling box $P_{XYZ|UVW}$, any input w induces a box partition of the bipartite box $P_{XY|UV}$ parametrized by z with $p^z := p(z|w)$ and $P^z_{XY|UV} := P_{XY|UV,Z=z,W=w}$.*

Lemma 2. *Given a bipartite non-signaling box $P_{XY|UV}$, let $\mathcal{W}$ be a set of box partitions $w = \{(p^z, P^z_{XY|UV})\}_z$. Then the tripartite box, where the input of the third party is $w \in \mathcal{W}$, defined by $P_{XYZ|UV,W=w}(z) := p^z \cdot P^z_{XY|UV}$ is non-signaling and has marginal box $P_{XY|UV}$.*

As explained in the introduction, it is crucial for our security analysis to assume that Alice and Bob have several input/output interfaces (whereas Eve's inputs and output may have an arbitrary structure). We then require the non-signaling condition to hold between all of the interfaces. We, therefore, extend Condition 1 from the tripartite to the $(2n + 1)$-partite case in the obvious way and call such a system $(2n + 1)$*-partite non-signaling* (see Fig. 8).

In order to study our particular protocol described in Sect. 1.7 we consider the case where Alice and Bob share $2n$ interfaces, each taking one bit input and giving one bit output.[5] Each input bit corresponds to the choice of a basis applied to measure one part of an entangled state and the output bit corresponds to the measurement result. In the case of a passive adversary, the distribution will

[5] We will write U for the random bit denoting Alice's input, bold-face letters $\mathbf{U}$ will denote an n-bit random variable (i.e., an n-bit vector), U_i a single random bit in this n-bit string, and lowercase letters the value that the random variable has taken. A similar notation is used for Alice's output X and Bob's input and output V and Y. No assumption is made about the range of Eve's input/output variables W and Z.

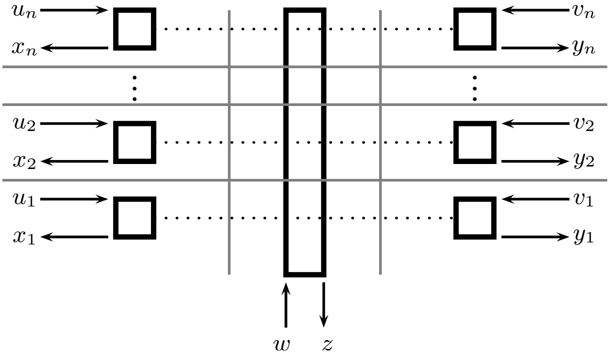


Fig. 8. Alice and Bob share n non-signaling boxes which are independent from their viewpoint. However, Eve can attack all of them at once. The gray lines stand for the non-signaling condition.

approximate the behavior of n non-local boxes. To prove security, however, we cannot make any assumptions about the distribution (which may be arbitrarily influenced by an adversary[6]). For this reason, our security proof only relies on the non-signaling condition, which we now reformulate for this specific case.

Condition 1′. The system $P_{\mathbf{XY}Z|\mathbf{UV}W}$ must not allow for signaling between any of the $2n+1$ marginal systems, i.e.,

$$\sum_{x_i} P_{\mathbf{XY}Z|\mathbf{UV}W}(\mathbf{x},\mathbf{y},z,\mathbf{u}\backslash u_i,u_i,\mathbf{v},w) = \sum_{x_i} P_{\mathbf{XY}Z|\mathbf{UV}W}(\mathbf{x},\mathbf{y},z,\mathbf{u}\backslash u_i,u_i',\mathbf{v},w)$$

for all $\mathbf{x}\backslash x_i, \mathbf{y}, z, , \mathbf{u}\backslash u_i, u_i, u_i', \mathbf{v}, w$, and where we used the notation $\mathbf{x}\backslash x_i$ to abbreviate $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots x_n$, i.e., all x_j for which $j \neq i$ (and similarly for signaling in all other directions).

Note that the set of possible attacks of an adversary is determined by Condition 1′ only. More precisely, the adversary, Eve, could choose an arbitrary behavior of the non-signaling box $P_{XYZ|UVW}$ satisfying Condition 1′ and has full access to the interface taking input W and giving output Z.

2.2 Security Definition

We define security in the context of *random systems* [38]. The closeness of two systems $\mathcal{S}_0$ and $\mathcal{S}_1$ can be measured by introducing a so-called *distinguisher*. A distinguisher $\mathcal{D}$ is itself a system, and it can interact with the other system. Assume the distinguisher is given at random either system $\mathcal{S}_0$ or $\mathcal{S}_1$; after interacting with the system, the distinguisher outputs a bit guessing whether it has interacted with system $\mathcal{S}_0$ or $\mathcal{S}_1$. The *distinguishing advantage between system* $\mathcal{S}_0$ *and* $\mathcal{S}_1$ is the maximum guessing advantage any distinguisher can have in this game.

[6] This scenario is analogous to Eve being able to do coherent attacks in a quantum key distribution protocol.

Definition 4. The *distinguishing advantage between two systems* $\mathcal{S}_0$ *and* $\mathcal{S}_1$ is

$$\delta(\mathcal{S}_0, \mathcal{S}_1) = \max_{\mathcal{D}}[P(B = 1|\mathcal{S} = \mathcal{S}_0) - P(B = 1|\mathcal{S} = \mathcal{S}_1)] \ .$$

Two systems $\mathcal{S}_0$ and $\mathcal{S}_1$ are called ϵ-indistinguishable if $\delta(\mathcal{S}_0, \mathcal{S}_1) \leq \epsilon$.

The probability of any event $\mathcal{E}$, defined by any of the input and output variables, when the distinguisher $\mathcal{D}$ is interacting with $\mathcal{S}_0$ or $\mathcal{S}_1$ cannot differ by more than this quantity. The reason is that otherwise this event could be used to distinguish the two systems.

Lemma 3. *Let* $\mathcal{S}_0$ *and* $\mathcal{S}_1$ *be* ϵ*-indistinguishable systems. Denote by* $P(\mathcal{E}|\mathcal{S}_0, \mathcal{D})$ *the probability of an event* $\mathcal{E}$*, defined by any of the input and output variables, given the distinguisher is interacting with the system* $\mathcal{S}_0$*. Then* $P(\mathcal{E}|\mathcal{S}_0, \mathcal{D}) \leq P(\mathcal{E}|\mathcal{S}_1, \mathcal{D}) + \epsilon$*.*

The security of a cryptographic primitive can be measured by its distance from an *ideal* system which is secure by definition. For example in the case of key distribution, the ideal system is the one which outputs a uniform and random key (bit string) S at one end and for which all other input/output interfaces are completely independent of this first interface. This key is secure by construction. If the *real* system generating a key is indistinguishable from the *ideal* one, this key is called secure.

Definition 5. *A key* S *is* ϵ-secure *if the system outputting* S *is* ϵ*-indistinguishable from an ideal system which outputs a uniform random variable* S *and for which all other input/output interfaces are completely independent of the random variable* S*.*

As a consequence of Lemma 3, the resulting security is *composable* [39], [40], [41].

For the security analysis, we consider an entanglement-based version of Protocol 1 (Sect. 1.7). This means that the protocol starts with step 2 and it is assumed that the $n + k$ quantum states have already been pre-distributed (possibly by an adversary). As described in Sect. 2.1, these states are modeled as non-signaling boxes. We model the public authenticated channel connecting Alice and Bob as an additional (signaling) system, as depicted in Fig. 9. Eve can wire-tap the public channel, choose an input on her part of the non-signaling box and obtain an output (i.e., measure her part of the quantum state). Similar to the quantum case, it is no advantage for Eve to make several box partitions (measurements) instead of a single one, as the same information can be obtained by making a refined box partition of the initial box. Without loss of generality, we can, therefore, assume that Eve gives a single input to the non-signaling box at the end (after all communication between Alice and Bob is finished). In our scenario, Eve, therefore, obtains all the communication exchanged over the public channel Q, can then choose the input to her interface of the non-signaling box W (which can depend on Q), and finally obtains the outcome of the box Z. As shown in Fig. 9, we may also define a lager box $\mathcal{S}_{real}$ which includes the behavior of the protocol executed by Alice and Bob and outputs S_A and S_B. According to

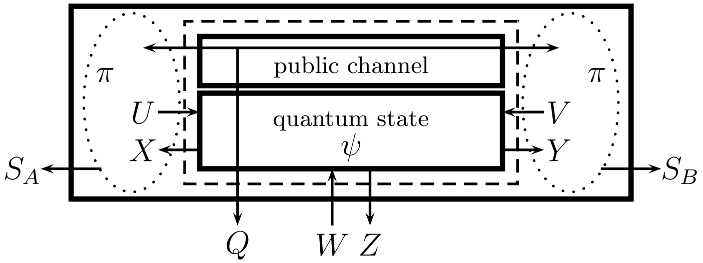


Fig. 9. Our system. Alice and Bob share a public authentic channel and a quantum state. When they apply a protocol π to obtain a key, all this can together be modeled as a system.

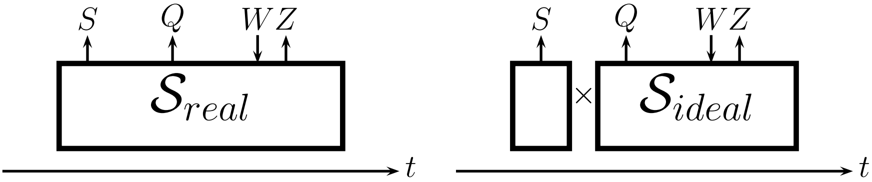


Fig. 10. An illustration of the security protocol: the real system (left) is compared to the ideal system (right). The distribution of S in the ideal case is $P_S(s) = 1/|\mathcal{S}|$.

Definition 5, the key S_A[7] is secure if the system $\mathcal{S}_{real}$ is ϵ-indistinguishable from the ideal system (see Fig. 10). For the security analysis it is useful to formulate this definition in terms of the distance from uniform.

Definition 6. *The* distance from uniform of S given $Z(w)$ and Q *is*

$$d(S|Z(w),Q) = 1/2 \sum_{s,q} \max_{w} \sum_{z} P_{Z,Q|W=w}(z,q) \cdot |P_{S|Z=z,Q=q,W=w}(s) - P_U| \ .$$

We have written $Z(w)$ because the eavesdropper can choose the input adaptively, and the choice of input changes the output distribution.

It is then straightforward to show the following Lemma 4.

Lemma 4. *A key S generated by a system as given in Fig. 9 is ϵ-secure if $d(S|Z(w),Q) \leq \epsilon$.*

3 Privacy Amplification

In this section, we prove the main technical result. We consider the situation where Alice and Bob share n imperfect PR boxes, and the key is computed by taking

[7] Note that we can consider the distance of S_A from an ideal key and the distance between S_A and S_B (probability of the keys to be unequal) separately. By the triangle inequality, the distance of the total real system from the ideal system is at most the sum of the two.

the XOR of all n output bits. We will show that taking the XOR of the outputs of several non-signaling boxes is a good privacy-amplification function in the sense that the resulting bit is almost perfectly secret (for sufficiently large n).

We now start with the statement and proof of our main claim.

Lemma 5. *Let a* $(2n+1)$*-partite non-signaling box* $P_{\boldsymbol{XYZ}|\boldsymbol{UV}W}$, $f(\boldsymbol{X}) := \bigoplus_i X_i$ *and* $Q := (\boldsymbol{U} = \boldsymbol{u}, \boldsymbol{V} = \boldsymbol{v})$*. Then*

$$d(f(\boldsymbol{X})|Z(W), Q) \leq 1/2 \cdot \sum_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{u},\boldsymbol{v}:\ x_i \oplus y_i \neq u_i \cdot v_i\ \ \forall i} P_{\boldsymbol{XY}|\boldsymbol{UV}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{v})\ .$$

Note that Alice and Bob estimate the average probability that their non-signaling boxes deviate from the perfect CHSH condition. Conditioned on this estimate of ε, the right-hand side is approximately equal to $1/2 \cdot (4\varepsilon)^n$.

We proceed in several steps. First, we show that the problem of finding the maximum distance from uniform of the XOR of several output bits can be cast as a linear optimization problem. Then, we show that this linear program describing n non-signaling boxes can be seen as the n-wise tensor product of the linear program describing a single non-signaling box — this is the crucial step. By using the product form of the linear program we can then show that there exists a dual feasible solution — i.e., an upper-bound on the distance from uniform — reaching the above value.

First note that, because of convexity, the maximal possible non-uniformity of the XOR of the output bits can be obtained by a box partition with only two outputs, 0 and 1. It is, therefore, sufficient to consider a box partition with only two elements $z = 0$ and $z = 1$. However, given one element of the box partition $(p, P^{Z=0}_{\mathbf{XY}|\mathbf{UV}})$, the second element $(1-p, P^{Z=1}_{\mathbf{XY}|\mathbf{UV}})$ is determined because their convex combination forms the marginal box, $P_{\mathbf{XY}|\mathbf{UV}}$. The distance from uniform of a random bit from the adversary's point of view can then be expressed only in terms of the one element of the box partition as

$$d(\bigoplus_i X_i|Z(\bar{w}), Q) = 2 \cdot p \cdot (P[\bigoplus_i X_i = 0|Z = 0, Q] - 1/2)\ .$$

This implies that finding the distance from uniform is equivalent to finding the "best" element of a box partition $(p, P^{Z=0}_{\mathbf{XY}|\mathbf{UV}})$. When can $(p, P^{Z=0}_{\mathbf{XY}|\mathbf{UV}})$ be element of a box partition? The criterion is given in Lemma 6. It follows from the positivity of probabilities and the linearity of the non-signaling conditions.

Lemma 6. *A non-signaling box* $P_{\boldsymbol{XY}|\boldsymbol{UV}}$ *has a box partition with element* $(p, P^{Z=0}_{\boldsymbol{XY}|\boldsymbol{UV}})$ *if and only if for all inputs and outputs* $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{v}$*,*

$$p \cdot P^{Z=0}_{\boldsymbol{XY}|\boldsymbol{UV}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{v}) \leq P_{\boldsymbol{XY}|\boldsymbol{UV}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{v})\ .$$

We can now show that the maximal distance from uniform which can be reached by a non-signaling adversary is the solution of a linear programming problem (see, e.g., [42] for a good introduction to linear programming). We introduce a new variable Δ. $\Delta(\mathbf{x}, \mathbf{y}|\mathbf{u}, \mathbf{v})$ can be defined as $2p \cdot P^{Z=0}(\mathbf{xy}|\mathbf{uv}) - P(\mathbf{xy}|\mathbf{uv})$.[8]

[8] In the following, we write $P(\mathbf{xy}|\mathbf{uv})$ instead of $P_{\mathbf{XY}|\mathbf{UV}}(\mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{v})$.

Lemma 7. *The distance from uniform of $\bigoplus_i X_i$ given $Z(W)$ and $Q := (\boldsymbol{U} = \boldsymbol{u}, \boldsymbol{V} = \boldsymbol{v})$ is*

$$d(\bigoplus_i X_i|Z(W), Q) = 1/2 \cdot b^T \cdot \Delta^* \ ,$$

where $b^T \cdot \Delta^*$ is the optimal value of the linear program

$$\begin{aligned}
\textit{max:}\quad & \sum_{(\boldsymbol{x},\boldsymbol{y}):f(\boldsymbol{x})=0} \Delta(\boldsymbol{xy}|\boldsymbol{uv}) - \sum_{(\boldsymbol{x},\boldsymbol{y}):f(\boldsymbol{x})=1} \Delta(\boldsymbol{xy}|\boldsymbol{uv}) \\
\textit{s.t.:}\quad & \sum_{\boldsymbol{x}} \Delta(\boldsymbol{xy}|\boldsymbol{uv}) - \sum_{\boldsymbol{x}} \Delta(\boldsymbol{xy}|\boldsymbol{u'v}) = 0 \ \forall \boldsymbol{y}, \boldsymbol{v}, \boldsymbol{u}, \boldsymbol{u'} \ \ \textit{(non-sig. Alice to Bob)} \\
& \sum_{\boldsymbol{y}} \Delta(\boldsymbol{xy}|\boldsymbol{uv}) - \sum_{\boldsymbol{y}} \Delta(\boldsymbol{xy}|\boldsymbol{uv'}) = 0 \ \forall \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{v'} \ \ \textit{(non-sig. Bob to Alice)} \\
& \Delta(\boldsymbol{xy}|\boldsymbol{uv}) \leq P(\boldsymbol{xy}|\boldsymbol{uv}) \ \ \forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{v} \ \ \textit{(Lemma 6)} \\
& \Delta(\boldsymbol{xy}|\boldsymbol{uv}) \geq -P(\boldsymbol{xy}|\boldsymbol{uv}) \ \ \forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{v} \ \ \textit{(positivity of probabilities)} \ .
\end{aligned}$$

Note that there is no normalization constraint on Δ because normalization follows from the non-signaling constraints. This linear program can easily be brought into the form

$$\begin{array}{lc} \text{max:} & b^T \cdot \Delta \\ \text{s.t.:} & A \cdot \Delta \leq c \end{array} \qquad \text{and its dual} \qquad \begin{array}{lc} \text{min:} & c^T \cdot \lambda \\ \text{s.t.:} & A^T \cdot \lambda = b \\ & \lambda \geq 0 \end{array} \tag{2}$$

Note that in the *dual* program, the marginal box as seen by Alice and Bob only appears in the objective function $c^T \cdot \lambda$. The feasible region is, therefore, completely independent of the marginal.

For the case of a single non-signaling box, A_1, b_1 and c_1 explicitly have the form

$$A_1 = \begin{pmatrix} A_1^{\text{n-s}} \\ -A_1^{\text{n-s}} \\ \mathbb{1}_{16} \\ -\mathbb{1}_{16} \end{pmatrix} \ , \qquad \begin{aligned} b_1 &= \begin{pmatrix} 1\,1\,0\,0\,{-1}\,{-1}\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \end{pmatrix} \\ c_1 &= \begin{pmatrix} 0_{16}\ 0_{16}\ P(xy|uv)\ P(xy|uv) \end{pmatrix} \ , \end{aligned}$$

$$\text{with } A_1^{\text{n-s}} = \begin{pmatrix}
1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1
\end{pmatrix} ,$$

and where $P(xy|uv)$ are the probabilities of Alice's and Bob's marginal box such as, for example, given in Fig. 11 below, but with the rows stack on top of each other to form a vector. The dual optimal solution λ_1 can easily be calculated as

$$\lambda_1^{*T} = \begin{pmatrix} 0.5\ 0\ 0.5\ 0\ 0.5\ 0\ 0.5\ 0\ 0\ 0.5\ 0\ 0.5\ 0\ 0.5\ 0\ 0.5 \\ 0\,1\,0\,1\,0\,0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0\,1 \end{pmatrix} .$$

By comparison, we see that for every x, y, u, v such that $x \oplus y \neq u \cdot v$, there is exactly one 1 in the second part of λ_1^* and everywhere else λ_1^* is 0. I.e., $c_1^T \cdot \lambda_1^* = \sum_{x,y,u,v: x \oplus y \neq u \cdot v} P_{XY|UV}(x, y, u, v)$.

Our main tool to show Lemma 5 will be to note that we can express the linear program describing n non-signaling boxes as the tensor product of the linear program describing one non-signaling box.

Lemma 8. *Denote by A_1, b_1 the vector and matrix associated with the linear program (2) for the case of a single non-signaling box. Then the value of the program A, b, c associated with n non-signaling boxes is equal to the value of the linear program defined by*

$$\begin{aligned} \textit{max:} \quad & (b_1^{\otimes n})^T \cdot \Delta \\ \textit{s.t.:} \quad & A_1^{\otimes n} \cdot \Delta \leq c \ . \end{aligned} \tag{3}$$

Now we consider the dual program of (3). It follows directly from its form that if λ_1 is a feasible dual solution for a single non-signaling box, then $\lambda_1^{\otimes n}$ is feasible for n non-signaling boxes.

Lemma 9. *For any λ_i which is dual feasible for the linear program A_1, b_1 associated with one non-signaling box, $\bigotimes_i \lambda_i$ is dual feasible for the linear program (3) associated with n non-signaling boxes.*

Inserting the explicit value of $\lambda = \lambda_1^{\otimes n}$ into the objective function $c^T \cdot \lambda$ concludes the proof of Lemma 5.

4 Full Key Agreement

4.1 Information Reconciliation and Privacy Amplification: From One to Several Bits

We have seen in Sect. 3 that it is possible to create a highly secure bit using a linear function — the XOR. But obviously, we would like to extract a secure key string, not only a single bit. Furthermore, Alice's and Bob's raw key bits (the output of the non-signaling boxes) will differ with some probability δ, therefore, they need to do information reconciliation before extracting the secret key. Both information reconciliation and privacy amplification can be done the same way: by applying a random linear function to the output bits, i.e., $[R, S] := A \odot \mathbf{X}$, where A is a $(r + s) \times n$-matrix over $GF(2)$ with $p(0) = p(1) = 1/2$ for all entries and we write $\odot$ for the matrix multiplication modulo 2. The first r bits R are released for information reconciliation, while the last s bits form the final key S.

It follows from a result of [43] about two-universal sets of hash functions and from a result of [44] about information reconciliation that in the limit of large n, $r = \lceil n \cdot h(\delta) \rceil$ (where δ is the probability that Bob's bit is different from Alice's, and h is the binary entropy function) is both necessary and sufficient for Bob to be able to correct the errors in his raw key.

In order to show that the key S is secure, we show that it is secure even given the bits R of the information-reconciliation scheme are released. Using the triangle inequality, we can reduce the question of the security of the whole key to the question of the security of each of the bits S_i, given all previous bits $S_1, \ldots, S_{i-1}$ and R. We then derive a bound on the distance from uniform of S using Lemma 5.

Lemma 10. *Let a $(2n+1)$-partite non-signaling box $P_{\boldsymbol{XYZ}|\boldsymbol{UVW}}$ such that the estimated average error is ε. Let $[R, S] := A \odot \boldsymbol{X}$, where A is a $(r+s) \times n$-matrix over $GF(2)$, and P_A the uniform distribution over all these matrices. $Q := (\boldsymbol{U} = \boldsymbol{u}, \boldsymbol{V} = \boldsymbol{v}, A)$. Then*

$$d(S|Z(W), Q, R) \leq 1/2 \cdot 2^{r+s} \cdot \left(\frac{1+4\varepsilon}{2}\right)^n .$$

4.2 Key Rate

The key rate is the length of the key divided by the number of non-signaling boxes used in the limit of a large number of boxes. Because we only need a small number of boxes for parameter estimation [45], this will asymptotically correspond to $q := s/n$. From Lemma 10 we can calculate the key rate by setting $r := h(\delta) \cdot n$ (see Sect. 1.7 for a detailed description of the Protocol 1).

Lemma 11. *Protocol 1 reaches a key rate q of*

$$q = 1 - h(\delta) - \log_2(1 + 4\varepsilon) . \tag{4}$$

Key agreement is possible if the parameters ε and δ are such that this quantity is positive, i.e., $\varepsilon < 2^{-h(\delta)-1} - 1/4$ (see Fig. 6).

4.3 The Quantum Regime

If the non-signaling boxes have the same error ε for all inputs, then $\delta = \varepsilon$ in (4) and the protocol does not reach a positive secret key rate for $\varepsilon = \frac{1+\sqrt{2}}{4}$, the minimum value reachable by quantum mechanics. In order to avoid this problem, we have chosen the bases in Protocol 1 (see Sect. 1.7) such that the corresponding non-signaling box gives highly correlated output bits given input $(0, 0)$ (see Fig. 11). Alice and Bob generate their raw key only from these outputs.[9] Note that in a noiseless setting, the distribution described in black font can be achieved by measuring a singlet state. In that case, Alice and Bob will have perfectly correlated bits (and, therefore, would not need to do any information reconciliation), and the parameter limiting Eve's knowledge is $\varepsilon = 0.1875$. The parameters δ and η (in light gray font in Fig. 11) are introduced to account for possible noise that may arrise in the practical realization of the scheme.

[9] Another way to reach a positive key rate in the quantum regime is to use a type of non-locality characterized by a different Bell inequality allowing for a higher violation in the quantum regime. See [36] for details.

U		0		1	
V	$Y \backslash X$	0	1	0	1
0	0	$\frac{1}{2}-\frac{\delta}{2}$	$\frac{\delta}{2}$	$\frac{3}{8}-\frac{\eta}{2}$	$\frac{1}{8}+\frac{\eta}{2}$
	1	$\frac{\delta}{2}$	$\frac{1}{2}-\frac{\delta}{2}$	$\frac{1}{8}+\frac{\eta}{2}$	$\frac{3}{8}-\frac{\eta}{2}$
1	0	$\frac{3}{8}-\frac{\eta}{2}$	$\frac{1}{8}+\frac{\eta}{2}$	$\frac{1}{8}+\frac{\eta}{2}$	$\frac{3}{8}-\frac{\eta}{2}$
	1	$\frac{1}{8}+\frac{\eta}{2}$	$\frac{3}{8}-\frac{\eta}{2}$	$\frac{3}{8}-\frac{\eta}{2}$	$\frac{1}{8}+\frac{\eta}{2}$

Fig. 11. The quantum box used for key agreement

5 Concluding Remarks and Open Questions

We propose a new efficient protocol for generating a secret key between two parties connected by a quantum channel whose security is guaranteed *solely* by the fact that no information is exchanged between the different measurement events. The method is based on non-locality which can be generated from entangled quantum states. The security proof, on the other hand, is *independent* of quantum physics once the non-local correlations are established and have been verified.

The *practical* relevance is that the resulting security is *device-independent*: We could even use devices manufactured by the adversary to do key agreement. The *theoretical* relevance is that the resulting protocol is secure if *either relativity or quantum theory is correct.* This is in the spirit of modern cryptography's quest to minimize assumptions on which security rests.

Our scheme requires space-like separation not only between events happening on Alice's and Bob's side, but also between events within the same laboratory. It is a natural open question whether the space-like-separation conditions can be relaxed. For instance, is it sufficient if they hold on one of the two sides? Or in one direction among the n events on each side? Obviously, the latter would be easy to guarantee in practice.

Acknowledgments. We thank Roger Colbeck, Matthias Fitzi, Severin Winkler, and Jürg Wullschleger for helpful discussions, and Hoi-Kwong Lo for bringing reference [17] to our attention. We also thank three anonymous reviewers as well as Ivan Damgård for their help in improving the presentation of our results. EH and SW are supported by the Swiss National Science Foundation (SNF) and by the ETH research commission. RR acknowledges support from the Swiss National Science Foundation (grant No. 200021-119868).

References

1. Hänggi, E., Renner, R., Wolf, S.: Quantum cryptography based solely on Bell's theorem (2009), arxiv:quant-ph/0911.4171
2. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Trans. on Information Theory 22(6), 644–654 (1976)

3. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21(2), 120–126 (1978)
4. Maurer, U.: A provably-secure strongly-randomized cipher. In: Damgård, I.B. (ed.) EUROCRYPT 1990. LNCS, vol. 473, pp. 361–373. Springer, Heidelberg (1991)
5. Dziembowski, S., Maurer, U.: The bare bounded-storage model: The tight bound on the storage requirement for key agreement. IEEE Trans. on Information Theory 54(6), 2790–2792 (2008)
6. Wyner, A.D.: The wire-tap channel. Bell System Technical J 54(8), 1355–1387 (1975)
7. Csiszár, I., Körner, J.: Broadcast channels with confidential messages. IEEE Trans. on Information Theory 24(3), 339–348 (1978)
8. Maurer, U.: Conditionally-perfect secrecy and a provably-secure randomized cipher. J. of Cryptology 5(1), 53–66 (1992)
9. Bennett, C.H., Brassard, G.: Quantum cryptography: public key distribution and coin tossing. In: Int. Conf. on Computers, Systems and Signal Processing (1984)
10. Ekert, A.K.: Quantum cryptography based on Bell's theorem. Phys. Rev. Lett. 67(6), 661–663 (1991)
11. Gisin, N., Fasel, S., Kraus, B., Zbinden, H., Ribordy, G.: Trojan-horse attacks on quantum-key-distribution systems. Phys. Rev. A 73(2), 022320 (2006)
12. Fung, C.H.F., Qi, B., Tamaki, K., Lo, H.K.: Phase-remapping attack in practical quantum-key-distribution systems. Phys. Rev. A 75(3), 032314 (2007)
13. Qi, B., Fung, C.H.F., Lo, H.K., Ma, X.: Time-shift attack in practical quantum cryptosystems. Quantum Information and Computation 7, 073–082 (2007)
14. Zhao, Y., Fung, C.H.F., Qi, B., Chen, C., Lo, H.K.: Quantum hacking: Experimental demonstration of time-shift attack against practical quantum-key-distribution systems. Phys. Rev. A 78(4), 042333 (2008)
15. Makarov, V.: Controlling passively quenched single photon detectors by bright light. New J. of Physics 11(6), 065003 (2009)
16. Scarani, V., Kurtsiefer, C.: The black paper of quantum cryptography: real implementation problems (2009)
17. Mayers, D.C., Yao, A.: Quantum cryptography with imperfect apparatus. In: FOCS 1998, pp. 503–509 (1998)
18. Barrett, J., Hardy, L., Kent, A.: No signalling and quantum key distribution. Phys. Rev. Lett. 95, 010503 (2005)
19. Acín, A., Massar, S., Pironio, S.: Efficient quantum key distribution secure against no-signalling eavesdroppers. New J. of Phys. 8(8), 126 (2006)
20. Scarani, V., Gisin, N., Brunner, N., Masanes, L., Pino, S., Acín, A.: Secrecy extraction from no-signalling correlations. Phys. Rev. A 74(4), 042339 (2006)
21. Acín, A., Gisin, N., Masanes, L.: From Bell's theorem to secure quantum key distribution. Phys. Rev. Lett. 97, 120405 (2006)
22. Acín, A., Brunner, N., Gisin, N., Massar, S., Pironio, S., Scarani, V.: Device-independent security of quantum cryptography against collective attacks. Phys. Rev. Lett. 98, 230501 (2007)
23. McKague, M.: Device independent quantum key distribution secure against coherent attacks with memoryless measurement devices. New J. of Phys. 11(10), 103037 (2009)
24. Terhal, B.M.: Is entanglement monogamous? IBM J. of Research and Development 48(1), 71–78 (2004)
25. Bell, J.S.: On the Einstein-Podolsky-Rosen paradox. Physics 1, 195–200 (1964)
26. Einstein, A., Podolsky, B., Rosen, N.: Can quantum-mechanical description of physical reality be considered complete? Phys. Rev. 47, 777–780 (1935)

27. Clauser, J.F., Horne, M.A., Shimony, A., Holt, R.A.: Proposed experiment to test local hidden-variable theories. Phys. Rev. Lett. 23(15), 880–884 (1969)
28. Popescu, S., Rohrlich, D.: Quantum nonlocality as an axiom. Found. Phys. 24(3), 379–385 (1994)
29. Cirel'son, B.S.: Quantum generalizations of Bell's inequality. Lett. in Math. Phys. 4(2), 93–100 (1980)
30. Bennett, C.H., Brassard, G., Robert, J.M.: Privacy amplification by public discussion. SIAM J. on Computing 17(2), 210–229 (1988)
31. Impagliazzo, R., Levin, L.A., Luby, M.: Pseudo-random generation from one-way functions. In: STOC 1989, pp. 12–24 (1989)
32. Bennett, C.H., Brassard, G., Crépeau, C., Maurer, U.: Generalized privacy amplification. IEEE Trans. on Information Theory 41(6), 1915–1923 (1995)
33. König, R., Maurer, U., Renner, R.: On the power of quantum memory. IEEE Trans. on Information Theory 51(7), 2391–2401 (2005)
34. Renner, R., Koenig, R.: Universally composable privacy amplification against quantum adversaries. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 407–425. Springer, Heidelberg (2005)
35. Hänggi, E., Renner, R., Wolf, S.: The impossibility of non-signaling privacy amplification (2008)
36. Masanes, L.: Universally composable privacy amplification from causality constraints. Phys. Rev. Lett. 102(14), 140501 (2009)
37. Masanes, L., Renner, R., Winter, A., Barrett, J., Christandl, M.: Security of key distribution from causality constraints (2009)
38. Maurer, U.: Indistinguishability of random systems. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 110–132. Springer, Heidelberg (2002)
39. Pfitzmann, B., Waidner, M.: A model for asynchronous reactive systems and its application to secure message transmission. In: SP 2001, p. 184 (2001)
40. Backes, M., Pfitzmann, B., Waidner, M.: A composable cryptographic library with nested operations. In: CCS 2003, pp. 220–230 (2003)
41. Canetti, R.: Universally composable security: a new paradigm for cryptographic protocols. In: FOCS 2001, p. 136 (2001)
42. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge University Press, Cambridge (2004)
43. Carter, J.L., Wegman, M.N.: Universal classes of hash functions (extended abstract). In: STOC 1977, pp. 106–112 (1977)
44. Brassard, G., Salvail, L.: Secret-key reconciliation by public discussion. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 410–423. Springer, Heidelberg (1994)
45. König, R., Renner, R.: A de Finetti representation for finite symmetric quantum states. J. Math. Phys. 46(122108) (2005)

New Generic Algorithms for Hard Knapsacks

Nick Howgrave-Graham[1] and Antoine Joux[2]

[1] 35 Park St, Arlington, MA 02474
nickhg@gmail.com
[2] DGA and Université de Versailles Saint-Quentin-en-Yvelines
UVSQ PRISM, 45 avenue des États-Unis, F-78035, Versailles CEDEX, France
antoine.joux@m4x.org

Abstract. In this paper, we study the complexity of solving hard knapsack problems, i.e., knapsacks with a density close to 1 where lattice-based low density attacks are not an option. For such knapsacks, the current state-of-the-art is a 31-year old algorithm by Schroeppel and Shamir which is based on birthday paradox techniques and yields a running time of $\tilde{O}(2^{n/2})$ for knapsacks of n elements and uses $\tilde{O}(2^{n/4})$ storage. We propose here two new algorithms which improve on this bound, finally lowering the running time down to either $\tilde{O}(2^{0.385\,n})$ or $\tilde{O}(2^{0.3113\,n})$ under a reasonable heuristic. We also demonstrate the practicality of these algorithms with an implementation.

1 Introduction

The 0–1 knapsack problem or subset sum problem is a famous NP-hard problem which has often been used in the construction of cryptosystems. An instance of this problem consists of a list of n positive integers $(a_1, a_2, \cdots, a_n)$ together with another positive integer S. Given an instance, there exist two forms of knapsack problems. The first form is the decision knapsack problem, where we need to decide whether S can be written as:

$$S = \sum_{i=1}^{n} \epsilon_i a_i,$$

with values of ϵ_i in $\{0, 1\}$. The second form is the computational knapsack problem where we need to recover a solution $\epsilon = (\epsilon_1, \cdots, \epsilon_n)$ if at least one exists.

The decision knapsack problem is NP-complete (see [7]). It is also well-known that given access to an oracle that solves the decision problem, the computational problem can be solved using n calls to this oracle. Indeed, assuming that the original knapsack admits a solution, we can easily obtain the value of ϵ_n by asking to the oracle whether the subknapsack $(a_1, a_2, \cdots, a_{n-1})$ can sum to S. If so, there exists a solution with $\epsilon_n = 0$, otherwise, a solution necessarily has $\epsilon_n = 1$. Repeating this idea, we obtain the bits of ϵ one at a time.

Knapsack problems were introduced in cryptography by Merkle and Hellman [18] in 1978. The basic idea behind the Merkle-Hellman public key cryptosystem is to hide an easy knapsack instance into a hard looking one. The

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 235–256, 2010.

scheme was broken by Shamir [23] using lattice reduction. After that, many other knapsack based cryptosystems were also broken using lattice reduction. In particular, the low-density attacks introduced by Lagarias and Odlyzko [15] and improved by Coster et al. [4] are a tool of choice for breaking many knapsack based cryptosystems. The density of a knapsack is defined as:

$$d = \frac{n}{\log_2(\max_i a_i)}.$$

More recently, Impagliazzo and Naor [13] introduced cryptographic schemes which are as secure as the subset sum problem. They classify knapsack problems according to their density. On the one hand, when $d < 1$ a given sum S can usually be inverted in a unique manner and these knapsacks can be used for encryption. On the other hand, when $d > 1$, most sums have many preimages and the knapsack can be used for hashing purposes. However, for encryption, the density cannot be too low, since the Lagarias-Odlyzko low-density attack can solve random knapsack problems with density $d < 0.64$ given access to an oracle that solves the shortest vector problem (SVP) in lattices. Of course, since Ajtai showed in [1] that the SVP is NP-hard for randomized reduction, such an oracle is not available. However, in practice, low-density attacks have been shown to work very well when the SVP oracle is replaced by existing lattice reduction algorithm such as LLL[1] [16] or the BKZ algorithm of Schnorr [20]. The attack of [4] improves the low density condition to $d < 0.94$. For high density knapsacks, with $d > 1$ there is variation of these lattice-based attacks presented in [14] that finds collisions in mildly exponential time $O(2^{n/1000})$ using the same lattice reduction oracle.

However, for knapsacks with density close to 1, there is no effective lattice-based approach to solve the knapsack problem. As a consequence, in this case, we informally speak of *hard* knapsacks. Note that, it is proved in [13, Proposition 1.2], that density 1 is indeed the hardest case. For hard knapsacks, the state-of-the-art algorithm is due to Schroeppel and Shamir [21,22] and runs in time $O(n \cdot 2^{n/2})$ using $O(n \cdot 2^{n/4})$ bits of memory. This algorithm has the same running time as the basic birthday based algorithm on the knapsack problem introduced by Horowitz and Sahni [10], but much lower memory requirements. To simplify the notation of the complexities in the sequel, we extensively use the soft-Oh notation. Namely, $\tilde{O}(g(n))$ is used as a shorthand for $O(g(n) \cdot \log(g(n))^i)$, for any fixed value of i. With this notation, the algorithm of Schroeppel and Shamir runs in time $\tilde{O}(2^{n/2})$ using $\tilde{O}(2^{n/4})$ bits of memory.

Since Wagner presented his generalized birthday algorithm in [25], it is well-known that when solving problems involving sums of elements from several lists, it is possible to obtain a much faster algorithm when a single solution out of many is sought. A similar idea was previously used by Camion and Patarin in [2] to attack the knapsack based hash function of [5]. In this paper, we introduce two new algorithms that improve upon the algorithm of Schroeppel and Shamir to solve knapsack problems. In some sense, our algorithms are a new development

[1] LLL stands for Lenstra-Lenstra-Lovász and BKZ for blockwise Korkine-Zolotarev.

of the generalized birthday algorithm. The main difference is that, instead of looking for one solution among many, we look for one of the many possible representations of a given solution.

The paper is organized as follows: In Section 2 we recall some background information on knapsacks, in Section 3 we briefly recall the algorithm of Schroeppel–Shamir and introduce a useful practical variant of this algorithm, in Section 4 we present our improved algorithms and in Section 5 we describe practical implementations on a knapsack with $n = 96$. Section 4 is divided into 3 subsections, in 4.1 we describe the basic idea that underlies our algorithm, in 4.2 we present a simple algorithm based on this idea and in 4.3 we give a heuristic improvement of this algorithm in the balanced case. Finally, in Section 6 we present several extensions and some possible applications of our new algorithms.

2 Background on Knapsacks

2.1 Modular Knapsacks

We speak of a modular knapsack problem when we want to solve:

$$\sum_{i=1}^{n} \epsilon_i \, a_i \equiv S \bmod M,$$

where the integer M is the modulus.

Up to polynomial factors, solving modular knapsacks and knapsacks over the integers are equivalent. Any algorithm that realizes one task can be used to solve the other. In one direction, given a knapsack problem over the integers and an algorithm that solves any modular knapsack, it is clear that solving the problem modulo $M = \max(S, \sum_{i=1}^{n} a_i) + 1$ yields all integral solutions. In the other direction, assume that the modular knapsack $(a_1, \cdots, a_n)$ with target sum $S \bmod M$ is given by representative a_i of the classes of modular numbers in the range $[0, M-1]$. In this case, it is clear that any sum of at most n such numbers is in the range $[0, nM-1]$. As a consequence, if S is also represented in the range $[0, M-1]$, it suffices to solve n knapsack problems over the integers with targets S, $S+M$, $\ldots$, $S+(n-1)M$.

2.2 Random Knapsacks

Given two parameters n and D, we define a *random knapsack with solution* on n elements with prescribed density D as a knapsack randomly constructed using the following process:

- Let $B(n, D) = \lfloor 2^{n/D} \rfloor$.
- Choose each a_i (for i from 1 to n) uniformly at random in $[1, B(n, D)]$.
- Uniformly choose a random vector ϵ in $\{0, 1\}^n$ and let $S = \sum_{i=1}^{n} \epsilon_i \, a_i$.

Note that the computed density d of such a random knapsack differs from the prescribed density. However, as n tends to infinity, the two become arbitrarily close with overwhelming probability. In [4], it is shown that there exists a lattice based algorithm that solves all but an exponentially small fraction of random knapsacks with solution, when the prescribed density satisfies $D < 0.94$.

2.3 Unbalanced Knapsacks

The random knapsacks from above may have arbitrary values in $[0, n]$ for the weight $\sum_{i=1}^{n} \epsilon_i$ of the solution. Yet, most of the time, we expect a weight close to $n/2$. For various reasons, it is also useful to consider knapsacks with different weights. We define an *α-unbalanced random knapsack with solution* on n elements given α and the density D as follows:

- Let $B(n, D) = \lfloor 2^{n/D} \rfloor$.
- Choose each a_i (for i from 1 to n) uniformly at random in $[1, B(n, D)]$.
- Let $\ell = \lfloor \alpha n \rfloor$ and uniformly choose a random vector ϵ with exactly ℓ coordinates equal to 1, the rest being 0s, in the set of $\binom{n}{\ell}$ such vectors. Let $S = \sum_{i=1}^{n} \epsilon_i \, a_i$.

Unbalanced knapsacks are natural to consider, since they already appear in the lattice based algorithms of [15,4], where the value of α greatly impacts the densities that can be attacked. Moreover, in our algorithms, even when initially solving regular knapsacks, unbalanced knapsacks may appear in the course of the computations.

When dealing with balanced knapsacks with exactly half zeros and ones, we also use the above definition and speak of 1/2-unbalanced knapsacks.

2.4 Complementary Knapsacks

Given a knapsack $a_1, \ldots, a_n$ with target sum S, we define its *complementary knapsack* to be the knapsack that contains the same elements and has target sum $\sum_{i=1}^{n} a_i - S$. The solution ϵ of the original knapsack and ϵ' of the complementary knapsacks are related by:

$$\text{For all } i\text{:} \quad \epsilon_i + \epsilon'_i = 1.$$

Thus, solving either of the two knapsacks also yields the result of the other knapsack. Moreover, the weight ℓ and ℓ' are related by $\ell + \ell' = n$. In particular, if a knapsack is α-unbalanced, its complementary knapsack is $(1-\alpha)$-unbalanced. As a consequence, in any algorithm, we may assume without loss of generality that $\ell \leq \lfloor n/2 \rfloor$ (or that $\ell \geq \lceil n/2 \rceil$).

2.5 Asymptotic Values of Binomials

Where knapsacks are considered, binomial coefficients are frequently encountered, we recall that the binomial coefficient $\binom{n}{\ell}$ is the number of distinct choices of ℓ elements within a set of n elements. We have:

$$\binom{n}{\ell} = \frac{n!}{\ell! \cdot (n-\ell)!}.$$

We often need to obtain asymptotic approximation for binomials of the form $\binom{n}{\alpha\, n}$ (or $\binom{n}{\lfloor \alpha\, n \rfloor}$) for fixed values of α in $]0,1[$. This is easily done by using Stirling's formula:

$$n! = (1+o(1))\,\sqrt{2\pi n}\left(\frac{n}{e}\right)^n.$$

Ignoring polynomial factors in n, we find:

$$\binom{n}{\alpha\, n} = \tilde{O}\left(\left(\frac{1}{\alpha^\alpha \cdot (1-\alpha)^{1-\alpha}}\right)^n\right).$$

Many of the algorithms presented in this paper involve complexities of the form $\tilde{O}(2^{c\, n})$, where a constant c is obtained by taking the logarithm in basis 2 of numbers coming from asymptotic estimates of binomials. In this case, to improve the readability of the complexity, we choose a decimal approximation $c_0 > c$ of c. This would allow us to rewrite the complexity as $O(2^{c_0\, n})$ or even $o(2^{c_0\, n})$. However, we prefer to stick to $\tilde{O}(2^{c_0\, n})$. A typical example is the $\tilde{O}(2^{0.3113\, n})$ time complexity of our fastest algorithm, which stands for $\tilde{O}\left(\binom{n}{n/4} \cdot 2^{-n/2}\right)$.

2.6 Distribution of Random Knapsack Sums

In order to analyze the behavior of our algorithms, we need to use information about the distribution of modular sums of the form:

$$\sum_{i=1}^{n} a_i x_i \pmod{M},$$

for a random knapsack modulo M and for n-tuples $(x_1, \cdots, x_n) \in \mathcal{B}$, where $\mathcal{B}$ is an arbitrary set of n-dimensional vectors, with coordinates modulo M. We use the following important theorem [19, Theorem 3.2]:

Theorem 1. *For any set $\mathcal{B} \subset \mathbb{Z}_M^n$, the identity:*

$$\frac{1}{M^n} \sum_{(a_1,\cdots,a_n)\in\mathbb{Z}_M^n} \sum_{c\in\mathbb{Z}_M} \left(P_{a_1,\cdots,a_n}(\mathcal{B},c) - \frac{1}{M}\right)^2 = \frac{M-1}{M|\mathcal{B}|}$$

holds, where $P_{a_1,\cdots,a_n}(\mathcal{B},c)$ denotes the probability that $\sum_{i=1}^n a_i x_i \equiv c \pmod{M}$ for a random $(x_1,\cdots,x_n)$ drawn uniformly from $\mathcal{B}$, i.e.:

$$P_{a_1,\cdots,a_n}(\mathcal{B},c) = \frac{1}{|\mathcal{B}|}\left|\left\{(x_1,\cdots,x_n)\in\mathcal{B} \text{ such that } \sum_{i=1}^n a_i x_i \equiv c \pmod{M}\right\}\right|.$$

This implies the immediate corollaries:

Corollary 1. *For any real $\lambda > 0$, the fraction of n-tuples $(a_1, \cdots, a_n) \in \mathbb{Z}_M^n$ for which there exists a $c \in \mathbb{Z}_M$ that satisfies $|P_{a_1,\cdots,a_n}(\mathcal{B}, c) - 1/M| \geq \lambda/M$ is at most:*

$$\frac{M^2}{\lambda^2\, |\mathcal{B}|}.$$

Corollary 2. *For any reals $\lambda > 0$ and $1 > \mu > 0$, the fraction of n-tuples $(a_1, \cdots, a_n) \in \mathbb{Z}_M^n$ for which there exist at least $\mu\, M$ values $c \in \mathbb{Z}_M$ that satisfy $|P_{a_1,\cdots,a_n}(\mathcal{B}, c) - 1/M| \geq \lambda/M$ is at most:*

$$\frac{M}{\lambda^2\, \mu\, |\mathcal{B}|}.$$

These two corollaries are used when $|\mathcal{B}|$ is larger than M. We also need two more corollaries, one for small values of $|\mathcal{B}|$ and one for $|\mathcal{B}| \approx M$:

Corollary 3. *For any reals $1 > \mu > 0$, if $m > 1$ denotes $M/|\mathcal{B}|$, the fraction of n-tuples $(a_1, \cdots, a_n) \in \mathbb{Z}_M^n$ such that less than $\mu\, |\mathcal{B}|$ values $c \in \mathbb{Z}_M$ have $P_{a_1,\cdots,a_n}(\mathcal{B}, c) \neq 0$ is at most:*

$$\frac{\mu}{(1-\mu)m}$$

Corollary 4. *For any reals $\lambda > 0$, the fraction of n-tuples $(a_1, \cdots, a_n) \in \mathbb{Z}_M^n$ that satisfy:*

$$\sum_{c \in \mathbb{Z}_M} P_{a_1,\cdots,a_n}(\mathcal{B}, c)^2 \geq \frac{M + |\mathcal{B}|}{\lambda M |\mathcal{B}|}$$

is at most λ.

3 The Algorithm of Schroeppel and Shamir

The algorithm of Schroeppel and Shamir was introduced in [21,22]. It allows to solve a generic integer knapsack problem on n-elements in time $\tilde{O}(2^{n/2})$ using a memory of size $\tilde{O}(2^{n/4})$. It improves on the birthday algorithm of Horowitz and Sahni [10] that can be applied on such a knapsack. We first recall this basic birthday algorithm, which is based on the rewriting of a knapsack solution as an equality:

$$\sum_{i=1}^{\lfloor n/2 \rfloor} \epsilon_i\, a_i = S - \sum_{i=\lfloor n/2 \rfloor + 1}^{n} \epsilon_i\, a_i,$$

where all the ϵs are 0 or 1. Thus, to solve the knapsack problem, we construct the set $\mathcal{S}^{(1)}$ containing all possible sums of the first $\lfloor n/2 \rfloor$ elements and $\mathcal{S}^{(2)}$ be the set obtained by subtracting from the target S any of the possible sums of the last $\lceil n/2 \rceil$ elements. Searching for collisions between the two sets, we discover all the solutions of the knapsack problem. This can be done in time and memory $\tilde{O}(2^{n/2})$ by fully computing the two sets, sorting them and looking up for collisions. In [21,22], Schroeppel and Shamir show that, in order to find these collisions, it is not necessary to store the full sets $\mathcal{S}^{(1)}$ and $\mathcal{S}^{(2)}$. Instead, they generate them on the fly using priority queues (based either on heaps or a Adelson-Velsky and Landis trees), requiring memory $\tilde{O}(2^{n/4})$.

Algorithm 1. Schroeppel-Shamir algorithm

```
Require: Knapsack element $a_1, \dots, a_n$. Knapsack sum $S$
  Let $q_1 = \lfloor n/4 \rfloor$, $q_2 = \lfloor n/2 \rfloor$, $q_3 = \lfloor 3\,n/4 \rfloor$
  Create $\mathcal{S}_L^{(1)}(\sigma)$ and $\mathcal{S}_L^{(1)}(\epsilon)$: list of all $\sum_{i=1}^{q_1} \epsilon_i\, a_i$ and list of $\epsilon_{1\cdots q_1}$ (in the same order)
  Create $\mathcal{S}_R^{(1)}(\sigma)$ and $\mathcal{S}_R^{(1)}(\epsilon)$: list of all $\sum_{i=q_1+1}^{q_2} \epsilon_i\, a_i$ and list of $\epsilon_{q_1+1\cdots q_2}$
  Create $\mathcal{S}_L^{(2)}(\sigma)$ and $\mathcal{S}_L^{(2)}(\epsilon)$: list of all $\sum_{i=q_2+1}^{q_3} \epsilon_i\, a_i$ and list of $\epsilon_{q_2+1\cdots q_3}$
  Create $\mathcal{S}_R^{(2)}(\sigma)$ and $\mathcal{S}_R^{(2)}(\epsilon)$: list of all $\sum_{i=q_3+1}^{n} \epsilon_i\, a_i$ and list of $\epsilon_{q_3+1\cdots n}$
  Call 4-way merge Algorithm 2 or 3 on $(\mathcal{S}_L^{(1)}(\sigma), \mathcal{S}_R^{(1)}(\sigma), \mathcal{S}_L^{(2)}(\sigma), \mathcal{S}_R^{(2)}(\sigma))$, $n$ and $S$.
  Store returned set in Sol
  for each $(i, j, k, l)$ in Sol do
    Concatenate $\mathcal{S}_L^{(1)}(\epsilon)[i]$, $\mathcal{S}_R^{(1)}(\epsilon)[j]$, $\mathcal{S}_L^{(2)}(\epsilon)[k]$ and $\mathcal{S}_R^{(2)}(\epsilon)[l]$ into $\epsilon$
    Output: "$\epsilon$ is a solution"
  end for
```

Algorithm 2. Original 4-Way merge routine

```
Require: Four input lists $(\mathcal{S}_L^{(1)}, \mathcal{S}_R^{(1)}, \mathcal{S}_L^{(2)}, \mathcal{S}_R^{(2)})$, knapsack size $n$, target sum $\mathcal{T}$
  Let $S_L^{(1)}$, $S_R^{(1)}$, $S_L^{(2)}$ and $S_R^{(2)}$ be the sizes of the corresponding arrays.
  Create priority queues $Q_1$ and $Q_2$
  Sort $\mathcal{S}_R^{(1)}$ and $\mathcal{S}_R^{(2)}$ in increasing order. Keep track of positions in InitPos$_1$ and InitPos$_2$
  for $i$ from 0 to $S_L^{(1)}$ do
    Insert $(i, 0)$ in $Q_1$ with priority $\mathcal{S}_L^{(1)}[i] + \mathcal{S}_R^{(1)}[0]$.
  end for
  for $i$ from 0 to $S_L^{(2)}$ do
    Insert $(i, S_R^{(2)} - 1)$ in $Q_2$ with priority $\mathcal{T} - \mathcal{S}_L^{(2)}[i] - \mathcal{S}_R^{(2)}[S_R^{(2)} - 1]$.
  end for
  Create empty list Sol
  while $Q_1$ and $Q_2$ are not empty do
    Peek at value $q_1$ of lowest priority element in $Q_1$.
    Peek at value $q_2$ of lowest priority element in $Q_2$.
    if $q_1 \le q_2$ then
      Get $(i, j)$ from $Q_1$
      if $j \ne S_R^{(1)} - 1$ then
        Insert $(i, j+1)$ in $Q_1$ with priority $\mathcal{S}_L^{(1)}[i] + \mathcal{S}_R^{(1)}[j+1]$.
      end if
    end if
    if $q_1 \ge q_2$ then
      Get $(k, l)$ from $Q_2$
      if $l \ne 0$ then
        Insert $(k, l-1)$ in $Q_2$ with priority $\mathcal{T} - \mathcal{S}_L^{(2)}[k] - \mathcal{S}_R^{(2)}[l-1]$.
      end if
    end if
    if $q_1 = q_2$ then
      Add $(i, \text{InitPos}_1[j], k, \text{InitPos}_2[l])$ to Sol
    end if
  end while
  Return list of solutions Sol
```

More precisely, let us define $q_1 = \lfloor n/4 \rfloor$, $q_2 = \lfloor n/2 \rfloor$, $q_3 = \lfloor 3\,n/4 \rfloor$. We introduce four sets $\mathcal{S}_L^{(1)}$, $\mathcal{S}_R^{(1)}$, $\mathcal{S}_L^{(2)}$ and $\mathcal{S}_R^{(2)}$ of size $O(2^{n/4})$ defined as follows:

- $\mathcal{S}_L^{(1)}$ is the set of pairs $(\sum_{i=1}^{q_1} \epsilon_i\, a_i, \epsilon_{1\cdots q_1})$ with $\epsilon_{1\cdots q_1} \in \{0,1\}^{q_1}$;
- $\mathcal{S}_R^{(1)}$ is the set of $(\sum_{i=q_1+1}^{q_2} \epsilon_i\, a_i, \epsilon_{q_1+1\cdots q_2})$ with $\epsilon_{q_1+1\cdots q_2} \in \{0,1\}^{q_2-q_1}$;
- $\mathcal{S}_L^{(2)}$ is the set of $(\sum_{i=q_2+1}^{q_3} \epsilon_i\, a_i, \epsilon_{q_2+1\cdots q_3})$ with $\epsilon_{q_2+1\cdots q_3} \in \{0,1\}^{q_3-q_2}$;
- $\mathcal{S}_R^{(2)}$ is the set of $(\sum_{i=q_3+1}^{n} \epsilon_i\, a_i, \epsilon_{q_3+1\cdots n})$ with $\epsilon_{q_3+1\cdots n} \in \{0,1\}^{n-q_3}$.

With these notations, solving the knapsack problem amounts to finding four elements $\sigma_L^{(1)}$, $\sigma_R^{(1)}$, $\sigma_L^{(2)}$ and $\sigma_R^{(2)}$ in the corresponding sets such that $S = \sigma_L^{(1)} + \sigma_R^{(1)} + \sigma_L^{(2)} + \sigma_R^{(2)}$. We call this a *4-way merge problem.*

The algorithm of Schroeppel and Shamir is described in Algorithm 1, using their original 4-way merge Algorithm 2 as a subroutine. Note that, in Algorithm 1, we describe each set $\mathcal{S}_X^{(i)}$ as two lists $\mathcal{S}_X^{(i)}(\sigma)$ and $\mathcal{S}_X^{(i)}(\epsilon)$ stored in the same order.

3.1 A Variation on the Schroeppel and Shamir Algorithm

In practice, the need for priority queues of large size makes the algorithm of Schroeppel and Shamir harder to implement and to optimize. Indeed, using large priority queues either introduces an extra factor in the memory usage or unfriendly cache behavior. As a consequence, we would like to avoid priority queues altogether. In order to do this, we present a variation on their algorithm, inspired by an algorithm presented in [3] that solves the problem of finding 4 elements from 4 distinct lists with bitwise sum equal to 0. Note that, from a theoretical point of view, our variation is not as good as the original algorithm of Schroeppel and Shamir, because for some exceptional knapsacks, it requires more memory.

The idea is to choose a modulus M near $2^{(1/4-\varepsilon)\,n}$ and to remark that the 4-way merge condition implies $\sigma_L^{(1)} + \sigma_R^{(1)} \equiv S - \sigma_L^{(2)} - \sigma_R^{(2)} \pmod{M}$. As a consequence, for any solution of the knapsack, there exists a value σ_M, such that:

$$\sigma_M = (\sigma_L^{(1)} + \sigma_R^{(1)}) \bmod M = (S - \sigma_L^{(2)} - \sigma_R^{(2)}) \bmod M.$$

Since, we cannot guess the correct value of σ_M, we simply loop over all possible values. This gives a new 4-way merge Algorithm 3, which can be used as a replacement for the original subroutine in Algorithm 1.

Informally, for each test value of σ_M, Algorithm 3 constructs the set of all sums $\sigma_L^{(1)} + \sigma_R^{(1)}$ congruent to σ_M modulo M. This is done by sorting $\mathcal{S}_R^{(1)}$ by values modulo M. Indeed, in this case, it suffices for each $\sigma_L^{(1)}$ in $\mathcal{S}_L^{(1)}$ to search the value $\sigma_M - \sigma_L^{(1)}$ in $\mathcal{S}_R^{(1)}$. Using this method, we construct the set $\mathcal{S}^{(1)}$ of the birthday paradox algorithm as a disjoint union of smaller sets $\mathcal{S}^{(1)}(\sigma_M)$, which are created one at a time within the loop on σ_M in Algorithm 2. Similarly, we implicitly construct $\mathcal{S}^{(2)}$ as a disjoint union of $\mathcal{S}^{(2)}(\sigma_M)$, but do not store it, instead searching for matching values in $\mathcal{S}^{(1)}(\sigma_M)$.

Algorithm 3. Modular 4-Way merge routine

```
Require: Four input lists $(\mathcal{S}_L^{(1)}, \mathcal{S}_R^{(1)}, \mathcal{S}_L^{(2)}, \mathcal{S}_R^{(2)})$, size $n$, target sum $\mathcal{T}$
Require: Memory margin parameter: $\varepsilon$
  Let $M$ be a random modulus in $[2^{(1/4-\varepsilon)\,n}, 2 \cdot 2^{(1/4-\varepsilon)\,n}]$
  Create list $\mathcal{S}_R^{(1)}(M)$ containing pairs $(\mathcal{S}_R^{(1)}[i] \bmod M, i)$ where $i$ indexes all of $\mathcal{S}_R^{(1)}$
  Create list $\mathcal{S}_R^{(2)}(M)$ containing pairs $(\mathcal{S}_R^{(2)}[i] \bmod M, i)$ where $i$ indexes all of $\mathcal{S}_R^{(2)}$
  Sort $\mathcal{S}_R^{(1)}(M)$ and $\mathcal{S}_R^{(2)}(M)$ by values of the left member of each pair
  Create empty list $Sol$
  for $\sigma_M$ from 0 to $M-1$ do
    Empty the list $\mathcal{S}^{(1)}$ (or create the list if $\sigma_M = 0$)
    for $i$ from 1 to size of $\mathcal{S}_L^{(1)}$ do
      Let $\sigma_L^{(1)} = \mathcal{S}_L^{(1)}[i]$ and $\sigma_t = (\sigma_M - \sigma_L^{(1)}) \bmod M$
      Binary search first occurrence of $\sigma_t$ in $\mathcal{S}_R^{(1)}(M)$
      for each consecutive $(\sigma_t, j)$ in $\mathcal{S}_R^{(1)}(M)$ do
        Add $(\sigma_L^{(1)} + \mathcal{S}_R^{(1)}[j]), (i, j))$ to $\mathcal{S}^{(1)}$
      end for
    end for
    Sort list $\mathcal{S}^{(1)}$ by values of the left member of each pair
    for $k$ from 1 to size of $\mathcal{S}_L^{(2)}$ do
      Let $\sigma_L^{(2)} = \mathcal{S}_L^{(2)}[k]$ and $\sigma_t = (\mathcal{T} - \sigma_M - \sigma_L^{(2)}) \bmod M$
      Binary search first occurrence of $\sigma_t$ in $\mathcal{S}_R^{(2)}$
      for each consecutive $(\sigma_t, l)$ in $\mathcal{S}_R^{(2)}(M)$ do
        Let $\mathcal{T}' = \mathcal{T} - \sigma_L^{(1)} - \mathcal{S}_R^{(2)}[l]$
        Binary search first occurrence of $\mathcal{T}'$ in $\mathcal{S}^{(1)}$
        for each consecutive $(T, (i, j))$ in $\mathcal{S}^{(1)}$ do
          Add $(i, j, k, l)$ to $Sol$
        end for
      end for
    end for
  end for
  Return list of solutions $Sol$
```

Algorithm 4. Our simple algorithm (Section 4.2)

```
Require: Knapsack elements $a_1, \ldots, a_n$. Knapsack sum $S$. Parameter $\beta$
  Let $M$ be a random prime close to $2^{\beta\, n}$
  Let $R_1$, $R_2$ and $R_3$ be random values modulo $M$.
  Solve the 1/8-unbalanced knapsack modulo $M$ with elements $a$ and target $R_1$.
  Solve the 1/8-unbalanced modular knapsack with target $R_2$.
  Solve the 1/8-unbalanced modular knapsack with target $R_3$.
  Solve the 1/8-unbalanced modular knapsack with target $S - R_1 - R_2 - R_3 \bmod M$.
  Create the 4 sets of non-modular sums corresponding to the above solutions.
  Do a 4-way merge (with early abort and consistency checks) on these 4 sets.
  Rewrite the obtained solution as a knapsack solution.
```

Complexity analysis. If we ignore the innermost loop that writes down the solution set Sol, the running time of the execution of the loop iteration corresponding to σ_M is $\tilde{O}(\mathcal{S}^{(1)}(\sigma_M) + \mathcal{S}^{(2)}(\sigma_M))$, with a polynomial factor in n that comes from sorting and searching. Summing over all iterations of the loop, we have a total running time of $\tilde{O}(\mathcal{S}^{(1)} + \mathcal{S}^{(2)}) = \tilde{O}(2^{n/2})$, unless Sol has a size larger than $\tilde{O}(2^{n/2})$.

Where memory is concerned, storing $\mathcal{S}_L^{(1)}$, $\mathcal{S}_R^{(1)}$, $\mathcal{S}_L^{(2)}$ and $\mathcal{S}_R^{(2)}$ costs $O(2^{n/4})$. However, the memory required to store the partitioned representation of $\mathcal{S}^{(1)}$ is $\max_{\sigma_M} \mathcal{S}^{(1)}(\sigma_M)$. Note that we cannot guarantee that this maximum remains small. A simple counterexample occurs when all a_i values (in the first half) are multiples of M. Indeed, in that case we find that $\mathcal{S}^{(1)}(0)$ has size $2^{n/2}$. In general, we do not expect such a bad behavior, more precisely, we have:

Theorem 2. *For any real $\varepsilon > 0$ and modulus M close to $2^{(1/4-\varepsilon)\,n}$, for a fraction at least $1 - 2^{-4\varepsilon\, n}$ of knapsacks with density $D < 4$ given by n-tuples $(a_1, \cdots, a_n)$ and target value T, Algorithm 1 using as 4-way merge routine Algorithm 3 finds all of the N_{Sol} solutions of the knapsack in time $\tilde{O}(\max(2^{n/2}, N_{Sol}))$ using memory $\tilde{O}(\max(2^{(1/4+\varepsilon)\,n}, N_{Sol}))$.*

Proof. The time analysis is given above. The bound on the memory use for almost all knapsacks comes from applying Corollary 1 with $\lambda = 1/2$ twice on the left and right-hand side subknapsacks on $n/2$ elements, using $\mathcal{B} = \{0,1\}^{n/2}$. We need to use the fact that a random knapsack taken uniformly at random with n/D-bit numbers is close to a random knapsack modulo M, when $D < 4$.

A high bit version. Instead of using modular values to partition the sets $\mathcal{S}^{(1)}$ and $\mathcal{S}^{(2)}$, another option is to look at the value of the $\lceil n/4 \rceil$ higher bits. Depending on the precise context, this option might be more practical than the modular version. In the implementation presented in Section 5, we make use of both versions.

Early abort with multiple solutions. When the number of solutions N_{Sol} is large, and we wish to find a single solution, we can use an early abort strategy. Heuristically , assuming that the σ_M values corresponding to the many solutions are well-distributed modulo M, this reduces the heuristic expected running time to $\tilde{O}(\max(2^{n/4}, 2^{n/2}/N_{Sol}))$.

3.2 Application to Unbalanced Knapsacks

The basic birthday algorithm, the algorithm of Schroeppel–Shamir and our variation can also, with some care, be applied to α-unbalanced knapsacks. In this case, if we let:

$$\mathcal{C}_\alpha = \left(\alpha^{-\alpha} \cdot (1-\alpha)^{\alpha-1}\right),$$

the time complexity is $\tilde{O}(\mathcal{C}_\alpha^{n/2})$ and the memory complexity is $\tilde{O}(\mathcal{C}_\alpha^{n/2})$ for the basic birthday algorithm, $\tilde{O}(\mathcal{C}_\alpha^{n/4})$ for the algorithm of Schroeppel and Shamir and $\tilde{O}(\mathcal{C}_\alpha^{(1/4+\varepsilon)\,n})$ for our variation.

Adapting to the unbalanced case. Letting $\ell = \lfloor \alpha n \rfloor$, if we assume that the solution of the knapsack has $\lfloor \ell/2 \rfloor$ elements coming from the first half, then the algorithms are easily adapted. With the basic birthday method, the only difference with the balanced case is that $\mathcal{S}^{(1)}$ now contains all sums of exactly $\lfloor \ell/2 \rfloor$ elements among the $\lfloor n/2 \rfloor$ first elements and $\mathcal{S}^{(2)}$ contains all sums of $\lceil \ell/2 \rceil$ among the last $\lceil n/2 \rceil$ elements. This restriction is important, because allowing more elements on either side makes the sets $\mathcal{S}^{(1)}$ or $\mathcal{S}^{(2)}$ too large and prevents us from reaching the expected complexity bound. With balanced knapsacks, this is not an issue because $\binom{n}{\lfloor n/2 \rfloor}$ and 2^n are within polynomial factors of each other.

However, nothing *a priori* guarantees that the solution satisfies the above assumption. If it does, we say, following [24], that we have a splitting family. When n is even, to obtain such a splitting family, we can use a method attributed to Coppersmith in [24]. The idea is to run the algorithm n times on n knapsacks whose target sums are all equal to S and whose elements are rotated copies of $a_1, \ldots, a_n$. Namely, the elements of the i-th knapsack are $a_j^{(i)} = a_{(i+j) \bmod n}$. To prove that this works, it suffices to show that a sliding windows of $n/2$ consecutive elements intersects the solution S in exactly $\lfloor \ell/2 \rfloor$ points at least once, see [24] for details. When n is odd, we instead attempt to solve the two knapsacks on $n-1$ elements a_1 to a_{n-1} and targets S and $S - a_n$, thus going back to the even case. Alternatively, it is also possible to use a randomized approach also due to Coppersmith and described in [24]. In fact, it suffices to randomize the order of the a_i for each new trial and take the first and second halves. Thanks to Stirling's formulae, this, on average, only requires $O(\sqrt{n})$ trials.

For applying the algorithm of Schroeppel–Shamir or our variation to unbalanced knapsacks, we need to assume that the number of elements in each of the four quarters is known in advance and is either equal to $\lfloor \ell/4 \rfloor$ or to $\lceil \ell/4 \rceil$. Assuming that n is a multiple of 4, this can be achieved in a deterministic way by first using a sliding windows to guarantee that the two halves contains $\lfloor \ell/2 \rfloor$ or to $\lceil \ell/2 \rceil$ elements, then, inside of each half, we use another sliding window to balance the number of elements within the corresponding quarter. At most, we need to try $n^3/4$ configurations. When n is not a multiple of 4, we first guess the value of ϵ in $(n \bmod 4)$ positions and we are back to a knapsack with a number of elements equal to a multiple of 4. It is also possible to use a randomized approach, with an expected number of trials $O(n^{3/2})$.

4 The New Algorithms

4.1 Basic Principle

In this section, we want to solve a generic knapsack problem on n-elements. We start from the basic knapsack equation:

$$S = \sum_{i=1}^{n} \epsilon_i a_i.$$

As explained in Section 2, by taking the complementary knapsack if required, we may assume that $\ell = \sum_{i=1}^{n} \epsilon_i \geq \lceil n/2 \rceil$.

We define the set $\mathcal{S}_{\lceil \ell/2 \rceil}$ as the set of all partial sums of $\lfloor \ell/2 \rfloor$ or $\lceil \ell/2 \rceil$ knapsack elements. Clearly, there exists pairs (σ_1, σ_2) of elements of $\mathcal{S}_{\lceil \ell/2 \rceil}$ such that $S = \sigma_1 + \sigma_2$. In fact, there exist many such pairs, corresponding to all the possible decompositions of the set of ℓ elements appearing in S into two subsets of size $\leq \lceil \ell/2 \rceil$. The number $\mathcal{N}_n$ of such decompositions is given either by the binomial $\binom{\ell}{\ell/2}$ for even ℓ or by $2\binom{\ell}{(\ell-1)/2}$ for odd ℓ.

The basic idea that underlies all algorithms presented in this paper is to focus on a small part on $\mathcal{S}_{\lceil \ell/2 \rceil}$, in order to discover one of these many solutions. We start by choosing a prime integer M near $\mathcal{N}_n$ and a random element R modulo M. Heuristically, we find that with some constant probability, there exists a decomposition of S into $\sigma_1 + \sigma_2$, such that $\sigma_1 \equiv R \pmod{M}$ and $\sigma_2 \equiv S - R \pmod{M}$. To find such a decomposition, it suffices to construct the two subsets of $\mathcal{S}_{\lceil \ell/2 \rceil}$ containing elements respectively congruent to R and $S-R$ modulo M. Using the asymptotic estimates of binomials, we find that the expected size of each of these subsets is:

$$\frac{\binom{n}{\lceil \ell/2 \rceil}}{M} \approx \frac{\binom{n}{\lceil \ell/2 \rceil}}{\binom{\ell}{\lceil \ell/2 \rceil}} = \tilde{O}(2^{0.3113\,n}).$$

The exponent 0.3113 is obtained by approximating the binomial in the worst case where $\ell \approx n/2$. Once these two subsets, respectively denoted by $\mathcal{S}^{(1)}_{\lceil \ell/2 \rceil}$ and $\mathcal{S}^{(2)}_{\lceil \ell/2 \rceil}$ are constructed, we need to find a collision between σ_1 and $S - \sigma_2$, with σ_1 in $\mathcal{S}^{(1)}_{\lceil \ell/2 \rceil}$ and σ_2 in $\mathcal{S}^{(2)}_{\lceil \ell/2 \rceil}$. Clearly, using a classical sort and match method, this can be done in time $\tilde{O}(2^{0.3113\,n})$. As a consequence, assuming that we can construct the sets $\mathcal{S}^{(1)}_{\lceil \ell/2 \rceil}$ and $\mathcal{S}^{(2)}_{\lceil \ell/2 \rceil}$ quickly enough ,we can hope to construct an algorithm with overall complexity $\tilde{O}(2^{0.3113\,n})$ for solving generic knapsacks,. The rest of this paper shows how this can be achieved and also tries to minimize the required amount of memory.

Application to unbalanced knapsacks. The above idea can directly be applied to unbalanced knapsacks with $\ell = \alpha n$ elements in the decomposition of S. This expected size of the subsets of $\mathcal{S}_{\lceil \ell/2 \rceil}$ can now be approximated by:

$$\frac{\binom{n}{\lceil \ell/2 \rceil}}{\binom{\ell}{\lceil \ell/2 \rceil}} = \tilde{O}\left(\left(\frac{2}{\alpha^{\alpha/2} \cdot (2-\alpha)^{(2-\alpha)/2}}\right)^n \cdot 2^{-\alpha n}\right).$$

Interestingly, when $\alpha < 1/2$ we obtain a smaller bound by considering the complementary knapsack. As a consequence, in order to preserve the usual convention $\alpha \leq 1/2$, it is useful to substitute α by $1-\alpha$, we obtain the bound:

$$\tilde{O}\left(\left((1-\alpha)^{(\alpha-1)/2} \cdot (1+\alpha)^{-(1+\alpha)/2}\right)^n \cdot 2^{\alpha n}\right).$$

The curve of the logarithm in base 2 of this bound is included in Figure 1.

4.2 Simple Algorithm

We first present a reasonably simple algorithm, which can achieve several trade-offs between time and memory. For simplicity, we assume that $\ell = \sum_{i=1}^{n} \epsilon_i = \lfloor n/2 \rfloor$. Should this not be the case, it would suffice to run the algorithm (possibly in the unbalanced version described below) for all values of $\ell \leq \lfloor n/2 \rfloor$. In such a sequence of executions, the instance with $\ell = \lfloor n/2 \rfloor$ dominates the running time and the total run time remains within the same bound.

In our simple algorithm, instead of considering decompositions of S into two sub-sums as in the previous section, we now consider decompositions into four parts and write:

$$S = \sigma_1 + \sigma_2 + \sigma_3 + \sigma_4,$$

where each σ_i belongs to the set $\mathcal{S}_{\lceil \ell/4 \rceil}$ of all partial sums of either $\lfloor \ell/4 \rfloor$ or $\lceil \ell/4 \rceil$ knapsack elements. The exact number $\mathcal{N}$ of such decompositions varies depending on the value of ℓ modulo 4, for example:

$$\mathcal{N} = \binom{\ell}{\ell/4, \ell/4, \ell/4, \ell/4} \quad \text{when } \ell \equiv 0 \pmod 4.$$

However, in any case, thanks to Stirling's formula, we find that $\mathcal{N} = \tilde{O}(2^n)$.

We now choose an integer M near $2^{\beta n}$ (with $1/4 < \beta < 1/3$) and three random elements R_1, R_2 and R_3 modulo M. We then search for a decomposition that satisfies the constraints $\sigma_1 \equiv R_1 \pmod M$, $\sigma_2 \equiv R_2 \pmod M$, $\sigma_3 \equiv R_3 \pmod M$ and $\sigma_4 \equiv S - R_1 - R_2 - R_3 \pmod M$. Clearly, the fourth condition is a consequence of the other three and we heuristically expect $\mathcal{N} M^{-3}$ solutions that satisfy the extra constraints. To make this heuristic expectation precise enough we need the following generalization to Corollary 2:

Corollary 5. *When* $\log_2(M) > (3 \log_2(3)/16)\, n \approx 0.2972\, n$*, for any reals* $\lambda > 0$ *and* $1 > \mu > 0$*, the fraction of n-tuples* $(a_1, \cdots, a_n) \in \mathbb{Z}_M^n$ *for which there exist at least* μM^3 *values* $(c_1, c_2, c_3) \in \mathbb{Z}_M$ *that satisfy* $|P_{a_1,\cdots,a_n}(\mathcal{B}, c_1, c_2, c_3) - 1/M^3| \geq \lambda/M^3$ *is at most:*

$$\frac{2M^3}{\lambda^2\, \mu\, |\mathcal{B}|},$$

where $\mathcal{B}$ *is the set of decomposition of a given solution as* $(x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)})$ *and* $P_{a_1,\cdots,a_n}(\mathcal{B}, c_1, c_2, c_3)$ *denotes the probability of the event:*

$$\sum_{i=1}^{n} a_i x_i^{(1)} \equiv c_1 \quad \textit{and} \quad \sum_{i=1}^{n} a_i x_i^{(2)} \equiv c_2 \quad \textit{and} \quad \sum_{i=1}^{n} a_i x_i^{(3)} \equiv c_3 \pmod M.$$

Proof. Refer to long version of this paper [12].

Heuristically, we also expect the corollary to hold as long as $\beta > 1/4$.

Once we have random values R_1, R_2 and R_3 that match a decomposition of the solution, we can find the solution as follows: we start by constructing the four subsets of $\mathcal{S}_{\lceil \ell/4 \rceil}$ containing elements respectively congruent to R_1, R_2, R_3

and $S - R_1 - R_2 - R_3$ modulo M. We denote these subsets by $\mathcal{S}^{(1)}_{\lceil \ell/4 \rceil}$, $\mathcal{S}^{(2)}_{\lceil \ell/4 \rceil}$, $\mathcal{S}^{(3)}_{\lceil \ell/4 \rceil}$ and $\mathcal{S}^{(4)}_{\lceil \ell/4 \rceil}$. Once this is done, we search for a knapsack solution by doing a 4-way merge of these sets. This strategy is outlined as Algorithm 4.

Constructing the subsets. To construct each of the subsets $\mathcal{S}^{(1)}_{\lceil \ell/4 \rceil}$, $\mathcal{S}^{(2)}_{\lceil \ell/4 \rceil}$, $\mathcal{S}^{(3)}_{\lceil \ell/4 \rceil}$ and $\mathcal{S}^{(4)}_{\lceil \ell/4 \rceil}$, we use the algorithm of Schroeppel and Shamir. Note that, since the solution we are searching is a sum of $\lfloor \ell/4 \rfloor$ or $\lceil \ell/4 \rceil$ elements, we need to use the algorithm in the unbalanced case, with $\alpha = 1/8$. Depending on the value of β, the set of solutions may be quite large. Indeed, the expected number of solutions is $\binom{n}{\lceil n/8 \rceil} \cdot 2^{-\beta n} = \tilde{O}(2^{(0.5436-\beta)\,n})$. Since this is bigger than the size of the subsets $\mathcal{S}^{(i)}_{\lceil \ell/4 \rceil}$, the memory complexity of Algorithm 1 is $\tilde{O}(2^{(0.5436-\beta)\,n})$, while its time complexity is $\tilde{O}(\max(2^{(0.5436-\beta)\,n}, 2^{0.272\,n}))$. For the theoretical analysis, we assume here that we are using the original 4-way merge algorithm of Schroeppel and Shamir whose complexity is always guaranteed.

Of course, since we are solving modular knapsack instances, we first need to transform the problems into (polynomially many instances of) integer knapsacks as explained in Section 2. In any case, note that the time and memory requirements of this stage are dominated by the complexity of the next stage.

Recovering the desired solution. Once the subsets $\mathcal{S}^{(1)}_{\lceil \ell/4 \rceil}$, $\mathcal{S}^{(2)}_{\lceil \ell/4 \rceil}$, $\mathcal{S}^{(3)}_{\lceil \ell/4 \rceil}$ and $\mathcal{S}^{(4)}_{\lceil \ell/4 \rceil}$ are constructed, it suffices to perform a 4-way merge of these sets using a slightly modified version of the modular[2] 4-way merge Algorithm 3. For this 4-way merge, we use a modulus M' coprime to M. We choose M' close to $\binom{n}{\lceil \ell/4 \rceil} 2^{-\beta n} \approx 2^{(0.5436-\beta)n}$.

The changes to Algorithm 3 are the following:

1. Rename the modulus as M'
2. Replace the "for" loop on the $\sigma_{M'}$ value, by a loop where each new value of $\sigma_{M'}$ is randomly selected.
3. At each merge, i.e. insertion in $\mathcal{S}^{(1)}$, *Sol* or (implicit) $\mathcal{S}^{(2)}$, add a consistency check to make sure that the corresponding subset sums do not overlap. If consistency check fails, skip the insertion.
4. Add an early abort criteria: stop the algorithm at the first insertion in *Sol*.

At the end of the algorithm, the consistent solution $\sigma_1 + \sigma_2 + \sigma_3 + \sigma_4 = S$ present in *Sol* can be translated into a solution of the knapsack problem.

Complexity analysis (sketch of proof). We have already seen that the time complexity of the subset construction phase is $\tilde{O}(\max(2^{(0.5436-\beta)n}, 2^{0.272\,n}))$ using memory $\tilde{O}(2^{(0.5436-\beta)\,n})$. To analyze the complexity of the recovery stage, we need to know the size of the intermediate set of sums $\mathcal{S}^1(\sigma_{M'})$. Note that

[2] Here, we cannot use the original 4-way merge, because we do not know how to analyze its complexity when early abort is used.

this set contains all choices of $\lceil \ell/2 \rceil$ elements among n that can be written as a sum $\sigma = \sigma_1 + \sigma_2$ satisfying a modular constraints, i.e., $\sigma_1 \equiv \sigma_{M'} \pmod{M'}$. By construction, we also have $\sigma_1 \equiv R_1 + R_2 \pmod{M}$.

Using the same techniques, we can also show that there exists a constant τ such that at least $\tau \min(2^{(1-3\beta)\,n}, 2^{(1/2-\beta)\,n})$ decompositions of the original solutions in two parts with $\sigma_1 \equiv R_1 + R_2 \pmod{M}$ are obtained. Let $\mathcal{B}$ denotes this set of accessible decompositions and look at the corresponding sums modulo $M' > |\mathcal{B}|$. Applying Corollary 3 with $\mu = 1/2$, we find that, for all but an exponentially small fraction of n-tuples $(a_1, \cdots, a_n)$, at least $|\mathcal{B}|/2$ different sums modulo M'. As a consequence, since the $\sigma_{M'}$ values are taken at random, the 4-way merge requires an expected number of iterations $M'/(2|\mathcal{B}|)$. Moreover, after $n\,M'/(2|\mathcal{B}|)$ iterations there is an overwhelming probability to find at least one such decomposition. Thus, the early abort occurs after $\tilde{O}(M'/(2|\mathcal{B}|))$ iterations.

It remains to analyze the time complexity of each iteration of the loop. It is dominated by the number of merged pairs that need to be tested for consistency. For any value of $\sigma_{M'}$ the number of pairs is the sum over c of the number of elements congruent to c modulo M' in the first list by the number of elements congruent to $\sigma_{M'} - c$ modulo M' in the second list. This is a scalar product of two vectors on M' elements. It is smaller than the product of the norms of the two vectors. We can bound the squared norm using Corollary 4, with $\lambda = 2^{-\varepsilon n}$. We find that for an exponentially small fraction λ of n-tuples, the number of pairs tested for consistency per iteration is $\tilde{O}(2^{\varepsilon n} M')$. Multiplying by the number of iterations, we find a total time $\tilde{O}(2^{\varepsilon n} {M'}^2/|\mathcal{B}|) = \tilde{O}(2^{(0.0872+\beta)\,n})$ when ε is small enough.

We should also state that the number of quadruples tested for consistency is $\tilde{O}(2^{0.3113\,n})$. Putting everything together, when $1/3 > \beta > 1/4$, we summarize the overall running time of the algorithm as $\tilde{O}(\max(2^{0.3113\,n}, 2^{(0.0872+\beta)n}))$ using $\tilde{O}(2^{(0.5436-\beta)n})$ units of memory. We recall that, when $\beta \leq 3\log_2(3)/16$ the analysis is only heuristic.

Some possible time-memory trade-offs. We now instantiate this simple algorithm by choosing values for β. A first option is to minimize the required amount of memory, this is achieved by taking β arbitrarily close to 1/3 and yields a running time $\tilde{O}(2^{0.421\,n})$, using $\tilde{O}(2^{0.211\,n})$ memory units. A second option is to look at the smallest value of β for which we can prove the algorithm, i.e., $\beta \approx 0.2972$, we have running time $\tilde{O}(2^{0.385\,n})$, using $\tilde{O}(2^{0.247\,n})$ memory units. A third heuristic option is to require the same amount of memory as in Schroeppel–Shamir, i.e. $\tilde{O}(2^{n/4})$, this occurs for $\beta \approx 0.2936$ and corresponds to a running time $\tilde{O}(2^{0.381\,n})$. Finally, we can minimize the running time by taking β close to 1/4 and obtain an algorithm with time complexity $\tilde{O}(2^{0.338\,n})$ and memory complexity $\tilde{O}(2^{0.294\,n})$.

For the choices $\beta < 1/4$, the time complexity becomes $\tilde{O}(2^{(0.5872-\beta)n})$ and increases again.

Complexity for unbalanced knapsacks. As in Section 3.1, this algorithm can be extended to α-unbalanced knapsacks, with $\alpha \leq 1/2$. Writing the time complexity as $\tilde{O}(2^{\mathcal{C}_\alpha n})$ and the memory complexity as $\tilde{O}(2^{\mathcal{D}_\alpha n})$, we have:

$$\begin{aligned}\mathcal{C}_\alpha &= 2\log_2\left(\frac{4}{\alpha^{\alpha/4}\cdot(4-\alpha)^{(4-\alpha)/4}}\right) - 2\alpha + 2\beta\alpha \quad \text{and} \\ \mathcal{D}_\alpha &= \ \ \log_2\left(\frac{4}{\alpha^{\alpha/4}\cdot(4-\alpha)^{(4-\alpha)/4}}\right) - 2\beta\alpha.\end{aligned}$$

As in the balanced case, the parameter β determines the chosen time-memory trade-off.

Knapsacks with multiple solutions. Note that nothing prevents the above algorithm from finding a large fraction of the solutions for knapsacks with many solutions. However, in that case, we need to take some additional precautions. We need to change the early abort strategy and to remove any duplicate representation of a given solution. We should remember that, if the number of solutions becomes too large it can dominate time and memory complexities.

For an application that would require all the solutions of the knapsack, it is also necessary to increase the running time. The reason is that this algorithm is probabilistic and that the probability of missing any given solution decreases exponentially as a function of the running time. Of course, when there is a large number N_{Sol} of solutions, the probability of missing at least one is multiplied by N_{Sol}. Heuristically, to balance this, we increase the running time by a factor of $\log(N_{Sol})$.

4.3 A Better Heuristic Algorithm

Despite the fact that the algorithm from Section 4.2 outperforms the method of Schroeppel and Shamir, it does not achieve the complexity expected from Section 4.1. Admittedly, with the choice of β that optimizes speed, it comes reasonably close. However, in this case, it requires more memory than we would expect. In order to further reduce the complexity, we propose a heuristic algorithm that more closely follows the basic idea from Section 4.1. More precisely, we need to write $S = \sigma_1 + \sigma_2$ and constrain σ_1 enough to lower the number of expected solutions close to 1. Once again, we assume, for simplicity, that n is even and that we are considering a 1/2-unbalanced knapsack.

We choose a modulus M close to $2^{\gamma n}$ with $\gamma \geq 1/2$ and thus need to consider on average $2^{(\gamma-1/2)n}$ different random values for σ_1 modulo M. For each of these $2^{(\gamma-1/2)n}$ random values, denoted by R, we need to compute the list of all solutions to the partial knapsack $\sigma_1 = R \pmod{M}$, the list of all solutions to $\sigma_2 = S - R \pmod{M}$ and finally to search for a collision between the integer values of σ_1 and $S - \sigma_2$.

Clearly, the list of values σ_1 (or σ_2) can be constructed by solving a modular knapsack problem involving about $n/4$ values chosen among n. After transforming the problem into integer knapsack problems, we simply use the algorithm from Section 4.2 in the 1/4-unbalanced case. This can be done using time $\tilde{O}(2^{0.2996\,n})$ and memory $\tilde{O}(2^{0.2123\,n})$, assuming that the number of solutions is no bigger than that. Since the number of expected solutions is $\binom{n}{n/4}/M = \tilde{O}(2^{(0.8113-\gamma)n})$, we choose γ between 0.5117 and 0.5990, in order to balance the number of solutions returned by the subroutine with some compromise between its time or memory. Here is a table that shows some achievable trade-offs:

γ	Time exponent	Memory exponent	Comment
0.5117	0.3113	0.2996	Lowest time
0.5177	0.3173	0.2936	Same memory as Algorithm 4
0.5375	0.3372	0.2737	Same time as Algorithm 4
0.5613	0.3609	1/4	Same memory as Schroeppel-Shamir
0.5990	0.3986	0.2123	Lowest memory

The behavior of this algorithm for α-unbalanced knapsacks is shown on Figure 1.

Complexity using recursion. Finally, we can use this heuristic approach recursively to slightly reduce the memory requirements. In fact, one level of recursion is enough. To solve an α-unbalanced knapsack, we cut it in two halves and solve the resulting $(\alpha/2)$-unbalanced knapsacks using the above heuristic method. Thanks to the faster runtime of the subroutine, we can use a different choice for γ and obtain the lowest runtime with less memory. More precisely, the memory use for α-unbalanced knapsacks is now equal to the running time of the heuristic algorithm on $(\alpha/2)$-unbalanced knapsacks. As a consequence, we can solve 1/2-unbalanced knapsacks in time $\tilde{O}(2^{0.3113\,n})$ using $\tilde{O}(2^{0.2936\,n})$ units of memory.

5 A Practical Experiment

In order to make sure that our new algorithms perform well in practice, we have benchmarked their performance by using a typical hard knapsack problem. We constructed it using 96 random elements of 96 bits each and then built the target S as the sum of 48 of these elements.

Variation of Schroeppel–Shamir algorithm. Concerning the implementation of Schroeppel–Shamir algorithm, we need to distinguish between two cases. Either we are given a good decomposition of the set of indices into four quarters, each containing half zeroes and half ones, or we are not. In the first case, we can reduce the size of the initial small lists to $\binom{24}{12} = 2\,704\,156$. In second case, two options are possible: we can run the previous approach on randomized decompositions until a good is found, which requires about 120 executions; or we can start from small lists of size $2^{24} = 16\,777\,216$.

For the first case, we used the variation of Schroeppel–Shamir presented in Section 3.1 with a prime modulus $M = 2\,704\,157$. Testing the full space of solutions requires $120 \times 37 = 4\,400$ days on a single Intel core 2 duo at 2.66 Ghz. It turns out that, despite higher memory requirements, the second option is the faster one and would require about 1 500 days on the same machine to enumerate the search space. The memory requirements are either 300 Mbytes of memory with initial lists of size $\binom{24}{12}$ and 1.8 Gbytes with initial lists of size 2^{24}.

Of course, the algorithm may succeed before finishing the full enumeration.

Our simple algorithm. As with the Schroeppel–Shamir algorithm, we need to distinguish between two cases, depending whether or not a good decomposition into four balanced quarters is initially known. When it is the case, our implementation recovers the correct solution in less than an hour on the same computer. When such a decomposition is not known in advance, we need about 120 randomized decompositions and find a solution in about 5 days. The parameters we use in the implementation are the following:

- For the main modulus that define the random values R_1, R_2 and R_3, we take $M = 1\,253\,839$.
- For the final merging of the four obtained lists, we use the modulus 2 493 709 and apply consistency checks and early abort.

The memory requirement are approximately 2.6 Gbytes of memory.

Our better heuristic algorithm. In our implementation of the algorithm, the small lists that occur at the innermost level are so small that we replaced Schroeppel–Shamir algorithm there by a basic birthday paradox method. Thus, we no longer need a decomposition of the knapsack into four balanced quarters. Instead, two balanced halves are enough. This means that when such a decomposition is not given, we only need to run the algorithm an average of 6.2 times to find a correct decomposition instead of 120 times. The parameters we use are:

- For the higher level modulus, we choose $M = 4\,194\,319 \cdot 58\,711 \cdot 613$.
- The innermost birthday paradox method is done modulo 613.
- Assembling the two half-knapsacks is performed modulo $58\,711 \cdot 613$.

In practice, using such composite moduli saves time and memory. With the above parameters, our implementation uses about 1.7 Gbytes and runs in approximately 1.5 hours given a correct decomposition[3] into two halves. Without such a decomposition, we need less than 10 hours to find a solution.

6 Possible Extensions and Applications

The algorithmic techniques presented in this paper can be applied to more than ordinary knapsacks. We already mentioned modular knapsacks in Section 2, we now describe a few more:

Approximate knapsack problems. A first problem we can consider is to find approximate solutions to knapsack problems. More precisely, given a knapsack $a_1, \ldots, a_n$ and a target S, we try to write:

$$S = \sum_{i=1}^{n} \epsilon_i \, a_i + \delta,$$

[3] More precisely, we found 30 copies of the solution in 157 585 seconds on a single core.

where δ is small, i.e. belongs to the range $[-B, B]$ for a specified bound B. As the modular knapsack problem, this can be solved by transforming it into a knapsack problem with several targets. Define a new knapsack $b_1, \ldots, b_n$ where b_i is the closest integer to a_i/B and let S' be the closest integer to S/B. To solve the original problem, it now suffices to find solutions to the new knapsack, with targets $S' - \lceil n/2 \rceil, \ldots, S' + \lceil n/2 \rceil$.

Vectorial knapsack problems. Another option is to consider knapsacks whose elements are vectors of integers and where the target is a vector. Without going into the details, it is clear that this is not going to be a problem for our algorithms. In fact, the decomposition into separate components can even make things easier. Indeed, if the individual components are of the right size, they can be used as a replacement for the modular criteria that determine whether we keep or remove partial sums.

Knapsacks with ϵ_i in $\{-1, 0, 1\}$. In this case, we can apply similar methods. However, we obtain different bounds, since the number of different representations of a given solution is no longer the same. For simplicity of presentation, we assume that n is a multiple of 3 and that the solution contains $n/3$ values of each type. A simple birthday approach works by searching for a collision between two sums of $n/3$ knapsack elements. It is equal to:

$$\binom{n}{n/3} \approx \tilde{O}(2^{0.9183\, n}).$$

Note that this is higher than the expected $3^{n/2}$. A slightly more complex approach splits the knapsack in two halves and search for a collision between a left and right sum, each containing one third each of of 0, 1 and -1. This yields the expected complexity $3^{n/2}$. Using our ideas and taking a collision between two half-sums each containing two-thirds of 0s and one sixth of each of 1 and -1 of the n elements, we find a complexity $\tilde{O}(2^{0.585\, n})$ to find one of the $2^{2n/3}$ possible decompositions.

Single solution out of many. When there are many possible solutions to a knapsack problem, we may wish to combine our idea with the generalized birthday algorithm of [25] and find one of the many solutions even faster. However, this approach is difficult to analyze in general.

Combination of the above and possible applications. In fact, it is even possible to address combinations of the above. As a consequence, this algorithm can be a very useful cryptanalytic tool. For example, the NTRU cryptosystem can be seen as an unbalanced, approximate modular vector knapsack. However, it has been shown in [11] that it is best to attack this cryptosystem by using a mix of lattice reduction and knapsack-like algorithms. As a consequence, deriving new bounds for attacking NTRU would require a complex analysis, which is out of scope for the present paper. In the same vein, Gentry's fully homomorphic scheme [8], also needs to be studied with our new algorithm in mind. Another possible application would be the SWIFFT hash function [17].

Note that, in all cases, our algorithms never affect asymptotic security of a cryptographic scheme, indeed, an algorithm with complexity $2^{0.3113n}$ remains exponential. However, depending on the initial designers hypothesis, recommended practical parameters may need to be increased. For the special case of NTRU, it can be seen that in [9] that the estimates are conservative enough not to be affected by our algorithms.

7 Conclusion, Open Problems

In this paper, we have proposed new algorithms to solve the knapsack problem and other related problems, which improve on the current state of the art. In particular, for the knapsack problem itself, this improves the 31-year old algorithm of Schroeppel and Shamir and gives a positive answer to the question posed in the Open Problem Garden [6] about knapsack problems: “Is there an algorithm that runs in time $2^{n/3}$?”. Many interesting related problems are still open:

- Find a fast deterministic algorithm to solve the knapsack problem. In particular, such an algorithm could show that a given knapsack does not have a solution.
- Devise a fast Las Vegas algorithm, i.e., a randomized algorithm that can prove that a given knapsack has no solution.
- Improve our algorithms by using a full recursive approach.
- Reduce the memory requirements. Surprisingly, general cycle finding techniques do not seem to apply in this case and do not yield a constant memory algorithm with time $\tilde{O}(2^{n/2})$.

Acknowledgements. We would like to thank Igor Shparlinski for useful discussions about exponential sums and distributions of knapsack outputs. The first author also appreciates time spent thinking about this problem at NTRU Cryptosystems, Inc.

References

1. Ajtai, M.: The shortest vector problem in L2 is NP-hard for randomized reductions (extended abstract). In: 30th ACM STOC, Dallas, Texas, USA, May 23–26, pp. 10–19. ACM Press, New York (1998)
2. Camion, P., Patarin, J.: The Knapsack hash function proposed at Crypto’89 can be broken. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 39–53. Springer, Heidelberg (1991)
3. Chose, P., Joux, A., Mitton, M.: Fast correlation attacks: An algorithmic point of view. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 209–221. Springer, Heidelberg (2002)

4. Coster, M.J., Joux, A., LaMacchia, B.A., Odlyzko, A.M., Schnorr, C.-P., Stern, J.: Improved low-density subset sum algorithms. Computational Complexity 2, 111–128 (1992)
5. Damgård, I.: A design principle for hash functions. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 416–427. Springer, Heidelberg (1990)
6. Open problem garden, http://garden.irmacs.sfu.ca
7. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco (1979)
8. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher, M. (ed.) 41st ACM STOC, Bethesda, MD, USA, May 2009, pp. 169–178. ACM Press, New York (2009)
9. Hirschorn, P.S., Hoffstein, J., Howgrave-Graham, N., Whyte, W.: Choosing NTRUEncrypt parameters in light of combined lattice reduction and MITM approaches. In: Abdalla, M., Pointcheval, D., Fouque, P.-A., Vergnaud, D. (eds.) ACNS 2009. LNCS, vol. 5536, pp. 437–455. Springer, Heidelberg (2009)
10. Horowitz, E., Sahni, S.: Computing partitions with applications to the knapsack problem. J. Assoc. Comp. Mach. 21(2), 277–292 (1974)
11. Howgrave-Graham, N.: A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 150–169. Springer, Heidelberg (2007)
12. Howgrave-Graham, N., Joux, A.: New generic algorithms for hard knapsacks, eprint.iacr.org or www.joux.biz/publications/Knapsacks.pdf
13. Impagliazzo, R., Naor, M.: Efficient cryptographic schemes provably as secure as subset sum. Journal of Cryptology 9(4), 199–216 (1996)
14. Joux, A., Granboulan, L.: A practical attack against knapsack based hash functions (extended abstract). In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 58–66. Springer, Heidelberg (1994)
15. Lagarias, J.C., Odlyzko, A.M.: Solving low-density subset sum problems. J. Assoc. Comp. Mach. 32(1), 229–246 (1985)
16. Lenstra, A.K., Lenstra Jr., H.W., Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann. 261, 515–534 (1982)
17. Lyubashevsky, V., Micciancio, D., Peikert, C., Rosen, A.: Swifft: A modest proposal for fft hashing. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 54–72. Springer, Heidelberg (2008)
18. Merkle, R., Hellman, M.: Hiding information and signatures in trapdoor knapsacks. IEEE Trans. Information Theory 24(5), 525–530 (1978)
19. Nguyen, P.Q., Shparlinski, I.E., Stern, J.: Distribution of modular sums and the security of the server aided exponentiation. Progress in Computer Science and Applied Logic 20, 331–342 (2001); Final Proceedings of Cryptography and Computational Number Theory workshop, Singapore (1999)
20. Schnorr, C.-P.: A hierarchy of polynomial time lattice basis reduction algorithms. Theoretical Computer Science 53, 201–224 (1987)
21. Schroeppel, R., Shamir, A.: A $T = O(2^{n/2}), S = O(2^{n/4})$ algorithm for certain NP-complete problems. In: FOCS, pp. 328–336 (1979)
22. Schroeppel, R., Shamir, A.: A $T = O(2^{n/2}), S = O(2^{n/4})$ algorithm for certain NP-complete problems. SIAM Journal on Computing 10(3), 456–464 (1981)
23. Shamir, A.: A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) Advances in Cryptology – CRYPTO 1982, Santa Barbara, CA, USA, pp. 279–288. Plenum Press, New York (1983)

24. Stinson, D.R.: Some baby-step giant-step algorithms for the low hamming weight discrete logarithm problem. Math. Comput. 71(237), 379–391 (2002)
25. Wagner, D.: A generalized birthday problem. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 288–303. Springer, Heidelberg (2002)

A Graph of Compared Complexities

In the following figure, we present the complexity of the algorithms discussed in the paper for α-unbalanced knapsacks.

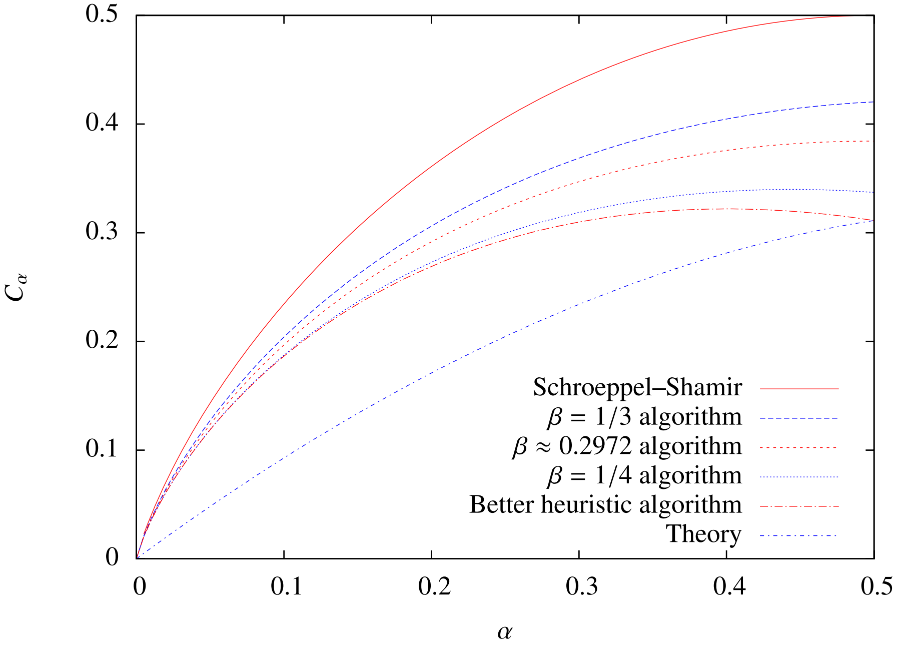


Fig. 1. Curves of time complexity exponent for varying balance factor

Lattice Enumeration Using Extreme Pruning

Nicolas Gama[1], Phong Q. Nguyen[2], and Oded Regev[3]

[1] GREYC and ENSICAEN, France
[2] INRIA and ENS, France
http://www.di.ens.fr/~pnguyen/
[3] Blavatnik School of Computer Science, Tel Aviv University, Israel
http://www.cs.tau.ac.il/~odedr/

Abstract. Lattice enumeration algorithms are the most basic algorithms for solving hard lattice problems such as the shortest vector problem and the closest vector problem, and are often used in public-key cryptanalysis either as standalone algorithms, or as subroutines in lattice reduction algorithms. Here we revisit these fundamental algorithms and show that surprising exponential speedups can be achieved both in theory and in practice by using a new technique, which we call *extreme pruning*. We also provide what is arguably the first sound analysis of pruning, which was introduced in the 1990s by Schnorr *et al.*

1 Introduction

A *lattice* is the set of all integer combinations of n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ in $\mathbb{R}^n$. These vectors are known as a *basis* of the lattice. The most basic computational problem involving lattices is the *shortest vector problem* (SVP), which asks to find a nonzero lattice vector of smallest norm, given a lattice basis as input. The inhomogeneous version of the problem is called the *closest vector problem* (CVP); here we are given an arbitrary vector in addition to the lattice basis and asked to find the lattice point closest to that vector.

Algorithms for these problems can be used to solve a wide range of problems, such as integer programming [16], factoring polynomials with rational coefficients [17], integer relation finding [15], as well as problems in communication theory (see [1,25] and references therein). They are also extremely useful in public-key cryptanalysis, notably they can break special cases of RSA and DSA (see [23] and references therein). And the growing interest in lattice-based cryptography further motivates their study.

There are two main algorithmic techniques for lattice problems. The first technique, known as *lattice reduction*, started with the celebrated LLL algorithm [17] and continued with blockwise algorithms [31,32,12] such as BKZ [32]. It works by applying successive elementary transformations to the input basis in an attempt to make its vectors shorter and more orthogonal. For usual parameters, such algorithms run in polynomial time, but the approximation factor they provide is asymptotically exponential (see [13] for experimental results). A second and more basic approach, which is the focus of our work, is the *enumeration technique*

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 257–278, 2010.

which dates back to the early 1980s with work by Pohst [27], Kannan [16], and Fincke-Pohst [11], and is still actively investigated (e.g., [32,1,14,28,30,20,35]). In its simplest form, enumeration is simply an exhaustive search for the best integer combination of the basis vectors. Enumeration algorithms run in exponential time (or worse) but find the shortest vector (as opposed to a loose approximation thereof).

The two approaches are often combined. First, blockwise lattice reduction algorithms rely on a subroutine to find short vectors in a low-dimensional lattice, whose dimension is a parameter known as the "block size". This subroutine is typically implemented through enumeration. Second, the running time of enumeration algorithms crucially depends on the quality of the input basis. Therefore, enumeration algorithms are almost never applied directly to a given basis; instead, one first applies lattice reduction and then runs an enumeration algorithm on the resulting reduced basis.

An alternative algorithmic technique for solving lattice problems was suggested in 2001 by Ajtai, Kumar, and Sivakumar [4] (see also [5,26,21,29] for recent improvements). Although this technique, known as *sieving*, leads to the asymptotically fastest algorithms for solving lattice problems exactly (running in time essentially $2^{O(n)}$), it also requires an exponential amount of space, and as a result, it is so far not useful in practice. We will not discuss this technique in the remainder of the paper.

Previous results. As mentioned above, the focus of our work is on *enumeration algorithms* which are important not just as standalone algorithms, but also as routines in lattice reduction algorithms. The basic enumeration algorithm (as in [11,32]) essentially amounts to an exhaustive search for an integer combination of the basis vectors whose norm is small enough, say, at most some given threshold R. The search can be seen as a depth-first search on a tree whose leaves correspond to lattice points, and whose internal nodes correspond to partial assignments to the coefficients of the integer combination, or geometrically, to the intersection of the lattice with subspaces (see Sect. 3 for a more detailed description). We include in the tree only those nodes whose norm is at most R.

In an attempt to speed up the running time of this algorithm, Schnorr, Euchner, and Hörner [32,33] suggested in the 1990s a modification of the basic enumeration algorithm, called *pruned enumeration.* The rough idea is to prune subtrees of the tree in which the "probability" of finding a desired lattice point is "too small".[1] By doing so, we effectively restrict our exhaustive search to a subset of all possible solutions. The hope is that although this introduces some probability of missing the desired vector, this "probability" would be small compared to the gain in running time.

Experimentally, this led to breaking certain instances of the Chor-Rivest cryptosystem, and as a result, the pruning algorithm of Schnorr and Hörner [33] made it into the popular mathematical library NTL [34], as a subroutine of BKZ [32]. Unfortunately, to the best of our knowledge, no careful analysis of

[1] Making these notions precise requires care, as we shall see later.

pruned enumeration was ever performed. The arguments that appear in the existing literature are, as far as we can tell, mostly intuitive and do not provide a proper analysis of pruned enumeration.[2]

Our results. We start out by providing what we believe is the first sound analysis of pruned enumeration. Our analysis applies to a very general family of pruned enumeration algorithms, in which the pruning is determined by an arbitrary *bounding function*, which provides for each level of the tree an upper bound on the distance of nodes that should be considered. As we will see, the running time of the enumeration algorithm is determined by the volume of certain high-dimensional bodies. Our analysis is based on two heuristic assumptions, both of which we believe are very reasonable. We also provide experimental evidence to back these assumptions.

Next, we use our analysis to understand the effect of various bounding functions on the performance of pruned enumeration algorithms. For instance, the analysis can show that well-chosen bounding functions lead asymptotically to an exponential speedup of about $2^{n/4} \approx 1.189^n$ over basic enumeration, while maintaining a success probability $\geq 95\%$.

But our main contribution is the realization that further exponential speedups can be obtained by using bounding functions that *significantly reduce* the search region. With such bounding functions, the probability of finding the desired vector is actually rather low (say, 0.1%), but surprisingly, the running time of the enumeration is reduced by a much more significant factor (say, much more than 1000). A rigorous explanation of why this happens will be given in Section 5. As a result, we can repeat the pruned enumeration algorithm several times (say, 1000) until the desired vector is found, and the total running time becomes significantly smaller than what one would obtain with standard pruned enumeration. We note that we must "reshuffle" the basis vectors before each enumeration as otherwise all the enumerations would behave identically (this will be explained in more detail when we discuss our second heuristic assumption).

We call this method, which we view as our main conceptual contribution, *extreme pruning.* We note that a similar idea is used in other algorithms; for instance, this is one of the underlying ideas in Lenstra's elliptic curve factoring method. We are not aware of any other application of this idea in the context of lattice algorithms. Our analysis shows that a well-chosen extreme pruning leads asymptotically to an exponential speedup of about $(2-\varepsilon)^{n/2} \approx 1.414^n$ over basic enumeration, which is roughly the square of the previous speedup $2^{n/4}$.

Experimental results. In practice, our best extreme pruning is able to find the shortest vector of dense knapsack lattices of dimension 110 (resp. 100) in less than 62.12 (resp. 1.73) CPU days of sequential computation on a single 1.86-Ghz core, with a negligible amount of memory and in a trivially parallelizable manner. With plain Schnorr-Euchner enumeration [32] on a BKZ-35 reduced basis, it would have taken $1.38 \cdot 10^9$ (resp. 482000) CPU years, so the speedup is about

[2] In Appendix A, we show some flaws in the analysis of Schnorr and Hörner [33].

$8.1 \cdot 10^9$ (resp. $1.0 \cdot 10^8$). We are currently running new experiments on random lattices (as used in [13]), and we expect similar exponential speedups. To the best of our knowledge, none of these dense lattices can be handled by standard lattice reduction: they are harder than the 350-dimensional lattice (solved in [22]) from the GGH challenges.

Open Questions. We expect extreme pruning to improve the performance of lattice reduction algorithms [32,13], but we leave it to future work to assess its precise impact. Our focus in this paper is on high-dimensional enumeration, whereas lattice reduction algorithms typically apply enumeration on blocks whose dimension is rather small; for instance, the experiments of [22] used a block size of 60. Currently, our extreme pruning algorithm improves enumeration by randomization, but still uses negligible space; it would be interesting to see if further improvements can be made by using more space. One way to do so would be to design new algorithms for the closest vector problem with preprocessing (CVPP). Indeed, a good CVPP algorithm can help to prune the enumeration tree in the following way: one would enumerate all the nodes at some depth k, and use the CVPP algorithm to discard those which do not lead to any leave, without having to compute all their children. Unfortunately, we have so far been unable to obtain an improvement in practice using state-of-the-art CVPP algorithms [2]. (But see [20] for a theoretically important algorithm that combines CVPP and enumeration.)

Roadmap. We start in Section 2 with some background and notation on lattices, and continue with a description of the basic enumeration algorithm in Section 3. In Section 4 we describe the pruned enumeration algorithm and give our rigorous analysis. Using that analysis, we introduce and analyze the extreme pruning algorithm in Section 5. Finally, we present our experimental results in Sect. 6. Further information is given in the Appendix: App. A discusses the Schnorr-Hörner pruning [33]; and App. B describes the code used in our experiments, which includes an apparently new implementation trick that speeds up enumeration.

2 Preliminaries

Lattices are discrete subgroups of $\mathbb{R}^m$. Any lattice L can be defined by a basis, which is a set of linearly independent vectors $(\mathbf{b}_1, \dots, \mathbf{b}_n)$ in $\mathbb{R}^m$ such that L is equal to the set $L(\mathbf{b}_1, \dots, \mathbf{b}_n) = \{\sum_{i=1}^n x_i \mathbf{b}_i, x_i \in \mathbb{Z}\}$ of all integer linear combinations of the $\mathbf{b}_i$'s. All the bases of L have the same number n of elements, called the dimension of L, and they all have the same volume, called the volume $\mathrm{vol}(L)$ or determinant of L. Throughout the paper, we use row representations of matrices. The Euclidean norm of a vector $\mathbf{v} \in \mathbb{R}^m$ is denoted $\|\mathbf{v}\|$. We denote by $\mathrm{Ball}_n(R)$ the n-dimensional Euclidean ball of radius R, and by $V_n(R) = R^n \cdot \frac{\pi^{n/2}}{\Gamma(n/2+1)}$ its volume. The n-dimensional unit sphere is denoted by S^{n-1}.

Shortest vector. A lattice L contains non-zero vectors of minimal Euclidean norm: this norm is called the *first minimum* $\lambda_1(L)$ of L. A vector of norm $\lambda_1(L)$ is called a *shortest vector* of L, and is in general unique up to the sign. Hermite's constant γ_n is the supremum of the ratio $(\lambda_1(L)/\mathrm{vol}(L)^{1/n})^2$ over all n-dimensional lattices. Minkowski's theorem shows that $\sqrt{\gamma_n}$ is smaller than the diameter of the n-dimensional ball of volume 1: $\sqrt{\gamma_n} \leq 2 \cdot V_n(1)^{-1/n} \leq \sqrt{n}$.

Orthogonalization. A basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ can be written uniquely as a product $B = \mu \cdot D \cdot Q$ where $\mu = (\mu_{i,j})$ is an $n \times n$ lower-triangular matrix with unit diagonal, D an n-dimensional positive diagonal matrix and Q an $n \times m$ matrix with orthonormal row vectors. Then μD is a lower triangular representation of B (with respect to Q), $B^* = DQ = (\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*)$ is the Gram-Schmidt orthogonalization of the basis, and D is the diagonal matrix formed by the $\|\mathbf{b}_i^*\|$'s. For all $i \in \{1, \ldots, n+1\}$, we denote by π_i the orthogonal projection on $(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})^\perp$. For all $i \in \{1, \ldots, n+1\}$, $\pi_i(L)$ is an $n+1-i$ dimensional lattice generated by the basis $(\pi_i(\mathbf{b}_i), \ldots, \pi_i(\mathbf{b}_n))$, with $\mathrm{vol}(\pi_i(L)) = \prod_{j=i}^{n} \|\mathbf{b}_j^*\|$.

Reduced bases. Lattice reduction algorithms aim to transform an input basis into a "high quality" basis. There are many ways to quantify the quality of bases produced by lattice reduction algorithms. One popular way, which is particularly useful for our purposes, is to consider the Gram-Schmidt norms $\|\mathbf{b}_1^*\|, \ldots, \|\mathbf{b}_n^*\|$. Intuitively speaking, a good basis is one in which this sequence never decays too fast. In practice, it turns out that the Gram-Schmidt coefficients of bases produced by the main reduction algorithms (such as LLL or BKZ) have a certain "typical shape", assuming the input basis is sufficiently random. This property was thoroughly investigated in [13,24]: accordingly, our speedup analysis assumes to simplify that $\|\mathbf{b}_i^*\|/\|\mathbf{b}_{i+1}^*\| \approx q$ where q depends on the reduction algorithm.

Gaussian Heuristic. The Gaussian Heuristic provides an estimate on the number of lattice points inside a "nice enough" set.

Heuristic 1. *Given a lattice L and a set S, the number of points in $S \cap L$ is approximately* $\mathrm{vol}(S)/\mathrm{vol}(L)$.

In some cases, this heuristic can be proved. For instance, Ajtai showed [3] that for any finite Borel set S of measure V which does not contain 0, the expectation of $S \cap L$ taken over a certain natural distribution on lattices L of volume D is V/D. In particular, the expectation of $\lambda_1(L)$ on random lattices of volume D is the radius of the n-dimensional ball of volume D, that is $D^{1/n} \cdot V_n(1)^{-1/n}$, which is often used as a "prediction" of $\lambda_1(L)$ for a "typical" lattice. There are also counterexamples to this heuristic (see, e.g., [19] for counterexamples in $\mathbb{Z}^n$).

3 Enumeration

We recall Schnorr-Euchner's enumeration algorithm [32] which is the enumeration algorithm used in practice, and analyze its cost.

3.1 Setting

To simplify the exposition, we assume in the rest of the paper the following setting. Let L be a lattice whose shortest vector $\mathbf{v}$ is unique (up to sign). Our goal is to find $\mathbf{v}$. (Our entire analysis can be extended in a straightforward manner to the closest vector problem.) We assume we are given a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of L and a very good upper bound R on $\lambda_1(L)$ so that finding $\pm\mathbf{v}$ amounts to finding any nonzero lattice vector $\mathbf{w} \in L$ such that $\|\mathbf{w}\| \leq R$, and therefore, we can easily check whether or not the solution is correct. In many practical situations, $\lambda_1(L)$ is known exactly: this is typically true in cryptographic situations, such as in CJLOSS lattices [7]. In the full version, we will explain how to adapt our analysis to the general case of SVP.

3.2 Description

To find $\pm\mathbf{v}$, enumeration goes through the *enumeration tree* formed by all vectors in the projected lattices $\pi_n(L)$, $\pi_{n-1}(L)$, $\ldots, \pi_1(L)$ of norm at most R. More precisely, the enumeration tree is a tree of depth n, and for each $k \in \{0, \ldots, n\}$, the nodes at depth k are all the vectors of the rank-k projected lattice $\pi_{n+1-k}(L)$ with norm at most R. In particular, the root of the tree is the zero vector (because $\pi_{n+1}(L) = \{0\}$), while the leaves are all the vectors of L of norm $\leq R$. The parent of a node $\mathbf{u} \in \pi_{n+1-k}(L)$ at depth k is by definition $\pi_{n+2-k}(\mathbf{u})$ at depth $k-1$. And we order the child nodes by increasing Euclidean norm: note that all ancestors of a given node are at most as long as the node, because they are projections of the node.

We note that the tree is symmetric, because if $\mathbf{x} \in L$ then $-\mathbf{x} \in L$. Thus, we halve the tree by restricting to "positive" nodes: we only consider nodes $\pi_{n+1-k}(\mathbf{u})$ where the last nonzero coordinate of $\mathbf{u} \in L$ with respect to $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is positive. From now on, by enumeration tree, we mean this halved enumeration tree, which has a single leaf, either $\mathbf{v}$ or $-\mathbf{v}$. The Schnorr-Euchner algorithm [32] performs a Depth First Search of the tree to find the single leaf. The more reduced the basis is, the less nodes in the tree, and the cheaper the enumeration.

Concretely, the shortest vector $\mathbf{v} \in L$ may be written as $\mathbf{v} = v_1\mathbf{b}_1 + \cdots + v_n\mathbf{b}_n$ where the v_i's are unknown integers and $\mathbf{b}_i = \mathbf{b}_i^* + \sum_{j=1}^{i-1} \mu_{i,j}\mathbf{b}_j^*$. Then $\mathbf{v} = \sum_{j=1}^{n} \left(v_j + \sum_{i=j+1}^{n} \mu_{i,j}v_i\right) \mathbf{b}_j^*$, which gives the norms of its projections as:

$$\|\pi_{n+1-k}(\mathbf{v})\|^2 = \sum_{j=n+1-k}^{n} \left(v_j + \sum_{i=j+1}^{n} \mu_{i,j}v_i\right)^2 \|\mathbf{b}_j^*\|^2, \ 1 \leq k \leq n \qquad (1)$$

Now, if $\mathbf{v}$ is a leaf of the tree, then the n inequalities $\|\pi_{n+1-k}(\mathbf{v})\| \leq R$ together with (1) enable us to perform an exhaustive search for the coordinates $v_n, v_{n-1}, \ldots, v_1$ of $\mathbf{x}$:

$$\sum_{j=n+1-k}^{n} \left(v_j + \sum_{i=j+1}^{n} \mu_{i,j}v_i\right)^2 \|\mathbf{b}_j^*\|^2 \leq R^2, \quad 1 \leq k \leq n,$$

which can be rewritten for $1 \leq k \leq n$ as

$$\left| v_{n+1-k} + \sum_{i=n+2-k}^{n} \mu_{i,j} v_i \right| \leq \frac{\sqrt{R^2 - \sum_{j=n+2-k}^{n} \left(v_j + \sum_{i=j+1}^{n} \mu_{i,j} v_i \right)^2 \|\mathbf{b}_j^*\|^2}}{\|\mathbf{b}_{n+1-k}^*\|} \tag{2}$$

We start with $k = 1$ in (2), that is: $0 \leq v_n \leq R/\|\mathbf{b}_n^*\|$ because we restricted to "positive" nodes. This allows to perform an exhaustive search for the integer v_n, and we do so by increasing values of v_n. Now, assume that the projection $\pi_{n+2-k}(\mathbf{v})$ has been guessed for some k: the integers $v_{n+2-k}, \ldots, v_n$ are known. Then (2) enables to compute an interval I_{n+1-k} such that $v_{n+1-k} \in I_{n+1-k}$, and therefore to perform an exhaustive search for v_{n+1-k}. A Depth First Search of the tree corresponds to enumerating I_{n+1-k} from its middle, by increasing values of $\|\pi_{n+1-k}(\mathbf{v})\|$, namely $v_{n+1-k} = \lfloor - \sum_{i=n+2-k}^{n} \mu_{i,j} v_i \rceil$, $\lfloor - \sum_{i=n+2-k}^{n} \mu_{i,j} v_i \rceil \pm 1$, and so on.

3.3 Complexity

The running time of the enumeration algorithm is N polynomial-time operations where N is the total number of tree nodes. Hence, in order to analyze this running time, we need to obtain good estimates of N. As already suggested by Hanrot and Stehlé [14], a good estimate of N can be derived from the Gaussian heuristic. More precisely, the number of nodes at level k is exactly half the number of vectors of $\pi_{n+1-k}(L)$ of norm $\leq R$ (where the half comes because we halved the tree). Since $\mathrm{vol}(\pi_{n+1-k}(L)) = \prod_{i=n+1-k}^{n} \|\mathbf{b}_i^*\|$, the Gaussian heuristic predicts the number of nodes at level k scanned by the Schnorr-Euchner algorithm to be close to

$$H_k = \frac{1}{2} \cdot \frac{V_k(R)}{\prod_{i=n+1-k}^{n} \|\mathbf{b}_i^*\|}. \tag{3}$$

If this holds, then $N \approx \sum_{k=1}^{n} H_k$. In Sect. 6.1, we present experiments that strongly support this heuristic estimate.

For a typical reduced basis, we have $\|\mathbf{b}_i^*\|/\|\mathbf{b}_{i+1}^*\| \approx q$ where q depends on the reduction algorithm (see [13]). The bound $R = \sqrt{\gamma_n}\mathrm{vol}(L)^{1/n}$ is optimal in the worst case. Since $\sqrt{\gamma_n} = \Theta(\sqrt{n})$, an elementary computation shows that (3) becomes:

$$H_k \approx \frac{\|\mathbf{b}_1\|^{n-k} 2^{O(n)}}{q^{(n-1-k)(n-k)/2} \mathrm{vol}(L)^{(n-k)/n}} = \frac{q^{(n-k)(n-1)/2} 2^{O(n)}}{q^{(n-1-k)(n-k)/2}} = q^{(n-k)k/2} 2^{O(n)},$$

where the right-hand term is always less than $q^{n^2/8} 2^{O(n)}$ because $(n-k)(k/2)$ is maximized for $k = n/2$. Hence:

$$H_k \lessapprox q^{n^2/8} 2^{O(n)}.$$

Thus, $\max_k H_k$ is super-exponential in n and is reached for $k \approx n/2$, which is consistent with experiments (see Fig. 1 of Sect. 5.2). For small values of n, the term $2^{O(n)}$ is not negligible, and may shift a bit the maximum index $k \approx n/2$.

We note that if we make the (reasonable) assumption that the location of the leaf is uniform, the number of nodes scanned by the enumeration algorithm will only be $N/2$ in expectation, and not N. For simplicity, we ignore this factor 2 in the sequel.

Finally, we mention that *rigorous* bounds on N exist. For instance, if the basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is LLL-reduced, and $R = \|\mathbf{b}_1\|$, then it is well-known that N is at most $2^{O(n^2)}$. Also, Hanrot and Stehlé [14] showed that if the basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is so-called quasi-HKZ-reduced, and $R = \|\mathbf{b}_1\|$, then $N \leq n^{n/(2e)+o(n)}$. See [14] for details.

4 Pruned Enumeration

Since enumeration is expensive, it is tempting not to enumerate all the tree nodes, by discarding certain branches. The idea of pruned enumeration goes back to Schnorr and Euchner [32], and was further studied by Schnorr and Hörner [33]. For instance, one might intuitively hope that $\|\pi_{n/2}(\mathbf{v})\|^2 \lessapprox \|\mathbf{v}\|^2/2$, which is more restrictive than the inequality $\|\pi_{n/2}(\mathbf{v})\|^2 \leq \|\mathbf{v}\|^2$ used by enumeration. Formally, pruning replaces each of the n inequalities $\|\pi_{n+1-k}(\mathbf{v})\| \leq R$ by $\|\pi_{n+1-k}(\mathbf{v})\| \leq R_k$ where $R_1 \leq \cdots \leq R_n = R$ are n real numbers defined by the pruning strategy. This means that one replaces R by R_k in each of the n inequalities (2).

At the end of their paper [32] , Schnorr and Euchner briefly proposed $R_k = R\min(1, \sqrt{(1.05)k/n})$, but did not provide any analysis, only limited experiments. Schnorr and Hörner [33] later proposed another choice of R_k's, which we discuss in App. A: we show that this pruning has flaws, and that the analysis of [33] is mostly incorrect; in particular, if the heuristic analysis of [33] was correct, it would imply polynomial-time algorithms for the shortest vector problem, while the problem is NP-hard under randomized reductions.

We now provide what we believe is the first rigorous analysis of pruned enumeration: The next two subsections deal with the running time and the success probability. In principle, this analysis can be used to optimize the choice of the bounding function in the pruning algorithm. However, we did not attempt to do that since this is not the main focus of our paper. Instead, our analysis will be used in the next section to show how exponential speedups (both in theory and in practice) can be achieved through the use of extreme pruning.

4.1 Running Time Analysis

The running time of the pruned enumeration algorithm is given by

$$T_{\text{node}} \cdot N$$

where T_{node} is the average amount of time spent processing one node in the enumeration tree, and N is the number of nodes in the pruned tree. In order to estimate N, we estimate the number of nodes at each level of the search tree using the Gaussian heuristic (Heuristic 1). As we shall see later, our estimates agree very nicely with the experiments, giving some further justification to the use of the Gaussian heuristic.

Our estimate is very similar to the one in Equation 3, except that instead of using balls of radius $\sqrt{R}$, we use *cylinder-intersections* of radii $(R_1, \ldots, R_k)$ for $1 \leq k \leq n$. More specifically, define the (k-dimensional) cylinder-intersection of radii $R_1 \leq \cdots \leq R_k$ as the set

$$C_{R_1,\ldots,R_k} = \left\{ (x_1, \ldots, x_k) \in \mathbb{R}^k,\ \forall j \leq k,\ \sum_{l=1}^{j} x_l^2 \leq R_j^2 \right\}.$$

Notice that the set of vertices in level k of the pruned tree correspond exactly to the points of the projected lattice $\pi_{n+1-k}(L)$ that are inside $C_{R_1,\ldots,R_k}$. Therefore, using the Gaussian heuristic, we can estimate the number of nodes in the enumeration tree using

$$N = N_{R_1,\ldots,R_n}(\|\mathbf{b}_1^*\|, \ldots, \|\mathbf{b}_n^*\|) = \frac{1}{2} \sum_{k=1}^{n} \frac{V_{R_1,\ldots,R_k}}{\prod_{i=n+1-k}^{n} \|\mathbf{b}_i^*\|} \tag{4}$$

where $V_{R_1,\ldots,R_k}$ denotes the volume of $C_{R_1,\ldots,R_k}$, and the factor half is as a result of the symmetry in the SVP problem.

There are several ways to compute or approximate the volume of cylinder intersections $V_{R_1,\ldots,R_k}$. The simplest and most naïve method, which is the one we used at first in our numerical optimizations, is based on a Monte Carlo method. Namely, by observing that the cylinder intersection $C_{R_1,\ldots,R_k}$ is contained in a ball of radius R_k, we can write

$$V_{R_1,\ldots,R_k} = V_k(R_k) \cdot \Pr_{\mathbf{u} \sim \mathrm{Ball}_k} \left(\forall j \in [1,k],\ \sum_{i=1}^{j} u_i^2 \leq \frac{R_j^2}{R_k^2} \right). \tag{5}$$

The number of samples required to estimate the above probability by Monte Carlo sampling is proportional to its inverse. One can speed things up significantly by replacing the ball with a smaller containing body (such as another cylinder intersection) whose volume is known and from which we can sample uniformly.

For certain interesting choices of radii $(R_1, \ldots, R_k)$, rigorous estimates can be obtained, as we shall see in Section 5. Moreover, one particular case in which we can compute the volume *exactly* is when $R_1 = R_2$, $R_3 = R_4$, etc. and k is even. This is because the distribution of the vector $(u_1^2 + u_2^2, u_3^2 + u_4^2, \ldots, u_{k-1}^2 + u_k^2)$ when $\mathbf{u}$ is chosen from Ball_k is given by a Dirichlet distribution with parameters $(1, \ldots, 1)$ ($k/2 + 1$ ones), which is simply a uniform distribution over the set of all vectors whose coordinates are non-negative and sum to at most 1 (see Page 593 of [9]). This leads to an easy way to compute the probability in Eq. 5 exactly (as it amounts to computing the volume of a certain polytope). This calculation can also be combined with the Monte Carlo simulation above, leading to much faster running times.

Finally, let us also mention that there is a large body of work showing *provably polynomial time* algorithms that provide a good approximation to the volume of *any* convex body (see, e.g., [10,18]). However, in practice these algorithms are rather slow and are therefore probably not too useful for our purposes.

4.2 Success Probability Analysis

We let p_{succ} denote the "probability" that the target vector is still in the tree after the pruning. Prior to our work, the implicit assumption was that one should choose a bounding function so as to minimize the running time while keeping p_{succ} reasonably high, say 95%. As we shall see in the next section, surprising speedups can be obtained through *extreme pruning*, i.e., when p_{succ} is very small.

Before proceeding with the analysis, we must explain what we mean by "probability", since the pruning algorithm is entirely deterministic. (We note that in previous work this was often glossed over; see App. A). In order to meaningfully talk about the success probability p_{succ}, we must assume some kind of distribution on the inputs (since the pruning algorithm is entirely deterministic). For that purpose, we make the following heuristic assumption on the input basis:

Heuristic 2. *The distribution of the coordinates of the target vector* $\mathbf{v}$*, when written in the normalized Gram-Schmidt basis* $(\mathbf{b}_1^*/\|\mathbf{b}_1^*\|, \ldots, \mathbf{b}_n^*/\|\mathbf{b}_n^*\|)$ *of the input basis, look like those of a uniformly distributed vector of norm* $\|\mathbf{v}\|$*.*

We use here the imprecise term 'looks like' on purpose. It should be interpreted simply as saying that the estimate on p_{succ} obtained by performing the analysis under the above assumption on the coordinates of the shortest vector corresponds to what one observes in practice on any reasonable distribution of inputs (see Sect. 6 for experiments). We note that Heuristic 2 would follow from a (stronger) natural heuristic on reduced bases:

Heuristic 3. *The distribution of the normalized Gram-Schmidt orthogonalization* $(\mathbf{b}_1^*/\|\mathbf{b}_1^*\|, \ldots, \mathbf{b}_n^*/\|\mathbf{b}_n^*\|)$ *of a random reduced basis* $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ *looks like that of a uniformly distributed orthogonal matrix.*

Here, we assume that the reduction is not too strong, that is, the reduced basis is made of vectors that are significantly longer than the shortest vector of the lattice. In typical randomly constructed lattices, the number of such vectors is exponential,[3] and hence, we may hope that the reduced basis is not oriented in any particular direction.

We now estimate the success probability p_{succ}. Let $\mathbf{v}$ be the target vector, and let $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ be its coordinates in the orthonormal basis $(\mathbf{b}_n^*/\|\mathbf{b}_n^*\|, \ldots, \mathbf{b}_1^*/\|\mathbf{b}_1^*\|)$ (notice that the coordinates are reversed, with x_1 corresponding to $\mathbf{b}_n$ etc.). By definition, $\mathbf{v}$ belongs to the pruned tree if and only

[3] Moreover, for any $c \geq 1$ and any n-dimensional lattice L, the number of lattice points of norm at most $2cV_n(1)^{-1/n}\text{vol}(L)^{1/n}$ is at least $\lceil c^n \rceil$. But this bound only applies for radii above Minkowski's upper bound, which is twice the Gaussian heuristic.

if for all $k = 1, \ldots, n$, $\sum_{j=1}^{k} x_j^2 \leq R_k^2$. By Heuristic 2, $\mathbf{x}$ is distributed like a uniform vector subject to the constraints that $\|\mathbf{x}\| = \|\mathbf{v}\|$. Hence, we can estimate p_{succ} as

$$p_{\text{succ}} = p_{\text{succ}}(R_1, \ldots, R_n) = \Pr_{u \sim S^{n-1}\|\mathbf{v}\|/R}\left(\forall j \in [1,n],\ \sum_{l=1}^{j} u_l^2 \leq \frac{R_j^2}{R_n^2}\right) \qquad (6)$$

where S^{n-1} denotes the unit sphere in n dimensions. As before, one can estimate this probability through Monte Carlo simulation, or compute it exactly in certain cases (e.g., using the fact that if u is chosen from S^{n-1} for even n, then $(u_1^2 + u_2^2, \ldots, u_{n-3}^2 + u_{n-2}^2)$ is distributed uniformly over all vectors whose coordinates are non-negative and sum to at most 1).

5 Extreme Pruning

In this section we present our main contribution, the extreme pruning algorithm, whose main idea is to apply pruning using bounding functions whose p_{succ} is very small. Our algorithm takes as input a lattice basis, as well as n real numbers $R_1^2 \leq \cdots \leq R_n^2 = R^2$ (the bounding function) where R_k corresponds to the pruning in depth k. The goal of the algorithm is to find a vector of length at most R, and is described in Algorithm 1.

Algorithm 1. Extreme Pruning

Repeat until a vector of length at most R is found:

1. Randomize the input basis, and apply basis reduction to it.
2. Run the enumeration on the tree pruned with radii $R_1, \ldots, R_n$, as explained in Sect. 4.

We are being deliberately imprecise in Step 1 of the algorithm. First, we do not know what the best method of randomization is, and it is quite likely that this does not matter much. In our experiments, we simply multiplied the input basis by some small unimodular matrix chosen at random, but one can also use other methods. Second, as we shall see in the analysis below, the choice of basis reduction has a great effect on the overall running time, and has to be set properly. The choice of basis reduction algorithm, as well as of the bounds $R_1, \ldots, R_n$ will be the topic of Sections 5.2 and 5.3. But first we analyze the expected running time of the algorithm.

5.1 Running Time Analysis

We now analyze the expected running time of the extreme pruning algorithm based on the analysis in Section 4.

First, we estimate the probability of success in each iteration of the algorithm by $p_{\text{succ}}(R_1, \dots, R_n)$, as in Eq. 6, which is based on Heuristic 2. Here, we explicitly perform a randomization before reduction, and we stress that Heuristic 2 produces estimates on p_{succ} that agree very nicely with our experiments (see Sect. 6 for more details).

Next, we estimate the running time of each iteration of the algorithm. Let us denote the (average) running time of Step 1 by T_{reduc}. Once a choice of randomization and reduction algorithm is fixed, this time can be easily estimated experimentally. The running time of Step 2 can be estimated by $N_{R_1,\dots,R_n}(\|\mathbf{b}_1^*\|, \dots, \|\mathbf{b}_n^*\|)$, as in Eq. 4. Notice that this running time depends on the Gram-Schmidt coefficients of the basis produced in Step 1, and might vary from one iteration to another. In order to simplify the analysis, we assume that these Gram-Schmidt coefficients are the same throughout all iterations, and denote them by $\bar{\mathbf{b}}_1^*, \dots, \bar{\mathbf{b}}_n^*$. This is partly justified by the observation that bases produced by known reduction algorithms have a clear shape that depends only on the reduction algorithm and not so much on the input basis. Alternatively, it should be straightforward to refine our analysis so that it takes into account the *distribution* of Gram-Schmidt coefficients produced in Step 1, as opposed just to their average. Yet another possibility is to modify the algorithm so that 'bad' bases (i.e., those that differ significantly from the average behavior) are discarded.

To summarize, we can estimate the expected time required to find the desired vector by

$$T_{\text{extreme}}(R_1, \dots, R_n, \bar{\mathbf{b}}_1^*, \dots, \bar{\mathbf{b}}_n^*) := \frac{T_{\text{reduc}} + T_{\text{node}} \cdot N_{R_1,\dots,R_n}(\bar{\mathbf{b}}_1^*, \dots, \bar{\mathbf{b}}_n^*)}{p_{\text{succ}}(R_1, \dots, R_n)}. \quad (7)$$

5.2 Choosing Parameters for the Experiments

In order to optimize the running time of the extreme pruning algorithm, we need to choose the basis reduction algorithm used in Step 1, as well as the bounding parameters $R_1 \le \dots \le R_n$ used in Step 2. These choices are crucial in determining the running time of the algorithm.

Since finding the exact optimum of the expression in Eq. (7) seems difficult, we decided to try numerical optimization. We wrote a program that starts from the linear bounding function $R_k^2 = (k/n) \cdot R^2$ and successively tries to apply small random modifications to it. After each such modification it checks if the expression in Eq. (7) decreased or not, with an exact computation based on the Dirichlet distribution (as described in Sect. 4.1). If it did, the modification is accepted; otherwise it is rejected.

This yielded bounding functions whose predicted running time is only 62.1 CPU days (including reduction time) in dimension 110 for hard knapsack lattices, which is significantly better than the linear bounding function (see Figure 1).

5.3 Asymptotic Analysis

In this section, we provide a rigorous justification to the effectiveness of extreme pruning. We do this through an analysis of three bounding functions, whose

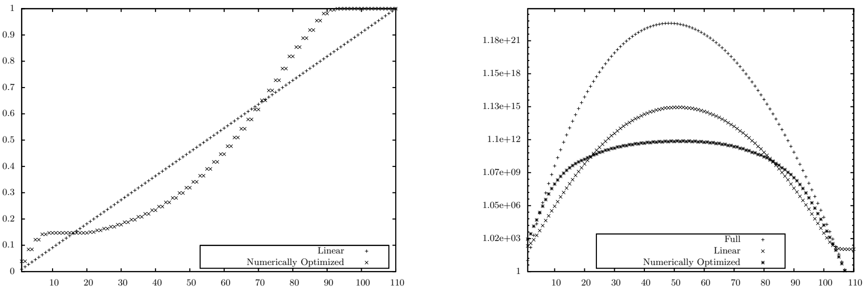


Fig. 1. On the left are the linear function and our best bounding function found by numerical optimization. On the right we compare the estimated expected number of nodes (with respect to the depth) visited in a run of full enumeration, extreme pruning with the linear function, and extreme pruning using our best bounding function. Note that basis reduction times are ignored here.

asymptotic behavior we can analyze (under our reasonable heuristic assumptions). The first is the linear bounding function. The speedup it offers over full enumeration is provably exponential, but it is not significantly better than what we can achieve using non-extreme pruning. This is perhaps not surprising since its success probability p_{succ} is $1/n$, which is relatively high (so this bounding function is not too 'extreme'). Our second example is a step bounding function. We show that this function obtains an exponential improvement over our best non-extreme bounding function in terms of the number of nodes scanned in the middle level of the tree. This function nicely highlights the reason extreme pruning is superior to non-extreme pruning. Our third bounding function is a piecewise linear bounding function. Its analysis combines the previous two analyses and leads to the best speedups we can rigorously demonstrate.

Throughout this section, we make the simplifying assumption that $\|\mathbf{v}\| = R$ (which, as discussed above, is the case in many scenarios, and is also the worst case for pruning algorithms). We also ignore the reduction time, which in practice might make the claimed speedups a bit smaller (but does not affect the asymptotics). Finally, instead of giving an absolute bound on the number of nodes in (each level of) the enumeration tree, we compare this number to that in the full enumeration tree. This allows us to focus on analyzing the volume of cylinder intersections, and ignore the properties of the given basis, which we leave for future work.

Linear pruning. We define the linear bounding function as $R_k^2 = ({}^k\!/_n) \cdot R^2$, for $k = 1, \ldots, n$. The motivation for this setting comes from the fact that if $\mathbf{v}$ is a uniformly random vector of length R (as we assume in our heuristic), then the expectation of the squared norm of its projection on the first k coordinates is exactly $({}^k\!/_n) \cdot R^2$. Although this bounding function leads to running times that are far from optimal, it is interesting as it can be analyzed rigorously and shown to provide an exponential speedup over full enumeration. In the full version, we prove the following claim, which states that the success probability of linear pruning is exactly $1/n$:

Claim. Let u be a vector uniformly distributed in the unit sphere S^{n-1}. Then,

$$\Pr_u\left(\forall j \in \{1,\ldots,n\},\ \sum_{i=1}^{j} u_i^2 \leq \frac{j}{n}\right) = \frac{1}{n}.$$

This also shows that linear pruning keeps an exponentially small proportion (between ${}^1\!/_k\,({}^k\!/_n)^{k/2}$ and $({}^k\!/_n)^{k/2}$ at each depth k) of nodes:

Corollary 1. *For any integer $n \geq 1$,*

$$V_n(1)/n \leq \mathrm{vol}\left\{u \in \mathrm{Ball}_n(1) : \forall j \in \{1,\ldots,n\},\ \sum_{i=1}^{j} u_i^2 \leq \frac{j}{n}\right\} \leq V_n(1). \qquad (8)$$

Hence, compared to full enumeration, linear pruning reduces the number of nodes at depth k by the multiplicative factor $(n/k)^{k/2}$, up to some polynomial factor. As we saw in Section 3, for typical reduced bases, most of the nodes in the enumeration tree are concentrated around $n/2$, in which case the speedup is approximately $2^{n/4} \approx 1.189^n$.

Step bounding function. We now present an example in which extreme pruning is significantly superior to non-extreme pruning, at least in the asymptotic sense.

Consider the step bounding function given by $R_k^2 = \alpha R^2$ for $1 \leq k \leq n/2$, and $R_k = R$ otherwise, where $\alpha > 0$ is a constant to be determined later. (We fix the location of the step to $n/2$ for simplicity; the analysis can be easily extended to any step function.) With this bounding function, the number of nodes in the middle level of the pruned enumeration tree compared to that in the middle level of the full enumeration tree is smaller by a factor of $\alpha^{\frac{n}{4}}$. We omit the analysis for other levels of the tree because our next bounding function will have a much better performance in this respect, and its analysis is not more complicated.

We now compute the success probability p_{succ}. Eq. 6 tells us that p_{succ} is equal to the probability that for a random point $\mathbf{u} = (u_1,\ldots,u_n)$ on the sphere S^{n-1}, we have $\sum_{i=1}^{n/2} u_i^2 \leq \alpha$. It follows from classical concentration inequalities (see, e.g., Lemma 2.2 in [8]) that this probability is $1 - 2^{-\Omega(n)}$ if $\alpha > 1/2$, and $2^{-\Omega(n)}$ if $\alpha < 1/2$. Hence, for $\alpha > 1/2$, we are in the non-extreme regime, and by choosing α close to $1/2$ we obtain that the number of nodes in the middle level of the pruned enumeration tree is smaller by a factor of about $2^{n/4} \approx 1.189^n$ than that in the full enumeration tree. Since most of the nodes are located at depths around $n/2$, this is a good approximation of the speedup offered by non-extreme pruning using a step function.

Let us now consider what happens when we take $\alpha < 1/2$ and move into the extreme pruning regime. By definition, $\sum_{i=1}^{n/2} u_i^2$ is distributed according to the beta distribution $\mathrm{Beta}(\frac{n}{4}, \frac{n}{4})$. Therefore, we can express p_{succ} using the regularized incomplete beta function as:

$$p_{\mathrm{succ}} = I_\alpha(\frac{n}{4}, \frac{n}{4}) \geq \frac{(\frac{n}{2}-1)!}{(\frac{n}{4})!(\frac{n}{4}-1)!}\,\alpha^{\frac{n}{4}}(1-\alpha)^{\frac{n}{4}} - 1 = \Omega\left(\tfrac{1}{\sqrt{n}}(4\alpha(1-\alpha))^{\frac{n}{4}}\right)$$

where the inequality follows from integration by parts of the incomplete beta function. Hence, the total number of middle level nodes that will be scanned in the extreme pruning algorithm (which can be seen as an approximation of the overall running time) is smaller than that in full enumeration by a factor of

$$\Omega\left(\tfrac{1}{\sqrt{n}}(4(1-\alpha))^{\frac{n}{4}}\right).$$

This expression is maximized for very small $\alpha > 0$, in which case the speedup is asymptotically roughly $4^{\frac{n}{4}} \approx 1.414^n$, which is greatly superior to the speedup of 1.189^n obtained with non-extreme pruning.

As this example nicely demonstrates, the advantage of extreme pruning comes from the fact that for small $\alpha > 0$, although the success probability is exponentially small, the volume of the search region decreases by a stronger exponential factor.

Piecewise linear bounding function. Consider the bounding function defined as $R_k^2 = ({}^{2k}/_{n})\alpha \cdot R^2$ for $k = 1, \ldots, n/2$, and $R_k^2 = (2\alpha - 1 + 2k(1-\alpha)/n) \cdot R^2$ otherwise, where we assume that $0 < \alpha < 1/4$ is a constant. In the full version, using similar arguments than for linear and step pruning, we show that: $p_{\text{succ}} \geq \Omega\left(n^{-5/2}(4\alpha(1-\alpha))^{\frac{n}{4}}\right)$. Ignoring polynomial factors, for sufficiently large n, the total number of level k nodes that will be scanned in the extreme pruning algorithm is shown to be smaller than that in full enumeration by a factor of

$$\Omega\Big(2^{\frac{n}{2}-\frac{k}{2}}\,\alpha^{\frac{n}{4}-\frac{k}{2}}\,(1-\alpha)^{\frac{n}{4}}\left(\frac{n}{k}\right)^{\frac{k}{2}}\Big)$$

for $k \leq n/2$ and by a factor of

$$\Omega\Big(2^{\frac{n}{2}-\frac{k}{2}}\,(1-\alpha)^{\frac{n}{2}-\frac{k}{2}}\left(\frac{n}{k}\right)^{\frac{k}{2}}\Big).$$

for $k > n/2$. We see that for levels around $n/2$ (say, $0.49n \leq k \leq 0.51n$), in order to maximize the expressions above, one should choose a small α, in which case the speedup is roughly of $2^{\frac{n}{2}} \approx 1.414^n$. For other values of k, the number of nodes may actually increase, but typically these levels contain a small fraction of the nodes, and the global asymptotical speedup is not affected.

6 Experiments

The setup. All our experiments are run on 64-bit Xeon processors with frequency 1.86 GHz, and compiled with g++ version 4.2.4 x86_64 (options -O9 -ffast-math -funroll-loops -ftree-vectorize). Running times are provided for a single core.

Implementation. For lattice reduction, we used fplll [6]'s implementation of LLL, and NTL [34]'s implementation of BKZ [32]. We implemented our own enumeration algorithms (see App. B) in basic C++, using double and long arithmetic; we plan to release the source codes. The input basis and its Gram-Schmidt orthogonalization were pre-computed with NTL [34] in RR precision, and then rounded to double precision before entering the enumeration procedure. While floating-point arithmetic is known to cause stability problems during LLL reduction, we did not experience such problems during enumeration, even up to dimension 110; we note that a rigorous analysis of enumeration with floating-point arithmetic has recently been done in [28].

Lattices. It is important to test algorithms on lattices that do not have a special structure that can be exploited by standard reduction algorithms. On the other hand, we need to be able to decide if the algorithm was successful, therefore $\lambda_1(L)$ must be known. For concreteness, we performed all our experiments on hard knapsack lattices, namely the so-called CJLOSS lattices [7] of density 0.94 where the knapsack solution was further chosen with exactly as many 0s as 1s. For these lattices, we can easily check whether the vector found is the shortest vector because it corresponds to knapsack solutions. It can also be checked experimentally that they are quite dense, in the sense that their first minimum $\lambda_1(L)$ is close to the Gaussian heuristic, which can be seen as an indication that they are hard instances of SVP (see [13]). Moreover, these lattices, we believe, serve as a good representative of hard lattices that typically occur in practice. In particular, we believe that the results reported here are not limited to CJLOSS lattices, but in fact represent a general phenomenon. We are currently running new experiments on random lattices (as used in [13]), for which we have a tight estimate on the first minimum $\lambda_1(L)$, but do not know shortest vectors in advance.

Rate of enumeration. The amount of time an enumeration algorithm spends per node in the tree might depend slightly on the depth of that node in the tree. Luckily, this dependence is typically not too strong, and more importantly, most nodes are concentrated around the same depth of the tree (see, e.g., Fig. 1). In all our experiments in dimensions 100–110, the running time of the enumeration algorithm was directly proportional to the number of nodes in the tree, and the rate of enumeration was very close to $0.94 \cdot 10^7$ nodes per second, or equivalently $0.3 \cdot 10^{15}$ nodes per core-year. This is faster than fplll's [6] implementation (which is itself an improvement over NTL [34]) by about 40%. This is due to a code optimization described in App. B.

Table 1. Pruning vs. full enumeration

dim	90	100	110	120
Full enumeration	$1.2 \cdot 10^{17}$	$1.6 \cdot 10^{20}$	$4.0 \cdot 10^{23}$	$1.9 \cdot 10^{27}$
Schnorr-Hörner	$2.3 \cdot 10^{12}$	$7.4 \cdot 10^{14}$	$4.7 \cdot 10^{17}$	$3.8 \cdot 10^{20}$
Linear pruning	$1.1 \cdot 10^{11}$	$2.6 \cdot 10^{13}$	$1.0 \cdot 10^{16}$	$8.3 \cdot 10^{18}$
Extreme pruning	n/a	$7.7 \cdot 10^{11}$	$2.5 \cdot 10^{13}$	n/a

Estimated running times. Table 1 compares the expected total number of nodes to be scanned under four different bounding functions for CJLOSS lattices in dimensions 90–120; this number of nodes is obtained by multiplying the expected number of nodes in the pruned tree (as estimated by the Gaussian heuristic) by $1/p_{\mathrm{succ}}$. Recall that $0.3 \cdot 10^{15}$ nodes represent one year of sequential computation. The first row corresponds to full enumeration under BKZ-35 reduced bases. The second row corresponds to Schnorr-Hörner's pruning under BKZ-35 reduced bases with an optimal choice of parameter p so that the success probability is still greater than 90% (this corresponds to $p = 48, 57, 67, 77$ for dimensions $90, 100, 110$, and 120 respectively). The third row corresponds to extreme pruning using the linear bounding function under BKZ-35 reduced bases. And the last row corresponds to extreme pruning using our best numerically optimized bounding function (which we computed only for dimensions 100 and 110) and the optimum BKZ-30 (resp. BKZ-32) in dim 100 (resp. 110).

Here we are only considering the number of nodes scanned, and ignoring the basis reduction time. Except our numerically optimized bounding function, it is negligible. For our best bounding function it adds about 50% to the total running time. Another caveat is that the number of nodes in full enumeration (as well as in Schnorr-Hörner pruning and linear pruning) decreases as we increase the block size of the reduction algorithm beyond 35. However, this does not decrease the overall running time by much since the running time of reduction algorithms depends exponentially on the block size.

Actual running time. The actual running times match the predictions well. In practice, extreme pruning is able to find the shortest vector of 0.94-density CJLOSS lattices of dimension 110 in less than 63 CPU days (including reduction time) of sequential computation on a single core, with a negligible amount of memory and in an easily parallelizable manner. With plain Schnorr-Euchner enumeration [32] on a BKZ-35 reduced basis, it would have taken $1.38 \cdot 10^9$ (resp. 482000) CPU years, so the speedup is about $8.1 \cdot 10^9$ (resp. $1.0 \cdot 10^8$).

6.1 Verifying the Heuristics

In this section we report on some experiments meant to verify our heuristic assumptions.

The accuracy of the Gaussian heuristic. Although the Gaussian heuristic was already suggested as a useful heuristic for analyzing the running time of enumeration algorithms (see [14]), we are not aware of any published experimental verification of the heuristic. We therefore ran a significant number of experiments comparing the actual number of nodes in the enumeration tree to the prediction given by the Gaussian heuristic. We tried both CJLOSS lattices and random lattices with several different reduction algorithms and with either full enumeration or pruned enumeration. In all cases, the estimates given by the Gaussian heuristic were very precise, and typically matched the exact count of nodes to within an error of at most 5%.

Randomness of reduced bases. The success of our experiments gives some evidence to the validity of Heuristics 2 and 3. In the full version, we report on additional experiments whose goal is to validate these heuristics directly. We note that the heuristics cannot apply to very strong reduction notions, where the first basis vector is with very high probability the shortest lattice vector: in such cases, there is no need for enumeration since the shortest vector is already provided to us. But it seems to apply to weaker yet still strong reduction notions, such as BKZ-30 in dimension 100.

Acknowledgements. Part of this work was supported by the Commission of the European Communities through the ICT program under contract ICT-2007-216676 ECRYPT II, by the Binational Science Foundation, by the Israel Science Foundation, by the European Commission under the Integrated Project QAP funded by the IST directorate as Contract Number 015848, by the Wolfson Family Charitable Trust, and by a European Research Council (ERC) Starting Grant.

References

1. Agrell, E., Eriksson, T., Vardy, A., Zeger, K.: Closest point search in lattices. IEEE Trans. on Info. Theory 48(8), 2201–2214 (2002)
2. Aharonov, D., Regev, O.: Lattice problems in NP intersect coNP. Journal of the ACM 52(5), 749–765 (2005); Preliminary version in FOCS 2004
3. Ajtai, M.: Generating random lattices according to the invariant distribution (Draft) (March 2006)
4. Ajtai, M., Kumar, R., Sivakumar, D.: A sieve algorithm for the shortest lattice vector problem. In: Proc. 33rd ACM Symp. on Theory of Computing (STOC), pp. 601–610 (2001)
5. Ajtai, M., Kumar, R., Sivakumar, D.: Sampling short lattice vectors and the closest lattice vector problem. In: Proc. of 17th IEEE Annual Conference on Computational Complexity (CCC), pp. 53–57 (2002)
6. Cadé, D., Pujol, X., Stehlé, D.: FPLLL library, version 3.0 (September 2008), `http://perso.ens-lyon.fr/damien.stehle`
7. Coster, M., Joux, A., LaMacchina, B., Odlyzko, A., Schnorr, C., Stern, J.: An improved low-density subset sum algorithm. In: Computational Complexity (1992)
8. Dasgupta, S., Gupta, A.: An elementary proof of the Johnson-Lindenstrauss lemma. Technical report, ICSI, Berkeley, TR-99-006 (1999)
9. Devroye, L.: Non-uniform random variate generation (1986), `http://cg.scs.carleton.ca/~luc/rnbookindex.html`
10. Dyer, M., Frieze, A., Kannan, R.: A random polynomial-time algorithm for approximating the volume of convex bodies. J. ACM 38(1), 1–17 (1991)
11. Fincke, U., Pohst, M.: Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. Mathematics of Computation 44(170), 463–471 (1985)
12. Gama, N., Nguyen, P.Q.: Finding short lattice vectors within Mordell's inequality. In: Proc. 40th ACM Symp. on Theory of Computing, STOC (2008)
13. Gama, N., Nguyen, P.Q.: Predicting Lattice Reduction. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 31–51. Springer, Heidelberg (2008)

14. Hanrot, G., Stehlé, D.: Improved analysis of Kannan's shortest lattice vector algorithm (extended abstract). In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 170–186. Springer, Heidelberg (2007)
15. Håstad, J., Just, B., Lagarias, J.C., Schnorr, C.-P.: Polynomial time algorithms for finding integer relations among real numbers. SIAM J. Comput. 18(5) (1989)
16. Kannan, R.: Improved algorithms for integer programming and related lattice problems. In: Proc. 15th ACM Symp. on Theory of Computing, STOC (1983)
17. Lenstra, A.K., Lenstra Jr., H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische Ann. 261, 513–534 (1982)
18. Lovász, L., Vempala, S.: Simulated annealing in convex bodies and an $O^*(n^4)$ volume algorithm. J. Comput. Syst. Sci. 72(2), 392–417 (2006)
19. Mazo, J.E., Odlyzko, A.M.: Lattice points in high dimensional spheres. Monatsheft Mathematik 17, 47–61 (1990)
20. Micciancio, D., Voulgaris, P.: A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations. In: Proc. 42nd ACM Symp. on Theory of Computing, STOC (2010)
21. Micciancio, D., Voulgaris, P.: Faster exponential time algorithms for the shortest vector problem. In: Proc. ACM-SIAM Symp. on Discrete Algorithms (SODA), pp. 1468–1480 (2010)
22. Nguyen, P.Q.: Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from Crypto 1997. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 288–304. Springer, Heidelberg (1999)
23. Nguyen, P.Q.: Public-key cryptanalysis. In: Luengo, I. (ed.) Recent Trends in Cryptography. Contemporary Mathematics, vol. 477. AMS–RSME (2009)
24. Nguyen, P.Q., Stehlé, D.: LLL on the average. In: Hess, F., Pauli, S., Pohst, M. (eds.) ANTS 2006. LNCS, vol. 4076, pp. 238–256. Springer, Heidelberg (2006)
25. Nguyen, P.Q., Vallée, B. (eds.): The LLL Algorithm: Survey and Applications. Information Security and Cryptography. Springer, Heidelberg (2009)
26. Nguyen, P.Q., Vidick, T.: Sieve algorithms for the shortest vector problem are practical. J. of Mathematical Cryptology 2(2), 181–207 (2008)
27. Pohst, M.: On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications. SIGSAM Bull. 15(1), 37–44 (1981)
28. Pujol, X., Stehlé, D.: Rigorous and efficient short lattice vectors enumeration. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 390–405. Springer, Heidelberg (2008)
29. Pujol, X., Stehlé, D.: Solving the shortest lattice vector problem in time $2^{2.465n}$, IACR eprint report number 2009/605 (2009)
30. Schnorr, C.-P.: Average time fast SVP and CVP algorithms for low density lattices. TR Goethe Universität Frankfurt (January 4, 2010)
31. Schnorr, C.-P.: A hierarchy of polynomial lattice basis reduction algorithms. Theoretical Computer Science 53(2-3), 201–224 (1987)
32. Schnorr, C.-P., Euchner, M.: Lattice basis reduction: improved practical algorithms and solving subset sum problems. Math. Programming 66, 181–199 (1994)
33. Schnorr, C.-P., Hörner, H.H.: Attacking the Chor-Rivest cryptosystem by improved lattice reduction. In: Guillou, L.C., Quisquater, J.-J. (eds.) EUROCRYPT 1995. LNCS, vol. 921, pp. 1–12. Springer, Heidelberg (1995)
34. Shoup, V.: Number Theory C++ Library (NTL) version 5.4.1, http://www.shoup.net/ntl/
35. Stehlé, D., Watkins, M.: On the extremality of an 80-dimensional lattice (2010) (manuscript)

A Schnorr-Hörner Pruning

Here we revisit the pruning strategy described by Schnorr and Hörner in [33], and analyze it in our framework. Their pruning strategy is implemented in NTL [34] as a subroutine to BKZ [32]. It turns out that their bounding function suffers from some fundamental flaws and is clearly inferior to our proposed bounding functions. Furthermore, we show that the analysis of [33] is not satisfying.

A.1 Description

In more detail, given a basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$, the Schnorr-Hörner pruning strategy is defined by the following bounding function, which is parameterized by an integer $p > 0$,

$$\begin{aligned} R_k^2 &= R^2 - \left(2^{-p}\,\mathrm{vol}(L(\mathbf{b}_1, \ldots, \mathbf{b}_{n-k}))/V_{n-k}\right)^{2/(n-k)} \\ &= R^2 - \frac{1}{\pi}\left(2^{-p}\,\mathrm{vol}(L(\mathbf{b}_1, \ldots, \mathbf{b}_{n-k}))\,\Gamma((n-k)/2+1)\right)^{2/(n-k)}, \end{aligned} \tag{9}$$

where $V_{n-k} = V_{n-k}(1)$ is the volume of the unit ball in $n-k$ dimensions. Note that [33] used a different description than the one we gave, but both descriptions can be shown to be equivalent.

In the full version, we rigorously analyze this pruning strategy, and show that it is inferior to our extreme pruning (see also Table 1). Here, we briefly mention several disadvantages. First, the fact that the bounding function depends on a parameter p is undesirable; the analysis of [33] does not give any clear indication on the optimal choice of p. Second, the bounding function may not be positive when p is too small, in which case failure is certain. Third, even for larger values of p, the bounding function may initially decrease, in which case some nodes enumerated in the top levels of the tree are guaranteed to lead to a dead end. In other words, by replacing their bounding function $R_1^2, \ldots, R_n^2$ with the bounding function defined by ${R'}_k^2 = \min(R_k^2, \ldots, R_n^2)$, we obtain exactly the same success probability at a lower running time.

A.2 The Analysis of Schnorr and Hörner

We now present some of our observations regarding the original analysis given by Schnorr and Hörner [33, Sect. 3] and why we believe it is flawed. It should be stressed that the analysis presented there is quite terse, and there is a possibility that our interpretation of it is not what the authors had in mind; yet we think that there is sufficient evidence to suggest that the use of their pruning function should be avoided and we feel that it is important to bring this to the community's attention.

The core of their analysis is [33, Thm. 2], which states that if $x_{n+2-k}, \ldots, x_n \in \mathbb{Z}$ are fixed, and if $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is a random basis of L such that its Gram-Schmidt coefficients $\mu_{i,j}$ are independent and uniformly distributed modulo 1, then the vector $\mathbf{t} = x_{n+2-k}\mathbf{b}_{n+2-k} + \cdots + x_n\mathbf{b}_n$ is such that $\mathbf{t} - \pi_{n+2-k}(\mathbf{t})$ is uniformly distributed modulo the lattice $\bar{L}$ spanned by $\mathbf{b}_1, \ldots, \mathbf{b}_{n+1-k}$, which implies, by [33, Lemma 1], that the expectation E of the number of $\underline{(x_1, \ldots, x_{n+1-k})} \in \mathbb{Z}^{n+1-k}$ such that $\|x_1\mathbf{b}_1 + \cdots + x_n\mathbf{b}_n\| \leq R$ is $V_{n+1-k}(\sqrt{R^2 - \|\pi_{n+2-k}(\mathbf{t})\|^2})/\mathrm{vol}(\bar{L})$. And [33] seems to interpret E as the expectation of the number of leaves (in the enumeration tree) which derive from the node $\pi_{n+2-k}(\mathbf{t})$ at depth $k-1$.

Then, [33] claims that Thm. 2 implies that the probability that the pruned enumeration misses the shortest vector is at most $2^{-p}c$ where c is said to be experimentally proportional to $c_{p,n}2^p$ for random LLL-reduced bases, where $c_{p,n}$ decreases to 0 as p increases. The failure probability is claimed to be experimentally ≤ 0.9 for $n < 30$ and $p = 7$. Finally, [33] claims that under heuristic arguments, they can show that for $p > \log_2 n$: given a random basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of L and a bound $R \leq \|\mathbf{b}_1\|$, their pruned enumeration performs on the average only $O(n^2 2^p)$ arithmetic steps to output a lattice vector $\mathbf{v} \in L$ such that $\|\mathbf{v}\| \leq R$ if $R \geq \lambda_1(L)$, or nothing if $R < \lambda_1(L)$. However, no proof is provided, and the claim looks suspicious: indeed, by taking $p = \lceil \log_2 n \rceil$, it would imply polynomial-time algorithms for the shortest vector problem (which is NP-hard under randomized reductions), because $n^2 2^p$ is polynomial in n.

There are further problems in the analysis of [33]. First of all, the assumption in Thm. 2 that the $\mu_{i,j}$'s of a random reduced basis are uniformly distributed is not supported by experiments: the experiments of [24] show that the distribution of the coefficients $\mu_{i,i-1}$ of a random LLL-reduced basis is far from being uniform. More importantly, the use of Thm. 2 in [33] to analyze the success probability of pruned enumeration is improper: if one selects $(x_{n+2-k}, \ldots, x_n)$ as the last $k-1$ coordinates of a fixed lattice vector, then these coordinates depend on the random basis, and therefore Thm. 2 cannot be applied. Similarly, the expectation E cannot be viewed as the number of leaves in the enumeration tree which derive from the node $\pi_{n+2-k}(\mathbf{t})$ at depth $k-1$, because when the random basis varies, the tree varies too, so for any choice of $x_{n+2-k}, \ldots, x_n \in \mathbb{Z}$, the node $\pi_{n+2-k}(\mathbf{t})$ may appear in one tree, but not in other trees, so this expectation of the number of leaves cannot be properly defined.

B Pseudo-code of the Pruned Enumeration Code

Here we provide the pseudo-code used to implement the enumeration algorithm in our experiments (see Algorithm 2): a detailed explanation appears in the full version. The code is based on the original Schnorr-Euchner enumeration algorithm [32], with several minor modifications. The first easy modification (see Line 10) is that we support a general bounding function $R_1 \leq \cdots \leq R_n$. The second modification (see Lines 1, 15-17, and 25) is a certain optimization that seems to give in practice a speedup by about 40%. To the best of our knowledge, this improvement has not appeared yet in the literature nor in any software package. Finally, another very minor modification is that we abort the procedure as soon as a vector shorter than $R = R_n$ is found.

Algorithm 2. Pruned Enumeration

```
Input: A basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$
    A bounding function $R_1^2 \leq \dots \leq R_n^2$
    The Gram-Schmidt coefficient matrix $\mu$ (a lower-triangular matrix with ones on the
    diagonal), together with the norms of the Gram-Schmidt vectors $\|\mathbf{b}_1^*\|^2, \dots, \|\mathbf{b}_n^*\|^2$.
Output: The coefficients of a lattice vector satisfying the bounds (if it exists)
$\sigma \leftarrow (0)_{(n+1)\times n}$; $r_0 = 0; r_1 = 1; \cdots ; r_n = n$
$\rho_1 = \rho_2 = \cdots = \rho_{n+1} = 0$;  // partial norm
$v_1 = 1; v_2 = \cdots = v_n = 0$  // current combination
$c_1 = \cdots = c_n = 0$  // centers
$w_1 = \cdots = w_n = 0$  // jumps
last_nonzero = 1;  // largest $i$ for which $v_i \neq 0$; zero if all $v_i = 0$
$k = 1$;
while true do
    $\rho_k = \rho_{k+1} + (v_k - c_k)^2 \cdot \|\mathbf{b}_k^*\|^2$  // compute norm squared of current node
    if $\rho_k \leq R_{n+1-k}^2$ (we are below the bound) then
        if $k = 1$ then
            return $(v_1, \dots, v_n)$; (solution found; program ends)
        else
            $k \leftarrow k - 1$ // going down the tree
            $r_{k-1} \leftarrow \max(r_{k-1}, r_k)$ // to maintain the invariant for $j < k$
            for $i = r_k$ downto $k + 1$ do $\sigma_{i,k} \leftarrow \sigma_{i+1,k} + v_i \mu_{i,k}$ endfor
            $c_k \leftarrow -\sigma_{k+1,k}$  // $c_k \leftarrow -\sum_{i=k+1}^{n} v_i \mu_{i,k}$
            $v_k \leftarrow \lfloor c_k \rceil$; $w_k = 1$
        end if
    else
        $k \leftarrow k + 1$ // going up the tree
        if $k = n + 1$ then
            return $\emptyset$ (there is no solution)
        end if
        $r_{k-1} \leftarrow k$ // since $v_k$ is about to change, indicate that $(i, j)$ for $j < k$ and
        $i \leq k$ are not synchronized
        // update $v_k$
        if $k \geq$ last_nonzero then
            last_nonzero $\leftarrow k$
            $v_k \leftarrow v_k + 1$;
        else
            if $v_k > c_k$ then $v_k \leftarrow v_k - w_k$ else $v_k \leftarrow v_k + w_k$
            $w_k \leftarrow w_k + 1$
        end if
    end if
end while
```

Algebraic Cryptanalysis of McEliece Variants with Compact Keys

Jean-Charles Faugère[1], Ayoub Otmani[2,3],
Ludovic Perret[1], and Jean-Pierre Tillich[2]

[1] SALSA Project - INRIA (Centre Paris-Rocquencourt)
UPMC, Univ Paris 06 - CNRS, UMR 7606, LIP6
104, avenue du Président Kennedy 75016 Paris, France
jean-charles.faugere@inria.fr, ludovic.perret@lip6.fr
[2] SECRET Project - INRIA Rocquencourt
Domaine de Voluceau, B.P. 105 78153 Le Chesnay Cedex, France
ayoub.otmani@inria.fr, jean-pierre.tillich@inria.fr
[3] GREYC - Université de Caen - Ensicaen
Boulevard Maréchal Juin, 14050 Caen Cedex, France

Abstract. In this paper we propose a new approach to investigate the security of the McEliece cryptosystem. We recall that this cryptosystem relies on the use of error-correcting codes. Since its invention thirty years ago, no efficient attack had been devised that managed to recover the private key. We prove that the private key of the cryptosystem satisfies a system of bi-homogeneous polynomial equations. This property is due to the particular class of codes considered which are alternant codes. We have used these highly structured algebraic equations to mount an efficient key-recovery attack against two recent variants of the McEliece cryptosystems that aim at reducing public key sizes. These two compact variants of McEliece managed to propose keys with less than 20,000 bits. To do so, they proposed to use quasi-cyclic or dyadic structures. An implementation of our algebraic attack in the computer algebra system Magma allows to find the secret-key in a negligible time (less than one second) for almost all the proposed challenges. For instance, a private key designed for a 256-bit security has been found in 0.06 seconds with about $2^{17.8}$ operations.

1 Introduction

Alternative cryptography. Despite the fact that several hard problems have been proposed as a foundation for public-key primitives, those effectively used are essentially classical problems coming from number theory: integer factorization (e.g. in RSA) and discrete logarithm (e.g. in Diffie-Hellman key-exchange). It is well-known that, although polynomial-time algorithms for those problems have not yet been found, they are not safe from a theoretic breakthrough that would endanger the security of the corresponding schemes. For instance, the emergence

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 279–298, 2010.

of a new computer model such as quantum computers would make schemes based on these classical number theory problems totally insecure.

The lack of diversity in public key cryptography has been identified as a major concern in the field of information security. A good illustration of the potential damage of such lack of diversity is hash zoo. The portfolio of hash functions used so far in practice was mainly restricted to the same type of functions which are now almost all broken. Although the American National Institute of Standards and Technology (NIST) issued an international call[1] to design a new standard hash function, the cryptography community will remain in a fuzzy situation until 2012 (date of final decision).

One of the main issues in public key cryptography is to identify hard problems that are not based on the classical ones coming from number theory. However, few emerged until now as a viable alternative. As pointed in [2], promising candidates include: the problem of solving multivariate equations over a finite field, the problem of finding a short vector in a lattice and the problem of decoding a linear code. Those problems known for being NP-hard are not concerned with the quantum computer threat.

McEliece cryptosystem. Among those problems, code-based cryptosystems seem to offer the most promising alternative. McEliece public key cryptosystem [25] is one of the oldest public-key cryptosystems. Its security relies on the difficulty of decoding a linear code. The main advantage of this system is to have very fast encryption and decryption functions. Depending on how the parameters are chosen for a fixed security level, this cryptosystem is about five times faster for encryption and about 10 to 100 times faster for decryption than RSA [10]. Furthermore, it has withstood many attacking attempts. After more than thirty years now, it still belongs to the very few public key cryptosystems which remain unbroken.

Following McEliece's pioneering work, several different public key cryptosystems based on the intractability of decoding a linear code have been proposed [28,20,31,23,7,6,4,3,5,27]. The original McEliece cryptosystem relies on Goppa codes whereas its variants suggested to use different codes. The Sidelnikov system [31] used Reed-Muller codes, the Janwa-Moreno system proposed to take algebraic geometric codes [23] and the Gabidulin-Paramonov-Tretjakov (GPT) cryptosystem considered Gabidulin codes devised for the rank-metric. LDPC codes have also been repeatedly suggested for this use. Niederreiter is the first in [28] to bring in a significant modification of the McEliece system. However his suggestion to use Generalized Reed-Solomon codes turned out to be an insecure solution [32]. Many of these schemes were broken [32,22,26,29,18,30,35]. All these attacks result in a total break of the system (the secret key, or an equivalent secret key is recovered from the knowledge of the public key). However, the original McEliece remains unbroken. The fact that the best known attacks are still exponential speaks for itself [33,8,9,19].

[1] http://csrc.nist.gov/groups/ST/hash/sha-3/index.html

Despite its impressive resistance against a variety of attacks and its fast encryption and decryption, McEliece cryptosystem has not stood up to RSA for practical applications. This is most likely due to the large size of the public key which is between several hundred thousand and several million bits. To overcome this limitation, a new trend initiated in [21] manages to decrease the key size by choosing the public matrix defining the code in a particular form; for instance with a quasi-cyclic structure [21]. This enables to decrease significantly the key size. The same idea was used in [4] with LDPC codes. Both schemes were broken in [29]. It should be noted that both proposals did not use the Goppa codes of the McEliece cryptosystem, and the attacks have no impact on its security.

This work was then followed by two independent proposals [5,27] that are based on the same kind of idea of using quasi-cyclic [5] or dyadic structure [27]. Both use the same type of codes called the alternant code family which contains Goppa codes. Actually the codes used in [27] are Goppa codes. This approach is quite attractive because it results in a drastic improvement of the public key size. In [5], the size ranges between $8,000$ and $20,000$ bits, whereas it lies between $4,000$ and $20,000$ bits in [27]. Until now, these new proposals seem to be immune against the attack suggested in [29].

Our contribution. In this paper we show that both schemes have a serious flaw that can be exploited to recover the private keys. We present an *algebraic cryptanalysis*[2] of the quasi-cyclic and dyadic schemes [5,27]. Algebraic cryptanalysis is a general framework that permits to assess the security of theoretically all cryptographic schemes. So far, such type of attacks has been applied successfully against several multivariate schemes and stream ciphers. To our knowledge, it is the first time that such an approach is used against code-based cryptosystems. The basic principle of this cryptanalysis is to associate to a cryptographic primitive a set of algebraic equations. The system of equations is constructed in such a way to have a correspondence between the solutions of this system, and a secret information of the cryptographic primitive (for instance the secret key of an encryption scheme). In McEliece, the algebraic system that we have to solve for recovering the secret-key has the following very specific structure:

$$\left\{ g_{i,0}Y_0X_0^j + \cdots + g_{i,n-1}Y_{n-1}X_{n-1}^j = 0 \;\middle|\; i \in \{0,\ldots,k-1\}, j \in \{0,\ldots,r-1\} \right\} \quad (1)$$

where the unknowns are the X_i's and the Y_i's and the $g_{i,j}$'s are known coefficients with $0 \leq i \leq k-1$, $0 \leq j \leq n-1$ that belong to a certain field $\mathbb{F}_q$ with $q = 2^s$. We look for solutions of this system in a certain extension field $\mathbb{F}_{q^m}$. Here k is an integer which is at least equal to $n - rm$. By denoting $\mathbf{X} \overset{\text{def}}{=} (X_0,\ldots,X_{n-1})$ and $\mathbf{Y} \overset{\text{def}}{=} (Y_0,\ldots,Y_{n-1})$ we will refer to such an algebraic system by $\mathsf{McE}_{k,n,r}(\mathbf{X},\mathbf{Y})$. The total number of equations is rk. The number of unknowns $2n$ and the maximum degree $r-1$ of the equations can be extremely high when cryptographic

[2] An independent and parallel work [34] took place that also proposed a cryptanalysis of these two schemes.

parameters are considered (e.g. $n = 1024$ and $r - 1 = 49$). Thus it is not clear whether an algebraic attack can be mounted efficiently in general.

However, in the case of the tweaked McEliece schemes we have either quasi-cyclic or dyadic [5,27]. It turns out that is possible to make use of this structure in order to reduce considerably the number of unknowns in the algebraic system (1). Moreover, it also appears that by using only the linear equations involving the Y_i's gives a linear space of solutions which is of small dimension. We will explain in Section 4 and Section 5 respectively how to solve the underlying algebraic systems. We will see how the public-key structure (quasi-cyclic or dyadic) impacts on the difficulty of solving the algebraic system (1). In particular, the structure induces an imbalance between the X and Y variables. From a practical point of view, we have been able to recover the secret-key via Gröbner bases computations in a negligible time (less than one second) for most of the parameters proposed in [5,27]. Before that, we briefly recall in the next section the McEliece scheme and we explain in Section 3 how we derive the algebraic system (1).

2 McEliece Public-Key Cryptosystem

We recall here how the McEliece public-key cryptosystem is defined.

Secret key: The triplet $(\boldsymbol{S}, \boldsymbol{G_s}, \boldsymbol{P})$ of matrices defined over a finite field $\mathbb{F}_q$ over q elements, with q being a power of two, that is $q = 2^s$. $\boldsymbol{G_s}$ is a full rank matrix of size $k \times n$, with $k < n$, $\boldsymbol{S}$ is of size $k \times k$ and is invertible, and $\boldsymbol{P}$ is permutation matrix of size $n \times n$. Moreover $\boldsymbol{G_s}$ defines a code (which is the set of all possible $\boldsymbol{uG_s}$ with $\boldsymbol{u}$ ranging over $\mathbb{F}_q^k$) which has a decoding algorithm which can correct in polynomial time a set of errors of weight at most t. This means that it can recover in polynomial time $\boldsymbol{u}$ from the knowledge of $\boldsymbol{uG_s} + \boldsymbol{e}$ for all possible $\boldsymbol{e} \in \mathbb{F}_q^n$ of Hamming weight at most t.

Public key: The matrix product $\boldsymbol{G} = \boldsymbol{SG_sP}$.

Encryption: A plaintext $\boldsymbol{u} \in \mathbb{F}_q^k$ is encrypted by choosing a random vector $\boldsymbol{e}$ in $\mathbb{F}_q^n$ of weight at most t. The corresponding ciphertext is $\boldsymbol{c} = \boldsymbol{uG} + \boldsymbol{e}$.

Decryption: $\boldsymbol{c}' = \boldsymbol{cP}^{-1}$ is computed from the ciphertext $\boldsymbol{c}$. Notice that $\boldsymbol{c}' = (\boldsymbol{uSG_sP}+e)\boldsymbol{P}^{-1} = \boldsymbol{uSG_s}+\boldsymbol{eP}^{-1}$ and that $\boldsymbol{eP}^{-1}$ is of Hamming weight at most t. Therefore the aforementioned decoding algorithm can recover in polynomial time $\boldsymbol{uS}$. This vector is multiplied by $\boldsymbol{S}^{-1}$ to obtain the plaintext $\boldsymbol{u}$.

This describes the general scheme suggested by McEliece. From now on, we say that $\boldsymbol{G}$ is the *public generator matrix* and the vector space $\mathscr{C}$ spanned by its rows is the *public code i.e.* $\mathscr{C} \stackrel{\text{def}}{=} \left\{\boldsymbol{uG} \mid \boldsymbol{u} \in \mathbb{F}_q^k\right\}$. What is generally referred to as the McEliece cryptosystem is this scheme together with a particular choice of the code, which consists in taking a binary Goppa code. This class of codes belongs to a more general class of codes, namely the alternant code family ([24, Chap. 12, p. 365]). The main feature of this last class of codes is the fact that they can be decoded in polynomial time.

3 Algebraic Approach

Setting up the algebraic system. We explain more precisely how we construct the algebraic system described in (1). As explained in the previous section, the McEliece cryptosystem relies on Goppa codes which belong to the class of *alternant codes* and inherit from this an efficient decoding algorithm. We will describe this class of codes in more details since both cryptosystems that we cryptanalyze use such codes. It is convenient to describe them through a *parity-check matrix.* This is an $r \times n$ matrix $\boldsymbol{H}$ defined over an extension $\mathbb{F}_{q^m}$ of the field over which the code is defined, which is such that

$$\{\boldsymbol{u}\boldsymbol{G_s} \mid \boldsymbol{u} \in \mathbb{F}_q^k\} = \{\boldsymbol{c} \in \mathbb{F}_q^n \mid \boldsymbol{H}\boldsymbol{c}^T = 0\}. \tag{2}$$

r satisfies in this case the condition $r \geq \frac{n-k}{m}$. In the case of alternant codes, there exists a parity-check matrix with a very special form related to Vandermonde matrices. More precisely there exist two vectors $\boldsymbol{x} = (x_0, \ldots, x_{n-1})$ and $\boldsymbol{y} = (y_0, \ldots, y_{n-1})$ in $\mathbb{F}_{q^m}^n$ such that $\boldsymbol{V}_r(\boldsymbol{x}, \boldsymbol{y})$ is a parity-check matrix , with

$$\boldsymbol{V}_r(\boldsymbol{x}, \boldsymbol{y}) \overset{\text{def}}{=} \begin{pmatrix} y_0 & \cdots & y_{n-1} \\ y_0 x_0 & \cdots & y_{n-1} x_{n-1} \\ \vdots & & \vdots \\ y_0 x_0^{r-1} & \cdots & y_{n-1} x_{n-1}^{r-1} \end{pmatrix}. \tag{3}$$

We use the following notation in what follows

Definition 1. *The alternant code $\mathscr{A}_r(\boldsymbol{x}, \boldsymbol{y})$ of order r over $\mathbb{F}_q$ associated to $\boldsymbol{x} = (x_0, \ldots, x_{n-1})$ where the x_i's are different elements of $\mathbb{F}_{q^m}$ and $\boldsymbol{y} = (y_0, \ldots, y_{n-1})$ where the y_i's are nonzero elements of $\mathbb{F}_{q^m}$ is defined by*

$$\mathscr{A}_r(\boldsymbol{x}, \boldsymbol{y}) = \{\boldsymbol{c} \in \mathbb{F}_q^n \mid \boldsymbol{V}_r(\boldsymbol{x}, \boldsymbol{y})\boldsymbol{c}^T = 0\}.$$

It should be noted that the public code in the McEliece scheme is also an alternant code. We denote here by the public code, the set of vectors of the form

$$\{\boldsymbol{u}\boldsymbol{G} \mid \boldsymbol{u} \in \mathbb{F}_q^k\} = \{\boldsymbol{c}\boldsymbol{S}\boldsymbol{G_s}\boldsymbol{P} \mid \boldsymbol{c} \in \mathbb{F}_q^k\}.$$

This is simple consequence of the fact that the set $\{\boldsymbol{u}\boldsymbol{S}\boldsymbol{G_s}\boldsymbol{P} \mid \boldsymbol{u} \in \mathbb{F}_q^k\}$ is obtained from the secret code $\{\boldsymbol{u}\boldsymbol{G_s} \mid \boldsymbol{u} \in \mathbb{F}_q^k\}$ by permuting coordinates in it with the help of $\boldsymbol{P}$, since multiplying by an invertible matrix $\boldsymbol{S}$ of size $k \times k$ leaves the code globally invariant. The key feature of an alternant code is the following fact

Fact 1. *There exists a polynomial time algorithm decoding an alternant code once a parity-check matrix $\boldsymbol{H}$ of the form $\boldsymbol{H} = \boldsymbol{V}_r(\boldsymbol{x}, \boldsymbol{y})$ is given.*

In other words, it is possible to break the McEliece scheme, if it is possible to find $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$ in $\mathbb{F}_{q^m}^n$ such that

$$\{\boldsymbol{x}\boldsymbol{G} \mid \boldsymbol{x} \in \mathbb{F}_q^k\} = \{\boldsymbol{y} \in \mathbb{F}_q^n \mid \boldsymbol{V}_r(\boldsymbol{x}^*, \boldsymbol{y}^*)\boldsymbol{y}^T = 0\}. \tag{4}$$

From the knowledge of this matrix $\boldsymbol{V}_r(\boldsymbol{x}^*, \boldsymbol{y}^*)$, it is possible to decode the public code, that is to say to recover $\boldsymbol{u}$ from $\boldsymbol{uG} + \boldsymbol{e}$. Finding such a matrix clearly amounts to find a matrix $\boldsymbol{V}_r(\boldsymbol{x}^*, \boldsymbol{y}^*)$ such that $\boldsymbol{V}_r(\boldsymbol{x}^*, \boldsymbol{y}^*)\boldsymbol{G}^T = \boldsymbol{0}$. By bringing in $2n$ variables $X_0, \ldots, X_{n-1}$ and $Y_0, \ldots, Y_{n-1}$ where X_i corresponds to x_i^* and Y_i to y_i^* we see that this is equivalent to solve the following system:

$$\left\{ g_{i,0} Y_0 X_0^j + \cdots + g_{i,n-1} Y_{n-1} X_{n-1}^j = 0 \;\middle|\; i \in \{0, \ldots, k-1\}, j \in \{0, \ldots, r-1\} \right\} \quad (5)$$

where the $g_{i,j}$'s are the entries of the known matrix $\boldsymbol{G}$ with $0 \le i \le k-1$ and $0 \le j \le r-1$.

The cryptosystems proposed in [5,27] follow the McEliece scheme [25] with the additional goal to design a public-key cryptosystem with very small key sizes. They both require to identify alternant codes having a property that allows matrices to be represented by very few rows. In the case of [5] circulant matrices are chosen whereas the scheme [27] focuses on dyadic matrices. These two families have in common the fact the matrices are completely described from the first row. The public generator matrix $\boldsymbol{G}$ in these schemes is a block matrix where each block is circulant in [5] and dyadic in [27].

We shall see that the algebraic approach previously described leads to a key-recovery in nearly all the parameters proposed in both schemes. The crucial point that makes the attack possible is the fact that we have a system with much less unknowns than in the case of the McEliece cryptosystem. This is due to both the particular structure of the matrices and their block form that describe the public alternant codes. We finally end this section with a simple remark which explains that we can basically set one of the Y_i's and two values of the X_i's to an arbitrary value in the algebraic system (1).

Proposition 1 ([24, Chap. 10, p. 305]). *Let $(X_0, \ldots, X_{n-1})$, $(Y_0, \ldots, Y_{n-1})$ be a solution of (1) and $a \neq 0$, b, $c \neq 0$ be elements of $\mathbb{F}_{q^m}$. Then $(aX_0 + b, \ldots, aX_{n-1} + b)$ and $(cY_0, \ldots, cY_{n-1})$ is also a solution of (1).*

Solving the Algebraic System. We describe now a general technique to solve the algebraic system $\mathsf{McE}_{k,n,r}(\mathbf{X}, \mathbf{Y})$ using Gröbner bases techniques [11,12,13,14]. Although the particular characteristics of the cryptosystems [5,27] studied here will further influence the shape of $\mathsf{McE}_{k,n,r}(\mathbf{X}, \mathbf{Y})$ (number of variables, number of equations, ...), we have designed a special strategy for taking advantage as much as possible of the intrinsic structure. We have made an implementation of the strategy described here. We will present the experimental results, as well as the improvements which are possible due to the quasi-cyclic and dyadic structures, in Section 4 (quasi-cyclic case) and Section 5 (dyadic case).

As a first general remark, it is readily seen that $\mathsf{McE}_{k,n,r}(\mathbf{X}, \mathbf{Y})$ is highly structured. For instance, it is very sparse as the only monomials occurring in the system are of the form $Y_i X_i^j$, with $0 \le i \le k-1$ and $0 \le j \le r-1$. It can also be noticed that each block of k equations is *bi-homogeneous*, *i.e.* homogeneous if the variables of $\mathbf{X}$ (resp. $\mathbf{Y}$) are considered alone. Note that such structure already appears in the cryptanalysis of the MinRank problem [15].

Due to the particular structure of the system, it makes sense to design a specific strategy for solving $\mathsf{McE}_{k,n,r}(\mathbf{X},\mathbf{Y})$. A simple way for solving this system would consist in generating the equations and try to solve it directly with a generic algorithm (for instance, the Gröbner basis algorithm available in the Magma computer algebra software). This approach fails for most challenges proposed in [27]. However, it is interesting to remark that this direct approach has been successful in practice for all challenges of [5]. We only mention that it takes between few minutes to 24 hours of computation using a negligible amount of memory. As a comparaison, the improved strategy that we will describe now permits to solve (almost) all the challenges of [5,27] in few seconds using also a negligible amount of memory.

The first fundamental remark is that there are k linear equations in the n variables of the block $\mathbf{Y}$ in $\mathsf{McE}_{k,n,r}(\mathbf{X},\mathbf{Y})$. This implies that all the variables of the block $\mathbf{Y}$ can be expressed in terms of $d \geq n-k$ variables. From now on, we will always assume that the variables of the block $\mathbf{Y}'$ only refer to these d *free* variables. The first step is then to rewrite the system (1) only in function of the variables of $\mathbf{X}$ and $\mathbf{Y}'$, *i.e.* the variables of $\mathbf{Y} \setminus \mathbf{Y}'$ are substituted by linear combinations involving only variables of $\mathbf{Y}'$. For the cryptosystems considered in this paper [5,27], the number of free variables d in $\mathbf{Y}'$ can be rather small (typically 1 or 2 for some challenges). Furthermore, the quasi-cyclic and dyadic structures provide additional linear equations in the variables of $\mathbf{X}$ and $\mathbf{Y}'$ which can be also used to rewrite/clean the system. In the sequel, we will denote by $\mathsf{McE}_{k,n,r}(\mathbf{X}',\mathbf{Y}')$ the system obtained from $\mathsf{McE}_{k,n,r}(\mathbf{X},\mathbf{Y})$ by removing all the linear equations in $\mathbf{X}$ and $\mathbf{Y}$.

The second crucial point is that as soon as the the projection of the solutions on the variables $\mathbf{Y}'$ are known, the system (1) simplifies to:

$$\left\{ g'_{i,0}X_0^j + \cdots + g'_{i,n-1}X_{n-1}^j = 0 \;\middle|\; i \in \{0,\ldots,k-1\}, j \in \{0,\ldots,r-1\} \right\}.$$

This system is readily solved by keeping only the equations in this system which correspond to powers of the X_i's which are powers of two. In other words we consider only the equations of the form $g'_{i,0}X_0^{2^j} + \cdots + g'_{i,n-1}X_{n-1}^{2^j} = 0$ for j in $\{0,\ldots,\lfloor \log_2(r-1) \rfloor\}$ and i in $\{0,\ldots,k-1\}$. Hence, we obtain a quasi bi-linear system because the system is always defined over a field of characteristic two. Moreover, it has very few monomials per equation and can be easily solved in practice by computing a Gröbner basis.

The most difficult part of the computation is to find a projection of the solutions with respect to the variables of the block $\mathbf{Y}'$. Notice that an exhaustive search on the d free variables of $\mathbf{Y}'$ leads to a practical attack for some of the challenges proposed in [5,27]. We will present below an even more efficient strategy to recover $\mathbf{Y}'$. More formally, let $\mathcal{I}$ be the ideal generated by $\mathsf{McE}_{k,n,r}(\mathbf{X}',\mathbf{Y}')$ and $\mathcal{V}$ be the corresponding variety *i.e.* the set of solutions. The goal is to compute the projection of $\mathcal{V}$, denoted by $\mathcal{V}'$, on the variables of $\mathbf{Y}'$. This can done by computing a Gröbner basis (w.r.t. a degree order) G_{deg} of $\mathcal{I} \cap \mathbb{F}_{q^m}[\mathbf{Y}']$. This is a classical problem in computer algebra which can be solved by using standard elimination techniques (for instance see [1, Chap. 2.3, p. 69] or [12, Chap. 3, p. 112]. In the appendix, we briefly recall basic facts about elimination theory. In our context, we have used a slightly modified version of F_4 [13] for computing a Gröbner basis $G_{\deg}$ of $\mathcal{I} \cap \mathbb{F}_{q^m}[\mathbf{Y}']$. Roughly speaking, the idea is to adapt the algorithm for performing the Gröbner basis computation in $\mathbb{F}_{q^m}[\mathbf{X}'][\mathbf{Y}']$, *i.e.* the set of polynomials in $\mathbf{Y}'$ whose coefficients are polynomials in $\mathbb{F}_{q^m}[\mathbf{X}']$. As for the usual F_4, we process degree by degree. However, we consider only the degree of the polynomials w.r.t. the variables of $\mathbf{X}'$. We stop the computation as soon as we have sufficiently many equations in $\mathbf{Y}'$, in other words, as soon as we detect that $\mathcal{V}'$ has a finite number of solution. Below, we describe more precisely the procedure which allows to compute a Gröbner basis $G_{\deg}$ of $\mathcal{I} \cap \mathbb{F}_{q^m}[\mathbf{Y}']$. As already explained, this is the most difficult part.

Algorithm 1. ComputeProjection

```
**Input :** The system $\mathsf{McE}_{k,n,r}(\mathbf{X}', \mathbf{Y}')$
**Output**: A Gröbner basis $G_{\deg}$ of $\mathcal{I} \cap \mathbb{F}_{q^m}[\mathbf{Y}']$
Let $F$ be the equations of degree $2^i, 1 \leq i \leq r-1$ of $\mathsf{McE}_{k,n,r}(\mathbf{X}', \mathbf{Y}')$
Let $F'$ be the system obtained from $F$ by fixing “randomly” some variables of $\mathbf{X}'$
Compute a Gröbner basis $G_{\deg}$ of $\mathcal{I} \cap \mathbb{F}_{q^m}[\mathbf{Y}']$ using the tweaked version of $\mathrm{F}_4$
**Return** $G_{\deg}$
```

Furthermore, to be sure that the variety $\mathcal{V}'$ associated to $\mathcal{I} \cap \mathbb{F}_{q^m}[\mathbf{Y}']$ has few solutions, we have to remove parasite solutions corresponding to $X_i = X_j$ or to $Y_j = 0$. A classical way to do that is to introduce new variables u_{ij} and v_i and add to $\mathsf{McE}_{k,n,r}(\mathbf{X}', \mathbf{Y}')$ equations of the form:

$$u_{ij} \cdot (X_i - X_j) + 1 = \text{and } v_i \cdot Y_i + 1 = 0.$$

In practice, we have not added all theses equations; but only few of them namely 4 or 5. The reason is that we do not want to add too many new variables. In addition, including few of such equations already permits to remove trivial solutions. We also have to remove some degree of freedom in the algebraic system by fixing randomly few variables of $\mathbf{X}$' as explained in Proposition 1. It is important to notice that since we are removing component of high dimension the new system is indeed much faster to solve.

Finally, we have not considered all the equations of $\mathsf{McE}_{k,n,r}(\mathbf{X}', \mathbf{Y}')$ to compute $G_{\deg}$. This system being naturally over-defined, we can “safely” remove some equations. Typically, it makes sense to consider the smaller subset of equations such that $\mathcal{V}'$ is zero-dimensional and for which we can efficiently compute $G_{\deg}$. The variety $\mathcal{V}'$ having few elements it is not difficult to recover this set from the knowledge of $G_{\deg}$.

4 Algebraic Cryptanalysis of the Quasi-Cyclic Variant

The scheme presented in [5] suggests to use block matrices where each block is a circulant matrix. The public code $\mathscr{C}$ suggested in [5] is defined on a field $\mathbb{F}_q = \mathbb{F}_{2^s}$ which is considered as a subfield of $\mathbb{F}_{q^m}$ for a certain integer m. Let α be a primitive element of $\mathbb{F}_{q^m}$. Let ℓ and N_0 be such that $q^m - 1 = N_0\ell$ and let β be an element of $\mathbb{F}_{q^m}$ of order ℓ. Although this is not explicitly stated in [5], it is readily checked that $\mathscr{C}$ is defined from an $r \times n$ parity-check matrix $\boldsymbol{H}$ over $\mathbb{F}_{q^m}$ which is the juxtaposition of n_0 ($n = \ell n_0$) matrices $\boldsymbol{H}^{(0)}, \cdots, \boldsymbol{H}^{(n_0-1)}$ of size $r \times \ell$. Each $\boldsymbol{H}^{(b)} = (h_{i,j}^{(b)})$ with $0 \leq b \leq n_0 - 1$, $0 \leq i \leq r - 1$ and $0 \leq j \leq \ell - 1$ is given by

$$h_{i,j}^{(b)} = \gamma_b \beta^{(d_b+j)e} \left(\alpha^{w_b} \beta^{d_b+j}\right)^i \tag{6}$$

where γ_b is a nonzero element of $\mathbb{F}_{q^m}$, d_b is an integer of $\{0, \ldots, \ell - 1\}$, e is an integer of $\{0, \ldots, \ell - 1\}$ and the w_b's are distinct integers of $\{0, \ldots, N_0 - 1\}$. From this, it is now clear that $\mathscr{C}$ is an alternant code $\mathscr{A}_r(\boldsymbol{x}, \boldsymbol{y})$ of order r associated to $\boldsymbol{x} = (x_0, \ldots, x_{n-1})$ and $\boldsymbol{y} = (y_0, \ldots, y_{n-1})$ which satisfy for any j in $\{0, \ldots, \ell - 1\}$

$$x_{b\ell+j} = \alpha^{w_b} \beta^{d_b+j} \tag{7}$$

$$y_{b\ell+j} = \gamma_b \beta^{(d_b+j)e}, \tag{8}$$

It can be checked (see [5]) that $\mathscr{C}$ has a public generating matrix $\boldsymbol{G}$ which is block circulant of size $k \times n$ with k of the form $k = k_0\ell$ for some integer k_0 (recall that $k \geq n - rm$).

We present now an algebraic attack against the quasi-cyclic variant proposed in [5] that recovers $\boldsymbol{x}$ and $\boldsymbol{y}$ by setting up an algebraic system of the form $\mathsf{McE}_{k,n,r}(\mathbf{X}, \mathbf{Y})$ from the equation $\boldsymbol{H}\boldsymbol{G}^T = 0$. This would also give a system with $2n$ unknowns. We can obtain a huge reduction of the number of unknowns by using Equations (7) and (8) which induce some linear relations between the x_i's and the y_i's. From these two equations we deduce that

$$x_{b\ell+j} = x_{b\ell} \beta^j \tag{9}$$

$$y_{b\ell+j} = y_{b\ell} \beta^{je}, \tag{10}$$

for any j in $\{0, \ldots, \ell - 1\}$ and j in $\{0, \ldots, n_0 - 1\}$. Furthermore, since in the cases considered in [5], e is small because it lies in the range $\{0, \ldots, \ell - 1\}$ and ℓ is less than 100, we may assume that:

Assumption 2. *The secret integer e such that $0 \leq e \leq \ell - 1$ is known.*

This enables to simplify the description of the system $\mathsf{McE}_{k,n,r}(\mathbf{X}, \mathbf{Y})$. By setting up the unknown X_b for $x_{b\ell}$ and the unknown Y_b for $y_{b\ell}$ we obtain the following algebraic system.

Proposition 2. *Let $\boldsymbol{G} = (g_{i,j})$ be the $k \times n$ public generator matrix with $k = k_0\ell$ and $n = n_0\ell$. For any $0 \leq w \leq r - 1$ and any $0 \leq i \leq k - 1$, the unknowns $X_0, \ldots, X_{n_0-1}$ and $Y_0, \ldots, Y_{n_0-1}$ should satisfy:*

$$\sum_{b=0}^{n_0-1} g'_{i,b,w} Y_b X_b^w = 0 \qquad \textit{where} \;\; g'_{i,b,w} \overset{def}{=} \sum_{j=0}^{\ell-1} g_{i,b\ell+j}\beta^{j(e+w)}. \tag{11}$$

Proof. We observe that

$$\begin{aligned}\sum_{j=0}^{n-1} g_{i,j} y_j x_j^w &= \sum_{b=0}^{n_0-1}\sum_{j=0}^{\ell-1} g_{i,b\ell+j} y_{b\ell+j} x_{b\ell+j}^w = \sum_{b=0}^{n_0-1}\sum_{j=0}^{\ell-1} g_{i,b\ell+j} y_{b\ell} x_{b\ell}^w \beta^{je}\beta^{jw} \\ &= \sum_{b=0}^{n_0-1} y_{b\ell} x_{b\ell}^w \left(\sum_{j=0}^{\ell-1} g_{i,b\ell+j}\beta^{je}\beta^{jw}\right)\end{aligned}$$

By setting X_b for $x_{b\ell}$ and Y_b for $y_{b\ell}$ we obtain the aforementioned system.

Theoretically by Proposition 1, we would be able to fix two variables, say X_0 and X_1, and one variable Y_j, for instance Y_0, to arbitrary values as long as $X_0 \neq X_1$ and $Y_0 \neq 0$. However, if we do it, we then lose the linear relations between the x_i's given in (9). Therefore we can only fix one X_i and one Y_i as stated in Lemma 1 that is straightforward to prove.

Lemma 1. *Let* $(X_0, \ldots, X_{n_0-1})$, $(Y_0, \ldots, Y_{n_0-1})$ *be a solution of (11). Then* $(aX_0, \ldots, aX_{n_0-1})$ *and* $(cY_0, \ldots, cY_{n_0-1})$ *is also a solution of (11) for any* $a \neq 0$ *and* $c \neq 0$ *of* $\mathbb{F}_{q^m}$.

Hence, the total number of unknowns is $2(n_0 - 1)$. Furthermore there are many redundant equations in Proposition 2. This comes from the block circulant form of $\boldsymbol{G}$. From this form we know that $g_{i\ell+u,b\ell+j} = g_{i\ell,b\ell+((j-u) \mod \ell)}$ for all u in $\{0, \ldots, \ell-1\}$ and i in $\{0, \ldots, k_0 - 1\}$. We also have:

$$\begin{aligned}g'_{i\ell+u,b,w} &= \sum_{j=0}^{\ell-1} g_{i\ell+u,b\ell+j}\beta^{j(e+w)} = \sum_{j=0}^{\ell-1} g_{i\ell,b\ell+((j-u) \mod \ell)}\beta^{j(e+w)} \\ &= \sum_{j=0}^{\ell-1} g_{i\ell,b\ell+j}\beta^{j(e+w)}\beta^{u(e+w)} = g'_{i\ell,b,w}\beta^{u(e+w)}\end{aligned}$$

We used the fact $\beta^{\ell(e+w)} = 1$. So for a given i, when u describes $\{0, \ldots, \ell-1\}$, the equations $\sum_{b=0}^{n_0-1} g'_{i\ell+u,b,w} Y_b X_b^w = 0$ are all equivalent. This means that instead of having rk equations we have only $\frac{rk}{\ell} = k_0 r$ algebraic equations.

Proposition 3. *The system (11) has* $(n_0 - 1)$ *unknowns* Y_i *and* $(n_0 - 1)$ *unknowns* X_i. *Furthermore, it has* k_0 *linear equations involving only the* Y_i*'s and* $(r-1)k/\ell = (r-1)k_0$ *polynomial equations involving the unknowns* $Y_i X_i^w$ *with* $w > 1$.

From now on, we will always assume that redundant equations have been removed and the variables X_0 and Y_0 are fixed. Finally, note that there are $d \overset{\text{def}}{=} n_0 - 1 - k_0$ free variables for the Y_i's.

Table 1. Cryptanalysis results for [5] ($m = 2$)

Challenge	q	ℓ	n_0	d	Security [5]	Unknowns	Equations	Time (Operations, Memory)
A_{16}	2^8	51	9	3	80	16	510	0.06 sec ($2^{18.9}$ op, 115 Meg)
B_{16}	2^8	51	10	3	90	18	612	0.03 sec ($2^{17.1}$ op, 116 Meg)
C_{16}	2^8	51	12	3	100	22	816	0.05 sec ($2^{16.2}$ op, 116 Meg)
D_{16}	2^8	51	15	4	120	28	1275	0.02 sec ($2^{14.7}$ op, 113 Meg)
A_{20}	2^{10}	75	6	2	80	10	337	0.05 sec ($2^{15.8}$ op, 115 Meg)
B_{20}	2^{10}	93	6	2	90	10	418	0.05 sec ($2^{17.1}$ op, 115 Meg)
C_{20}	2^{10}	93	8	2	110	14	697	0.02 sec ($2^{14.5}$ op, 115 Meg)
QC$_{600}$	2^8	255	15	3	600	28	6820	0.08 sec ($2^{16.6}$ op, 116 Meg)

We now present experimental results obtained when solving the system described in (11) using the strategy described in Section 3. The experimental results have been obtained with several Xeon bi-processor 3.2 Ghz, with 16 Gb of Ram. The instances of our problem have been generated using the Magma software. We used the Magma version 2.15 for our computations. The F$_5$ [14] algorithm has been implemented in C language in the FGb software. We used this implementation for computing the first Gröbner basis (i.e. which is used in Algorithm 1). All the other computations are performed under Magma including factorizing some univariate polynomials and computing Gröbner bases using the F$_4$ [13] algorithm. The most important observation is that we have been able to solve all the challenges of [5] in a negligible time because the dimension $d = n_0 - 1 - k_0$ of the vector space solution for the Y_i's is very small. We also proposed a challenge QC$_{600}$ for showing the behaviour of our attack for high security levels.

5 Algebraic Cryptanalysis of the Dyadic Variant

The cryptosystem presented in [27] considers particular alternant codes called *quasi-dyadic* Goppa codes. Goppa codes form an important subclass of alternant codes. Goppa codes are defined by means of a polynomial $G(X)$ of degree ℓ with coefficients in $\mathbb{F}_{q^m}$ and for which the sequence $\boldsymbol{x}$ is assumed not to contain any root of $G(X)$. The alternant code defined by the parity-check matrix $\boldsymbol{V}_\ell(\boldsymbol{x},\boldsymbol{y})$ with $y_i = G(x_i)^{-1}$ is called a Goppa code over $\mathbb{F}_q$ and is denoted by $\mathscr{G}(\boldsymbol{x}, G)$. A detailed description of the key generation is given in Appendix B. We only provide important facts that are useful for recovering the private key. We first state an important result that shows that $\boldsymbol{G}$ defines actually an alternant code. The proof is given in Appendix C. The last important fact to know about $\boldsymbol{G}$ is that it is a $k \times n$ matrix over $\mathbb{F}_q$ such that $n = n_0\ell$ and $k \geq n - m\ell$ where n_0, ℓ are given integers.

Proposition 4. *The code defined by the public generator matrix $\boldsymbol{G}$ is an alternant code $\mathscr{A}_\ell(\boldsymbol{x},\boldsymbol{y})$ where for any $0 \leq j \leq n_0 - 1$ and $0 \leq i, i' \leq \ell - 1$, we have the following equations:*

$$\begin{cases} y_{j\ell+i} & = y_{j\ell} \\ x_{j\ell+i} + x_{j\ell} & = x_i + x_0 \\ x_{j\ell+(i\oplus i')} & = x_{j\ell+i} + x_{j\ell+i'} + x_{j\ell} \end{cases} \tag{12}$$

where $\oplus$ is the bitwise exclusive-or on the binary representation of the indices.

The cryptanalysis of the system consists in defining n_0 unknowns $Y_0, \ldots, Y_{n_0-1}$ that play the role of the y_j's and n unknowns $X_0, \ldots, X_n$ that represent the x_j's. We know specify the system of equations that we obtain directly from Proposition 4.

Proposition 5. *For any w, j, i and i' such that $0 \le w \le \ell - 1$, $0 \le j \le n_0 - 1$ and $1 \le i, i' \le \ell - 1$, we have:*

$$\begin{cases} \sum_{j=0}^{n_0-1} Y_j \sum_{l=0}^{\ell-1} g_{i,j\ell+l} X_{j\ell+l}^{w} = 0 \\ \\ X_{j\ell+i} + X_{j\ell} + X_i + X_0 = 0 \\ \\ X_{j\ell+(i\oplus i')} + X_{j\ell+i} + X_{j\ell+i'} + X_{j\ell} = 0 \end{cases} \tag{13}$$

It is possible to simplify System (13) by observing, thanks to the third equation, that actually many variables X_i's can be expressed in function of some few variables, namely X_{2^j} with $0 \le j \le \log_2(\ell - 1)$ and X_b with $0 \le b \le n_0 - 1$.

Corollary 1. *For any $1 \le i \le \ell - 1$, if we write the binary decomposition of $i = \sum_{j=0}^{\log_2(\ell-1)} \eta_j 2^j$ then the following equation holds:*

$$X_i = X_0 + \sum_{j=0}^{\log_2(\ell-1)} \eta_j (X_{2^j} + X_0).$$

We are also able to provide the exact number of unknowns we can fix to arbitrary values.

Lemma 2. *Let $(X_0, \ldots, X_{n-1})$, $(Y_0, \ldots, Y_{n-1})$ be a solution of (13) and $a \neq 0$, b,c $\neq 0$ be elements of $\mathbb{F}_{q^m}$. Then $(aX_0 + b, \ldots, aX_{n-1} + b)$ and $(cY_0, \ldots, cY_{n-1})$ is also a solution of (13).*

Proof. The only fact to prove is that $(X_0 + b, \ldots, X_{n-1} + b)$ is also a solution of the last two equations in (13). It is readily checked since $\mathbb{F}_{q^m}$ is of charateristic two.

We can now completely give the effective number of equations after elimination of redundant equations.

Proposition 6. *The system (13) has $n_0 - 1$ unknowns Y_i and $n_0 - 2 + \log_2(\ell)$ unknowns X_i. Furthermore, it has $n_0 - m$ linear equations involving only the Y_i's, and $(\ell - 1)\ell(n_0 - m)$ polynomial equations involving the unknowns $Y_i X_i^w$ with $w > 1$.*

Proof. The number of variables Y_j is $(n_0 - 1)$ since we can choose $Y_0 = 1$. As for the variables X_j, we observe that they can all be expressed only in function of X_{2^j} and $X_{i\ell}$ with $0 \leq j \leq \log_2(\ell - 1)$ and $0 \leq i \leq n_0 - 1$. So the number of unknowns X_j is $\log_2(\ell - 1) + 1 + n_0 - 2$ since we can fix two different arbitrary values for two variables, say X_0 and X_ℓ (Lemma 2). Using the fact that $\log_2(\ell - 1) = \log_2(\ell) - 1$ since ℓ is a power of 2, we get the claimed number of unknowns. Furthermore, because of the dyadicity of $\boldsymbol{G}$, the equations obtained with $w = 0$ are identical when $\boldsymbol{g}$ describes all the rows of a dyadic block of $\boldsymbol{G}$. This does not appear when $w > 1$. So we have $k/\ell = n_0 - m$ linear equations that involve the Y_i's and $(\ell - 1)k = (\ell - 1)\ell(n_0 - m)$ polynomial equations that contain variables of the form $Y_i X_i^w$ where $w > 1$.

We now present in Table 2 the experimental results we obtained when we solve the system described in (13) using the strategy described in Section 3. As previously, the experimental results have been obtained with several Xeon bi-processor 3.2Ghz, with 16 Gb of Ram. The instances of our problem have been generated using the Magma software. We used the Magma version 2.15 for our computations. The F_5 [14] algorithm has been implemented in C language in the FGb software. We used this implementation for computing the first Gröbner basis (*i.e.* which is used in Algorithm 1). All the other computations are performed under Magma including factorizing some univariate polynomials and computing Gröbner bases using the F_4 algorithm [13]. Table 2 also shows the impact of the degree extension m. Indeed, computation times indicate that the solutions are easy to find if m is small. This phenomenon is directly related to the size of the solution space of the variables Y_i. We have seen in Section 3 that such variables satisfy a system of linear equations. From Proposition 6, the number of linear equations is $k_0 = n_0 - m$ whereas the number of unknowns Y_i is $n_0 - 1$. Thus the dimension of the vector space solution for the Y_i's is $m - 1$. We also give in Table 2 new parameters (Dyadic_{256} and Dyadic_{512}) that are generated by "extrapolating" challenges given in [27]. We observe that this solution space is manageable in practise as long as $m < 16$ because we did not succeed to find an efficient way to solve the challenges of [27] when $m = 16$.

Table 2. Cryptanalysis results for [27]

Challenge	q	m	ℓ	n_0	Security	Unknowns	Equations	Time (Operations, Memory)
Table 2	2^2	8	64	56	128	115	193, 584	1, 776.3 sec ($2^{34.2}$ op, 360 Meg)
Table 2	2^4	4	64	32	128	67	112, 924	0.50 sec ($2^{22.1}$ op, 118 Meg)
Table 2	2^8	2	64	12	128	27	40, 330	0.03 sec ($2^{16.7}$ op, 35 Meg)
Table 3	2^8	2	64	10	102	23	32, 264	0.03 sec ($2^{15.9}$ op, 113 Meg)
Table 3	2^8	2	128	6	136	16	65, 028	0.02 sec ($2^{15.4}$ op, 113 Meg)
Table 3	2^8	2	256	4	168	13	130, 562	0.11 sec ($2^{19.2}$ op, 113 Meg)
Table 5	2^8	2	128	4	80	12	32, 514	0.06 sec ($2^{17.7}$ op, 35 Meg)
Table 5	2^8	2	128	5	112	14	48, 771	0.02 sec ($2^{14.5}$ op, 35 Meg)
Table 5	2^8	2	128	6	128	16	65, 028	0.01 sec ($2^{16.6}$ op, 35 Meg)
Table 5	2^8	2	256	5	192	15	195, 843	0.05 sec ($2^{17.5}$ op, 35 Meg)
Table 5	2^8	2	256	6	256	17	261, 124	0.06 sec ($2^{17.8}$ op, 35 Meg)
Dyadic_{256}	2^4	4	128	32	256	68	455, 196	7.1 sec ($2^{26.1}$ op, 131 Meg)
Dyadic_{512}	2^8	2	512	6	512	18	1, 046, 532	0.15 sec ($2^{19.7}$ op, 38 Meg)

6 Conclusion

We described in this paper a new algebraic approach to assess the security of the McEliece cryptosystem. We showed that the private key of this scheme is a solution of a very structured system of bi-homogeneous polynomial equations in two sets of unknowns Y_i and X_i. The solutions belong to a finite field $\mathbb{F}_{q^m}$ whereas the coefficients of the system are in $\mathbb{F}_q$ for some known integers m and q. This system comes from the particular structure of alternant codes used in McEliece. Indeed, the Goppa codes as proposed in [25] form a subfamily of alternant codes. Furthermore, the system is composed of two parts of equations: one part that consists of linear equations that involve only the unknowns Y_i and a second part where the equations involve terms of the form $Y_i X_i^j$.

We applied this approach to two new cryptosystems [27,5] that are variants of the McEliece scheme. Both aim at reducing the public keys by using very structured block matrices (cyclic matrices in [5] and dyadic matrices in [27]). We show that our new cryptanalytic point of view is very efficient for all the parameters proposed in [5]. An implementation in Magma validates our attack and shows that the private key can be found in a negligible time. For the scheme [27], we are also able to fully recover the private key in almost all cases. An implementation in Magma shows that this can be done in time comparable to [5] as long as the dimension m of the solution vector space of the Y_i's is small.

Thanks to a very recent development [16] on the solving of bihomogeneous bilinear systems, it is very likely that the solving technique presented here can be replaced by a new version of F_5 dedicated to bi-linear systems. In our case, we can obtain a (quasi) bilinear system when we consider the equations involving terms of the form $Y_i X_i^{2^j}$. Moreover, this will permit to precisely estimate the complexity of the attack presented in this paper and will provide a concrete criteria to evaluate the security of future compact McEliece's variants.

Finally, it would be interesting to study if this algebraic approach followed here can be improved in order to mount an attack on the original McEliece cryptosystem. In this case however, there are much more unknowns than in the cases considered here and there is much more freedom left on the Y_i's by looking at the linear equations involving only the Y_i's.

References

1. Adams, W., Loustaunau, P.: An Introduction to Gröbner Bases. American Mathematical Society, Providence (July 1994)
2. Avanzi, R.: Lightweight asymmetric cryptography and alternatives to rsa, ecrypt european network of excellence in cryptology (2005), http://www.ecrypt.eu.org/ecrypt1/documents/D.AZTEC.2-1.2.pdf
3. Baldi, M., Bodrato, M., Chiaraluce, G.F.: A new analysis of the McEliece cryptosystem based on QC-LDPC codes. In: Ostrovsky, R., De Prisco, R., Visconti, I. (eds.) SCN 2008. LNCS, vol. 5229, pp. 246–262. Springer, Heidelberg (2008)
4. Baldi, M., Chiaraluce, G.F.: Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes. In: IEEE International Symposium on Information Theory, Nice, France, March 2007, pp. 2591–2595 (2007)

5. 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)
6. Berger, T.P., Loidreau, P.: How to mask the structure of codes for a cryptographic use. Designs Codes and Cryptography 35(1), 63–79 (2005)
7. Berger, T.P., Loidreau, P.: Designing an efficient and secure public-key cryptosystem based on reducible rank codes. In: Canteaut, A., Viswanathan, K. (eds.) INDOCRYPT 2004. LNCS, vol. 3348, pp. 218–229. Springer, Heidelberg (2004)
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.: Explicit bounds for generic decoding algorithms for code-based cryptography. In: Pre-proceedings of WCC 2009, pp. 168–180 (2009)
10. 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)
11. Buchberger, B.: Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal. PhD thesis, Innsbruck (1965)
12. Cox, D.A., Little, J.B., O'Shea, D.: Ideals, Varieties, and algorithms: an Introduction to Computational Algebraic Geometry and Commutative Algebra. Undergraduate Texts in Mathematics. Springer, New York (2001)
13. Faugère, J.-C.: A new efficient algorithm for computing gröbner bases (f4). Journal of Pure and Applied Algebra 139(1-3), 61–88 (1999)
14. Faugère, J.-C.: A new efficient algorithm for computing gröbner bases without reduction to zero: F5. In: ISSAC 2002, pp. 75–83. ACM press, New York (2002)
15. Faugère, J.-C., Levy-dit Vehel, F., Perret, L.: Cryptanalysis of minrank. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 280–296. Springer, Heidelberg (2008)
16. Faugère, J.-C., El Din, M.S., Spaenlehauer, P.-J.: Gröbner bases of bihomogeneous ideals generated by polynomials of bidegree (1,1): Algorithms and complexity. CoRR, abs/1001.4004 (2010)
17. Faugère, J.-C., Gianni, P.M., Lazard, D., Mora, T.: Efficient computation of zero-dimensional gröbner bases by change of ordering. J. Symb. Comput. 16(4), 329–344 (1993)
18. Faure, C., Minder, L.: Cryptanalysis of the McEliece cryptosystem over hyperelliptic curves. In: Proceedings of the eleventh International Workshop on Algebraic and Combinatorial Coding Theory, Pamporovo, Bulgaria, June 2008, pp. 99–107 (2008)
19. Finiasz, M., Sendrier, N.: Security bounds for the design of code-based crypto systems. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 88–105. Springer, Heidelberg (2009)
20. Gabidulin, E., Paramonov, A.V., Tretjakov, O.V.: Ideals over a non-commutative ring and their applications to cryptography. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 482–489. Springer, Heidelberg (1991)
21. Gaborit, P.: Shorter keys for code based cryptography. In: Ytrehus, Ø. (ed.) WCC 2005. LNCS, vol. 3969, pp. 81–91. Springer, Heidelberg (2006)
22. Gibson, J.K.: Severely denting the Gabidulin version of the McEliece public key cryptosystem. Design Codes and Cryptography 6(1), 37–45 (1995)

23. Janwa, H., Moreno, O.: McEliece public key cryptosystems using algebraic-geometric codes. Designs Codes and Cryptography 8(3), 293–307 (1996)
24. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes, 5th edn. North-Holland, Amsterdam (1986)
25. McEliece, R.J.: A Public-Key System Based on Algebraic Coding Theory, pp. 114–116. Jet Propulsion Lab. (1978); DSN Progress Report 44
26. Minder, L., Shokrollahi, A.: Cryptanalysis of the Sidelnikov cryptosystem. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 347–360. Springer, Heidelberg (2007)
27. 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)
28. 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 (1985)
29. Otmani, A., Tillich, J.P., Dallot, L.: Cryptanalysis of McEliece cryptosystem based on quasi-cyclic ldpc codes. In: Proceedings of First International Conference on Symbolic Computation and Cryptography, Beijing, China, April 28-30, pp. 69–81. LMIB Beihang University (2008)
30. Overbeck, R.: Structural attacks for public key cryptosystems based on Gabidulin codes. J. Cryptology 21(2), 280–301 (2008)
31. Sidelnikov, V.M.: A public-key cryptosytem based on Reed-Muller codes. Discrete Mathematics and Applications 4(3), 191–207 (1994)
32. Sidelnikov, V.M., Shestakov, S.O.: On the insecurity of cryptosystems based on generalized Reed-Solomon codes. Discrete Mathematics and Applications 1(4), 439–444 (1992)
33. Stern, J.: A method for finding codewords of small weight. In: Wolfmann, J., Cohen, G. (eds.) Coding Theory 1988. LNCS, vol. 388, pp. 106–113. Springer, Heidelberg (1988)
34. Gauthier Umana, V., Leander, G.: Practical key recovery attacks on two McEliece variants (2009), http://eprint.iacr.org/2009/509
35. Wieschebrink, C.: Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes. eprint 452 (2009), http://eprint.iacr.org/2009/452.pdf

A Gröbner Basics

The classical approach for computing a Gröbner basis of $\mathcal{I} \cap \mathbb{F}_{q^m}[\mathbf{Y}']$ can be described as follows. A reader already familiar with polynomial system solving can skip this part. We have to choose a suitable ordering on the monomials (for a definition of such orders, see for instance [12, Chap. 2, p. 52]). In particular, we have to select an elimination ordering ([1, Chap. 2.3, p. 69]) on the blocks $\mathbf{X}'$, $\mathbf{Y}'$ such that the variables occurring in $\mathbf{X}$ are greater that those of $\mathbf{Y}'$ (denoted by $\mathbf{X}' >> \mathbf{Y}'$). According to [1, Theo. 2.3.4, Chap. 2.3, p. 69], this elimination ordering will permit to compute a Gröbner basis G_{deg} of $\mathcal{I} \cap \mathbb{F}_{q^m}[\mathbf{Y}']$ with respect to a degree order on the variables of $\mathbf{Y}'$ (*i.e.* this is the order induced when "removing" the variables of the block $\mathbf{X}'$ in the elimination ordering). In theory, to compute the variety $\mathcal{V}'$ associated to $\mathcal{I} \cap \mathbb{F}_{q^m}[\mathbf{Y}']$, we have to perform a change of ordering on G_{deg} to compute Gröbner basis G_{lex} of $\mathcal{I} \cap \mathbb{F}_{q^m}[\mathbf{Y}']$. If we assume that $\mathcal{V}'$ is *zero-dimensional* (*i.e.* has a finite number of solutions so that $\#\mathcal{V}' < \infty$), then an efficient tool to perform the change of ordering is the FGLM algorithm [17]. The complexity of computing G_{lex} from G_{deg} with FGLM is polynomial in the size of $\mathcal{V}'$, *i.e.* $\mathcal{O}\big((\#\mathcal{V}')^3\big)$. In our case, the size of $\mathcal{V}'$ is very small (< 10).

We have used a slightly modified version of F_4 [13] for computing a Gröbner basis G_{deg} of $\mathcal{I} \cap \mathbb{F}_{q^m}[\mathbf{Y}']$. The idea is to adapt the algorithm for performing the Gröbner basis computation in $\mathbb{F}_{q^m}[\mathbf{X}'][\mathbf{Y}']$, *i.e.* the set of polynomials in $\mathbf{Y}'$ whose coefficients are polynomials in $\mathbb{F}_{q^m}[\mathbf{X}']$. As for the usual F_4, we process degree by degree. However, we consider only the degree of the polynomials w.r.t. the variables of $\mathbf{X}'$. We stop the computation as soon as we have sufficiently many equations in $\mathbf{Y}'$ (for instance, as soon as we detect that $\mathcal{V}'$ has a finite number of solution, *i.e.* of dimension zero). The modified version is defined below.

```
Input: { $\mathbf{X}$' and $\mathbf{Y}$'
       { $F$ a finite subset of $\mathbb{F}_{q^m}[\mathbf{X}', \mathbf{Y}']$
       { $<$ a monomial admissible order
Output: a finite subset of $\mathbb{F}_{q^m}[\mathbf{Y}']$.
$G := F$ and $P := \{\mathrm{CritPair}(f,g) \mid (f,g) \in G^2 \text{ with } f \neq g\}$
while $P \neq \emptyset$ and $\dim(G \cap \mathbb{F}_{q^m}[\mathbf{Y}']) > 0$ do
    $d := \min\{\deg_{\mathbf{X}'}(p) \mid p \in P\}$ minimal partial degree of critical pairs
    Extract from $P$, $P_d$ the list of critical pairs of degree $d$
    $R :=$ MATRIX_REDUCTION($\mathrm{Left}(P_d) \cup \mathrm{Right}(P_d), G$)
    for $h \in R$ do
        $P := P \cup \{\mathrm{CritPair}(h,g) \mid g \in G\}$
        $G := G \cup \{h\}$
return $G \cap \mathbb{F}_{q^m}[\mathbf{Y}']$
```

Fig. 1. Algorithm F_4 (modified version)

For the definition of MATRIX_REDUCTION, and CritPair, we refer to [13]. Briefly, the first function performs the usual polynomial reduction of Buchberger's algorithm [12] using linear algebra. The second function selects critical pairs with respect to a defined strategy.

B Description of the Variant Based on Dyadic Goppa Codes

The cryptosystem presented in [27] considers particular alternant codes called *quasi-dyadic* Goppa codes. Goppa codes form an important subclass of alternant codes. Goppa codes are defined by means of a polynomial $G(X)$ of degree ℓ with coefficients in $\mathbb{F}_{q^m}$ and for which the sequence $\boldsymbol{x}$ is assumed not to contain any root of $G(X)$. The alternant code defined by the parity-check matrix $\boldsymbol{V}_\ell(\boldsymbol{x},\boldsymbol{y})$ with $y_i = G(x_i)^{-1}$ is called a Goppa code over $\mathbb{F}_q$ and is denoted by $\mathscr{G}(\boldsymbol{x}, G)$. It has dimension $n - m\ell$ and minimum distance $d \geq \ell + 1$ [24, Chap. 12, p. 340]. In the special case where the roots $\boldsymbol{z} = (z_0, \dots, z_{\ell-1})$ of $G(X)$ are distinct and all belong to $\mathbb{F}_{q^m}$ then $\mathscr{G}(\boldsymbol{x}, G)$ admits a parity-check matrix $\boldsymbol{C}(\boldsymbol{z},\boldsymbol{x})$ in Cauchy form [24, p. 345].

The scheme in [27] considers a Goppa code that admits a parity-check matrix that is both a Cauchy matrix and a block matrix where each block is dyadic. An $\ell \times \ell$ matrix $\boldsymbol{\Delta} = (\Delta_{i,j})$ with $0 \leq i \leq \ell - 1$ and $0 \leq j \leq \ell - 1$ is *dyadic* if and only if $\Delta_{i,j} = h_{i \oplus j}$ where $\oplus$ is the bitwise exclusive-or on the binary representation of the indices and $\boldsymbol{h} = (h_0, \dots, h_{\ell-1})$ is the first row of $\boldsymbol{\Delta}$. Let $\boldsymbol{h} = (h_0, \dots, h_{N-1})$ be a vector of $\mathbb{F}_{q^m}^N$ with $\ell \leq N$. We denote by $\boldsymbol{\Delta}_\ell(\boldsymbol{h}) = (\Delta_{i,j})$ the $\ell \times N$ matrix such that $\Delta_{i,j} = h_{i\oplus j}$. One can easily observe that $\boldsymbol{\Delta}_\ell(\boldsymbol{h})$ is the juxtaposition of N_0 dyadic matrices of size $\ell \times \ell$ when $N = N_0\ell$ for some integer N_0. Proposition 7 proved in [27, Theorem 2] characterizes dyadic Cauchy matrices.

Proposition 7. *A necessary and sufficient condition for $\boldsymbol{\Delta}_\ell(\boldsymbol{h})$ to be a Cauchy matrix $\boldsymbol{C}(\boldsymbol{z},\boldsymbol{x})$ is that $\mathbb{F}_{q^m}$ is of characteristic 2 and for any i, j in $\{0, \dots, N-1\}$ we have:*

$$\frac{1}{h_{i\oplus j}} = \frac{1}{h_j} + \frac{1}{h_i} + \frac{1}{h_0}. \tag{14}$$

Furthermore, for any $\theta \in \mathbb{F}_{q^m}$ and for any $z_i^* = 1/h_i + \theta$ and $x_j^* = 1/h_j + 1/h_0 + \theta$, the Cauchy matrix $\boldsymbol{C}(\boldsymbol{z}^*, \boldsymbol{x}^*)$ is equal to $\boldsymbol{\Delta}_\ell(\boldsymbol{h})$.

Indeed, the public generator matrix $\boldsymbol{G}$ is a $k \times n$ block matrix where each block is an $\ell \times \ell$ dyadic matrix with ℓ being a power of 2. The entries of $\boldsymbol{G}$ belong to $\mathbb{F}_q$ and the integers k and n are chosen such that $n = n_0\ell$ and $k = n - m\ell = \ell(n_0 - m)$ where n_0 is some integer and m defines the extension $\mathbb{F}_{q^m}$. The matrix $\boldsymbol{G}$ is obtained from a secret $\ell \times n$ block parity-check matrix $\boldsymbol{H} = \left(\boldsymbol{\Delta}_\ell(\boldsymbol{f}_0) | \cdots | \boldsymbol{\Delta}_\ell(\boldsymbol{f}_{n_0-1})\right)$ where each block $\boldsymbol{\Delta}_\ell(\boldsymbol{f}_j)$ is an $\ell \times \ell$ dyadic matrix and for any $0 \leq j \leq n_0 - 1$, $\boldsymbol{f}_j$ is a vector of $\mathbb{F}_{q^m}^\ell$ such that $\boldsymbol{f}_j = \gamma_j \left(h_{\omega_j \ell \oplus d_j}, h_{(\omega_j\ell+1)\oplus d_j}, \dots, h_{((\omega_j+1)\ell-1)\oplus d_j}\right)$ where $\boldsymbol{h} = (h_0, \dots, h_{N-1})$ is a random vector of $\mathbb{F}_{q^m}^N$ that satisfies Equation (14) and such that $N = N_0\ell$ for some integer $N_0 \gg n_0$. The integers ω_j, d_j are chosen such that $0 \le \omega_j \le N_0 - 1$ and $0 \le d_j \le \ell - 1$. The coefficients γ_j are non zero elements of $\mathbb{F}_{q^m}$. Note that the integers ω_j's are different. The secret key consists then of the vectors $\boldsymbol{h}$, $\boldsymbol{\omega} = (\omega_0, \dots, \omega_{n_0-1})$, $\boldsymbol{d} = (d_0, \dots, d_{n_0-1})$ and $\boldsymbol{\gamma} = (\gamma_0, \dots, \gamma_{n_0-1})$.

C Proof of Proposition 4

Lemma 3. *Let $N = N_0\ell$ with $\ell = 2^e$ for some integer e and let $\boldsymbol{h}$ be a vector in $\mathbb{F}_{q^m}^N$ that satisfies Equation (14). Let $\mathscr{G}(\boldsymbol{a}^*, G)$ be the Goppa code such that $\boldsymbol{a}^* = (a_0^*, \dots, a_{N-1}^*)$ is defined by $a_j^* = 1/h_j + 1/h_0$ for $0 \le j \le N-1$ and $G(X) = \prod_{i=0}^{\ell-1}(X - z_i)$ with $z_i = 1/h_i$. Then for any i, j in $\{0, \dots, N-1\}$ we have $a_{i\oplus j}^* = a_i^* + a_j^*$ and for any $0 \le j \le N_0 - 1$ and $0 \le i, i' \le \ell - 1$*

$$G(a_{j\ell+i}^*)^{-1} = \prod_{l=j\ell}^{(j+1)\ell-1} h_l$$

Proof. The property that $a_{i\oplus j}^* = a_i^* + a_j^*$ comes from Equation (14). Furthermore, we have:

$$G(a_{j\ell+i}^*)^{-1} = \prod_{\ell=0}^{\ell-1}(z_\ell - a_{j\ell+i}^*)^{-1} = \prod_{\ell=0}^{\ell-1}(1/h_\ell + 1/h_{j\ell+i} + 1/h_0)^{-1} = \prod_{\ell=0}^{\ell-1} h_{j\ell+\ell}$$

which terminates the proof.

We remark in particular that we have $G(a_{j\ell+i}^*) = G(a_{j\ell}^*)$ for any $0 \le j \le n_0 - 1$ and $0 \le i \le \ell - 1$. The next lemma we give without proof shows that the action of a dyadic permutation can be simply characterized as a translation.

Lemma 4. *Let t and d two integers such that $0 \le d \le \ell - 1$. For any vector $\boldsymbol{v} = (v_0, \dots, v_{\ell-1})$, we have:*

$$\boldsymbol{v} \times \boldsymbol{\Delta}_\ell(\boldsymbol{b}_d) = (v_d, v_{1\oplus d}, \dots, v_{(\ell-1)\oplus d}) \tag{15}$$

where the vector $\boldsymbol{b}_d = (b_{d,0}, \dots, b_{d,\ell-1})$ is such that $b_{d,j} = 0$ if $j \ne d$ and $b_{d,d} = 1$.

We are now prepared to prove Proposition 4. Let $(\boldsymbol{h}, \boldsymbol{\omega}, \boldsymbol{d}, \boldsymbol{\gamma})$ be the private key and let $\boldsymbol{G}$ be the public generator matrix. We shall see that a parity-check matrix for the code generated by $\boldsymbol{G}$ is $\boldsymbol{V}_\ell(\boldsymbol{a}, \boldsymbol{\lambda})$ with $a_{j\ell+i} = a_{(\omega_j t+i)\oplus d_\ell}^*$ and $\lambda_{j\ell+i} = \gamma_j\ G(a_{\omega_j t}^*)^{-1}$ where $\boldsymbol{a}^*$ and $G(X)$ are defined as in Lemma 3. Indeed, we know that the code defined by the parity-check matrix $\boldsymbol{\Delta}_\ell(\boldsymbol{h})$ is also defined by the parity-check matrix $\boldsymbol{V}_\ell(\boldsymbol{a}, \boldsymbol{\lambda})$ where $\lambda_j = G(a_j)^{-1}$ for any $0 \le j \le N - 1$. Recall from Lemma 3 that $G(a_{j\ell+i}) = G(a_{j\ell})$ for any $0 \le j \le N_0 - 1$ and $0 \le i \le \ell - 1$. The role of $\boldsymbol{\omega}$ is to pick n_0 dyadic blocks from $\boldsymbol{\Delta}_\ell(\boldsymbol{h})$. These blocks correspond to the columns $a_{\omega_j\ell}^*, \dots, a_{(\omega_j+1)\ell-1}^*$ of $\boldsymbol{V}_\ell(\boldsymbol{a}, \boldsymbol{\lambda})$ when j describes $\{1, \dots, n_0\}$. These columns are then multiplied by a dyadic permutation matrix $\boldsymbol{\Delta}_\ell(\boldsymbol{b}_{d_\ell})$ which leads to reorder the columns as $a_{\omega_j\ell\oplus d_j}, \dots, a_{((\omega_j+1)\ell-1)\oplus d_j}$ according to

Lemma 4. Finally, each dyadic block is scaled by γ_j which means that if we set $\lambda_{j\ell+i} = \gamma_j \; G(a_{\omega_j \ell})^{-1}$ then $\boldsymbol{V}_\ell(\boldsymbol{a}, \boldsymbol{\lambda})$ is another parity-check matrix of the code generated by $\boldsymbol{G}$. We are now going to show that for any $0 \leq j \leq n_0 - 1$ and $0 \leq i, i' \leq \ell - 1$, we have the following equations:

$$\begin{cases} \lambda_{j\ell+i} & = \lambda_{j\ell} \\ a_{j\ell+i} + a_{j\ell} & = a_i + a_0 \\ a_{j\ell+i\oplus i'} & = a_{j\ell+i} + a_{j\ell+i'} + a_{j\ell} \end{cases} \tag{16}$$

It is clear from Lemma 3 that $\lambda_{j\ell+i} = \lambda_{j\ell}$. On the other hand, $a_{j\ell+i} = a_{(\omega_j \ell+i)\oplus d_j}$ $= 1/h_{(\omega_j \ell+i)\oplus d_j} + 1/h_0$. From Equation (14) we thus have:

$$\begin{aligned} a_{j\ell+i} &= \frac{1}{h_{\omega_j \ell+i}} + \frac{1}{h_{d_j}} = \frac{1}{h_{\omega_j \ell}} + \frac{1}{h_i} + \frac{1}{h_0} + \frac{1}{h_{d_j}} \\ &= \frac{1}{h_{\omega_j \ell \oplus d_j}} + \frac{1}{h_i} + \frac{1}{h_0} = a_{j\ell} + \frac{1}{h_i} + \frac{1}{h_0}. \end{aligned}$$

We observe in particular that $a_i + a_0 = 1/h_i + 1/h_0$ and since this quantity does not depend on ℓ, this is equivalent to say that $a_{j\ell+i} + a_{j\ell} = a_i + a_0$. Before proving the third equation, we can first see that $a_{j\ell+i\oplus i'} + a_{j\ell} = a_{i\oplus i'} + a_0$. So if we know that $a_{i\oplus i'} = a_i + a_{i'} + a_0$ then we would get $a_{i\oplus i'} = a_{j\ell+i} + a_{j\ell} + a_{i'}$ which finally implies $a_{i\oplus i'} = a_{j\ell+i} + a_{j\ell+i'} + a_0$ that leads to the expected result. Now we have $a_{i\oplus i'} = a_{(\omega_1 \ell+i+i')\oplus d_1} = a_{\omega_1 \ell+i+i'} + a_{d_1} = a_{\omega_1 \ell+i} + a_{i'} + a_{d_1} = a_{(\omega_1 \ell+i)\oplus d_1} + a_{i'}$. Therefore we obtain:

$$a_{i\oplus i'} = a_i + a_{i'} + a_{\omega_1 \ell} + a_{d_1} + a_{\omega_1 \ell} + a_{d_1} = a_i + a_{i'} + a_0.$$

Key Recovery Attacks of Practical Complexity on AES-256 Variants with up to 10 Rounds

Alex Biryukov[1], Orr Dunkelman[2,4], Nathan Keller[3,4], Dmitry Khovratovich[1], and Adi Shamir[4]

[1] University of Luxembourg, Luxembourg
[2] École Normale Supérieure,
45 rue d'Ulm, 75230 Paris, France
[3] Einstein Institute of Mathematics, Hebrew University.
Jerusalem 91904, Israel
[4] Faculty of Mathematics and Computer Science
The Weizmann Institute
Rehovot 76100, Israel

Abstract. AES is the best known and most widely used block cipher. Its three versions (AES-128, AES-192, and AES-256) differ in their key sizes (128 bits, 192 bits and 256 bits) and in their number of rounds (10, 12, and 14, respectively). While for AES-128, there are no known attacks faster than exhaustive search, AES-192 and AES-256 were recently shown to be breakable by attacks which require 2^{176} and $2^{99.5}$ time, respectively. While these complexities are much faster than exhaustive search, they are completely non-practical, and do not seem to pose any real threat to the security of AES-based systems.

In this paper we aim to increase our understanding of AES security, and we concentrate on attacks *with practical complexity*, i.e., attacks that can be experimentally verified. We show attacks on reduced-round variants of AES-256 with up to 10 rounds with complexity which is feasible. One of our attacks uses only two related keys and 2^{39} time to recover the complete 256-bit key of a 9-round version of AES-256 (the best previous attack on this variant required 4 related keys and 2^{120} time). Another attack can break a 10-round version of AES-256 in 2^{45} time, but it uses a stronger type of *related subkey attack* (the best previous attack on this variant required 64 related keys and 2^{172} time). While the full AES-256 cannot be directly broken by these attacks, the fact that 10 rounds can be broken with such a low complexity raises serious concerns about the remaining safety margin offered by AES-256.

1 Introduction

AES (Advanced Encryption Standard) is an iterated block cipher which was selected by NIST in October 2000 after a three year competition. It was made a national and international standard, and replaced DES as the most widely deployed block cipher in both software and hardware applications.

The three standardized versions of AES are called AES-128, AES-192, and AES-256. They differ from each other in the key length (128, 192, and 256 bits)

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 299–319, 2010.

and the number of rounds (10, 12, and 14, respectively). Their data encryption rounds are all the same, but the details of the key schedule are slightly different since different amounts of key material are available and required in the three variants. Their security was thoroughly analyzed by the NSA, which declared in 2003 that none of them has any known vulnerability and that the longer-key variants AES-192 and AES-256 can be used to protect top secret US governmental data [11].

The situation started to change in the spring of 2009, when Biryukov, Khovratovich and Nikolić [4] found a key recovery attack on AES-256 with related keys and time complexity of 2^{131}. The attack was completely non-practical, but it was the first time that anyone had published an attack on the full AES cipher which was faster than exhaustive search. Shortly afterwards, Biryukov and Khovratovich [3] reduced the time complexity of the attack on AES-256 to $2^{99.5}$, and described the first attack on AES-192 which was faster than exhaustive search (requiring 2^{176} instead of 2^{192} time). As a result, AES is no longer considered to be theoretically secure, but the crucial question all of us are facing is how far it is from becoming practically insecure.

The practicality of various types of cryptanalytic attacks depends on many factors: Attacks based on a few ciphertexts are better than attacks that require many ciphertexts, known plaintext attacks are better than chosen plaintext attacks, nonadaptive attacks are better than adaptive attacks, single key attacks are better than related key attacks, etc. Since it is difficult to quantify the relative importance of all these factors in different scenarios, we usually concentrate on the total running time of the attack, which is a single well defined number. While one can argue about the exact transition point between cryptanalytic attacks of practical and theoretical time complexity, it is reasonable to place it at around 2^{64} basic instructions. Since optimized AES implementations require about 16 clock cycles per byte and each plaintext has 16 bytes, this is approximately equal to 2^{56} AES encryptions. This choice of threshold is supported by the fact that 2^{55} evaluations of DES were carried out on special purpose hardware several years ago, while a collision finding attack on SHA-1 which was expected to take 2^{61} in a large distributed effort, was abandoned due to lack of progress.

To try and estimate the security margin left in a given cryptosystem, we can take two different approaches. One of them is to compare the time complexity of the best known attack on the full cryptographic scheme with this threshold. This was the approach that motivated [3] and [4], and in this sense AES still seems to be very secure. However, attacks can only get better over time, and in particular they tend to exhibit “round creep” that slowly increases the number of rounds which can be attacked with practical complexity. A second approach (and the one taken in this paper) is thus to compare this number to the total number of rounds in the iterated cryptosystem. An example which demonstrates the difference between these two approaches is the comparison between Serpent and Rijndael made during the AES competition. The best attacks on their full versions had exactly the same time complexity (namely, that of exhaustive search).

Table 1. Summary of attacks on AES-256

Rounds	Scenario	Time	Data	Memory	Result	Section
8	Key Diff. – CP	2^{31}	2^{31}	2	Distinguisher	Sect. 6.1
8	Subkey Diff. – CC	$2^{26.5}$	$2^{26.5}$	$2^{26.5}$	35 subkey bits	Sect. 6.2
9	Key Diff. – CP	2^{39}	2^{38}	2^{32}	Full key	Sect. 4.1.3
9	Subkey Diff. – CC	2^{32}	2^{32}	2^{32}	56 key bits	Sect. 4.2
10	Subkey Diff. – CP	2^{49}	2^{48}	2^{33}	Distinguisher	Sect. 5.1
10	Subkey Diff. – CC	2^{45}	2^{44}	2^{33}	35 subkey bits	Sect. 5.2
11	Subkey Diff. – CP	2^{70}	2^{70}	2^{33}	50 key bits	Sect. 6.3

CP — Chosen plaintext, CC — Chosen ciphertext

However, Rijndael was designed to be as fast as possible (with a relatively small security margin), whereas Serpent was designed to have a large security margin (at the expense of speed on some platforms), which made it more resistant against future cryptanalytic developments. As an extreme example, we would feel very uncomfortable using a theoretically secure n round block cipher if we knew that its $n-1$ round version can be attacked with practical complexity, since such a scheme is "one idea away from disaster".

What we show in this paper is that this type of security margin in AES is significantly smaller than generally believed. In particular, we describe several key derivation attacks of practical complexity on AES-256 when its number of rounds is reduced to approximately that of AES-128. The best previously published attacks on such variants were far from practical, requiring 4 related keys and 2^{120} time to break a 9-round version of AES-256 [9], and 64 related keys and 2^{172} time to break a 10-round version of AES-256 ([9], see also [2]).[1] In this paper we describe an attack on 9-round AES-256 which can find its complete 256-bit key in 2^{39} time by using only the simplest type of related keys (in which the chosen plaintexts are encrypted under two keys whose XOR difference can be chosen in many different ways). Our best attack on 10-round AES-256 requires only two keys and 2^{45} time, but it uses a stronger type of *related subkey attack*. These attacks can be extended into a quasi-practical 2^{70} attack on 11-round AES-256, and into a trivial 2^{26} attack on 8-round AES-256. We summarize the complexities of our attacks in Table 1. The attacks are particularly well suited to AES-256 in counter mode of operation (AES-CTR), since the adversary can get all the chosen plaintexts he needs by starting from just two chosen initial values and running the counter mode in a natural way.

This paper is organized as follows. In Section 2 we describe the AES cipher and its different versions, and in Section 3 we discuss various types of related key

[1] For comparison, the best practical single-key attack on AES-256 is a SQUARE attack on 6 rounds which requires $6 \cdot 2^{32}$ chosen plaintexts and has time complexity of 2^{44} [7].

attacks. Our attacks on 9-round variants of AES-256 are described in Section 4, and our attacks on 10-round variants of AES-256 are described in Section 5. In Section 6 we briefly outline several other attacks on variants of AES-256 with a smaller or larger number of rounds, and in Section 7 we describe how to choose more naturally looking plaintexts in our attack. We conclude with a discussion of our results in Section 8.

2 Description of AES-256

AES-256 is an iterated block cipher which encrypts 128-bit plaintexts with 256-bit keys. It has 14 rounds, where each round applies four basic operations in the following order:

- SubBytes (SB) is a nonlinear byte-wise substitution that applies the same 8×8 S-box to every byte.
- ShiftRows (SR) is a cyclic shift of the i'th row by i bytes to the left.
- MixColumns (MC) is a matrix multiplication over a finite field applied to each column.
- AddRoundKey (ARK) is an exclusive-or with the round subkey.

Before the first round an additional whitening ARK operation is performed, and in the last round the MC operation is omitted. AES-128 and AES-192 use exactly the same round function, but the number of rounds is reduced to 10 and 12, respectively.

Next we describe the key schedule of AES-256. The supplied 256-bit key is divided into 8 words of 32 bits each $(W[0], W[1], \cdots, W[7])$. To generate the 15 subkeys of 128 bits (which consist of 60 words of 32 bits), the following algorithm is used:

- For $i = 8$ till $i = 59$ do the following:
  - If $i \equiv 0 \bmod 8$, then $W[i] = W[i-8] \oplus \mathrm{SB}(\mathrm{RotByte}(W[i-1])) \oplus \mathrm{Rcon}[i/8]$,
  - If $i \equiv 4 \bmod 8$, then $W[i] = W[i-8] \oplus \mathrm{SB}(W[i-1]))$,
  - Else $W[i] = W[i-8] \oplus W[i-1]$,

where RotByte represents one byte rotation (i.e., $(a_0, a_1, a_2, a_3) \to (a_1, a_2, a_3, a_0)$), and *Rcon* denotes an array of fixed constants.

The key schedules of AES-128 and AES-192 are slightly different, since they have to apply more mixing operations to the shorter key in order to produce the slightly smaller number of subkeys for the various rounds. This small difference in the key schedules plays a major role in making AES-256 more vulnerable to our attacks, in spite of its longer key and supposedly higher security. For more details about all aspects of the design of AES, we refer the reader to [6].

2.1 Our Notations

The rounds of AES-256 are numbered $0, 1, \ldots, 13$. The subkey used at the end of round i is denoted by K^i, and the whitening subkey is denoted by K^{-1}.

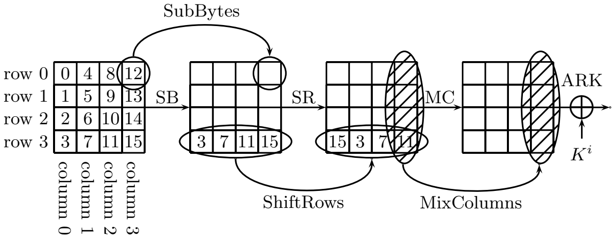


Fig. 1. The AES round function and byte marking conventions

The XOR difference between the subkeys K^i produced by two related keys is denoted by $\Delta(K^i)$. The 128-bit state at the input to round i is denoted by I^i. The XOR difference between the states I^i produced during two related encryptions is denoted by $\Delta(I^i)$.

Any 128-bit intermediate state or subkey is described by a 4×4 byte matrix, whose bytes are numbered as described in Figure 1. The rows and the columns of this matrix are numbered 0,1,2,3. Byte j of subkey K^i or of state I^i is denoted by K^i_j or I^i_j, respectively. Similarly, a difference in this byte is denoted by $\Delta(K^i_j)$ or $\Delta(I^i_j)$, respectively. When we want to refer to more than one byte, we list the relevant bytes in the subscript, as in $\Delta(I^i_{j,k,l,m})$.

3 Related-Key Attacks

The related-key attack model [1,10] is a class of cryptanalytic attacks in which the adversary knows or chooses a relation between several keys and is given access to encryption/decryption functions with all these keys. The goal of the adversary is to find the actual keys. the relation between the keys is commonly selected as XOR or rotation, but other bijective relations are also considered. In the simplest form of this attack, this relation is just an XOR with a constant: $K_2 = K_1 \oplus C$, where the constant C is chosen by the adversary. This type of relation allows the adversary to trace the propagation of XOR differences induced by the key difference C through the key schedule of the cipher. However, more complex forms of this attack allow other (possibly non-linear) relations between the keys. For example, in some of the attacks described in this paper the adversary chooses a desired XOR relation in the second subkey, and then defines the implied relation between the actual keys as: $K_2 = F^{-1}(F(K_1) \oplus C) = R_C(K_1)$ where F represents a single round of the AES-256 key schedule, and the constant C is chosen by the adversary. We call such attacks *related subkey attacks*, and we emphasize that once the second key is computed via the relation, all its subkeys are computed in the standard way, i.e., consistent with the full evolution of the key schedule.

The choice of the relation between secret keys gives additional power to the adversary compared to other cryptanalytic attacks in which the adversary can manipulate only the plaintexts and/or the ciphertexts. The simpler the relation is, the easier it is for an adversary to manipulate the key in the desired fashion. For example, the key exchange protocol 2PKDP [13], allows an adversary to XOR the unknown key with a constant. Other related key attacks, such as those presented in [8,12], discuss practical attacks on well known schemes under different key relations.

Even though related-key attacks may not be a realistic threat in many cryptographic applications, resistance to such attacks is an important design goal for new block ciphers, and in fact it was one of the stated design goals of the Rijndael algorithm, which was selected as the Advanced Encryption Standard. Designers usually try to build primitives which can be automatically used without further analysis in the widest possible set of applications, protocols, or modes of operation. For example, block ciphers susceptible to related-key differential attacks may lead to insecure compression functions, if they are instantiated by a Davies-Meyer construction [4]. Moreover, history shows that users of cryptography tend to use (or misuse) such cryptographic primitives in very creative manners, forcing the cryptographers to design schemes which resemble ideal primitives under the broadest possible set of scenarios.

4 Attacks on 9 Round Variants of AES-256

4.1 A Related-Key (XOR Difference) Attack

In this section we present an attack on 9-round AES-256, which is based on the simplest form of a related key attack, in which plaintexts can be encrypted under two unknown keys, whose XOR difference can be chosen by the adversary. The number of chosen plaintexts used in the attack is 2^{38}, and the most time consuming part of the attack is to ask the legitimate user to prepare all their 2^{39} corresponding ciphertexts under the two keys. Once this is done, the derivation of the complete 256-bit key requires less than 2^{39} time.

4.1.1 The Related-Key Differentials Used in Our Attacks

The basic differential characteristic we use is an 8-round differential, whose first seven rounds are part of the longer differential path introduced in [4], but its eighth round is different (see Figure 2). Since we use a related key attack, each one of the characteristics is a combination of a key characteristic and a state characteristic, which are intertwined in a way that maximizes their total probability. We first describe the key difference part of the characteristics (which is independent of the data), and how it generates the various subkey differences we would like to have.

Let a and b be *any two 32-bit words.* When we choose the key difference of the differential as $\Delta(K) = (b, b, b, b, a, 0, a, 0)$, we can easily show that the ten 128-bit subkeys used in the 9-round version of AES-256 (including the initial whitening subkey) have the following difference structure with probability 1:

$$\begin{gathered}(b,b,b,b||a,0,a,0)\\(b,0,b,0||a,a,0,0)\\(b,b,0,0||a,0,0,0)\\(b,0,0,0||a,a,a,a)\\(c,c,c,c||d,e,d,e),\end{gathered}$$

where

$$c = b \oplus SB(RotByte(a)), \qquad d = a \oplus SB(c), \qquad e = d \oplus a,$$

and each row describes the differences of two additional subkeys, starting with the original key difference in the first row. This is an amazingly long trail for a key schedule which tries to be nonlinear, and it clearly indicates that this part of the design of AES-256 is seriously flawed.

To create the desired cancellations between this key characteristic and the state characteristic, we have to impose additional constraints on the choice of the two words a and b. Let α be any non-zero byte value (there are 255 ways to choose it, and each choice will lead to a different related key attack of the same complexity). By the construction of the SubBytes operation, there exists a unique byte value β such that the differential $\alpha \rightarrow \beta$ through the SubBytes operation holds with probability 2^{-6} (see [6]). Let b be the 32-bit column vector $b = MC((\beta,0,0,0)^T)$, and let a be the 32-bit column vector $a = (\alpha,0,0,0)^T$. Note that with this choice, the top three bytes of c are equal to the corresponding known values in b, and thus the only effect of the nonlinearity of the SubBytes operation on the subkey differences is to make the lowest byte of c and the four bytes of d unknown to the adversary. We denote these bytes by c_3 and (d_0, d_1, d_2, d_3).

The input difference in the state part of the characteristic is (b,b,b,b) (i.e., the same input difference b in each column). The subkey difference $\Delta(K^{-1})$ used in the whitening phase cancels the identical plaintext difference, and thus the difference $\Delta(I^0)$ is zero. The zero difference remains unchanged until the ARK operation at the end of round 0. The subkey difference $\Delta(K^0)$ inserts difference α into two bytes of the state. With probability 2^{-12}, this difference evolves through the SB operation to difference β, that is transformed through the MC operation to b, which is then cancelled with the subkey difference $\Delta(K^1)$, resulting in $\Delta(I^2) = 0$. The zero difference is preserved, until the subkey difference $\Delta(K^2)$ inserts difference α into two bytes of the state. This difference is again cancelled with probability 2^{-12} with the subkey difference $\Delta(K^3)$, resulting in $\Delta(I^4) = 0$. The subkey difference $\Delta(K^4)$ inserts difference α into one byte of the state, that is cancelled with the key difference $\Delta(K^5)$ with probability 2^{-6}. The zero difference is preserved again, until the subkey difference $\Delta(K^6)$ inserts difference α in four bytes of the state. With probability 2^{-24}, this difference evolves to difference (b,b,b,b) after the MC operation of round 7. Since the subkey difference $\Delta(K^7)$ is (c,c,c,c), we have $\Delta(I^8) = (f,f,f,f)$, where $f = b \oplus c = (0,0,0,b_3 \oplus c_3)^T$. This is the output difference of the differential. Overall, the differential $(b,b,b,b) \to (f,f,f,f)$ for rounds 0–7 holds with probability 2^{-54} for the subkey differences presented above. The differential characteristic is depicted in Figure 2.

In our attack we do not use this basic differential characteristic as is, but rather several "truncated" variants in which some of the differential conditions are relaxed:

1. **Main Differential.** In this differential we relax the differential conditions on three of the four active S-boxes in round 7, and leave only the condition in the SB operation in byte 0. That is, we require that the output difference of the SB operation in byte 0 is β, and do not restrict the output differences of the SB operation in bytes 4,8,12. As a result, the difference in the first column after the MC operation is b, and thus $\Delta(I^8_{0,1,2}) = 0$. The differences in the other columns, as well as $\Delta(I^8_3)$, are unknown. The probability of this truncated differential is 2^{-36}.
2. **Shifted Main Differential.** This truncated differential is almost identical to the previous one. The only difference is that we keep the differential condition in byte 12 of round 7, instead of byte 0.
3. **Complementary Differential for the 9-Round Attack.** In this differential we consider only rounds 0–6, and relax the differential condition in round 5. Since the input difference to round 5 is non-zero only in byte I^5_0, and there is no differential condition, the difference $\Delta(I^6)$ is in the entire first column (bytes $I^6_{0,1,2,3}$). This difference evolves to differences in all the 16 bytes in $\Delta(I^7)$, but since the MC operation is linear, there is a total of only 256 possible differences in the four bytes of each column. The probability of this truncated differential is 2^{-24}.
4. **Differential for the 8-Round Attack.** For the sake of completeness, we also describe a simplified differential which is used later in Section 6.1 to attack an 8-round version of AES-256 with a lower complexity. In this differential we consider rounds 0–7, and relax all the differential conditions in round 7. Since the difference $\Delta(I^7)$ is non-zero only in bytes 0,4,8,12, and since in the 8-round variant of AES-256 there is no MC operation at round 7, the ciphertext difference in bytes 1,2,5,6,9,10,13,14 is known to the adversary (it is equal to the difference chosen in the respective bytes of b). This supplies the adversary with a 64-bit filtering, which will be sufficient to discard all the wrong pairs. The probability of this truncated differential is 2^{-30}.

4.1.2 Preliminaries for the 9-Round Attack

We now describe several simple properties of the round function of AES, which are exploited by our attack.

1. **Observation A.** Consider a pair that satisfies the main differential described in Section 4.1.1. As noted above, the difference in columns 1,2,3 after the MC operation of round 7 is unknown. However, due to the properties of the SB and MC operations, there are only 127 possible values for the

difference in each of these columns. Moreover, these 127 differences assume 127 different values in each of the four bytes. As a result, if the adversary knows the difference in one byte, she can obtain immediately the difference in the other three bytes of the same column, along with a one-bit filtering (since only 127 out of the 256 byte values are possible differences). Similarly, if a pair satisfies the complementary differential described in Section 4.1.1, then if the adversary knows the difference $\Delta(I^7)$ in one byte, she can obtain immediately the difference in the other three bytes of the same column (though, without the additional filtering since in this case there are 256 possible differences).

2. **Observation B.** Given an input and an output difference to the SB operation, there are three possibilities:
   (a) With probability 1/256, there exist four pairs of actual values that satisfy both the input and the output difference.
   (b) With probability 126/256, there exist two such pairs.
   (c) With probability 129/256, there exist no such pairs, and thus the impossible input/output difference pair can be discarded immediately.

   When there exist possible pairs, they can be found immediately by a table look-up. In order to do this, the adversary prepares in advance the difference distribution table which stores for each possible input/output difference, the actual values of the pairs satisfying these differences. The time required to prepare the table is 2^{16} evaluations of SB, and the required memory is 2^{17} bytes. Each look-up operation in this table allows the adversary to either discard the input/output difference pair (with probability $\approx 1/2$), or to find immediately the actual input and output values of the two bytes (with two or four possibilities).
3. **Observation C.** Consider the subkey differences between the related-keys used in the attack. It turns out that the unknown difference bytes c_3, d_0, d_1, d_2, d_3 can take on only 127 out of the 256 possible values, and (except for d_3), these 127 values are known to the adversary in advance. Indeed, since by the key schedule, $c = b \oplus SB(RotByte(a))$, the adversary knows that $x = c_3 \oplus b_3$ is one of the 127 differences such that the differential $\alpha \to x$ through SB is possible. Since b_3 is known to the adversary, the 127 possible values of c_3 are also known. Similarly, since $d = a \oplus SB(c)$ and $(c_0, c_1, c_2) = (b_0, b_1, b_2)$ are known to the adversary, she can find the 127 possible values for each of the bytes d_0, d_1, d_2.

4.1.3 A Detailed Description of the 9-Round Attack

We now present the actual algorithm used by the adversary to derive the key information from the given ciphertexts, along with some textual explanations:

1. **Data Generation**
   (a) Choose 2^{37} arbitrary plaintexts P, and add to them the 2^{37} chosen plaintexts $P' = P \oplus (b, b, b, b)$. Ask the user to encrypt each one of these 2^{38} plaintexts under the two unknown keys K and $K' = K \oplus (b, b, b, b, a, 0, a, 0)$. Each one of the 2^{37} choices of P provides two different

pairs of encryption operations $(((P, K), (P', K'))$ and $((P, K'), (P', K)))$ which have the desired input difference $\Delta(P) = (b, b, b, b)$, along with the desired key difference $\Delta(K) = (b, b, b, b, a, 0, a, 0)$, and thus we get a total of 2^{38} such pairs from 2^{38} plaintexts using 2^{39} total time. In the sequel we treat each pair of corresponding ciphertexts (C, C') as if it is a "right pair" with respect to the main differential presented in Section 4.1.1 (i.e., assume that it satisfies all the conditions in the differential).

(b) Insert the 2^{38} ciphertext pairs into a hash table indexed by the difference in the three bytes 0,10,13. We note that if a pair is a right pair, then since $\Delta(I^8_{0,1,2}) = 0$, the ciphertext difference in bytes $(0, 13, 10)$ is equal to (d_0, d_1, d_2). Hence, the pairs are divided into 2^{24} sets according to the possible values of (d_0, d_1, d_2), and the attack is sequentially applied to each set (which contains $2^{38-24} = 2^{14}$ pairs on average).

(c) Note that by Observation C, only $127^3 \approx 2^{21}$ of the values (d_0, d_1, d_2) are possible, and hence the rest of the attack can be applied only to the $2^{24-3} = 2^{21}$ possible sets.

2. **First Filtering Step.** For each set of pairs corresponding to a possible value of (d_0, d_1, d_2), perform the following operations:

(a) Guess the byte K^8_{12} and partially decrypt the ciphertext pairs through the last round to get $\Delta(I^8_{12})$. (Note that $\Delta(K^8_{12}) = d_0 \oplus \alpha$ is now known to the adversary). Check whether the obtained difference $\Delta(I^8_{12})$ is possible (see Observation A). Half of the pairs are expected to pass this filtering.

(b) Use the difference $\Delta(I^8_{12})$ to find the differences $\Delta(I^8_{13,14})$ (see Observation A). Using the ciphertext difference in bytes 9,6, find the input and output differences to the SB operation in bytes 13,14. (Note that the corresponding key differences, $\Delta(K^8_6) = d_2$ and $\Delta(K^8_9) = d_1$, are now known to the adversary). Check whether this input/output difference pair is possible, and if it is possible, retrieve the corresponding actual values of the pairs (see Observation B). This is a two-bit filtering, and hence $2^{14} \cdot 2^{-1} \cdot 2^{-2} = 2^{11}$ pairs are expected to remain, and each pair suggests two pairs of actual values on average for each byte. Each such pair of actual values can be combined with the corresponding ciphertext pair to get a suggestion for the subkey bytes K^8_6 or K^8_9.

(c) Go over all the 2^{16} possible values of $K^8_{6,9}$ and check how many times each value is suggested. Discard all the values which are suggested fewer than four times. Note that the correct value is suggested by all the right pairs, and hence it gets at least four suggestions with probability about 0.57. On the other hand, the probability that a wrong subkey is suggested at least four times is approximately $\binom{2^{13}}{4} \cdot 2^{-64} = (2/3) \cdot 2^{-16}$. Hence, less than one subkey suggestion remains on average (If no subkey suggestions remain, which happens frequently, the set is discarded).

(d) Discard all the pairs that lead to a "wrong" value of $K^8_{6,9}$ (as we said before, all the right pairs suggest the correct value). Only a few wrong pairs (at most four) are expected to remain, and only these pairs are considered in the rest of the attack.

3. **Second Filtering Step.** This step is actually a repetition of Step 2 with a different column. For each remaining value of $(d_0, d_1, d_2, K^8_{12})$ and the corresponding remaining pairs, perform the following:
   (a) Guess the subkey byte K^8_8 and partially decrypt the ciphertext pairs through the last round to obtain $\Delta(I^8_8)$. Use this difference to retrieve $\Delta(I^8_{9,10})$ (or discard the pair if the difference is impossible). Using the ciphertext difference in bytes 5,2, find the input and output differences to the SB operation in bytes 9,10. Check whether this input/output difference pair is possible, and if it is, retrieve the corresponding actual values of the pairs and use them to get suggestions for the subkey $K^8_{5,2}$. Go over all the 2^{16} possible values of $K^8_{5,2}$ and check how many times each value is suggested. Discard all the subkey values that are suggested fewer than four times.
   (b) At this stage, the probability that a wrong subkey is suggested at least four times is extremely low (about 2^{-64}), and hence it is expected that all the wrong values of $(d_0, d_1, d_2, K^8_{12}, K^8_8)$ will be discarded. Thus, the adversary obtains the right pairs, and the correct values of d_0, d_1, d_2, and six of the sixteen bytes of K^8 (namely, $=K^8_{2,5,6,8,9,12}$).
4. **Retrieving the Rest of K^8**
   (a) For each remaining pair, guess the values c_3 and d_3. Using the known input and output differences of the SB operations applied to bytes 3,7,11,15 of round 8, obtain suggestions for the subkey bytes $K^8_{3,7,11,15}$. Discard all the values that are suggested fewer than four times. Since only four pairs are expected to remain at this stage, only the correct subkey value is likely to remain, along with the correct values of c_3 and d_3.
   (b) Repeat Steps 1,2,3 of the attack with the *shifted main differential* (presented in Section 4.1.1) instead of the main differential used until this step, and guessing the subkeys $K^8_{0,4}$ instead of $K^8_{8,12}$. This retrieves the subkey bytes $K^8_{0,1,4,10,13,14}$, which completes the derivation of K^8. Note that the time complexity of this step is significantly lower than the time complexity of Steps 1,2,3, since the correct values of d_0, d_1, and d_2 are already known to the adversary.
5. **Retrieving K^7.** At this stage the adversary already knows the full value of the last subkey K^8, and hence can peel off the last round. Note that the "ciphertexts" referred to in this part are actually the outputs of round 7.
   (a) For the right pairs with respect to the main differential, guess the value of $K^7_{0,4,8,12}$ and partially decrypt the ciphertexts through round 7 to obtain $\Delta(I^7_{0,4,8,12})$. Check whether $\Delta(I^7_{0,4,8,12}) = (\alpha, \alpha, \alpha, \alpha)$. If not, discard the guessed key bytes. Since all the right pairs suggest the correct subkey value, only the correct value is likely to remain. Note that since this step can be performed in each byte independently, its time complexity is negligible.
   (b) Use the key schedule to obtain two possible values for each of the bytes $K^7_{13,14,15}$. (In the key schedule algorithm, the column value $K^7_{12,13,14,15}$

is the input to SB operations for which both the input and output differences are known. Hence, two suggestions for the actual value can be obtained by Observation B).

(c) This step uses the *complementary differential* presented in Section 4.1.1. Consider any subset of 2^{26} pairs of plaintexts amongst the 2^{38} pairs used in the attack (using more pairs at this stage will be a waste of time). For each pair, assume that it is a right pair with respect to the complementary differential.

   i. Partially decrypt the ciphertext pairs through round 7 to obtain the differences $\Delta(I^7_{0,1,4,6,8,11})$. Note that the partial decryption is possible since the subkey bytes $K^7_{0,4,8,13,14,15}$ are already known to the adversary.

   ii. Consider each of Columns 0,1,2 of $\Delta(I^7)$ separately, and check whether the difference in the two "known" bytes (e.g., bytes 0,1 in Column 0) agrees with one of the column differences which are possible for right pairs w.r.t. the complementary differential (as explained in Section 4.1.1, there is a total of 256 such differences in each column). In each of the columns, this yields an 8-bit filtering, and hence about four pairs are expected to pass to the next step.

   iii. For each of the remaining pairs, use the known differences $\Delta(I^7_{0,4,8,12})$ to retrieve the full difference $\Delta(I^7)$ (see Observation A). Then, use the input and output differences to all the SB operations in round 7 to obtain two suggestions for the actual values in each byte, and use these suggestions to get two suggestions for each byte of the subkey K^7. Discard all the subkey values that are suggested fewer than three times. Since all the right pairs suggest the correct subkey value, it is easy to show that only the correct subkey values are likely to remain.

(d) At this stage the adversary knows the full values of K^7 and K^8, and hence she can find the original 256-bit key K by running the (invertible) key schedule of AES-256 backwards from the known values of these two consecutive subkeys.

The Complexity of the Attack. The data complexity is 2^{38} chosen plaintexts (composed of 2^{37} arbitrary plaintexts P along with their 2^{37} companions P'). It is easy to see that the most time-consuming step of the attack is Step 2(c), that takes $2^{21} \cdot 2^8 \cdot 2^{16} = 2^{45}$ simple table lookup operations. To make a fair comparison, we have to remember that each 9-round AES-256 encryption requires $9 \times 16 = 144 \approx 2^7$ SB operations, and thus the 2^{45} simple operations required to carry out the complete attack are likely to be faster than the 2^{39} encryption operations required to prepare all the ciphertexts. The RAM memory requirements of the attack are negligible.[2] The probability of success is about 57% (required for the success of Step 2(c)), but can be made arbitrarily high by using additional pairs.

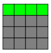
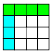
[2] The data can be stored on a hard disk, as we need to write it once, and read it once, while operating each time on a small amount of plaintext/ciphertext pairs which easily fit 1 MByte of memory.

4.2 A Related-Subkey Attack

If we relax the conditions on the key relation, a more efficient attack can be obtained. Recall that the XOR condition we imposed on the key in the previous attack directly implied the same XOR condition on the subkeys K^{-1} and K^0 which were used in the whitening phase and in the first round. In the new attack, we impose a XOR condition on the two subkeys K^0 and K^1 used in the first and second round. Let us define $\Delta(K^0) = (b, b, b, b)$ and $\Delta(K^1) = (a, 0, a, 0)$, where a and b are the same as in the previous attack. The key difference $\Delta(K)$ can then be defined by running the key schedule backwards as (f, a, a, a, b, b, b, b), where f is a full-column unknown difference.

A differential characteristic for 9 rounds, based on this key difference, is depicted in Figure 3 in the center. It contains 13 active S-boxes in the state. The input of the first four S-boxes and the output of the last four S-boxes are not specified, and the other S-boxes yield the desired values with probability 2^{-6}. Therefore, the plaintext difference $\Delta(P)$ is specified in 9 bytes, and the ciphertext difference $\Delta(C)$ is specified in 12 bytes.

The best attack runs in the chosen-ciphertext scenario as follows:

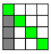
1. Prepare two structures of 2^{31} ciphertexts each, whose active bytes are located in row 0, and where byte 0 takes on only 128 possible values. The constants differ according to $\Delta(C)$ .
2. Ask for the decryption of the structures under K and K', respectively.
3. Select all the plaintext pairs that satisfy $\Delta(P)$ .
4. Every candidate pair proposes two candidates for each of six key bytes (K_4^{-1}), (K_8^{-1}), (K_{12}^{-1}), (K_{12}^0), (K_{14}^0), (K_{15}^0), from plaintext difference bytes 4, 8, 12, 3, 1, 2, respectively.

Let us compute the number of right pairs. Of the 2^{62} possible pairs, about 2^{30} pairs survive the 32-bit filter in the last round, producing a zero difference in I^7. The five S-boxes, whose inputs are specified, reduce the number of right pairs to one. The single right pair can be detected because the fixed 72-bit plaintext difference filters out all the 2^{62} wrong ciphertext pairs. The plaintext difference in bytes 1, 2, 3, 4, 8, 12 proposes two candidates for each one of the corresponding key bytes. As a result, we can recover 56 bits of the key using only 2^{32} time and 2^{32} chosen ciphertexts.

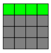
5 Attacks on 10 Round Variants of AES-256

In this section we describe two attacks on the 10-round variant of AES-256. Both of them are based on a key relation which is defined by imposing a fixed difference on two consecutive subkeys. We also have to start from an odd round so that the 10-round attack is run on rounds 1–10 of AES-256. We define $\Delta(K^2) = (b, b, b, b)$ and $\Delta(K^3) = (a, 0, a, 0)$, where a and b are the same values defined in the previous attacks. As a result, column 0 of $\Delta(K^1)$ and byte 0 of $\Delta(K^0)$ are not known to the adversary (see Figure 3, right).

5.1 Chosen-Plaintext Attack

A differential characteristic for 10 rounds, based on this key difference, contains 17 active S-boxes in the state. Compared to the 9-round differential characteristic, the last 7 rounds remain the same, and the input of three of the four active S-boxes in the second round is restricted. Therefore, 8 S-boxes behave as expected with probability 2^{-6}. The plaintext difference $\Delta(P)$ is specified in 12 bytes .

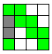
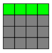
The algorithm used by the adversary is as follows:

1. Prepare 2^{16} structures of 2^{32} plaintexts each, whose active bytes are located on the main diagonal. The constants differ according to $\Delta(P)$ and so that the total number of distinct plaintexts is 2^{48}.
2. Encrypt all the structures under both K and K'.
3. Select all the ciphertexts pairs that satisfy $\Delta(C)$ , and whose plaintexts belong to the same structure.

Let us compute the number of right pairs. Of the 2^{80} possible pairs about $2^{80-24} = 2^{56}$ have zero difference in bytes 1,2,3 of I^2. The four S-boxes in round 2 can be used as a 26-bit filter, so 2^{30} pairs come out of the first two rounds. The next five S-boxes, whose inputs are specified, reduce the expected number of right pairs to one. The single right pair can be detected because the fixed 96-bit plaintext difference filters out all the 2^{80} candidate ciphertext pairs. As a result, we also get a distinguisher whose total complexity is 2^{48} data, 2^{49} time, and 2^{33} memory.

5.2 Chosen-Ciphertext Attack

In this attack we relax the input to one of the S-boxes in round 2. As a result, the plaintext difference is specified in 8 bytes only, so a chosen-ciphertext attack is more practical in terms of the data requirements. Our best attack runs as follows:

1. Prepare 2^{12} structures of 2^{32} ciphertexts each, whose active bytes are located in row 0, and decrypt all the texts with K. The constants differ according to $\Delta(P)$  and so that the total number of distinct plaintexts would be 2^{44}.
2. Apply to each one of these ciphertexts the difference $\Delta(C)$  and decrypt all of them with the related key K'.
3. Select all the plaintext pairs that satisfy $\Delta(P)$.
4. Every candidate pair proposes two candidates for each of five key bytes $(K^9_{12}), (K^{10}_0), (K^{10}_4), (K^{10}_8), (K^{10}_{12})$.

Let us compute the number of right pairs. Of 2^{76} possible pairs about 2^{44} pairs have zero difference in I^9. The seven S-boxes with input restrictions in rounds 2–10 provide a 42-bit filter, which reduces the number of right pairs to $2^{44-42} = 4$. The plaintext difference is specified in 64 bits, so $2^{76-64} = 2^{12}$ pairs come out of the last filter.

The wrong pairs are filtered at the bottom of the differential. We guess $\Delta(K^9_{12})$ and thus derive the full $\Delta(K^{10})$. Then each candidate pair proposes sixteen 32-bit key candidates, or 2^{16} candidates in total. The probability that four wrong pairs propose the same values is 2^{-32}, and is still very low when we combine all the guesses. As a result, we get two candidates for each of the five key bytes, which provide 35 bits of information about the key. The total complexity of this attack is 2^{44} data, 2^{45} time, and 2^{33} memory.

6 Attacks on Other Variants of AES-256

6.1 A Related-Key Distinguisher for 8 Rounds

The basic distinguishing attack in this case uses the simplified 8-round differential presented in Section 4.1.1. The attack itself is very simple: the adversary asks for the encryption of 2^{30} pairs of plaintexts with the input and key differences of the differential. For each pair, he checks whether the difference in bytes 1,5,9,13 of the ciphertext is equal to the known value of b_1, and whether the difference in bytes 2,6,10,14 of the ciphertext is equal to the known value of b_2. Since this is a 64-bit filter, for a random permutation all the pairs are likely to be discarded, while for an 8-round AES-256, one pair will remain with probability $1 - 1/e \approx 0.63$.

This efficient distinguishing attack was verified experimentally. We sampled 100 pairs of related keys (for a specific value of α and its corresponding β). For each such pair, we took 2^{32} random pairs with input difference (b, b, b, b), and encrypted them under the related keys. As the probability of the 7-round differential characteristic is 2^{-30}, the expected number of right pairs in each experiment is 4, where the actual number is distributed like a Poisson random variable with a mean value of 4. We compare our results and the expected values in Table 2.

We also did a second experiment, where the plaintexts where chosen in a counter mode manner. In each experiment, we picked at random two related keys and two

Table 2. The Number of Right Pairs in 100 Experiments

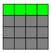
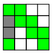
Right Pairs	0	1	2	3	4	5	6	7	8	9	10	11	12
Theory ($Poi(4)$)	1.8	7.3	14.7	19.5	19.5	15.6	10.4	6.0	3.0	1.3	0.5	0.2	0.06
Experiment 1 (random plaintexts)	0	10	18	10	28	18	6	8	1	0	0	0	1
Experiment 2 (counter mode)	1	3	17	19	23	18	4	4	8	2	1	0	0

IVs satisfying the input difference, and tried 2^{32} consecutive counters. In the 100 experiments, we encountered a distribution which follows the same Poisson distribution, thus proving that these attacks can be applied even when counter mode is used. The exact distribution is given in Table 2 as Experiment 2.

6.2 A Related-Subkey Attack on 8 Rounds

In this attack we consider AES-256 reduced to 8 rounds and starting from an odd round. We take the differential for the related-subkey attack on 9 rounds, cut the first round, and relax the input difference of the first two active S-boxes (Figure 3, left). As a result, the plaintext difference $\Delta(P)$ is specified in 8 bytes, and the ciphertext difference $\Delta(C)$ is specified in 12 bytes. There are three active S-boxes such that both their input and output differences are fixed.

The best attack runs in the chosen-ciphertext scenario as follows:

1. Prepare two structures of about $2^{25.5}$ ciphertexts, where bytes in row 0 are active, but do not run through all their possible values. The constants differ according to $\Delta(C)$ .
2. Encrypt structures with K and K', respectively.
3. Detect all the plaintext pairs that satisfy $\Delta(P)$ .
4. Every candidate pair proposes two candidates for each one of six key bytes (K_4^{-1}), (K_8^{-1}), (K_{12}^{-1}), (K_{12}^{0}), (K_{14}^{0}), (K_{15}^{0}).

Let us compute the number of right pairs. Of 2^{51} possible pairs, about 2^{19} pairs survive the 32-bit filter in the last round, producing a zero difference in I^7. The three S-boxes, whose inputs are specified, reduce the number of right pairs to two. All the 2^{61} wrong pairs are discarded by the 64-bit plaintext difference filter.

The two right pairs provide information on the five bytes of the last subkeys (see the attack on 9 rounds), which are recovered after we guess $\Delta(K_0^8)$. As a result, we recover 35 bits of the subkey with $2^{26.5}$ time, data, and memory complexity.

6.3 Related-Subkey Attacks on 11 Rounds

The related-subkey differentials can be extended in several ways to 11 rounds. However, the best attacks we get are beyond the practical 2^{56} bound, so we only briefly sketch the underlying ideas.

6.3.1 Differential

An 11-round differential (rounds 1–11) is obtained by adding one round at the end of the 10-round related-subkey differential (Figure 3, right). The final round is then similar to the last round of the 9-round related-key differential from Section 4.1.1. We are flexible in the number of active S-boxes in round 10 whose output differences are restricted. As we fix more S-boxes, we get a better ciphertext filter, but a lower overall probability of the differential. We can also change the parity of rounds and start from round 0 instead of round 1; this makes the scenario more practical but adds one more unknown byte difference to the plaintext.

6.3.2 Attacks

The following attacks can be applied to 11-round AES-256:

1. Start from an odd round and restrict the output difference of three active S-boxes in round 10. The data and time complexity would be about 2^{70}, and the last steps are done with non-trivial key ranking. About 50 bits of the subkey are recovered.
2. Restrict one more active S-box. Then the data and time complexities increase to 2^{76}, but it is easier to discard the wrong pairs.
3. Start from an even round (as in the original AES) and restrict three S-boxes in round 9 (former 10). The data complexity would be about 2^{70}, and the time complexity about 2^{75}.
4. Start from an even round and restrict two S-boxes. The data complexity drops to 2^{63}, but the time complexity increases to 2^{90} due to complicated filtering and key ranking.

7 The Choice of the Data and the Key Differences

The data and key differences in all the differentials we use depend on two byte values α and β, such that $\alpha \rightarrow \beta$ through the S-box holds with probability 2^{-6}. For each value of α there exists a unique such value β, and vice versa. There are no other restrictions on α and β in our attacks, and in fact it is even possible to use other values of α and β for which the differential $\alpha \rightarrow \beta$ through the S-box holds with probability 2^{-7}. However, such choices lead to slightly less efficient attacks.

By using this considerable freedom in the choice of α and β, the adversary can try to achieve several goals:

1. **Reducing the Hamming Weight of the Data and Key Differences.** Since in actual attacks the key difference is likely to be caused by applying physical faults to the encryption device, it is desirable to make the required changes as small as possible. The minimal possible Hamming Weight of the key difference is 24 bits, and it is obtained by taking $\alpha = 05_x, \beta = 08_x$ or $\alpha = 0A_x, \beta = 04_x$.
2. **Restricting the Plaintext Bytes to ASCII Characters.** Recall that in ASCII characters, the MSB in each byte is zero. Hence, if the plaintext difference used in the attack has value zero in the MSB of each byte, this increases the probability that if the initial plaintext consists of ASCII characters, the "modified" plaintext will also consist of ASCII characters. Such difference can be obtained by fixing the two MSBs of β to be zeros (e.g., $\beta = 08_x$).
3. **Adapting the Plaintext Difference to Numeric Characters.** By choosing $\beta = 01_x$, the plaintext difference in all its bytes is only in the two LSBs. As a result, we can choose numeric plaintexts whose modified versions also contain only numeric characters.

As a final comment, consider the case of an AES cryptosystem which is used in counter mode. Its plaintexts are defined by a fixed prefix, followed by a 64-bit counter. When we XOR our fixed difference to a sequence of 2^t such consecutive ciphertexts, we get another sequence of 2^t plaintexts which has the same structure (but with a different fixed prefix and a different counting order). Consequently, instead of repeatedly forcing the cryptosystem to encrypt our 2^t chosen plaintexts, we can just force it to start from two chosen starting points, and let the natural counting process in this mode of operation generate all the other ciphertexts we need in our attack.

8 Conclusions

This paper continues the deterioration in the security of AES which took place in 2009. The main problem seems to be the key schedule of AES-256, which is "not of industrial strength": It does not mix the initial key sufficiently, it is too linear, and as a result it has unusually long key differentials of probability 1. In addition, the similarity between the key schedule and the data encryption in AES makes it possible to repeatedly cancel data differences with corresponding key differences over many rounds. Ironically, the new attacks work best against AES-256 (which was supposed to be the strongest member of the AES family), and do not currently seem to work against AES-128.

The attacks described in this paper clearly have a practical time complexity. The number of chosen plaintexts they need is comparable to this time complexity, and can be considered practical when the adversary has the encryption device in his possession. The most problematic aspect of the attack is its reliance on related keys, which is not universally accepted as a practical attack model. However, we believe that it is important to consider such attacks for several reasons: First of all, resistance against the largest possible variety of attacks should be an essential part of the certification process for new ciphers, even if these attacks do not seem to pose any immediate risk. In addition, AES was specifically designed to resist such attacks (see [5]), and its failure to do so raises serious doubts about its overall security. Finally, implementors should be aware of the possibility that such attacks exist, since they may become practical in some particular modes of operation (e.g., when the block cipher is employed in a MAC which uses the chosen input blocks as keys), or when some key bits can be flipped with a laser beam in a fault attack. The most disturbing aspect of the new attacks is that AES-256 can no longer be considered as a safe black box construction, which can be dropped into any security application with little thought about how it is used.

Acknowledgements. Nathan Keller was partially supported by the Adams Fellowship and by the Koshland center for basic research. Dmitry Khovratovich is supported by PRP "Security & Trust" grant of the University of Luxembourg.

References

1. Biham, E.: New Types of Cryptanalytic Attacks Using Related Keys. J. Cryptology 7(4), 229–246 (1994)
2. Biham, E., Dunkelman, O., Keller, N.: Related-Key Boomerang and Rectangle Attacks. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 507–525. Springer, Heidelberg (2005)
3. Biryukov, A., Khovratovich, D.: Related-Key Cryptanalysis of the Full AES-192 and AES-256. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 1–18. Springer, Heidelberg (2009)
4. Biryukov, A., Khovratovich, D., Nikolic, I.: Distinguisher and Related-Key Attack on the Full AES-256. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 231–249. Springer, Heidelberg (2009)
5. Daemen, J., Rijmen, V.: AES proposal: Rijndael. NIST AES proposal (1998)
6. Daemen, J., Rijmen, V.: The Design of Rijndael. In: AES — the Advanced Encryption Standard. Springer, Heidelberg (2002)
7. Ferguson, N., Kelsey, J., Lucks, S., Schneier, B., Stay, M., Wagner, D., Whiting, D.: Improved Cryptanalysis of Rijndael. In: Schneier, B. (ed.) FSE 2000. LNCS, vol. 1978, pp. 213–230. Springer, Heidelberg (2001)
8. Fluhrer, S.R., Mantin, I., Shamir, A.: Weaknesses in the Key Scheduling Algorithm of RC4. In: Vaudenay, S., Youssef, A.M. (eds.) SAC 2001. LNCS, vol. 2259, pp. 1–24. Springer, Heidelberg (2001)
9. Kim, J., Hong, S., Preneel, B.: Related-Key Rectangle Attacks on Reduced AES-192 and AES-256. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 225–241. Springer, Heidelberg (2007)
10. Knudsen, L.R.: Cryptanalysis of LOKI91. In: Zheng, Y., Seberry, J. (eds.) AUSCRYPT 1992. LNCS, vol. 718, pp. 196–208. Springer, Heidelberg (1992)
11. National Security Agency (NSA). National Policy on the Use of the Advanced Encryption Standard (AES) to Protect National Security Systems and National Security Information (June 2003), `http://www.cnss.gov/Assets/pdf/cnssp_15_fs.pdf`
12. Tews, E., Weinmann, R.-P., Pyshkin, A.: Breaking 104 bit WEP in less than 60 seconds. In: Kim, S., Yung, M., Lee, H.-W. (eds.) WISA 2007. LNCS, vol. 4867, pp. 188–202. Springer, Heidelberg (2008)
13. Tsudik, G., Van Herreweghen, E.: On Simple and Secure Key Distribution. In: ACM Conference on Computer and Communications Security, pp. 49–57 (1993)

A Comments on Figures 2 and 3

Figures 2 and 3 depict differential characteristics that are used in the related-key attacks. Figure 3 deals with related-subkey attacks, when the differentials coincide in the last six rounds. For simplicity, we depict the common part of the related-subkey differentials only once for the 8-round differential characteristic. The other related-subkey differentials differ in the top rounds only, so we omit the 6.5-round part with seven active S-boxes.

In Figure 3 we also show how the data we start with is structured, and how the number of right pairs decreases during the encryption process. In the beginning of the attack, n STR above k T stands for n structures with k texts each. Similarly, k P stands for k right pairs, and n-bit F stands for an n-bit filter.

Colors. We extensively use colors while depicting differentials. In order to visually demonstrate the textual explanations, we provide the used color scheme.

🟩	Arbitrary difference
🟪	Fixed S-box input difference
🟦	Fixed S-box output difference
🟥	Key schedule: the S-box application to 🟪
▨ ▨ ▨ ▨	MixColumns expansion of 🟦
▧ ▧ ▧ ▧	Key schedule: the S-box application to ▨▨▨▨

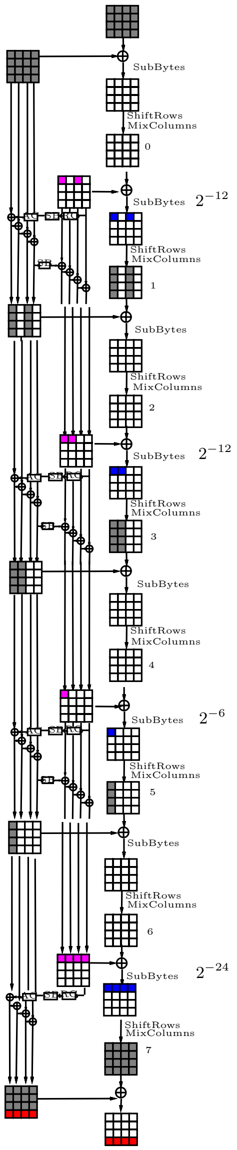


Fig. 2. Related-key differential for 8-round AES-256

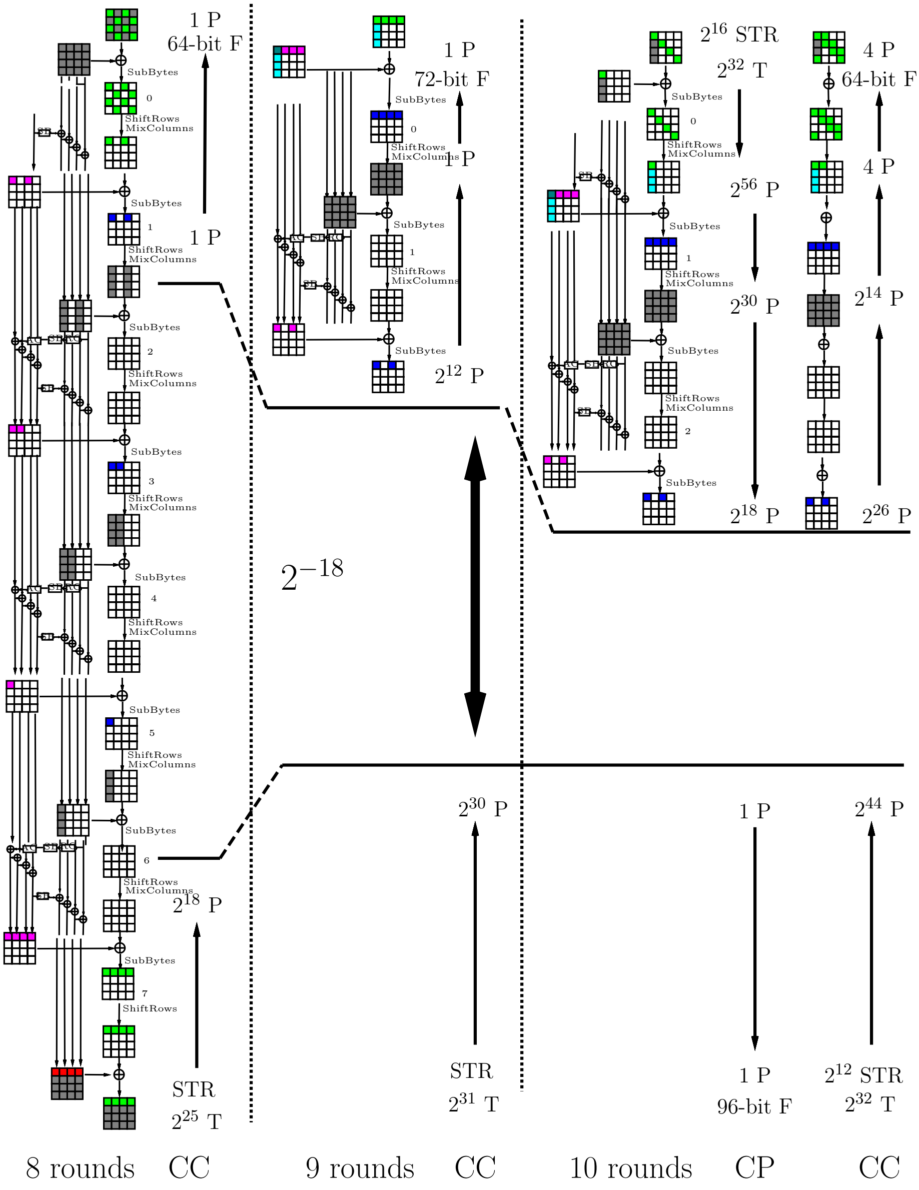


Fig. 3. Related-key attacks on 8-, 9- and 10-round AES-256

Cryptography between Wonderland and Underland

Moti Yung

Google Inc. and Department of Computer Science, Columbia University
moti@cs.columbia.edu

Abstract. Cryptography is a very broad field, interdisciplinary in nature, and connected to many other areas (in mathematics, computer science, computer systems and engineering). On the one hand, in theoretical cryptography many new notions have been defined, constructed and improved, especially new protocols and cryptosystems that are very powerful and surprising, including solving challenging and even seemingly paradoxical problems. On the other hand, cryptography is often required in actual computing systems, where the computing and communication infrastructure is very dynamic and evolves in a very fast pace. Thus, actual systems may need solutions that are highly constrained, non trivial, and not covered by merely combining existing cryptographic tools and protocols in a black-box fashion. These solutions are the subject of industrial development of specific cryptographic systems that are much less known than their theoretical counterparts. We discuss the interplay between theory of cryptographic protocols and actual industrial cryptographic systems, the differences in specifying, analyzing, modeling, designing and validating in each sub-area, as well as the similarity and the mutual influence between the two sub-areas.

A Tale of Two Sub-areas

Modern cryptography is famous for far reaching developments in many areas that are reported in theoretical and experimental papers. In particular, the area of designing public key cryptosystems and cryptographic protocol has been very fruitful, where amazing developments have been taking place. Working in this area, formalizing new problems, solving them and proving their security, then improving, refining and re-defining the problem is fascinating. Working this way, in fact, often feels like having an adventure in Wonderland (this is obviously said in an allusion to Lewis Carroll's "Alice's Adventures in Wonderland").

However, this area of cryptography is often criticized as being only theoretical. It is often believed by system designers that cryptographers finish their usefulness once they design ciphers and similar cryptographic functions (e.g., a hash function) and a few building block protocols (SSL/TLS, IPSEC). Systems researchers have said that these building blocks can then be deployed in systems by general system designers and essentially solve all problems. Ignoring the fact that it is typical to hear cross-field criticism among researchers, it is indeed the

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 320–321, 2010.

case that for various tasks, standard general cryptographic solutions do work. However, standard components usually serve well typical standard systems (for which system researchers may not be needed as well). For more sophisticated systems (e.g., fast and safe cryptography on specific hardware platforms such as smartcards) cryptographers need to be heavily involved (and they are).

The thesis of this presentation is that cryptographic protocol design of special-purpose industrial solutions is not different, and it needs cryptographic sophistication as well. This area presents challenges that require understanding of the underlying system, the available technologies, the system's goals and specifications, involved costs and financial risks, the relevant business goals, as well as threats and their implications. However, to get secure and safe solutions it requires involvement of cryptographic protocol designers. Note that even though this area is less publicized and is based quite often on an oral tradition (given its industrial nature), it is, nevertheless, highly exciting to develop these types of solutions. These solution need to satisfy a large set of constraints, such as having the right performance parameters, providing right level of usability, being cost effective, having robust engineering, and assuring "the right level" of security given the underlying working environment. Contributing to lasting solutions that employ its underlying cryptography correctly is indeed a fascinating area, and working in this area often feels like having an adventure in Underland (this is said in an allusion to another series of fantasy novels: Suzanne Collins' "The Underland Chronicles").

Working in both areas of cryptographic protocol design (the theoretical and actual) gives one a large spectrum of appreciation of what is the state of the art and what is actually required in systems. When designing "a system that incorporates cryptographic subsystem," there are unique advantages to a team that is aware of the state of the art of the theoretical cryptographic literature. Further, the mode of thinking about problems from their specifications and security requirements, the formalization of threat as an adversary arguments, the careful design, and the need to scrutinize it, which dominates the theoretical work, are, all, applicable to working on actual systems incorporating cryptographic components. Yet, theory alone is not enough; specific system knowledge (as described above) is a must, and working closely with the entire engineering team is also a must for achieving successful contributions. This enables adapting the best solutions necessary to a specific setting, based on the full range of the specialized system goals and constraints.

The presentation will cover case studies of actual systems, designed for different purposes and facing different threats. The influence of "theoretical thinking" will be argued, as well as the added value of "specialized systems thinking" which is beyond theory. The general notions of transferring ideas from theory to systems, as well as turning systems ideas to subjects of new theoretical studies will be presented as well.

Automatic Search for Related-Key Differential Characteristics in Byte-Oriented Block Ciphers: Application to AES, Camellia, Khazad and Others

Alex Biryukov and Ivica Nikolić*

University of Luxembourg
{alex.biryukov,ivica.nikolic}uni.lu

Abstract. While differential behavior of modern ciphers in a single secret key scenario is relatively well understood, and simple techniques for computation of security lower bounds are readily available, the security of modern block ciphers against related-key attacks is still very ad hoc. In this paper we make a first step towards provable security of block ciphers against related-key attacks by presenting an efficient search tool for finding differential characteristics both in the state and in the key (note that due to similarities between block ciphers and hash functions such tool will be useful in analysis of hash functions as well). We use this tool to search for the best possible (in terms of the number of rounds) related-key differential characteristics in AES, byte-Camellia, Khazad, FOX, and Anubis. We show the best related-key differential characteristics for 5, 11, and 14 rounds of AES-128, AES-192, and AES-256 respectively. We use the optimal differential characteristics to design the best related-key and chosen key attacks on AES-128 (7 out of 10 rounds), AES-192 (full 12 rounds), byte-Camellia (full 18 rounds) and Khazad (7 and 8 out of 8 rounds). We also show that ciphers FOX and Anubis have no related-key attacks on more than 4-5 rounds.

Keywords: Cryptanalysis tool, search for best differential characteristics, related-key attack, open key, AES, Camellia, Khazad, Anubis, FOX.

1 Introduction

Proving security of modern block ciphers against differential [6] and linear cryptanalysis [28] has become a well understood and relatively simple task. Many of the modern ciphers are constructed as so-called substitution-permutation networks (SPN) — they consist of layers of non-linear substitution boxes (S-boxes) and diffusion layers built from linear or affine functions. The designer simply has to use diffusion layers with high (or maximal) branch number which is typically achieved by using maximum distance separable matrices [13]. Using such diffusion layers one can prove lower bounds on the number of active S-boxes

* This author is supported by the Fonds National de la Recherche Luxembourg grant TR-PHD-BFR07-031.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 322–344, 2010.

for a certain number of internal rounds. The designer then picks the number of rounds for which the probability of the best differential or linear characteristic is lower than 2^{-k} where k is the key size of a cipher. The resultant cipher is then provably secure against standard differential and linear attacks.

Such reasoning however holds only in the single key model, and does not extend to the case of related-key attacks [4]. In this class of cryptanalytic attacks the attacker knows or chooses the relation between several keys and is given access to encryption/decryption functions with all these keys. The goal of the attacker is to find the actual keys. The relation between the secret keys is a function chosen by the attacker with some extra care taken to avoid trivial attacks, and quite often it is just a XOR with a chosen constant. Security of most modern block ciphers against related-key attacks still relies on heuristic and ad hoc arguments. This situation is very similar to the heuristic security that we have for the modern hash functions, which is due to a lack of proper tools and methodologies for the analysis of differentials of non-bijective functions.

In this paper we make a step in the direction of provable security of modern block ciphers (and by analogy of modern hash functions), by presenting an efficient tool that can evaluate and help to prove bounds for the security of block-ciphers (hash functions) against differential related-key (open-key or chosen message) attacks.

Automatic search for best differential characteristics and linear approximations in a single key scenario was first performed by Matsui [29] for DES. Algorithms for automatic search of differential characteristics for MD4 were presented in [34,15], and for MD5 in [35]. De Cannière and Rechberger in [11] described a method that finds characteristics in SHA-1 in an automatic way and produced the best known collision trails for SHA-1. A typical problem that arises when trying to construct a tool for automatic search of characteristics is the size of the search space. The search space is exponential in the size of the block and the key which makes straightforward approaches infeasible for 128-bit block 128-256 bit key ciphers[1]. Therefore often the most important task for producing an efficient tool is the reduction in the size of the internal state by using some equivalent representation of the state, but with smaller size.

In that respect it is natural to look at byte (or word)-oriented ciphers which constitute a large fraction of modern ciphers. A natural compact (sometimes known as truncated) representation would shrink each byte into a single bit, representing by 0 a byte without difference and by 1 a byte with a difference. In such representation 16-byte block state, 16-32 byte key would translate into 16 and 16-32 bits respectively. These numbers are low, and give hope that a search of the whole $2^{32} - 2^{48}$ space of related-key differential characteristics might be possible. The main problem with this representation is a very heavy branching which will happen in the linear diffusion layers and on the XORs. Such representation alone will only allow to search for the most basic and short characteristics which happen with probability close to 1.

[1] Note that full search was feasible for 64-bit block cipher DES, due to its Feistel structure, which reduces the search space to about 2^{32}.

Our Contribution. Our goal is to perform a full search for related-key differential characteristic and to be able to find or to prove the non-existence of characteristics similar to those that were used in the recent attack on AES-256 [10]. In this paper we achieve this goal for all versions of AES and for several other ciphers. At the basis of our related-key search algorithm, further denoted as a *tool*, lies Matsui's approach for search of the best differential characteristics, with several important modifications. Depending on the key schedule of a cipher, we differentiate three classes of block ciphers. This is done to improve the efficiency. For each of the classes we introduce a special modification to Matsui's algorithm to obtain the final tool. The internal representation of the difference in a cipher (state and subkeys) plays a very important role for constructing a feasible tool. Using only compact representation may lead to a high branching (caused by XORs or other linear-diffusion transforms such as MixColumns in AES) when trying to build all possible one round characteristics. We completely eliminate the branching in the state of a cipher by using a special representation that takes into account the properties of the matrix used in the linear-diffusion layer. The related-key differential characteristics produced by our tool, fix only the positions of the active bytes[2]. To produce standard differential characteristics, i.e. characteristic with exact values of the differences in the active bytes, one has to fix the byte differences corresponding to the possible transitions of the differences trough the S-boxes.

We apply the tool to different byte-oriented block ciphers. The tool finds related-key characteristics for the full-round ciphers, or if such characteristics do not exist, for the maximal number of rounds for which they exist. We provide the best possible differential characteristics for all the versions of AES. For AES-128 it is on 5 rounds out of 10 (this also means that AES-128 is secure against straightforward related-key attacks after 6 rounds). For AES-192 it is on 11 rounds – just one round short of the total 12 rounds. The characteristic for AES-256 is on all 14 rounds, and it is the same characteristic (and the only one on 14 rounds) that was given in [10]. Then we present boomerang attack on AES-128 reduced to 7 rounds, and improve the complexity of the best attack on AES-192 by a factor 2^7. We analyze the version of Camellia without the FL functions, and where the rotation constants in the key schedule are multiplies of 8. For this "byte-Camellia", the best related-key differential characteristic is on 8 rounds (out of 18). Additionally, we launch a chosen-key attack and find a characteristic on all 18 rounds of byte-Camellia. For Khazad first we find a related-key characteristic on 7 rounds (out of 8), and then we show a boomerang attack on 7 rounds, and a chosen-key attack on the full-round Khazad. For the ciphers FOX and Anubis, we show that related-key differential characteristics cannot exist on more than 4-5 rounds. The summary of our results is given in Tables 1,2. Due to space limitations, we will not describe the ciphers that we

[2] Note that while our characteristics allow certain flexibility in the values due to the compact representation of differences – they are not *truncated differentials* since our goal is to find fully specified differential characteristics, rather than just truncated characteristics.

Table 1. Summary of attacks on the ciphers examined in the paper

Cipher	Attack/Result	Rounds	Data	Workload	Reference
AES-128	Collisions	7	2^{32}	2^{128}	[16]
	Partial sum	7	$2^{128} - 2^{119}$	2^{120}	[14]
	Impossible diff.	7	$2^{112.2}$	$2^{117.2}$	[26]
	Boomerang - RK	7	2^{97}	2^{97}	Section 3.2
AES-192	Rectangle - RK	9	2^{64}	2^{143}	[18]
	Rectangle - RK	10	2^{125}	2^{182}	[22]
	Boomerang - RK	12	2^{123}	2^{176}	[9]
	Boomerang - RK	12	2^{116}	2^{169}	Section 3.3
AES-256	Rectangle - RK	10	2^{114}	2^{173}	[5,22]
	Subkey Diff.	10	2^{48}	2^{49}	[8]
	Differential - RK	14	2^{131}	2^{131}	[10]
	Boomerang - RK	14	$2^{99.5}$	$2^{99.5}$	[9]
Camellia-128	Impossible	11	2^{118}	2^{126}	[27]
byte-Camellia-128	Chosen-key dist.	18	$2^{6\cdot 17}$	$2^{6\cdot 17}$	Section 4.2
Khazad	Slide attack[a]	5	2^{98}	2^{104}	[7]
	Integral	5	2^{64}	2^{91}	[31]
	Boomerang[a] - RK	7	2^{50}	2^{50}	Section 5.2
	Chosen-key dist.	8	2^{55}	2^{55}	Section 5.3

[a] The attack works for a weak key class, and the workload includes the effort to find related keys from the class.

Table 2. The upper bounds on the probabilities of the related-key differential characteristics for full or round-reduced ciphers examined in the paper. The probabilities for a higher number of rounds are below 2^{-k}, where k is the key size.

Cipher	Rounds	Workload	Section
AES-128	5	$2^{6\cdot 17}$	3.2
AES-192	11	$2^{6\cdot 31}$	3.2
AES-256	14[b]	2^{131}	3.2
byte-Camellia-128	8	$2^{6\cdot 19}$	4.1
Khazad	7	$2^{5\cdot 19}$	5.1

[b] The same characteristics as in [10].

analyze, we refer the reader to [13,1,3,20,2], and will use the original notation proposed by the designers in these papers.

2 A Tool for Search of Related-Key Differential Characteristics

Related-key differentials, introduced by Biham in [4], unlike the traditional single-key differentials that have difference only in the plaintext, have difference in the key as well. A related-key differential is specified with two input differences: Δ_P in the plaintext and Δ_K in the key, and an output difference Δ_C

in the ciphertext. A pair of plaintexts (P_1, P_2) and a pair of keys (K_1, K_2) follow the related-key differential in the cipher $E_K(P)$, if $P_1 \oplus P_2 = \Delta_P, K_1 \oplus K_2 = \Delta_K$ and $E_{K_1}(P_1) \oplus E_{K_2}(P_2) = \Delta_C$. A popular technique to find lower bounds on the probability of differentials is via finding probabilities of the best differential characteristics. A related-key differential characteristic besides the differences in the key, plaintext and ciphertext, also fixes the differences in the state and the subkeys after each round of the cipher.

There are a couple of approaches to construct a tool for search of the best round-reduced (related- or single-key) differential characteristics. One approach is using dynamic programming. Let $\Delta X_i, \Delta Y_j, \Delta Z_k$ be the differences only in the plaintext in the case of single-key, or in both the plaintext and the subkeys in case of related-key differentials. First, all one round characteristics $\Delta X_i \rightarrow \Delta Y_j$ are built, i.e. the attacker tries all possible starting differences X_i, and for each of them goes through one round of the cipher and obtains the differences Y_j. Distinct starting differences X_{i_1}, X_{i_2} can produce the same difference Y_j. For each Y_j only the characteristics, that have the highest probability are left. Next, the attacker builds again all one round characteristics $Y_j \rightarrow Z_k$, for different Y_j (but only those Y_j that were obtained in the first step). Again, for each Z_j he selects only the characteristics that have the highest probability. As a result, he had built the optimal two-round characteristics $X_i \rightarrow Y_j \rightarrow Z_k$. This procedure is repeated until the target n-round differential characteristic is built. The time complexity of the dynamic programming approach is linear in the number of one round characteristics, but it requires a lot of memory for storing the intermediate round values $\Delta Y, \Delta Z$, etc. That is why we will use the second approach, similar to the one used by Matsui in [29] for finding the best differential and linear characteristics in DES. It requires relatively small memory and the time complexity highly depends on the probability of the best round-reduced differential characteristics. The algorithm works by induction: to find the best n-round characteristic first it finds the best $1, 2, \ldots, n-1$ round characteristics. At some stage it requires building all one round characteristics – their number depends on the size of the search space. Thus a straightforward application of Matsui's search to modern ciphers would immediately fail but there is some hope for byte-oriented ciphers if one switches to compact representations in which each byte is replaced by a single bit: a byte with a difference, also called an *active* byte, is replaced by 1, a byte without a difference — by 0. This means that the difference in n-byte cipher can be represented as n-bit vector.

The compact representation seems optimal, yet several improvements to Matsui's algorithm are still required. Almost all byte-oriented ciphers are designed as substitution-permutations networks (SPN), i.e. they have a layer of S-boxes (S-layer) and a linear diffusion layer – a simple multiplication of the input by a matrix A (P-layer). When the S-boxes are bijective (a property common to most ciphers developed in the last 15 years), then an active byte stays active (and vice-versa) before and after the S-box. Hence, the S-layer does not alter the compact representation. On the other hand, the P-layer can change the number of the active bytes as well as their positions (depending on the exact values of the differences in the active bytes, and the branch number of the matrix A) and therefore it introduces a branching. Thus, besides the traditional compact representation, further denoted as S-value, we will introduce additional representation, called P-value. Indeed, the difference in n-byte cipher will be represented as $2n$-bit vector, where the first n coordinates (bits) are the S-value coordinates, and the next n are the P-value coordinates. The P-value of a difference is obtained when S-value goes through a P-layer and it is the same as the previous S-value (see Fig.1 for clarification). For example, in AES, if the value of a difference of some column is (0,1,0,0,0,0,0,0) (i.e. there is a difference only in the second byte of the column, the difference is of a type $(0, x, 0, 0)^T$) before the MixColumn, then after the MixColumn it is (0,0,0,0,0,1,0,0) meaning: $A(0, x, 0, 0)^T$, where A is the MixColumn matrix and x is an arbitrary non-zero byte value (i.e. it is a four-byte difference, obtained when some column with a difference only in the second byte was multiplied by the MixColumn matrix). Note that the representation can always be reduced to only S-value (although often not uniquely). For example, the above vector (0,0,0,0,0,1,0,0) can be represented as (1,1,1,1,0,0,0,0). The P-values reduce the branching as well: it is better to XOR two P-values, then to reduce them to only S-values and then XOR them. For example, if we XOR two differences (0,0,0,0,0,1,0,0) and (0,0,0,0,0,1,0,0) then the result can be (0,0,0,0,0,1,0,0) or (0,0,0,0,0,0,0,0). On the other hand, if we first reduce them to only S-values, then we will get the values (1,1,1,1,0,0,0,0) and (1,1,1,1,0,0,0,0). Obviously, XOR of these two values gives 2^4 possible outputs. In the states of the ciphers, after each transform, we will have either only non-zero S-value or only non-zero P-value of a difference (but never both), and hence we can effectively eliminate any branching in the state (the branching goes into the key). However even in such representations the search space of 128-bit block, 256-bit key cipher would be 16+32 bits, i.e. 2^{48}. Another complication is that if one would like to search for differential characteristics rather than truncated differentials one will need to pay in heavy branching at every XOR operation both in the state and in the key-schedule, which makes the search completely infeasible. Hence, depending on the key schedule, we would like to propose different variants of the tool to solve these problems:

1. The first variant is the original Matsui's approach itself. It applies to ciphers that *have minimal branching in the key schedule, with subkeys consecutively obtained one from another*. This means that once the difference in the subkey K_i is fixed, the difference in the subkey K_{i+1} can easily and almost uniquely be determined. Let $\Delta X \rightarrow \Delta Y$ be one round differential characteristic, where ΔX is the input difference in **both the state and the subkey**, and ΔY is the output difference, and let $W(\Delta X \rightarrow \Delta Y)$ be the weight function of this characteristic – the probability cost required to produce a pair that follows the characteristic (the exact definition of W is given later). Let $W_1, W_2, \ldots, W_{n-1}$ be the weights of the best $1, 2, \ldots, (n-1)$-round characteristic found previously with the algorithm and let $\widetilde{W_n}$ be the weight of

some (not necessarily optimal) n-round characteristic D_n. The search for the best n-round characteristic in pseudo code is described in Alg. 13. In short, first the algorithm builds all possible one round characteristics with a weight at most $\tilde{W_n} - W_{n-1}$. This constraint is introduced to filter some of the one round characteristics: if the weight of the first round is more than $\tilde{W_n} - W_{n-1}$ then it can not be extended to an n-round characteristic because the weight of $n-1$ rounds is at least W_{n-1} so in total it will have a weight more than the previously found characteristic of weight $\tilde{W_n}$. Each of the good one round characteristics (the one that pass the filter) is extended (when possible) to n rounds by the NextRound procedure. One call of this procedure extends the characteristic by one additional round. Again, it extends only the characteristics that satisfy the weight condition, by checking if the sum of the weights of the r and $n-r$ characteristics is not greater than the weight $\tilde{W_n}$ of the already known differential D_n.

2. The second variant of the tool is for ciphers that *have possibly high branching in the key schedule, with subkeys consecutively obtained one from another*. A good example of this type of key scheduling is the one in AES (subkey K_{i+1} is obtained from K_i in one iteration, but due to XORs in the key schedule, there is a lot of branching). If we try to apply the variant 1 of the tool to this type of ciphers we would have to build all one round characteristics (with differences in the state and the subkey). Yet, the high branching in the subkey, blows the number of characteristic out of proportion, and the search becomes infeasible. That is why we have to modify the tool for this special case of ciphers. Let ΔS_r be the difference in the state of round r after the XOR of the subkey K_r and let ΔK_r be the difference in this subkey. To add one more round to this characteristic one can proceed as follows:

   - take ΔS_r and go through all one round transformations of the state to build $\Delta\tilde{S}_{r+1}$ which is the difference in the state of round $r+1$ just before the XOR of the subkey K_{r+1}
   - take any ΔS_{r+1}
   - XOR $\Delta\tilde{S}_{r+1}$ and ΔS_{r+1} to produce ΔK_{r+1}
   - check if ΔK_{r+1} can be obtained from ΔK_r in one round.

   This way, instead of building all one round characteristics in the subkey, we only have to check if some subkey difference can be transformed to another difference in one subkey round (see Fig. 1). The number of these transitions that has to be checked is related to the size and branching of the state. Usually the state has smaller size than the subkey, and with the right representation it can have minimal or no branching, leading to a feasible search. Let ΔP be the plaintext difference, $\Delta S_r, \Delta K_r$ be the difference in the state and the subkey of round r, $\Delta\tilde{S}_r$ the difference in the state of round r just before the subkey XOR, $W(\Delta S_r \rightarrow \Delta\tilde{S}_{r+1})$ the probability of the characteristic $\Delta S_r \rightarrow \Delta\tilde{S}_{r+1}$. The notions of W_i are the same as in the previous variant. For the sake of clarity we assume there is no whitening key. The variant 2 of our search tool is described in Alg. 2.

3. The third variant of the tool applies to ciphers that *have key schedule with subkeys that are not successively obtained one from another*. Usually, the key schedule of these ciphers applies heavy transformations to the master key to obtain another key, and then combines these two keys (often with linear transformations) to get all subkeys. To build the tool we will use the following strategy: 1) from the master key, obtain all the subkeys, and 2) apply the first variant of the tool – build the characteristics for the state, but use the obtained subkeys (instead of building characteristics for subkeys). Let $\Delta X \xrightarrow{\Delta K} \Delta Y$ denote one round characteristic where $\Delta X, \Delta Y$ are the differences in the initial and final states, and ΔK is the subkey used in that round. The rest of the notions are the one used in variant 1. The pseudo code of the third variant is given in Alg. 3.

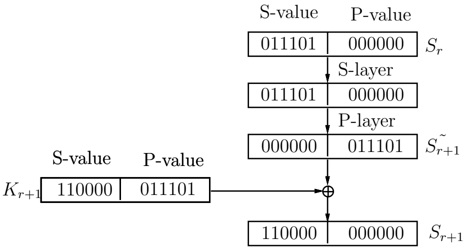


Fig. 1. The variant 2 of the tool with S- and P-value representations

Algorithm 1. Search of n-round differential characteristics - Variant 1

```
for all $\{\Delta X \to \Delta Y | W(\Delta X \to \Delta Y) + W_{n-1} \le \tilde{W_n}\}$ do
   Call NextRound($\Delta Y$, $W(\Delta X \to \Delta Y)$, 2)
end for

NextRound($\Delta Y$, $w$, $r$)
for all $\{\Delta Z | \Delta Y \to \Delta Z$ and $W(\Delta Y \to \Delta Z) + w + W_{n-r} \le \tilde{W_n}\}$ do
   if $r = n$ then
      Update $D_n$
      $\tilde{W_n} \leftarrow w + W(\Delta Y \to \Delta Z)$
   else
      Call NextRound($\Delta Z, w + W(\Delta Y \to \Delta Z), r + 1$)
   end if
end for
```

Now, let us determine the weight function $W(\Delta_1 \to \Delta_2)$ of the one round characteristics $\Delta_1 \to \Delta_2$. In the attacks on AES [10,9], the attacker pays only for the active S-boxes (active bytes that go through S-boxes) in the state and the subkey in each round of the cipher. Hence, we will use the same definition: $W(\Delta_1 \to \Delta_2)$ is defined as the number of active S-boxes in the state and the subkey in the one round characteristic $\Delta_1 \to \Delta_2$.

Algorithm 2. Search of n-round differential characteristics - Variant 2

```
for all $\Delta P, \Delta S_1$ do
  Obtain $\Delta \tilde{S_1}$ from $\Delta P$
  $\Delta K_1 = \Delta \tilde{S_1} \oplus \Delta S_1$
  Call NextRound($\Delta S_1, \Delta K_1, W(\Delta P \rightarrow \Delta \tilde{S_1})$, 2)
end for

NextRound($\Delta S_{r-1}, \Delta K_{r-1}, w, r$)
Obtain $\Delta \tilde{S_r}$ from $\Delta S_{r-1}$
if $W(\Delta S_{r-1} \rightarrow \Delta \tilde{S_r}) + w + W_{n-r} \leq \tilde{W_n}$ then
  for all $\Delta S_r$ do
    $\Delta K_r = \Delta \tilde{S_r} \oplus \Delta S_r$
    if $r = n$ then
      Update $D_n$
      $\tilde{W_n} \leftarrow w + W(\Delta S_{r-1} \rightarrow \Delta \tilde{S_r})$
    else
      Call NextRound($\Delta S_r, \Delta K_r, w + W(\Delta S_{r-1} \rightarrow \Delta \tilde{S_r}), r + 1$)
    end if
  end for
end if
```

Algorithm 3. Search of n-round differential characteristics - Variant 3

```
for all master $\Delta K|$ obtain subkeys $\Delta K_1, \ldots \Delta K_n$ with weight $W_K$ do
  for all $\{\Delta X \xrightarrow{\Delta K_1} \Delta Y | W(\Delta X \xrightarrow{\Delta K} \Delta Y) + W_K + W_{n-1} \leq \tilde{W_n}\}$ do
    Call NextRound($\Delta Y$, $W(\Delta X \xrightarrow{\Delta K} \Delta Y)$, 2)
  end for
end for

NextRound($\Delta Y$, $w$, $r$)
for all $\{\Delta Z | \Delta Y \xrightarrow{\Delta K_r} \Delta Z$ and $W(\Delta Y \xrightarrow{\Delta K_r} \Delta Z) + w + W_K + W_{n-r} \leq \tilde{W_n}\}$ do
  if $r = n$ then
    Update $D_n$
    $\tilde{W_n} \leftarrow w + W(\Delta Y \xrightarrow{\Delta K_r} \Delta Z) + W_K$
  else
    Call NextRound($\Delta Z, w + W(\Delta Y \rightarrow \Delta Z), r + 1$)
  end if
end for
```

When searching for n-round differential characteristic, the upper bounds on the weight of these characteristics are limited by the maximal number of active S-boxes that a characteristic can have. These upper bounds, depend on the key size and the difference propagation probability of the S-boxes which is usually 2^{-7} (sometimes 2^{-6} or even 2^{-5}). The weight of the n-round differential characteristic for a cipher with k-bit key, and S-boxes with maximal difference propagation probability 2^{-l} is upper bounded by $\lfloor \frac{k}{l} \rfloor$. The related-key differential characteristics produced by the tool have fixed positions of the active bytes, while the exact values are undefined. To produce standard differential characteristics, one has to find the exact values of the active bytes (the differences in the active bytes). The probability of the standard characteristics may be lower than the one predicted by the tool, but never higher, because the tool assumed that all active S-boxes hold with maximal differential probability, while in practice (in the case of standard characteristic) some S-boxes may hold with lower probability.

A new class of attacks, presented in [23,10], called *open-key* attacks, gives the attacker the full freedom of *knowing* or even *choosing* the key. In return, the attacker has to demonstrate some non-trivial property of the cipher which differentiates it from an ideal cipher. The motivation behind these attacks is that ciphers are often used as building blocks for some other cryptographic primitives, such as hash functions. There, the attacker has a full freedom of choosing all input parameters. An interesting approach is applicable to all ciphers in the chosen-key attack model. We call this approach *divide-and-conquer* technique. Let us have some related-key differential characteristic for a cipher. Since we control both the key and the state (it is a chosen-key attack), we can find a good pair of keys and states that follow the characteristic by the following method: 1) first find a good pair of keys that follow the differential characteristic only in the key, 2) once the subkeys are fixed, find a good pair of plaintexts that follow the differential characteristic in the state. It means we can split the whole characteristic in two halves: the one in the key, and the one in the state, and *instead of multiplying their probabilities, we can add them.* We will launch chosen related-key differential attacks on the full-round ciphers, in the cases when (secret) related-key characteristics do not exist. Note that proving the resistance against the chosen related-key differential attacks is still an open problem because it is unclear how to estimate the upper bound on the weighs of these characteristics since the number of rounds that can be covered for free varies from 1 in the rebound attack [30], 2 in the Super-Sbox [17], and even more in the tool of Khovratovich et al [21].

3 AES

The 128-bit block version of Rijndael[13] has been standardized by NIST as Advanced Encryption Standard (AES) in November 2001 [32]. It supports three different key sizes: 128, 192, and 256 bits, denoted as AES-128, AES-192, and AES-256, respectively. Various cryptanalytic results were published on AES, and until recently, the best attacks presented non-random properties of 7/10/10 rounds (out of 10/12/14 rounds) of AES-128/192/256 [14,16,23,18,22]. A breakthrough in analysis of AES have been the results [10,9]. In [10] a related-key attack on all 14 rounds of AES-256 was presented. In [9], boomerang attacks on full-round AES-192 and AES-256 were shown.

AES is an SPN cipher. The subkeys are generated consecutively one from another, but there is a lot of branching caused by the XORs of columns in the key schedule. Hence, *we will use variant 2 of the tool.* The state goes through four transformations: S-box layer, ShiftRows, linear-diffusion layer called MixColumns, and XOR of the key. In the tool we will use the following optimal representation of the state trough one round: the beginning state (before the S-boxes) can have non-zero only S-value (but zero P-value), after the S-box layer and after ShiftRows has again only non-zero S-value, after the MixColumns has non-zero only P-value, and after the subkey XOR again it has only non-zero S-value. This way, there is no branching in the state. The subkeys then can be determined as a XOR of a state of only P-value (the one after MixColumns) and a state of only S-value (the next round state, just before the S-boxes), hence they have columns that can have both non-zero S- and P-values. To use variant 2 of the tool we would have to be able to determine if the difference in the subkey K_{i+1} can be obtained from the difference in K_i. One subkey round consists of XOR of columns, application of S-boxes, and rotation of a column. If we represent the columns of the subkeys simply with only S-value, all of the above transforms can be easily checked. Therefore, each column of the subkeys is reduced only to S-value: 1) convert P-value into S-value, 2)XOR the obtained S-value with the initial S-value. Note, reduction to S-value as well as the XOR introduce branching, but the search is still feasible.

3.1 Best Round-Reduced Differential Characteristics for AES

We have applied the tool to all three versions of AES. The maximal difference propagation of the S-box in AES is 2^{-6}. Since the key sizes are 128,192, and 256, we can allow no more than 21, 31, and 42 active S-boxes in the characteristics for AES-128, AES-192, and AES-256, respectively. For AES-128 we found differentials characteristics on 4 rounds with 13 active S-boxes, and on 5 rounds with 17 active S-boxes. *In AES-128 there are no 6-round related-key differential characteristics.* For AES-192 we found differential characteristics up to 11 rounds out of 12. The characteristic on 11 rounds has 31 active S-boxes (20 in the state, and 11 in the key). For AES-256 we found unique differential characteristic on all 14 rounds, but this is the same characteristic that was presented in [10]. The characteristic from [10] is optimal for 9-14 rounds, but we have found better characteristic on 8 rounds (10 active S-boxes instead of 14). The 5-round differential characteristic for AES-128, and 11-round for AES-192 are presented in Fig. 2 in the Appendix. Regarding chosen-key attacks, in all versions of AES, there are no differential characteristic on 10 rounds with 21 or less active S-boxes in the state.

3.2 Related-Key Boomerang Attack on 7-Round AES-128

Let us show a boomerang attack on 7 rounds of AES-128. We will use two 3-round differential characteristics: a top 3-round truncated differential characteristic (4-1-4-16) with no key difference, and a 3-round related-key bottom differential characteristic with 5 active S-boxes in the state and 1 in the subkeys. They are presented in the Figure 4 in the Appendix. Note that if we extend the bottom characteristic for one additional round then the difference in the ciphertexts is fixed in 9 bytes, 4 bytes have equal difference and 3 bytes have a random difference. The difference δ between these two ciphertexts can have $2^{4\cdot 7} = 2^{28}$ distinct values (1-bit of freedom is lost for each of the 4 active S-boxes, since given a fixed input difference only 2^7 output differences are possible). Let Δ be the difference between K^4 and let the key schedule transform this difference into Δ' in K^7. Instead of guessing 2^{28} possibilities of the bottom difference for each ciphertext (which would increase the pressure on our filters) we guess 31 bit of the key: seven bits of $k^6_{1,3}$ and full $k^7_{1,1}, k^7_{1,2}, k^7_{1,3}$. This guess allows us to work on both faces of the bottom characteristic, since unlike in most related-key boomerang attacks our boomerang has only two related keys, instead of four.

The attack works as follows: For each guess of 2^{31} bits of the key

1. Prepare a structure of plaintexts P_i with all the possible 2^{32} four byte values on the main diagonal and the other bytes fixed.
2. Enrypt all the plaintexts P_i with the secret key K and obtain ciphertexts C_i.
3. For each ciphertexts C_i compute the correct difference δ using the 31-bit key guess, and obtain $D_i = C_i \oplus \delta$.
4. Decrypt all D_i with the key which is computed from the last subkey: $K^7 \oplus \Delta'$ and obtain plaintexts Q_i.
5. Sort all Q_i by 12 non-diagonal bytes. Pick only the pairs (Q_i, Q_j) that have zero difference in these 12-bytes. If none are found then goto 1.
6. Check the candidate quartet against 8 active S-boxes at the top (four on both sides of the boomerang) which gives an 8-bit filter.
7. Do the key counting step with the remaining quartet candidates.

Let us calculate the data and time requirements of the attack. A pair of plaintexts passes the first round with a probability 2^{-22} (MixColumns from four to one active byte, the position and the value of the active byte is irrelevant). The next two rounds are passed with probability 1, so in the third round with have a pair of states with all bytes active. In the second characteristic, from the bottom up assume that the initial 31-bit guess was correct, then in the next three rounds we have five active S-boxes which hold with probability 2^{-30} (when each is 2^{-6}). Yet for the two pairs of ciphertexts we only need the same difference after the fourth round (from bottom up) and therefore the two S-boxes of this round can be counted only once (on the one side of the boomerang the pair passes the layer of S-boxes with one of the 2^{14} possible differences, on the parallel side of the boomerang the pair matches this difference with probability 2^{-14}). So the two pairs of ciphertexts pass the second characteristic with a probability of $2^{-(3\cdot 6+3\cdot 6+2\cdot 7)} = 2^{-50}$. Now that we have passed with low cost the 4th round layer of S-boxes where the top-down characteristic had 16 active S-boxes we switch to the last phase of the boomerang attack, where the effect of the mixing of the third round can be undone for free because are guaranteed to have the same difference as in the forward direction. In the second round we pay again 2^{-24} for four to one active byte of MixColumns(2^{-22} if we do not require the boomerang to return in exactly the same 4 bytes). The next round is done for free so we obtain two plaintexts with a difference only in four diagonal bytes.

Hence the total probability of the boomerang is $2^{-22-50-24} = 2^{-96}$. Each structure of 2^{32} plaintexts contains 2^{63} pairs with a difference in the four diagonal bytes. Hence, to find two good boomerang quartets we need $2^{96-63+1} = 2^{34}$ structures or $2^{34+32} = 2^{66}$ chosen plaintexts and $2^{31} \cdot 2^{66} = 2^{97}$ adaptive chosen ciphertexts. The average amount of false quartets for all 2^{31} key guesses which satisfy our 96+8 = 104-bit filtering condition is $2^{31+34+63-104} = 2^{24}$. Note that each boomerang quartet suggest 31-bit value for the key guess at the bottom as well as 16 guesses for 64 bits at the top (corresponding to 4 active S-boxes in the plaintext at each side). Since we requested two good boomerang quartets they will vote together for the correct keys while the remaining 2^{24} false quartets would vote randomly. We expect that none of the false quartets survive this 91-bit key voting step.

At this point the attacker can either finish the attack with an exhaustive search of about 2^{96} steps or by repeating the boomerang attack starting from another 4 active S-boxes in the plaintext.

3.3 Related-Key Boomerang Attacks on AES-192 and AES-256

We tweaked our tool to produce the optimal differential characteristics for a boomerang attack on AES-192. The tool produced a top differential characteristic other than the one presented in [9], with the same bottom characteristic. The two characteristics are shown at Fig.3 in the Appendix. The ladder switch between the two characteristics in round 6 is simpler: due to the switch there are no active S-boxes in this round. The top characteristic has 2 active in round 3, and 1 in round 4, while the bottom characteristic has 1 active in round 7, 8, and 10, and 2 active in round 9. Hence, the probability of the boomerang is $2^{-6\cdot(2+1+1+1+2+1)} = 2^{-48}$ compared to the boomerang in [9] with a probability 2^{-55}. A rough estimate between these two attacks gives us a speed-up of 2^7: the new boomerang attack requires 2^{116} data, and 2^{169} time.

For AES-256, on 6 and 7 rounds there are only two characteristics with 5 active S-boxes (and no characteristics with less active S-boxes), and these are the exact characteristics used in the boomerang attack on AES-256 in [9]. On the other hand, 8-round characteristic has at least 10 active S-boxes, hence using it in a boomerang attack will blow up the complexity above the best known attack. Therefore, we believe that the characteristics used for the attack on AES-256 in [9] are optimal.

4 Camellia

Camellia [1] is a 128-bit SPN block cipher with 128, 192, and 256-bit keys. We will analyze Camellia with 128-bit keys, without the FL functions. This version has 18 rounds, and so far, the best cryptanalytical results are truncated differential of 8 rounds [25], and impossible differential on 11 rounds [27]. The key schedule of Camellia is not byte oriented because the rotation constants are not a multiple of 8. In order to test our tool we will make it byte oriented, by using the following rotation constants for rounds 1-18: 0, 0, 16, 16, 16, 16, 48, 48, 48, 64, 64, 64, 96, 96, 96, 96, 112, 112. We call this version — byte-Camellia. Note that, since we choose the rotation constants as close as possible to the original constants, a differential characteristic in the key schedule of byte-Camellia, may be suitable for the original Camellia. For that to happen, the positions of the active *bits* in an active byte have to be invariant of small rotations. On the other hand, trying all possible combinations of active bits, i.e. building all possible differential characteristics in the key schedule for the original version of Camellia, seems too much time consuming. Hence, we will analyze only byte-Camellia.

The key schedule of Camellia-128 applies transforms (4 rounds) to the master key K_L, to produce another key K_A and it uses these two values to generate the subkeys in a linear way. Therefore, *we will use variant 3 of the tool.* Internally in the tool, in all steps we will use only the S-type representation.

4.1 Best Round-Reduced Differential Characteristics for Byte-Camellia

The maximal difference propagation probability of the S-boxes in Camellia is 2^{-6}. Therefore, we can allow no more than $\lfloor \frac{128}{6} \rfloor = 21$ active S-box in the characteristic of the key and the state. With this type of limitations, the tool produced the best related-key differential characteristic. It is on 8 rounds, and it has 20 active S-boxes.

4.2 Chosen-Key Attack on Full-Round Byte-Camellia

When searching for chosen related-key characteristics in byte-Camellia, we can spend 21 active S-box in each, the key and the state (using the divide-and-conquer technique). With these weight limitations our tool was able to produce a good characteristic on all 18 rounds of byte-Camellia. The characteristic has 17 active S-boxes in the key, and 15 in the state (see Fig. 4 in the Appendix). The characteristic can be used to show that 256-bit double-block-length [19] hash function construction initiated with byte-Camellia-128 cipher, can be distinguished from a random function.

5 Khazad

Khazad [3] is a 64-bit block cipher with a key size of 128 bits. It is an SPN with 8 rounds. The best attacks go only up to 5 rounds: an integral attack [31] with 2^{91} complexity and a class of 2^{64} weak keys which can be attacked in 2^{40} steps using a slide attack [7].

The subkeys in the key schedule of Khazad are obtained consecutively from one another using a Feistel function. Therefore, *we will use variant 2 of the tool.* The small key and block sizes, in addition to the low branching in the key schedule allows to use the variant 1 as well. The optimal representation is similar to the one used in the tool for AES. In the state, after the S-boxes (γ) we will have non-zero only S-value, and after the linear-diffusion layer (θ) only P-value.

5.1 Best Round-Reduced Differential Characteristics for Khazad

The maximal difference propagation of the S-boxes in Khazad is 2^{-5}. Hence, a differential characteristic for Khazad cannot have more than 25 active S-boxes, at most 12 can be in the state. With this type of limitations, the tool was able to produce interesting results. The best related-key differential characteristics for 4,5,6, and 7 rounds have 9, 10, 19, and 20 active S-boxes, respectively. The related-key attacks based on such characteristics would be the new best attacks on Khazad up to 7 rounds. The 7-round characteristic is presented at Fig.5 in the Appendix.

5.2 Related-Key Boomerang Attacks on 7 Rounds of Khazad

Let us improve the probability of the 7-round attack by using a boomerang attack. We will use two 4 round characteristics (See Fig.5 in the Appendix). The four related keys K^A, K^B, K^C, and K^D, are obtained as follows: 1) fix any K^A, i.e. (K^A_{-2}, K^A_{-1}), 2) produce (K^A_0, K^A_1) from K^A, and fix K^B such that $K^B = (K^A_0, K^A_1) \oplus (\Delta K_0, \Delta K_1)$, 2) obtain (K^A_6, K^A_7) and (K^B_6, K^B_7) and then fix $K^C = (K^A_6, K^A_7) \oplus (\Delta K_6, \Delta K_7)$, $K^D = (K^B_6, K^B_7) \oplus (\Delta K_6, \Delta K_7)$. The pink difference was chosen such that after γ and θ it could produce gray difference with a probability 2^{-5}. Let us find the complexity of the attack. We start with the same one byte difference (the pink byte) in the plaintext and the subkey K_0, hence there are no active bytes in the state in round 1 and 2. The difference in the subkey K_3, as well as in the state, (denoted with the grey bytes) is obtained when the pink byte goes through γ and θ, and hence it happens with 2^{-5}. At the end of round 4 we can switch to the bottom characteristic. The ciphertext difference is fully determined. We pay 2^{-5} in round 7 so the blue byte in the state after the inverse S-box will become pink (and then cancel with the pink difference in the key). To get a zero difference in the subkey K_5 we pay additional 2^{-5}. In round 4 we switch the state to the top characteristic. An important moment is the switch in the keys. When the gray difference in the top characteristic between K^A and K^B in the subkey K_3 is the same as the gray difference in the bottom characteristic between K^A and K^C (and K^B and K^D) in the subkey K_3, then the switch in the key is for free (this is due to the Feistel switch[3], See [9]). Then not only the difference in K_3 between K^C and K^D will be the same as between K^A and K^B, but their value will be equal to the values of K^A_3 and K^B_3 and hence will go through the S-boxes producing the same values. Therefore, we pay additional 2^{-5} for each of the differences in subkey K_3 (instead of a switching cost of 2^{-64}!). After the switch to the top characteristic, we pay 2^{-5} in the state of round 3 to get the same pink difference which will cancel after the key XOR. We pay additional 2^{-5} for the zero difference in the subkey K_1. The rest of the characteristic holds with probability 1. The probability of the whole boomerang attack is $2^{-(2\cdot5+2\cdot5+2\cdot5+2\cdot5+5+5)} = 2^{-50}$. This translates into a boomerang attack in a class of weak keys that works for 1 out of 2^{30} related-key

[3] We have tested the Feistel switch in the key schedule of Khazad, and a related-key quartet, following the whole 7-round differential characteristic, was found.

quartets, with a complexity 2^{20} encryptions/decryptions. Moreover if we relax constraints on the difference in the key we can increase the size of the weak key class to 1 out of every 2^8 keys which can be attacked with complexity of 2^{49} encryptions and analysis steps. In both cases when the boomerang returns we know the plaintext difference and thus we have a 64 bit filter which allows us to filter out all the wrong quartets. Returning boomerang provides us with 7 bits of information about the key byte K_2^0 since we know the input and output difference for the active S-box of the key schedule and similarly about 7 bits of the key of K_6^0. One can extend this attack into a full key recovery attack via auxiliary techniques.

5.3 Chosen-Key Attack on Full-Round Khazad

The 7-round related-key differential characteristic can easily be extended at the top for an additional round and then used in a chosen-key attack (See Fig. 5 in the Appendix). Since we control the exact values of the key and the state, we will use the divide-and-conquer technique, and first fix the keys satisfying the characteristic in the key, and then find a proper pair of plaintexts that follow the characteristic in the state. We can use the rebound attack [30] and fix one round for free in both the key and in the state. For the key, we can fix the round for $\varDelta K_4$ (or $\varDelta K_6$), and obtain a characteristic in the key that holds with probability 2^{-55}. In the state, we will fix the values in the first round, hence the characteristic in the state holds with 2^{-10}. The S-boxes are non-injective regarding the difference, i.e. if we fix the input and output difference of the S-box, then there is a solution with probability $\frac{1}{2}$. Therefore, we introduce a possible one bit difference in each byte of the plaintext so that there will always be a solution for the S-box input/output differences. The total complexity of the chosen-key distinguisher is bounded by the probability of the characteristic in the key and is 2^{55}. The input difference is fixed in 56 bits, while the output is fixed in all 64-bits. This shows that full Khazad has properties which are not present in an ideal cipher. This also means, for example, that 256-bit Tandem-DM[24] hash function construction initiated with Khazad cipher, can be distinguished from a random function.

6 FOX and Anubis

The ciphers FOX [20] and Anubis[2] have highly non-linear key schedule, leading to a potentially low key agility. This property becomes important when the key of the cipher is frequently changed, for example when the cipher is used in some hash function construction. Yet, with the respect to related-key differentials, the key schedule of these ciphers is exceptionally resistant. The large number of S-boxes in the schedule ensures that a related-key differential attack cannot be launched on more than some very modest number of rounds.

We have analyzed FOX64 – the 64-bit block version of FOX with 128-bit key and 16 rounds. Each round key of this cipher is produced from the master key with a sequence of transformations $NL64$. We have found empirically, by checking all the possible key differential characteristics, that the minimal number of active S-boxes in $NL64$ is 7. This means that in any related-key differential characteristic, for each round of FOX64 one has to spend at least 7 S-boxes *only* for producing the round key. The maximal difference propagation probability of the S-box in FOX is 2^{-4}, while the key is 128 bits. Hence, we can conclude that for FOX64 there is no related-key differential characteristic on more than $\lfloor \frac{128}{4 \cdot 7} \rfloor = 4$ rounds.

Anubis, a 128-bit block cipher, supports variety of key sizes from 128 bits to 320 while it has $8 + \frac{keysize}{32}$ rounds. We will focus on key sizes up to 256 bits. The maximal difference probability of the S-boxes is 2^{-5}, hence the maximal number of active S-boxes in a characteristic can not be greater than $\lfloor \frac{256}{5} \rfloor = 51$. The key schedule of Anubis is SPN with an additional S-box layer at the end of each round (as well as ω and τ transformations). This means that in a characteristic of the key schedule, each active S-box should be counted twice, except for the S-boxes in the first round (which are counted only once). The branch number of the linear-diffusion layer is 5, hence in four consecutive rounds there are at least $5^2 = 25$ active S-boxes. Therefore, in 5 rounds of the key schedule, there are at least $2 \cdot 25$ active S-boxes in the last 4 rounds, and at least 1 in the first round, or in total at least 51 active S-boxes in the 5-round characteristic of the key schedule. Hence, in Anubis there are no related-key differential characteristics on more than 5 rounds.

7 Conclusions and Future Research

We presented a tool for search of related-key differential characteristics in various ciphers. It produced the best round-reduced differential characteristics, which helped to improve the best known attacks on AES-128, AES-192, byte-Camellia-128, and Khazad. It also allowed to prove security bounds against simple related-key attacks for AES-128, FOX and Anubis. The tool runs in a range of few hours (Khazad, byte-Camellia) to several days (AES-128) and weeks (AES-192) and takes from several megabytes to 25 Gbytes (byte-Camellia) of memory for the transition lookup tables in the key schedule.

The tool was implemented as described in the paper and while it produced a lot of interesting results, a couple of open problems emerged. The first one is how to deal with ciphers that have small part that is not byte oriented such as Camellia which has rotations in the key schedule. Second, it is very interesting to adapt the tool to hash functions. This would mean that the problem of very large internal state of some modern hash functions has to be solved. The idea of applying a similar tool for finding related-key differential characteristics in DES and in non-byte oriented ciphers also seems attractive. Producing chosen-key differential characteristics for 9 and 10 rounds of AES-128 is still an open problem.

References

1. Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J., Tokita, T.: Camellia: A 128-bit block cipher suitable for multiple platforms - design and analysis. In: Stinson, D.R., Tavares, S.E. (eds.) SAC 2000. LNCS, vol. 2012, pp. 39–56. Springer, Heidelberg (2000)
2. Barreto, P., Rijmen, V.: The Anubis Block Cipher. Submission to the NESSIE Project (2000)
3. Barreto, P., Rijmen, V.: The Khazad Legacy-Level Block Cipher. Submission to the NESSIE Project (2000)
4. Biham, E.: New types of cryptanalytic attacks using related keys. J. Cryptology 7(4), 229–246 (1994)
5. Biham, E., Dunkelman, O., Keller, N.: Related-key boomerang and rectangle attacks. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 507–525. Springer, Heidelberg (2005)
6. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. J. Cryptology 4(1), 3–72 (1991)
7. Biryukov, A.: Analysis of involutional ciphers: Khazad and Anubis. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 45–53. Springer, Heidelberg (2003)
8. Biryukov, A., Dunkelman, O., Keller, N., Khovratovich, D., Shamir, A.: Key recovery attacks of practical complexity on AES variants with up to 10 rounds. In: EUROCRYPT 2010 (to appear, 2010)
9. Biryukov, A., Khovratovich, D.: Related-key cryptanalysis of the full AES-192 and AES-256. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 1–18. Springer, Heidelberg (2009)
10. Biryukov, A., Khovratovich, D., Nikolić, I.: Distinguisher and related-key attack on the full AES-256. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 231–249. Springer, Heidelberg (2009)
11. Cannière, C.D., Rechberger, C.: Finding SHA-1 characteristics: General results and applications. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 1–20. Springer, Heidelberg (2006)
12. Chowdhury, D.R., Rijmen, V., Das, A. (eds.): INDOCRYPT 2008. LNCS, vol. 5365. Springer, Heidelberg (2008)
13. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer, Heidelberg (2002)
14. Ferguson, N., Kelsey, J., Lucks, S., Schneier, B., Stay, M., Wagner, D., Whiting, D.: Improved cryptanalysis of Rijndael. In: Schneier, B. (ed.) FSE 2000. LNCS, vol. 1978, pp. 213–230. Springer, Heidelberg (2000)
15. Fouque, P.-A., Leurent, G., Nguyen, P.: Automatic search of differential path in MD4. Cryptology ePrint Archive, Report 2007/206 (2007)
16. Gilbert, H., Minier, M.: A collision attack on 7 rounds of Rijndael. In: AES Candidate Conference, pp. 230–241 (2000)
17. Gilbert, H., Peyrin, T.: Super-Sbox Cryptanalysis: Improved Attacks for AES-like permutations. In: FSE 2010 (to appear, 2010)
18. Gorski, M., Lucks, S.: New related-key boomerang attacks on AES. In: Chowdhury, et al. (eds.) [12], pp. 266–278
19. Hirose, S.: Some plausible constructions of double-block-length hash functions. In: Robshaw [33], pp. 210–225
20. Junod, P., Vaudenay, S.: FOX: A new family of block ciphers. In: Handschuh, H., Hasan, M.A. (eds.) SAC 2004. LNCS, vol. 3357, pp. 114–129. Springer, Heidelberg (2004)

21. Khovratovich, D., Biryukov, A., Nikolić, I.: Speeding up collision search for byte-oriented hash functions. In: Fischlin, M. (ed.) CT-RSA 2009. LNCS, vol. 5473, pp. 164–181. Springer, Heidelberg (2009)
22. Kim, J., Hong, S., Preneel, B.: Related-key rectangle attacks on reduced AES-192 and AES-256. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 225–241. Springer, Heidelberg (2007)
23. Knudsen, L.R., Rijmen, V.: Known-key distinguishers for some block ciphers. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 315–324. Springer, Heidelberg (2007)
24. Lai, X., Massey, J.L.: Hash function based on block ciphers. In: Rueppel, R.A. (ed.) EUROCRYPT 1992. LNCS, vol. 658, pp. 55–70. Springer, Heidelberg (1993)
25. Lee, S., Hong, S., Lee, S., Lim, J., Yoon, S.: Truncated differential cryptanalysis of Camellia. In: Kim, K.-c. (ed.) ICISC 2001. LNCS, vol. 2288, pp. 32–38. Springer, Heidelberg (2002)
26. Lu, J., Dunkelman, O., Keller, N., Kim, J.: New impossible differential attacks on AES. In: Chowdhury, et al (eds.) [12], pp. 279–293
27. Lu, J., Kim, J., Keller, N., Dunkelman, O.: Improving the efficiency of impossible differential cryptanalysis of reduced Camellia and MISTY1. In: Malkin, T. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 370–386. Springer, Heidelberg (2008)
28. Matsui, M.: Linear cryptoanalysis method for DES cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1994)
29. Matsui, M.: On correlation between the order of S-boxes and the strength of DES. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 366–375. Springer, Heidelberg (1994)
30. Mendel, F., Rechberger, C., Schläffer, M., Thomsen, S.S.: The rebound attack: Cryptanalysis of reduced Whirlpool and Grøstl. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 260–276. Springer, Heidelberg (2009)
31. Muller, F.: A new attack against Khazad. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 347–358. Springer, Heidelberg (2003)
32. National Institute of Standards and Technology. Advanced encryption standard (AES). FIPS 197 (November 2001)
33. Robshaw, M.J.B. (ed.): FSE 2006. LNCS, vol. 4047. Springer, Heidelberg (2006)
34. Schläffer, M., Oswald, E.: Searching for differential paths in MD4. In: Robshaw (ed.) [33], pp. 242–261
35. Stevens, M.: Fast collision attack on MD5. Cryptology ePrint Archive, Report 2006/104 (2006)

Appendix

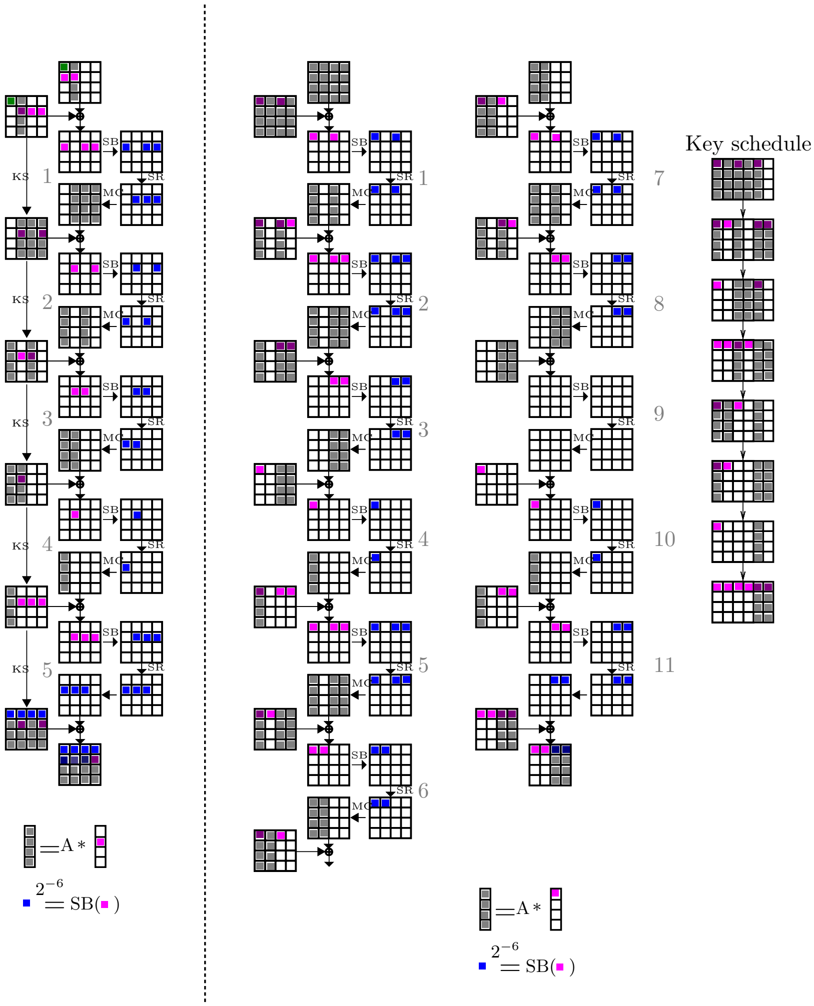


Fig. 2. The best characteristics for AES-128(left) and AES-192(right). The first one is on 5 rounds, while the second one is on 11 rounds.

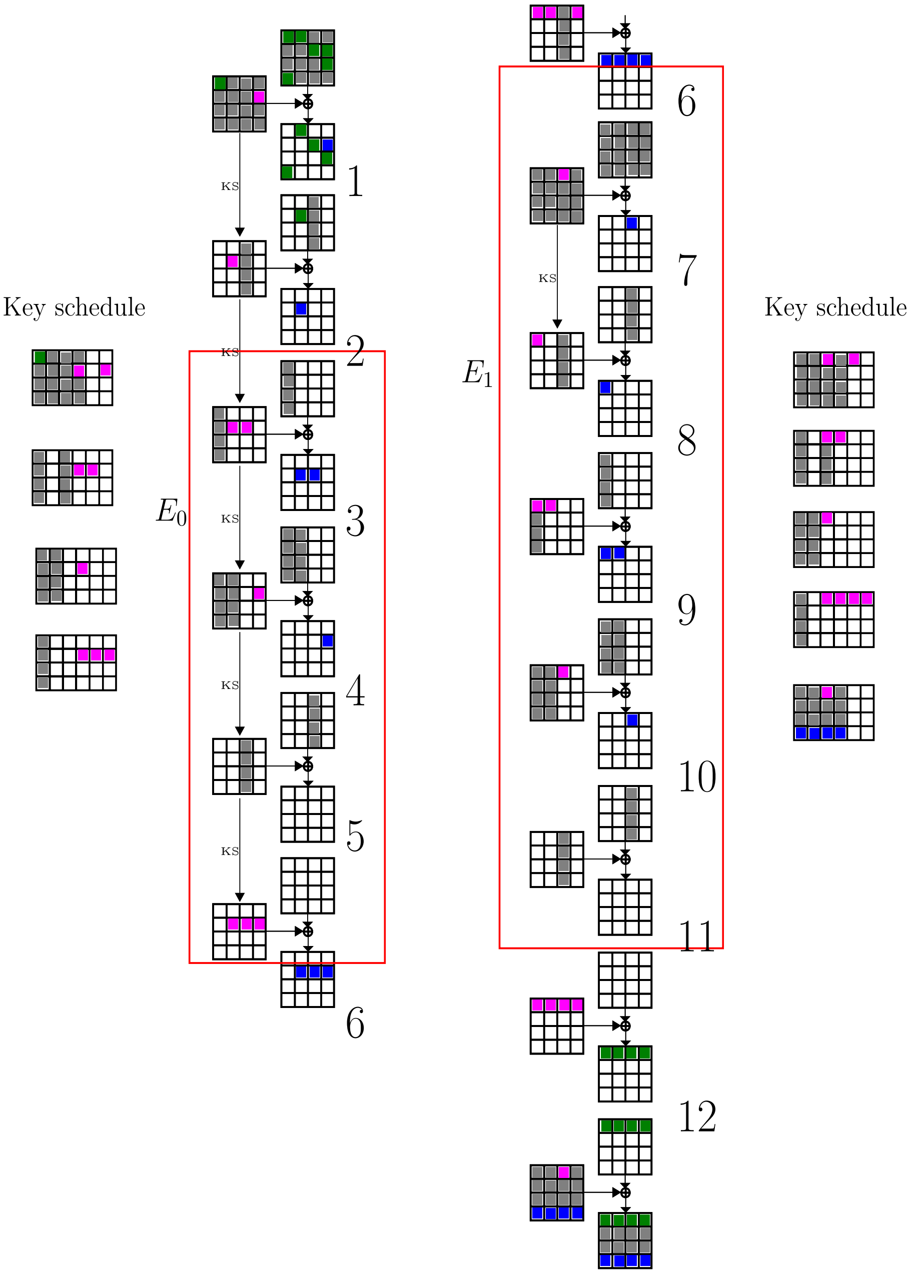


Fig. 3. A related-key boomerang attack on AES-192. The bottom characteristic is the same as in [9].

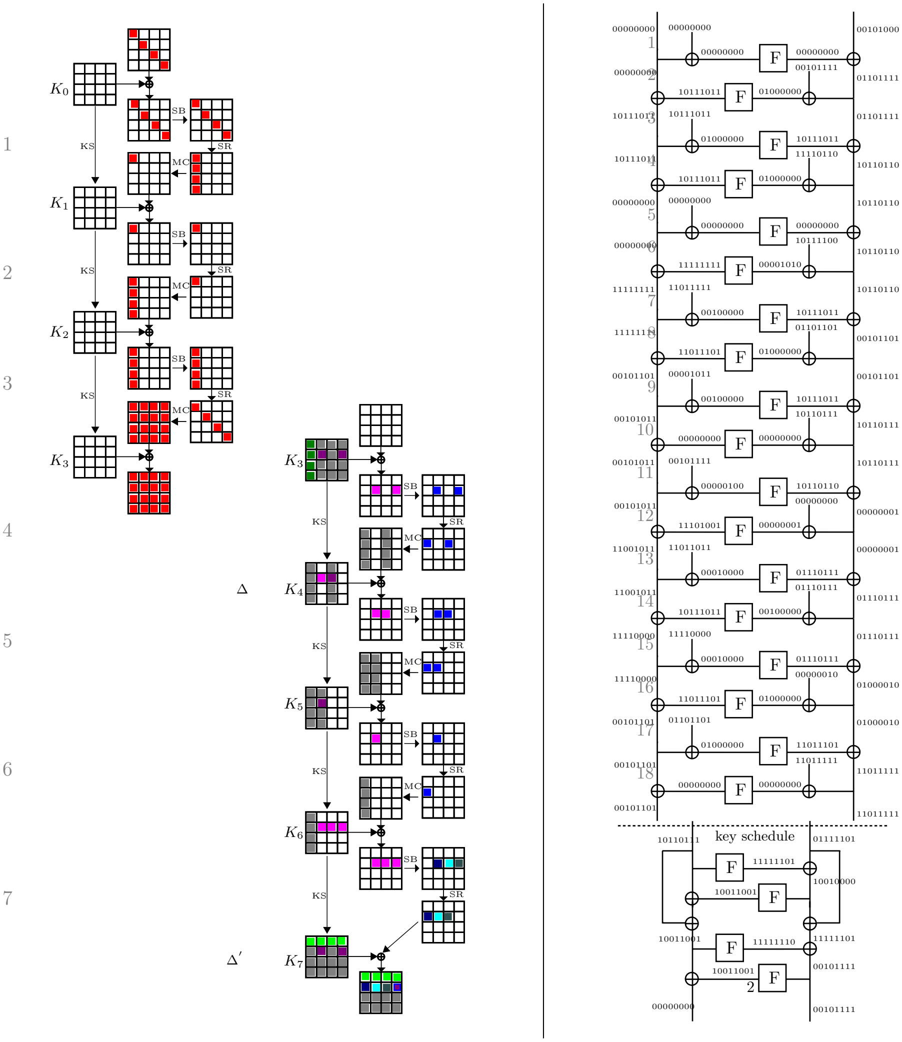


Fig. 4. On the left, two characteristics for the related-key boomerang attack on AES-128 reduced to seven rounds. On the right, the related-key differential characteristic (top in the state, bottom in the key) on full-round byte-Camellia-128 for the chosen-key distinguisher. The characteristic has compact representation, the actual differences are to be fixed. The key XOR is depicted separately from the function F.

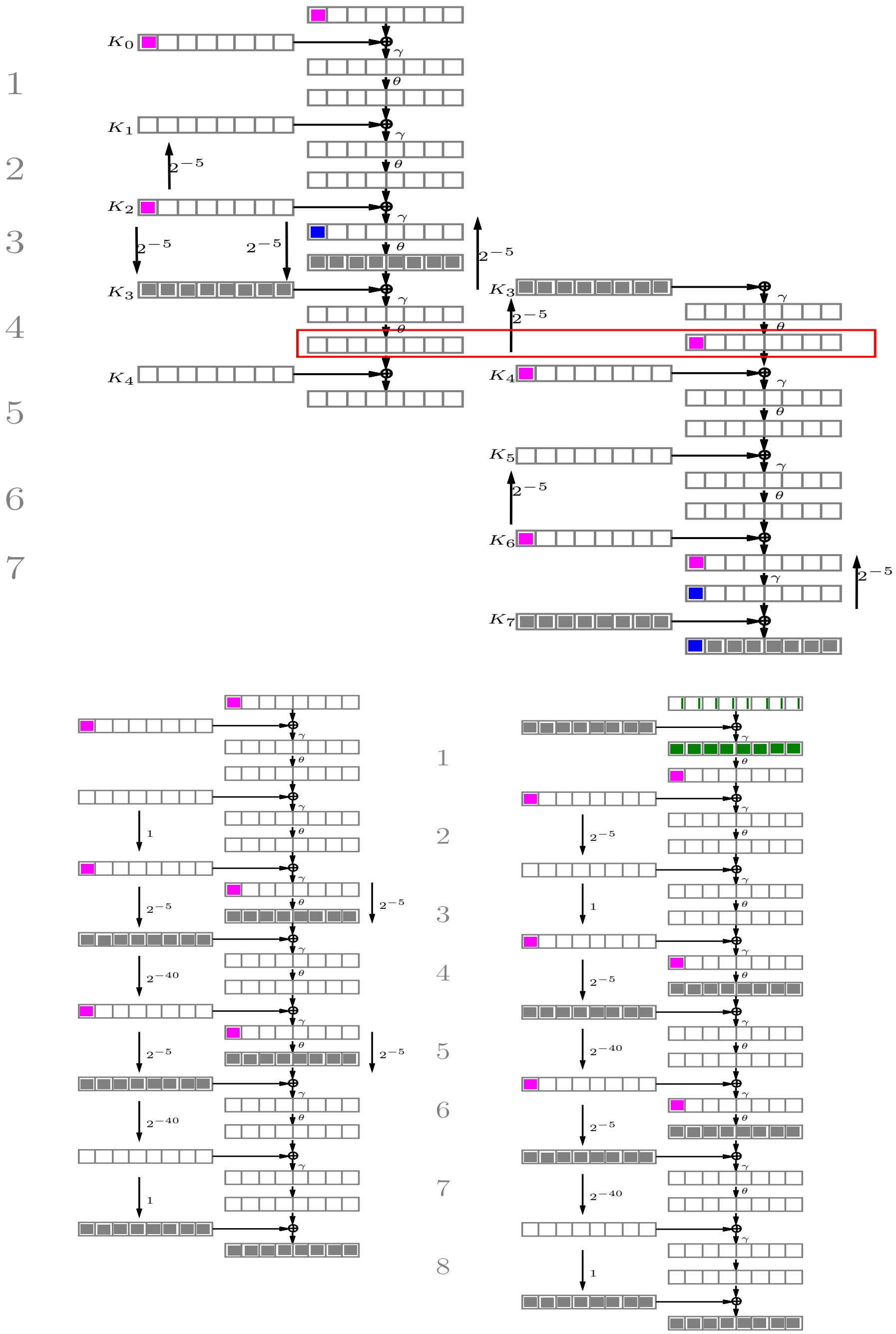


Fig. 5. Characteristics for the related-key boomerang attack on 7-round Khazad (top), related-key differential characteristic on 7 rounds (bottom-left), and related-key differential characteristic on 8 rounds used for a chosen-key distinguisher (bottom-right)

Plaintext-Dependent Decryption: A Formal Security Treatment of SSH-CTR[⋆]

Kenneth G. Paterson[⋆⋆] and Gaven J. Watson[⋆⋆⋆]

Information Security Group,
Royal Holloway, University of London,
Egham, Surrey, TW20 0EX, U.K.
{kenny.paterson,g.watson}@rhul.ac.uk

Abstract. This paper presents a formal security analysis of SSH in counter mode in a security model that accurately captures the capabilities of real-world attackers, as well as security-relevant features of the SSH specifications and the OpenSSH implementation of SSH. Under reasonable assumptions on the block cipher and MAC algorithms used to construct the SSH Binary Packet Protocol (BPP), we are able to show that the SSH BPP meets a strong and appropriate notion of security: indistinguishability under buffered, stateful chosen-ciphertext attacks. This result helps to bridge the gap between the existing security analysis of the SSH BPP by Bellare *et al.* and the recently discovered attacks against the SSH BPP by Albrecht *et al.* which partially invalidate that analysis.

Keywords: SSH; counter mode; security proof.

1 Introduction

SSH is one of the most widely used secure network protocols. Originally designed as a replacement for insecure remote login procedures which sent information in plaintext, it has since become a general purpose tool for securing Internet traffic. The current version of SSH, SSHv2, was designed in 1996, and it is this version to which we refer throughout this paper. The SSHv2 protocols are defined in a collection of RFCs [4,11,12,13,14].

The SSH Binary Packet Protocol (BPP), as specified in [13], is the component of SSH that is responsible for providing confidentiality and integrity services to all messages exchanged over an SSH connection. It was subjected to a formal cryptographic security analysis using the methods of provable security by Bellare *et al.* [3]. Bellare *et al.* introduced a stateful security model and notion for SSH-style protocols. They also proved that several minor variants of the SSH

[⋆] This research was supported in part by the European Commission under contract ICT-2007-216676 (ECRYPT-II).

[⋆⋆] This author supported by an EPSRC Leadership Fellowship, EP/H005455/1.

[⋆⋆⋆] This author supported by an EPSRC Industrial CASE studentship sponsored by BT Research Laboratories.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 345–361, 2010.

BPP meet their security notion, given reasonable assumptions about the cryptographic primitives. In particular, they showed that, while the SSH BPP using CBC mode encryption with IV chaining (SSH-IPC) is *in*secure, the SSH BPP using either CBC mode encryption with explicit random IVs and random padding (SSH-$NPC), or counter mode encryption (SSH-CTR), is secure in their model.

However, the recent work of Albrecht *et al.* [1] has demonstrated plaintext recovery attacks against both SSH-IPC and SSH-$NPC, despite the proof of security for SSH-$NPC in [3]. The attacks in [1] exploit several features that are intrinsic to the SSH specification and to implementations, but that are not captured in the security model of [3]: firstly, the decryption process depends on the packet length field, which itself forms part of the plaintext data; secondly, data can be delivered to the decrypting party in a byte-by-byte manner by an attacker, allowing the attacker to observe the behaviour of the decrypting party after each byte is received; and, thirdly, the attacker can distinguish various kinds of decryption failure (most importantly, the attacker can tell exactly when a MAC fails to verify). As a consequence of these attacks, versions 5.2 and higher of OpenSSH, the leading implementation of SSH, now negotiate the selection of counter mode in preference to CBC mode. This follows the recommendation of the CPNI vulnerability announcement [7]. OpenSSH versions 5.2 and higher also include specific counter-measures for CBC mode to frustrate the CBC-specific attacks of [1].

No attacks are known against the SSH BPP using counter mode, and the security model and proof for the relevant scheme SSH-CTR provided in [3] does rule out many classes of attack. Yet it is evident, in view of the attacks in [1], that the current formal security analysis of SSH-CTR in [3] is inadequate. In particular, the current analysis of SSH-CTR does not take into account the plaintext-dependent nature of the decryption process, nor the ability of the attacker to interact in a byte-by-byte manner with the decryption process. Indeed, the length field which turns out to be so critical to breaking SSH in [1] is ignored in the security analysis of [3], while it is assumed in [3] that ciphertexts are processed in an atomic fashion. Moreover, while the model of [3] does include errors arising from cryptographic processing, it does not do so in a way that accurately reflects the reality of SSH implementations such as OpenSSH – in the model of [3], any error condition leads to an identical error message, while in reality, the error type and the timing of the error can both leak to the adversary. This additional information was also exploited in the attacks of [1].

1.1 Our Contribution

This paper aims to bridge the gap between the current security analysis of the SSH-CTR in [3] on the one hand, and the reality of the SSH specifications in the RFCs and the OpenSSH implementation of the SSH BPP using counter mode on the other. We develop a security model for the SSH BPP that extends the stateful model introduced in [3] and that is driven by our desire to more closely align the security model with the SSH specifications and the OpenSSH implementation. We focus on the OpenSSH implementation in preference to any of the many other SSH implementations available because of its widespread use [10]. A novel aspect of our security model is its ability to allow the attacker to interact with the decryption oracle in a byte-by-byte fashion, with ciphertext bytes being buffered until they can be processed. Novel aspects of our description of the SSH BPP using counter mode include its provision for plaintext-dependent decryption, and accurate modeling of all the error events that arise during decryption in the OpenSSH implementation of the SSH BPP in counter mode. We prove that the SSH BPP using counter mode is secure in our model, under standard assumptions concerning the cryptographic components used in the construction. This requires significant reworking of the security analysis for counter mode in [3] to take account of the new features of our model and our description of the SSH BPP. Our analysis is sufficient to show that the SSH BPP using counter mode is immune to the type of attacks reported in [1].

While our analysis is quite specific to the SSH BPP in counter mode, we believe that the modeling and proof techniques developed here should be much more widely applicable: all reasonably complex secure communication protocols involve handling of error and other management messages, and many such protocols allow for the adversary to interact with the decryption process in a fine-grained manner (rather than in a "ciphertext-atomic" manner). More generally, we hope that our practice-driven, provable security analysis of the SSH BPP will serve as an example to show that provable security techniques have an important role to play in analyzing protocols that are used in the real world, whilst taking into account low-level, code-oriented behaviours of the cryptographic elements of the protocols.

1.2 Paper Organisation

We begin by giving a description of the SSH Binary Packet Protocol in Section 2, using this to identify the key features required in our modeling of the SSH BPP and its security. In Section 3 we define the building blocks that we use to define the SSH BPP's Encode-then-Encrypt&MAC encryption scheme. Section 4 gives the definitions of our new security models. Section 5 contains our proof of security for SSH using counter mode encryption. Section 6 presents our conclusions.

2 SSH Binary Packet Protocol

The SSH Binary Packet Protocol (BPP) is defined in RFC 4253 [13]. The SSH BPP provides both confidentiality and integrity of messages sent over an SSH connection using an encode-then-encrypt&MAC construction. A message is first encoded by prepending a 4 byte packet length field and 1 byte padding length field and appending a minimum of 4 bytes of random padding. The packet length field specifies the total length of the encoded message excluding the packet length field itself. This encoded message is then encrypted. There are various algorithms supported for encryption, but here, in the light of the attacks in [1], we only consider stateful counter mode encryption, as specified for SSH in RFC 4344 [4]. Since the

SSH BPP is specified in a blockwise manner, SSH still appends padding even when using counter mode encryption. The final ciphertext is the concatenation of the encoded-then-encrypted message and a MAC value. The MAC value is computed over the concatenation of a 32-bit packet sequence number and the encoded (but not encrypted) message. The sequence number is not sent over the channel but is maintained separately by both communicating parties.

2.1 Modeling the SSH BPP and Its Security

We now give a high-level description of the main features of our model for the SSH BPP and its security, explaining how these arise from features of the SSH BPP specification and specific implementations.

As with the model of [3], our model for the SSH BPP is a stateful one, reflecting the protocol's use of per-packet sequence numbers. We also wish to give the adversary access to encryption and decryption oracles in a left-or-right indistinguishability game. We next discuss how these oracles should be defined, with further details to follow in the sections ahead. At this point, our model begins to significantly diverge from the model of [3].

When decrypting a ciphertext, the receiver should first decrypt the first block received and retrieve the packet length field in order to determine how much more data must be received before the MAC tag is obtained. According to RFC 4253 [13]:

> *"Implementations SHOULD decrypt the length after receiving the first 8 (or cipher block size, whichever is larger) bytes of a packet."*

Thus we may expect that an SSH implementation will enter into a wait state, awaiting further data, unless sufficient data has already arrived to complete the packet. Informally speaking, this renders the entire decryption process plaintext-dependent, in the sense that the number of ciphertext bytes required before the decryption process can complete (possibly with an error message because of a MAC verification failure) is determined by the initial bytes of the plaintext. Moreover, because SSH is implemented over TCP, the attacker can deliver as few or as many bytes of ciphertext at a time as he wishes to the decrypting party. These facts are exploited in the attacks against the SSH BPP in CBC mode in [1]. Thus our security analysis for the SSH BPP needs to consider the length field and how its processing affects security, as well as allowing the adversary to deliver data to the decryption oracle in a byte-by-byte manner in the security model. However, it should be noted that the plaintext message is not made available to the adversary in a byte-by-byte manner as it is decrypted. Instead, in implementations, the plaintext is buffered until sufficient data has arrived that the MAC can be checked. Our model, therefore, needs to allow byte-by-byte delivery of ciphertext data, but also to include a buffered decryption process.

In fact, the situation is more complicated than this because implementations of SSH also follow the advice in RFC 4253 [13] to perform sanity checking of the length field as soon as it is obtained from the first block of ciphertext:

> *"...implementations SHOULD check that the packet length is reasonable in order for the implementation to avoid denial of service and/or buffer overflow attacks."*

What is "reasonable" is not defined in the RFCs, and specific implementations adopt various practices. Version 5.2 of OpenSSH implements a particular set of checks, and tries to tear down the SSH connection with an error message in the event that these checks fail. This error condition is generally quite easy to distinguish from a MAC failure in an attack because an SSH implementation can be made to pass through a wait state before the MAC failure. The distinguishability of these different error conditions is used in the attacks against OpenSSH in CBC mode in [1]. So a security model for the SSH BPP should include errors arising from length checking as well as from MAC failures, and should report these errors in such a way that they can be distinguished by the adversary. Additional errors may arise after MAC checking, because of a failure of the decoding algorithm applied to the recovered, encoded message. Again, the model should reflect this possibility. To comply with the SSH specifications, all of these errors should be "fatal", leading to the destruction of the SSH connection. However, note that an adversary may be able to prevent such error messages from reaching the peer of party initiating the tear-down. We handle this aspect by having separate states for the encryption and decryption oracles in our model, and with an error arising during decryption leading to the loss of the decryption oracle, but not the encryption oracle, and vice-versa.

It is notable that SSH attempts to hide the packet length field by encrypting it. However, a simple extension of the attacks in [1] shows that this is futile: an attacker who can detect the start of a new packet simply needs to flip a bit somewhere in the ciphertext after the length field and wait for a MAC failure. Simple arithmetic involving the number of ciphertext bytes delivered before the MAC failure is seen then tells the attacker what the content of the packet length field was. Of course, the cost of this attack is to lose the SSH connection. However, it shows that the length field cannot be hidden from an active attacker. For this reason, we will insist that, in our left-or-right indistinguishability game, all pairs of messages submitted to the encryption oracle should have the same length when encoded, so that they cannot be trivially distinguished using the above attack.

3 Definitions

3.1 Notation

First let us begin by defining some notation. For a string x, let $|x|$ denote the length of x in bytes, and let $x[i]$ denote the i-th block of x, where, throughout, blocks consist of L bytes. Let $x[1 \ldots n]$ denote the concatenation of the blocks $x[1], x[2], \ldots, x[n]$ of x and let $x \| y$ denote the concatenation of strings x and y. Let ε denote the empty string. Let $\langle i \rangle_t$ denote the t-byte binary representation of integer i, where $0 \leq i < 2^{8t}$.

3.2 Building Blocks

Based on the discussion in the previous section, we now define the primitives which form the building blocks in our description of the SSH BBP's encode-then-encrypt&MAC construction. These building blocks are an encoding scheme $\mathcal{EC}$, an encryption scheme (we consider only counter mode encryption) and a message authentication scheme $\mathcal{MA}$.

Encoding Scheme: The *encoding scheme* $\mathcal{EC} = (\mathsf{enc}, \mathsf{dec})$ used in SSH consists of an encoding algorithm enc and a decoding algorithm dec. The encoding algorithm enc is stateful and randomised, takes as input a message m and outputs two messages (m_e, m_t). Here as in [3], m_e denotes the encoded message which will be used by any future encryption process and m_t denotes the encoded message which will be used by a MAC tagging algorithm. As required by the SSH BPP, the encoding algorithm prepends some length information about the message and appends some padding.

The decoding algorithm dec is stateful and deterministic. It takes as input the full encoded message $m_e = m_e[1 \ldots n]$, strips off all length fields and outputs the decoded message m. However, if it is unable to parse the message correctly an error message $\perp_P$ is output. Note that our definition of dec is slightly different to that in [3] which had two outputs m and m_t. Note also that dec will only be called during the decryption process for SSH if both length checking and MAC checking have not returned errors. For correctness of the encoding scheme, we require that for any m with $\mathsf{enc}(m) = (m_e, m_t) \neq (\perp, \perp)$, we have $\mathsf{dec}(m_e) \neq \perp_P$.

```
Algorithm enc($m$)
  if $st_e = \perp$ then
    return ($\perp, \perp$)
  end if
  if $SN_e \geq 2^{32}$ or $|m| \geq 2^{32} - 5$ then
    $st_e \leftarrow \perp$
    return ($\perp, \perp$)
  else
    $PL \leftarrow L - ((|m| + 5) \bmod L)$
    if $PL < 4$ then
      $PL \leftarrow PL + L$
    end if
    $PD \xleftarrow{r} \{0,1\}^{8 \cdot PL}$
    $LF \leftarrow (1 + |m| + PL)$
    $m_e \leftarrow \langle LF \rangle_4 \| \langle PL \rangle_1 \| m \| PD$
    $m_t \leftarrow SN_e \| m_e$
    $SN_e \leftarrow SN_e + 1$
    return ($m_e, m_t$)
  end if
```

```
Algorithm dec($m_e$)
  if $st_d = \perp$ then
    return $\perp$
  end if
  if $SN_d \geq 2^{32}$ then
    $st_d \leftarrow \perp$
    return $\perp$
  else
    Attempt to parse $m_e$ as:
    $\langle LF \rangle_4 \| \langle PL \rangle_1 \| m \| PD$ where
    $PL \geq 4$, $|PD| = PL$ and $|m| \geq 0$.
    if parsing fails then
      $st_d \leftarrow \perp$
      return $\perp_P$
    else
      $SN_d \leftarrow SN_d + 1$
      return $m$
    end if
  end if
```

Fig. 1. Encoding Scheme for SSH

The specific encoding scheme used by the SSH BPP specification is shown in Figure 1. Here, L denotes the block-size in bytes of the block cipher in use (or the default value of 8 if a stream cipher such as ARCFOUR is being used), LF denotes the length field, PL denotes the padding length and PD denotes the padding bytes. The padding bytes are assumed to be random in our security analysis, though our security results also hold for any distribution on the padding bytes (including fixed bytes). We test that the message m submitted for encoding contains at most $2^{32}-6$ bytes, so that the length of the encoded message can be recorded in the 4-byte length field. Each of the two algorithms enc, dec maintains a separate state of the form (st, SN), initially set to $(\varepsilon, 0)$. In each case, the first component st maintains the status of the algorithm, i.e. if the algorithm is in an error state or not. This is used to model the effect of an SSH connection tear-down when an error occurs. The second component SN denotes a 32-bit sequence number. Note that RFC 4344 [4] states that when the sequence number SN wraps around, new keys must be negotiated. For simplicity in our analysis, we model this by forcing st_e (or st_d) to $\perp$ when SN_e (or SN_d) reaches 2^{32}. In our full model of the SSH BPP, this has the effect of removing the adversary's access to the encryption or decryption oracle. This ensures that each value of SN_e or SN_d is used only once, and is equivalent to enforcing rekeying when the relevant sequence number wraps around. Note that in [3], the equivalent state consists of a single value which is "over-loaded" to carry both the algorithm status and sequence number. For concreteness, Figure 1 shows the specific parsing steps carried out by OpenSSH during decoding. Other implementations may perform different checks here.

Encryption Scheme: The construction of SSH that we consider uses counter mode encryption of a block cipher, and is called SSH-CTR in [3]. When we come to formally analyze the security of SSH-CTR, we will regard the block cipher as being a pseudorandom function (prf) family rather than as a pseudorandom permutation family. This allows us to directly use some of the results from [2]. Our definition for a prf family can be found in the full version of this paper [9].

We give a formal definition for counter mode encryption based on a prf family F, $\mathrm{CTR}[F] = (\mathcal{K}\text{-CTR}, \mathcal{E}\text{-CTR}, \mathcal{D}\text{-CTR})$ in [9]. The key generation algorithm $\mathcal{K}$-CTR outputs a random k-bit key K_e for the underlying prf family F, therefore specifying a function F_{K_e} having l-bit inputs and L-byte outputs. Note that in practice we have $l = 8L$ since all block ciphers have equal input and output size. The key generation algorithm also outputs a random l-bit initial counter ctr, which is used to initialise counters in the encryption and decryption algorithms $\mathcal{E}$-CTR, $\mathcal{D}$-CTR.

We also define the scheme $\mathrm{CTR}^{\mathcal{EC}}[F]$ to be a combination of counter mode encryption and the encoding/decoding scheme from Figure 1. Full details of this scheme appear in [9]. This construction is not used in SSH, but is needed as a step in our security analysis in Section 5.

Message Authentication Scheme: A *message authentication scheme* (MAC) $\mathcal{MA} = (\mathcal{K}_t, \mathcal{T}, \mathcal{V})$ consists of three algorithms. The key generation algorithm $\mathcal{K}_t$

returns a key K_t. The tag algorithm $\mathcal{T}$, which may be stateful and randomised, takes as input the key K_t and an encoded message m_t and returns a tag τ. The verification algorithm $\mathcal{V}$, which is deterministic and stateless, takes as finput the key K_t and an encoded message m_t and a candidate tag τ' and outputs a bit. For any key K_t, message m_t and internal state of $\mathcal{T}_{K_t}$, we require that $\mathcal{V}_{K_t}(m_t, \mathcal{T}_{K_t}(m_t)) = 1$.

3.3 Encode-then-Encrypt&MAC

With the above components defined, we are now ready to define SSH-CTR. Note that our version is significantly different from that considered in [3] because of the new features that we discussed in Section 2.1.

Our construction of SSH-CTR is an Encode-then-Encrypt&MAC construction with plaintext-dependent decryption. We define SSH-CTR = ($\mathcal{K}$-SSH-CTR, $\mathcal{E}$-SSH-CTR, $\mathcal{D}$-SSH-CTR) in Figure 2. This makes use of the encoding scheme $\mathcal{EC}$ described in Section 3.2, the encryption scheme CTR[F] and a message authentication scheme $\mathcal{MA}$, where the length of the MAC tag is `maclen`. It also makes use of a length checking algorithm len that we discuss below. Note that this construction is stateful. The encryption state arises from the counter mode state ctr_e combined with the state (st_e, SN_e) of the algorithm enc. The decryption state arises from the counter mode state ctr_d, the state (st_d, SN_d) of the algorithm dec, and the ciphertext buffer `cbuff`. We will refer to the scheme SSH-CTR[F] whenever we wish to highlight the scheme's reliance on a particular function family F in the encryption component.

The key generation algorithm $\mathcal{K}$-SSH-CTR selects keys for counter mode encryption and the MAC algorithm uniformly at random from the relevant keyspaces. This represents a significant abstraction from reality in our description of SSH-CTR, since in practice these keys and the initial counter value ctr are derived in a pseudorandom manner from the keying material established during SSH's key exchange protocol. The decryption algorithm $\mathcal{D}$-SSH-CTR is considerably more complex than one might expect. This complexity is required to accurately model all the features of the SSH specification and the OpenSSH implementation. $\mathcal{D}$-SSH-CTR operates in 3 distinct stages.

In Stage 1, a sequence of ciphertext bytes c of arbitrary length is received and appended to the ciphertext buffer `cbuff`.

In Stage 2 of $\mathcal{D}$-SSH-CTR, once sufficient bytes have arrived to process the first block of ciphertext, the packet length field is extracted, and length checking is performed by making a call to the function len. This accords with our discussion in Section 2.1. The function len is shown as part of Figure 2. It takes as input a single block of plaintext, and returns either the content of the length field (as an integer) or a failure symbol $\perp_L$. The exact details of length checking, and how to behave if length checking fails, is implementation-specific and not specified in the RFCs. Figure 2 shows the exact checks carried out by OpenSSH version 5.2 in counter mode; our subsequent analysis still holds so long as the algorithm at a minimum checks that the total number of encrypted bytes (i.e. excluding the MAC tag) indicated by the length field is a multiple of the block-size L,

```
Algorithm $\mathcal{K}$-SSH-CTR$(k)$
  $K_e \xleftarrow{r} \mathcal{K}_e(k)$
  $K_t \xleftarrow{r} \mathcal{K}_t(k)$
  $ctr \xleftarrow{r} \{0,1\}^l$
  return $K_e, K_t$
```

```
Algorithm $\mathcal{E}$-SSH-CTR$_{K_e,K_t}(m)$
  if $st_e = \perp$ then
    return $\perp$
  end if
  $(m_e, m_t) \leftarrow \mathsf{enc}(m)$
  if $m_e = \perp$ then
    $st_e \leftarrow \perp$
    return $\perp$
  else
    $c \leftarrow \mathcal{E}\text{-CTR}_{K_e}(m_e)$
    $\tau \leftarrow \mathcal{T}_{K_t}(m_t)$
    return $c \| \tau$
  end if
```

```
Algorithm $\mathsf{len}(m)$ $(|m| = L)$
  Parse $m$ as $\langle LF \rangle_4 \| R$
  if $LF \leq 5$ or $LF \geq 2^{18}$ then
    return $\perp_L$
  else if $LF + 4 \bmod L \neq 0$ then
    return $\perp_L$
  else
    return $LF$
  end if
```

```
Algorithm $\mathcal{D}$-SSH-CTR$_{K_e,K_t}(c)$
  if $st_d = \perp$ then
    return $\perp$
  end if
  {Stage 1}
  cbuff $\leftarrow$ cbuff$\|c$
  {Stage 2}
  if $m_e = \varepsilon$ and $|\mathtt{cbuff}| \geq L$ then
    Parse cbuff as $\tilde{c} \| A$ (where $|\tilde{c}| = L$)
    $m_e[1] \leftarrow \mathcal{D}\text{-CTR}_{K_e}(\tilde{c})$
    $LF \leftarrow \mathsf{len}(m_e[1])$
    if $LF = \perp_L$ then
      $st_d \leftarrow \perp$
      return $\perp_L$
    else
      need $= 4 + LF +$ maclen
    end if
  end if
  {Stage 3}
  if $|\mathtt{cbuff}| \geq L$ then
    if $|\mathtt{cbuff}| \geq$ need then
      Parse cbuff as $\bar{c}[1\ldots n] \| \tau \| B$,
      where $|\bar{c}[1\ldots n] \| \tau| =$ need,
      and $|\tau| =$ maclen
      $m_e[2\ldots n] \leftarrow \mathcal{D}\text{-CTR}_{K_e}(\bar{c}[2\ldots n])$
      $m_e \leftarrow m_e[1] \| m_e[2\ldots n]$
      $m_t \leftarrow SN_d \| m_e$
      $v \leftarrow \mathcal{V}_{K_t}(m_t, \tau)$
      if $v = 0$ then
        $st_d \leftarrow \perp$
        return $\perp_A$
      else
        $m \leftarrow \mathsf{dec}(m_e)$
        $m_e \leftarrow \varepsilon$, cbuff $\leftarrow B$
        return $m$
      end if
    end if
  end if
```

Fig. 2. SSH-CTR, SSH using counter mode encryption

and fails if this is not the case. For further discussion, see the full version [9]. Note that when length checking fails in OpenSSH version 5.2 in counter mode, an error message is sent and the SSH connection is torn down. We model this by outputting a length error $\perp_L$ and setting the state st_d to $\perp$. Because the first action of $\mathcal{D}$-SSH-CTR is to simply return $\perp$ if st_d is already equal to $\perp$, our description of SSH-CTR models the subsequent connection tear-down seen in OpenSSH. If the length checks pass, then $\mathcal{D}$-SSH-CTR proceeds to use the returned value of LF to determine the value of `need`, which is the number of additional ciphertext bytes that are needed before the entire ciphertext (including MAC tag) is adjudged to have arrived. This makes the decryption algorithm plaintext-dependent and no further output is produced by $\mathcal{D}$-SSH-CTR until the complete ciphertext has arrived and its MAC has been checked.

In Stage 3 of $\mathcal{D}$-SSH-CTR, ciphertext bytes that have been buffered in `cbuff` during Stage 1 are processed. Note that our model allows the recipient to receive more data than he expects; this data is denoted by B in Stage 3. This data is assumed to be the start of the next ciphertext message and so we reinitialise `cbuff` with this data at the end of Stage 3. Once the buffer contains sufficient data (as determined by the variable `need`), the decryption algorithm uses counter mode to obtain the encoded plaintext m_e and the message m_t to be verified by the MAC algorithm (this consists of m_e with the sequence number prepended). The MAC tag is then checked, and, if it verifies successfully, the encoded plaintext m_e is passed to the `dec` algorithm (as defined in Figure 1). Notice that three types of error can arise during this stage: a failure of the MAC verification, resulting in output $\perp_A$, a failure of parsing during decoding, resulting in output $\perp_P$, or a wrap-around of the sequence number SN_d during decoding, resulting in output $\perp$. When any of these errors arises, the state st_d of the decryption algorithm is set as $\perp$. This state is checked at the start of every oracle query and if it equals $\perp$, then an error message $\perp$ is returned. In this way, our description of SSH-CTR models the subsequent connection tear-down seen in OpenSSH.

This description of SSH-CTR faithfully models OpenSSH in counter mode, in the sense of having buffered, plaintext-dependent decryption, and with errors arising at exactly the same points during decryption and based on the same failure conditions that are tested in OpenSSH. There are other ways in which to implement SSH and still be RFC-compliant. For example, the full decoding of the message, and hence parsing checks, could be performed before the MAC verification, as is the case in the construction of SSH-CTR given in [3].

4 Security Models

4.1 Chosen Plaintext Security

We begin by extending the usual left-or-right (LOR) indistinguishability game for a CPA adversary from [2] to handle stateful encryption and leakage of length information. This extension is only needed at intermediate steps in our security analysis, while we are primarily interested in the security of the SSH BPP under chosen ciphertext attacks. For this reason, we content ourselves with chosen plaintext security definitions that are tied to the particular schemes SSH-CTR$[F]$ and CTR$^{\mathcal{EC}}[F]$ that we need to analyze.

In the usual LOR-CPA model the adversary is given access to a left-or-right encryption oracle $\mathcal{E}(\mathcal{LR}(\cdot,\cdot,b))$, where $b \in \{0,1\}$. This oracle takes as input two messages m_0 and m_1. If $b = 0$ it outputs the encryption of m_0 and if $b = 1$ it outputs the encryption of m_1. It is the adversary's challenge to determine the bit b. The advantage of such an adversary is defined in the usual way. Our extension of the LOR-CPA model makes it stateful and incorporates leakage of a length field. To achieve the former, we incorporate explicit sequence numbers in the model. To achieve the latter, we provide the adversary with access to a length revealing oracle $\mathcal{L}(\cdot)$ whose operation is specific to the particular scheme under study. For the schemes SSH-CTR[F] and CTR$^{\mathcal{EC}}$[F], the oracle takes as input a block c which is treated as the first block of a new message; the oracle decrypts this block to retrieve the length field and performs the required length checking functions, and then outputs either the length field LF or the symbol $\perp_L$ signifying an invalid length field. We require that $\mathcal{L}(\cdot)$ maintains its own view of any internal state of the underlying encryption scheme, according to the queries it receives. For the schemes we consider, this is done by increasing a counter value ctr_l by a number that is determined by the length field, and increasing a sequence number SN_l by 1, each time the oracle is called; at the start of the security game, ctr_l and SN_l are set to the corresponding values held at the encryption oracle. The detailed operation of the length oracle associated with the schemes SSH-CTR[F] and CTR$^{\mathcal{EC}}$[F] can be found in the full version [9]. We name our new model LOR-LLSF-CPA, where “LLSF” stands for “length leaking stateful”.

In [3], decryption queries are defined to be either “in-sync” or “out-of-sync” with respect to the sequence number at the encryption oracle. We introduce a similar concept for length oracle queries in our next definition:

Definition 1. ***[LOR-LLSF-CPA]***
Consider the stateful encryption scheme $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ with an associated length oracle $\mathcal{L}(\cdot)$. Let $b \in \{0,1\}$ and $k \in \mathbb{N}$. Let $\mathcal{A}$ be an attacker that has access to the oracles $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$ and $\mathcal{L}(\cdot)$ The game played is as follows:

$$\begin{array}{l} \boldsymbol{Exp}^{lor\text{-}llsf\text{-}cpa\text{-}b}_{\mathcal{E},\mathcal{A}}(k) \\ K \xleftarrow{r} \mathcal{K}(k) \\ b' \leftarrow \mathcal{A}^{\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b)),\mathcal{L}(\cdot)} \\ \textbf{\textit{return}}\ b' \end{array}$$

For all queries (m_0, m_1) to $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$, we require that $|\mathsf{enc}(m_0)| = |\mathsf{enc}(m_1)|$. In this model the adversary has the possibility of making three different types of query to $\mathcal{L}$. Let SN_e denote the sequence numbers at the encryption oracle and let SN_l denote the sequence numbers at the length oracle.

- *A query c to $\mathcal{L}$ when the length oracle has sequence number SN_l is said to be in-sync if c is equal to the first block of ciphertext output by the encryption oracle when it had sequence number $SN_e = SN_l$.*
- *A query c to $\mathcal{L}$ when the length oracle has sequence number SN_l is said to be an out-of-sync current state query if c is not equal to the first block of ciphertext output by the encryption oracle when it had sequence number $SN_e = SN_l$.*
- *A query to $\mathcal{L}$ when the length oracle has sequence number SN_l is said to be an out-of-sync future state query if $SN_l > SN_e$, where SN_e is the sequence number used by the encryption oracle when responding to its most recent query.*

We require that the response to any further length oracle queries following the first out-of-sync query is $\perp$.

The attacker wins when $b' = b$*, and its advantage is defined to be:*

$$\boldsymbol{Adv}_{\mathcal{SE},\mathcal{A}}^{lor\text{-}llsf\text{-}cpa}(k) = \Pr[\boldsymbol{Exp}_{\mathcal{SE},\mathcal{A}}^{lor\text{-}llsf\text{-}cpa\text{-}1}(k) = 1] - \Pr[\boldsymbol{Exp}_{\mathcal{SE},\mathcal{A}}^{lor\text{-}llsf\text{-}cpa\text{-}0}(k) = 1].$$

The advantage function of the scheme is defined to be

$$\boldsymbol{Adv}_{\mathcal{SE}}^{lor\text{-}llsf\text{-}cpa}(k, t, q_e, \mu_e, q_l) = \max_{\mathcal{A}}\{\boldsymbol{Adv}_{\mathcal{SE},\mathcal{A}}^{lor\text{-}llsf\text{-}cpa}(k)\}$$

for any integers t, q_e, μ_e, q_l*. The maximum is over all adversaries* $\mathcal{A}$ *with time complexity* t*, making at most* q_e *queries to the encryption oracle, totalling at most* μ_e *bits in each of the left and right inputs, and* q_l *queries to the length revealing oracle.*

4.2 Chosen Ciphertext Security

Now we consider chosen ciphertext attackers. We introduce a new security notion for left-or-right indistinguishability against chosen-ciphertext attackers for buffered, stateful decryption (LOR-BSF-CCA). In this model, which extends the IND-SFCCA model of [3], the adversary is given access to an encryption oracle and to a *buffered* decryption oracle. The model applies for any encryption scheme in which the decryption oracle maintains a buffer of as-yet-unprocessed ciphertext bytes `cbuff` and in which encryption and decryption states include sequence numbers which are incremented after each successful operation. For reasons explained in Section 2.1, we need to limit the attacker's queries to the encryption oracle to pairs of messages (m_0, m_1) having the same length when encoded.

Definition 2. ***[LOR-BSF-CCA]***
Consider the symmetric encryption scheme $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ *with buffered, stateful decryption. Let* $b \in \{0,1\}$ *and* $k \in \mathbb{N}$*. Let* $\mathcal{A}$ *be an attacker that has access to the oracles* $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$ *and* $\mathcal{D}_K(\cdot)$*. The game played is as follows:*

```
$\boldsymbol{Exp}_{\mathcal{SE},\mathcal{A}}^{lor\text{-}bsf\text{-}cca\text{-}b}(k)$
$K \overset{r}{\leftarrow} \mathcal{K}(k)$
$b' \leftarrow \mathcal{A}^{\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b)),\mathcal{D}_K(\cdot)}(k)$
$\boldsymbol{return}\ b'$
```

We require that for all queries (m_0, m_1) *to* $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$*,* $|\mathsf{enc}(m_0)| = |\mathsf{enc}(m_1)|$*. In this model the adversary has the possibility of making three different types of decryption query. Let* SN_e *denote the sequence numbers at the encryption oracle and let* SN_d *denote the sequence numbers at the decryption oracle. Recall that, since the adversary can deliver ciphertexts in a byte-wise fashion to the decryption oracle, the same value of* SN_d *may be involved in processing a sequence of ciphertext queries.*

- *The sequence of decryption queries corresponding to the sequence number SN_d is said to be in-sync if, after input of the final query in the sequence, the ciphertext buffer* `cbuff` *has as a prefix the output from the encryption oracle for sequence number $SN_e = SN_d$. The response from an in-sync query is not returned to the adversary.*
- *The sequence of decryption queries corresponding to the sequence number SN_d is said to be an out-of-sync current state query if, after input of the final query in the sequence, the ciphertext buffer* `cbuff` *does not have the output from the encryption oracle for sequence number $SN_e = SN_d$ as a prefix.*
- *The sequence of decryption queries corresponding to the sequence number SN_d is said to be an out-of-sync future state query if $SN_d > SN_e$, where SN_e is the sequence number used by the encryption oracle when responding to its most recent query.*

The response to any further decryption queries following an out-of-sync query is the $\perp$ symbol.

The attacker wins when $b' = b$, and its advantage is defined to be:

$$\boldsymbol{Adv}^{lor\text{-}bsf\text{-}cca}_{\mathcal{SE},\mathcal{A}}(k) = \Pr[\boldsymbol{Exp}^{lor\text{-}bsf\text{-}cca\text{-}1}_{\mathcal{SE},\mathcal{A}}(k) = 1] - \Pr[\boldsymbol{Exp}^{lor\text{-}bsf\text{-}cca\text{-}0}_{\mathcal{SE},\mathcal{A}}(k) = 1].$$

The advantage function of the scheme is defined to be

$$\boldsymbol{Adv}^{lor\text{-}bsf\text{-}cca}_{\mathcal{SE}}(k, t, q_e, \mu_e, q_d, \mu_d) = \max_{\mathcal{A}}\{\boldsymbol{Adv}^{lor\text{-}bsf\text{-}cca}_{\mathcal{SE},\mathcal{A}}(k)\}$$

for any integers $t, q_e, \mu_e, q_d, \mu_d$. The maximum is over all adversaries $\mathcal{A}$ with time complexity t, making at most q_e queries to the encryption oracle, totalling at most μ_e bits in each of the left and right inputs, and at most q_d series of queries to the decryption oracle, totalling at most μ_d bits.

In the model above, the response from an in-sync decryption query is not returned to the adversary. This is required in order to prevent the obvious and trivial attack in which the adversary simply queries the decryption oracle with the output from the encryption oracle. We include in-sync decryption queries in order to permit the adversary to observe the system's behaviour in encrypting messages of its choice and to let the adversary advance the sequence numbers maintained at the encryption and decryption oracles to values of its choice. We make the restriction that only one out-of-sync query is allowed for the same reason that this restriction is made in [3]: if the first out-of-sync query does not decrypt successfully, the decryption oracle enters a halting state anyway, while if it does, then our security analysis will show that the adversary has broken the strong unforgeability of the MAC scheme. Our security model and analysis can be extended to handle multiple out-of-sync decryption queries.

The specific decryption oracle we consider when analyzing the security of SSH-CTR operates exactly as the decryption algorithm $\mathcal{D}$-SSH-CTR in Section 3.3: the oracle takes as input an arbitrary number of bytes which is then added to `cbuff`; the decryption process uses the first plaintext block to determine how many bytes of ciphertext are needed to complete the packet; and the decryption process involves length checking, MAC checking, and decoding, with each of these steps potentially outputting a distinct error message. Also note that for SSH-CTR, the decryption oracle acts as a "bomb" oracle: when an error of any type occurs this oracle simply outputs $\perp$ in response to any further query. This models an attempt by the decrypting party to initiate an SSH connection tear-down. However, note that our model for SSH-CTR has separate states for encryption and decryption, so that the encryption oracle is not "lost" if the decryption oracle is. This allows us to model an adversary that outputs the relevant error messages. This description of SSH-CTR in the context of the LOR-BSF-CCA model is sufficiently rich to give the attacker all the capabilities exploited in the attacks of Albrecht *et al.* [1]. Thus, if we can prove SSH-CTR to be secure in the LOR-BSF-CCA sense, then attacks of the kind developed in [1] will be prevented.

4.3 Integrity of Ciphertexts

We next extend the INT-SFCTXT model from [3] to include buffered decryption. We call our new model "integrity of ciphertexts for buffered, stateful decryption" or INT-BSF-CTXT. The model again applies for any encryption scheme in which the decryption oracle maintains a buffer of as-yet-unprocessed ciphertext bytes `cbuff` and in which encryption and decryption states include sequence numbers which are incremented after each successful operation.

In this INT-BSF-CTXT model, the adversary has access to encryption and decryption oracles, and is considered successful if it is able to make an out-of-sync sequence of decryption queries that results in an output from the decryption oracle that is not a member of the set $\{\perp_L, \perp_A, \perp_P, \perp\}$. Again, the specific decryption oracle that we consider when analyzing the security of SSH-CTR operates exactly as the decryption algorithm $\mathcal{D}$-SSH-CTR in Section 3.3. The formal definition of the INT-BSF-CTXT model can be found in the full version of this paper [9].

4.4 Security of Message Authentication Schemes

Finally, we define two security notions for MACs. We will use the LOR-DCPA notion from [3], for distinct plaintext privacy of message authentication schemes. We will also use the standard SUF-CMA model for strong unforgeability of MACs. The formal definitions for these notions can also be found in the full version [9].

5 Security Analysis

We will now present our main result, Theorem 1. This theorem provides a concrete security guarantee for the scheme SSH-CTR$[F]$ in terms of security properties of the prf family F and MAC scheme $\mathcal{MA}$ used in its construction. The structure of our proof follows that in [3], but with significant modifications being needed to handle the new features of our security model and adversary. Our proof is valid no matter what length checks are performed by the encoding scheme, so long as the minimal length check described previously is included. Our proof is also valid (and in fact can be tightened slightly) if the random padding bytes in the encoding scheme are replaced by fixed bytes. It is also valid no matter what specific parsing checks are carried out, provided that the encoding scheme is correct. With the exception of our main result, the proofs are given in the full version of this paper [9].

Theorem 1. *Let SSH-CTR[F] be the combined encryption scheme for the encoding scheme $\mathcal{EC}$, counter mode encryption CTR[F] and a message authentication scheme $\mathcal{MA}$. Then for $q_e, q_d \leq 2^{32}$, $\mu_e \leq 8L2^l - 8q_e(8+L)$ and any t, k, μ_d, we have:*

$$\begin{aligned}&\boldsymbol{Adv}^{lor\text{-}bsf\text{-}cca}_{SSH\text{-}CTR[F]}(k,t,q_e,\mu_e,q_d,\mu_d)\\ &\leq 2\boldsymbol{Adv}^{suf\text{-}cma}_{\mathcal{MA}}(k,t,q_t,\mu_t,q_v,\mu_v) + 2\boldsymbol{Adv}^{prf}_{F}(k,t',q_F) + 4\boldsymbol{Adv}^{prf}_{\mathcal{T}}(k,t'',q_t)\end{aligned}$$

where $q_t = q_e$, $\mu_t \leq \mu_e + 8(L+12)q_e$, $q_v = q_d$, $\mu_v \leq \mu_d + 32q_d$, $q_F \leq q_l + \mu_e/8L + q_e(1+8/L)$, $t' = O(t)$ and $t'' = O(t)$.

Proof of Theorem 1: This follows from Theorem 2 and Lemmas 1, 2, 3, 4 and 5. □

The following is an extension of a result of Bellare and Namprempre [5]; here we consider buffered, stateful decryption and include in our model potential errors arising from length checking, MAC failures and parsing failures.

Theorem 2. *Let SSH-CTR[F] be the combined encryption scheme for the encoding scheme $\mathcal{EC}$, counter mode encryption CTR[F] and a message authentication scheme $\mathcal{MA}$. Then for any $k, t, q_e, \mu_e, q_d, \mu_d$, we have:*

$$\begin{aligned}&\boldsymbol{Adv}^{lor\text{-}bsf\text{-}cca}_{SSH\text{-}CTR[F]}(k,t,q_e,\mu_e,q_d,\mu_d)\\ &\quad\leq 2\boldsymbol{Adv}^{int\text{-}bsf\text{-}ctxt}_{SSH\text{-}CTR[F]}(k,t,q_e,\mu_e,q_d,\mu_d) + \boldsymbol{Adv}^{lor\text{-}llsf\text{-}cpa}_{SSH\text{-}CTR[F]}(k,t,q_e,\mu_e,q_l)\end{aligned}$$

where $q_l = q_d$.

Lemma 1. *Let SSH-CTR[F] be the combined encryption scheme for the encoding scheme $\mathcal{EC}$, counter mode encryption CTR[F] and a message authentication scheme $\mathcal{MA}$. Then for $q_e, q_d \leq 2^{32}$ and any k, t, μ_e, μ_d, we have:*

$$\boldsymbol{Adv}^{int\text{-}bsf\text{-}ctxt}_{SSH\text{-}CTR[F]}(k,t,q_e,\mu_e,q_d,\mu_d) \leq \boldsymbol{Adv}^{suf\text{-}cma}_{\mathcal{MA}}(k,t,q_t,\mu_t,q_v,\mu_v)$$

where $q_t = q_e$, $\mu_t \leq \mu_e + 8(L+12)q_e$, $q_v = q_d$, and $\mu_v \leq \mu_d + 32q_d$.

Lemma 2. *Let SSH-CTR[F] be the combined encryption scheme for the encoding scheme $\mathcal{EC}$, counter mode encryption CTR[F] and a message authentication scheme $\mathcal{MA}$. Then for $q_e, q_l \leq 2^{32}$ and any k, t, μ_e, we have:*

$$\begin{aligned}&\boldsymbol{Adv}^{lor\text{-}llsf\text{-}cpa}_{SSH\text{-}CTR[F]}(k,t,q_e,\mu_e,q_l)\\ &\quad\leq \boldsymbol{Adv}^{lor\text{-}llsf\text{-}cpa}_{CTR^{\mathcal{EC}}[F]}(k,t',q_e,\mu_e,q_l) + 2\boldsymbol{Adv}^{lor\text{-}dcpa}_{\mathcal{MA}}(k,t'',q_t,\mu_t)\end{aligned}$$

where $q_t = q_e$, $t' = O(t)$, $t'' = O(t)$, and $\mu_t \leq \mu_e + 16(L+12)q_e$.

Lemma 3. *Suppose F is a prf family with input length l bits and output length L bytes. Let $R = Rand^{l \to L}$ be the set of all functions mapping l-bit strings to L-byte strings. Then for any k, t, q_e, μ_e, q_l, we have:*

$$\begin{aligned}&\boldsymbol{Adv}^{\textit{lor-llsf-cpa}}_{CTR^{\mathcal{EC}}[F]}(k, t, q_e, \mu_e, q_l)\\ &\quad \leq 2\boldsymbol{Adv}^{prf}_{F}(k, t', q_F) + \boldsymbol{Adv}^{\textit{lor-llsf-cpa}}_{CTR^{\mathcal{EC}}[R]}(k, t, q_e, \mu_e, q_l)\end{aligned}$$

where $q_F \leq q_l + \mu_e/8L + q_e(40 + 8(3 + L))/8L$ and $t' = O(t)$.

Lemma 4. *For any k, t, q_l, q_e and $\mu_e \leq 8L2^l - 8q_e(8 + L)$ we have:*

$$\boldsymbol{Adv}^{\textit{lor-llsf-cpa}}_{CTR^{\mathcal{EC}}[R]}(k, t, q_e, \mu_e, q_l) = 0.$$

Lemma 5. *Let $\mathcal{MA}$ be a message authentication scheme. Then for any k, t and q_t, we have:*

$$\boldsymbol{Adv}^{\textit{lor-dcpa}}_{\mathcal{MA}}(k, t, q_t, \mu_t) \leq 2\boldsymbol{Adv}^{prf}_{\mathcal{T}}(k, t', q_t)$$

where $t' = O(t)$.

6 Conclusion

We have extended the security model of Bellare *et al.* [3] to develop a model suited to analyzing the SSH BPP. We gave a description of SSH-CTR that is closely linked to the specification of SSH in the RFCs and the OpenSSH implementation of SSH. We then proved the security of SSH-CTR in the extended model. Our approach is sufficiently powerful to incorporate the attacks of Albrecht *et al.* [1]. This helps to close the gap that exists between the formal security analysis of SSH and the way in which SSH should be (and is in practice) implemented.

Our approach can be seen as an attempt to expand the scope of provable security to incorporate the fine details of cryptographic implementations. We grant the attacker a much wider and more realistic set of ways of interacting with the SSH protocol than in the previous analysis of [3]. We believe that our approach captures more of the cryptographically relevant features of the SSH BPP, including plaintext-dependent, byte-wise decryption and detailed modeling of the errors that can arise during cryptographic processing in the SSH BPP.

References

1. Albrecht, M.R., Paterson, K.G., Watson, G.J.: Plaintext recovery attacks against SSH. In: IEEE Symposium on Security and Privacy, pp. 16–26. IEEE Computer Society, Los Alamitos (2009)
2. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption. In: Proceedings of 38th Annual Symposium on Foundations of Computer Science (FOCS 1997), pp. 394–403. IEEE, Los Alamitos (1997)

3. Bellare, M., Kohno, T., Namprempre, C.: Breaking and provably repairing the SSH authenticated encryption scheme: A case study of the encode-then-encrypt-and-MAC paradigm. ACM Transactions on Information and Systems Security 7(2), 206–241 (2004)
4. Bellare, M., Kohno, T., Namprempre, C.: The Secure Shell (SSH) Transport Layer Encryption Modes. RFC 4344 (January 2006), http://www.ietf.org/rfc/rfc4344.txt
5. 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)
6. Canvel, B., Hiltgen, A.P., Vaudenay, S., Vuagnoux, M.: Password interception in a SSL/TLS channel. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 583–599. Springer, Heidelberg (2003)
7. CPNI Vulnerability Advisory. Plaintext recovery attack against SSH (November 14, 2008), http://www.cpni.gov.uk/Docs/Vulnerability_Advisory_SSH.txt (revised November 17, 2008)
8. Krawczyk, H.: The order of encryption and authentication for protecting communications (or: How secure is SSL?). In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 310–331. Springer, Heidelberg (2001)
9. Paterson, K.G., Watson, G.J.: Plaintext-Dependent Decryption: A Formal Security Treatment of SSH-CTR. Cryptology ePrint Archive, Report 2010/095 (2010), http://eprint.iacr.org/2010/095
10. SSH usage profiling, http://www.openssh.org/usage/index.html
11. Ylonen, T., Lonvick, C.: The Secure Shell (SSH) Protocol Architecture. RFC 4251 (January 2006), http://www.ietf.org/rfc/rfc4251.txt
12. Ylonen, T., Lonvick, C.: The Secure Shell (SSH) Authentication Protocol. RFC 4252 (January 2006), http://www.ietf.org/rfc/rfc4252.txt
13. Ylonen, T., Lonvick, C.: The Secure Shell (SSH) Transport Layer Protocol. RFC 4253 (January 2006), http://www.ietf.org/rfc/rfc4253.txt
14. Ylonen, T., Lonvick, C.: The Secure Shell (SSH) Connection Protocol. RFC 4254 (January 2006), http://www.ietf.org/rfc/rfc4254.txt

Computational Soundness, Co-induction, and Encryption Cycles

Daniele Micciancio

University of California at San Diego,
9500 Gilman Dr., Mail Code 0404,
La Jolla, CA 92093, USA
daniele@cs.ucsd.edu

Abstract. We analyze the relation between induction, co-induction and the presence of encryption cycles in the context of computationally sound symbolic equivalence of cryptographic expressions. Our main finding is that the use of co-induction in the symbolic definition of the adversarial knowledge allows to prove soundness results without the need to require syntactic restrictions, like the absence of encryption cycles, common to most previous work in the area. Encryption cycles are relevant only to the extent that the key recovery function associated to acyclic expressions can be shown to have a unique fixed point. So, when a cryptographic expression has no encryption cycles, the inductive (least fixed point) and co-inductive (greatest fixed point) security definitions produce the same results, and the computational soundness of the inductive definitions for acyclic expressions follows as a special case of the soundness of the co-inductive definition.

Keywords: Computational soundness, co-induction, greatest fixed points, formal methods for security, symbolic encryption, encryption cycles.

1 Introduction

The symbolic approach to security analysis (pioneered by Dolev and Yao in [1]) has been very useful in the construction and application of automated reasoning tools for the analysis of cryptographic protocols, like the Murphi model checker [2] and the Isabelle theorem prover [3], just to name two representative examples. However, the simplicity of the associated adversarial model (which enables the construction of automated analysis tools) is also the main weakness of symbolic security analysis: security is guaranteed only against attackers that abide to the rules of the Dolev-Yao model. In practice, one needs security against any (computationally feasible) attack as typically considered in modern computational cryptography. In the last few years, starting with the seminal work of Abadi and Rogaway [4], there has been considerable progress in understanding the relation between symbolic security analysis, and computational cryptography. Yet, it is fair to say that many problems related to the connection of symbolic and computational cryptography are still wide open.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 362–380, 2010.

The aim of this paper is to explore one specific aspect that sets the symbolic and computational models apart, and that has not received much attention so far: the use of induction versus co-induction in security proofs. We do so in the simplest possible setting considered in the literature: the indistinguishability of cryptographic expressions, i.e., expressions like $(\{\!|d_1|\!\}_{k_1}, \{\!|k_1|\!\}_{k_2})$, where $\{\!|m|\!\}_k$ represents the encryption of message m under key k. These are the expressions typically used to model messages in cryptographic protocols. For example, the above expression may be used to represent the message in a protocol where a long term key k_2 is used to encrypt a session key k_1, which in turn is used to encrypt the actual message d_1. The standard notion of equivalence in cryptography is computational indistinguishability: two expressions are equivalent if no probabilistic polynomial time adversary can distinguish the probability distributions naturally associated to the two expressions in an actual execution of the protocol. In the symbolic setting, equivalence is usually defined by mapping each expression to a corresponding pattern. For example, the expression $(\{\!|d_1|\!\}_{k_1}, \{\!|d_1|\!\}_{k_2}, k_2)$ may be mapped to the pattern $(\{\!|\Box|\!\}_{k_1}, \{\!|d_1|\!\}_{k_2}, k_2)$ to model the fact that an adversary observing the messages $\{\!|d_1|\!\}_{k_1}$,$\{\!|d_1|\!\}_{k_2}$ and k_2, can recover the key k_2, decrypt the second ciphertext to d_1, and even detect that the first ciphertext uses a key different from k_2 (e.g., because decryption under k_2 fails), but cannot tell that the first ciphertext encrypts the same message d_1 as the second.

In the seminal paper [4], Abadi and Rogaway showed that the meaning associated to cryptographic expressions by standard symbolic methods is computationally sound, in the sense that (under appropriate restrictions) if two expressions are symbolically equivalent (i.e., they have the same pattern), then the associated probability distributions are computationally indistinguishable.

Induction versus co-induction. As with most work in the area of formal analysis of security protocols, Abadi and Rogaway adopt an inductive approach to the symbolic modeling of adversarial knowledge: initially the attacker does not know any key and tries to learn as many keys as possible from a given cryptographic expression through the application of Dolev-Yao rules.[1] Technically, the knowledge of the adversary can be defined by associating to each cryptographic expression e a corresponding key recovery operator $\mathcal{F}_e$ (mapping sets of keys to sets of keys) which roughly corresponds to a single application of the Dolev-Yao decryption rules. The adversarial knowledge (obtained from observing the expression e) can be characterized as the *least fixed point* of the key recovery operator $\mathcal{F}_e$, i.e., the smallest set of keys S such that $\mathcal{F}_e(S) = S$. Operationally, this least fixed point can be obtained by starting from the empty set of keys

[1] A typical Dolev-Yao rule is that given a key k and the encryption $\{\!|m|\!\}_k$ of some message m under k, one can compute the plaintext m. Such rules are intended to capture the security features of the cryptographic operations used in the construction of messages, and the whole framework relies on the postulate that the adversary cannot perform any other operation. So, for example, given the cipher-text $\{\!|m|\!\}_k$, one cannot recover the message m, unless the encryption key k is already known.

$\emptyset$ (modeling the adversary's initial knowledge),[2] and applying the key recovery operator $\mathcal{F}_e$ to obtain more and more keys

$$\emptyset \subset \mathcal{F}_e(\emptyset) \subset \mathcal{F}_e^2(\emptyset) \subset \ldots \subset \mathcal{F}_e^m(\emptyset) = \mathcal{F}_e^{m+1}(\emptyset)$$

until the least fixed point $\mathcal{F}_e^m(\emptyset)$ is reached, and no additional keys can be recovered by further applications of $\mathcal{F}_e$.

In this paper we propose a dual, co-inductive approach. Technically, we propose to define the set of recoverable keys as the *greatest fixed point* of $\mathcal{F}_e$, i.e., the *largest* set of keys S such that $S = \mathcal{F}_e(S)$. As before, the greatest fixed point can be obtained by repeatedly applying the key recovery operator, but this time starting from the set of all keys **Keys**, and resulting in a sequence of smaller and smaller sets[3]

$$\mathbf{Keys} \supset \mathcal{F}_e(\mathbf{Keys}) \supset \ldots \supset \mathcal{F}_e^m(\mathbf{Keys}) = \mathcal{F}_e^{m+1}(\mathbf{Keys})$$

until the greatest fixed point $\mathcal{F}_e^m(\mathbf{Keys})$ is reached. Informally, we start from the set of all keys that appear in the expression in the role of plaintext of some encryption, and then iterate through the following process: the new set of keys is the set of all keys that can be deduced from the expression using the current set of keys for decryption. This is now the set of exposed keys. Intuitively, this corresponds to starting from the assumption that no key is guaranteed to be secure, and proving that more and more keys (namely, those in the complement of the sets $\mathcal{F}_e^i(\mathbf{Keys})$) are hidden to the adversary. As we are going to explain, this technical change in the definition of symbolic security has far reaching consequences when it comes to computational soundness.

Encryption cycles. In order to prove their soundness theorem, Abadi and Rogaway [4] need to impose a simple, but fundamental, technical restriction: the cryptographic expressions should not contain *encryption cycles*, e.g., sequences of messages of the form

$$\{\!\!\{k_1\}\!\!\}_{k_2}, \{\!\!\{k_2\}\!\!\}_{k_3}, \ldots, \{\!\!\{k_{n-1}\}\!\!\}_{k_n}, \{\!\!\{k_n\}\!\!\}_{k_1},$$

where each key k_i is encrypted under the next key $k_{(i \bmod n)+1}$ in the sequence, circularly. While encrypting a key with itself is typically considered a dangerous cryptographic practice, encryption cycles do occur in a small number of applications (e.g., credential systems [5], encrypted data backups, etc.), and the problem of designing encryption schemes supporting such a use has been the subject of many recent papers [6,7,8,9,10]. In the symbolic security setting it is customary to assume that encryption cycles are *secure*, in the sense that an adversary observing a sequence of circularly encrypted keys, cannot recover any of them.

[2] The knowledge of the keys of corrupted parties can be modeled by including those keys as part of the expression e.

[3] We remark that $\mathcal{F}_e(S)$ is defined as the set of keys that can be immediately recovered from the expression e using the keys in S for decryption. In particular, $\mathcal{F}_e(S)$ does not necessarily contain S as a subset, e.g., if some keys in S only occur in e as encryption keys, but never as (possibly encrypted) messages.

Our contribution. The main contribution of this paper is to highlight the relation of encryption cycles to inductive and co-inductive definitions of security. Specifically, we prove that

- (Theorem 1) if the set of recoverable keys is defined by co-induction (i.e., as the greatest fixed point of the key recovery operator), then the computational soundness result of Abadi and Rogaway holds without the need to impose syntactic restrictions: if two expressions (with or without encryption cycles) are symbolically equivalent, then their computational counterparts are indistinguishable.
- (Theorem 2) if an expression has no encryption cycles, then the associated key recovery function has a unique fixed point. In particular, the least and greatest fixed point coincide, and the conditional result of Abadi and Rogaway for acyclic expressions follows from the unrestricted result in the co-inductive setting.

Our results show that what sets the symbolic and computational frameworks apart (e.g., with respect to their ability to deal with encryption cycles,) is not the inherent difference between the computational and symbolic protocol execution models. Rather, it is the modeling of adversarial knowledge, which is typically *inductive* in the case of symbolic analysis, while intrinsically *co-inductive* in the computational setting.

At the technical level, our main computational soundness result (Theorem 1) is fairly general, and applicable to classes of cryptographic expressions that occur in many application domains, like secure multicast key distribution [11,12,13], and cryptographically controlled access to XML documents [14]. A follow-up paper [15] demonstrates the generality of our techniques using Theorem 1 to establish a computational soundness theorem for expressions with pseudo-random keys, as those used in [11,12,13]. As in this work, the results of [15] hold without the need to impose any syntactic restriction on the expressions.

We remark that the uniqueness of the fixed point for acyclic expressions is a purely symbolic result: neither the statement nor proof of Theorem 2 requires the use of the computational execution model. In fact, the proof is simple enough that one could model and verify it using an automated theorem prover. This fact, together with the simplicity of our computational soundness theorem (compared to analogous results from [4] and related papers), suggest that our greatest fixed point framework may be a useful tool even when one is interested in computational soundness with respect to the traditional inductive security definition. Specifically, in order to prove such computational soundness results one can

- first prove computational soundness for the corresponding co-inductive definition of security (possibly using Theorem 1), and
- then find and check (possibly with the help of automated symbolic reasoning tools) syntactic restrictions under which the inductive and co-inductive symbolic security definitions coincide.

So, even if induction may be the most intuitive and preferred way to analyze security protocols in practice, we believe that the co-inductive method would still be a valuable tool to establish the computational soundness of the inductive symbolic analysis.

Related work. Computationally sound symbolic analysis has been the topic of many recent works. This paper is most closely related to the line of work initiated by Abadi and Rogaway in [4], where secrecy properties with respect to passive adversaries are considered. Subsequent developments along the same lines include [16,12,11,14]. We mention that other approaches to symbolic analysis (e.g., [17]) inherit certain co-inductive ideas from the underlying process calculus, e.g., the use of bisimulation to define the equivalence between cryptographic processes. However, those frameworks are substantially more elaborate than the simple computational soundness setting considered in this paper, and their use of co-induction is quite different.

The problem of dealing with encryption cycles is a classic one in cryptography, already mentioned in the seminal paper [18] introducing the modern notion of computational security for encryption. Following [4], the problem has attracted renewed interest, both within the computational and symbolic setting. Two opposite approaches to resolving the discrepancy with respect to encryption cycles were proposed in [19,20].

In [20], Adao, Bana, Herzog and Scedrov prove a soundness theorem in the presence of key cycles using a strong security notion for encryption recently proposed in [21,5]. This notion, called security under "Key Dependent Messages" or "Key Dependent Input", allows encrypted messages to depend on the secret decryption key. At the time of [5,21,20], no scheme achieving this security notion was known in the standard model, and the only solutions (proposed in [5,21]) relied on the random oracle heuristic. Since then, the problem of building KDM-secure cryptographic primitives has been investigated in various works [7,6,8,10]. Similar results in the presence of active adversaries are given in [22]. In this paper, we do not consider the extended notions of computational security employed in these works, except for a brief discussion in Section 5. Rather, we focus on the question of the relation between symbolic and computational security when the standard computational security notion of indistinguishability under chosen plaintext attacks (still the golden standard in cryptography in the setting of passive attacks) is employed.

A different approach is used in[19], where Laud addresses the problem of reconciling symbolic and computational analysis in the presence of key cycles by strengthening the symbolic adversary. Specifically, Laud augments the entailment relation used in inductive approaches with a special rule that explicitly allows the symbolic adversary to break encryption cycles. As a result, Laud proves a computational soundness theorem for encrypted expressions (essentially equivalent to our Corollary 1) that does not require syntactic restrictions. Interestingly, greatest fixed point computations were suggested [23, equation 15] as an *algorithmic tool* to evaluate Laud's entailment relation. The main difference between [19,23] and our work is that [19,23] retain the inductive framework (and entailment relation, see Section 2) for modeling the adversarial knowledge, and resolve the encryption cycles issue using ad-hoc methods. Here we establish a close connection between greatest fixed points and cryptographic expressions at the semantic (computational soundness) level, and present a general approach (based on the use of co-induction) that can be generalized to a larger class of cryptographic expressions, e.g., the expressions with pseudo-random keys [13,12], secret sharing schemes [14], etc.

Organization. The rest of the paper is organized as follows. In Section 2 we present some preliminary definitions on symbolic expressions. (For an overview of the computational cryptography notions used in this paper the reader is referred to the appendix.) In Section 3 we present our main technical results. In Section 4 we illustrate our results on a simple example expression. Section 5 concludes with a discussion of future research directions and open problems.

2 Preliminaries

In this section we review the results and standard notation used in previous papers, mostly following the seminal work of Abadi and Rogaway [4]. For an overview of standard computational cryptography definitions and how symbolic expressions are evaluated to probability distributions over bitstrings, the reader is referred to the Appendix. Let $\mathbf{Exp}(\mathbf{Keys}, \mathbf{Data})$ be the set of cryptographic expressions built from two (disjoint) sets of key and data symbols $\mathbf{Keys}$,$\mathbf{Data}$, using pairing and encryption operations. Formally, $\mathbf{Exp}(\mathbf{Keys}, \mathbf{Data})$ is the set of expressions generated by the grammar

$$\mathbf{Exp} ::= \mathbf{Data} \mid \mathbf{Keys} \mid (\mathbf{Exp}, \mathbf{Exp}) \mid \{\!|\mathbf{Exp}|\!\}_{\mathbf{Keys}}, \tag{1}$$

where (e_1, e_2) denotes the concatenation of e_1 and e_2, and $\{\!|e|\!\}_k$ denotes the encryption of e under k. Define also the set of *patterns*

$$\mathbf{Pat}(\mathbf{Keys}, \mathbf{Data}) \subset \mathbf{Exp}(\mathbf{Keys} \cup \{\circ\}, \mathbf{Data} \cup \{\square\}), \tag{2}$$

where $\circ$ and $\square$ are two special symbols (not in $\mathbf{Keys}$ or $\mathbf{Data}$) which denote unknown keys or data respectively.[4] Notice that expressions are just a special case of patterns, while patterns can be regarded (at least syntactically) as expressions over an extended set of keys and data that include the special symbols $\circ$ and $\square$. This justifies the use (common throughout this paper) of the same symbols e, e_1, e_2 to denote both expressions and patterns. As a notational convention, we do not write the special key symbol $\circ$ when it occurs as an encryption key. We also assume the paring operation $(\cdot, \cdot)$ is right associative, and omit unnecessary parenthesis. So, for example, we write (e_1, e_2, e_3) and $\{\!|e_1, e_2|\!\}$ instead of $(e_1, (e_2, e_3))$ and $\{\!|(e_1, e_2)|\!\}_\circ$.

[4] To be precise, not all expressions in $\mathbf{Exp}(\mathbf{Keys} \cup \{\circ\}, \mathbf{Data} \cup \{\square\})$ are valid patterns. Formally, the set of patterns is defined as the image $\mathbf{p}(\mathbf{Exp}(\mathbf{Keys}, \mathbf{Data}), \mathcal{P}(\mathbf{Keys}))$ of the function $\mathbf{p}$ given in Figure 2, where $\mathcal{P}(\mathbf{Keys})$ is the power-set of $\mathbf{Keys}$. The reader can safely ignore this technical detail, which is important only when mapping patterns to probability distributions over bit-strings.

$$\begin{aligned}
\mathbf{Keys}(d) &= \emptyset \\
\mathbf{Keys}(k) &= \{k\} \cap \mathbf{Keys} \\
\mathbf{Keys}(e_1, e_2) &= \mathbf{Keys}(e_1) \cup \mathbf{Keys}(e_2) \\
\mathbf{Keys}(\{\!|e|\!\}_k) &= (\{k\} \cap \mathbf{Keys}) \cup \mathbf{Keys}(e)
\end{aligned}$$

$$\begin{aligned}
\mathbf{Parts}(d) &= \{d\} \\
\mathbf{Parts}(k) &= \{k\} \\
\mathbf{Parts}(e_1, e_2) &= \mathbf{Parts}(e_1) \cup \mathbf{Parts}(e_2) \\
\mathbf{Parts}(\{\!|e|\!\}_k) &= \{\{\!|e|\!\}_k\} \cup \mathbf{Parts}(e)
\end{aligned}$$

Fig. 1. The keys and parts of a pattern

The *keys* and *parts* of an expression or pattern are defined in the obvious way according to the rules given in Figure 1. Notice that the special symbol $\circ$ is never included among the keys of a pattern. With this notation, the set of keys $k \in \mathbf{Keys}$ that occur only as encryption subscripts in an expression (but never as messages) is precisely $\mathbf{Keys}(e) \setminus \mathbf{Parts}(e)$. Keys are usually viewed as bound names up to renaming. (E.g., as in the spi calculus [24].) Formally, two expressions or patterns e_1, e_2 are *equivalent up to renaming* (written $e_1 \cong e_2$), if there exists a bijection $\mu\colon \mathbf{Keys}(e_1) \to \mathbf{Keys}(e_2)$ such that $\mu(e_1) = e_2$, where μ acts on e_1 as a substitution. Notice that, by definition, μ only acts on **Keys** and maps the special symbol $\circ$ always to $\circ$.

The symbolic equivalence of cryptographic expressions is defined by means of a pattern function $\mathbf{p}$ (mapping expressions to corresponding patterns) and the auxiliary function $\mathtt{struct}$, both defined in Figure 2. Intuitively, $\mathtt{struct}(e)$

$$\begin{aligned}
\mathbf{p}(d, T) &= d \\
\mathbf{p}(k, T) &= k \\
\mathbf{p}((e_1, e_2), T) &= (\mathbf{p}(e_1, T), \mathbf{p}(e_2, T)) \\
\mathbf{p}(\{\!|e|\!\}_k, T) &= \begin{cases} \{\!|\mathbf{p}(e, T)|\!\}_k & \text{if } k \in T \\ \{\!|\mathtt{struct}(e)|\!\}_k & \text{if } k \notin T \end{cases}
\end{aligned}$$

$$\begin{aligned}
\mathtt{struct}(d) &= \square \\
\mathtt{struct}(k) &= \circ \\
\mathtt{struct}((e_1, e_2)) &= (\mathtt{struct}(e_1), \mathtt{struct}(e_2)) \\
\mathtt{struct}(\{\!|e|\!\}_k) &= \{\!|\mathtt{struct}(e)|\!\}
\end{aligned}$$

Fig. 2. Rules defining the pattern function $\mathbf{p}\colon \mathbf{Pat}(\mathbf{Keys}, \mathbf{Data}) \times \mathcal{P}(\mathbf{Keys}) \to \mathbf{Pat}(\mathbf{Keys}, \mathbf{Data})$ and auxiliary function $\mathtt{struct}\colon \mathbf{Pat}(\mathbf{Keys}, \mathbf{Data}) \to \mathbf{Pat}(\emptyset, \emptyset)$, where $k \in \mathbf{Keys} \cup \{\circ\}$, $d \in D \cup \{\square\}$, and $e, e_1, e_2 \in \mathbf{Pat}(\mathbf{Keys}, \mathbf{Data})$

represents structural information about e (e.g., its size) that may be leaked when encrypting e under standard computational encryption schemes, and $\mathbf{p}(e, T)$ is the pattern observable in e using the keys in T for decryption. Informally, $\mathtt{struct}(e)$ is obtained by replacing all keys and data symbols in e by $\circ$ and $\Box$ respectively, and $\mathbf{p}(e, T)$ is obtained replacing all subexpressions $\{\!|e'|\!\}_k$ in e such that $k \notin T$ by $\{\!|\mathtt{struct}(e')|\!\}_k$. For example,

$$\mathbf{p}((\{\!|d_1|\!\}_{k_1}, \{\!|d_2|\!\}_{k_1}, \{\!|d_1, d_2|\!\}_{k_2}), \{k_1\}) = (\{\!|d_1|\!\}_{k_1}, \{\!|d_2|\!\}_{k_1}, \{\!|\Box, \Box|\!\}_{k_2}).$$

The pattern in this example models the fact that, using the key k_1, an adversary observing the message $(\{\!|d_1|\!\}_{k_1}, \{\!|d_2|\!\}_{k_1}, \{\!|d_1, d_2|\!\}_{k_2})$ can detect that the first two ciphertexts are the encryption of d_1 and d_2 under k_1. The adversary can also determine that the third ciphertext uses a key different from k_1 (e.g., because decryption under k_1 fails), and encodes a message which is about the same size as the concatenation of d_1 and d_2 (e.g., by looking at the length of the ciphertext). However, the adversary cannot extract any other information about the third message. In particular, it cannot detect that the third message is indeed the concatenation of the first two.

Going back to the definition of symbolic equivalence, each expression is mapped to a pattern

$$\mathbf{pattern}(e) = \mathbf{p}(e, \mathbf{recoverable}(e)) \tag{3}$$

where $\mathbf{recoverable}(e) \subseteq \mathbf{Keys}$ is a set (to be defined) which informally consists of all keys that can be "recovered" by an adversary observing e. Two expressions e_1, e_2 are considered symbolically equivalent if $\mathbf{pattern}(e_1) \cong \mathbf{pattern}(e_2)$, i.e., if they have the same pattern up to key renaming.

In most previous work (starting from the original Dolev-Yao paper [1], and including the seminal contribution of Abadi and Rogaway [4]) the set of recoverable keys is defined as

$$\mathbf{recoverable}(e) = \{k : e \vdash k\}$$

where the entailment relation $\vdash$ is the *smallest* binary relation over the set $\mathbf{Exp}(\mathbf{Keys}, \mathbf{Data})$ such that

1. $e \vdash e$ for all $e \in \mathbf{Exp}(\mathbf{Keys}, \mathbf{Data})$,
2. if $e \vdash (e_1, e_2)$ then $e \vdash e_1$ and $e \vdash e_2$, and
3. if $e \vdash \{\!|e_1|\!\}_k$ and $e \vdash k$, then $e \vdash e_1$.

Informally, the entailment relation $\vdash$ represents the capabilities of a Dolev-Yao adversary, that given e, tries to learn as much as possible from e. For example the last rule stipulates that if the adversary can recover both the ciphertext $\{\!|e_1|\!\}_k$ and the key k, then she can decrypt and recover the plaintext e_1 too.

We remark that other definitions of recoverable keys have been considered in the literature. Most notably, in an effort to remove the syntactic restriction to acyclic expressions, Laud [19] has proposed an alternative definition of the entailment relation that strengthens the Dolev-Yao adversary by explicitly allowing him to break the encryption cycles. Formally, Laud defines $\mathbf{recoverable}(e) = \{k : e \vdash_\emptyset k\}$ where the entailment relation $\vdash_S$ is defined as the *smallest* relation satisfying the following conditions

1. $e \vdash_S e$,
2. if $e \vdash_S (e_1, e_2)$ then $e \vdash_S e_1$ and $e \vdash_S e_2$,
3. if $e \vdash_S \{\!\!\{e'\}\!\!\}_k$ then $e \vdash_{S \cup \{k\}} e'$,
4. if $e \vdash_{S \cup \{k\}} e'$ and $e \vdash_S k$ then $e \vdash_S e'$,
5. if $e \vdash_S e'$ and $S \subseteq S'$ then $e \vdash_{S'} e'$,
6. if $e \vdash_{S \cup \{k\}} k$ then $e \vdash_S k$.

Intuitively, the relation $e \vdash_S e'$ models the fact that expression e' can be recovered from expression e using the keys in S for decryption. So, for example, rule 5 simply states that increasing the number of available decryption keys does not decrease our ability to recover information from e. Rules 1 and 2 are the same as for the entailment relation $\vdash$ used by Abadi and Rogaway. Rules 3 and 4 together imply the standard decryption rule: if $e \vdash_S \{e'\}_k$ and $e \vdash_S k$, then $e \vdash_S e'$. The main novelty in Laud's definition is rule 6, which captures the idea that the adversary can break encryption cycles: if decrypting under k allows to recover k, then k is part of an encryption cycle and it can be recovered by the adversary.

3 Computationally Sound Greatest Fixed Point Semantics

In order to compare our results to prior work, it is convenient to give a different, but equivalent definition of the set of recoverable keys. First of all, we extend the pattern computation function $\mathbf{p}$ of Abadi and Rogaway [4] to include patterns in its domain. This is done in the obvious way, namely, we let

$$\mathbf{p} \colon \mathbf{Pat}(\mathbf{Keys}, \mathbf{Data}) \times \mathcal{P}(\mathbf{Keys}) \to \mathbf{Pat}(\mathbf{Keys}, \mathbf{Data})$$

be the function defined precisely by the same rules already given in Figure 2. Next, we introduce a key recovery function

$$\mathbf{r} \colon \mathbf{Pat}(\mathbf{Keys}, \mathbf{Data}) \to \mathcal{P}(\mathbf{Keys})$$

which is, in a sense, a counterpart to the pattern computation function $\mathbf{p}$ of [4]. Intuitively, the function $\mathbf{r}$ maps the expression or pattern e to the set of keys recoverable from *all parts* of e. For the class of patterns used in this paper, the function $\mathbf{r}$ can be simply defined as

$$\mathbf{r}(e) = \{k \in \mathbf{Keys}(e) \colon k \in \mathbf{Parts}(e)\} = \mathbf{Keys}(e) \cap \mathbf{Parts}(e), \tag{4}$$

i.e., $\mathbf{r}(e)$ is the set of all keys that appear in e as a message. In other words, $\mathbf{r}(e)$ includes all keys of e, except those that occur exclusively as encryption subscripts.

We observe that the functions $\mathbf{p}$ and $\mathbf{r}$ satisfy the following fundamental properties:

$$\mathbf{p}(e, \mathbf{Keys}) = e \tag{5}$$

$$\mathbf{p}(\mathbf{p}(e, S), T) = \mathbf{p}(e, S \cap T) \tag{6}$$

$$\mathbf{r}(\mathbf{p}(e, T)) \subseteq \mathbf{r}(e) \tag{7}$$

These are all very natural requirements. Properties (5) and (6) just say that $\mathbf{p}$ makes keys *act* on the patterns, or, more precisely, $(\mathcal{P}(\mathbf{Keys}), \cap)$ acts[5] as a monoid on the set $\mathbf{Pat}(\mathbf{Keys}, \mathbf{Data})$. The third property (7) states that the action $\mathbf{p}(\cdot, T)$ does not increase the amount of information recoverable from (the parts of) a pattern. When $\mathbf{p}$ and $\mathbf{r}$ satisfy properties 5-7, we say that "$\mathbf{p}$ is an $\mathbf{r}$-projection". We will see later that these are the only properties needed to instantiate our general framework, but for now the reader may want to focus on the specific functions $\mathbf{p}$ and $\mathbf{r}$ defined in Figure 2 and (4).

The functions $\mathbf{p}, \mathbf{r}$ are used to associate to each $e \in \mathbf{Exp}(\mathbf{Keys}, \mathbf{Data})$ a corresponding key recovery operator

$$\mathcal{F}_e \colon T \mapsto \mathbf{r}(\mathbf{p}(e, T)) \tag{8}$$

that maps any $T \subseteq \mathbf{Keys}$ to the set of keys recoverable from all the parts of the observable pattern $\mathbf{p}(e, T)$. The function $\mathcal{F}_e$ models the process of using a set of keys T to break an expression e into parts, and then using all such parts to recover as many keys as possible. In Theorem 1 we will show that for any expression e, the key recovery operator (8) is a monotone function. In particular, $\mathcal{F}_e$ admits both a least and a greatest fixed point

$$\mathrm{fix}(\mathcal{F}_e) = \bigcup_n \mathcal{F}^n(\emptyset) \qquad \mathrm{FIX}(\mathcal{F}_e) = \bigcap_n \mathcal{F}^n(\mathbf{Keys}).$$

It is a well known fact that the set of keys $\{k \colon e \vdash k\}$ recoverable by a Dolev-Yao adversary is precisely the least fixed point of $\mathcal{F}_e$. So, the Abadi-Rogaway definition of the pattern of an expression can be reformulated as $\mathbf{pattern}(e) = \mathbf{p}(e, \mathrm{fix}(\mathcal{F}_e))$.

Our general framework is very similar to the one of Abadi and Rogaway, and we adopt most definitions given so far. The only difference is that, instead of defining recoverable keys as the least fixed point of $\mathcal{F}_e$, we take the greatest fixed point and let

$$\mathbf{Pattern}(e) = \mathbf{p}(e, \mathrm{FIX}(\mathcal{F}_e)). \tag{9}$$

As usual, two expressions are symbolically equivalent if they have the same pattern (9) up to key renaming. We refer the reader to Section 4 for an example of use of the greatest fixed point patterns.

In this section we prove that our new greatest fixed point symbolic semantics is computationally sound, i.e., for any two expressions e_1, e_2, if $\mathbf{Pattern}(e_1) \cong \mathbf{Pattern}(e_2)$, then the probability distributions $[\![e_1]\!]$ and $[\![e_2]\!]$ are computationally indistinguishable. We do so in a very general way, applicable to a wider class of cryptographic expressions than considered in this paper and in [4], as demonstrated in follow-up work [15]. Theorem 1 below states that, as long as properties (5-7) are satisfied, in order to establish the computational soundness of the greatest fixed point symbolic semantics (9) it is enough to test the following simpler condition: for any pattern e, the probability distributions $[\![e]\!]$, and $[\![\mathbf{p}(e, \mathbf{r}(e))]\!]$

[5] Recall that an action of a monoid $(G, \cdot)$ on a set A is a binary operation $\times$ mapping $A \times G$ to A such that $(a \times g_1) \times g_2 = a \times (g_1 \cdot g_2)$ and $a \times 1_G = a$.

are computationally indistinguishable. Informally, this condition states that the keys $\mathbf{r}(e)$ recoverable from all parts of a pattern do not increase our knowledge about the pattern. This is a non-trivial assumption, as it depends on the security of the encryption scheme, but still it is a much easier-to-check condition than the conclusion of the soundness theorem. In particular, the indistinguishability of $[\![\mathbf{p}(e,\mathbf{r}(e))]\!]$ and $[\![e]\!]$ can be usually proved in a fairly direct way, starting from the definition of secure encryption scheme, without the need to go through a complex hybrid argument.

Theorem 1. *Let* **Keys** *and* **Data** *be two (disjoint) sets of key and constant symbols, and let* $\mathbf{p}$ *and* $\mathbf{r}$ *be functions such that* $\mathbf{p}$ *is an* $\mathbf{r}$*-projection. Then, for any expression* $e \in \mathbf{Exp}(\mathbf{Keys},\mathbf{Data})$*, the key recovery operator* $\mathcal{F}_e(T) = \mathbf{r}(\mathbf{p}(e,T))$ *is a monotone function, and the greatest fixed point semantics* $\mathbf{Pattern}(e) = \mathbf{p}(e, \mathit{FIX}(\mathcal{F}_e))$ *is well defined. Moreover, if, for any* $e \in \mathbf{Pat}(\mathbf{Keys},\mathbf{Data})$*, the distributions* $[\![e]\!]$ *and* $[\![\mathbf{p}(e,\mathbf{r}(e))]\!]$ *are computationally indistinguishable, the distribution* $[\![e]\!]$ *is computationally indistinguishable from* $\mathbf{Pattern}(e)$*. In particular, for any two expressions* $e_1, e_2 \in \mathbf{Exp}(\mathbf{Keys},\mathbf{Data})$*, if* $\mathbf{Pattern}(e_1) \cong \mathbf{Pattern}(e_2)$*, then the distributions* $[\![e_1]\!]$ *and* $[\![e_2]\!]$ *are computationally indistinguishable.*

Proof. First of all, we show that the key recovery operator is monotone. Let $S \subseteq T \subseteq \mathbf{Keys}$ be two sets of keys. From the definition of $\mathcal{F}_e$ and properties (6–7), we obtain

$$\begin{aligned}\mathcal{F}_e(S) &= \mathbf{r}(\mathbf{p}(e,S))\\ &= \mathbf{r}(\mathbf{p}(e,T\cap S))\\ &= \mathbf{r}(\mathbf{p}(\mathbf{p}(e,T),S))\\ &\subseteq \mathbf{r}(\mathbf{p}(e,T)) = \mathcal{F}_e(T).\end{aligned}$$

So, $\mathcal{F}_e$ is a monotone operator and it admits a greatest fixed point $\mathrm{FIX}(\mathcal{F}_e) = \bigcap_i \mathcal{F}^i(\mathbf{Keys})$.

Now consider an expression e and the corresponding pattern $\mathbf{pattern}(e) = \mathbf{p}(e, \mathrm{FIX}(\mathcal{F}_e))$, and assume without loss of generality that $\mathbf{Keys} = \mathbf{Keys}(e)$, so that $n = |\mathbf{Keys}|$ is polynomially bounded in the size of e. Since $\mathcal{F}_e$ is a monotone function, we have $\mathrm{FIX}(\mathcal{F}_e) = \mathcal{F}_e^n(\mathbf{Keys})$, where $n = |\mathbf{Keys}|$ is the length of the longest chain in $\mathcal{P}(\mathbf{Keys})$. We will show that for every i, $[\![\mathbf{p}(e,\mathcal{F}_e^{i+1}(\mathbf{Keys}))]\!]$ is computationally indistinguishable from $[\![\mathbf{p}(e,\mathcal{F}_e^{i}(\mathbf{Keys}))]\!]$. It follows, by transitivity, that $[\![\mathbf{p}(e,\mathrm{FIX}(\mathcal{F}_e))]\!] = [\![\mathbf{p}(e,\mathcal{F}_e^{n}(\mathbf{Keys}))]\!]$ is computationally indistinguishable from $[\![e]\!] = [\![\mathbf{p}(e,\mathbf{Keys})]\!] = [\![\mathbf{p}(e,\mathcal{F}_e^{0}(\mathbf{Keys}))]\!]$. More specifically, any probabilistic polynomial time algorithm distinguishing $[\![e]\!]$ from $[\![\mathbf{pattern}(e)]\!]$ with advantage δ can be turned into a probabilistic polynomial time algorithm that distinguishes $[\![\mathbf{p}(e,\mathcal{F}_e^{i+1}(\mathbf{Keys}))]\!]$ from $[\![\mathbf{p}(e,\mathcal{F}_e^{i}(\mathbf{Keys}))]\!]$ for some i with advantage δ/n.

Fix the value of the index i, and let $T = \mathcal{F}_e^i(\mathbf{Keys})$ and $e' = \mathbf{p}(e,T)$. Clearly, $\mathcal{F}_e(\mathbf{Keys}) \subseteq \mathbf{Keys}$ because $\mathbf{Keys}$ is the set of all keys in e, and from the monotonicity of $\mathcal{F}_e$ we get that $\mathcal{F}_e^{i+1}(\mathbf{Keys}) \subseteq \mathcal{F}_e^{i}(\mathbf{Keys})$ for all $i \geq 0$. In particular, $\mathcal{F}_e(T) \subseteq T$. We want to prove that $[\![\mathbf{p}(e, \mathcal{F}_e(T))]\!]$ is indistinguishable from $[\![\mathbf{p}(e, T)]\!]$. Notice that, using the definition of $\mathcal{F}_e(T) = \mathbf{r}(\mathbf{p}(e, T))$, we get

$$\begin{aligned}\mathbf{p}(e', \mathbf{r}(e')) &= \mathbf{p}(\mathbf{p}(e, T), \mathcal{F}_e(T))\\ &= \mathbf{p}(e, T \cap \mathcal{F}_e(T))\\ &= \mathbf{p}(e, \mathcal{F}_e(T)).\end{aligned}$$

Remember that by hypothesis, $[\![\mathbf{p}(e', \mathbf{r}(e'))]\!]$ is computationally indistinguishable from $[\![e']\!]$. Therefore, $[\![\mathbf{p}(e, \mathcal{F}_e(T))]\!] = [\![\mathbf{p}(e', \mathbf{r}(e'))]\!]$ is indistinguishable from $[\![e']\!] = [\![\mathbf{p}(e, T)]\!]$ as claimed.

We remark that in Theorem 1 we have assumed that e is an expression for simplicity only. The same result (and proof) holds true also when $e \in \mathbf{Pat}(\mathbf{Keys}, \mathbf{Data})$ is an arbitrary pattern. In the following corollary we apply Theorem 1 to the functions $\mathbf{p}$ and $\mathbf{r}$ defined in Figure 2 and (4).

Corollary 1. *If $\mathcal{E}$ is a (length regular) semantically secure encryption scheme, then for any two expressions $e_1, e_2 \in \mathbf{Exp}(\mathbf{Keys}, \mathbf{Data})$ such that $\mathbf{Pattern}(e_1) \cong \mathbf{Pattern}(e_2)$, the distributions $[\![e_1]\!]$ and $[\![e_2]\!]$ are computationally indistinguishable.*

Proof. We already observed that $\mathbf{p}$ and $\mathbf{r}$ satisfy properties (5–7). In order to apply Theorem 1 and conclude that $[\![e_1]\!]$ is indistinguishable from $[\![e_2]\!]$, we only need to prove that for any pattern e, the distributions $[\![\mathbf{p}(e, \mathbf{r}(e))]\!]$ and $[\![e]\!]$ and computationally indistinguishable. To this end, assume for contradiction that there exists an efficient algorithm $\mathcal{D}$ that distinguishes distribution $[\![e]\!]$ from $[\![\mathbf{p}(e, \mathbf{r}(e))]\!]$ with non-negligible probability. (See Definition 2 in Appendix.) We use $\mathcal{D}$ to construct an efficient adversary that breaks the indistinguishability of the encryption scheme $\mathcal{E}$ used to evaluate the patterns. Let $T = \mathbf{Keys}(e) \setminus \mathbf{Parts}(e)$ be the set of all encryption keys in e that do not also appear in e as a message. We define an adversary $\mathcal{A}$ that is given access to $|T|$ encryption oracles $\mathcal{E}_b^t(\cdot, \cdot)$ (indexed by $t \in T$). The adversary $\mathcal{A}$ chooses keys $\sigma(k)$ independently at random for all $k \in \mathbf{Keys}(e) \setminus T = \mathbf{r}(e)$. It then evaluates the expression e according to the usual evaluation rules, except for subexpressions of the form $\{\!|e'|\!\}_k$ where $k \in T$. These are evaluated using oracle $\mathcal{E}_b^k$. When $\mathcal{A}$ is done evaluating e, it submits the resulting string to the distinguisher $\mathcal{D}$. Notice that when $b = 1$, the adversary $\mathcal{A}$ produces a query which is distributed identically to $[\![e]\!]$, while when $b = 0$ the distribution is $[\![\mathbf{p}(e, \mathbf{r}(e))]\!]$. So, $\mathcal{A}$ will have the same advantage in breaking the encryption scheme as $\mathcal{D}$ has in distinguishing $[\![e]\!]$ from $[\![\mathbf{p}(e, \mathbf{r}(e))]\!]$.

Corollary 1 is very similar in spirit to the soundness result proved by Abadi and Rogaway in [4]. However, our proof of Corollary 1 is much simpler than the original argument given by Abadi and Rogaway, which requires the expressions e_1, e_2 to be acyclic. The main difference is our use of greatest fixed points in the definition of adversarial knowledge, while [4] uses the traditional least fixed point definition. At first sight, the two results may seem incomparable, since they use different definitions of patterns. The following theorem bridges the gap between the two (inductive and co-inductive) definitions of pattern, showing that acyclic expressions have a unique fixed point. So, under the acyclicity hypothesis of [4] (common to most other work on computationally sound symbolic cryptography) the traditional least fixed point semantics and the new greatest fixed point semantics are identical.

Theorem 2. *If $e \in \mathbf{Exp}(\mathbf{Keys}, \mathbf{Data})$ is an acyclic expression, then $\mathit{fix}(\mathcal{F}_e) = \mathit{FIX}(\mathcal{F}_e)$.*

Proof. Assume $\mathrm{fix}(\mathcal{F}_e) \neq \mathrm{FIX}(\mathcal{F}_e)$. We prove that e contains an encryption cycle. Since $\mathrm{fix}(\mathcal{F}_e) \subset \mathrm{FIX}(\mathcal{F}_e)$, the set $T = \mathrm{FIX}(\mathcal{F}_e) \setminus \mathrm{fix}(\mathcal{F}_e)$ is not empty. Notice that all $k \in T$ necessarily belong to $\mathbf{r}(e)$ because by monotonicity

$$T \subseteq \mathrm{FIX}(\mathcal{F}_e) = \mathcal{F}_e(\mathrm{FIX}(\mathcal{F}_e)) \subseteq \mathcal{F}_e(\mathbf{Keys}(e)) = \mathbf{r}(\mathbf{p}(e, \mathbf{Keys}(e))) = \mathbf{r}(e). \tag{10}$$

However, all occurrences of $k \in T$ in e must be under the scope of an encryption operator $\{\!\!\{ \dots k \dots \}\!\!\}_{k'}$ with $k' \notin \mathrm{fix}(\mathcal{F}_e)$, because $k \notin \mathrm{fix}(\mathcal{F}_e)$. Again, from (10), we get that at least some occurrence of k in e must not be encrypted under keys outside of $\mathrm{FIX}(\mathcal{F}_e)$. It follows, that k must be encrypted under some key $k \in \mathrm{FIX}(\mathcal{F}_e) \backslash \mathrm{fix}(\mathcal{F}_e) = T$. Consider now the "encrypt" relation, restricted to the keys in T: for any $k_1, k_2 \in T$, k_1 encrypts k_2 (in e) if e contains a subexpression $\{\!\!\{e'\}\!\!\}_{k_1}$ such that $k_2 \in \mathbf{Parts}(e')$. We just proved that all keys in T are encrypted in e under some key in T, i.e., all nodes T in the graph of the "encrypt" relation, have in-degree at least one. Since T is a non-empty finite set, it must necessarily contain a cycle.

4 Example

In this section we illustrate our greatest fixed point symbolic framework on a simple example expression. Let

$$e = (\{\!\!\{k_1, \{\!\!\{\{\!\!\{k_4\}\!\!\}_{k_3}\}\!\!\}_{k_4}\}\!\!\}_{k_2}, \{\!\!\{k_2\}\!\!\}_{k_1}).$$

The set of recoverable keys associated to this expression is defined as the greatest fixed point of the key recovery operator $\mathcal{F}_e$. This fixed point is computed as follows. Start from the set $K_0 = \{k_1, k_2, k_3, k_4\}$ of all keys in the expression, and apply $\mathcal{F}_e$ to it to obtain the set

$$\begin{aligned} K_1 &= \mathcal{F}_e(K_0) \\ &= \mathbf{r}(\mathbf{p}(e, K_0)) \\ &= \mathbf{r}(\{\!\!\{(k_1, \{\!\!\{\{\!\!\{k_4\}\!\!\}_{k_3}\}\!\!\}_{k_4})\}\!\!\}_{k_2}, \{\!\!\{k_2\}\!\!\}_{k_1}) \\ &= \{k_1, k_2, k_4\}. \end{aligned}$$

As we apply $\mathcal{F}_e$ to K_1 we obtain

$$\begin{aligned} K_2 &= \mathcal{F}_e(K_1) \\ &= \mathbf{r}(\mathbf{p}(e, K_1)) \\ &= \mathbf{r}(\{\!\!\{(k_1, \{\!\!\{\{\!\!\{\circ\}\!\!\}_{k_3}\}\!\!\}_{k_4})\}\!\!\}_{k_2}, \{\!\!\{k_2\}\!\!\}_{k_1}) \\ &= \{k_1, k_2\}. \end{aligned}$$

If we apply $\mathcal{F}_e$ once more we obtain

$$\begin{aligned} K_3 &= \mathcal{F}_e(K_2) \\ &= \mathbf{r}(\mathbf{p}(e, K_2)) \\ &= \mathbf{r}(\{\!|(k_1, \{\!|\{\!|\circ|\!\}|\!\}_{k_4})|\!\}_{k_2}, \{\!|k_2|\!\}_{k_1}) \\ &= \{k_1, k_2\}. \end{aligned}$$

Notice that we obtained a decreasing sequence of sets

$$\begin{aligned} \mathcal{F}_e^0(\mathbf{Keys}) &= \{k_1, k_2, k_3, k_4\} \\ &\supset \mathcal{F}_e^1(\mathbf{Keys}) \\ &= \{k_1, k_2, k_4\} \\ &\supset \mathcal{F}_e^2(\mathbf{Keys}) = \{k_1, k_2\} \\ &= \mathcal{F}_e^3(\mathbf{Keys}) \end{aligned}$$

and $\mathcal{F}_e^i(\mathbf{Keys}) = \{k_1, k_2\}$ for all $i \geq 2$. This is the greatest fixed point of the operator $\mathcal{F}_e$, so the symbolic semantics of expression e is

$$\mathbf{Pattern}(e) = \mathbf{p}(e, \{k_1, k_2\}) = \{\!|(k_1, \{\!|\{\!|\circ|\!\}|\!\}_{k_4})|\!\}_{k_2}, \{\!|k_2|\!\}_{k_1}).$$

This pattern tells us that the keys k_1 and k_2 are not guaranteed to be hidden from an adversary when a computational encryption scheme (satisfying the standard notion of indistinguishability against chosen plaintext attack) is used. On the other hand, the adversary cannot recover they keys k_3 and k_4, even if k_4 is part of an encryption cycle.

5 Discussion and Open Problems

We presented a general framework for the computationally sound symbolic analysis of cryptographic expressions, as those used to model messages in security protocols. The framework is essentially the same as the standard one proposed by Abadi and Rogaway [4], with the only difference that the adversarial knowledge is defined by co-induction (using greatest fixed points), rather than induction (using least fixed points). This simple change brings the computational and symbolic definitions much closer to each other.

We believe that our observations improve our understanding of the relation between symbolic and computational cryptography, and open up several new interesting research directions. In retrospect, the fact that co-inductive methods (in the symbolic setting) result in a closer connection to computational security should not come too much as a surprise, since the methods of computational cryptography (e.g., the notion of computational indistinguishability, a form of observational equivalence) have a very strong co-inductive flavor. Acyclicity and similar syntactic restrictions are not a peculiarity of [4]: most work on computationally sound symbolic security analysis (with just a few rare exceptions like [19]) seem to require restrictions of this sort. Our results suggest the use of co-induction in the symbolic modeling of adversarial knowledge as a general method to prove closer connections between symbolic and computational security in other settings. There is a need for more work in the area of co-inductive symbolic security analysis, and such work is likely to provide a better bridge between symbolic and computational cryptography than traditional methods based on induction.

It is natural to ask how co-induction relates to recent constructions of circularly secure encryption [8,25], i.e., computational encryption schemes that remain secure even in the presence of encryption cycles. We remark that [8,25] achieve circular security by building encryption schemes satisfying very strong homomorphic properties that allow, for example, to build the encryption of k under k (i.e., an encryption cycle of length 1) given the encryption of 0 under k, and similarly for longer cycles. We conjecture that if the Dolev-Yao deduction rules are properly modified to model encryption schemes with special homomorphic properties (as those used in [8,25]), then the resulting key recovery operator $\mathcal{F}_e$ associated to any expression (with or without encryption cycles) would always have a unique fixed point. We leave a full investigation of computational soundness of encryption schemes with special properties to future work.

The generality of our approach (at least in the setting of secrecy properties in the presence of passive adversaries) has recently been demonstrated in [15], where Theorem 1 is used to establish the computational soundness of symbolic expressions with pseudorandom keys, as those employed in multicast key distribution protocols [11,12,13]. As in this paper, the result of [15] does not require the expressions to be acyclic or satisfy any syntactic restriction. We expect similar results can also be obtained for cryptographic expressions that make use of secret sharing schemes (as those employed in [14] in the analysis of cryptographically controlled access to XML documents), and most other cryptographic primitives achieving secrecy goals.

The main open problem at this point is to extend our co-inductive framework to prove computational soundness results in the presence of active adversaries, as those considered in [26]. We remark that moving from passive adversaries to active attacks requires substantial changes in the execution model. In a passive attack, an adversary only gets to see the sequence of messages transmitted during the execution of the protocol. So the entire adversary's view of the system can be modeled by a sequence of expressions (or even a single expression containing their concatenation.) In an active attack scenario, the adversary interacts with the honest parties, intercepting and injecting messages in the communication network. Security properties no longer pertain exclusively what information can be learned by the adversary, but also how the adversary can influence the messages. A general approach to computational soundness in the presence of active adversaries has been proposed in [26], where security properties are modeled as sets of traces, e.g., sequences of events that can occur during a run of the protocol. We leave the development of a co-inductive framework for the study of cryptographic trace properties in the presence of active attacks as an open problem.

Acknowledgments. The author thanks the anonymous referees for their helpful comments. This research was supported in part by NSF under grants CNS-0430595 and CNS-0831536. 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 National Science Foundation.

References

1. Dolev, D., Yao, A.: On the security of public key protocols. IEEE Transactions on Information Theory 29(2), 198–208 (1983)
2. Mitchell, J.C., Mitchell, M., Stern, U.: Automated analysis of cryptographic protocols using Murphi. In: Proceedings of SSP 1997, pp. 141–151. IEEE Computer Society, Los Alamitos (1997)
3. Paulson, L.C.: The inductive approach to verifying cryptographic protocols. Journal of Computer Security 6(1-2), 85–128 (1998)
4. Abadi, M., Rogaway, P.: Reconciling two views of cryptography (The computational soundness of formal encryption). Journal of Cryptology 15(2), 103–127 (2002)
5. Camenish, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer, Heidelberg (2001)
6. 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)
7. Halevi, S., Krawczyk, H.: Security under key-dependent inputs. In: Computer and communications security – Proceedings of CCS 2007, Alexandria, VA, USA, pp. 466–475. ACM, New York (2007)
8. 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)
9. Adão, P., Bana, G., Scedrov, A.: Computational and information theoretic soundness and completeness of formal encryption. In: Proceedings of CSFW 2005, June 2005, pp. 170–184. IEEE Computer Society, Los Alamitos (2005)
10. 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)
11. Micciancio, D., Panjwani, S.: Corrupting one vs. corrupting many: the case of broadcast and multicast encryption. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 70–82. Springer, Heidelberg (2006)
12. Micciancio, D., Panjwani, S.: Adaptive security of symbolic encryption. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 169–187. Springer, Heidelberg (2005)
13. Micciancio, D., Panjwani, S.: Optimal communication complexity of generic multicast key distribution. IEEE/ACM Transactions on Networking 16(4), 803–813 (2008); Preliminary version in Eurocrypt 2004
14. Abadi, M., Warinschi, B.: Security analysis of cryptographycally controlled access to XML documents. Journal of the ACM 55(2), 1–29 (2008); Prelim. version in PODS 2005

15. Micciancio, D.: Pseudo-randomness and partial information in symbolic security analysis. Report 2009/249, IACR ePrint archive (2009), http://eprint.iacr.org/2009/249
16. Abadi, M., Jürjens, J.: Formal eavesdropping and its computational interpretation. In: Kobayashi, N., Pierce, B. (eds.) TACS 2001. LNCS, vol. 2215, pp. 82–94. Springer, Heidelberg (2001)
17. Mitchell, J.C., Ramanathan, A., Scedrov, A., Teague, V.: A probabilistic polynomial-time calculus for the analysis of cryptographic protocols. Theoretical Computer Science 353(1-3), 118–164 (2006); Preliminary version in MFPS 2001
18. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sience 28(2), 270–299 (1984); Preliminary version in Proc. of STOC 1982
19. Laud, P.: Encryption cycles and two views of cryptography. In: Proceedings of NORDSEC 2002, Karlstad University Studies, Karlstad, Sweden, November 2002, vol. 31, pp. 85–100 (2002)
20. Adão, P., Bana, G., Herzog, J., Scedrov, A.: Soundness of formal encryption in the presence of key-cycles. In: di Vimercati, S.d.C., Syverson, P.F., Gollmann, D. (eds.) ESORICS 2005. LNCS, vol. 3679, pp. 374–396. Springer, Heidelberg (2005)
21. 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)
22. Backes, M., Pfitzmann, B., Scedrov, A.: Key-dependent message security under active attacks - BRSIM/UC-soundness of Dolev-Yao-style encryption with key cycles. Journal of Computer Security 16(5), 497–530 (2008); Preliminary version in CSF 2007
23. Laud, P., Vene, V.: A type system for computationally secure information flow. In: Liśkiewicz, M., Reischuk, R. (eds.) FCT 2005. LNCS, vol. 3623, pp. 365–377. Springer, Heidelberg (2005)
24. Abadi, M., Gordon, A.: A calculus for cryptographic protocols: the spi calculus. In: Proceedings of CCS 1997, pp. 36–47 (1997)
25. 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)
26. Micciancio, D., Warinschi, B.: Soundness of formal encryption in the presence of active adversaries. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 133–151. Springer, Heidelberg (2004)
27. Goldreich, O.: Foundations of Cryptography. Basic Tools, vol. I. Cambridge University Press, Cambridge (2001)
28. Goldreich, O.: Foundation of Cryptography. Basic Applications, vol. II. Cambridge University Press, Cambridge (2004)

Appendix

In the computational setting, given an encryption scheme $\mathcal{E}$, each expression $e \in \mathbf{Exp}(\mathbf{Keys}, \mathbf{Data})$ naturally maps to a probability distribution $[\![e]\!]$ over bitstrings. Two expressions e_1, e_2 are equivalent in the computational setting if the corresponding probability distributions $[\![e_1]\!] \equiv [\![e_2]\!]$ are computationally indistinguishable. In this appendix we briefly recall all the basic computational security definitions used in this paper. The reader is referred to any standard textbook (e.g., [27,28]) for details.

Encryption. A (symmetric) encryption scheme is defined as a pair of (probabilistic) polynomial time encryption and decryption algorithms $\mathcal{E}, \mathcal{D}$ such that $\mathcal{D}(k, \mathcal{E}(k, m)) = m$ for any message m and key k. Here the message m is an arbitrary string, and the key k is a uniformly random string of some fixed length ℓ that depends on the desired security level. The encryption scheme is considered secure if it satisfies the following property, called semantic security or indistinguishability under chosen plaintext attack.

Definition 1. *An encryption scheme $(\mathcal{E}, \mathcal{D})$ is indistinguishable under chosen plaintext attack if, for any probabilistic polynomial time adversary $\mathcal{A}$, the following holds. Choose a bit b and a key k of length ℓ uniformly at random and run $\mathcal{A}$ on input ℓ and with access to an encryption oracle $O_b(m)$ that outputs $\mathcal{E}(k, m)$ if $b = 1$, or $\mathcal{E}(k, 0^{|m|})$ if $b = 0$. The attacker $\mathcal{A}$ is required to run in time polynomial in the security parameter ℓ, and is supposed to guess the value of b. Then the quantity $|\Pr\{\mathcal{A}^{O_1}(\ell) = 1\} - \Pr\{\mathcal{A}^{O_0}(\ell) = 1\}|$ is negligible in the security parameter ℓ, i.e., it is smaller than $1/\ell^c$ for any constant c and sufficiently large ℓ.*

The above definition can be proved equivalent (via a standard hybrid argument) to a seemingly stronger definition where the attacker is given access to several encryption oracles, each encrypting under an independently chosen random key.

Definition 2. *An encryption scheme $(\mathcal{E}, \mathcal{D})$ is indistinguishable under chosen plaintext attack if, for any probabilistic polynomial time adversary $\mathcal{A}$ and polynomial p, the following holds. Choose a bit b and $n = p(\ell)$ keys $k_1, \ldots, k_n$ of length ℓ each, uniformly and independently at random and run $\mathcal{A}$ on input ℓ and with access to an encryption oracle $O_b(i, m)$ that outputs $\mathcal{E}(k_i, m)$ if $b = 1$, or $\mathcal{E}(k_i, 0^{|m|})$ if $b = 0$. The attacker $\mathcal{A}$ is required to run in time polynomial in the security parameter ℓ, and is supposed to guess the value of b. Then the quantity $|\Pr\{\mathcal{A}^{O_1}(\ell) = 1\} - \Pr\{\mathcal{A}^{O_0}(\ell) = 1\}|$ is negligible in the security parameter ℓ, i.e., it is smaller than $1/\ell^c$ for any constant c and sufficiently large ℓ.*

Computational equivalence between probability distributions over bitstrings is defined below.

Definition 3. *Let $\{A_i^0\}$ and $\{A_i^1\}$ be two probability ensembles, i.e., two sequences of probability distributions over bitstrings. $\{A_i^0\}$ and $\{A_i^1\}$ are computationally indistinguishable if for any probabilistic polynomial time adversary D, the quantity $|\Pr\{D(A_i^0) = 1\} - \Pr\{D(A_i^1) = 1\}|$ is negligible in i.*

Computational evaluation. Cryptographic expressions can be evaluated using a computational encryption scheme $\mathcal{E}$. In order to map the expressions to strings we need also to fix a string value γ_d for every piece of data $d \in \mathbf{Data}$ appearing in the expression, and a pairing function $\gamma \colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$.

We first define the evaluation $\sigma[\![e]\!]$ of an expression $e \in \mathbf{Exp}(\mathbf{Keys}, \mathbf{Data})$ with respect to a fixed key assignment $\sigma \colon \mathbf{Keys} \to \{0,1\}^\ell$. The value $\sigma[\![e]\!]$ is defined by induction on the structure of the expression e by the rules $\sigma[\![d]\!] = \gamma_d$,

$\sigma[\![k]\!] = \sigma(k)$, $\sigma[\![(e_1, e_2)]\!] = \gamma(\sigma[\![e_1]\!], \sigma[\![e_2]\!])$, and $\sigma[\![\{\!|e|\!\}_k]\!] = \mathcal{E}(\sigma(k), \sigma[\![e]\!])$ where all applications of the encryption algorithm $\mathcal{E}$ are performed using independent randomness. The computational evaluation $[\![e]\!]$ of an expression e is defined as the probability distribution obtained by first choosing a random key assignment σ (by setting $\sigma(k) \in \{0,1\}^\ell$ to an independently and randomly chosen value for each key symbol $k \in \mathbf{Keys}$) and then computing $\sigma[\![e]\!]$.

Length conventions and pattern evaluation. Since computational encryption is not usually required to hide the length of the input, it is natural to require that all functions operating on messages are length-regular, i.e., the length of the output depends only on the length of the input. Throughout the paper we assume that the functions $d \mapsto \gamma_d$, $\gamma(\cdot,\cdot)$ and $\mathcal{E}$ are length regular, i.e., $|\gamma_d|$ is the same for all $d \in \mathbf{Data}$, $|\sigma(k)| = \ell$ for all keys k, $|\gamma(x_1, x_2)|$ depends only on $|x_1|$ and $|x_2|$, and $|\mathcal{E}(k, x)|$ depends only on $|\sigma(k)| = \ell$ and $|x|$. Under these assumptions, it is easy to see that any two expressions $e, e' \in \mathbf{Exp}(\mathbf{Keys}, \mathbf{Data})$ with the same structure $\texttt{struct}(e) = \texttt{struct}(e')$ are always evaluated to strings of exactly the same length $|\sigma[\![e]\!]| = |\sigma[\![e']\!]|$. Using this fact, the computational evaluation function $\sigma[\![e]\!]$ is extended to patterns by defining $\sigma[\![\texttt{struct}(e)]\!] = 0^{|\sigma[\![e]\!]|}$. Notice that the definition is well given because $|\sigma[\![e]\!]|$ depends only on $\texttt{struct}(e)$, and not on the specific expression e.

Encryption Schemes Secure against Chosen-Ciphertext Selective Opening Attacks

Serge Fehr[1], Dennis Hofheinz[2,⋆], Eike Kiltz[1,⋆⋆], and Hoeteck Wee[3,⋆⋆⋆]

[1] CWI, Amsterdam
[2] Karlsruhe Institute of Technology
[3] Queens College, CUNY

Abstract. Imagine many small devices send data to a single receiver, encrypted using the receiver's public key. Assume an adversary that has the power to adaptively corrupt a subset of these devices. Given the information obtained from these corruptions, do the ciphertexts from *uncorrupted* devices remain secure?

Recent results suggest that conventional security notions for encryption schemes (like IND-CCA security) do not suffice in this setting. To fill this gap, the notion of *security against selective-opening attacks (SOA security)* has been introduced. It has been shown that lossy encryption implies SOA security against a *passive*, i.e., only eavesdropping and corrupting, adversary (SO-CPA). However, the known results on SOA security against an *active* adversary (SO-CCA) are rather limited. Namely, while there exist feasibility results, the (time and space) complexity of currently known SO-CCA secure schemes depends on the number of devices in the setting above.

In this contribution, we devise a new solution to the selective opening problem that does not build on lossy encryption. Instead, we combine techniques from non-committing encryption and hash proof systems with a new technique (dubbed "cross-authentication codes") to glue several ciphertext parts together. The result is a rather practical SO-CCA secure public-key encryption scheme that does not suffer from the efficiency drawbacks of known schemes. Since we build upon hash proof systems, our scheme can be instantiated using standard number-theoretic assumptions such as decisional Diffie-Hellman (DDH), decisional composite residuosity (DCR), and quadratic residuosity (QR). Besides, we construct a conceptually very simple and comparatively efficient SO-CPA secure scheme from (slightly enhanced) trapdoor one-way permutations.

We stress that our schemes are completely independent of the number of challenge ciphertexts, and we do not make assumptions about the underlying message distribution (beyond being efficiently samplable). In particular, we do not assume efficient conditional re-samplability of the message distribution. Hence, our schemes are secure in *arbitrary* settings, even if it is not known in advance how many ciphertexts might be considered for corruptions.

[⋆] Part of this work performed while at CWI. Supported by an NWO Veni grant.

[⋆⋆] Supported by the research program Sentinels (http://www.sentinels.nl).

[⋆⋆⋆] Supported by PSC-CUNY Award # 6014939 40 and the US Army Research laboratory and the UK Ministry of Defence under agreement number W911NF-06-3-0001.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 381–402, 2010.

1 Introduction

The generally accepted notion of security for public-key encryption is indistinguishability of ciphertexts under chosen-ciphertext attacks (IND-CCA, cf. [27, 30, 16]). For IND-CCA security, it must not be possible to tell which one of two adversarially chosen messages is encrypted, even when given access to a decryption oracle. The notion of IND-CCA security has proved extremely useful. On the one hand, it essentially captures the notion of a secure channel against active attacks (see [9, 12]). On the other hand, efficient IND-CCA secure encryption schemes can be constructed under standard number-theoretic assumptions (e.g., [13, 26, 23]).

However, there are realistic scenarios in which IND-CCA security is not known to provide security. For instance, consider a setting in which a large (and possibly a priori unknown) number of small devices send data to a single receiver. Each device encrypts its messages using the receiver's public key. Now assume an adversary that has the power to adaptively corrupt a subset of these devices. Say that, upon corrupting a device, the adversary learns the device's complete internal state, including the random coins used during previous encryptions. In that sense, the adversary may ask for *selective openings* of ciphertexts. The obvious question is: do the *unopened* ciphertexts remain secure? That is, can the adversary conclude anything about the plaintexts sent by uncorrupted devices, beyond of course what is implied already by the revealed plaintexts? While intuitively, the answer should be "no" for a secure public-key encryption system, IND-CCA security does not seem to be immediately useful in this setting. (E.g., [21] shows that whenever encryption constitutes a commitment to the respective message, the scheme cannot be proven secure using black-box techniques. This holds independent of whether the scheme is IND-CCA secure or not.) We clarify that the problem becomes moot if the senders can erase their randomness after sending the encrypted messages (cf. [1]). However, reliable erasure is difficult on a real system. As such, we will only focus on solutions that do not require erasures.

So far, only little is known on the construction of public key encryption schemes that are secure under selective opening attacks (SOA secure) as discussed above. Concretely, [3, 5] have shown that every *lossy* encryption scheme (cf. [29]) is SOA secure against *passive* (i.e., eavesdropping) adversaries. This yields a generic construction of SOA secure encryption that allows for fairly efficient instantiations. However, [3, 5] leave open the question of designing schemes that are SOA secure against *active* adversaries.

Our contribution. We construct practical public key encryption schemes that are SO-CCA secure, i.e., SOA secure against active attacks. Interestingly, we substantially deviate from previous techniques to obtain SOA security. To explain our approach, let us briefly sketch how [3, 5] employ lossy encryption to achieve SOA security.

(Passive) SOA security from lossy encryption. Lossy encryption schemes have the property that the scheme's "real" public key can be substituted with a "lossy" public key. Real and lossy keys are computationally indistinguishable, so – at least in a passive security experiment – this change cannot be detected by an adversary. Now lossy keys have the property that encryptions performed with them yield "lossy" ciphertexts that are statistically independent of the plaintext. In particular, a given lossy ciphertext can be – in general inefficiently – explained (or, opened) as an encryption of an arbitrary plaintext. Consequently, an SOA adversary cannot distinguish real keys, ciphertexts, and openings from those implied by lossy keys. But in the lossy case, the adversary's view is statistically independent of unopened messages; SOA security follows.

SOA-CCA security from lossy encryption, and its limitations. Now consider an active SOA adversary (i.e., one that is equipped with a decryption oracle). To prove SO-CCA security, now additionally adversarial decryption queries have to be answered. Obviously, this is impossible with fully lossy keys (i.e., keys that *always* encrypt to ciphertexts that are independent of the plaintext). In the IND-CCA case (see [29]), the solution to this dilemma is to make sure that *only* the challenge ciphertext is lossy. Technically, the security proof consider an "all-but-one" (ABO) public key. The ABO public key only encrypts the challenge ciphertext into a lossy ciphertext, and the corresponding ABO secret key can be used to decrypt any ciphertext except the lossy challenge ciphertext (and thus can be used to answer decryption queries).

This technique works well in the IND-CCA case, since there we have only one challenge ciphertext. However, with SOA security, we have to deal with a – possibly huge – vector of challenge ciphertexts that correspond to all openable ciphertexts. We would need "all-but-many" public keys that allow to make *only* the challenge ciphertexts lossy. (In fact, this is the exact approach taken by [20].) However, a counting argument shows that now the public-key size is at least linear in the maximal number of challenge ciphertexts. In realistic examples as the one above, there might be thousands of openable challenge ciphertexts. Hence, the lossy encryption approach leads to fairly impractical schemes, which have huge keys, and which actually achieve only *bounded* SO-CCA security. The latter means that the number of challenge ciphertexts for which the scheme is secure, is limited once the public key is chosen. If the number of potentially openable ciphertexts happens to exceed this limit, nothing is guaranteed anymore.

Another limitation of this approach is that, unless a lossy ciphertext is *efficiently* openable (a property which is not known to hold for most lossy encryption schemes), the lossy encryption paradigm only achieves (bounded) so-called IND-SO-CCA security. This in particular means that SOA security is only guaranteed for joint message distributions that are *efficiently conditionally re-samplable*. This means that even when *conditioned* on an arbitrary fixed subvector of messages, the remaining messages need to be efficiently samplable.[1] Many realistic settings (e.g., encryptions of ciphertexts, commitments, or signatures for fixed messages) correspond to *not* efficiently conditionally re-samplable message distributions. So without extra assumptions, lossy encryption implies only bounded SOA security in a restricted class of settings.

Our approach. We show SOA security using techniques from non-committing, resp. deniable encryption (e.g., [10, 15, 24, 11]). Non-committing encryption (NCE) schemes allow for "equivocable" ciphertexts that are computationally indistinguishable from real

[1] We remark that it is not obvious from [20] that their IND-SO-CCA secure scheme (Section 4.6) requires this additional condition on the distribution of the challenge messages. However, if this condition is not satisfied, then the challenger in the ideal game (in the definition of IND-SO-CCA security) is *inefficient*, and as such it *cannot* be argued in the security proof that in the ideal game the real public key can be replaced by a lossy key.

ciphertexts, but can be efficiently opened arbitrarily.[2] To achieve security against selective opening attacks, we rely on an idea from the deniable encryption scheme of Canetti et al. [11]. In their scheme, an encryption of 0 corresponds to a random string and that of 1 corresponds to a pseudorandom string (with a sparse range); it is easy to see that 1-encryptions are equivocable and can be opened as both 0 and 1. We will similar ideas in our schemes, which allows us to turn *all* SOA challenge ciphertexts into equivocable ones *one by one*. (Recall that in a sense, the reason why the lossy encryption paradigm does not mesh well with SO-CCA security is that lossy encryption only provides a handle to turn *all* challenge ciphertexts into lossy ones *at once*.) Finally, when all challenge ciphertexts are equivocable, we can argue that they do not contain any information about the unopened messages, and SOA security follows. Unlike previous constructions based on lossy encryption, we do not change the distribution of the public key in either our simulation or in the analysis.

We stress that the complexity our scheme does not depend on the number of challenge ciphertexts. So at the time of, say, constructing a PKI using our scheme, the number of potentially openable ciphertexts does not have to be known. We also remark that our approach achieves SOA security against *arbitrary* message distributions. We do not need to make extra assumptions on the underlying encryption scheme, or on the message distribution.

We first showcase our approach with a conceptually very simple scheme that is SO-CPA secure, i.e., SOA secure against passive attacks. Interestingly, we can base our proof upon general complexity assumptions, i.e., on the assumption of (a slightly enhanced version of) trapdoor one-way permutations. Going further, by our discussion above, NCE techniques do not necessarily suffer from the limitations of lossy encryption when it comes to active attacks. However, we have yet to describe how to handle decryption queries in the security proof, and, indeed, the simple SO-CPA secure scheme needs to be adjusted in several non-trivial ways in order to obtain our SO-CCA secure scheme.

Our scheme. In our SO-CCA secure scheme, encryption of a (multi-bit) message is performed bitwise, with one ciphertext element per bit. If the plaintext bit is 1, the corresponding ciphertext element X is an element of the language $\mathcal{L}$ associated with a hash proof system (HPS, cf. [14]). If the bit is 0, the ciphertext element is a random element, which will most likely be not in $\mathcal{L}$. Additionally, the ciphertext contains an authentication tag T, whose key K is the HPS key[3] associated to X in case $X \in \mathcal{L}$ (computed with the help of the witness), and a random key is taken in case $X \notin \mathcal{L}$. Decryption checks if the authentication tag T is verified correctly by the HPS key $\hat{K}$ computed from X (by means of the HPS secret key), which is the case iff $X \in \mathcal{L}$, i.e., 1 was encrypted. This approach is somewhat similar to the original Cramer-Shoup cryptosystem ([13, 14]), only that the HPS keys are used for authentication and not to directly pad a message.

[2] NCE talks about openings in which *secret keys*, as opposed to encryption randomness, are released. As a consequence, NCE schemes are comparatively inefficient and have severe limitations (see [28]). Our work shows that when "opening" refers to encryption randomness only, then NCE techniques allow for quite practical schemes.

[3] We adopt the notation of [22, 25] to view a HPS as a key encapsulation mechanism, i.e., to call HPS instances "ciphertexts" and HPS proofs "keys."

Opening a ciphertext part as an encryption of 1 means releasing a witness for $X \in \mathcal{L}$. Opening as an encryption of 0 means releasing the randomness used to randomly sample X. The crucial observation now is that 1-encryptions are equivocable: to open a 1-encryption as a 0-encryption, simply claim that X and K were randomly sampled, and provide the corresponding coins. Hence, equivocating all challenge ciphertexts means substituting them by all-one encryptions. This can be done as follows. For any $X \notin \mathcal{L}$, first the corresponding randomly chosen key K is replaced by the corresponding HPS key (which does not change the adversary's view due to statistical properties of the HPS), and then X is replaced by $X \in \mathcal{L}$ (which is indistinguishable to the adversary due to the assumed hardness of $\mathcal{L}$).

In order to have CCA security, it is important that the above changes can be done (and argued) while at the same time being able to answer decryption queries. This is indeed the case in our construction since decryption queries can be answered with the help of the HPS secret key, while the hardness of distinguishing $X \in \mathcal{L}$ from $X \notin \mathcal{L}$ holds even when given the HPS secret key.

The formal security proof uses ideas similar to those of Cramer and Shoup. We stress, however, that our proof is structured quite differently, since additional complications arise due to the fact that each ciphertext contains several X's (one for each plaintext bit), and we have several challenge ciphertexts. Due to this, it will be crucial how exactly and in which order the challenge ciphertexts are substituted by all-one encryptions. Furthermore, we need an authentication tag T that allows to "glue" together in a non-malleable way the L HPS ciphertexts $X_1, \dots, X_L$, obtained by encrypting an L-bit message, via their corresponding keys $K_1, \dots, K_L$.

Cross-authentication code. In order to "glue" HPS ciphertexts together, we make use of a new kind of information-theoretic authentication technique, which we call *cross-authentication*. Recall that in standard authentication, the authentication tag is computed from the message and the key, and can then be used to verify the authenticity of the message with the help of the key. In a cross-authentication code (XAC), the authentication tag is instead computed from a *list* $K_1, \dots, K_L$ of keys (and there is no designated message). It should be possible to verify the correctness of the tag T with *any* single key K_i from the list, and it should be hard for an adversary to forge a tag T' that is accepted by one of the keys, even if the adversary is given all the remaining keys and a correctly computed tag T. To the best of our knowledge, this concept has not been studied before. It is an important ingredient to our construction but might also find other applications as well. We give a formal definition and propose an efficient construction.

Other related work. Dwork et al. [17] study SOA security of commitments, and provide a connection to the Fiat-Shamir methodology. Hemenway et al. [20] were the first to devise SO-CCA secure public-key encryption schemes. Their most efficient schemes have compact ciphertexts of size independent of the number of challenge ciphertexts. Yet, all their constructions follow the lossy encryption paradigm and thus suffer from the drawbacks that are inherent to that approach. Hence, unless the lossy encryption satisfies some additional property, they only prove the weaker IND-SO-CCA security notion, which in particular requires the distribution of the challenge messages to be efficiently conditionally re-samplable. Furthermore, the size of their public and secret keys still depends on the number of challenge ciphertexts. In contrast, our constructions are comparatively efficient, completely independent of the number of challenge ciphertexts, and do not make assumptions about the distribution of the challenge messages (beyond the usual requirement of being efficiently samplable). Bellare et al. [4] propose a (passively) SOA secure identity-based encryption scheme that is also based on NCE techniques. However, their result does not directly yield a SO-CCA secure public-key encryption scheme, say, by applying the IBE→PKE transformation of Boneh, Canetti, Halevi, and Katz [8]. (In a nutshell, the reason is that [8] use a one-time signature scheme that may lose its guarantees under selective opening attacks.)

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 X, we denote by $x \leftarrow X$ the process of sampling x uniformly from X. For a probabilistic algorithm A, we denote $y \leftarrow A(x; R)$ 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 $R \in \mathcal{R}_A$, and we write $y_1, \ldots, y_m \leftarrow A(x)$ for $y_1 \leftarrow A(x), \ldots, y_m \leftarrow A(x)$ with fresh randomness in each execution. By $\mathsf{time}_A = \mathsf{time}_A(k) \in \mathbb{N} \cup \{\infty\}$, we denote the supremum of the running time of an algorithm A when running on security parameter k. If time_A is polynomial in k, then A is PPT.

Trapdoor one-way permutations and collision resistant hashing. Informally, a trapdoor one-way permutation should be hard to invert, unless given a trapdoor.

Definition 1 (Trapdoor one-way permutation). *A family of trapdoor one-way permutations $\mathcal{F}$ consists of three PPT algorithms* Gen, Eval *and* Inv *with the following properties.* $\mathsf{Gen}(1^k)$ *outputs the description of a permutation $f : \mathcal{D}_f \to \mathcal{D}_f$ and a trapdoor τ, and* $\mathsf{Eval}(f, x) = f(x)$ *and* $\mathsf{Inv}(\tau, x) = f^{-1}(x)$ *for all $x \in \mathcal{D}_f$. Furthermore, for every PPT algorithm A, the following function is negligible in k:*

$$\Pr\left[A(pk, f(x)) = x \mid (f, \tau) \leftarrow \mathsf{Gen}(1^k), x \leftarrow \mathcal{D}_f\right].$$

Note that we do not distinguish between the function f and its description output by Gen. Furthermore, to simplify notation, we usually leave the algorithms Gen, Eval and Inv implicit and write $(f, f^{-1}) \leftarrow \mathcal{F}$ to denote that a public/secret-key pair is generated using $\mathsf{Gen}(1^k)$, and we write $f(x)$ and $f^{-1}(x)$ to denote that $\mathsf{Eval}(f, x)$ and $\mathsf{Inv}(\tau, x)$ are executed.

Informally, a hash function H is collision resistant if it is infeasible to find two distinct preimages x, x' with $\mathsf{H}(x) = \mathsf{H}(x')$.

Definition 2 (Collision-resistant hash function). *A* collision-resistant hash function *$\mathcal{H}$ with domain $\mathcal{D} = \mathcal{D}_k$ and range $\mathcal{R} = \mathcal{R}_k$ consists of two PPT algorithms* Gen *and* Eval *with the following properties.* $\mathsf{Gen}(1^k)$ *outputs the description of a function* $\mathsf{H} : \mathcal{D} \to \mathcal{R}$ *such that* $\mathsf{Eval}(K, x) = \mathsf{H}(x)$ *for all $x \in \mathcal{D}$. Furthermore, for every PPT algorithm B, the following function is negligible in k:*

$$\mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},B}(k) := \Pr\left[x \neq x' \wedge \mathsf{H}(x) = \mathsf{H}(x') \mid \mathsf{H} \leftarrow \mathsf{Gen}(1^k), (x, x') \leftarrow B(\mathsf{H})\right]$$

Similarly to above, we do not distinguish between the function H and its description output by Gen and we usually leave the algorithms Gen and Eval implicit and write $\mathsf{H} \leftarrow \mathcal{H}$ to denote that H is generated by Gen.

Encryption schemes and security under selective openings. A public-key encryption scheme consists of three 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 and all $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$, and with overwhelming probability over $C \leftarrow (pk, M)$.

Following [17, 21, 3, 5, 20], we present a definition for security under selective openings that captures security of an encryption scheme under adaptive attacks. The definition is simulation-based (much like semantic security [19]), and demands that whatever an adversary that sees a vector of ciphertexts deduces can also be deduced by a simulator that does not see any ciphertexts. To model adaptive corruptions, our notion also allows both adversary and simulator to request "openings" of adaptively selected ciphertexts. (Since the simulator does not actually get to see any ciphertexts, it may only ask to see selected components of an initially unknown message vector.)

Definition 3 (SO-CPA, SO-CCA security). *A public-key encryption scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* chosen-plaintext secure under selective openings *(short:* SO-CPA secure*) iff for every polynomially bounded* $n = n(k) > 0$*, every PPT function* R*, and every stateful PPT machine* A *(the adversary), there is a stateful PPT machine* S *(the simulator), such that* $\mathsf{Adv}^{\mathsf{cpa\text{-}so}}_{\mathsf{PKE},A,S,R}$ *is negligible. Here*

$$\mathsf{Adv}^{\mathsf{cpa\text{-}so}}_{\mathsf{PKE},A,S,R}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{cpa\text{-}so\text{-}real}}_{\mathsf{PKE},A,R}(k) = 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{so\text{-}ideal}}_{S,R}(k) = 1\right],$$

where the experiments $\mathsf{Exp}^{\mathsf{cca\text{-}so\text{-}real}}_{\mathsf{PKE},A,R}(k)$ *and* $\mathsf{Exp}^{\mathsf{so\text{-}ideal}}_{S,R}(k)$ *are defined as follows:*

```
Experiment $\mathsf{Exp}^{\mathsf{cpa\text{-}so\text{-}real}}_{\mathsf{PKE},A,R}$
  $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$
  $\mathcal{M} \leftarrow A(\mathtt{dist}, pk)$
  $\mathbf{M} := (M^i)_{i \in [n]} \leftarrow \mathcal{M}$
  $\mathbf{R} := (R^i)_{i \in [n]} \leftarrow (\mathcal{R}_{\mathsf{Enc}})^n$
  $\mathbf{C} := (C^i)_{i \in [n]} := (\mathsf{Enc}(pk, M^i; R^i))_{i \in [n]}$
  $I \leftarrow A(\mathtt{select}, \mathbf{C})$
  $out_A \leftarrow A(\mathtt{output}, (M^i, R^i)_{i \in I})$
  return $R(\mathcal{M}, \mathbf{M}, out_A)$
```

```
Experiment $\mathsf{Exp}^{\mathsf{so\text{-}ideal}}_{S,R}$
  $\mathcal{M} \leftarrow S(\mathtt{dist})$
  $\mathbf{M} := (M^i)_{i \in [n]} \leftarrow \mathcal{M}$
  $I \leftarrow S(\mathtt{select}, (1^{|M^i|})_{i \in [n]})$
  $out_S \leftarrow S(\mathtt{output}, (M^i)_{i \in I})$
  return $R(\mathcal{M}, \mathbf{M}, out_S)$
```

Furthermore, we define

$$\mathsf{Adv}^{\mathsf{cca\text{-}so}}_{\mathsf{PKE},A,S,R}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{cca\text{-}so\text{-}real}}_{\mathsf{PKE},A,R}(k) = 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{so\text{-}ideal}}_{S,R}(k) = 1\right]$$

for an experiment $\mathsf{Exp}^{\mathsf{cca\text{-}so\text{-}real}}_{\mathsf{PKE},A,R}$ *that is defined like* $\mathsf{Exp}^{\mathsf{cpa\text{-}so\text{-}real}}_{\mathsf{PKE},A,R}$*, but grants the adversary (in all stages of the attack) access to a decryption oracle* $\mathsf{Dec}(sk, \cdot)$*. We require that* A

never queries $\mathsf{Dec}(sk, \cdot)$ *on a challenge ciphertext* C^i. *We say that* PKE *is* chosen-ciphertext secure under selective openings *(short:* SO-CCA secure*) if for all* n, R, *and* A *there exists* S *such that* $\mathsf{Adv}^{\mathsf{cca\text{-}so}}_{\mathsf{PKE},A,S,R}(k)$ *is negligible.*

A few remarks about Definition 3 are in place:

- We assume that the distribution $\mathcal{M}$ that A outputs is encoded as a circuit that samples n-tuples of messages according to this distribution. Since A is PPT, this enforces efficient samplability of $\mathcal{M}$. Efficient samplability of $\mathcal{M}$ is a standard and much weaker requirement than the *efficient conditional re-samplability* requirement from the indistinguishability-based selective opening security definitions IND-SO-ENC [3, 5] or IND-SO-CCA2 [20]. We also note that since A chooses $\mathcal{M}$ adaptively (i.e., dependent on pk), SO-CCA security as defined above implies IND-CCA security (see [2] for a convenient formalization).
- We stress that Definition 3 requires the specified security property to hold for *any* (polynomially bounded) n. This is in contrast to the schemes in [20], in which the public key pk depends on n, so once pk is chosen, security is only guaranteed for challenge ciphertexts of bounded length.
- Our notion of “opening of a ciphertext” corresponds to sender corruptions: as an opening, we release plaintext and encryption randomness, but not decryption key. While this clearly poses a significant restriction, it is in a certain sense the best we can hope for without resorting to non-black-box or non-committing encryption techniques (see [21, Section 5]).
- Like [17, 21, 3, 5, 20], we model only one layer of adaptivity. (That is, the adversary may choose only once a subset of ciphertexts to be opened.) More realistic notions would model several stages of adaptive corruption, but would also be substantially more complicated in description and handling. We stress that our SO-CCA secure encryption scheme to be presented does not rely on the assumption of only one corruption stage.
- We allow the length of the messages transmitted by the various senders to vary depending on the randomness of the message distribution $\mathcal{M}$ and the identity of the sender, and we provide this information (i.e. the message lengths $|M^1|, \ldots, |M^n|$) to the simulator. Indeed, we cannot prevent the adversary from always choosing to corrupt the $n/2$ senders that send the longest messages.

Sender-equivocable encryption schemes. We formalize the notion of sender equivocability, which (for CPA security) is similar to non-committing encryption except the adversary is only allowed to corrupt the sender but not the receiver. In addition, we require that to equivocate, the simulator only needs to know the random coins used to generate the simulated ciphertext (and not those for the simulated public key). This latter requirement is needed because unlike the set-up for non-committing encryption, all ciphertexts are generated using the same public key in the selective opening attacks.

Definition 4 (NC-CPA, NC-CCA security). *A public-key encryption scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* sender-equivocable *(short:* NC-CPA secure*) iff there is a stateful PPT machine* S *(the simulator) such that for every stateful PPT machine* A *(the adversary)* $\mathsf{Adv}^{\mathsf{cpa\text{-}nc}}_{\mathsf{PKE},A,S}$ *is negligible. Here*

$$\mathsf{Adv}^{\text{cpa-nc}}_{\mathsf{PKE},A,S}(k) := \Pr\left[\mathsf{Exp}^{\text{cpa-nc-real}}_{\mathsf{PKE},A}(k) = 1\right] - \Pr\left[\mathsf{Exp}^{\text{cpa-nc-ideal}}_{\mathsf{PKE},A}(k) = 1\right],$$

where the experiments $\mathsf{Exp}^{\text{cca-nc-real}}_{\mathsf{PKE},A}(k)$ *and* $\mathsf{Exp}^{\text{cpa-nc-ideal}}_{\mathsf{PKE},A}(k)$ *are defined as follows:*

```
Experiment $\mathsf{Exp}^{\text{cpa-nc-real}}_{\mathsf{PKE},A}$
  $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$
  $(M, z) \leftarrow A(\mathtt{dist}, pk)$
  $R \leftarrow \mathcal{R}_{\mathsf{Enc}}$
  $C := \mathsf{Enc}(pk, M; R)$
  return $A(\mathtt{output}, M, C, R, z)$
```

```
Experiment $\mathsf{Exp}^{\text{cpa-nc-ideal}}_{\mathsf{PKE},A}$
  $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$
  $(M, z) \leftarrow A(\mathtt{dist}, pk)$
  $C \leftarrow S(\mathtt{sim}, pk, 1^{|M|})$
  $R \leftarrow S(\mathtt{open}, M)$
  return $A(\mathtt{output}, M, C, R, z)$
```

Furthermore, we define

$$\mathsf{Adv}^{\text{cca-nc}}_{\mathsf{PKE},A,S}(k) := \Pr\left[\mathsf{Exp}^{\text{cca-nc-real}}_{\mathsf{PKE},A}(k) = 1\right] - \Pr\left[\mathsf{Exp}^{\text{cca-nc-ideal}}_{\mathsf{PKE},A}(k) = 1\right]$$

for an experiment $\mathsf{Exp}^{\text{cca-nc-real}}_{\mathsf{PKE},A}$ *that is defined like* $\mathsf{Exp}^{\text{cpa-nc-real}}_{\mathsf{PKE},A}$ *but grants the adversary* A *(in all stages of the attack) access to a decryption oracle* $\mathsf{Dec}(sk, \cdot)$*. We also consider an experiment* $\mathsf{Exp}^{\text{cca-nc-ideal}}_{\mathsf{PKE},A}$ *that is defined like* $\mathsf{Exp}^{\text{cpa-nc-ideal}}_{\mathsf{PKE},A}$*, but also grants* A *access to* $\mathsf{Dec}(sk, \cdot)$*. In both experiments, we require that* A *never queries* $\mathsf{Dec}(sk, \cdot)$ *on the challenge ciphertext* C*. We say that* PKE *is* chosen-ciphertext secure under selective openings *(short:* NC-CCA secure*) if there exists* S *such that for all* A*,* $\mathsf{Adv}^{\text{cca-nc}}_{\mathsf{PKE},A,S}(k)$ *is negligible.*

The next lemma says that if an encryption scheme is NC-CPA secure (resp. NC-CCA secure), then it is also SOCPA secure (resp. SOCCA secure). An analogous statement was shown in [10] in the context of non-committing encryption and adaptive corruptions; the main technical difference is that we achieve security amidst selective opening attacks with respect to a single public key.

Lemma 1 (NC-CPA security implies SO-CPA security). *Suppose* PKE *is NC-CPA secure with simulator* S*. Then, for every adversary* A *and every function* R*, there exists an adversary* B *and a simulator* S'*, such that*

$$\left|\mathsf{Adv}^{\text{cpa-so}}_{\mathsf{PKE},A,S',R}(k)\right| \leq n \left|\mathsf{Adv}^{\text{cpa-nc}}_{\mathsf{PKE},B,S}(k)\right|. \tag{1}$$

We have $\mathsf{time}_{S'} \approx \mathsf{time}_A + n \cdot \mathsf{time}_S + \mathsf{time}_R$*. Moreover, if* PKE *is NC-CCA secure, then we have that*

$$\left|\mathsf{Adv}^{\text{cca-so}}_{\mathsf{PKE},A,S',R}(k)\right| \leq n \left|\mathsf{Adv}^{\text{cca-nc}}_{\mathsf{PKE},B,S}(k)\right|. \tag{2}$$

with the same relation $\mathsf{time}_{S'} \approx \mathsf{time}_A + n \cdot \mathsf{time}_S + \mathsf{time}_R$*.*

The proof idea is very simple: the SOCPA simulator S' generates n equivocable ciphertexts independently, one for each sender and forward these ciphertexts to the adversary A. When A asks for an opening set I, S' relays this set to its own experiment, receives the corresponding messages in I, and opens the ciphertexts in the simulation suitably.

Proof (sketch). We first establish the claim for NC-CPA vs SO-CPA. Here, the simulator S' internally simulates a copy of A and proceeds as follows:

- On input $\mathtt{dist}$, run $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$ and output $\mathcal{M} \leftarrow A(\mathtt{dist}, pk)$;
- On input $(\mathtt{select}, (1^{|M^i|})_{i\in[n]})$, run $\mathbf{C} = (C^i)_{i\in[n]} \leftarrow (S(\mathtt{sim}, pk, 1^{|M^i|}))_{i\in[n]}$ and output $I \leftarrow A(\mathtt{select}, \mathbf{C})$
- On input $(\mathtt{output}, (M^i)_{i\in I})$, compute $R^i \leftarrow S(\mathtt{open}, M^i)$ for $i \in I$ and return $out_A \leftarrow A(\mathtt{output}, (M^i, R^i)_{i\in I})$

The analysis proceeds via a series of games, where in Game j, $j = 0, 1, \ldots, n$, the first j ciphertexts are generated using $S(\mathtt{sim}, pk)$, and the corresponding randomness using $S(\mathtt{open}, pk, M^i)$. The last $n-j$ ciphertexts are generated using $\mathsf{Enc}(pk, M^i; R^i)$ with randomness R^i. We claim that the sum (over $j = 1, \ldots, n$) of the distinguishing probabilities between Game $j-1$ and Game j is bounded by $n\mathsf{Adv}^{\mathsf{cpa\text{-}nc}}_{\mathsf{PKE},B,S}(k)$, where B uniformly guesses $j \in [n]$, and internally simulates a copy of A as follows:

- On input $\mathtt{dist}, pk$, compute $\mathcal{M} \leftarrow A(\mathtt{dist}, pk)$ and $\mathbf{M} = (M^i)_{i\in[n]} \leftarrow \mathcal{M}$, and output $(M, z) = (M_j, \mathbf{M})$.
- On input $(\mathtt{output}, M, C, R, z)$, compute

$$C^i = \begin{cases} S(\mathtt{sim}, pk, 1^{|M^i|}) & \text{if } i < j \\ C & \text{if } i = j \\ \mathsf{Enc}(pk, M^i; R^i) & \text{if } i > j \end{cases}$$

compute $I \leftarrow A(\mathtt{select}, \mathbf{C})$, $out_A \leftarrow A(\mathtt{output}, (M^i, R^i)_{i\in I})$ and output $R(\mathcal{M}, \mathbf{M}, out_A)$.

For NC-CCA vs SO-CCA, the simulator is exactly as above, except it also simulates $\mathsf{Dec}(sk, \cdot)$ which it can since it knows (pk, sk).

3 Warmup: An NC-CPA Secure Scheme

We focus on constructing NC-CPA and NC-CCA secure schemes, which by Lemma 1, are respectively SO-CPA and SO-CCA secure.

Ingredients. As a warmup for our NC-CCA secure scheme, and to explain one of the key ideas, we construct an efficient NC-CPA secure scheme from a slightly enhanced version of trapdoor one-way permutations. Namely, we require that there exist algorithms for sampling the domain $\mathcal{D}_f$, and for explaining an arbitrary $x \in \mathcal{D}_f$ as a result of sampling $\mathcal{D}_f$:

Definition 5 (Efficiently samplable and explainable domain). *A domain $\mathcal{D}_f$ is* efficiently samplable and explainable *iff there exist PPT algorithms* Sample *and* $\mathsf{Explain}$ *such that* $\mathsf{Sample}(\mathcal{D}_f; R)$ *is uniformly distributed over $\mathcal{D}_f$ for $R \leftarrow \mathcal{R}_{\mathsf{Sample}}$, and* $\mathsf{Explain}(\mathcal{D}_f, x)$ *outputs R that is uniformly distributed subject to* $\mathsf{Sample}(\mathcal{D}_f; R) = x$ *for any $x \in \mathcal{D}_f$.*

Explainability is a vital property in the construction of non-committing encryption schemes (see Damgård and Nielsen [15]; there, an essentially equivalent property is called "invertible sampling"). We stress that the domain of most "natural" trapdoor one-way permutations satisfies Definition 5.[4] Note that for families of trapdoor one-way permutations, explainability implies that the family is *enhanced* in the sense of Goldreich [18], Appendix C.1.

Hence, let $\mathcal{F}$ be a family of trapdoor one-way permutations $f : \mathcal{D}_f \to \mathcal{D}_f$ with efficiently samplable and explainable domain $\mathcal{D}_f$ (for every $f \in \mathcal{F}$), and hard-core predicate $h : \mathcal{D}_f \to \{0,1\}$. For $(f, f^{-1}) \leftarrow \mathcal{F}$ and $\ell = \ell(k)$, define

$$\mathsf{BM}_{f,\ell}(x) := (h(x), h(f(x)), \ldots, h(f^{\ell-1}(x))) \in \{0,1\}^{\ell}.$$

It is well-known that BM is pseudorandom, even given $f^{\ell}(x)$. Formally:

Theorem 1 (Blum and Micali [6]). *Let $\mathcal{F}$ a family of trapdoor one-way permutations $f : \mathcal{D}_f \to \mathcal{D}_f$ with hard-core predicate $h : \mathcal{D}_f \to \{0,1\}$. Then, for every PPT distinguisher D and every polynomially bounded $\ell = \ell(k)$, the function*

$$\mathsf{Adv}^{\mathsf{prg}}_{\mathcal{F},\ell,D}(k) := \Pr\left[D(f^{\ell}(x), \mathsf{BM}_{f,\ell}(x)) = 1\right] - \Pr\left[D(x, K) = 1\right]$$

is negligible in k, where $(f, f^{-1}) \leftarrow \mathcal{F}$, $x \leftarrow \mathcal{D}_f$, and $K \leftarrow \{0,1\}^k$.

The scheme. For $\mathcal{F}$ as above and a message space of $\{0,1\}$, our NC-CPA secure encryption scheme $\mathsf{NCCPA} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is defined as:

$\mathsf{Gen}(1^k)$. Sample $(f, f^{-1}) \leftarrow \mathcal{F}$, and return $(pk, sk) = (f, f^{-1})$.

$\mathsf{Enc}(pk, M; R)$. Parse $pk = f$, $M \in \{0,1\}$, and $R = (R^x, K_0) \in \mathcal{R}_{\mathsf{Sample}} \times \{0,1\}^k$. Set $x \leftarrow \mathsf{Sample}(\mathcal{D}_f; R^x)$ and return

$$C := (y, K) := \begin{cases} (f^k(x), \mathsf{BM}_{f,k}(x)) & \text{if } M = 1 \\ (x, K_0) & \text{if } M = 0. \end{cases}$$

$\mathsf{Dec}(sk, C)$. Parse $sk = f^{-1}$ and $C = (y, K)$. Return $M = 1$ if $\mathsf{BM}_{f,k}(f^{-k}(y)) = K$, and $M = 0$ else.[5]

Note that 1-encryptions are always correctly decrypted, while 0-encryptions are wrongly decrypted to 1 with probability 2^{-k}. Furthermore, larger messages can be encrypted by concatenating ciphertexts. (This does not affect NCCPA's NC-CPA security.)

Equivocable ciphertexts and sketch of security proof. The key to proving NC-CPA security is that 1-encryptions are equivocable. More concretely, the NC-CPA simulator S proceeds as follows:

[4] Damgård and Nielsen [15] show that any dense subset $\mathcal{D}_f$ of an efficient samplable domain is both efficiently samplable and explainable as long as $\mathcal{D}_f$ admits an efficient membership test. For the trapdoor permutations based on RSA, the public index is a RSA modulus N and the domain $\mathbb{Z}_N^*$ clearly satisfies these properties. For Rabin's trapdoor permutations based on modular squaring, the public index is a Blum integer N and we need to modify the domain to be the group of signed quadratic residues in $\mathbb{Z}_N^*$.

[5] Note that $\mathsf{BM}_{f,k}(f^{-k}(y))$ can be computed from f^{-1} and y alone.

- On input $(\mathtt{sim}, pk)$ where pk is the public key f, it returns a random 1-encryption given by $(y, K) = (f^k(x), \mathsf{BM}_{f,k}(x))$ with randomness R;
- On input $(\mathtt{open}, M)$ for $M \in \{0, 1\}$, it returns $(\mathsf{Explain}(\mathcal{D}_f, y), K)$ if $M = 0$ and R if $M = 1$.

A straightforward hybrid argument to BM's pseudorandomness shows that this simulation achieves a computationally indistinguishable view for A in real experiment and ideal simulation. We obtain:

Theorem 2 (NCCPA **is NC-CPA secure).** *For every adversary A and every function R, there exists a simulator S, and a distinguisher D, such that*

$$\left|\mathsf{Adv}^{\mathsf{cpa\text{-}so}}_{\mathsf{SOCCA},A,S,R}(k)\right| \leq n \left|\mathsf{Adv}^{\mathsf{prg}}_{\mathcal{F},k,D}(k)\right| . \tag{3}$$

We have $\mathsf{time}_S \approx \mathsf{time}_A$ *and* $\mathsf{time}_D \approx \mathsf{time}_A + \mathsf{time}_R$.

We omit a more detailed proof, since the proof of Theorem 2 is similar to, but conceptually simpler than the upcoming proof for our NC-CCA secure scheme.

Relation to non-committing encryption. We point out that NCCPA can be seen as a variant of non-committing encryption schemes in [10, 24]. Compared with these schemes, our scheme is more efficient and conceptually simpler. It also allows for an unbounded usage, since we only need to provide encryption random coins (but not secret keys) upon an opening. As such, NCCPA serves as a useful tool to explain how we use equivocable ciphertexts to prove security under selective openings. The main technical difficulties lie in designing and analyzing a chosen-ciphertext secure scheme. This will turn out to be a delicate task that requires some more preparation.

4 Hash Proof Systems with Explainable Domains

We recall the notions of a subset membership problem and of an (extended) hash proof system, as introduced in Cramer and Shoup [14]. In our definitions, we require all properties to hold *perfectly*; this can be relaxed by allowing a negligibly small error probability (which includes that sampling algorithms may produce near-uniform output).

Definition 6 (Subset membership problem). *A* subset membership problem SMP *consists of the following PPT algorithms.*

System parameter generation. $\mathsf{SysGen}(1^k)$ *outputs system parameters ρ that defines a set $\mathcal{X}_\rho$ of ciphertexts and a language $\mathcal{L}_\rho \subseteq \mathcal{X}_\rho$. $\mathcal{X}_\rho$ is required to be efficiently recognizable (given ρ).*

Sampling from $\mathcal{L}_\rho$. $\mathsf{SampleL}(\mathcal{L}_\rho; W)$ *uniformly samples $X \leftarrow \mathcal{L}_\rho$ using randomness W.*

A subset membership problem SMP *is called* hard *iff $\mathcal{X}_\rho$ and $\mathcal{L}_\rho$ are computationally indistinguishable. Concretely, for every PPT distinguisher D, the following function is negligible:*

$$\mathsf{Adv}^{\mathsf{sm}}_{\mathsf{EHPS},D}(k) := \Pr\left[D(X) = 1 \mid X \leftarrow \mathcal{X}_\rho \setminus \mathcal{L}_\rho\right] - \Pr\left[D(X) = 1 \mid X \leftarrow \mathcal{L}_\rho\right]$$

where in both probabilities $\rho \leftarrow \mathsf{SysGen}(1^k)$.

Definition 7 (EHPS). *An* extended hash proof system *(short:* EHPS*)* EHPS *for a subset membership problem* SMP *associates with each $\rho \leftarrow \mathsf{SysGen}(1^k)$ an efficiently recognizable set of keys $\mathcal{K}_\rho$ and an efficiently recognizable set of tags $\mathcal{T}_\rho$, and consists of the following PPT algorithms:*

Individual key generation. $\mathsf{HashGen}(\rho)$ *outputs a public key hpk and a secret key hsk. We assume that hpk and hsk both contain ρ.*

Secret evaluation. $\mathsf{SEval}(hsk, X, t)$ *computes a key $K \in \mathcal{K}_\rho$. We also write $K = hsk(X, t)$.*

Public evaluation (with witness). $\mathsf{PEval}(hpk, X, W, t)$ *computes a key $K \in \mathcal{K}_\rho$. We require correctness in the sense of* $\mathsf{PEval}(hpk, X, W, t) = \mathsf{SEval}(hsk, X, t)$ *for all $\rho \leftarrow \mathsf{SysGen}(1^k)$, $(hpk, hsk) \leftarrow \mathsf{HashGen}(\rho)$, $X \leftarrow \mathsf{SampleL}(\mathcal{L}_\rho; W)$, and all $t \in \mathcal{T}_\rho$.*

By definition, in an EHPS the public key hpk uniquely determines the action of SEval for ciphertexts $X \in \mathcal{L}_\rho$. An EHPS typically becomes interesting/useful when on the other hand the action of SEval for ciphtertexts $X \in \mathcal{X}_\rho \setminus \mathcal{L}_\rho$ is "very undetermined". We capture this as follows.

Definition 8 (2-universal). *An EHPS (for* SMP*) is* 2-universal *iff for all possible $\rho \leftarrow \mathsf{SysGen}(1^k)$, all hpk in the range of* $\mathsf{HashGen}(\rho)$*, and all distinct $(X_1, t_1), (X_2, t_2)$ in $(\mathcal{X}_\rho \setminus \mathcal{L}_\rho) \times \mathcal{T}$,*

$$\Pr\bigl[hsk(X_2, t_2) = K_2 \bigm| hsk(X_1, t_1) = K_1\bigr] = \frac{1}{|\mathcal{K}_\rho|},$$

where the probability is over possible hsk with $(hpk, hsk) \leftarrow \mathsf{HashGen}(\rho)$.

In addition to the above (standard) properties, we will also need the following non-standard requirements.

Definition 9 (Sparseness of the language). *An subset membership problem* SMP *has a* sparse language *if for $\rho \leftarrow \mathsf{SysGen}(1^k)$ and $X \leftarrow \mathcal{X}_\rho$, the probability that $X \in \mathcal{L}_\rho$ is negligible.*

Definition 10 (Explainable ciphertexts and keys). *We say that a subset membership problem* SMP *has* explainable ciphertexts *if the set $\mathcal{X}_\rho$ is efficiently samplable and explainable in the sense of Definition 5. Similarly, an extended hash proof system* EHPS *has* explainable keys *if the set $\mathcal{K}_\rho$ is efficiently samplable and explainable.*

We point out that explainable keys can actually be assumed without loss of generality, because $\mathcal{K}_\rho$ can always be efficiently mapped into $\mathcal{K}'_\rho = \{0,1\}^m$ by means of a suitable (almost) balanced function, such that uniform distribution in $\mathcal{K}_\rho$ induces (almost) uniform distribution in $\mathcal{K}'_\rho$, and where m is linear in $\log(|\mathcal{K}_\rho|)$. The requirement on the ciphertexts to be explainable, on the other hand, is a real restriction on the SMP; nevertheless, several suitable SMPs do satisfy this requirement and have a sparse language, as we will outline next.

Examples of suitable SMPs. The DDH-based SMPs from Cramer and Shoup [14] satisfy all our requirements, assuming that the platform group $\mathbb{G}$ is efficiently samplable and explainable in the sense of Definition 5. One popular such group in which DDH is assumed to be hard is the unique q-order subgroup of $\mathbb{Z}_p^*$, where $p = 2q + 1$ is a safe prime. Another one is the elliptic curve $\mathbb{G}_1$ from [7, Section 5.1]. The Paillier-based SMP from [14] fulfils our requirements as well. Finally, the SMP from [14] based on quadratic residuosity satisfies all our requirements *except* for a sparse language (Definition 9). However, the SMP that consists of, say, k parallel copies of the QR SMP from [14] (and where the EHPS key is the product of the individual keys) *has* a sparse language and satisfies our remaining requirements.

5 Cross-Authentication Codes

We introduce here a new information-theoretic authentication technique, which will play an important role in our construction of a SO-CCA-secure encryption scheme. However, the technique may also be useful in other contexts. Cross-authentication, as we call our technique, allows to compute an authentication tag T for a *list* $K_1, \dots, K_L$ of keys, with the following two properties. The tag T can be verified by any single key K_i from the list, and without knowledge of K_i it is information-theoretically hard to forge a tag T' that is correctly verified by K_i, even when given a correctly computed tag T and all the other keys $K_{\neq i} = (K_j)_{j \neq i}$.

Below is the formal definition followed by an efficient example construction.

Definition 11 **(L-Cross-authentication code).** *For $L \in \mathbb{N}$, an L-*cross-authentication code *(short: L-*XAC*)* XAC *consists of a key space $\mathcal{XK}$ and a tag space $\mathcal{XT}$ and of three PPT algorithms* XGen, XAuth *and* XVer. $\mathsf{XGen}(1^k)$ *produces a uniformly random key $K \in \mathcal{XK}$,* $\mathsf{XAuth}(K_1, \dots, K_L)$ *outputs a tag $T \in \mathcal{XT}$, and* $\mathsf{XVer}(K, i, T)$ *outputs a decision bit. The following is required:*

Correctness. *For all $i \in [L]$, the probability*

$$\mathsf{fail}_{\mathsf{XAC}}(k) := \Pr\left[\mathsf{XVer}(K_i, i, \mathsf{XAuth}(K_1, \dots, K_L)) \neq 1\right],$$

is negligible, where $K_1, \dots, K_L \leftarrow \mathsf{XGen}(1^k)$ in the probability.

Security against impersonation and substitution attacks. $\mathsf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k)$ *and* $\mathsf{Adv}^{\mathsf{sub}}_{\mathsf{XAC}}(k)$ *as defined below are both negligible:*

$$\mathsf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k) := \max_{i,T'} \Pr\left[\mathsf{XVer}(K, i, T') = 1 \mid K \leftarrow \mathsf{XGen}(1^k)\right]$$

where the max *is over all $i \in [L]$ and $T' \in \mathcal{XT}$, and*

$$\mathsf{Adv}^{\mathsf{sub}}_{\mathsf{XAC}}(k) := \max_{i, K_{\neq i}, F} \Pr\left[\begin{array}{l} T' \neq T \wedge \\ \mathsf{XVer}(K_i, i, T') = 1 \end{array} \middle| \begin{array}{l} K_i \leftarrow \mathsf{XGen}(1^k), \\ T := \mathsf{XAuth}(K_1, \dots, K_L), \\ T' \leftarrow F(T) \end{array}\right].$$

where the max *is over all $i \in [L]$, all $K_{\neq i} = (K_j)_{j \neq i} \in \mathcal{XK}^{L-1}$ and all (possibly randomized) functions $F : \mathcal{XT} \to \mathcal{XT}$.*

Note that by taking $\mathcal{R}_{\mathsf{XGen}}$ as key space, instead of $\mathcal{XK}$, we may without loss of generality assume that $\mathcal{XK}$ is of the form $\mathcal{XK} = \{0,1\}^r$ (and XGen simply outputs its randomness).

Example of a L-XAC. Let $\mathbb{F}$ be a finite field of size q, where q depends on k (e.g. $q = 2^k$). Set $\mathcal{XK} = \mathbb{F}^2$ and $\mathcal{XT} = \mathbb{F}^L \cup \{\perp\}$, and let XGen produce a random key in $\mathcal{XK} = \mathbb{F}^2$. For $K_1 = (a_1, b_1), \dots, K_L = (a_L, b_L) \in \mathcal{XK}$, the authentication tag $T = \mathsf{XAuth}(K_1, \dots, K_L)$ is given by the unique vector $T = (T_0, \dots, T_{L-1}) \in \mathbb{F}^L$ such that $p_T(a_i) = b_i$ for $i = 1, \dots, L$, where $p_T(x) = T_0 + T_1 x + \cdots + T_{L-1}x^{L-1} \in \mathbb{F}[x]$. T can be computed efficiently by solving the linear equation system $\mathbf{A}T = B$, where $\mathbf{A} \in \mathbb{F}^{L \times L}$ is the Vandermonde matrix whose i-th row is given by $1, a_i, a_i^2 \dots, a_i^{L-1}$, and where $B \in \mathbb{F}^L$ is the column vector with entries $b_1, \dots, b_L$. If $\mathbf{A}T = B$ admits more than one or no solution, then T is set to $\perp$ instead. For any $T \in \mathcal{XT}$, $K = (a,b) \in \mathcal{XK}$ and $i \in [L]$, the verification $\mathsf{XVer}(K,i,T)$ outputs 1 if and only if $T \neq \perp$ and $p_T(a) = b$.

Lemma 2. *The above L-XAC* XAC *satisfies:*

$$\mathsf{fail}_{\mathsf{XAC}}(k) \leq \frac{L(L-1)}{2q}\ , \quad \mathsf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k) \leq \frac{1}{q} \quad \textit{and} \quad \mathsf{Adv}^{\mathsf{sub}}_{\mathsf{XAC}}(k) \leq 2 \cdot \frac{L-1}{q}\ .$$

Proof. Correctness: By construction, $\mathsf{XVer}(K_i, i, \mathsf{XAuth}(K_1, \dots, K_L)) = 1$ except if the Vandermonde matrix $\mathbf{A}$ is singular. The Vandermonde determinant $\det(\mathbf{A})$ is well known to be non-zero unless $a_i = a_j$ for some $i \neq j$, where the latter happens with probability at most $\frac{1}{2}L(L-1)/|\mathbb{F}|$.

Security against impersonation attack: Consider an arbitrary but fixed $T' \in \mathcal{XT}$. If $T' = \perp$ then $\mathsf{XVer}(K,i,T) = 0$ for any choice of K and i. Else, if $T' \in \mathbb{F}^L$, then the probability (over the uniformly random choice of b) that $p_{T'}(a) = b$ is $1/|\mathbb{F}|$.

Security against substitution attack: Consider an arbitrary $i \in [L]$. For concreteness, but without loss of generality, we may assume $i = L$. We fix arbitrary values for $K_1 = (a_1, b_1), \dots, K_{L-1} = (a_{L-1}, b_{L-1})$. We may assume those a_i's to be pairwise distinct, since otherwise T will be $\perp$ for any choice of K_L and then the probability of finding T' that is accepted by K_L is upper bounded by $\mathsf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k)$. We first slightly modify the computation of T as follows. Instead of setting T to $\perp$ as soon as $\det(\mathbf{A}) = 0$, we distinguish between the case where $\mathbf{A}T = B$ has no solution and where it has multiple solutions for T. In the former case, T is still set to $\perp$, but in the latter, T is chosen uniformly at random from all the solutions. Note that this modification makes the computation of T randomized (at least in general), but the definition of $\mathsf{Adv}^{\mathsf{sub}}_{\mathsf{XAC}}(k)$ still makes sense. This modification changes the value of $\mathsf{Adv}^{\mathsf{sub}}_{\mathsf{XAC}}(k)$ by at most $\varepsilon_{\text{multi}} = \Pr[\mathbf{A}T = B \text{ has multiple solutions}]$, where the probability is over the choice of K_L.

In the following argument, we consider the above modified version of XAC. The probability $\mathsf{Adv}^{\mathsf{sub}}_{\mathsf{XAC}}(k)$ is upper bounded by the corresponding probability conditioned on $T \neq \perp$ plus the probability that $T = \perp$. Since the latter probability equals $\varepsilon_{\text{no}} = \Pr[\mathbf{A}T = B \text{ has no solution}]$, we can focus on the former while book-keeping the "error" accumulated so far: $\varepsilon_{\text{multi}} + \varepsilon_{\text{no}} = \Pr[\det(\mathbf{A}) = 0] = \Pr[a_L \in \{a_1, \dots, a_{L-1}\}] = (L-1)/|\mathbb{F}|$. In the following argument, we consider an arbitrary $T \neq \perp$, and we consider the corresponding (conditional) probability distribution of K_L. It holds that

$K_L = (a_L, b_L)$ is uniformly distributed in $\mathbb{F}^2$ subject to $p_T(a_L) = b_L$. This in particular implies that a_L on its own is uniformly distributed. Consider now an arbitrary choice for $T' \in \mathcal{XT}$ (computed from $K_1, \ldots, K_{L-1}$ and T). T' is required to be different from T, and we may assume that $T' \neq \bot$, since otherwise $\mathsf{XVer}(K, L, T) = 0$ holds with certainty. By linearity, $p_{T'}(a_L) = b_L$ holds exactly if $p_{T'-T}(a_L) = 0$. However, by the Schwartz-Zippel lemma, $p_{T'-T}(a_L) = 0$ holds with probability at most $\deg(p_{T'-T}(x))/|\mathbb{F}| \leq (L-1)/|\mathbb{F}|$ for a uniformly random $a_L \in \mathbb{F}$. Taking into account $\varepsilon_{\text{multi}}$ and ε_{no} from further up, this proves the claim.

6 Our NC-CCA Secure Scheme

Ingredients. For our encryption scheme with message space $\{0,1\}^L$, we need the following.

1. A hard subset membership problem SMP with sparse language $\mathcal{L}_\rho$ and explainable ciphertexts $\mathcal{X}_\rho$.
2. A 2-universal extended hash proof system EHPS for SMP with tags $\mathcal{T}_\rho$ and explainable keys $\mathcal{K}_\rho$.
3. A collision-resistant hash function $\mathcal{H}$ with domain $(\mathcal{X}_\rho)^L$ and range $\mathcal{T}_\rho$.
4. An L-cross-authentication code XAC with key space $\mathcal{XK} = \mathcal{K}_\rho$ and tag space $\mathcal{XT}$.

From the remarks after Definition 10 and 11 it follows that the efficient samplability and explainability of $\mathcal{K}_\rho$ and the requirement on $\mathcal{XK}$ to coincide with $\mathcal{K}_\rho$ pose no real restriction. In fact, all of these ingredients exist under standard number-theoretic assumptions such as decisional Diffie-Hellman (DDH), decisional composite residuosity (DCR), and quadratic residuosity (QR).

The scheme. We define our encryption scheme $\mathsf{NCCCA} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as follows:

$\mathsf{Gen}(1^k)$**.** Run $\rho \leftarrow \mathsf{SysGen}(1^k)$, $(hpk, hsk) \leftarrow \mathsf{HashGen}(\rho)$ and $\mathsf{H} \leftarrow \mathcal{H}$. Return public key $pk = (hpk, \mathsf{H})$ and secret key $sk = (hsk, \mathsf{H})$.

$\mathsf{Enc}(pk, M; R)$**.** Parse $pk = (hpk, \mathsf{H})$, $M = (M_1, \ldots, M_L) \in \{0,1\}^L$, and $R = (W_i, R_i^X, R_i^K)_{i \in [L]} \in (\mathcal{R}_{\mathsf{SampleL}} \times \mathcal{R}_{\mathsf{Sample}} \times \mathcal{R}_{\mathsf{Sample}})^L$. For $i \in [L]$, set

$$X_i := \begin{cases} \mathsf{Sample}(\mathcal{X}_\rho; R_i^X) \in \mathcal{X}_\rho & \text{if } M_i = 0 \\ \mathsf{SampleL}(\mathcal{L}_\rho; W_i) \in \mathcal{L}_\rho & \text{if } M_i = 1 \end{cases},$$

and compute $t := \mathsf{H}(X_1, \ldots, X_L)$. Then, for $i \in [L]$, set the keys

$$K_i := \begin{cases} \mathsf{Sample}(\mathcal{K}_\rho; R_i^K) & \text{if } M_i = 0 \\ \mathsf{PEval}(hpk, X_i, W_i, t) & \text{if } M_i = 1 \end{cases}$$

and compute the tag $T := \mathsf{XAuth}(K_1, \ldots, K_L)$. Return $C = (X_1, \ldots, X_L, T)$.

$\mathsf{Dec}(sk, C)$**.** Parse $sk = (hsk, \mathsf{H})$ and $C = (X_1, \ldots, X_L, T) \in \mathcal{X}_\rho^L \times \mathcal{XT}$. Set $t := \mathsf{H}(X_1, \ldots, X_L)$. For $i \in [L]$, let $\overline{K}_i := hsk(X_i, t)$, and $M_i := \mathsf{XVer}(\overline{K}_i, i, T)$. Return $M := (M_1, \ldots, M_L)$.

Lemma 3 (Correctness of NCCCA**).** *For any pk in the range of* Gen*, any M, and any $C \leftarrow \mathsf{Enc}(pk, M)$, we have $\mathsf{Dec}(sk, C) = M$ except with probability at most $L \cdot \max\{\mathsf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k), \mathsf{fail}_{\mathsf{XAC}}(k)\}$.*

Proof. If $M_i = 1$, then $\overline{K}_i = hsk(X_i, t) = \mathsf{PEval}(hpk, X_i, W_i, t) = K_i$ by completeness of EHPS, and so $\mathsf{XVer}(\overline{K}_i, i, T) = 1$ except with probability $\mathsf{fail}_{\mathsf{XAC}}(k)$ by correctness of XAC. On the other hand, for $M_i = 0$, EHPS's universality implies that $\overline{K}_i = hsk(X^f, t)$ is uniformly random, even given pk, C, and M. Hence, the probability that $\mathsf{XVer}(\overline{K}_i, i, T) = 1$ is at most $\mathsf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}$(k). The statement follows by a union bound over $i \in [L]$.

Equivocable ciphertexts. As with our earlier scheme NCCPA, NCCCA has the property that 1-encryptions are equivocable. Specifically, we can construct a NC-CCA simulator S that proceeds as follows:

- On input $(\mathtt{sim}, pk, 1^L)$ where pk is the public key (hpk, H), it generates an *equivocable* ciphertext of the form

$$C = (X'_1, \ldots, X'_L, T) = (\mathsf{SampleL}(\mathcal{L}_\rho; W'_1), \ldots, \mathsf{SampleL}(\mathcal{L}_\rho; W'_L), T) \quad (4)$$

  for uniformly chosen $W'_i \in \mathcal{R}_{\mathsf{SampleL}}$ and $T := \mathsf{XAuth}(K'_1, \ldots, K'_L)$ with $K'_i := \mathsf{PEval}(hpk, X_i, W_i, t)$.
- On input $(\mathtt{open}, M)$ for an arbitrary $M \in \{0,1\}^L$, such a C can be explained as an encryption of M by releasing $R = \big(W_i, R^X_i, R^K_i\big)_{i \in [L]}$ with $W_i = W'_i$ if $M_i = 1$, and $(R^X_i, R^K_i) = (\mathsf{Explain}(\mathcal{X}_\rho, X'_i), \mathsf{Explain}(\mathcal{K}_\rho, K'_i))$ if $M_i = 0$.

Our security proof shows that equivocated ciphertexts and their openings are indistinguishable from authentic ones, even given a decryption oracle.

7 Security Analysis

Theorem 3 (NCCCA **is NC-CCA secure).** *There exists a simulator S such that for every adversary A there exists a subset membership distinguisher D and an adversary B on* H*'s collision resistance property such that $\mathsf{time}_D, \mathsf{time}_B \approx \mathsf{time}_A$ and*

$$\left|\mathsf{Adv}^{\mathsf{cca\text{-}nc}}_{\mathsf{NCCCA},A,S}(k)\right| \leq L \cdot \left|\mathsf{Adv}^{\mathsf{sm}}_{\mathsf{EHPS},D}(k)\right| + 2L^2 q \cdot \mathsf{Adv}^{\mathsf{xac}}_{\mathsf{XAC}}(k) + \mathsf{Adv}^{\mathsf{cr}}_{\mathsf{H},B}(k) + \frac{L(L-1)}{|\mathcal{L}_\rho|}, \quad (5)$$

where $\mathsf{Adv}^{\mathsf{xac}}_{\mathsf{XAC}}(k) = \max\big\{\mathsf{Adv}^{\mathsf{sub}}_{\mathsf{XAC}}(k), \mathsf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k)\big\}$ and q is an upper bound on the number of decryption queries A performs.

Before going into the formal proof below, we briefly give a high-level description of the reasoning. The goal is to replace the challenge ciphertext by an equivocable ciphertext. We replace the challenge ciphertexts as follows, one-by-one for every X^*_m (that is not already in $\mathcal{L}_\rho$) within every challenge ciphertext C^*. First, instead of choosing the corresponding key K^*_m at random whenever $M_m = 0$, K^*_m is always computed as HPS key $K^*_m = hsk(X^*_m, t^*)$. Next, $X^*_m \notin \mathcal{L}_\rho$ is replaced by $X^*_m \in \mathcal{L}_\rho$, yielding an equivocable ciphertext.

We now briefly argue why these modifications do not (significantly) alter the adversary A's view. In order to argue that the modification to the choice of K^*_m does not change A's view, it is crucial that A has no information on the HPS secret key hsk beyond the public key hpk. In order to guarantee this, we first slightly modify the decryption procedure Dec used to answer the decryption queries so that Dec does not make any use of hsk: rather than verifying the XAC tag T_i, the decrypted message bit M_i is directly set to 0 whenever $X_i \notin \mathcal{L}_\rho$. By universality of the hash proof system and the security of XAC against impersonation attacks, it follows that this modification does not significantly change A's view. Note that with this modified decryption procedure, the resulting game is not efficient anymore, but this fine for arguing that choosing K^*_m as HPS key instead of random does not change A's view, since this is an information-theoretic argument. However, this step would be a problem for justifying the switch from $X^*_m \notin \mathcal{L}_\rho$ to $X^*_m \in \mathcal{L}_\rho$. Therefore, before doing the latter switch, the modified decryption procedure is replaced again by the original procedure Dec. Again, this change to the decryption procedure can be argued to have little effect on A's view by universality of the hash proof system and security of XAC. However, in this case things are slightly more subtle because if $X_i = X^*_m$ and $t = t^*$, then A now knows an XAC tag that is verified by the HPS key $K_i = hsk(X_i, t)$, namely T^*. But if indeed $t = t^*$ then the collision resistance of H ensures that A has to submit a *different* XAC tag. Hence security against substitution attacks of XAC ensures that the tag will be rejected. Thus both decryption processes decrypt to the same message bit and are hence indistinguishable.

Proof. We proceed in a series of games. Generally, we will denote the output of Game i by out_i.

Game -2 is the original real experiment $\mathsf{Exp}^{\mathsf{cca\text{-}nc\text{-}real}}_{\mathsf{NCCCA},A}$. By definition,

$$\Pr[out_{-2} = 1] = \mathsf{Exp}^{\mathsf{cca\text{-}nc\text{-}real}}_{\mathsf{NCCCA},A}(k). \tag{6}$$

Let $M^* = (M^*_1, \ldots, M^*_\ell)$ denote the message chosen by A; C^* be the challenge ciphertext handed to A; and C^j be A's j-th decryption query. Write $C^* = (X^*_1, \ldots, X^*_L, T^*)$, $C^j = (X^j_1, \ldots, X^j_L, T^j)$, and similarly for the variables t^*, K^j_i, etc. Without loss of generality, we assume that A always makes $q = q(k)$ decryption queries.

In **Game** -1, we abort the experiment (with output 1) as soon as $X^*_i = X^*_{i'}$ for some distinct $i, i' \in [L]$. A counting argument and a union bound show

$$|\Pr[out_{-1} = 1] - \Pr[out_{-2} = 1]| \leq \frac{L(L-1)}{|\mathcal{L}_\rho|}. \tag{7}$$

In **Game** 0, we abort the experiment (with output 1) as soon as A submits a decryption query C^j with

$$t^j = \mathsf{H}(X^j_1, \ldots, X^j_L) = (X^*_1, \ldots, X^*_L) = t^*$$

for some ℓ. A straightforward reduction shows that

$$\Pr[out_0 = 1] - \Pr[out_{-1} = 1] = \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},B}(k) \tag{8}$$

for a suitable B that simulates Game 0.[6]

[6] If H is only *target* collision resistant, a reduction with a multiplicative loss of q can be conducted.

From **Game** 0 **up to Game** L, we will stepwise replace the challenge ciphertext C^* with an equivocable ciphertext of the form (4). Specifically, **Game** m with $0 \leq m \leq L$ coincides with **Game** 0 except that X_i^* and K_i^* with $i \leq m$ are computed as $X_i^* := \mathsf{SampleL}(\mathcal{L}_\rho; W_i^*) \in \mathcal{L}_\rho$ and $K_i^* := \mathsf{PEval}(hpk, X_i^*, W_i^*, t)$, no matter what M_i^* is, and X_i^* is opened suitably to M_i^* as explained at the end of Section 6. Looking ahead, we point out that the final **Game** L, in which C^* is equivocable, is identical to the ideal experiment $\mathsf{Exp}^{\mathsf{cca\text{-}nc\text{-}ideal}}_{\mathsf{NCCCA},S}$ for the simulator S described earlier. We now show indistinguishability between **Games** m **and** $m+1$ for any $0 \leq m \leq L-1$. We do this in several steps.

Game $m.1$ is identical to **Game** m above.

In **Game** $m.2$, we slightly modify the decryption oracle. Recall that from each EHPS ciphertext X_i of a decryption query C, a key $\overline{K}_i = hsk(X_i, t)$ is computed and $M_i := \mathsf{XVer}(\overline{K}_i, i, T)$ is returned. We change this to $M_i := 0$ iff X_i is inconsistent in the sense of $X_i \notin \mathcal{L}_\rho$. (Note that this makes Game $m.2$ inefficient.)

Let $\mathsf{bad}_{m.i.1}$ denote the event that in Game $m.1$, there is a EHPS ciphertext X_i in some C^j that is inconsistent in the sense $X_i \notin \mathcal{L}_\rho$, but $\mathsf{XVer}(\overline{K}_i, i, T) = 1$. Let $\mathsf{bad}_{m.2}$ be the corresponding event in Game $m.2$. By construction, it holds that Game $m.1$ and Game $m.2$ are identical as long as the respective events $\mathsf{bad}_{m.1}$ and $\mathsf{bad}_{m.2}$ do not occur, and $\Pr[\mathsf{bad}_{m.1}] = \Pr[\mathsf{bad}_{m.2}]$. We postpone the proof of the following claim:

Lemma 4. $\Pr[\mathsf{bad}_{m.2}] \leq Lq \cdot \mathsf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k)$.

It follows that

$$|\Pr[out_{m.2} = 1] - \Pr[out_{m.1} = 1]| \leq \Pr[\mathsf{bad}_{m.2}] \leq Lq \cdot \mathsf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k)\,. \tag{9}$$

Note that the adversary's view in Game $m.2$ depends only on hpk. Namely, while the experiment uses hsk to decrypt consistent EHPS ciphertexts efficiently, by completeness of EHPS, this does not release any information on hsk beyond hpk.

In **Game** $m.3$, instead of choosing $K_m^* \in \mathcal{K}_\rho$ uniformly, using Sample, if $M_m^* = 0$, we compute

$$K_m^* := hsk(X_m^*, t^*)$$

as in a hypothetical decryption of C^*. (Later, if C^* is to be opened, K_m^* is explained as being randomly through $\mathsf{Sample}(\mathcal{K}_\rho)$, using coins $\mathsf{Explain}(\mathcal{K}_\rho, K_m^*)$.) Since the only information about *hsk beyond hpk* is released while computing K_m^*, the universality of EHPS guarantees that K_m^* looks uniform. Concretely,

$$\Pr[out_{m.3} = 1] = \Pr[out_{m.2} = 1]\,. \tag{10}$$

In **Game** $m.4$, we reverse the changes from Game $m.2$. That is, decryption does not set $M_i := 1$ iff $X_i \in \mathcal{L}_\rho$, but again computes $M_i := \mathsf{XVer}(\overline{K}_i, i, T)$. Note that this makes Game $m.4$ efficient again.

Let $\mathsf{bad}_{m.3}$ denote the event that in Game $m.3$, there is a EHPS ciphertext X_i in some C^j that is inconsistent in the sense $X_i \notin \mathcal{L}_\rho$, but $\mathsf{XVer}(\overline{K}_i, i, T) = 1$. Let $\mathsf{bad}_{m.4}$ be the corresponding event in Game $m.4$. Similar to above, it holds that Game $m.3$ and Game $m.4$ are identical as long as the respective events $\mathsf{bad}_{m.3}$ and $\mathsf{bad}_{m.4}$ do not occur, and $\Pr[\mathsf{bad}_{m.3}] = \Pr[\mathsf{bad}_{m.4}]$. We postpone the proof of the following claim:

Lemma 5. $\Pr[\mathsf{bad}_{m.3}] \le Lq \cdot \max\{\mathsf{Adv}^{\mathsf{sub}}_{\mathsf{XAC}}(k), \mathsf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k)\}$.

Writing $\mathsf{Adv}^{\mathsf{xac}}_{\mathsf{XAC}}(k) := \max\{\mathsf{Adv}^{\mathsf{sub}}_{\mathsf{XAC}}(k), \mathsf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k)\}$, it follows that

$$|\Pr[out_{m.4} = 1] - \Pr[out_{m.3} = 1]| \le \Pr[\mathsf{bad}_{m.3}] \le Lq \cdot \mathsf{Adv}^{\mathsf{xac}}_{\mathsf{XAC}}(k). \tag{11}$$

In **Game** $m.5$, we do not sample a random $X^*_m \leftarrow \mathcal{X}_\rho$ if $M^*_m = 0$, but instead a consistent $X^*_m \in \mathcal{L}_\rho$. Concretely, the experiment always runs $X^*_m \leftarrow \mathsf{SampleL}(\mathcal{L}_\rho; W^*_m)$. (Later, if C^* is to be opened, X^*_m is explained as being randomly through $\mathsf{Sample}(\mathcal{X}_\rho)$, using random coins $\mathsf{Explain}(\mathcal{X}_\rho, X^*_m)$.) Since Game $m.4$ is again efficient , we can use the subset membership assumption to obtain

$$\frac{1}{L}\sum_{m\in[L]} (\Pr[out_{m.5} = 1] - \Pr[out_{m.4} = 1]) = \mathsf{Adv}^{\mathsf{sm}}_{\mathsf{EHPS},D}(k) \tag{12}$$

for a suitable D that guesses m uniformly and simulates Game $m.4$, resp. Game $m.5$, (implicitly) depending on its challenge.

Because $K^*_m = hsk(X^*_m, t^*) = \mathsf{PEval}(hpk, C^*_m, W^*_m, t^*)$ in Game $m.5$, Game $m.5$ is nothing but a reformulation of Game $m + 1$. Hence, summing up (9,10,11,12) over $m \in [L]$ yields

$$|\Pr[out_L = 1] - \Pr[out_0 = 1]| \le L \cdot \left|\mathsf{Adv}^{\mathsf{sm}}_{\mathsf{EHPS},D}(k)\right| + 2L^2 q \cdot \mathsf{Adv}^{\mathsf{xac}}_{\mathsf{XAC}}(k). \tag{13}$$

It is left to observe that in **Game** L, the experiment is exactly that of $\mathsf{Exp}^{\mathsf{so\text{-}ideal}}_{\mathsf{NCCCA},S}$ for the NC-CCA simulator S described earlier. Therefore,

$$\Pr\left[\mathsf{Exp}^{\mathsf{so\text{-}ideal}}_{\mathsf{NCCCA},S}(k) = 1\right] = \Pr[out_L = 1]. \tag{14}$$

Combining (6,7,8,13,14) finishes the proof.

We catch up with the proofs of the two technical lemmas:

Proof (of Lemma 4). Let $\mathsf{bad}_{m.2.j.i}$ denote the event that in Game $m.2$, the EHPS ciphertext X_i in some C^j is inconsistent in the sense $X_i \notin \mathcal{L}_\rho$, but $\mathsf{XVer}(\overline{K}^j_i, i, T^j) = 1$. Note $\mathsf{bad}_{m.2} = \bigvee_{(j,i)\in[q]\times[L]} \mathsf{bad}_{m.2.j.i}$.

Fix $(j, i) \in [q] \times [L]$. If $X^j_i \notin \mathcal{L}_\rho$, universality of EHPS implies that $\overline{K}^j_i = hsk(X^j_i, t^j)$ is uniformly random and independent of A's view. (Recall that A's view in Game $m.2$ depends only on hpk.) Hence

$$\Pr[\mathsf{bad}_{m.2.j.i}] \le \mathsf{Adv}^{\mathsf{imp}}_{\mathsf{XAC}}(k),$$

and a union bound over j and i shows the claim.

Proof (of Lemma 5). Let $\mathsf{bad}_{m.3.j.i}$ denote the event that in Game $m.3$, the EHPS ciphertext X_i in some C^j is inconsistent in the sense $X_i \notin \mathcal{L}_\rho$, but $\mathsf{XVer}(\overline{K}^j_i, i, T^j) = 1$. Note $\mathsf{bad}_{m.3} = \bigvee_{(j,i)\in[q]\times[L]} \mathsf{bad}_{m.3.j.i}$.

Fix $(j, i) \in [q] \times [L]$. We may assume that $X^j_i \notin \mathcal{L}_\rho$ (as necessary for $\mathsf{bad}_{m.3.j.i}$). Suppose first that $(X^j_i, t^j) \ne (X^*_m, t^*)$. Recall that A's information on hsk in Game $m.3$ is restricted to hpk and $K^*_m = hsk(X^*_m, t^*)$. Thus, EHPS's 2-universality implies that $\overline{K}_i^j = hsk(X_i^j, t^j)$ is uniformly random and independent of A's view. By XAC's security against impersonation attacks,

$$\Pr\left[\mathsf{bad}_{m.3.j.i} \mid (X_i^j, t^j) \neq (X_m^*, t^*)\right] \leq \mathsf{Adv}_{\mathsf{XAC}}^{\mathsf{imp}}(k). \tag{15}$$

Now suppose $(X_i^j, t^j) = (X_m^*, t^*)$. By our changes in Games 0 and 1, we may assume that $(X_{i'}^j)_{i' \in [L]} = (X_{i'}^*)_{i' \in [L]}$, so that necessarily $m = i$ and $\overline{K}_i^j = K_i^* = hsk(X_i^*, t^*)$. Furthermore, for the decryption query to be valid, $T^j \neq T^*$ has to hold. EHPS's universality implies that $K_i^* = hsk(X_i^*, t^*)$ is uniformly distributed and the only information A has on K_i^* is T^*. By XAC's security against substitution attacks,

$$\Pr\left[\mathsf{bad}_{m.3.j.m} \mid (X_i^j, t^j) = (X_m^*, t^*)\right] \leq \mathsf{Adv}_{\mathsf{XAC}}^{\mathsf{sub}}(k). \tag{16}$$

A union bound and summing up (15,16) shows the inequality part of the lemma. The equality part follows by noting $\Pr[\mathsf{bad}_{m.3}] = \Pr[\mathsf{bad}_{m.4}]$, as in the proof of Lemma 4.

Acknowledgements. This work was initiated at the Securing Cyberspace Reunion Conference in 2009; we thank IPAM and the conference organizers for their hospitality. We are also grateful to the anonymous referees for helpful comments.

Note: 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.

References

[1] Beaver, D., Haber, S.: Cryptographic protocols provably secure against dynamic adversaries. In: Rueppel, R.A. (ed.) EUROCRYPT 1992. LNCS, vol. 658, pp. 307–323. Springer, Heidelberg (1992)

[2] 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)

[3] Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (2009)

[4] Bellare, M., Waters, B., Yilek, S.: Identity-based encryption secure under selective opening attack (manuscript, 2010)

[5] Bellare, M., Yilek, S.: Encryption Schemes Secure under Selective Opening Attack. Cryptology ePrint Archive, Report 2009/101 (2009)

[6] Blum, M., Micali, S.: How to generate cryptographically strong sequences of pseudorandom bits. SIAM Journal on Computing 13(4), 850–864 (1984)

[7] Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)

[8] Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM Journal on Computing 36(5), 915–942 (2006)

[9] Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: 42nd FOCS, October 2001, pp. 136–145. IEEE Computer Society Press, Los Alamitos (2001)

[10] Canetti, R., Feige, U., Goldreich, O., Naor, M.: Adaptively secure multi-party computation. In: 28th ACM STOC, pp. 639–648. ACM Press, New York (1996)

[11] 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)

[12] Canetti, R., Krawczyk, H., Nielsen, J.B.: Relaxing chosen-ciphertext security. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 565–582. Springer, Heidelberg (2003)

[13] 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)

[14] 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)

[15] Damgård, I., Nielsen, J.B.: Improved non-committing encryption schemes based on general complexity assumptions. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 432–450. Springer, Heidelberg (2000)

[16] Dolev, D., Dwork, C., Naor, M.: Nonmalleable cryptography. SIAM Journal on Computing 30(2), 391–437 (2000)

[17] Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.: Magic functions. Journal of the ACM 50(6), 852–921 (2003)

[18] Goldreich, O.: Foundations of Cryptography: Basic Applications, vol. 2. Cambridge University Press, Cambridge (2004)

[19] Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)

[20] Hemenway, B., Libert, B., Ostrovsky, R., Vergnaud, D.: Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. Cryptology ePrint Archive, Report 2009/088 (2009)

[21] Hofheinz, D.: Possibility and impossibility results for selective decommitments. IACR ePrint Archive (2008)

[22] 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)

[23] 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)

[24] Katz, J., Ostrovsky, R.: Round-optimal secure two-party computation. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 335–354. Springer, Heidelberg (2004)

[25] 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)

[26] 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)

[27] Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: 22nd STOC. ACM Press, New York (1990)

[28] Nielsen, J.B.: Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 111–126. Springer, Heidelberg (2002)

[29] Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: 40th STOC, pp. 187–196. ACM Press, New York (2008)

[30] 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)

Cryptographic Agility and Its Relation to Circular Encryption

Tolga Acar[1], Mira Belenkiy[1], Mihir Bellare[2], and David Cash[2]

[1] eXtreme Computing Group, Microsoft Research, Redmond, WA
{tolga,mibelenk}@microsoft.com
http://research.microsoft.com/en-us/people/tolga
[2] Department of Computer Science & Engineering,
University of California San Diego, CA
{mihir,cdc}@cs.ucsd.edu
http://www.cs.ucsd.edu/users/{mihir,cdc}

Abstract. We initiate a provable-security treatment of cryptographic *agility*. A primitive (for example PRFs, authenticated encryption schemes or digital signatures) is agile when multiple, individually secure schemes can securely share the same key. We provide a surprising connection between two seemingly unrelated but challenging questions. The first, new to this paper, is whether wPRFs (weak-PRFs) are agile. The second, already posed several times in the literature, is whether every secure (IND-R) encryption scheme is secure when encrypting cycles. We resolve the second question in the negative and thereby the first as well. We go on to provide a comprehensive treatment of agility, with definitions for various different primitives. We explain the practical motivations for agility. We provide foundational results that show to what extent it is achievable and practical constructions to achieve it to the best extent possible. On the theoretical side our work uncovers new notions and relations and settles stated open questions, and on the practical side it serves to guide developers.

1 Introduction

This paper initiates a provable-security treatment of cryptographic *agility*. Agility considers a set of schemes, all meeting some base notion of security, and requires that security is maintained when multiple schemes use the same key. Agility may be considered for any cryptographic primitive: PRFs, authenticated symmetric encryption, collision-resistant hashing, IND-CCA public-key encryption, whatever. To illustrate let us jump right to the example where we have the most interesting results. Then we will back up and discuss motivation and other results.

Are wPRFs agile? Let F be a family of functions and consider a game that picks a random key K and challenge bit b and gives the adversary an oracle $\mathbf{Fn}$ that takes no inputs. Each time it is called, $\mathbf{Fn}$ picks random x, y, returning $(x, F_K(x))$ if $b = 1$ and (x, y) otherwise. Naor and Reingold [24] call F a weak-PRF (wPRF) if the adversary can't guess b.

Why this notion? Being a wPRF is, in practice, a much weaker assumption on a blockcipher than the usual PRF or PRP one. Yet powerful results by Naor

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 403–422, 2010.

and Reingold [24], Maurer and Sjödin [21] and Maurer and Tessaro [22] show that symmetric cryptography can be efficiently and securely based on wPRFs.

Letting F^1, F^2 be wPRFs having keys of the same length, consider a game that picks a *single* random key K and challenge bit b and gives the adversary an oracle **Fn** that, on input $i \in \{1,2\}$, picks random x, y, returning $(x, F^i_K(x))$ if $b = 1$ and (x, y) otherwise. It's just like the previous game, but with two function families, and *both are being evaluated with the same key.* We say that the pair $\{F^1, F^2\}$ is agile if an adversary can't guess b in the above game. Now consider the following statement or conjecture:

wPRF-A : <u>Every</u> pair $\{F^1, F^2\}$ of wPRFs is agile.

Is the statement true? Our first guess was yes, because the randomness of the inputs means the two functions are unlikely to ever be evaluated on the same point, and then it is hard to see what harm there is in their using the same key. But attempts to prove this failed. It is unclear how to reduce the agility of $\{F^1, F^2\}$ to their individual, assumed wPRF securities because reduction-based proof methods break down totally when the key is the same for both functions. Does that mean the statement is false? To demonstrate that, we need a counter-example, meaning specific families F^1, F^2 that (under some assumption) are wPRFs but we have an attack showing $\{F^1, F^2\}$ is not agile. However, an example is not immediate, again due to the fact that the attacker has no control on the inputs to the functions, these being chosen at random by the game.

We clarify that the question is *not* whether there *exists* a pair $\{F^1, F^2\}$ that is agile. We are not asking for a construction of F^1, F^2 that can securely share a key. Indeed, such a construction is trivial: just let F be a wPRF and let $F^1 = F^2 = F$. The question is whether *all* pairs $\{F^0, F^1\}$ of wPRFs are agile.

We have still to motivate *why* we should care whether security is maintained when two schemes use the same key, a practice that cryptographers would typically frown upon. But we will soon explain important practical reasons for this concern. Furthermore, our focus on wPRFs is not arbitrary. We will see that wPRFs are "agility catalysts" in the sense that if they are agile then we can make a host of other primitives agile as well. So the above question —is **wPRF-A** true or not— is central.

We find it intriguing that so basic and simply stated a question is hard to answer. We will obtain the answer by turning to something that seems different but eventually isn't.

Are IND-R encryption schemes CYC-secure? IND-R (INDistinguishability from Random) [26] is a strong notion of CPA-privacy for symmetric encryption schemes that is met by common blockcipher modes of operation (CBC, CTR). It implies IND-CPA. CYC asks for privacy when encrypting "cycles" of the form $\mathcal{E}(K_1, K_2), \mathcal{E}(K_2, K_1)$. Cyclic security was introduced by Camenisch and Lysyanskaya [13] and is of interest as a simple and basic instance of the security of encrypting key-dependent messages concurrently considered by Black, Rogaway and Shrimpton [10]. Now consider the following statement or conjecture:

IND-is-CYC : EVERY IND-R SYMMETRIC ENCRYPTION SCHEME IS CYC-SECURE.

Broadly speaking, this asks whether "normal" security implies security when encrypting cycles. In their work presenting a particular, public-key encryption scheme shown to securely encrypt cycles if the DDH assumption holds, Boneh, Halevi, Hamburg and Ostrovsky [11] explicitly ask, and leave open, the above question. (Up to details of the definitions.) Black, Rogaway and Shrimpton [10] pose it too. Haitner and Holenstein's black-box separations for key-dependent message security [16] only consider stronger forms of security, and do not apply to this question.

THE CONNECTION. We have just stated two open problems that on the face of it are quite different. The first is about wPRFs and the second about symmetric encryption, which are different primitives. In the first case, the issue is sharing a key between two schemes. In the second, no key sharing is involved and we refer to standard notions. Yet, we show the two problems are related. Specifically, we show in Theorem 1 that

wPRF-A $\Rightarrow$ **IND-is-CYC**.

That is, if *every* pair $\{F^1, F^2\}$ of wPRFs is agile (can securely share a key) then *every* IND-R symmetric encryption scheme is CYC-secure (can securely encrypt cycles).

SETTLING BOTH QUESTIONS. So far the above is an instance of what Karp called the "If pigs could whistle then horses could fly" approach that aims to understand open questions in cryptography and complexity theory by relating them to each other. As above, this approach can turn up interesting relations between seemingly unrelated problems. But it doesn't settle them. However, in this case, we can go further. We provide in Theorem 2 a direct and explicit counter-example to show that **IND-is-CYC** is false, resolving the above-mentioned open problem. Our **wPRF-A** $\Rightarrow$ **IND-is-CYC** connection then implies that **wPRF-A** is also false, settling the question of whether wPRFs are agile. The counter-example is a symmetric encryption scheme shown to be IND-R under the SXDH assumption of [3] but shown by attack to not be CYC-secure.

This result refuting **IND-is-CYC** is strengthened by the fact that IND-R is a very strong version of (CPA) privacy and our formalization of CYC is a very weak one. (The formal definitions are in the body of the paper.) Thus, even strong "normal" security fails to imply weak security for encrypting cycles. Interest in this question is witnessed by the work of Backes, Pfitzmann and Scedrov [5] who have previously shown that IND-CPA does not imply a formalization CYC-BPS of cyclic security. However, their counter-example encryption scheme is stateful while ours is stateless, and also IND-R $\Rightarrow$ IND-CPA and CYC-BPS $\Rightarrow$ CYC, making their result weaker than ours.

EXTENSIONS AND IMPLICATIONS. Above we discussed **IND-is-CYC** in the symmetric setting. Our result showing it is false, however, extends to the public-key setting, showing that IND-CPA (semantic security) does not imply security of encrypting cycles, answering the open question of Boneh, Halevi, Hamburg and Ostrovsky [11]. Our result confirms that to achieve circular-security, one needs novel, dedicated schemes and analyses, vindicating work in this line [11,12,2].

Context. Let us now back up to provide some context for agility and describe our other contributions in this area. Cryptographic code usually has a suite of allowed schemes of any particular type. (For example, authenticated encryption.) But new standards or proposed standards appear at a rapid rate. Cryptographic code written today needs to be able to easily incorporate schemes that will appear in the future. This has been recognized and enunciated in developer forums, where the term "agility" has been used to refer to the ability to easily add schemes to an existing suite by structuring code to allow schemes to be substituted in a blackbox manner. The IETF is currently considering adding agility to the widely deployed RADIUS protocol [25]. Resources for software professionals, like a recent Microsoft Developer Network Magazine article [28], encourage agility.

Keys for use with the existing schemes will, however, already have been distributed. Changing them or getting new ones distributed and certified is difficult and error-prone. Agility, thus, would ask that it be possible to maintain the existing key, using this single key with multiple schemes, both new and old. (The presence of new schemes will not preclude use of the old ones. Data encrypted under old schemes and then stored still has to be decrypted, and legacy systems must be supported.)

Agility is of course possible only among schemes that have keys of the same type or length. (An algorithm with a 128-bit key and another with a 256-bit key shouldn't share a key.) But key-compatibility is common given that many schemes will use the same underlying blockciphers or hash functions. Popular proposed or standardized symmetric authenticated encryption (AE) schemes like CCM [29], OCB [26], CWC [20], GCM [23], and EAX [9], for example, all use a 128-bit AES key.

Cryptographers have always recommended key separation, usually interpreted as asking that schemes for different purposes not use the same key. Thus, MAC and symmetric encryption should use different keys, as should public-key encryption and digital signatures. Assuming this type of separation is in place, the issue is whether different schemes for the *same* goal, for example authenticated encryption or PRF, may securely use the same key. This is what agility captures.

We clarify that agility is about *individually secure* schemes sharing a key. It is not about what happens when a scheme is broken and replaced by another that is (hopefully) secure. When that happens, you should not retain the old key since the attacks on the old scheme may already have compromised it.

In their work on chosen-protocol attacks, Kelsey, Schneier and Wagner [19] point both to the danger of using the same key across different schemes and the pressures that are likely to make this happen, the latter including the cost of certification of new keys, the spread of cryptographic APIs, and the limits posed on key-storage by smartcards. They are concerned mostly with different primitives (they call them protocols, for example, encryption and digital signature) sharing a key. Agility can be viewed as a class of chosen-protocol attacks in which the schemes, or protocols, are all for the same goal.

The present paper can serve as a developer guide for agility, pointing out what is possible and what is not. We provide formal definitions that enable a rigorous treatment of agility. With regard to results, on the positive side, we provide practical constructions, showing how to use PRFs and wPRFs as catalysts to confer agility on higher-level primitives like authenticated encryption. On the negative side we show that agility for the full set of schemes meeting some notion is usually unlikely. Let us now expand on all this.

Definitions. Agility is novel, definitionally, in that, unlike standard definitions of security, which apply to individual schemes, agility is a property of a *set* Γ of schemes that individually already meet some base notion of security. Thus, Γ might be the set of all PRFs or some subset thereof. It is as though one moves up one level in “types.”

We appropriately extend the game defining base security so that the key is chosen just once yet an adversary can, via a *scheme argument*, pass in different schemes that will all use this key. The set Γ is said to be a-agile ($a \in \mathbb{N}$) with respect to the base security notion if, for all compatible, size a subsets Π of Γ, the adversary advantage is negligible when its scheme arguments are drawn from Π. (Compatible means the schemes in the set have keys of the same type and length.) In the body of the paper we exemplify with detailed definitions for the case of PRFs, wPRFs and authenticated encryption. In this framework, what we called wPRF agility above is the 2-agility of the set Γ of all wPRFs.

Foundations. The most basic theoretical question is whether a primitive (for example, PRF, AE) is agile, by which we mean that the set of *all* schemes that are individually secure is a-agile for $a \geq 2$. We answer this for a variety of primitives. Fig. 1 summarizes our findings. As it shows, collision-resistant hash functions, when formalized as keyed families, are agile. (Practical functions like MD5, SHA1, SHA256 being unkeyed are trivially agile.) IND-CPA-secure public-key encryption schemes are also agile. So two RSA-based public-key encryption schemes can share the same keys as long as only IND-CPA-security is desired. PRFs, MACs, IND-CPA secure symmetric encryption schemes, AE schemes, IND-CCA-secure public-key encryption schemes and digital signatures are *not* agile. We present counter-examples in the body of the paper for some of these, and others are similar.

Primitive	**Agile?**
PRFs, wPRFs, MACs, IND-CPA symmetric encryption, symmetric authenticated encryption, IND-CCA public-key encryption, digital signatures	No
Collision-resistant hash functions, IND-CPA public-key encryption	Yes

Fig. 1. Agility status of some basic primitives

The above results are relatively straightforward. The most interesting question was whether wPRFs are agile. As discussed at length above, we have answered the question in the negative by first making a connection to cyclic encryption and then answering an open question there.

The following shows why our focus on PRFs and wPRFs is not arbitrary and also shows how, despite the above, to get strong agility in practice.

PRF-derived agility. Our DtE (Derive-then-Encrypt) transform associates to a given PRF ff and a given AE scheme es a new AE scheme $\mathsf{es}_{\mathsf{ff}}$ in which ff under the base key is used to derive a subkey that is then used for es. This turns out to have strong agility properties. Specifically, let Γ be the set of all AE schemes $\mathsf{es}_{\mathsf{ff}}$ as es ranges over *all* AE schemes. Then, for any a, the set Γ is a-agile with respect to AE. The short rendition of this is that AE has now, effectively, become agile. The lack of agility in the primitive itself has been circumvented by using it not directly but within the scope of our construction which can in fact maintain a single key and yet be able to swap in and securely use *any* AE scheme. This is of direct interest in practice where, as we have seen, there are numerous existing and emerging options for AE such as CCM, OCB, CWC, GCM and EAX. Even this (small) set of schemes is probably not agile. But it becomes so when used via our construction.

The above requires that the ff scheme be fixed. But it too is a primitve for which agility may be desirable. If we want to be able to use arbitrary PRFs, the above-noted lack of agility of the primitive means we are out of luck. But in practice these are blockciphers for which there may be only a small set of relevant choices. (For example, all AES finalists.) This set may in fact be agile.

wPRF-derived agility. DtE uses a PRF to make AE agile. Could we use a wPRF instead? This is attractive for two reasons. The first is that a wPRF is a weaker assumption on a blockcipher than a PRF. The second is that, as a result, a set of blockciphers is more likely to be agile with respect to wPRF than to PRF. (We cannot of course hope for agility with respect to all wPRFs since that class is not agile. As above, however, we'd like to get it for as large a subset of the class as possible.) However, the obvious way to extend the construction, namely upon encryption to pick a random R, use $F_K(R)$ as the AE key where K is the base key and F our wPRF, and return R with the ciphertext, fails to achieve AE, even in the absence of agility. What we instead observe is that some of the existing transforms of wPRFs to PRFs from [24,21,22] have a form that make them agility preserving, meaning that if the wPRF is drawn from an agile set then the result is an agile set of PRFs. This yields a construct that is more robust and in practice more agile than DtE but also more expensive.

Related work. Agility is part of the broader issue of the security of key reuse [19]. Haber and Pinkas [15] analyze the security of several specific constructions of public-key encryption and digital signatures when a single public/secret key pair is used both for encryption and signing or for two encryption schemes or digital signature schemes simultaneously. They do not consider the general problem of key reuse and focus on public-key primitives.

Key-dependent message security and its special case, circular security, were defined in concurrent works [10,13] and recent work gives several constructions of various primitives meeting different flavors of security [17,18,11,4,12,2]. At the end of Section 4 we discuss exactly how our counterexample for circular security fits into prior work.

2 Preliminaries

Notation and conventions. If x is a string then $|x|$ denotes its length, and if S is a set then $|S|$ denotes its size. The empty string is denoted ε. If $a = (a_1, \ldots, a_n)$ then $(a_1, \ldots, a_n) \leftarrow a$ means we parse a as shown. Unless otherwise indicated, an algorithm may be randomized. "PT" stands for "polynomial time." By $y \leftarrow A(x_1, x_2, \ldots; r)$ we denote the operation of running A on inputs $x_1, x_2, \ldots$ and coins $r \in \{0,1\}^*$. We denote by $y \stackrel{\$}{\leftarrow} A(x_1, x_2, \ldots)$ the operation of picking r at random and letting $y \leftarrow A(x_1, x_2, \ldots; r)$. (The coins are chosen from a space that may depend on the inputs.) We denote by $[A(x_1, x_2, \ldots)]$ the set of all possible outputs of A on inputs $x_1, x_2, \ldots$.

Games. Our definitions and proofs use the language of code-based games [8]. Recall that a game —look at Fig. 2 for examples— has an (optional) **Initialize** procedure, procedures to respond to adversary oracle queries, and a **Finalize** procedure. A game G is executed with an adversary A as follows. First, **Initialize**(if present) executes, and its outputs are the inputs to A. Then A executes, its oracle queries being answered by the corresponding procedures of G. When A terminates, its output becomes the input to the **Finalize** procedure. The output of the latter, denoted G^A, is called the output of the game, and we let "G^A" denote the event that this game output takes value true. Boolean flags are assumed initialized to false. The running time of an adversary is the worst case time of the execution of the adversary with the game defining its security, so that the execution time of the called game procedures is included.

Function families. The (common) syntax we use for PRFs and wPRFs is more general than may be usual because we will (later) need to consider schemes defined via families of groups. An FF-scheme ("FF" stands for "Function Family") $\mathsf{ff} = (\mathsf{ff.Pg}, \mathsf{ff.Kg}, \mathsf{ff.f}, \mathsf{ff.DomR}, \mathsf{ff.RngR})$ consists of a parameter generator, a key generator, an evaluator, a domain recognizer and a range recognizer, all PT algorithms, the last three deterministic. We require that $\mathsf{ff.f}(\mathit{pars}, K, \cdot) : \mathsf{ff.Dom}(\mathit{pars}) \to \mathsf{ff.Rng}(\mathit{pars})$ for every $k \in \mathbb{N}$, $\mathit{pars} \in [\mathsf{Pg}(1^k)]$ and $K \in [\mathsf{Kg}(\mathit{pars})]$, where $\mathsf{ff.Dom}(\mathit{pars}) = \{x : \mathsf{ff.DomR}(\mathit{pars}, x) = 1\}$ and $\mathsf{ff.Rng}(\mathit{pars}) = \{y : \mathsf{ff.RngR}(\mathit{pars}, y) = 1\}$. We require that one can sample from $\mathsf{ff.Dom}(\mathit{pars})$ and $\mathsf{ff.Rng}(\mathit{pars})$ in PT on input pars. We also require that $|\mathsf{ff.Dom}(\mathit{pars})| \geq 2^k$ for all $\mathit{pars} \in [\mathsf{ff.Pg}(1^k)]$ and all $k \in \mathbb{N}$. (PRFs on tiny domains are trivially constructed and don't even imply one-way functions. This convention rules them out.)

Encryption syntax. Our syntax for encryption is general enough to cover both symmetric and asymmetric encryption, which will save us from repeating

```
proc KeySetup(ff)
    $pars \xleftarrow{\$} \mathsf{ff.Pg}(1^k)$ ; $K \xleftarrow{\$} \mathsf{ff.Kg}(pars)$
    $b \xleftarrow{\$} \{0,1\}$
    Return $pars$
proc Fn(ff, $x$)
    If $\mathsf{ff.DomR}(pars, x) = 0$ then return $\perp$
    If $b = 1$ then $y \leftarrow \mathsf{ff.f}(pars, K, x)$
    Else $y \xleftarrow{\$} \mathsf{ff.Rng}(pars)$
    Return $y$
proc Finalize($b'$)
    Return $(b' = b)$
```

```
proc KeySetup(ff)
    $pars \xleftarrow{\$} \mathsf{ff.Pg}(1^k)$ ; $K \xleftarrow{\$} \mathsf{ff.Kg}(pars)$
    $b \xleftarrow{\$} \{0,1\}$
    Return $pars$
proc Fn(ff)
    $x \xleftarrow{\$} \mathsf{ff.Dom}(pars)$
    If $b = 1$ then $y \leftarrow \mathsf{ff.f}(pars, K, x)$
    Else $y \xleftarrow{\$} \mathsf{ff.Rng}(pars)$
    Return $(x, y)$
proc Finalize($b'$)
    Return $(b' = b)$
```

Fig. 2. Game $\mathsf{FF.PR.Gm}_k$, on the left, and game $\mathsf{FF.wPR.Gm}_k$, on the right, for $k \in \mathbb{N}$

similar security definitions for both cases. A ENC-scheme ("ENC" stands for encryption) $\mathsf{es} = (\mathsf{es.Pg}, \mathsf{es.Kg}, \mathsf{es.Enc}, \mathsf{es.Dec}, \mathsf{es.MsgR}, \mathsf{es.CtxtR})$ consists of 6 PT algorithms: a parameter generator, a key generator, encryption and decryption algorithms, a plaintext recognizer and a ciphertext recognizer. The decryption algorithm is deterministic. On input $pars \in [\mathsf{es.Pg}(1^k)]$, the key generator outputs a triple (ek, dk, pk), where ek is an encryption key, dk is a decryption key and pk is a public key. We say that the encryption scheme is symmetric if it is always the case that $pk = \perp$. (In which case we may assume wlog $ek = dk$.) We say it is asymmetric if it is always the case that $ek = pk$. We require that there is a polynomial $r(\cdot)$, called the *number of coins used by* es.Enc, such that es.Enc draws its coins from $\{0,1\}^{r(k)}$ whenever its first input is $pars \in [\mathsf{es.Pg}(1^k)]$. We require that $\mathsf{es.Enc}(pars, ek, \cdot; R)$: $\mathsf{es.Msg}(pars) \rightarrow \mathsf{es.Ctxts}(pars)$ and $\mathsf{es.Dec}(pars, dk, \mathsf{es.Enc}(pars, ek, M; R)) = M$ for all $R \in \{0,1\}^{r(k)}$, all $M \in \mathsf{es.Msg}(pars)$, all $(ek, dk, pk) \in [\mathsf{es.Kg}(pars)]$, all $pars \in [\mathsf{es.Pg}(1^k)]$ and all $k \in \mathbb{N}$, where $\mathsf{es.Msg}(pars) = \{M : \mathsf{es.MsgR}(pars, M) = 1\}$ and $\mathsf{es.Ctxts}(pars) = \{C : \mathsf{es.CtxtR}(pars, C) = 1\}$. We require that one can sample from $\mathsf{es.Msg}(pars)$ and $\mathsf{es.Ctxts}(pars)$ in PT on input $pars$.

3 Agility Definitions

The notions of security we usually define apply to schemes, saying what it means for the scheme to be secure. (For example, a function family is a PRF if ...). Agility is different. It is not a property of an individual scheme but of a set of schemes relative to some (standard) security notion for these schemes. Thus, we might have a set of PRFs and talk of their agility with respect to the PRF notion.

The template for an agility definition is as follows. We start with a syntax. (For example, FF for function families or ENC for encryption schemes.) We then provide a sequence of games to capture agility with respect to a (usually standard) underlying notion of security for individual schemes. (For example, games

$\mathsf{FF.PR.Gm}_k$, $k \in \mathbb{N}$, on the left side of Fig. 2. The underlying notion here, called PR for pseudorandomness, is the standard PRF notion.) The unusual feature of the games is to have a *scheme argument*, meaning the adversary may provide procedures a scheme (of the syntax being considered) whose algorithms the game then uses. Agility of a set Π of schemes is measured by allowing the adversary to use *different* members of Π (the choice at its discretion) in the role of scheme argument, *with the underlying key remaining the same.* (For this to be possible, Π must be consistent in the sense that all its schemes have keys of the same syntactic form.) Some advantage will be associated to an adversary and Π, and thence we will get a definition of agility for Π. Restricting attention to a set Π consisting of a single scheme (corresponding to an adversary whose scheme argument is always this one scheme) recovers the base underlying security notion (for example, PRF) for this scheme, thereby saving us from defining it separately and also confirming that agility is a natural extension of the base notion.

We could carry through the above in a fully general way, but it is likely to be hard to parse. Instead, we exemplify with agility definitions for three primitives important to this paper, namely PRFs, wPRFs and authenticated encryption. To expose the underlying unity, however, we use a uniform notation, where Sec-security of Syntax-schemes, for example, refers to security of schemes of the shown syntax with regard to the shown base notion of security. We hope the reader will forgive the standard notion of a PRF ending up, for this reason, being called PR-security of an FF-scheme.

PRF Agility. Say a set Π of FF-schemes is *compatible* if all $\mathsf{ff} \in \Pi$ have the same parameter generator and also all $\mathsf{ff} \in \Pi$ have the same key generator. Consider the games $\mathsf{FF.PR.Gm}_k$ ($k \in \mathbb{N}$) on the left side of Fig. 2. Call an adversary A Π-*restricted* if the scheme arguments in its queries are all drawn from Π, it makes only one **KeySetup** query, this being its first oracle query, and it never repeats an oracle query. (All this must hold with probability 1 regardless of how queries are answered. Π being compatible means the parameter and key generation algorithms invoked by **KeySetup** will be the same regardless of the FF-scheme that it is provided as input.) Let $\mathbf{Adv}^{\mathsf{PR}}_{\Pi,A}(k) = 2\Pr[\mathsf{FF.PR.Gm}^A_k] - 1$ for any compatible set Π of FF-schemes and Π-restricted adversary A. ("PR" stands for "pseudorandom".)

We say that a compatible set Π of FF-schemes is *agile with respect to* PR if the function $\mathbf{Adv}^{\mathsf{PR}}_{\Pi,A}(\cdot)$ is negligible for all PT, Π-restricted adversaries A. We say that a (not necessarily compatible) set Γ of FF-schemes is a-*agile with respect to* PR ($a \in \mathbb{N}$) if every size a, compatible subset $\Pi \subseteq \Gamma$ of Γ is agile with respect to PR.

We recover the usual notion of an FF-scheme ff being a PRF —which we call PR-security here for uniformity— as agility of the singleton set $\{\mathsf{ff}\}$ with respect to PR. To spell it out, FF-scheme ff is PR-*secure* if the function $\mathbf{Adv}^{\mathsf{PR}}_{\{\mathsf{ff}\},A}(\cdot)$ is negligible for all PT $\{\mathsf{ff}\}$-restricted adversaries A. Then ff is PR-secure iff it is a PRF, and the set $\mathsf{FF.PR.Sch}$ of all PR-secure FF-schemes is the set of all PRFs.

wPRF agility. The games $\mathsf{FF.wPR.Gm}_k$ ($k \in \mathbb{N}$) are now those on the right side of Fig. 2. Let $\mathbf{Adv}^{\mathsf{wPR}}_{\Pi,A}(k) = 2\Pr[\mathsf{FF.wPR.Gm}^A_k] - 1$ for any compatible set Π of FF-schemes and Π-restricted adversary A. ("wPR" stands for "weakly pseudorandom".) We say that a compatible set $\mathsf{\Pi}$ of FF-schemes is *agile with respect to* wPR if the function $\mathbf{Adv}^{\mathsf{wPR}}_{\mathsf{\Pi},A}(\cdot)$ is negligible for all PT, $\mathsf{\Pi}$-restricted adversaries A. We say that a (not necessarily compatible) set $\mathsf{\Gamma}$ of FF-schemes is *a-agile with respect to* wPR ($a \in \mathbb{N}$) if every size a, compatible subset $\mathsf{\Pi} \subseteq \mathsf{\Gamma}$ of $\mathsf{\Gamma}$ is agile with respect to wPR. As before we recover the usual notion of an FF-scheme ff being a wPRF, which we call wPR-security here, as agility of the singleton set $\{\mathsf{ff}\}$ with respect to wPR, and let $\mathsf{FF.wPR.Sch}$ be the set of all wPR-secure FF schemes.

Agility for authenticated encryption. Early definitions of AE [7] gave separate privacy and integrity requirements. Our agility games $\mathsf{ENC.AuE.Gm}_k$ ($k \in \mathbb{N}$) given in Fig. 3, where $\mathsf{es} = (\mathsf{es.Pg}, \mathsf{es.Kg}, \mathsf{es.Enc}, \mathsf{es.Dec}, \mathsf{es.MsgR}, \mathsf{es.CtxtR})$ is a ENC-scheme, are instead based on a unified definition in the style of Rogaway and Shrimpton [27]. The privacy requirement is indistinguishability from random [26] (IND-R), a strengthening of the usual notion of [6] that tends to be naturally achieved by block cipher modes of operation [6]. The games of course have the scheme argument that is central to agility. The definitions proceed in direct analogy to the above. To detail them, first say a set $\mathsf{\Pi}$ of ENC-schemes is *compatible* if all $\mathsf{es} \in \mathsf{\Pi}$ have the same parameter generator and also all $\mathsf{es} \in \mathsf{\Pi}$ have the same key generator. Call an adversary A $\mathsf{\Pi}$-*restricted* if the scheme arguments in its queries are all drawn from $\mathsf{\Pi}$ and it makes only one **KeySetup** query, this being its first oracle query. Let $\mathbf{Adv}^{\mathsf{AuE}}_{\mathsf{\Pi},A}(k) = 2\Pr[\mathsf{ENC.AuE.Gm}^{A}_{k}] - 1$ for any compatible set $\mathsf{\Pi}$ of ENC-schemes and $\mathsf{\Pi}$-restricted adversary A. ("AuE" stands for "authenticated encryption".) We say that a compatible set $\mathsf{\Pi}$ of ENC-schemes is *agile with respect to* AuE if the function $\mathbf{Adv}^{\mathsf{AuE}}_{\mathsf{\Pi},A}(\cdot)$ is negligible for all PT, $\mathsf{\Pi}$-restricted adversaries A. We say that a (not necessarily compatible) set $\mathsf{\Gamma}$ of ENC-schemes is *a-agile with respect to* AuE ($a \in \mathbb{N}$) if every size a, compatible subset $\mathsf{\Pi} \subseteq \mathsf{\Gamma}$ of $\mathsf{\Gamma}$ is agile with respect to AuE. We recover the usual notion of an ENC-scheme es being an authenticated encryption scheme, which we call AuE-security here, as agility of the singleton set $\{\mathsf{es}\}$ with respect to AuE and let $\mathsf{ENC.AuE.Sch}$ be the set of all AuE-secure ENC schemes.

```
proc KeySetup(es)
  $pars \xleftarrow{\$} \mathsf{es.Pg}(1^k)$ ; $(ek, dk, pk) \xleftarrow{\$} \mathsf{es.Kg}(pars)$
  $S \leftarrow \emptyset$ ; $b \xleftarrow{\$} \{0,1\}$
  Return $(pars, pk)$

proc RoR(es, $M$)
  If $M \notin \mathsf{es.Msg}(pars)$ Then Return $\perp$
  If $b = 1$ Then $C \xleftarrow{\$} \mathsf{es.Enc}(pars, ek, M)$
  Else $C \xleftarrow{\$} \mathsf{es.Ctxts}(pars)$
  $S \leftarrow S \cup \{(\mathsf{es}, C)\}$
  Return $C$

proc Dec(es, $C$)
  If $(\mathsf{es}, C) \in S$ Then Return $\perp$
  If $b = 1$ Then $M \leftarrow \mathsf{es.Dec}(pars, dk, C)$
  Else $M \leftarrow \perp$
  Return $M$

proc Finalize($b'$)
  Return $(b' = b)$
```

Fig. 3. Game $\mathsf{ENC.AuE.Gm}_k$ for $k \in \mathbb{N}$

This definition is for both symmetric and asymmetric schemes even though the latter can only meet it if no **Dec** queries are allowed because the $\mathsf{IND\text{-}R}$ notion of privacy obtained by incorporating the latter restriction will be useful later.

4 Negative Results

We consider the central foundational question about agility, namely whether it can be achieved for the set of *all* secure schemes of a given type. We begin by showing how to rule this out quite simply for PRFs and AE. Similar methods yield negative results for many other primitives, but *not* for wPRFs. We establish the connection between the latter and circular encryption, and then provide our negative result on circular encryption, namely that IND-R does not imply CYC. This will be used to establish non-agility of wPRFs.

Non-agility of PRFs. PRF agility is important because PRFs model blockciphers, for which agility is important in practice, and also (cf. Section 5) because PRFs are "universal" with regard to providing agility in the sense that if a set of PRFs is agile we can use it to build a class of authenticated encryption schemes that is agile with respect to *arbitrary* substitution of the encryption. The following says the set $\mathsf{FF.PR.Sch}$ of all PRFs is not a-agile for $a \geq 2$ under the minimal assumption that PRFs exist.

Proposition 1. *Let $a \geq 2$. If the set $\mathsf{FF.PR.Sch}$ of all PR-secure FF-schemes is not empty then it is not a-agile with respect to PR.*

Proof (Proposition 1). Let $\mathsf{ff} \in \mathsf{FF.PR.Sch}$. We construct a PR-secure scheme $\overline{\mathsf{ff}}$ such that the set $\Pi = \{\mathsf{ff}, \overline{\mathsf{ff}}\}$ is not 2-agile, meaning the two PRFs cannot securely use the same key. The Proposition follows.

For the construction, we assume points in the range of ff are bitstrings. This is wlog since they can always be encoded as such. The parameter generator, key generator and domain recognizer of $\overline{\mathsf{ff}}$ are the same as those of ff. On input pars, K, x, the evaluator $\overline{\mathsf{ff}}.\mathsf{f}$ lets $y \leftarrow \mathsf{ff.f}(\mathit{pars}, K, x)$ and returns the bitwise complement $\overline{y}$ of y. The new FF-scheme has range defined by $\overline{\mathsf{ff}}.\mathsf{Rng}(\mathit{pars}) = \{\, \overline{y} \,:\, y \in \mathsf{ff.Rng}(\mathit{pars}) \,\}$.

It is easy to see that $\overline{\mathsf{ff}}$ is PR-secure (meaning, is a PRF) assuming ff is. The interesting question is what happens when they share a key. Consider the Π-restricted adversary A that on input pars, begins with a $\mathbf{KeySetup}(\mathsf{ff})$ query. Then it lets $x \xleftarrow{\$} \mathsf{ff.Dom}(\mathit{pars})$ and lets $y \leftarrow \mathbf{Fn}(\mathsf{ff}, x)$ and $z \leftarrow \mathbf{Fn}(\overline{\mathsf{ff}}, x)$. (Note the definition of a Π-restricted adversary required it to not repeat an oracle query. This condition is met because $(\mathsf{ff}, x) \neq (\overline{\mathsf{ff}}, x)$.) If $z = \overline{y}$ it outputs 1, else 0.

We assume $|\mathsf{ff.Rng}(\mathit{pars})| \geq 2$. This is wlog because there are standard ways to extend the range of a PRF. Now we claim that $\mathbf{Adv}^{\mathsf{PR}}_{\Pi,A}(\cdot) \geq 1/2$, which shows that Π is not 2-agile as desired. We justify the claim as follows. If $b = 1$ in game $\mathsf{FF.PR.Gm}_k$ then $z = \overline{y}$ and A returns 1. If $b = 0$, it returns 1 with the probability that $z = \overline{y}$ when z is drawn at random from $\overline{\mathsf{ff}}.\mathsf{Rng}(\mathit{pars})$ and y is drawn at random from $\mathsf{ff.Rng}(\mathit{pars})$. But both sets $\mathsf{ff.Rng}(\mathit{pars})$ and $\overline{\mathsf{ff}}.\mathsf{Rng}(\mathit{pars})$ have size at least two, so the probability is at most $1/2$. □

The above says the class $\mathsf{FF.PR.Sch}$ of all PRFs is not a-agile for $a \geq 2$. But it is still possible that some proper subsets Γ of $\mathsf{FF.PR.Sch}$ are a-agile for some $a \geq 2$. This is interesting for practice, where one may be interested in a certain specific and quite small collection of schemes, and is why we defined agility for subsets of $\mathsf{FF.PR.Sch}$ rather than merely for the whole.

Extensions. Similar ideas exclude agility for many other primitives. Let us illustrate by sketching a counterexample to show that $\mathsf{ENC.AuE.Sch}$ is not a-agile with respect to AuE for any $a > 1$. Given $\mathsf{es} \in \mathsf{ENC.AuE.Sch}$ we construct $\overline{\mathsf{es}} \in \mathsf{ENC.AuE.Sch}$ which given $pars, K, M$ lets $C \stackrel{\$}{\leftarrow} \mathsf{es}(pars, K, M)$ and returns $\overline{C}$. We claim $\{\mathsf{es}, \overline{\mathsf{es}}\}$ is not agile. This is because an attacker can query $\mathbf{RoR}(\mathsf{es}, M)$ to get back C and then query $\mathsf{Dec}(\overline{\mathsf{es}}, \overline{C})$ to get back a message that will be M if the challenge bit b was 1 and is unlikely to be M otherwise. However, this type of approach does not work to prove non-agility of wPRFs because the inputs to the functions are not under adversary control. On the other hand, a proof that wPRFs are agile also seems out of reach of reduction-based techniques. We turn to resolving this.

Auxiliary definitions for encryption. We say that ENC-scheme es is $\mathsf{IND\text{-}R}$-secure if $\mathbf{Adv}^{\mathsf{AuE}}_{\{\mathsf{es}\},A}(\cdot)$ is negligible for all PT, $\{\mathsf{es}\}$-restricted adversaries A that make no **Dec** queries. This strong version of privacy under CPA from [26] implies the standard IND-CPA and is achieved by blockcipher modes of operation like CTR and CBC [6].

Say es can *encrypt its own keys* if $dk \in \mathsf{es.Msg}(pars)$ for every $(ek, dk, pk) \in [\mathsf{es.Kg}(pars)]$, every $pars \in [\mathsf{es.Pg}(1^k)]$ and every $k \in \mathbb{N}$. For such an encryption scheme, let $\mathbf{Adv}^{\mathsf{CYC}}_{\mathsf{es},A}(k) = \Pr[\mathsf{ENC.CYC.Gm}^{A}_{k}]$ where the game in question is shown in Fig. 4. Say es is CYC-secure if $\mathbf{Adv}^{\mathsf{CYC}}_{\mathsf{es},A}(\cdot)$ is negligible for all PT A. This asks that es have pseudorandom ciphertexts under a weak type of circular-encryption attack. The adversary is given access to samples, each of which is either a circular encryption of two keys or a pair of random strings, and is challenged to distinguish these two cases. Normal chosen-plaintext queries are not allowed, so CYC-security does not imply IND-CPA. Since our results are negative, this only strengthens them.

The definitions we have just given are for both the symmetric and asymmetric case. We will first be interested in the former.

Relating wPRF agility and encryption security. We show that if every pair of wPRFs is 2-agile, then every $\mathsf{IND\text{-}R}$-secure symmetric ENC-scheme is CYC-secure. We say that an FF-scheme ff has *bit-output* if there is a polynomial $r(k) \geq k$ such that $\mathsf{ff.Rng}(pars) = \{0,1\}^{r(k)}$ for all $pars \in [\mathsf{ff.Pg}(1^k)]$ and all $k \in \mathbb{N}$.

Theorem 1. (**wPRF-A $\Longrightarrow$ IND-is-CYC**) *Suppose the set* $\mathsf{FF.wPR.Sch}$ *of all* wPR*-secure* FF*-schemes is* 2*-agile with respect to* wPR*, and further that* wPR*-secure* FF*-schemes with bit-output exist. Then every* $\mathsf{IND\text{-}R}$*-secure symmetric encryption scheme that can encrypt its own keys is also* CYC*-secure.*

```
proc Initialize
$pars \xleftarrow{\$} \mathsf{es.Pg}(1^k)$ ; $b \xleftarrow{\$} \{0,1\}$
$(ek_1, dk_1, pk_1) \xleftarrow{\$} \mathsf{es.Kg}(pars)$
$(ek_2, dk_2, pk_2) \xleftarrow{\$} \mathsf{es.Kg}(pars)$
Return $(pars, pk_1, pk_2)$

proc Cyc()
If $b = 1$ then
    $C_1 \xleftarrow{\$} \mathsf{es.Enc}(pars, ek_1, dk_2)$
    $C_2 \xleftarrow{\$} \mathsf{es.Enc}(pars, ek_2, dk_1)$
Else
    $C_1 \xleftarrow{\$} \mathsf{es.Ctxts}(pars)$
    $C_2 \xleftarrow{\$} \mathsf{es.Ctxts}(pars)$
Return $(C_1, C_2)$

proc Finalize($b'$)
Return $(b' = b)$
```

```
Algorithm $\mathsf{ff}_i.\mathsf{Pg}(1^k)$    // $i = 1, 2$
$fpars \xleftarrow{\$} \mathsf{ff.Pg}(1^k)$ ; $epars \xleftarrow{\$} \mathsf{es.Pg}(1^k)$
Return $pars \leftarrow (fpars, epars)$

Algorithm $\mathsf{ff}_i.\mathsf{Kg}((fpars, epars))$    // $i = 1, 2$
$L \xleftarrow{\$} \mathsf{ff.Kg}(fpars)$
$(K_1, K_1, \perp) \xleftarrow{\$} \mathsf{es.Kg}(epars)$
$(K_2, K_2, \perp) \xleftarrow{\$} \mathsf{es.Kg}(epars)$
Return $(L, K_1, K_2)$

Algorithm $\mathsf{ff}_1.\mathsf{f}((fpars, epars), (L, K_1, K_2), x)$
$r \leftarrow \mathsf{ff.f}(fpars, L, x)$ ; $y \leftarrow \mathsf{es.Enc}(epars, K_1, K_2; r)$
Return $y$

Algorithm $\mathsf{ff}_2.\mathsf{f}((fpars, epars), (L, K_1, K_2), x)$
$r \leftarrow \mathsf{ff.f}(fpars, L, x)$ ; $y \leftarrow \mathsf{es.Enc}(epars, K_2, K_1; r)$
Return $y$
```

Fig. 4. Game $\mathsf{ENC.CYC.Gm}_k$ on the right, for $k \in \mathbb{N}$. On the right, algorithms for FF-schemes $\mathsf{ff}_1, \mathsf{ff}_2$ of the proof of Theorem 1.

To prove this, we start with an IND-R-secure symmetric ENC-scheme es and then build a pair of wPRFs. Assuming wPRFs are 2-agile, this pair is 2-agile as a special case. We will then prove CYC-security of es based on the 2-agility of the wPRF pair.

Accordingly, let $\mathsf{es} = (\mathsf{es.Pg}, \mathsf{es.Kg}, \mathsf{es.Enc}, \mathsf{es.Dec}, \mathsf{Enc.MsgR}, \mathsf{Enc.CtxtR})$ be a symmetric ENC-scheme, and let $r(\cdot)$ be the number of coins used by es.Enc. Let $\mathsf{ff} = (\mathsf{ff.Pg}, \mathsf{ff.Kg}, \mathsf{ff.f}, \mathsf{ff.DomR}, \mathsf{ff.RngR})$ be a FF-scheme such that $\mathsf{ff.Rng}(pars) = \{0,1\}^{r(k)}$ for all $pars \in [\mathsf{ff.Pg}(1^k)]$ and all $k \in \mathbb{N}$. That an FF-scheme with such range exists follows from the assumption that FF-schemes with bit-output exist, for we can reduce output size by truncation or increase it by application of a PRG.

For $i = 1, 2$ we now define FF-scheme $\mathsf{ff}_i = (\mathsf{ff}_i.\mathsf{Pg}, \mathsf{ff}_i.\mathsf{Kg}, \mathsf{ff}_i.\mathsf{f}, \mathsf{ff}_i.\mathsf{DomR}, \mathsf{ff}_i.\mathsf{RngR})$. The parameter, key-generation and evaluator algorithms are in Fig. 4. Since es is symmetric, we are assuming wlog that the encryption and decryption keys are the same, so that output of the j-th execution of $\mathsf{es.Kg}(epars)$ in the code of $\mathsf{ff}_i.\mathsf{Kg}((fpars, epars))$ has the form $(K_j, K_j, \perp)$ $(j = 1, 2)$. The FF-schemes ff_1 and ff_2 are identical except for how their evaluators compute the output value y, where the roles of K_1 and K_2 are reversed. We let $\mathsf{ff}_1.\mathsf{DomR} = \mathsf{ff}_2.\mathsf{DomR} = \mathsf{ff.DomR}$, and $\mathsf{ff}_1.\mathsf{RngR} = \mathsf{ff}_2.\mathsf{RngR} = \mathsf{es.CtxtR}$. Since $\mathsf{ff}_1, \mathsf{ff}_2$ have the same parameter and key-generation algorithms, $\{\mathsf{ff}_1, \mathsf{ff}_2\}$ is compatible. The following says that each of ff_1 and ff_2, *taken individually*, is wPR-secure.

Lemma 1. *Suppose symmetric* ENC*-scheme* es *is* IND-R*-secure and* FF*-scheme* ff *is* wPR*-secure. Let* $\mathsf{ff}_1, \mathsf{ff}_2$ *be constructed from them as described above. Then* ff_1 *and* ff_2 *are both* wPR*-secure.*

The proof is in [1]. The next lemma says that if $\{\mathsf{ff}_1, \mathsf{ff}_2\}$ is agile with respect to wPR, then es is CYC-secure.

Lemma 2. *Suppose* $\mathsf{ff}_1, \mathsf{ff}_2$ *are constructed as described above from symmetric* ENC*-scheme* es *and* wPR*-secure* FF*-scheme* ff*. Suppose* $\{\mathsf{ff}_1, \mathsf{ff}_2\}$ *is agile with respect to* wPR*. Then* es *is* CYC*-secure.*

Theorem 1 follows from these two lemmas. The proof of Lemma 2 is in [1].

The SXDH assumption. Our counterexample encryption scheme that is $\mathsf{IND\text{-}R}$-secure but not CYC-secure relies on the SXDH assumption [3] which we now formalize. A *group scheme* is a PT algorithm GS that on input 1^k outputs $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2)$, where p is a k-bit prime, $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are descriptions of groups of order p, $\mathbf{e}$: $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is a non-degenerate bilinear map, and g_i is a generator for G_i, $i = 1, 2$. We assume that one can recognize and multiply elements of the groups involved as well evaluate $\mathbf{e}(\cdot, \cdot)$ in time polynomial in k. The Symmetric External Diffie-Hellman (SXDH) assumption [3] is that the Decisional Diffie-Hellman problem is hard in both G_1 and G_2. Formally, let $\mathbf{Adv}^{\mathsf{SXDH}}_{\mathsf{GS},A}(k) = 2\Pr[\mathsf{SXDH}^{A}_{\mathsf{GS},k}] - 1$ where the game is in Fig. 5. The SXDH problem is said to be hard for GS if $\mathbf{Adv}^{\mathsf{SXDH}}_{\mathsf{GS},A}(\cdot)$ is negligible for every PT A. We also assume that a group scheme comes equipped with a PT "key derivation function" H that, for $i = 1, 2$, takes input $\mathit{pars} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2)$ together with i and $Z \in \mathbb{G}_i$, and returns a point $H(\mathit{pars}, i, Z) \in \mathbb{Z}_p$. The only requirement we place on H is that for all $\mathit{pars} \in [\mathsf{GS}(1^k)]$ and both $i = 1, 2$, if Z is uniformly distributed over $\mathbb{G}_i$, then $H(\mathit{pars}, i, Z)$ is uniformly distributed over $\mathbb{Z}_p$. This requirement can be relaxed to allow a negligible deviation from uniform.

IND-R-but-not-IND-CYC encryption schemes. The following says that if SXDH is true then we can build counterexample encryption schemes, both symmetric and asymmetric, which are $\mathsf{IND\text{-}R}$-secure (and hence IND-CPA-secure) but are not CYC-secure.

Theorem 2. (**SXDH** $\Longrightarrow$ NOT **IND-is-CYC**) *Suppose there exists a group scheme in which the SXDH problem is hard. There there exist symmetric and asymmetric* ENC*-schemes which are* $\mathsf{IND\text{-}R}$*-secure but not* CYC*-secure.*

```
proc Initialize
$(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2) \stackrel{\$}{\leftarrow} \mathsf{GS}(1^k)$ ; $x_1, x_2, y_1, y_2, z_1, z_2 \stackrel{\$}{\leftarrow} \mathbb{Z}_p$ ; $b \stackrel{\$}{\leftarrow} \{0, 1\}$
If $(b = 1)$ then $(X_1, Y_1, Z_1, X_2, Y_2, Z_2) \leftarrow (g_1^{x_1}, g_1^{y_1}, g_1^{x_1 y_2}, g_2^{x_2}, g_2^{y_2}, g_2^{x_2 y_2})$
Else $(X_1, Y_1, Z_1, X_2, Y_2, Z_2) \leftarrow (g_1^{x_1}, g_1^{y_1}, g_1^{z_1}, g_2^{x_2}, g_2^{y_2}, g_2^{z_2})$
Return $((p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2), X_1, Y_1, Z_1, X_2, Y_2, Z_2)$
proc Finalize($b'$)
Return $(b' = b)$
```

Fig. 5. Game $\mathsf{SXDH}_{\mathsf{GS},k}$, for $k \in \mathbb{N}$, used to define the hardness of the SXDH problem in group scheme GS

```
Algorithm $\mathsf{es}_j.\mathsf{Pg}(1^k)$   // $j = 1, 2$
    $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2) \xleftarrow{\$} \mathsf{GS}(1^k)$
    $pars \leftarrow (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2)$
    Return $pars$

Algorithm $\mathsf{es}_1.\mathsf{Kg}(pars)$
    $x_1, x_2 \xleftarrow{\$} \mathbb{Z}_p^*$
    $X_1 \leftarrow g_1^{x_1}$ ; $X_2 \leftarrow g_2^{x_2}$
    $dk \leftarrow (x_1, x_2)$ ; $ek \leftarrow (X_1, X_2)$
    Return $(ek, dk, \perp)$

Algorithm $\mathsf{es}_2.\mathsf{Kg}(pars)$
    $x_1, x_2 \xleftarrow{\$} \mathbb{Z}_p^*$
    $X_1 \leftarrow g_1^{x_1}$ ; $X_2 \leftarrow g_2^{x_2}$
    $dk \leftarrow (x_1, x_2)$ ; $ek \leftarrow (X_1, X_2)$
    Return $(ek, dk, ek)$
```

```
Algorithm $\mathsf{es}_j.\mathsf{Enc}(pars, ek, (m_1, m_2))$   // $j = 1, 2$
    $(X_1, X_2) \leftarrow ek$ ; $y_1, y_2, u_1, u_2 \xleftarrow{\$} \mathbb{Z}_p$
    $Y_1 \leftarrow g_1^{y_1}$ ; $U_1 \leftarrow g_1^{u_1}$ ; $Z_1 \leftarrow X_1^{y_1}$ ; $T_1 \leftarrow X_1^{u_1/m_2}$
    $Y_2 \leftarrow g_2^{y_2}$ ; $U_2 \leftarrow g_2^{u_2}$ ; $Z_2 \leftarrow X_2^{y_2}$ ; $T_2 \leftarrow X_2^{u_2/m_1}$
    $c_1 \leftarrow m_1 + H(pars, 1, Z_1)$
    $c_2 \leftarrow m_2 + H(pars, 2, Z_2)$
    $C \leftarrow (Y_1, U_1, T_1, Y_2, U_2, T_2, c_1, c_2)$
    Return $C$

Algorithm $\mathsf{es}_j.\mathsf{Dec}(pars, dk, C)$   // $j = 1, 2$
    $(Y_1, U_1, T_1, Y_2, U_2, T_2, c_1, c_2, ) \leftarrow C$
    $(x_1, x_2) \leftarrow dk$
    $m_1 \leftarrow c_1 - H(pars, 1, Y_1^{x_1})$
    $m_2 \leftarrow c_2 - H(pars, 2, Y_2^{x_2})$
    Return $(m_1, m_2)$
```

Fig. 6. ENC-scheme es_1 is symmetric and es_2 is asymmetric. For $j = 1, 2$ the message space is $\mathsf{es}_j.\mathsf{Msg}(pars) = \mathbb{Z}_p^* \times \mathbb{Z}_p^*$ and the ciphertext space is $\mathsf{es}_j.\mathsf{Ctxts}(pars) = \mathbb{G}_1^3 \times \mathbb{G}_2^3 \times \mathbb{Z}_p^2$.

To prove this, let GS be a group scheme for which SXDH is hard. For $j \in \{1, 2\}$, Fig. 6 associates to GS the ENC-scheme $\mathsf{es}_j = (\mathsf{es}_j.\mathsf{Pg}, \mathsf{es}_j.\mathsf{Kg}, \mathsf{es}_j.\mathsf{Enc}, \mathsf{es}_j.\mathsf{Dec}, \mathsf{es}_j.\mathsf{MsgR}, \mathsf{es}_j.\mathsf{CtxtR})$. ENC-scheme es_1 is symmetric and ENC-scheme es_2 is asymmetric. Notice both schemes can encrypt their own keys. (That is, the decryption keys are in the message space.) As described the schemes do not use coins that are bitstrings of some length $r(\cdot)$ depending only on the security parameter as our definition requires, but they may be easily modified to do this while retaining the attributes given by the Lemmas below. The following says that both schemes are IND-R-secure (and hence IND-CPA secure) assuming SXDH.

Lemma 3. *Let* $\mathsf{es}_1, \mathsf{es}_2$ *be the* ENC*-schemes associated to group scheme* GS *via Fig. 6. Suppose the SXDH problem is hard in* GS. *Then* $\mathsf{es}_1, \mathsf{es}_2$ *are* IND-R*-secure.*

The proof is in [1]. Now we show, however, that the schemes are not circular secure.

Lemma 4. *Let* $\mathsf{es}_1, \mathsf{es}_2$ *be the* ENC*-schemes associated to group scheme* GS *via Fig. 6. Then* $\mathsf{es}_1, \mathsf{es}_2$ *are not* CYC*-secure.*

Proof. We describe a PT adversary A such that $\mathbf{Adv}^{\mathsf{CYC}}_{\mathsf{es}_j, A}(k) \geq 1 - 2^{-k+1}$ for both $j = 1, 2$. The adversary ignores its input public key pk and hence works against both the symmetric and asymmetric versions of the scheme. $A(pk)$ issues a single query to **Cyc** and receives a pair (C_1, C_2) whose component ciphertexts it parses as $(Y_1, U_1, T_1, Y_2, U_2, T_2, c_1, c_2) \leftarrow C_1$ and $(\hat{Y}_1, \hat{U}_1, \hat{T}_1, \hat{Y}_2, \hat{U}_2, \hat{T}_2, \hat{c}_1, \hat{c}_2) \leftarrow C_2$. A returns 1 if $\mathbf{e}(U_1, \hat{U}_2) = \mathbf{e}(T_1, \hat{T}_2)$ and 0 otherwise. For the analysis, let $dk_1 = (x_1, x_2)$ and $dk_2 = (\hat{x}_1, \hat{x}_2)$ be the decryption keys chosen in the game. If $b = 1$, then

$$\mathbf{e}(T_1, \hat{T}_2) = \mathbf{e}(X_1^{u_1/\hat{x}_2}, \hat{X}_2^{\hat{u}_2/x_1}) = \mathbf{e}(U_1^{x_1/\hat{x}_2}, \hat{U}_2^{\hat{x}_2/x_1}) = \mathbf{e}(U_1, \hat{U}_2),$$

so A outputs 1. If $b = 0$, then the ciphertexts were sampled at random from $\mathbb{G}_1^3 \times \mathbb{G}_2^3 \times \mathbb{Z}_p^2$, so $\mathbf{e}(T_1, \hat{T}_2)$ and $\mathbf{e}(U_1, U_2)$ are uniformly random and independent elements of $\mathbb{G}_T$, and thus A returns 1 with probability $1/p$. □

Theorem 2 follows from these two lemmas.

Non-agility of wPRFs. We can now combine Theorems 1 and 2 to rule out agility of wPRFs:

Theorem 3. *Let $a \geq 2$. Suppose there exists a group scheme in which the SXDH problem is hard and further that* wPR*-secure* FF*-schemes with bit-output exist. Then the set* FF.wPR.Sch *of all* wPR*-secure* FF*-schemes is not a-agile.*

An explicit example of a pair $\{\mathsf{ff}_1, \mathsf{ff}_2\}$ of wPR-secure FF-schemes that is not 2-agile can be obtained by combining the proofs of the two theorems. However, it turns out we can give a simpler example by directly using the techniques behind the proof of Theorem 2, constructing $\mathsf{ff}_1, \mathsf{ff}_2$ as follows. For $j \in \{1, 2\}$ let $\mathsf{ff}_j.\mathsf{Pg}(1^k)$ return $pars = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2) \overset{\$}{\leftarrow} \mathsf{GS}(1^k)$, let $\mathsf{ff}_j.\mathsf{Kg}(pars)$ return $x \overset{\$}{\leftarrow} \mathbb{Z}_p^*$, let $\mathsf{ff}_j.\mathsf{Dom}(pars) = \mathbb{Z}_p$ and $\mathsf{ff}_j.\mathsf{Rng}(pars) = \mathbb{G}_j^2$. For an input $y \in \mathbb{Z}_p$, let $\mathsf{ff}_1.\mathsf{f}(pars, x, y) = (g_1^y, g_1^{y/x})$ and $\mathsf{ff}_2.\mathsf{f}(pars, x, y) = (g_2^y, g_2^{xy})$. Individually, ff_1 and ff_2 can be proven to be secure wPRFs under appropriate and relatively standard assumptions. But if the same key x is simultaneously used for both function families, an obvious distinguishing attack in the same spirit as the one above against $\{\mathsf{es}_1, \mathsf{es}_2\}$ gives an adversary high advantage.

One might ask what is the value of Theorem 1 given this direct counterexample. First, we believe Theorem 1 is interesting in its own right as a connection between seemingly unrelated primitives. Also, if there are no group schemes in which SXDH is hard, Theorem 1, which is unconditional, still stands, and could lead to either positive or negative results depending on the veracity of the underlying conjectures. Notice that if wPRFs are shown agile, our results not only imply that all IND-R-secure encryption schemes are CYC-secure but also that there are no group schemes where SXDH is true.

Separating semantic and circular security of PKE. The public-key case of Theorem 2 resolves the following question posed by Boneh, Halevi, Hamburg and Ostrovsky [11]: does there exist an IND-CPA-secure public-key encryption that becomes insecure when a 2-cycle is published? (Our es_2, being IND-R-secure, is of course IND-CPA-secure. Two-cycle security as per [11], was, however, a weaker requirement than ours, being of an IND-CPA flavor rather than our IND-R flavor. But it is not hard to show that our es_2 fails this cyclicity notion as well.)

Let us review the status of this question. As noted by Goldwasser and Micali [14], it is not hard to see that semantic security does not guarantee that it is safe to encrypt a secret key under its corresponding public key. That is, one can give a scheme that is semantically-secure but is no longer secure when the adversary is given an encryption of the secret key. A natural question is if it is safe to encrypt larger cycles. Backes et al. [5] showed it is not safe for stateful symmetric encryption, but an adversary could only break circular-security when given access to encryptions of each key under its initial state. Boneh et al. [11] gave a simple public-key scheme that was one-way secure during a chosen-plaintext attack but not one-way after a circular encryption was published. Neither of the techniques in these works seemed to generalize to resolve the question for semantic-versus-circular security of public-key encryption, which was particularly relevant given recent effort towards constructing circular-secure public-key schemes.

5 Positive Results

The above may make us pessimistic about achieving agility but there is good news as well. First, certain primitives are agile. Second, there are steps we can take to get strong agility in practice for primitives like AE. The idea is to not use the key directly with AE but instead use a subkey derived based on the description of the AE scheme. The latter brings out the key role of PRFs and wPRFs in agility. Let us expand on these items.

Agile primitives. Collision-resistant hash functions, formalized as keyed families, are agile. IND-CPA secure public-key encryption schemes are agile. (But not IND-CCA-secure public-key encryption schemes, and not IND-CPA symmetric encryption schemes!) In both cases the reason is simple, namely that one only needs access to public information (the hashing key or public encryption key) to simulate an adversary.

PRF-based agility for AE. The existence and continued appearence of new AE schemes makes the agility of AE important. We have seen that we can't get agility for all AE schemes. Arguably, in practice, however, it may be enough to get it for a subset of them, such as CCM, OCB, CWC, GCM, EAX. However, the designs are sufficiently related that we suspect even this small set is in fact *not* agile! (That is, using the same key for all of them at the same time is insecure.)

We now show how to circumvent these difficulties and achieve AE agility, not only for the above schemes, but for all AE schemes, by using the schemes not directly but inside a construction. The requirement is a PRF that is either fixed or itself drawn from a small, agile space. This requirement is not too onerous because there are more proposals and choices for higher level primitives like AE than for the blockciphers that instantiate PRFs in practice. (Typically, one just uses AES.)

For our construction and analysis that follows now, we introduce some notation to ensure that the components we are using "fit together." Let $\ell(\cdot)$ be a polynomial. Let $\mathsf{FF.PR.Sch}[\ell]$ be the set of all $\mathsf{ff} \in \mathsf{FF.PR.Sch}$ such that $\mathsf{ff.Dom}(\mathit{pars}) = \{0,1\}^*$ and $\mathsf{ff.Rng}(\mathit{pars}) = \{0,1\}^{\ell(k)}$ for all $\mathit{pars} \in [\mathsf{FF.Pg}(1^k)]$ and all $k \in \mathbb{N}$. Let $\mathsf{ENC.AuE.Sch}[\ell]$ be the set of all $\mathsf{es} \in \mathsf{ENC.AuE.Sch}$ such that $\mathsf{es.Kg}(\mathit{pars})$ induces a uniform distribution on $\{0,1\}^{\ell(k)}$ for all $\mathit{pars} \in [\mathsf{es.Pg}(1^k)]$ and all $k \in \mathbb{N}$. We let $\langle \mathsf{es} \rangle \in \{0,1\}^*$ be some unique string-encoding of the description of es in the sense that no two schemes in $\mathsf{ENC.AuE.Sch}$ have the

```
Algorithm $\mathsf{es}_{\mathsf{ff}}.\mathsf{Pg}(1^k)$
$fpars \xleftarrow{\$} \mathsf{ff}.\mathsf{Pg}(1^k)$ ; $epars \xleftarrow{\$} \mathsf{es}.\mathsf{Pg}(1^k)$
Return $(fpars, epars)$

Algorithm $\mathsf{es}_{\mathsf{ff}}.\mathsf{Kg}((fpars, epars))$
$K \xleftarrow{\$} \mathsf{ff}.\mathsf{Kg}(fpars)$
Return $(K, K, \perp)$

Algorithm $\mathsf{es}_{\mathsf{ff}}.\mathsf{Enc}((fpars, epars), K, M)$
$K_{\mathsf{es}} \leftarrow \mathsf{ff}.\mathsf{f}(fpars, K, \langle \mathsf{es} \rangle)$
$C \xleftarrow{\$} \mathsf{es}.\mathsf{Enc}(epars, K_{\mathsf{es}}, M)$
Return $C$

Algorithm $\mathsf{es}_{\mathsf{ff}}.\mathsf{Dec}((fpars, epars), K, C)$
$K_{\mathsf{es}} \leftarrow \mathsf{ff}.\mathsf{f}(fpars, K, \langle \mathsf{es} \rangle)$
$M \leftarrow \mathsf{es}.\mathsf{Dec}(epars, K_{\mathsf{es}}, C)$
Return $M$
```

Fig. 7. The symmetric ENC-scheme scheme $\mathsf{es}_{\mathsf{ff}}$ associated to symmetric ENC-scheme es and FF-scheme ff

same encoding. Such an encoding always exists since schemes are finite tuples of algorithms and thus have finite descriptions.

Theorem 4. *Let $\ell(\cdot)$ be a polynomial. Let $\Gamma \subset \mathsf{FF.PR.Sch}[\ell]$ be a compatible, finite set that is agile with respect to* PR*. Then, for every $a \in \mathbb{N}$, the set $\{ \mathsf{es}_{\mathsf{ff}} : \mathsf{es} \in \mathsf{ENC.AuE.Sch}[\ell],\ \mathsf{ff} \in \Gamma \}$ is a-agile with respect to* AuE*.*

In particular, for $\mathsf{ff} \in \mathsf{FF.PR.Sch}[\ell]$ and every $a \in \mathbb{N}$, the set $\{ \mathsf{es}_{\mathsf{ff}} : \mathsf{es} \in \mathsf{ENC.AuE.Sch}[\ell] \}$ is a-agile with respect to ENC.AuE. The proof of Theorem 4 is in [1].

wPRF-based agility for AE. We would like to use wPRFs in place of PRFs because wPRF is a weaker assumption on a blockcipher than a PRF and, as a result, a set of blockciphers is more likely to be agile with respect to wPRF than to PRF. (We cannot of course hope for agility with respect to all wPRFs since that class is not agile. But we'd like to get it for as large a subset of the class as possible.) We show this is possible. This explains our interest in wPRFs and their importance in the agility domain.

The obvious modification to the above construction when ff is a wPRF rather than a PRF is for $\mathsf{es}_{\mathsf{ff}}((fpars, epars), K, M)$ to pick a random R, let $K_{\mathsf{es}} \leftarrow \mathsf{ff}.\mathsf{f}(fpars, K, R)$, let $C \xleftarrow{\$} \mathsf{es}.\mathsf{Enc}(epars, K_{\mathsf{es}}, M)$, and return (C, R) as the ciphertext, R being included to allow decryption. However it is easy to see that this is not secure. Even ignoring agility, $\mathsf{es}_{\mathsf{ff}}$ fails to be a secure AE scheme in general. Instead, we consider the constructions of PRFs from wPRFs due to Naor and Reingold [24], Maurer and Sjödin [21] and Maurer and Tessaro [22]. Some of the constructed PRFs make only blackbox appeal to a fixed number of wPRFs on independent keys. These types of constructions are agility-preserving in the sense that the set of constructed PRFs obtained by using wPRFs from a set Γ is agile with respect to PR if Γ was agile with respect to wPR. Now, we can use our construction above.

Acknowledgments

The third and fourth authors were supported in part by NSF grants CNS-0627779 and CCF-0915675. We thank Tom Ristenpart for useful pointers.

References

1. Acar, T., Belenkiy, M., Bellare, M., Cash, D.: Cryptographic agility and its relation to circular encryption. Cryptology ePrint Archive (2010), http://eprint.iacr.org/
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. Ateniese, G., Camenisch, J., Hohenberger, S., de Medeiros, B.: Practical group signatures without random oracles. Cryptology ePrint Archive, Report 2005/385 (2005), http://eprint.iacr.org/
4. Backes, M., Dürmuth, M., Unruh, D.: OAEP is secure under key-dependent messages. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 506–523. Springer, Heidelberg (2008)
5. 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. Comput. Secur. 16(5), 497–530 (2008)
6. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption. In: 38th FOCS, October 1997, pp. 394–403. IEEE Computer Society Press, Los Alamitos (1997)
7. 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)
8. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (2006)
9. Bellare, M., Rogaway, P., Wagner, D.: The EAX mode of operation. In: Roy, B.K., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 389–407. Springer, Heidelberg (2004)
10. 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)
11. 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)
12. 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)
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. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)
15. Haber, S., Pinkas, B.: Securely combining public-key cryptosystems. In: ACM CCS 2001, November 2001, pp. 215–224. ACM Press, New York (2001)
16. 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)

17. Halevi, S., Krawczyk, H.: Security under key-dependent inputs. In: Ning, P., di Vimercati, S.D.C., Syverson, P.F. (eds.) ACM CCS 2007, October 2007, pp. 466–475. ACM Press, New York (2007)
18. 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)
19. Kelsey, J., Schneier, B., Wagner, D.: Protocol interactions and the chosen protocol attack. In: Christianson, B., Lomas, M. (eds.) Security Protocols 1997. LNCS, vol. 1361, pp. 91–104. Springer, Heidelberg (1998)
20. Kohno, T., Viega, J., Whiting, D.: CWC: A high-performance conventional authenticated encryption mode. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 408–426. Springer, Heidelberg (2004)
21. Maurer, U.M., Sjödin, J.: A fast and key-efficient reduction of chosen-ciphertext to known-plaintext security. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 498–516. Springer, Heidelberg (2007)
22. Maurer, U.M., Tessaro, S.: Basing PRFs on constant-query weak PRFs: Minimizing assumptions for efficient symmetric cryptography. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 161–178. Springer, Heidelberg (2008)
23. McGrew, D.A., Viega, J.: The security and performance of the Galois/counter mode (gcm) of operation. In: Canteaut, A., Viswanathan, K. (eds.) INDOCRYPT 2004. LNCS, vol. 3348, pp. 343–355. Springer, Heidelberg (2004)
24. Naor, M., Reingold, O.: Synthesizers and their application to the parallel construction of pseudo-random functions. J. Comput. Syst. Sci. 58(2), 336–375 (1999)
25. Nelson, D.: Crypto-agility requirements for remote dial-in user service (radius). IETF Network Working Group Internet-Draft (November 2008), http://tools.ietf.org/html/draft-ietf-radext-crypto-agility-requirements-01
26. Rogaway, P., Bellare, M., Black, J., Krovetz, T.: OCB: A block-cipher mode of operation for efficient authenticated encryption. In: ACM CCS 2001, November 2001, pp. 196–205. ACM Press, New York (2001)
27. 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)
28. Sullivan, B.: Cryptographic agility. Microsoft Developer Network Magazine (August 2009), http://msdn.microsoft.com/en-us/magazine/ee321570.aspx
29. Whiting, D., Housley, R., Ferguson, N.: Counter with CBC-MAC (CCM). RFC 3610 (Informational) (September 2003)

Bounded Key-Dependent Message Security

Boaz Barak[1,⋆], Iftach Haitner[2], Dennis Hofheinz[3,⋆⋆], and Yuval Ishai[4,⋆⋆⋆]

[1] Princeton University
boaz@cs.princeton.edu
[2] Microsoft Research
iftach@microsoft.com
[3] Karlsruhe Institute of Technology
Dennis.Hofheinz@kit.edu
[4] Technion and UCLA
yuvali@cs.technion.ac.il

Abstract. We construct the first public-key encryption scheme that is proven secure (in the standard model, under standard assumptions) even when the attacker gets access to encryptions of arbitrary efficient functions of the secret key. Specifically, under either the DDH or LWE assumption, and for arbitrary but fixed polynomials L and N, we obtain a public-key encryption scheme that resists key-dependent message (KDM) attacks for up to $N(k)$ public keys and functions of *circuit size* up to $L(k)$, where k denotes the size of the secret key. We call such a scheme *bounded KDM secure.* Moreover, we show that our scheme suffices for one of the important applications of KDM security: ability to securely instantiate symbolic protocols with axiomatic proofs of security.

We also observe that any fully homomorphic encryption scheme that additionally enjoys circular security and circuit privacy is *fully KDM secure* in the sense that its algorithms can be independent of the polynomials L and N as above. Thus, the recent fully homomorphic encryption scheme of Gentry (STOC 2009) is fully KDM secure under certain non-standard hardness assumptions.

Finally, we extend an impossibility result of Haitner and Holenstein (TCC 2009), showing that it is impossible to prove KDM security against a family of query functions that contains exponentially hard pseudorandom functions if the proof makes only a *black-box* use of the query function and the adversary attacking the scheme. This shows that the non-black-box use of the query function in our proof of security is inherent.

Keywords: KDM/clique/circular security; fully homomorphic encryption; formal security.

⋆ Supported by NSF grants CNS-0627526, CCF-0426582 and CCF-0832797, US-Israel BSF grant 2004288 and Packard and Sloan fellowships.

⋆⋆ Part of work performed while at CWI and supported by an NWO Veni grant.

⋆⋆⋆ Supported by BSF grant 2008411, ISF grant 1310/06, and NSF grants 0830803, 0716835, 0627781.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 423–444, 2010.

1 Introduction

An encryption scheme is *key-dependent message (KDM) secure* if it is secure even against an attacker who has access to encryptions of messages that depend on the secret key. This strong notion of security, introduced by Black et al. [6], tries to capture scenarios where there could be correlations between the secret key and the encrypted messages. At a first glance, it may seem that such correlations only arise from bugs or errors on part of the protocol designer, and hence achieving such a strong security is not of much importance. It turns out, however, that such attacks naturally occur when considering complex systems. For example, in some popular disk encryption utilities, the disk encryption key can end up being stored in the page file, and thus is encrypted along with the disk content [7]. In addition, Camenisch and Lysyanskaya [9] showed that schemes with a certain restricted form of KDM security known as "circular security" are useful for constructing *Anonymous Credential Systems*. Finally, and perhaps most importantly, KDM security naturally arises as the right notion when one wishes to securely instantiate symbolic protocols with an axiomatic proof of formal security (see Section 6).

For a while, building a KDM-secure encryption scheme in the standard model, under any well studied hardness assumption, seemed too hard a nut to crack. The only scheme that was shown to resist any kind of KDM attacks was given by Black et al. [6] in the random-oracle model. Yet, in recent years KDM-secure encryption schemes were given for some non-trivial families of functions. This line of work started with the works of Halevi and Krawczyk [18] and Hofheinz and Unruh [19], who gave private-key encryption schemes secure against significantly restricted classes of KDM queries. Concretely, [18] prove security against arbitrary but fixed KDM queries that are known in advance, and against KDM queries that do not depend on certain "protected" parts of the key. The constructions from [19] obtain statistical KDM security in the presence of sufficiently few (arbitrary) KDM queries, as well as a stateful KDM-secure scheme in which KDM queries may only depend on the current state (but not on previous states).

A major step was taken by Boneh, Halevi, Hamburg, and Ostrovsky [7] who presented, under the *decisional Diffie-Hellman (DDH)* assumption, a public-key encryption scheme that is $N(k)$-circular secure for every polynomial N, and in fact is secure against the more general family of attacks allowing the adversary access to encryptions of arbitrary affine functions of the vector of $N(k)$ secret keys. Applebaum, Cash, Peikert, and Sahai [3] presented more efficient schemes that are secure against a similar family of key-dependent attacks, whose security is based on different assumptions: the *learning parity with noise (LPN)* assumption in the secret-key case and the *learning with errors (LWE)* assumption in the public-key case. In a recent independent work, Brakerski, Goldwasser, and Kalai [8] presented a transformation from a KDM secure scheme satisfying a certain property (in particular satisfied by the DDH and LWE based schemes of [7, 3]) into a scheme that is KDM secure with respect to a larger class of functions. While their transformation cannot be used to achieve security against all circuits of size $p(n)$, it has the benefit of depending only on the number of functions in the class, and being independent of their circuit size or number of keys. In particular they achieve KDM security with respect to the class of constant degree polynomials and any polynomial number of keys.

Despite the above progress, the families of functions for which KDM security was achieved prior to our work (in the standard model, under standard assumptions) was still quite restricted. In particular, these families were not sufficiently rich for several of the applications of KDM security in the context of complex systems and formal protocols. A partial explanation for this rather limited success was recently given by Haitner and Holenstein [17], who showed the impossibility of obtaining KDM security based on standard assumptions and using standard techniques. (In Section 1.2, we will describe their results in more detail, since we will later extend them to our case of bounded KDM security.)

1.1 Our Results

Our main result is the following:

Theorem 1 (Informal). *Under the DDH or LWE assumption, for any given polynomials $L = L(k)$ and $N = N(k)$, there exists a public-key encryption scheme that is KDM-secure with respect to the class of circuits of size $L(k)$, and for $N(k)$ independent keys, where k denotes the size of the keys.*

We call such a scheme a *bounded KDM-secure* encryption scheme. (This is in contrast with a *fully*- or *unbounded*-KDM scheme, where the circuit size and the number of keys can be an arbitrarily large polynomial in the security parameter, independent of the scheme's complexity.) We argue that this is the first encryption scheme (under standard cryptographic assumptions) that handles a rich enough function class to capture most "real life" KDM attacks.

The original motivation for KDM security was to securely instantiate symbolic cryptographic protocols that have a formal proof of security in some axiomatic system. As further evidence for the usefulness of bounded KDM security, we show that our notion is strong enough for this application:

Theorem 2 (Informal). *Let P be a symbolic protocol with operations such as public-key encryption and digital signatures. Then, instantiating P with a bounded KDM-secure[1] encryption scheme provides a computationally sound implementation.*

This yields the first soundness result without restrictions (such as assuming protocols without key-cyclic expressions) in the standard model.

Finally, we show that the above positive results are tight, by extending an impossibility result of Haitner and Holenstein [17] in the following sense:

[1] Actually, the precise notion we use is *length-dependent* KDM security (see Definition 5). This is a slight strengthening of bounded KDM security, and our scheme satisfies this stronger notion as well.

Theorem 3 (Informal). *An encryption scheme cannot be proven to be KDM-secure against a family of functions that contains exponentially hard pseudorandom functions, if the proof of security only accesses the query function and the adversary attacking the scheme in a black-box manner (i.e., as oracles).*

Remarks. We note the following points about our result:

1. *Efficiency.* Our scheme, although polynomial time, is not practically efficient as it uses the garbled circuit construction and its ciphertext length is at least L, where L is a bound on the circuit size of the KDM function. There are more efficient candidate KDM-secure cryptosystems if one is willing to settle for non-standard assumptions or the random oracle model.
2. *Full KDM security.* Although we only prove our scheme to be bounded KDM secure, it is of course possible that it is KDM secure with respect to *any* efficient KDM function. In fact, there seems to be an interesting obstacle to any KDM attack on our scheme. Suppose that we instantiate the scheme to be secure with respect to KDM functions of size k^3. Now suppose that there is a successful KDM attack against it, and for simplicity assume the attack consists of getting one encryption of $h(sk)$ where h is some efficiently computable function. Then the success of this attack implies that either DDH is false (assuming we instantiate our scheme from the DDH assumption), or that h has no circuit of size k^3. Hence, a proof that this construction is insecure against a polynomial-time KDM attacker will provably demonstrate than either DDH is false, or that $\mathbf{P} \not\subseteq \mathbf{Size}(k^2)$ (we lose a factor of k because h has a k-bit output). The latter is a widely believed fact, but its proof would be considered a major breakthrough in complexity theory. (Also, it is not at all clear how to derive such a conclusion directly from the DDH assumption— typically in cryptography we need to use *subexponential* hardness assumptions to get such a condition.) More generally, a successful attack is some way to *certify* that h is hard— even though it is easy in time k^3 to generate a random function outside of $\mathbf{Size}(k^2)$, it is not at all clear how to generate such a function along with a publicly verifiable certificate of hardness.
3. *Black-box-ness.* Our scheme makes a non-black-box use of the KDM function h, where Theorem 3 shows that this use is *inherent.*

Applications to formal security. A central motivation for the study of KDM security lies in the connection between formal and computational cryptography. In formal cryptography (starting with [12, 13, 23]), cryptographic operations like encryption or digital signatures are abstracted as symbolic operators that (only) adhere to natural rules. Given such rules, a simple calculus enables machine-assisted security analysis.

It was proven by Adão et al. [2] that *fully* KDM-secure encryption schemes imply computational soundness for *arbitrary* symbolic protocols. We reconsider their proof and show (Theorem 9) that bounded KDM security of the type that we achieve suffices. Hence, our combined results give the first encryption scheme (under standard cryptographic assumptions) whose security implication can be verified using formal security methods.

We stress that the clique security achieved by [7, 3] only enables to apply these formal methods to a very limited class of applications. For more details see Section 6.

1.2 Our Techniques

We now give an informal overview of the proof of Theorem 1. The following exposition focuses on a scheme that is secure against a *single-key* KDM attack. That is, there is only one public/private key pair (pk, sk) of length k, and the attacker can obtain encryptions of messages of the form $h(sk)$ for an arbitrary function h of circuit complexity at most $L(k)$. (Here $L = L(k)$ is an arbitrary fixed polynomial which affects the complexity of the encryption and decryption, but not the complexity of key generation.) The multiple-key case raises some additional subtleties that we ignore for the moment.

Recall that a *homomorphic* public-key encryption scheme is a public-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ that also has an additional algorithm Eval for evaluating functions on an encrypted message. Concretely, Eval takes the public key pk, an encryption of a message M, and a description of a function h from some family $\mathcal{H}$, and outputs a string from which $h(M)$ can be efficiently decrypted using the secret key sk. Our starting point is the following observation: a sufficiently strong homomorphic encryption is in fact also KDM-secure (with respect to the same class of functions $\mathcal{H}$), where "sufficiently strong" means that the scheme satisfies the following additional properties:

1. Self-referential (i.e., 1-circular) security: $\mathsf{Enc}_{pk}(sk)$ is indistinguishable from $\mathsf{Enc}_{pk}(0^k)$.
2. Strong function privacy: For every $h \in \mathcal{H}$ and plaintext M, $\mathsf{Eval}_{pk}(h, \mathsf{Enc}_{pk}(M))$ is indistinguishable from $\mathsf{Enc}_{pk}(h(M))$, even against a distinguisher that knows the secret key.

The basic idea for proving the KDM-security of such a scheme is that a distinguisher between $\mathsf{Enc}_{pk}(h(sk))$ and $\mathsf{Enc}_{pk}(h(0^k))$ can be used to distinguish between $\mathsf{Enc}_{pk}(sk)$ and $\mathsf{Enc}_{pk}(0^k)$ by simply running Eval with the function h (and thus the "KDM queries" are useless). When turning this idea into a proof one sees that it is crucial that function privacy hold even with respect to a distinguisher that knows the secret key.

This observation already implies that Gentry's recent breakthrough fully homomorphic encryption scheme [15] is fully KDM-secure, assuming that it is circular-secure (an assumption which is anyway necessary in Gentry's case to get a truly fully homomorphic encryption, where the public key does not grow with the depth of the circuit).[2] Since all *natural* candidates for public-key encryption schemes are not known to be 1-circular insecure, we find this observation interesting, as the assumption of circular security seems cleaner and more conservative than assuming full KDM security. (In particular, it is more easily "falsifiable" in the sense of Naor [25].)

[2] As Gentry notes, if one assumes his scheme is circular-secure then it also enjoys strong statistical function privacy.

In fact, it turns out that it suffices to have only *weak* function privacy, requiring that $\mathsf{Eval}_{pk}(h, \mathsf{Enc}_{pk}(M))$ be indistinguishable from $\mathsf{Eval}_{pk}(h', \mathsf{Enc}_{pk}(M))$ for h, h', M such that $h(M) = h'(M)$ (again indistinguishability is with respect to attackers who know the secret key).[3] See Theorem 5 for the details.

The latter observation suggests an approach to get KDM security for circuits of size L under standard assumptions. Consider any two-message protocol for evaluating a *universal function* with security against semi-honest parties. Such a protocol takes an input M from a receiver and a circuit h from a sender, and delivers the output $h(M)$ to the receiver. Given any such protocol and a standard public-key encryption (PKE), one can construct a homomorphic scheme with *weak* function privacy as follows. The public key is the public key pk of the PKE. The encryption of M under pk is a triple (C, pk', C') where C is an encryption of M under pk, pk' is the receiver's first message in the protocol on input M, and C' is an encryption under pk of the secret randomness sk' used to generate pk' (which is needed to recover the output of the protocol). The algorithm $\mathsf{Eval}((C, pk', C'), h)$ returns the sender's response to pk' on input h along with C'. Given sk, the output of Eval can be used to decrypt $h(M)$ by first recovering sk' and then computing the receiver's output in the protocol.

The advantage of this approach is that it can be instantiated under standard assumptions by using Yao's protocol [28]. More concretely, a secure two-message protocol for the universal function can be obtained by combining Yao's garbled circuit construction and any *two-message oblivious transfer* (OT) [27, 14] protocol.[4] Unlike the alternative of using *fully* homomorphic encryption, however, this protocol has the caveat that the communication must grow with the size of h, and hence (weak) function privacy can only hold with respect to the class $\mathcal{H}$ of all circuits with some a-priori size bound L.

A more subtle problem is that of making the homomorphic scheme constructed in the above way circular-secure. Indeed, encrypting the secret key of the homomorphic scheme with its own public key results in a circular dependency between the underlying PKE and the two-message protocol: the secret key sk of the PKE is encrypted using the "public key" pk' of the protocol, whereas the "secret key" sk' of the protocol is encrypted using the public key pk of the PKE. Even if the PKE is circular-secure, it is not clear that this property will be respected by the above construction.

Our way to handle this difficulty is by introducing a new notion that we call "targeted encryption", which is aimed towards resolving the above dependency

[3] This is indeed a weaker notion since if $y = h(M) = h'(M)$ then one can see that strong function privacy implies that $\mathsf{Eval}(h, \mathsf{Enc}(M)) \approx \mathsf{Enc}(y) \approx \mathsf{Eval}(h', \mathsf{Enc}(M))$. Intuitively, weak function privacy allows Eval to map ciphertexts from one domain to a different domain, while this is ruled out by strong function privacy.

[4] Two-message OT is a protocol comprised of one message from the receiver and one message from the sender, where the receiver has an input selection bit s and the sender has a pair of input strings X_0, X_1. In the end of the protocol the receiver only learns X_s and the sender learns nothing about s. Here we need k parallel instances of OT, where the receiver has a k-bit input selection vector s and the sender has k pairs of strings.

when applied to a two-message protocol based on Yao's technique. Targeted encryption can be viewed as a circular-secure extension of both public-key encryption and two-message OT. Loosely speaking, one can think of this as an OT protocol where the receiver has *no secret information* apart from the input selection vector s. This may look strange at first, and indeed it can be shown to be inherently at odds with the standard notion of OT, which requires that the sender learn nothing about s. But it turns out that one can obtain a meaningful relaxation of the above notion that is strong enough for our purposes. We then show that both the schemes of Boneh et al. [7] and Applebaum et al. [3] can be used to construct targeted encryption. The key property we use is that both schemes enjoy KDM security against *affine functions*, and in fact this is proven by giving a public algorithm to compute an encryption of any affine function of the secret key. We show that such an algorithm implies targeted encryption.

As mentioned above, multiple-key security adds some additional difficulties. In particular, targeted encryption on its own does not seem sufficient for *multiple key* security, and to handle this case we need to appeal to the multiple-key circular security of the underlying schemes.

To show our negative result (Theorem 3), we employ the techniques of Haitner and Holenstein [17]. Concretely, they showed that an encryption scheme cannot be proved to be KDM-secure against the family of *all* functions, if the proof of security only accesses the query function and the adversary attacking the scheme in a black-box manner (i.e., as oracles).[5] Here we extend this result to *every* family of functions that contains exponentially hard pseudorandom functions. There was no prior scheme that was shown (under a standard assumption) to be secure with respect to such a family, although many of the applications of KDM security require that the KDM function can be a cryptographic primitive such as a signature, a hash function, etc.

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 X, we denote by $x \leftarrow X$ the process of sampling x uniformly from X. For a probabilistic algorithm A, we denote by $y \leftarrow A(x)$ the process of running A on input x and with uniform randomness, and assigning y the result. If A runs in time polynomial in the security parameter k, then A is a PPT machine. (We always assume that k can be efficiently computed from the input to the algorithm even if it not explicitly given.) A function $f : \mathbb{N} \to [0, 1]$ is *negligible* iff $\forall c \in \mathbb{N}\, \exists k_0 \in \mathbb{N}\, \forall k > k_0 : |f(k)| < k^{-c}$. We say f is *overwhelming* iff $1 - f$ is negligible. Two collections $X = (X_k)_{k\in\mathbb{N}}$ and $Y = (Y_k)_{k\in\mathbb{N}}$ of random variables are *computationally indistinguishable*, written

[5] They also showed that it is impossible to prove (in a black-box way) that a trapdoor-permutation based scheme is KDM-secure against a family of t-wise independent hash functions, for t that is longer than the ciphertext size (here a non-black-box access to the query function is allowed).

$X \overset{c}{\approx} Y$, iff for every nonuniform polynomial-time distinguisher D, we have that $\mathsf{Pr}\left[D(1^k, X_k) = 1\right] - \mathsf{Pr}\left[D(1^k, Y_k) = 1\right]$ is negligible. We use $\circ$ for concatenation.

Encryption schemes. A public-key encryption (PKE) scheme with message space $\mathcal{M} = \mathcal{M}_k$ and secret key space $\mathcal{K} = \mathcal{K}_k$, consists of three algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ — Key generation $\mathsf{Gen}(1^k)$ outputs a public key pk and a secret key $sk \in \mathcal{K}_k$. Encryption $\mathsf{Enc}_{pk}(M)$ takes a public key pk and a message $M \in \mathcal{M}_k$, 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 require $\mathsf{Dec}_{sk}(C) = M$ for all $M \in \mathcal{M}_k$, all (pk, sk) in the range of $\mathsf{Gen}(1^k)$, and all C in the range of $\mathsf{Enc}_{pk}(M)$. For simplicity, we will assume from now on that both the key space and the message space are $\{0,1\}^k$. Our definitions and results, however, can be easily adapted to the case of messages of arbitrary length.

2.1 Garbled Circuits

An essential building block of our KDM secure encryption scheme is Yao's garbled circuit construction, attributed to [28]. Informally, the variant of this construction on which we rely transforms any circuit h with k input bits along with k pairs of random keys $(K_{1,0}, K_{1,1}), \ldots, (K_{k,0}, K_{k,1})$ into a "garbled circuit" GC such that the following properties hold:

- For any input $x \in \{0,1\}^k$ and any choice of $2k$ keys, the output $h(x)$ can be efficiently decoded (without knowing h) from GC and the k keys K_{i,x_i} corresponding to x.
- GC together with the k keys corresponding to x computationally hide all information about h other than the size of h and $h(x)$.
- GC alone computationally hides all information about h other than its size,

where the last two properties hold with respect to a random choice of the keys and a random execution of the transformation. The existence of a construction satisfying the above requirements is formally captured by the following theorem. See the full version of this paper [5] for a derivation of this theorem from the literature.

Theorem 4 (Garbled circuits). *Suppose that one-way functions exist. Then there is a pair of polynomial-time randomized algorithms* $(\mathsf{Garble}, \mathsf{GCEval})$ *that for security/input parameter k, output parameter m, and circuit size parameter s satisfy the following:*

Syntax. Garble *takes a $2k$ key tuple $\overline{K} = \{K_{i,b}\}_{i\in[k], b\in\{0,1\}}$, where $K_{i,b} \in \{0,1\}^k$, and a size s circuit describing a function $h : \{0,1\}^k \to \{0,1\}^m$, and outputs a "garbled circuit" GC.* GCEval *takes an input $x \in \{0,1\}^k$, a k key tuple, and a garbled circuit GC and outputs $y \in \{0,1\}^m$.*

Correctness. *We require that if $GC = \mathsf{Garble}(\overline{K}, h)$ then $\mathsf{GCEval}(\overline{K}_x, GC) = h(x)$, where we define $\overline{K}_x = \{(x_i, K_{i,x_i})\}_{i\in[k]}$.* [6]

[6] For ease of notation we assume that the input x is included in the description of $\overline{K}_x$. This is needed to guarantee correctness even when a pair of keys happen to be identical. Alternatively, we could avoid giving x as input to GCEval by either settling for statistical correctness or allowing the keys to be correlated.

Security against receiver. *For every polynomials* $s(k), m(k)$*, every* $x \in \{0,1\}^k$ *and every* $h, h' : \{0,1\}^k \to \{0,1\}^{m(k)}$ *of size* $s(k)$ *such that* $h(x) = h'(x)$*, if* $\overline{K}$ *is chosen at random then*

$$\overline{K}_x \circ \mathsf{Garble}(\overline{K}, h) \overset{c}{\approx} \overline{K}_x \circ \mathsf{Garble}(\overline{K}, h')$$

Security against outsiders. *For every polynomials* $s(k), m(k)$ *and every* $h, h' : \{0,1\}^k \to \{0,1\}^{m(k)}$ *of size* $s(k)$*, if* $\overline{K}$ *is chosen at random then*

$$\mathsf{Garble}(\overline{K}, h) \overset{c}{\approx} \mathsf{Garble}(\overline{K}, h')$$

2.2 Key-Dependent Message Security

Loosely speaking, the notion of *key-dependent message (KDM) security* gives an adversary access to encryptions of messages of the form $h(sk)$, where $h : \mathcal{K} \to \mathcal{M}$ is a function that the adversary can choose from some family. The formal definition below is taken from Black et al. [6] and allows the function to depend on some $N = N(k)$ secret keys. While handling multiple keys is important for the application to formal cryptography (see Section 6), much of the technical challenge is already manifested in the case $N = 1$, and so the reader may wish to focus on this case initially.

Definition 1 (KDM security). *Let* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public-key encryption scheme with message space* $\mathcal{M}$ *and secret key space* $\mathcal{K}$*. Let* $\overline{pk} := (\overline{pk}_1, \ldots, \overline{pk}_N)$ *and* $\overline{sk} := (\overline{sk}_1, \ldots, \overline{sk}_N)$ *be public, resp., secret key vectors, where* $N = N(k) > 0$ *is a positive-valued function. Let* A *be a* PPT *machine. Let*

- $\mathsf{Real}_{\overline{pk},\overline{sk}}$ *be the oracle that on input a function* $h : \mathcal{K}^N \to \mathcal{M}$ *(encoded as a circuit) and* $\mu \in [N]$ *returns* $C \leftarrow \mathsf{Enc}(\overline{pk}_\mu, h(\overline{sk}))$*, and*
- $\mathsf{Fake}_{\overline{pk}}$ *be the oracle that on input* h, μ *as above returns* $C \leftarrow \mathsf{Enc}(\overline{pk}_\mu, 0^k)$*.*

The KDM advantage *of* A *is*

$$\mathsf{Adv}^{\mathsf{KDM}}_{\mathsf{PKE},A}(k) := \Pr\left[A^{\mathsf{Real}_{\overline{pk},\overline{sk}}(\cdot,\cdot)}(\overline{pk}) = 1\right] - \Pr\left[A^{\mathsf{Fake}_{\overline{pk}}(\cdot,\cdot)}(\overline{pk}) = 1\right]$$

where $(\overline{pk}_i, \overline{sk}_i) \leftarrow \mathsf{Gen}(1^k)$ *for* $i \in [N]$ *in both probabilities. We say that* PKE *is* KDM secure with respect to a function class $\mathcal{H}$ *iff for every polynomial* N *and every* PPT A *that only queries its oracle with functions* $h \in \mathcal{H}$*, the advantage function* $\mathsf{Adv}^{\mathsf{KDM}}_{\mathsf{PKE},A}$ *is negligible in the security parameter.* PKE *is* fully KDM secure *iff* PKE *is KDM secure with respect to the class* $\mathcal{H}$ *that consists of all functions.*

Examples of KDM function classes. The following examples of KDM function classes will be important for us.

Clique/circular security. Let $\mathcal{S}_N$ consist of all functions $h_i : (\{0,1\}^k)^N \to \{0,1\}^k$ for $i \in [N]$, where $h_i(\overline{sk}_1, \ldots, \overline{sk}_N) = \overline{sk}_i$. Thus, KDM security with respect to $\mathcal{S}_N$ allows the adversary to obtain encryptions $\mathsf{Enc}_{\overline{pk}_i}(\overline{sk}_j)$ for every $i, j \in [N]$. This was called "clique security" by Boneh et al. [7] who gave a scheme that is KDM secure with respect to $\mathcal{S}_N$ for every N that is polynomial in the security parameter. (See Applebaum et al. [3] for another construction.) Security with respect to $\mathcal{S}_N$ implies "N-circular security". This notion, defined by [9] states that for independently generated N key pairs $(pk_1, sk_1), \ldots, (pk_N, sk_N)$, the vector of N encryptions $\mathsf{Enc}_{pk_1}(sk_2), \mathsf{Enc}_{pk_2}(sk_3), \ldots, \mathsf{Enc}_{pk_N}(sk_1)$ is indistinguishable from $\mathsf{Enc}_{pk_1}(0^k), \ldots, \mathsf{Enc}_{pk_N}(0^k)$.

Bounded security. Let $\mathcal{C}_{N,L}$ consist of all functions $h : (\{0,1\}^k)^N \to \{0,1\}^k$ that can be described with circuits of size at most L. We say that a scheme is (N, L) *bounded KDM secure*, if it is KDM secure with respect to $\mathcal{C}_{N(k),L(k)}$, where k denotes both the security parameter and the secret key size.[7]

Full (unbounded) security. Full KDM security is equivalent to requiring that a scheme is KDM secure with respect to $\mathcal{C}_{N,L}$ for every polynomials, in the security parameter, N and L. Note that this definition seems like the best one should look for, since a PPT adversary cannot generate circuits (i.e., queries) of superpolynomial size.

Finally, we say that a scheme has *single-key KDM security*, if in the KDM attack above the number of keys N is restricted to being 1. This notion makes sense with respect to bounded/unbounded security, where in the case of or clique or circular security it is equivalent to "self reference security" — the adversary has access to $\mathsf{Enc}_{pk}(sk)$.

2.3 KDM Security from Homomorphic Encryption

In this section we observe that one can get KDM security from a certain kind of homomorphic encryption schemes.

Definition 2 (Homomorphic encryption). *Let $\mathcal{H} = \{\mathcal{H}_k\}$ be a sequence of sets of Boolean circuits. A tuple of algorithms $\xi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ is a* homomorphic encryption scheme *with respect to $\mathcal{H}$, if $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a public key encryption scheme, and in addition for every $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$, $h \in \mathcal{H}_k$ and message $M \in \mathcal{M}$*

$$\mathsf{Dec}_{sk}\big(\mathsf{Eval}_{pk}(h, \mathsf{Enc}_{pk}(M))\big) = h(M)$$

We say that ξ satisfies strong (statistical) function-privacy *if for every $h \in \mathcal{H}_k$, pk in the range of $\mathsf{Gen}(1^k)$ and $M \in \mathcal{M}$, $\mathsf{Eval}_{pk}(h, \mathsf{Enc}_{pk}(M)) \overset{s}{\approx} \mathsf{Enc}_{pk}(h(M))$.*

We say that ξ satisfies weak (statistical) function-privacy *if for every $h, h' \in \mathcal{H}_k$, pk in the range of $\mathsf{Gen}(1^k)$ and $M \in \mathcal{M}$, if $h(M) = h'(M)$ then $\mathsf{Eval}_{pk}(h, \mathsf{Enc}_{pk}(M)) \overset{s}{\approx} \mathsf{Eval}_{pk}(h', \mathsf{Enc}_{pk}(M))$.*[8]

[7] Requiring the secret key to be at most k prevents trivialities such as making the key so big that $L(k)$-sized circuits don't have time to read it. In fact, our scheme will satisfy a slightly stronger notion which is that the key generation algorithm will be completely independent of N and L, see Definition 5.

[8] One can naturally define *computational* versions of weak and strong function privacy, in which case one needs to allow the distinguisher to get the secret key as part of the input, and in some applications also the randomness used to generate this secret key.

We say that a scheme is *fully homomorphic* if **(1)** for every polynomial $s = s(k)$ it is homomorphic with respect to the family $\mathcal{H} = \{\mathcal{H}_k\}$, where $\mathcal{H}_k$ is the set of all Boolean circuits of size at most $s(k)$, and **(2)** the running time (and hence also output size) of both the encryption and decryption algorithm is a fixed polynomial in the security parameter k. It was a longstanding open problem to come up with even a plausible candidate for such a scheme, until this was achieved this year by Gentry [15], who gave such a candidate based on ideal lattices.[9] If a scheme satisfies only **(1)** (but not necessarily **(2)**) then we say that it is *size-dependent* homomorphic encryption. There is a trivial construction of a size dependent homomorphic encryption: just have Eval concatenate the circuit to the ciphertext. Using Yao's garbled circuit construction and two-message OT one can get a size-dependent homomorphic encryption with weak function privacy. In contrast, *strong* function privacy for this class $\mathcal{H}$ implies condition **(2)**.

As mentioned in Section 1.2, we observe that a homomorphic encryption scheme with respect to a class $\mathcal{H}$ that is strongly function-private and is circular secure, is also KDM secure with respect to the same class. This already implies that Gentry's scheme is fully KDM secure under certain assumptions that do not refer to *full* KDM security (i.e., hardness of a certain bounded-distance decoding problem on ideal lattices, a sparse subset sum problem, and assuming the scheme is *circular* secure). Moreover, for this application we can relax the condition to *weak* function-privacy:

Theorem 5. *Suppose that there is a homomorphic encryption scheme with respect to a class $\mathcal{H}$ that is weakly function private and* 1*-circular secure. Then there is a single-key KDM-secure scheme with respect to the same class $\mathcal{H}$.*

Proof Sketch. Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be the homomorphic encryption scheme. Our encryption scheme $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ will be as follows:

Key Generation. $\mathsf{Gen}'(1^k)$ runs $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$ and outputs the same secret key sk and as public key the concatenation of pk and $C = \mathsf{Enc}_{pk}(sk)$.

Encryption. $\mathsf{Enc}'_{pk,C}(M)$ outputs $\mathsf{Eval}_{pk}(const^M, C)$, where $const^M$ is the constant function that always outputs M.

Decryption. We have $\mathsf{Dec}' = \mathsf{Dec}$.

Correctness follows easily from the homomorphic property. For security, consider a KDM attacker, that queries an oracle with h and gets back $\mathsf{Enc}'_{pk,C}(h(sk)) = \mathsf{Eval}_{pk}(const^{h(sk)}, C)$. We proceed by a hybrid argument. Suppose that the oracle was changed so that it returned $\mathsf{Eval}_{pk}(h, C)$. Since C is an encryption of sk, and obviously $const^{h(sk)}(sk) = h(sk)$, and the scheme satisfies weak function-privacy this will not change the attacker's output distribution (Since we need the secret key to compute $const^{h(sk)}$, we will need here to use the fact that weak function-privacy holds even with respect to distinguishers that know the secret key).

[9] The fully homomorphic version of Gentry's scheme requires three assumptions: hardness of a certain bounded-distance decoding problem on ideal lattices, hardness of a sparse version of subset sum, and circular security of his basic ideal-lattice based scheme.

The new oracle, however, can be simulated by the attacker on its own (since it does not use the secret key at all, but only h and C). Hence, we complete the proof by appealing to the circular security of the encryption, to argue that C might as well be an encryption of "junk". ∎

As a corollary, assuming the *circular* security of a version of Paillier's cryptosystem [26, 11], the homomorphic PKE construction from [20] yields a KDM-secure encryption scheme with respect to the class of branching programs of a bounded (polynomial) *length*, but unbounded (polynomial) *size*. In other words, the length of the ciphertexts should only depend on the length of branching programs computing the KDM function but not on their size. Compared to the alternative based on the circular-secure version of Gentry's scheme, the conclusion is much weaker but the assumption is different (and seemingly more conservative).

3 Targeted Encryption

The main tool we use to realize our KDM secure scheme is a new notion we call *targeted encryption*. This is a variant of a public key encryption scheme that has the following curious property: the encryption algorithm gets, apart from the message x, two additional inputs: an index $i \in [k]$ (where k is the bit length of the secret key), and a bit b. The decryption algorithm successfully retrieves x if the i^{th} bit of the secret key is b, but otherwise gets no information about x.[10]

Definition 3 (Targeted encryption). *An* targeted encryption scheme TES *consists of a tuple of algorithms* $(\mathsf{TGen}, \mathsf{TEnc}, \mathsf{TDec})$ *such that on security parameter k,* TGen *outputs a pair (pk, sk) with $sk = (sk_1, \ldots, sk_k) \in \{0,1\}^k$ and:*

Targeted decryption. *For every message $x \in \{0,1\}^n$ and index $i \in [k]$,*

$$\mathsf{TDec}_{sk}(\mathsf{TEnc}_{pk,i,sk_i}(x)) = x\ .$$

I.e., it is possible for a sender, given (i, b), to encrypt a message x such that the following hold: if the i^{th} bit of the secret key is b, then the receiver decrypts this message successfully.

(Statistical) security against receiver. *For every $x, x' \in \{0,1\}^n$ and index $i \in [k]$,*

$$\mathsf{TEnc}_{pk,i,1-sk_i}(x) \overset{\mathrm{s}}{\approx} \mathsf{TEnc}_{pk,i,1-sk_i}(x')\ .$$

I.e., if the i^{th} bit of the secret key is not b, then the receiver gets no information about the message x.[11]

[10] We do not actually require a targeted encryption to also have a standard ("untargeted") encryption algorithm, that always succeeds although this can easily be achieved by, say, concatenating two encryptions using parameters $i, 0$ and $i, 1$. Later, to achieve multiple-key security, we will need to assume such an algorithm with particular properties, see Section 5.

[11] For our purposes we can relax this notion to computational indistinguishability with respect to distinguishers that get the secret key as additional input.

Security against outsiders. *For every $x, x' \in \{0,1\}^n$, index $i \in [k]$, and $b \in \{0,1\}$,*

$$pk \circ \mathsf{Enc}_{pk,i,b}(x) \overset{c}{\approx} pk \circ \mathsf{Enc}_{pk,i,b}(x') \ .$$

I.e., outsiders, who do not know the secret key, get no information about the encryption, even if the i^{th} bit of sk does equal b.

The next theorem states that targeted encryption scheme can be obtained from either the DDH or the LWE assumptions.

Theorem 6. *Suppose that (1) the DDH Assumption holds,* or *(2) the LWE assumption holds (with certain parameters),*[12] *then there exists a targeted encryption scheme.*

Theorem 6 is proven by showing that targeted encryption is implied by both the work of Boneh et al. [7] and the work of Applebaum et al. [3] (see the full version [5] of this paperfor the formal proof). The idea of the proof is as follows. Both works give schemes that are KDM secure with respect to affine functions over $\mathbb{Z}_q^k$ for some number q, where k being the secret key size. Their proofs,[13] however, actually give the following stronger homomorphic property: there exists an algorithm Eval that gets the public key and an affine function $h : \mathbb{Z}_q^k \rightarrow \mathbb{Z}_q^k$, and outputs an encryption of $h(sk)$ that is statistically indistinguishable from $\mathsf{Enc}_{pk}(h(sk))$. Note that this is a property that indeed immediately implies KDM security for affine functions. We will use this property to get the following targeted encryption scheme: to encrypt a message $x \in \{0,1\}^n$ so that it can only be decrypted if the i^{th} bit of the key is b, we view x as an element inside $\mathbb{Z}_q$ (using some natural embedding for large enough q, where if n is too large, we encrypt x in chunks) and choose a random $r \in \mathbb{Z}_q$ and use the encryption of scheme to encrypt $r \cdot (sk_i - b) + x$. Note that this is an affine function of sk,[14] and its value is independent of x if $sk_i \neq b$, but is equal to x otherwise. Some complications arise from the fact that in [7] the group is actually given "in the exponent", where in [3] the key is not a bit string, but rather a vector in $\mathbb{Z}_q^k$. Nevertheless, these issues can be easily handled in both cases.

Discussion — Targeted encryption and oblivious transfer. Recall that in a (one out of two) *oblivious transfer* (OT) protocol, a sender holds a pair of values (x_0, x_1), and a receiver has a bit b. At the end of the protocol, the receiver learns x_b and learns nothing about x_{1-b}, while the sender learns nothing about b. A *two-message* OT protocol is one that consists of only two messages — the first from the receiver and the second from the sender. It is easy to see that any two-message OT implies a public-key cryptosystem (with the first message

[12] The exact group for DDH and parameters for LWE are inherited from the assumptions [7, 3]; one important note is that we need to assume LWE for a prime modulus that is polynomial in the security parameter.

[13] In [7]'s case, the above is true for what they call their "expanded" scheme.

[14] Indeed this is the function $h(sk) = \langle \overline{r}, sk \rangle + x'$, where $\overline{r}_i = r$, $\overline{r}_j = 0$ for $j \neq i$, and $x' = x - b \cdot r$.

being the public key); in addition, almost all popular candidates for public-key cryptosystems imply two-message OT protocols.

A targeted encryption scheme can be thought of as a type of “self-referential” OT where the receiver’s input selection bits are equal to the secret information it keeps after its first message (i.e., the secret key). It does not satisfy, however, the standard notion of OT, since the sender is not guaranteed to learn *nothing* about this secret key (although the “security against outsiders” property does imply that the sender cannot recover it completely). We note that it is possible (though we do not need to use this fact in this paper) to transform an OT with such a guarantee into a full-fledged OT, using the techniques of [16, 10].

4 Our Bounded KDM Secure Construction

Let k be the security parameter. Let $\mathsf{TES} = (\mathsf{TGen}, \mathsf{TEnc}, \mathsf{TDec})$ be a targeted encryption scheme. We will construct the following PKE scheme $\mathsf{bKDM} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ that is parameterized over polynomials N and L.

Key generation. $\mathsf{Gen}(1^k)$ samples and outputs $(pk, sk) \leftarrow \mathsf{TGen}(1^k)$.

Encryption. $\mathsf{Enc}_{pk}(M)$ chooses $2k$ random strings $\overline{K} = (K_{i,b})_{(i,b)\in[k]\times\{0,1\}}$ and computes the garbled circuit transformation on $\overline{K}$ and the constant function $const^M$ that outputs M on *every* input $x \in \{0,1\}^k$. We use S, which is some fixed polynomial $S(N, L)$ to be specified later, as the size parameter for the garbled circuit transformation. Let GC be the resulting output. Enc also computes for every $(i, b) \in [k] \times \{0, 1\}$ the value $\tilde{K}_{i,b} = \mathsf{TEnc}_{pk,i,b}(K_{i,b})$ and outputs $C := (GC, (\tilde{K})_{i,b})_{(i,b)\in[k]\times\{0,1\}})$ as the ciphertext.

Decryption. $\mathsf{Dec}_{sk}(GC, (\tilde{K}_{i,b})_{i,b})$ parses $sk = (sk_1, \ldots, sk_k) \in \{0,1\}^k$ and computes the value $K_i = \mathsf{TDec}_{sk}(\tilde{K}_{i,sk_i})$ for every $i \in [k]$. Then, it outputs the result of evaluating the garbled circuit GC on $K_1, \ldots, K_k$.

It is easily verified that the decryption will indeed output $const^M(sk) = M$. We would like to emphasize that key generation does *not* depend on L or N, only encryption does. Hence, we can generate and distribute keys even without knowing L and N in advance.

4.1 Single-Key Security of the Construction

We now show that bKDM is KDM secure for the case of a single key (i.e., $N = 1$). In Section 5, we show that if the underlying targeted encryption scheme is circular secure (when suitably interpreted as a PKE scheme), bKDM actually is secure for an arbitrary number of keys.

Theorem 7. *If* $\mathsf{TES} = (\mathsf{TGen}, \mathsf{TEnc}, \mathsf{TDec})$ *is a targeted encryption scheme, then for every polynomial* L*,* bKDM *instantiated with* $S(N, L) = L$ *is* $(1, L)$*-bounded KDM secure.*

Proof. Fix $N = 1$, an arbitrary L and a PPT adversary A on bKDM’s bounded KDM security. In order to keep the notations simple, we concentrate on the single query case (i.e., the attacker only asks a single key related query). The multi query case, however, easily follows from the same lines. We proceed in games. Let X_i be A's output in Game i, and write $X_i \approx X_j$ as a shorthand for $\mathsf{Pr}[X_i = 1] - \mathsf{Pr}[X_j = 1] \in \mathsf{negl}$. See Table 1 for an overview of all games used in the proof. In all the following games, (sk, pk) are chosen at random using TGen and the oracles get $h : \{0,1\}^k \to \{0,1\}^k$ as input.

Game 0 is the real KDM game. Namely, the oracle $\mathsf{Real}_{pk,sk}$ returns the ciphertext $\mathsf{Enc}_{pk}(h(sk))$. Recall that this is computed by (1) choosing a random $2k$ key tuple $\overline{K}$, (2) encrypting the keys using TEnc to obtain a tuple of ciphertexts $\tilde{K}$ where $\tilde{K}_{i,b} = \mathsf{TEnc}_{pk}(K_{i,b})$ for every $(i, b) \in [k] \times \{0, 1\}$, and (3) computing $GC = \mathsf{Garble}(\overline{K}, const^{h(sk)})$. Ciphertext is $C := (GC, \tilde{K})$.

In **Game** 1 the oracle sets $\tilde{K}_{i,b}$ to $\mathsf{TEnc}_{pk}(0^k)$, instead of $\mathsf{TEnc}_{pk}(K_{i,b})$, for every (i, b) with $sk_i \neq b$. (Note that we still use the original $\overline{K}$ in the garbled circuit construction.) Since GC is independent from the random coins used to encrypt $\tilde{K}$, the "security against receiver" property of TES yields that $X_0 \approx X_1$.

In **Game** 2 the oracle uses h instead of $const^{h(sk)}$ in the garbled circuit construction (i.e., it computes $GC = \mathsf{Garble}(\overline{K}, h)$). Since $h(sk) = const^{h(sk)}(sk)$ and only the keys $\overline{K}_{sk} = (\overline{K}_{i,sk_i})_i$ are used outside the garbled circuit construction, the security against receiver of the garbled circuit construction yields that $X_1 \approx X_2$. We note that the only role of the secret key in this game, is for deciding which elements of $\tilde{K}$ are replaced by encryptions of 0^k.

In **Game** 3 we go back to using the original $\tilde{K}$ (also for the (i, b) with $b \neq sk_i$). Again, the "security against receiver" property of TES implies that $X_2 \approx X_3$. Note that in this game the encryption oracle does not use the secret key *at all.*

We define **Game** 4 to be the variant of **Game** 3 in which we set $\tilde{K}_{i,b} = \mathsf{TEnc}_{pk}(0^k)$ for every i, b (i.e., we ignore the value of $\overline{K}$ for this part). Since the secret key is never used in either Game 3 or Game 4, the "security against

Table 1. Overview of the games used in the proof of Theorem 7. We use $\boxed{\text{boxes}}$ to highlight the component that changed from the previous hybrid, and note in the remark the justification for the fact that the hybrid is indistinguishable from the previous one.

Game	Oracle needs	$\tilde{K}_{i,sk_i}$	$\tilde{K}_{i,1-sk_i}$	Function in GC	Remark
0	pk, sk	$\mathsf{TEnc}(K_{i,sk_i})$	$\mathsf{TEnc}(K_{i,1-sk_i})$	$const^{h(sk)}$	Real KDM game
1	pk, sk	$\mathsf{TEnc}(K_{i,sk_i})$	$\boxed{\mathsf{TEnc}(0^k)}$	$const^{h(sk)}$	TES's sec. ag. recv.
2	pk, sk	$\mathsf{TEnc}(K_{i,sk_i})$	$\mathsf{TEnc}(0^k)$	$\boxed{h}$	GC's sec. ag. recv.
3	$\boxed{pk}$	$\mathsf{TEnc}(K_{i,sk_i})$	$\boxed{\mathsf{TEnc}(K_{i,1-sk_i})}$	h	TES's sec. ag. recv.
4	pk	$\boxed{\mathsf{TEnc}(0^k)}$	$\boxed{\mathsf{TEnc}(0^k)}$	h	TES's sec. ag. outs.
5	pk	$\mathsf{TEnc}(0^k)$	$\mathsf{TEnc}(0^k)$	$\boxed{const^{0^k}}$	GC's sec. ag. outs.
6	pk	$\boxed{\mathsf{TEnc}(K_{i,sk_i})}$	$\boxed{\mathsf{TEnc}(K_{i,1-sk_i})}$	$const^{0^k}$	TES's sec. ag. outs.
					fake KDM game

outsiders" property of the TES implies that $X_3 \approx X_4$. Note that in this game, the vector $\overline{K}$ is independent of $\tilde{K}$.

In **Game** 5 we change h to const^{0^k} in the garbled circuit construction. Since no information on the key vector $\overline{K}$, except for the garbled circuit itself, is given in both oracles, the "security against outsiders" property of the garbled circuit construction implies that $X_4 \approx X_5$. (Note that we need to use the "security against outsiders" and not the "security against receiver" property, since obviously we have no guarantee that $h(sk) = \mathit{const}^{0^k}(sk)$.)

We define **Game** 6 to be the game in which we go back to using the real $\tilde{K}$. Since the oracles do not use the secret key, we get that $X_5 \approx X_6$. Observe that the encryption oracle is exactly the Fake oracle as per Definition 1, and hence we have completed the proof. ∎

5 Multiple Key Security

While the notion of KDM security for a single key is challenging and elegant, many of the applications actually require KDM security in the presence of arbitrarily (polynomially) many keys. Hence, let now the number of keys N be an arbitrary polynomial in the security parameter. We will prove that our scheme bKDM from Section 4 is (N, L)-bounded KDM secure, but now under an additional assumption, and with different parameters.

Complication and central idea. Recall the proof of Theorem 7. There, we have first substituted the function $\mathit{const}^{h(sk)}$ that is evaluated by GC by the KDM query function h itself. By the secrecy against receiver property of the garbled circuit, we could argue that this change goes unnoticed by the receiver. This modification was a crucial step in our proof, since it allowed to construct the garbled circuit without knowing sk. Recall that in multiple secret keys case, the security is defined with respect to N public and secret keys pairs $((\overline{pk}_1, \overline{sk}_1), \dots, (\overline{pk}_N, \overline{sk}_N))$, and the adversary gets encryptions of a query function $h = h(\overline{sk})$ under arbitrary $\overline{pk}_\mu$ for $\mu \in [N]$. Hence, we cannot simply substitute $\mathit{const}^{h(\overline{sk})}$ with h directly (the secrecy against receiver property of the garbled circuit would not help in this case, since we cannot claim that $h(\overline{sk}_\mu) = \mathit{const}^{h(\overline{sk})}$).[15] Instead, we will substitute $\mathit{const}^{h(\overline{sk})}$ with a function h' for which $h'(\overline{sk}_\mu) = h(\overline{sk})$. This function h' contains an encryption of $\overline{sk}$ under the receiver's public key $\overline{pk}_\mu$. In this, we will have to interpret bKDM's underlying targeted encryption scheme TES as a *circular-secure* encryption scheme. (Circular security is required to guarantee that we can later replace these encryptions of secret keys with 0^k-encryptions.) Since our targeted encryption scheme instance is based on the clique-secure encryption schemes of [7, 3], it already has this property. The remaining part of the proof follows the proof of Theorem 7.

Definition 4 (Augmented targeted encryption). *An* augmented targeted encryption scheme $\mathsf{ATES} = (\mathsf{TGen}, \mathsf{TEnc}, \mathsf{TDec}, \mathsf{Enc}, \mathsf{Dec})$ *is a targeted*

[15] $h(\overline{sk}_\mu)$ is not even well defined; h is expecting a vector of (secret) keys as input, and not a single key.

encryption scheme $(\mathsf{TGen}, \mathsf{TEnc}, \mathsf{TDec})$*, complemented by* PPT *algorithms* Enc*,* Dec *for (un-targeted) encryption and decryption. We require that* $(\mathsf{TGen}, \mathsf{Enc}, \mathsf{Dec})$ *is a public-key encryption scheme with message space* $\mathcal{M} \subseteq \{0,1\}^k$*. In particular,* $\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(M)) = M$ *for all* $(pk, sk) \leftarrow \mathsf{TGen}(1^k)$ *and* $M \in \{0,1\}^k$*.*

We say that ATES is circular secure if $(\mathsf{TGen}, \mathsf{Enc}, \mathsf{Dec})$ is. We stress that our both TES instances from Theorem 6 are circular secure augmented targeted encryption schemes with the natural encryption and decryption algorithms from Boneh et al. [7] and Applebaum et al. [3] respectively. The following theorem implies our main result (i.e., Theorem 1). We provide a proof in the full version [5] of this paper.

Theorem 8. *If* $\mathsf{ATES} = (\mathsf{TGen}, \mathsf{TEnc}, \mathsf{TDec}, \mathsf{Enc}, \mathsf{Dec})$ *is a circular secure augmented targeted encryption scheme, then for every polynomials* L *and* N*,* bKDM *instantiated with a suitable polynomial* $S(N, L)$ *is bounded KDM secure.*

6 Application to Formal Cryptography

One of the main motivations to study KDM security lies in the connection between formal and computational cryptography. In formal cryptography (starting with [12, 13, 23]), cryptographic operations like encryption or digital signatures are abstracted as symbolic operators that (only) adhere to natural rules like $D_K(E_K(M)) = M$ for symmetric encryption and decryption operators E and D. A simple calculus like this enables machine-assisted security analysis (e.g., [21, 22]). It is not a priori clear, however, that security properties proved in the symbolic calculus also hold for the computational implementation of the protocol.

Computational soundness. Abadi and Rogaway [1] were the first to relate the formal and computational views on cryptography. Specifically, they showed that every symbolically proven property also holds in the computational world, assuming a suitable computational implementation. This is usually referred to as a soundness result, and suitable computational implementations are dubbed sound. In order to provide computational soundness in this sense in face of a passive adversary, an encryption scheme essentially needs to be IND-CPA secure.

Key-cyclic expressions. There is a technical nuisance, however, that limits the generality and expressivity of [1]'s approach. Namely, the soundness result only holds for protocols that do *not* contain key-cyclic expressions. That is, only protocols in which no expressions with cyclic dependencies of encryption keys (such as $E_{K_1}(E_{K_2}(K_1))$) appear are considered. This is for the following reason: in the symbolic setting, the natural deduction rules explicitly require secret keys for decryption. Hence, the encrypted plaintexts in such expressions are secret by definition in the symbolic world (i.e., there is no formal way to apply, say, D_{K_2} on the ciphertext $E_{K_1}(E_{K_2}(K_1))$). On the other hand, key-dependent messages like the one above, are not modeled in standard (computational) security experiments for encryption schemes.[16] Hence, there is an asymmetry between symbolic and computational setting, and any soundness result that connects symbolic encryption and *standard* computational encryption notions has to exclude key-cyclic expressions.

Soundness from Bounded KDM Security. It was informally claimed by Black et al. [6], and formally proven by Adão et al. [2], that *fully* KDM-secure encryption schemes imply computational soundness for *arbitrary* symbolic protocols. Since we can only achieve bounded KDM security against arbitrary circuits up to a certain size, we ask whether bounded KDM security suffices for computational soundness of arbitrary symbolic protocols. The answer we give is essentially affirmative.

To do so, we introduce the following slight strengthening of bounded KDM security:

Definition 5 (Length-dependent bounded KDM security). *A PKE scheme with message space $\mathcal{M} \subseteq \{0,1\}^*$ is N*-key length-dependent bounded KDM secure*, if it is KDM secure with respect to the circuit class of all $h : (\{0,1\}^k)^N \to \{0,1\}^{\lfloor\sqrt{|h|}\rfloor}$, where $|h|$ is the circuit size of h.*

That is, length-dependent KDM secure schemes are secure against larger KDM queries if longer messages are encrypted. We stress that our scheme bKDM from Section 4 is N-key length-dependent bounded KDM secure, if we choose L suitably (e.g., $L = |M|^{2.1}$) during encryption. Namely, bKDM's key generation algorithm does not depend on N or L, and the proofs of Theorems 7 and 8 do not use that L is fixed.

Theorem 9 (Following [1, 2]: Bounded KDM security implies soundness). *Let* bKDM *be an N-key length-dependent KDM secure PKE scheme, and let P be a symbolic protocol with N parties in the setting of Adão et al. [2]. Then* bKDM *provides a computationally sound implementation of P.*

We stress that the choice of symbolic setting [2] was made only for simplicity. We provide a proof outline in the full version [5] of this paper.

Application of our results. Theorem 9 can be instantiated with our scheme bKDM from Section 4. (As argued above, bKDM actually is N-key length-dependent bounded KDM secure.) This yields the first encryption scheme that provides soundness under a standard computational assumption.

Relation to circular security and extensibility. For extremely simple calculi that only feature public-key encryption, along with a few syntactic operations like pairings of terms, already clique security may enable soundness. (This is so since all key-dependent encryptions that can possibly occur in a symbolic

[16] Some subsequent soundness results (e.g., [4, 24]) consider an active adversary and require IND-CCA security. We stress that does not change the technical complications regarding key-cyclic expressions.

protocol can be traced back to simple encryptions of the form $E_{K_i}(K_j)$, assuming a suitable way to encrypt longer terms in chunks.) Nevertheless, we stress that our scheme bKDM allows much richer classes of calculi. For instance, the above soundness proof also works in the presence of signatures, so that terms of the form $E_{K_i}(sig(K_j), K_\ell)$ may occur. (The crucial observation is that we can suitably pad, e.g., signatures such that the signing algorithm can be expressed as a length-dependent KDM circuit.) On the other hand, clique security, or even security against a polynomial number of arbitrary but predetermined KDM functions is *not* sufficient to treat such richer classes of calculi.

7 Extending Haitner and Holenstein's Impossibility Result

In this section we observe that a result of Haitner and Holenstein [17], showing that there is no KDM-secure scheme with a proof of security which makes a black-box use of both the adversary and the KDM function, can be extended to rule out not just *full* KDM security but also *bounded* KDM security. The idea is simple: while this result used a *random* function h for the KDM function, a *pseudorandom* function could work just as well.

The following definition is adopted from [17].

Definition 6 (Cryptographic games). *A* cryptographic game *is a (possibly inefficient) random system Γ, where on security parameter k, Γ interacts with an attacker A and may output* 1*. We define the* game value *of such an interaction, denoted $\Gamma_A(1^k)$, as the probability that Γ outputs* 1 *in the end of the interaction with A, where the probability is taken over the random coins of Γ and A. A cryptographic game is* non-interactive *if it consists of two messages, from Γ to A and back.*

Examples:

OWF. The security of a one-way function f is equivalent to requiring that the value of the following game is negligible for any efficient A. On security parameter k, the system Γ selects a random $x \in \{0,1\}^k$ and sends $y = f(x)$ to the adversary. Γ outputs 1 if A outputs $x' \in f^{-1}(y)$.

DDH. The security of the DDH hardness assumption is equivalent to requiring that the value of the following game is at most negligibly bounded from $\frac{1}{2}$ for any efficient adversary. Let G be an appropriate DDH group (e.g., Z_p^* for some prime p) and let g be a generator in the group. The system Γ chooses a random bit b, sends the tuple (g^x, g^y, g^z) to the challenger A, where x and y are random exponents, and $z = x \cdot y$ if $b = 0$ and a random value otherwise. Γ outputs 1 iff A has guessed b correctly.

Definition 7 (Strongly-black-box reductions). *An encryption scheme* $(\mathsf{Enc}, \mathsf{Dec})$ *has a δ-*strongly-black-box *reduction from its KDM security to a cryptographic game Γ with respect to a family of query function $\mathcal{Q}$, if there exists an* *oracle-aided algorithm R with the following guarantee: Let A be an efficient adversary that breaks the KDM security of the scheme using query functions from $\mathcal{Q}$ with advantage $\epsilon_A = \epsilon_A(k)$ (i.e., On security parameter k, A distinguishes between encryptions of functions of the secret key and encryptions of garbage with advantage $\epsilon_A(k)$). Then the value of $\Gamma_{R^A}(1^k) \geq \delta(k, \epsilon_A(k))$. Here, we require that R treat the query functions it gets from A as black boxes — all it can do is to query them on arbitrary chosen inputs.*

Informally, we say that a proof for the KDM security of a scheme is strongly-black-box with respect to a game Γ and a family of query function $\mathcal{Q}$, if the value of $\delta(k, \epsilon_A(k))$ for every non-negligible $\epsilon_A(k)$ is considered a "break" of Γ (i.e., $\delta(k, \epsilon_A(k)) > \frac{1}{2} + \mathsf{negl}$ for the DDH game). We remark that *all* known KDM constructions in the literature have strongly-black-box reductions with respect to the relevant hardness assumption (e.g., DDH) and the class of query functions they are secure against.

Definition 8 (Pseudorandom functions (PRF)). *An ensemble of functions $\mathcal{F} = \{\mathcal{F}_k = \{f \colon \{0,1\}^{m(k)} \mapsto \{0,1\}^{\ell(k)}\}\}$ is* pseudorandom*, if, on security parameter k, an efficient adversary cannot distinguish with more than negligible advantage between a random $f \in \mathcal{F}_k$, and a truly random function defined on the same input and output domains. Here, the adversary may only access the function as a black box. The ensemble is α-*exponential hard *for a constant $\alpha > 0$, if no adversary that runs in time 2^{n^α} wins in the above game with advantage greater than 2^{k^α}.*

Theorem 10. *Let $(\mathsf{Enc}, \mathsf{Dec})$ be a δ-strongly-black-box reduction from its KDM security to a non-interactive cryptographic game Γ with respect to a family of query functions $\mathcal{Q} = \{\mathcal{Q}_k\}$.*[17] *Assume that $\mathcal{Q}_k$ contains the family of functions $\mathcal{G}_k = \{g_k(x) = f(x, 0^{t(k)-k}) \oplus r \colon f \in \mathcal{F}_{t(k)}, r \in \{0,1\}^{\ell(k)}\}$, where $t(k) \geq 2k$ and $\mathcal{F}_{t(k)} = \{f \colon \{0,1\}^{t(k)} \mapsto \{0,1\}^{\ell(k)}\}$ is an α-exponential hard PRF with $t(k)^\alpha \geq 2k$. Then, there exists an efficient algorithm A with $\Gamma_A \geq \delta(k, 1 - 2^{-k}) - 2^{-k}$.*

In particular, giving such a strongly-black-box reduction implies that either the class of query function considered is weak (does not contain exponentially hard PRF), or the game Γ can be efficiently broken with probability $\delta(k, 1-2^{-k}) - 2^{-k}$.

Proof (sketch). The proof is similar to the proof of [17, Theorem 5]. Consider the following (inefficient) adversary A for breaking the KDM security of $(\mathsf{Enc}, \mathsf{Dec})$ with respect to $\mathcal{Q}$. On security parameter k, choose a random $g \in \mathcal{G}_k$ and make a KDM query to obtain a ciphertext C. Then check (via exhaustive search) if there exists $sk \in \{0,1\}^k$ such that $\mathsf{Dec}_{sk}(C) = g(sk)$. If positive return 1, otherwise return 0. It is easy to verify that A breaks the KDM security with advantage $1-2^{-k}$ (the probability that a decryption of random ciphertext C equals to $g(sk)$

[17] Theorem 10 can be shown to hold also against all natural interactive games (see [17] for details). For the sake of simplicity, however, we choose to focus here on the non-interactive case.

for some sk, is bounded by $\sum_{sk} \Pr[\mathsf{Dec}_{sk}(C) = g(sk)] = \sum_{sk} \Pr_r[\mathsf{Dec}_{sk}(C) = f(sk) \oplus r] \leq 2^k \cdot 2^{-2k} = 2^{-k}$). More interestingly, we notice that the probability that R^A sends a ciphertext $C = \mathsf{Enc}_{sk}(g(sk))$ to A without previously making the query $g(sk)$ is bounded by 2^{-2k}. Assume otherwise, then R^A is an algorithm that runs in time poly $\cdot 2^k$ and breaks the security of $\mathcal{F}$. It follows that we can emulate the execution of R^A: throughout the execution, keep track of all queries that R makes to g, and let T denote the list of queries. When R queries A on a ciphertext C, act as the inefficient A above, but *only* with respect to the secret keys in T. The above observation yields that we emulate R^A with error bounded by 2^{-2k}. ■

References

[1] Abadi, M., Rogaway, P.: Reconciling two views of cryptography (the computational soundness of formal encryption). Journal of Cryptology 15(2), 103–127 (2002)

[2] Adão, P., Bana, G., Herzog, J., Scedrov, A.: Soundness of formal encryption in the presence of key-cycles. In: di Vimercati, S.d.C., Syverson, P.F., Gollmann, D. (eds.) ESORICS 2005. LNCS, vol. 3679, pp. 374–396. Springer, Heidelberg (2005)

[3] 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)

[4] Backes, M., Pfitzmann, B., Waidner, M.: A composable cryptographic library with nested operations (extended abstract). In: ACM CCS 2003, pp. 220–230 (2003); Full version in IACR Cryptology ePrint Archive 2003/015 (January 2003), http://eprint.iacr.org/

[5] Barak, B., Haitner, I., Hofheinz, D., Ishai, Y.: Bounded key-dependent message security. Cryptology ePrint Archive, Report 2009/511 (2009), http://eprint.iacr.org/

[6] 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)

[7] 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)

[8] Brakerski, Z., Goldwasser, S., Kalai, Y.: Circular-secure encryption beyond affine functions. Cryptology ePrint Archive, Report 2009/485 (2009), http://eprint.iacr.org/

[9] 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, p. 93. Springer, Heidelberg (2001)

[10] Crépeau, C., Kilian, J.: Weakening security assumptions and oblivious transfer. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 2–7. Springer, Heidelberg (1990)

[11] Damgård, I., Jurik, M.J.: A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In: Kim, K.-c. (ed.) PKC 2001. LNCS, vol. 1992. Springer, Heidelberg (2001)

[12] Dolev, D., Yao, A.C.: On the security of public key protocols. IEEE Transactions on Information Theory 29(2), 198–208 (1983)

[13] Even, S., Goldreich, O.: On the security of multi-party ping-pong protocols. In: FOCS 1983 (1983)

[14] Even, S., Goldreich, O., Lempel, A.: A randomized protocol for signing contracts. Commun. ACM (1985)

[15] Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC (2009)

[16] Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: STOC (1989)

[17] 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)

[18] Halevi, S., Krawczyk, H.: Security under key-dependent inputs. In: ACM CCS 2007 (2007)

[19] 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)

[20] Ishai, Y., Paskin, A.: Evaluating branching programs on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 575–594. Springer, Heidelberg (2007)

[21] Kemmerer, R., Meadows, C., Millen, J.: Three systems for cryptographic protocol analysis. Journal of Cryptology 7(2), 79–130 (1994)

[22] Lowe, G.: Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In: Margaria, T., Steffen, B. (eds.) TACAS 1996. LNCS, vol. 1055. Springer, Heidelberg (1996)

[23] Merritt, M.: Cryptographic Protocols. PhD thesis, Georgia Institute of Technology (1983)

[24] Micciancio, D., Warinschi, B.: Soundness of formal encryption in the presence of active adversaries. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 133–151. Springer, Heidelberg (2004)

[25] Naor, M.: On cryptographic assumptions and challenges. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 96–109. Springer, Heidelberg (2003)

[26] Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, p. 223. Springer, Heidelberg (1999)

[27] Rabin, M.: How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory (1981)

[28] Yao, A.C.: How to generate and exchange secrets. In: FOCS 1986 (1986)

Perfectly Secure Multiparty Computation and the Computational Overhead of Cryptography

Ivan Damgård[1], Yuval Ishai[2,⋆], and Mikkel Krøigaard[3,⋆⋆]

[1] University of Aarhus, Denmark
ivan@cs.au.dk
[2] Technion and UCLA
yuvali@cs.technion.ac.il
[3] Eindhoven University of Technology
m.kroigaard@tue.nl

Abstract. We study the following two related questions:

- What are the minimal computational resources required for general secure multiparty computation in the presence of an honest majority?
- What are the minimal resources required for two-party primitives such as zero-knowledge proofs and general secure two-party computation?

We obtain a nearly tight answer to the first question by presenting a *perfectly* secure protocol which allows n players to evaluate an arithmetic circuit of size s by performing a total of $\mathcal{O}(s \log s \log^2 n)$ arithmetic operations, plus an additive term which depends (polynomially) on n and the circuit depth, but only logarithmically on s. Thus, for typical large-scale computations whose circuit width is much bigger than their depth and the number of players, the amortized overhead is just polylogarithmic in n and s. The protocol provides perfect security with guaranteed output delivery in the presence of an active, adaptive adversary corrupting a $(1/3 - \varepsilon)$ fraction of the players, for an arbitrary constant $\varepsilon > 0$ and sufficiently large n. The best previous protocols in this setting could only offer *computational* security with a computational overhead of $\mathrm{poly}(k, \log n, \log s)$, where k is a computational security parameter, or perfect security with a computational overhead of $\mathcal{O}(n \log n)$.

We then apply the above result towards making progress on the second question. Concretely, under standard cryptographic assumptions, we obtain zero-knowledge proofs for circuit satisfiability with 2^{-k} soundness error in which the amortized computational overhead per gate is only *polylogarithmic* in k, improving over the $\omega(k)$ overhead of the best previous protocols. Under stronger cryptographic assumptions, we obtain similar results for general secure two-party computation.

1 Introduction

This work studies two different but closely related questions: the complexity of *secure multiparty computation* (MPC) in the presence of an honest majority,

[⋆] Supported by BSF grant 2008411, ISF grant 1310/06, and NSF grants 0830803, 0716835, 0627781.

[⋆⋆] Most of the work done at the University of Aarhus.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 445–465, 2010.

and the complexity of *two-party* cryptographic primitives such as zero-knowledge proofs and secure two-party computation.

1.1 The Complexity of MPC

We consider the question of MPC over secure point-to-point channels in the presence of an active (malicious) adversary, who may corrupt up to some constant fraction δ of the n players. In this work we focus on the case of an honest majority, where $\delta < 1/2$. Unlike the case of MPC with no honest majority, in this case it is possible to guarantee output delivery and provide unconditional security. Following the initial feasibility results of [17,3,8,27], a long sequence of works, initiated by [14,15,19,10], attempted to minimize the communication and computation resources required for general MPC in this setting.

To make the question cleaner and less sensitive to variations in the model, we adopt the following standard conventions. First, to measure the growth of complexity with the number of players, we consider n as a parameter which tends to infinity. A large value of n captures not only computations which combine inputs from many players, but also "cloud computing" scenarios in which a large number n of untrusted or unreliable *servers* are used to distribute computations on inputs that originate from a small number of *clients* or even from just a single client. Second, to eliminate from consideration an *additive* overhead which depends (polynomially) on n and a security parameter[1] but does not grow with the complexity of the functionality f, we assume the circuit complexity of f to be much bigger than n. This is in line with most typical MPC application scenarios, and may capture both complex computations on small inputs and simple computations on massive inputs.

More concretely, we consider the task of securely evaluating a function f represented by a boolean circuit C whose inputs and outputs are arbitrarily partitioned between the n players. We let k denote a security parameter, such that the simulation error of the protocol is bounded by 2^{-k}. (This should hold for computationally unbounded adversaries in the case of statistical security and for 2^k-bounded adversaries in the case of computational security; the parameter k can be ignored in the case of perfect security.) We say that a general MPC protocol has *computational overhead* $c(n, k, s)$ if for all positive integers n, k, s, and circuit C of size s, the total number of bit operations[2] performed by all n players together is at most $s \cdot c(n, k, s) + \mathrm{poly}(n, k, \log s)$. The computational overhead can be thought of as the amortized multiplicative price for achieving security: the ratio between the cost of securely distributing an expensive task between n players and the cost of a centralized (insecure) solution for the same task.

Note that the computational overhead of a protocol implies a similar bound on its *communication* overhead with respect to the circuit size. However, in light

[1] Such an overhead is very sensitive to the underlying network and MPC model, and is required in our settings even for performing the simple MPC task of broadcasting a single bit.

[2] Our count of bit operations includes both local computations and point-to-point communication.

of Gentry's recent candidate for a fully homomorphic encryption scheme [16], the circuit size should no longer be generally seen as a barrier for the communication complexity of MPC. This notion still looks meaningful in the setting of unconditional security or for circuits whose input or output length are comparable to their size. See Section 8 for further discussion.

The computation and communication overhead of the first general MPC protocols [17,3,8] were large polynomials in n, k (e.g., $\mathcal{O}(n^8)$ for a naive implementation of the perfectly secure BGW protocol over a point-to-point network [3,19]). Following a long sequence of works (see [13] for a survey) the current state of the art can be summarized as follows. For simplicity, we do not state the resilience level of each protocol. Using a general protocol composition technique from [6,18,13], all protocols can be made nearly optimally resilient with the same asymptotic overhead.

In the setting of computational security, an overhead of $c(n, k, s) = \text{poly}(k, \log n, \log s)$ was achieved in [13]. This protocol can be realized with a constant number of rounds under standard cryptographic assumptions.

In the case of unconditional security, all efficient MPC protocols from the literature require the round complexity to grow with the circuit depth d. Since all players in these protocols are active in each round, we redefine computational overhead for the unconditional case to allow an additive term of $\text{poly}(n, k, d, \log s)$ (the exponent of d should be extremely low here, or the term can become dominant). The computational overhead of the best *perfectly* secure protocol prior to this work [2] was $n \cdot \text{polylog}(n)$. This protocol has a similar communication overhead. In the case of *statistical* security and protocols which take inputs from and deliver outputs to only a *constant* number of clients (but still distribute the computation among n servers) a variant of the protocol from [11] based on algebraic geometric secret-sharing [9] (see [20,22]) has computation overhead of $k \cdot \text{polylog}(n)$ and communication overhead of $\mathcal{O}(1)$.

This state of the art leaves open several natural questions:

- Can the computational overhead be simultaneously sublinear in both n and k in *any* MPC model? This question turns out to be relevant for the applications discussed in Section 1.3 below.
- Can the computational overhead be sublinear in n with *perfect* security, or alternatively with statistical security even when inputs can originate from all players (as opposed to a constant number of clients as in [11,22])? These questions are open even for the easier case of *communication* overhead.

1.2 Our Results

We present a *perfectly* secure general MPC protocol whose computational overhead is *polylogarithmic* in n, answering the above questions affirmatively.

More concretely, the protocol can tolerate an active, adaptive adversary corrupting up to a $1/3 - \varepsilon$ fraction[3] of the players, for an arbitrary constant

[3] In our model we assume that only point-to-point channels are available, in which case it is impossible to achieve unconditional security with guaranteed output delivery if at least 1/3 of the players can be corrupted.

$\varepsilon > 0$ and all sufficiently large n. The computational (and communication) complexity required for evaluating a boolean circuit C of size s and depth d is $\text{polylog}(n) \log s \cdot s + d^2 \cdot \text{poly}(n, \log s)$. If C is an *arithmetic* circuit over a finite field of size bigger than n, the total computational work involves $\mathcal{O}(\log^2 n \log s \cdot s) + d^2 \cdot \text{poly}(n, \log s)$ arithmetic operations and the communication includes $\mathcal{O}(\log n \log s \cdot s) + d^2 \cdot \text{poly}(n, \log s)$ field elements.

Alternatively, in the case where d^2 is too large, we provide an option to increase the circuit size by a factor $\log d$ while decreasing the d^2 factor to $d \log d$. The intuition is that the first factor on the second term is dX, where X is defined as follows. Dividing the circuit into layers in the natural way, we define the number X to be the maximal number of layers reachable by one wire from any given layer. In general, $X = \mathcal{O}(d)$ and so the factor is d^2. With our alternative approach, $X = \mathcal{O}(\log d)$ and so the factor is $d \log d$. The real calculation is a bit more involved, but this is the basic idea.

Thus, with the above alternative, the computational complexity for an arithmetic circuit becomes $\mathcal{O}(\log^2 n \log s \log d \cdot s) + d \log d \cdot \text{poly}(n, \log s)$, and similarly for the other complexities.

Since the modification of the circuit increases its size by a factor $\log d$, it is not always the best solution. Only for circuits with a large depth is the alternative a good choice. Furthermore, the d^2 factor is the result of a somewhat pessimistic worst-case analysis, and for most typical circuits the additive term grows only linearly with d.

As a final remark about our protocol, it seems "lean" enough to be implemented in practice. This should be contrasted with the previous best protocol from [13], which involves a distributed evaluation of a pseudorandom function for every gate in the circuit.

Techniques. Our protocol employs several techniques that were used in previous works along this line, including the share-packing technique from [14], allowing to secret-share a block of secrets with a low amortize cost, and the efficient verifiable secret sharing protocol from [2,13]. The main technical challenge is to perform "non-homogenous" computations on pairs of blocks, i.e., ones that are different from coordinate-wise addition or multiplication of blocks. We address this challenge by embedding the computation in a special form of a universal circuit based on the so-called Beneš network [5,29]. The high level idea is that the structure of the circuit reduces the computation in a given layer of the circuit to an arbitrary permutation *between* blocks (which can be done locally), homogenous operations, and a logarithmic number of distinct permutations *within* blocks. We propose an efficient procedure for the latter. See Section 4 for a more detailed technical overview.

An independently interesting contribution is a new methodology for the security analysis of honest-majority MPC protocols. Similarly to most protocols of this type, our protocol is composed from subprotocols that generate auxiliary secret shared values to help in the computation, a subprotocol for sharing the inputs, and finally a "layer-protocol" that performs secure computation corresponding to one layer of the circuit, i.e., it starts with the shares of values going into the layer, consumes some auxiliary shared values, and outputs shares of values coming out of the layer. Our proof of security first proves all subprotocols to be UC secure. We then define a functionality $\mathcal{F}_i$ that takes inputs from the players and outputs shares of the values output by the i'th layer of the circuit (where layer 0 just produces the inputs to the circuit). We then show that $\mathcal{F}_0$ can be implemented by calling the auxiliary subprotocols, and $\mathcal{F}_i$ for $i > 0$ can be (UC-)implemented by calling $\mathcal{F}_{i-1}$ and then executing the layer-protocol.

We believe this may be the first example of a general honest-majority MPC protocol with a fully modularized proof of security. The main challenge is that it is non-trivial to define functionalities for the subprotocols such that 1) the subprotocol actually realizes the functionality and 2) the functionality provides what is needed in the larger context. It is well known that even for a simple task such as digital signatures, defining the "right" functionality is not easy.

In our case, the main idea turn out to be that a functionality that is supposed to output shares of some secrets, should not simply choose those shares on its own and send them to the players, although that may seem like the most natural approach. Instead, our functionalities ask the adversary which shares it wants the corrupted players to get, and the functionality then chooses shares for the honest players conditioned on the shares obtained from the adversary and the secret. In a sense, this models the fact that we do not care about the distribution of shares the adversary sees, as long as the secret is safe.

1.3 The Computational Overhead of Cryptography

A somewhat unexpected motivation for this work comes from the recent applications of honest-majority MPC to two-party primitives such as zero-knowledge proofs and general secure two-party computation [20,22]. We note that these general tasks can be used as building blocks for more specialized two-party tasks such as identification or different flavors of signatures.

The computation and communication overhead of standard two-party cryptographic primitives can be defined similarly to the overhead of MPC as defined above, except that here n is viewed as a constant and s corresponds to work required for an insecure implementation (e.g., length of message in case of encryption, or size of witness verification circuit in the case of zero-knowledge). For instance, typical implementations of encryption have a constant communication overhead, but a poly(k) computation overhead.[4] In contrast, for typical implementations of zero-knowledge proofs or secure two-party computation protocols from the literature, both the communication and computation overhead are poly(k).

In [21] it was shown that, under plausible assumptions, various primitives including encryption, signatures, and secure two-party computation in the *semi-honest* model can be implemented with a constant computational overhead. For

[4] Since for the purpose of concreteness we consider attackers that run in time 2^k, this requires to assume that the underlying hardness assumption is 2^{n^ε}-strong for some $\varepsilon > 0$.

primitives such as encryption, commitment, hashing, and signatures, constructions with polylog(k) overhead relying on lattice-based assumptions or error-correcting codes were given in [26,24,1].

Obtaining similar results for zero-knowledge proofs and secure two-party computation against malicious parties is one of the main questions left open in [21]. Combining our main result with general transformations from [20,22], we can make progress on the this question. Concretely, under standard cryptographic assumptions (e.g., assuming 2^{n^ε}-hardness of decoding random linear codes [1]), our main result yields zero-knowledge proofs for circuit satisfiability with 2^{-k} soundness error and simulation error, in which the amortized computational overhead per gate is only *polylogarithmic* in k, improving over the $\omega(k)$ overhead of the best previous protocols under *any* assumptions. Under stronger cryptographic assumptions, we obtain similar results for general secure two-party computation with simulation error 2^{-k}. Both types of protocols are unconditionally secure when implemented in the natural hybrid model (i.e., using ideal commitments in the case of zero-knowledge, or oblivious transfer in the case of secure computation). This implies that all "cryptographic" computations can be done during a preprocessing stage, before the actual inputs are known. See Section 7 for more details.

2 The Model

We consider the standard setting of *perfectly* UC-secure MPC [7], with guaranteed output delivery, over a synchronous network of secure point-to-point channels. Our protocols also employ a *broadcast* primitive, but since the number of broadcasts will be small they can be simulated over point-to-point channels without affecting the amortized overhead.

The players in our protocol are divided into three categories: *input clients* who contribute inputs, *output clients* who receive outputs, and n *servers* who help distribute the computation. To simplify the asymptotic complexity expressions, the number of clients is assumed to be $\mathcal{O}(n)$. Note that a player in the protocol is permitted to have one or more roles, and therefore this client-server model generalizes the usual model where every player has all three roles. The adversary is unbounded, active and adaptive, may corrupt up to t servers and any number of clients, where t is some constant fraction of n. (Concretely, one can use $t = n/8$ in the basic version of our protocol.)

We assume that the functionality f computed by the protocol is described by an arithmetic circuit C over a finite field $\mathbb{Z}_p$, where $p > 2n$. (In the case of boolean circuits, we can use the least p which satisfies this requirement. This results in an additional logarithmic communication overhead and polylogarithmic computation overhead.) The inputs and outputs of C may be arbitrarily partitioned between the input clients and the outputs clients, respectively.

It will be convenient to partition the gates into *layers*, such that each layer gets its input only from the previous layers and provides output to subsequent layers. This can be done by partitioning the gates according to the length of a longest path from an input. The *size* of the circuit C is written as $|C|$, and it is defined to be the number of gates plus the number of wires. Its *depth* is the length of the longest path from an input to an output, which is equal to the number of layers in the case of layered circuits.

Finally, since our efficiency goals are impossible to meet if each server needs to read an entire description of C, we separate the *protocol compilation* from the *protocol execution*. The protocol compiler takes a description of an arithmetic circuit C (whose inputs and outputs are partitioned between the clients) and a number of servers n and generates the "code" of each player in the protocol. When analyzing the complexity of the protocol we count only the cost of the protocol execution (combined over all players), but note that the protocol compilation can be performed with the same asymptotic computational cost as executing the protocol.

3 Packed Secret-Sharing

We will use the packed secret-sharing technique introduced by Franklin and Yung [14]. This is similar to standard Shamir secret-sharing [28] over $\mathbb{Z}_p$, but where a block of l different values $(x_1, .., x_l)$ are shared at once using a polynomial that evaluates to $x_1, ..., x_l$ in l distinct points. For privacy if t players are corrupted, the polynomial must be random of degree at most $d = t + l - 1$. We need that, from a set of n shares, one from each player, where at most t are incorrect, the correct block of secrets can be efficiently determined, even if the polynomial has degree up to $2d$. This will be the case if we set $t = n/8$ and $l = n/4$. Also, to have enough distinct evaluation points, we need that $p > 2n$. This is the same variant of packed secret sharing as was used in [13], which we refer to for further details.

Denote by $[x]_d$ a packed secret-sharing of the block x using a polynomial of degree at most d. Any vector of shares $\{s_1, \ldots, s_n\}$ among n servers is called *d-consistent* if the shares correctly match a degree at most d polynomial in the n first points and therefore uniquely defines a block of secrets.

Throughout the paper we will need many different protocols dealing with block sharings. Most notably we need verifiable secret-sharing for the input and reconstruction with error correction for the output. In Section 5 on page 453 we describe the known protocols that we will use.

4 Overview of the Protocol

Using packed secret sharing, it is straightforward to do secure addition or multiplication on l values in parallel, at the price of what a single operation would cost using normal secret sharing. This was already observed in [14] and can be used to compute the circuit C securely and efficiently if we arrange it such that every layer contains only one type of gates, and if we can produce sets of shared blocks $S_1, S_2, ..$ such that blocks in S_i contain the i'th input bit to the gates in a given layer, in some fixed order. We will call this a *correct line-up* for the given layer.

Demanding correct line-up is a problem, however: It implies that the values in the computation will have to be permuted between layers in arbitrary ways that depend on the concrete circuit. This is not easy to implement efficiently using packed secret sharing. We solve this problem by first constructing from C a new circuit C' that computes the same function but is more well-behaved. More precisely, we have

Lemma 1. *Given an arithmetic circuit C that is at least l gates wide, there is an efficient algorithm to transform it into another circuit C' with the following properties:*

1. *$C'(x) = C(x)$ for all inputs x.*
2. *Every layer contains only one type of gate.*
3. *If all values are stored in blocks using packed secret sharing where the block size l is a 2-power, the action between any two layers to achieve correct line-up is to permute the blocks and then in some blocks permute the elements within the block, where the same permutation applies to all blocks in the layer[5]. In the entire circuit, only $\log l$ different permutations are needed to handle permutations within blocks.*
4. *$|C'| = \mathcal{O}(|C| \log |C| + \text{depth}(C)^2 n \log^3 |C|)$, $\text{depth}(C') = \mathcal{O}(\log^2 |C| \text{depth}(C))$.*

The restriction on the width of the circuit is fairly insignificant, since n is generally small compared to the circuit size. Some of the layers in C' will not be a block wide, but since those layers also do not require a permutation, it will cause no problems.

We show in the full version (available on ePrint) [12] how this construction works in detail. The basic idea is to handle the arbitrary permutations needed in C by inserting a small piece of circuitry that permutes the values as desired. This subcircuit can be made very regular using permutation networks as described by Waksman [29]. These are based on Beneš networks [5]. It follows from the construction that C' only contains addition, multiplication and H-gates, where H swaps two input values x, y or leaves them alone, depending on a control-bit c: $\text{H}(x, y, c) = (cx + (1-c)y, cy + (1-c)x)$.

Now, given the input arithmetic circuit C, we first transform it into C' as described in the lemma. We begin our actual computation by secret-sharing the input values in blocks of size $l = \Theta(n)$, where l is a 2-power, and we then go through C' layer by layer, computing at each stage the output values from the layer in packed secret-shared form. Once we have the output from the last layer, shares of these are sent to the output clients for reconstruction.

Going into each layer we permute the shared blocks we have so far as needed to get correct line-up for the layer, and then do the computation required. The only non-trivial issue is how to permute the elements inside a shared block, i.e., how to compute $[\pi(x)]_d$ from $[x]_d$ for a permutation π. The idea is to first precompute pairs of the form $[r]_d, [\pi(r)]_d$ for random blocks r. We show below how to generate many such pairs using the same π at a small amortized cost per

[5] In some cases, it may additionally be necessary to discard some blocks.

pair. This is sufficient, since by the above lemma, we only need a small number of different permutations. The idea then is to reveal $x + r$ to a single server, who then locally computes $\pi(x + r)$ and secret-shares it, proving in the process that $[\pi(x+r)]_d$ was correctly formed. This can be done efficiently if we do many blocks in parallel. Then, given $[\pi(x + r)]_d = [\pi(x) + \pi(r)]_d$ and $[\pi(r)]_d$, players subtract shares locally to get $[\pi(x)]_d$.

5 Subprotocols

In the previous sections, we have covered how to evaluate a circuit C by transforming it into C' and computing layer by layer. We begin this section by listing known protocols that we will be using for this. Subsequently we cover new protocols we propose.

Known protocols. From [13] we borrow the following protocols:

- `Share`(D, d): A dealer D computes shares of a block of l secrets using a degree d polynomial and sends a share to each player. Communication is $\mathcal{O}(n)$ and computation is $\mathcal{O}(n \log n)$.
- `Reco`(R, d): Assumes a block has been shared using a polynomial of degree at most d. All players send their shares of the block to R, who uses standard error correction techniques to reconstruct the block. Communication is $\mathcal{O}(n)$ and computation is $\mathcal{O}(n \log n)$.
- `RobustShare`(d): This protocol basically implements verifiable secret-sharing for one or more dealers who want to secret-share $\Theta(n)$ blocks each using polynomials of degree d. The functionality it implements, $\mathcal{F}_{RobustShare}$, is shown in Figure 1 on the next page.
- `RanDouSha`(d): Generates a vector of random blocks and a degree d and a degree $2d$ sharing of each block. More precisely, it implements the functionality shown in Figure 2 on the following page.
- `RobustReshare`(d, d'): Takes as input a number of secret shared blocks. For each input $[x]_d$ it outputs a new sharing $[x]_{d'}$. However, it does not keep x secret.
- `SemiRobustShare`(d): Same as `RobustShare`(d), but the adversary can cause some of the honest dealers to fail. However, during the entire global protocol, he can only make up to t honest dealers fail.

For every protocol above except for the first two, the communication complexity is $\mathcal{O}(\beta n^2)$, and the computational complexity is $\mathcal{O}(\beta n^2 \log n)$, for handling β groups of $\Theta(n)$ blocks. In both cases we must additionally pay $\mathcal{O}(n^2)$ per complaint. Complaints are handled as in our protocol `RandomPairs` in Figure 4 on page 456. Since each complaint results in at least one corrupted player being eliminated from the protocol, at most t complaints can occur in total.

Furthermore, there is a minimal cost for these protocols, since they are built to handle groups of blocks and not just single blocks at a time. `RobustShare` for example always costs at least as much as for $\beta = n$. For a protocol like

1. Receive from all honest players the identities of the dealers and the number of blocks they want to share. Abort if the input is inconsistent. Receive also a set of input blocks to share from each honest dealer.
2. Send "Shares?" to the adversary together with the identities of the dealers and the number of blocks they want to share.
3. Receive from the adversary, for each block to be shared by an honest dealer, one share for each corrupted player (this should be thought of as the shares the adversary wants the corrupted players to receive). For each corrupt dealer, receive a polynomial of degree at most d.
4. For each block to be shared by an honest dealer, choose a random polynomial of degree at most d that is consistent with the block and the shares the adversary chose for the corrupted players. Compute and send the resulting shares to the honest players, and send the entire polynomial to the dealer.
5. For each block to be shared by a corrupt dealer, if the adversary sent a polynomial of correct degree, compute shares using this polynomial and send them to the players, otherwise tell all players that the dealer failed.

Fig. 1. The functionality $\mathcal{F}_{RobustShare}$

SemiRobustShare, it is possible to handle $\beta = 1$ efficiently, but then we need to add $\mathcal{O}(n^3)$ for n broadcasts. However, as we will show later, these cases make no difference in our final complexity; for this we do not care about how well our protocols handle a small number of elements, we care about how they scale.

In [13] there is a proof of perfect privacy and correctness for each of the protocols above, but it was not proved there that RanDouSha and RobustShare implement the corresponding functionalities. A proof of this follows quite easily from correctness and privacy in the same way as in the proof for the protocol RandomPairs, which we present in detail below.

We define functionalities only for some of the protocols above. The rest are mentioned because we use them as parts of other protocols. The final UC proof in the full version [12] only requires these parts to have perfect privacy and correctness.

5.1 Permuting Elements within a Block

The basic idea behind our protocols for permuting the set of elements within each block for a vector of blocks was already explained in Section 4. To use this idea, we need to be able to produce pairs of sharings $[r]_d, [\pi(r)]_d$ for random r's, and a server needs to be able to secret-share blocks while showing that they were correctly permuted. First we present the protocol RandomPairs for producing the required permuted pairs. The protocol for resharing and proving is simpler and yet very similar, and for that case we provide only a sketch. The protocol makes use of *hyperinvertible matrices.* A matrix is hyperinvertible if any intersection

1. Each honest player sends a natural number r to $\mathcal{F}_{double}$. If the honest players sent different values for r, $\mathcal{F}_{double}$ halts and outputs ABORT. Otherwise, send r and message "Shares?" to the adversary.
2. The adversary chooses $2r$ sets of shares for the corrupted players.
3. $\mathcal{F}_{double}$ chooses r random blocks $(x_1, \ldots, x_r)$ and creates random sharings $([x_1]_d, \ldots, [x_r]_d)$ and $([x_1]_{2d}, \ldots, [x_r]_{2d})$ such that they are consistent with the shares submitted by the adversary.
4. $\mathcal{F}_{double}$ outputs the resulting shares to the players.

Fig. 2. The functionality $\mathcal{F}_{double}$

between k rows and k columns of the matrix is invertible. In [2], it is described how such a matrix can be constructed. We refer to [2] for the details, but it is important to note, as was also done in [13], that we may use the $\mathcal{O}(n \log n)$ FFT algorithms to multiply our hyperinvertible matrices onto vectors.

Creating Permuted Pairs. The functionality $\mathcal{F}_{pairs}$ shown in Figure 3 details our requirements for the creation of permuted pairs. It works almost exactly like $\mathcal{F}_{double}$.

1. Each honest player sends a natural number r and a permutation π to $\mathcal{F}_{pairs}$. If the honest players sent different values for r or π, $\mathcal{F}_{pairs}$ halts and outputs ABORT. Otherwise, send r and message "Shares?" to the adversary
2. The adversary chooses $2r$ sets of shares for the corrupted players.
3. $\mathcal{F}_{pairs}$ chooses r random blocks $(x_1, \ldots, x_r)$ and chooses random sharings $([x_1], \ldots, [x_r])$ and $([\pi(x_1)], \ldots, [\pi(x_r)])$ such that they are consistent with the shares submitted by the adversary.
4. $\mathcal{F}_{pairs}$ outputs the chosen shares to the players.

Fig. 3. The functionality $\mathcal{F}_{pairs}$

An observation is needed before we present the protocol. Say we have some permutation π on l different elements, a vector of random blocks $(x_1, \ldots, x_n)$, and a vector of $y_i = \pi(x_i)$. Now suppose we apply some m by n matrix M and get the resulting vectors $(x'_1, \ldots, x'_m)$ and $(y'_1, \ldots, y'_m)$.

Applying M to a vector of blocks corresponds to applying M to l different vectors at once. Permuting all blocks and then applying M clearly has the same result as applying M and then permuting the resulting blocks. More precisely, after applying M, $\pi(x'_i) = y'_i$.

We now present the protocol `RandomPairs`. It is run in parallel for all of the players with the restriction that $n - 3t = \Omega(n)$. The matrix M is hyperinvertible of dimension n by $n - 2t$, and X is hyperinvertible of dimension $n - 2t$ by $n - 2t$. The protocol is shown in Figure 4 on the next page.

Proposition 1. *The protocol* `RandomPairs` *securely realizes* $\mathcal{F}_{pairs}$ *in the UC model with perfect security against an active and adaptive adversary corrupting at most* t *players, where* $n - 3t = \Omega(n)$*.* `RandomPairs` *creates* $\Theta(n^2)$ *permuted pairs at a time with a communication complexity of* $\mathcal{O}(n^3)$*, and a computational complexity of* $\mathcal{O}(n^3 \log n)$*. In both cases, we add* $\mathcal{O}(n^2)$ *per complaint.*

Proof. The proof is divided into three parts. The first two are correctness and simulation, and together they prove security in the UC model. The last part deals with the complexity.

Correctness: To show correctness, we must prove that all generated pairs are consistently shared and correctly permuted. Consider the set of players $\mathcal{P}$. If we denote by $\mathcal{P}'$ the subset of non-eliminated players, we know that by the end of the elimination step, only sharings coming from players in $\mathcal{P}'$ will be used.

We know that for any dealer $D \in \mathcal{P}'$, there are no conflicts $\{P_i, D\} \in \mathcal{C}$ for any $P_i \in \mathcal{P}'$. If there were such conflicts, they would have caused the elimination of either D or P_i in the elimination phase. This means that all honest players in

1. **Sharing**
   For each player D acting as dealer, and each group g of pairs to make, run the following in parallel:
   (a) D picks random blocks $(x_1, \dots, x_{n-2t})$ and $(y_1, \dots, y_{n-2t}) = (\pi(x_1), \dots, \pi(x_{n-2t}))$.
   (b) D shares the x_i and the y_i using protocol `Share`.
   (c) All players calculate
   $$([x'_1], \dots, [x'_n]) = M([x_1], \dots, [x_{n-2t}])$$
   $$([y'_1], \dots, [y'_n]) = M([y_1], \dots, [y_{n-2t}]).$$
   (d) For all i, all players P_j send their shares of $[x'_i]$ and $[y'_i]$ to P_i.
   (e) For all i, the dealer D sends all shares of $[x'_i]$ and $[y'_i]$ to P_i.
2. **Checking**
   Initialize $\mathcal{C} = \emptyset$. This set will contain sets of conflicting players. Now for each player P_i in parallel:
   (a) P_i checks that the sharings received for x'_i and y'_i by all D for all groups are consistent, and that $y'_i = \pi(x'_i)$. For any pair (P_j, D) where this check went well, P_i also checks that he received the same shares from all pairs of dealers D and P_j. If all goes well, he broadcasts a 1, and a 0 is broadcast if one or more checks fail.
   (b) If P_i broadcast a 0, he now proceeds to broadcast the number of complaints he intends to make. The complaints are then handled as described in the following. If at any point P_i broadcasts badly formatted complaints or the same complaint more than once, P_i is immediately eliminated and ignored.
   (c) If a dealer D dealt inconsistent shares or the pairs were not correctly permuted, P_i broadcasts (CONFLICT, P_i, D). All players include the set $\{P_i, D\}$ in $\mathcal{C}$.
   (d) Otherwise, if P_i sees that it has received different shares from some P_j and D for a group g, it broadcasts (CONFLICT, $D, P_j, g, share_D, share_{P_j}, w$), where w indicates whether it is a conflict with shares of $[x'_i]$ or $[y'_i]$. Such conflicts are sent out for any relevant cases, but at most one conflict is sent out for any specific pair (D, P_j).
      i. If D finds that $share_D$ does not match what he sent to P_i, he broadcasts (CONFLICT, D, P_i), and it is recorded in $\mathcal{C}$.
      ii. If P_j finds that $share_{P_j}$ does not match what he sent to P_i, he broadcasts (CONFLICT, P_j, P_i). This is recorded in $\mathcal{C}$.
      iii. If neither D nor P_j broadcasts a conflict, the conflicting set $\{D, P_j\}$ is included in $\mathcal{C}$.
3. **Elimination**
   All players now locally run the following elimination algorithm:
   (a) If there is a pair $\{P_i, P_j\} \in \mathcal{C}$ such that neither player has been eliminated so far, eliminate both players by removing them from the set $\mathcal{S}$ of player.
   (b) Keep all pairs $([x_i], [y_i])$ shared by non-eliminated players, throw away the rest.
4. **Postprocessing phase**
   (a) Reorder the players such that 1 through $n - 2t$ are non-eliminated.
   (b) (x_i^j, y_i^j) is the i'th pair of blocks known to the j'th player, for all non-eliminated j, and for each group.
   (c) Every player calculates
   $$([a_i^1], \dots, [a_i^{n-2t}]) = X^{-1}([x_i^1], \dots, [x_i^{n-2t}])$$
   $$([b_i^1], \dots, [b_i^{n-2t}]) = X^{-1}([y_i^1], \dots, [y_i^{n-2t}]).$$
   for all $i \in \{1, \dots, n - 3t\}$, and for each group.
   (d) For each group, the output is given by the pairs $([a_i^j], [b_i^j])$ for $i, j \in \{1, \dots, n - 3t\}$.

Fig. 4. Protocol `RandomPairs`

$\mathcal{P}'$ agree that the shares they have received from dealers $D \in \mathcal{P}'$ are consistent and represent correctly permuted pairs, and furthermore these shares agree with all shares received from $P_j \in \mathcal{P}'$.

Now consider all non-eliminated honest players. We know that at least for every two players eliminated, one of the players must have been corrupted. Therefore, we have at least $n - 2t$ honest players in $\mathcal{P}'$. Now select exactly $n - 2t$ of those and form the set H. It can be seen then that

$$([x'_i])_{P_i \in H} = M_H([x_i])_{1 \leq i \leq n-2t},$$

where M_H is a matrix containing only the rows of M with indices corresponding to the players in H. Since M_H is a square submatrix of a hyperinvertible matrix, it must be invertible. This means that

$$([x_i])_{1 \leq i \leq n-2t} = M_H^{-1}([x'_i])_{P_i \in H}.$$

The calculations above also hold for the y_i. We know that all pairs (x'_i, y'_i) where $P_i \in H$ are guaranteed to be consistently shared and correctly permuted. Applying the linear transformation M_H^{-1} preserves this property, and so we know that all of the original pairs (x_i, y_i) must be correct as long as the dealer is in $\mathcal{P}'$, but these are exactly the pairs we keep after the elimination phase.

Following the elimination phase, new pairs are created by applying yet another linear transformation. As before, linear transformations preserve the consistency of sharings and the property that pairs are correctly permuted, and thus correctness is ensured.

Simulation: To prove UC security, we must also show that we can construct a simulator $\mathcal{S}$ such that any environment $\mathcal{Z}$ cannot distinguish between the real world where it communicates with the adversary $\mathcal{A}$ and the ideal world where it communicates with $\mathcal{S}$. We do this by first proving perfect privacy (i.e. we prove that the adversary's view is independent of the secrets shared), and then we show how to use this and correctness to build a simulator.

For perfect privacy, all values seen by the adversary should be independent of the secret, which in this case is the set of output pairs. Throughout the protocol, $\mathcal{A}$ learns openings of sharings from honest players, and it knows its own sharings as well. It is these values that should be independent of the output. More specifically, we need only examine sharings by non-eliminated players, since the others are not used to create the output.

First, we prove that the sharings distributed by non-eliminated honest players are independent of the sharings opened towards $\mathcal{A}$. For any honest dealer and any group, let $I = \{1, \ldots, n-3t\}$ be the indices of the initial blocks and R those of the remaining blocks. Now choose a set C of size t that contains all indices of the corrupted players. The corrupted players now know openings of

$$([x'_i])_{i \in C} = M_C^I([x_i])_{i \in I} + M_C^R([x_i])_{i \in R},$$

where M_A^B means the matrix M restricted to rows in A and columns in B. A similar equation holds for the y'_i. Since $|C| = |R|$, there is exactly one choice of blocks in R that matches what the adversary can see for any set of blocks in I. In other words, the blocks opened to $\mathcal{A}$ are independent of the ones dealt by the honest dealers.

The final output blocks are created using the sharings from all non-eliminated servers, possibly including some corrupted servers. Therefore, we must also prove that the final outputs are independent of sharings from non-eliminated corrupt players. For the a_i^j and any group (the proof is the same for the b_i^j), let $I =$

$\{1, \ldots, n-3t\}$ be the set of the initial $n-3t$ indices, R the subsequent t, and C a set of size t containing the indices of all non-eliminated corrupted players (fill the rest of C with other players if there are less than t). The adversary knows x_i^j for all $j \in R$, so the sharings known to $\mathcal{A}$ are

$$([x_i^j])_{j\in C} = X_C^I([a_i^j])_{j\in I} + X_C^R([a_i^j])_{j\in R},$$

for all i. Since $|C| = |R|$, and since X is hyperinvertible, X_C^R is invertible. Therefore, for any set of blocks known to the adversary, there is exactly one choice of blocks $[a_i^j]_{j\in R}$ not output for any set of output blocks. In other words, the blocks dealt by $\mathcal{A}$ are independent of the output blocks. This concludes our proof of privacy.

We can now show how to construct a simulator $\mathcal{S}$. It simply runs dummy versions of the honest players and lets the execute the protocol with $\mathcal{A}$. We know that any values seen by $\mathcal{A}$ during the protocol are independent of the actual secrets shared, so the values generated by $\mathcal{S}$ towards $\mathcal{A}$ must be correctly distributed. When the protocol is done, the shares for corrupted players generated by the simulated run is fed into $\mathcal{F}_{pairs}$. The functionality now chooses the output sharing so to match these values, i.e. the honest players obtain shares that are consistent with a set of correctly distributed secrets and with the shares held by the adversary. By correctness of the protocol, this matches exactly the distribution of the output of a real protocol run.

The very last part of the proof is to deal with adaptive corruptions. First of all, if an honest player is corrupted during the protocol run but before we receive outputs from $\mathcal{F}_{pairs}$, we may simply open up one of the dummy parties to the adversary and continue from there. The only difficult part is if a server is corrupted after the output sharings have been chosen, because in that case the view of a dummy party does not match the output sharings. To adjust the view of a dummy party to the actual output shares of $\mathcal{F}_{pairs}$, we examine how these shares are constructed. We start by adjusting the shares of the $[a_i^j]$ for $j \in I$ (all of the following works in the same way for the b_i^j). The adversary knows the full sharings of

$$([x_i^j])_{j\in C} = X_C^I([a_i^j])_{j\in I} + X_C^R([a_i^j])_{j\in R},$$

so for those we simply pick the correct shares of $[a_i^j]$ for $j \in R$ to match the adjusted shares for $j \in I$. Now calculate $([x_i^j])_j = X([a_i^j])_j$ to find the remaining shares owned by the newly corrupted player. This of course means that the other dummy parties have to adjust their sharings from this point. The last problem is x_i^j created by this player. We can easily adjust its sharing of those values to match what we need, but it also needs to match the values opened to the adversary during the sharing of them. Luckily, we already know that this is simply a matter of adjusting the randomness used in the sharing.

Complexity: We now examine the complexity of the protocol. Going through each step of the protocol and remembering that every server is a dealer, we see that each step has a maximum communication complexity of $\mathcal{O}(n^3)$. Clearly this is also the total communication complexity. The computational complexity is $\mathcal{O}(n^3 \log n)$ plus the cost of each complaint, since in the slowest step, every server must check the consistency of $\Theta(n)$ sharings by interpolation, which can be done by using $\mathcal{O}(n \log n)$ FFT. Every complaint adds $\mathcal{O}(n^2)$ to both complexities for the broadcast.

Permuting Elements within Blocks. The next subprotocol `PermuteWithinBlocks`, and it is shown in Figure 5 takes as input the shares of blocks $([x_1], \ldots, [x_n])$, a vector of random pairs $(([s_1], [\pi(s_1)]), \ldots, ([s_n], [\pi(s_n)]))$, and the permutation π. It outputs shares of new sharings $([\pi(x_1)], \ldots, [\pi(x_n)])$. For this protocol, we prove correctness and privacy here, and use these properties in the simulation proof for the main protocol.

1. For every n input blocks, we do the following.
2. The servers locally compute $[x_i + s_i] = [x_i] + [s_i], \quad 1 \leq i \leq n$.
3. The servers select the non-eliminated server j that has least recently been chosen in this way and invoke `Reco` to reconstruct the $[x_i + s_i]$ to j.
4. Server j locally computes $\pi(x_i)$ for all i.
5. Server i uses protocol `Permuted` to share $[\pi(x_i + s_i)]$ for all i, proving in the process that it has been consistently shared and permuted. If `Permuted` outputs FAIL, return to step 3 (see description of `Permuted` in the text).
6. The players locally compute $[\pi(x_i)] = [\pi(x_i + s_i)] - [\pi(s_i)], \quad 1 \leq i \leq n$.

Fig. 5. Protocol `PermuteWithinBlocks`

Note that we only run the protocol for n blocks at a time to limit the cost of `Permuted` failing. For efficiency, we must work on at least n blocks at a time, so this is the natural choice. The protocol `Permuted` that was mentioned above is an adaptation of `RandomPairs`: there is only one dealer, server j. Rather than sharing both the x_i's and $\pi(x_i)$'s, the server shares only $\pi(x_i)$, since servers already have shares of the x_i's in question. However some extra random x_i's are added to ensure privacy (recall that `RandomPairs` requires extra random blocks that will not be output). Otherwise, we do exactly the same as in `RandomPairs` but if FAIL if server j is eliminated we stop immediately and output fail. The postprocessing phase is omitted, since there is only a single dealer who is allowed to know the (masked) secret.

It is perfectly private and correct by for the same reason that `PermutedPairs` is. As for the complexities, we consider permuting β groups of $\Theta(n)$ blocks (i.e. we permute $\Theta(\beta n)$ blocks). Ignoring broadcasts for a moment, we see that communication is at its most expensive when initially sharing, which costs $\mathcal{O}(\beta n^2)$. The most expensive computational step is still checking, which costs $\mathcal{O}(\beta n^2 \log n)$. For both computation and communication, we need to add $\mathcal{O}(n^3)$ in broadcast costs in both cases (regardless of the number of groups) and a further $\mathcal{O}(n^2)$ per complaint.

For the protocol `PermuteWithinBlocks`, it is clear that we still have privacy, since random blocks are added before opening. Correctness is trivial from the construction. As for the complexities, the most expensive step is `Permuted`. So both computational and communication complexities are as above, with the exception that the cost is multiplied by the number of times we fail and have to rerun Permuted. Since each failure results in at least one corrupt player being eliminated, the worst case is having to rerun t times.

5.2 Multiplications

As explained earlier, our circuit consists of only addition, multiplication and H-gates, where $\mathrm{H}(x, y, c) = (cx + (1 - c)y, cy + (1 - c)x)$. Since addition is trivially done by local computation, it is sufficient to explain how to handle multiplications. In order to do this, we need the protocol RobustReshare; as mentioned above it coverts a vector of blocks from being shared with degree d_1 to shares with degree d_2. In a nutshell, it publicly reconstructs the values and then reshares them. Assume that we are given shared blocks $[x]_d, [y]_d$ with degree d and sharings $[r]_d, [r]_{2d}$ of the same r but with degree d and $2d$. The protocol Multiply then works as shown in Figure 6.

1. For every pair of blocks x, y to multiply, we assume sharings $[r]_d, [r]_{2d}$ are available. The servers locally compute $[xy + r]_{2d} = [x]_d[y]_d + [r]_{2d}$.
2. RobustReshare is run to obtain $[xy + r]_d$ for all x, y.
3. For every x, y the servers locally compute $[xy]_d = [xy + r]_d - [r]_d$.

Fig. 6. Protocol Multiply

The pairs $[r]_d, [r]_{2d}$ we need can be generated using RanDouSha mentioned above. Correctness follows from correctness of RobustReshare. Privacy follows from privacy of RanDouSha since we can then assume the r is uniformly random from the adversary's point of view. The complexity is clearly dominated by RobustReshare whose complexity was covered earlier.

6 The Main Protocol

The final protocol is described in Figure 8 on the facing page, while the functionality realized is in Figure 7. This leads to:

1. The input clients send their inputs $(x_1, \ldots, x_r)$ to $\mathcal{F}_C$.
2. F_C distributes $(y_1, \ldots, y_t) = C(x_1, \ldots, x_r)$ to the intended output clients.

Fig. 7. The functionality $\mathcal{F}_C$ for the circuit C

Theorem 1. *There exists $0 < \delta < 1/3$ such that given n servers and an arithmetic circuit C that is at least $\Omega(n)$ gates wide, the protocol EvalCircuit realizes $\mathcal{F}_C$ with perfect security in the UC model against an active and adaptive adversary corrupting up to $t < \delta n$ servers.*

The total communication complexity is

$$\mathcal{O}(\log n \log |C| \cdot |C|) + \mathrm{poly}(n, \log |C|) \cdot \mathrm{depth}(C)^2,$$

Preprocessing: Transform C into C'.

Step 0**:** Input clients invoke the functionality $\mathcal{F}_{RobustShare}$ to share their inputs to the servers. The servers invoke $\mathcal{F}_{pairs}$ and $\mathcal{F}_{double}$ to create a set P_i of pairs and a set DS_i of double sharings for every layer $1 \leq i \leq d$ of C', where $d = \mathrm{depth}(C')$.

Step i**:** For $1 \leq i \leq d$, we have from the previous layers the set I_i of inputs for this layer as well as pairs and double sharings P_i and DS_i for this layer. Layer i is evaluated on I_i by the servers through local computations and a constant number of calls to `Multiply`.
The outputs of the layer may need to be permuted. If the blocks are to be permuted, they are permuted by local computation. If the elements within the blocks need to be permuted, the servers invoke `PermuteWithinBlocks` on the blocks in question.

Step $d+1$**:** The servers open sharings to the relevant output clients using `Reco`.

Fig. 8. Protocol `EvalCircuit`

while the total computational complexity is

$$\mathcal{O}(\log^2 n \log|C| \cdot |C|) + \mathrm{poly}(n, \log|C|) \cdot \mathrm{depth}(C)^2.$$

The actual threshold in Theorem 1 on the facing page is quite far from the optimal $n/3$ bound. To improve on this, we may use the *player virtualization* technique by Bracha [6] in the same way it was used in [13], to which we refer for the details of the construction. The basic idea is to construct virtual servers that run our protocol. To simulate each virtual server, a subset of the servers run a less efficient protocol, the inner protocol, that has a high threshold.

The difference from [13] is that here we are interested in perfect security. Therefore we need an inner protocol that also has perfect security. To this end, we can employ the BGW protocol [3]. Since it has threshold $n/3$, the construction from [13] gives us a threshold of $n/3 - \varepsilon$ for sufficiently large n, where $\varepsilon > 0$ may be chosen arbitrarily.

The construction increases both the computational and communication complexities to be the sum of the previous computational and communication complexities. Therefore, the new bound for both will be the old computational bound.

Because of space limitations, the proof of Theorem 1 on the preceding page is given in the full version (on ePrint) [12].

In the full version we also prove Corollary 1, which is a reduction in the complexity in some cases, namely when the depth is large and when X (the maximal number of connections from one layer to others) is large.

Corollary 1. *With the modification of the full version, the complexities of Theorem 1 can be altered to*

$$\mathcal{O}(\log \mathrm{depth}(C) \log n \log|C| \cdot |C|) + \mathrm{poly}(n, \log|C|) \cdot \mathrm{depth}(C) \log \mathrm{depth}(C)$$

for communication and

$$\mathcal{O}(\log \mathrm{depth}(C) \log^2 n \log|C| \cdot |C|) + \mathrm{poly}(n, \log|C|) \cdot \mathrm{depth}(C) \log \mathrm{depth}(C)$$

for computation.

7 Application to Two-Party Cryptography

In this section we sketch the application of our main result to reducing the computational overhead of zero-knowledge proofs and secure two-party computation.

In [20] it is shown how to obtain a zero-knowledge proof for the satisfiability of a circuit C from any MPC protocol for n servers in which one client ("the prover") has an input w and another client ("the verifier") should output $C'(w)$, where C' is a constant-depth circuit of roughly the same size as C which is easily determined by C. If the MPC protocol is adaptively secure against an active adversary who corrupts the prover and a constant fraction of the servers, the resulting zero-knowledge protocol will have soundness error of $2^{-\Omega(n)}$ plus the correctness error of the MPC protocol. The simulation error corresponds to that of the MPC protocol. The efficiency of the zero-knowledge protocol is essentially the same as that of the MPC protocol, excluding the cost of n commitments to strings whose total size is roughly the communication complexity of the MPC protocol.

The above transformation was combined with the MPC techniques from [11,9] to yield zero-knowledge proofs with a *constant* communication overhead. However, to guarantee soundness error of 2^{-k}, the *computational overhead* of this protocol must be $\Omega(k)$, even if ideal commitments are used. Plugging in our main result, we obtain a perfect zero-knowledge protocol in the *commitment-hybrid model* (i.e., using ideal commitments) in which both the communication and computation overhead are polylogarithmic in k. As a side benefit, the perfect security of our protocol allows for a simpler and more round-efficient transformation into a zero-knowledge proof protocol (see [20], Section 4).

To implement the commitment-hybrid model, we can use the constant overhead constructions from [21] or the polylog-overhead constructions from [1]. The latter have the advantage of relying on fairly standard cryptographic assumptions, related to the intractability of decoding random linear codes or learning with errors.

We note that in the case of zero-knowledge *arguments* (with computational soundness), it is possible to combine the PCP-based approach of [23,25] for efficient arguments with state of the art PCP constructions [4] and efficient lattice-based constructions of collision-resistant hash functions [26,24] to get alternative constructions with polylogarithmic computational overhead. However, other than offering only computational soundness, the resulting protocol requires stronger assumptions, inherits the complex and seemingly impractical nature of current PCP constructions, and does not allow to eliminate the need for cryptography using preprocessing.

We finally note that similar results can be obtained in the more general context of secure two-party computation. One approach to obtain these results is to apply the GMW-compiler [17], with the efficient zero-knowledge proofs described above, to a constant-overhead protocol for the semi-honest model from [21]. The latter protocol relies on the existence of a pseudorandom generator stretching n bits to n^2 bits in which each bit of the output depends on just a constant number of input bits — a plausible but nonstandard assumption. Another approach, which can offers *unconditional* security in the OT-hybrid model, is to instantiate the protocol compiler from [22] with our main protocol as the "outer protocol".

8 On the Relevance of Gentry's Scheme

The recent breakthrough of Gentry [16], suggesting the first plausible candidate for a fully homomorphic encryption scheme, has a great impact on the theoretical efficiency of MPC. By distributing the key generation and decryption of Gentry's scheme between the n players, it is possible to obtain general constant-round MPC protocols whose communication complexity only depends on n and the length of the inputs and outputs of C rather than the size of C. We note, however, that this protocol can only provide *computational* security (under a non-standard assumption) and, perhaps more importantly, its computational overhead involves a large polynomial in the security parameter. The high computational cost seems to make Gentry's scheme, in its current form, too inefficient for practical purposes. Finally, for circuits whose output length is not much smaller than their size (as in the case of performing a large number of simple computations), even the communication overhead of this protocol becomes a large polynomial in k and n. In contrast, our protocol has the same overhead even in this case. In light of the above, it seems fair to conclude that Gentry's result has limited relevance to the results of the present work from both a theoretical and from a practical point of view.

References

1. 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)
2. Beerliová-Trubíniová, Z., Hirt, M.: Perfectly-secure MPC with linear communication complexity. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 213–230. Springer, Heidelberg (2008)
3. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In: STOC, pp. 1–10. ACM, New York (1988)
4. Ben-Sasson, E., Goldreich, O., Harsha, P., Sudan, M., Vadhan, S.P.: Short pcps verifiable in polylogarithmic time. In: IEEE Conference on Computational Complexity, pp. 120–134. IEEE Computer Society, Los Alamitos (2005)
5. Benes, V.E.: Optimal rearrangable multistage connecting networks. The Bell System Technical Journal 43, 1641–1656 (1964)
6. Bracha, G.: An O(log n) expected rounds randomized byzantine generals protocol. J. ACM 34(4), 910–920 (1987)
7. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: FOCS 2001: Proceedings of the 42nd IEEE symposium on Foundations of Computer Science, Washington, DC, USA, pp. 136–145. IEEE Computer Society, Los Alamitos (2001)
8. Chaum, D., Crépeau, C., Damgard, I.: Multiparty unconditionally secure protocols. In: STOC 1988: Proceedings of the twentieth annual ACM symposium on Theory of computing, pp. 11–19. ACM, New York (1988)

9. Chen, H., Cramer, R.: Algebraic geometric secret sharing schemes and secure multiparty computations over small fields. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 521–536. Springer, Heidelberg (2006)
10. Cramer, R., Damgård, I., Nielsen, J.B.: Multiparty computation from threshold homomorphic encryption. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 280–299. Springer, Heidelberg (2001)
11. Damgård, I., Ishai, Y.: Scalable secure multiparty computation. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 501–520. Springer, Heidelberg (2006)
12. Damgård, I., Ishai, Y., Krøigaard, M.: Perfectly secure multiparty computation and the computational overhead of cryptography. Cryptology ePrint archive, report 2010/131 (2010), http://eprint.iacr.org/
13. Damgård, I., Ishai, Y., Krøigaard, M., Nielsen, J.B., Smith, A.: Scalable multiparty computation with nearly optimal work and resilience. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 241–261. Springer, Heidelberg (2008)
14. Franklin, M.K., Yung, M.: Communication complexity of secure computation (extended abstract). In: STOC, pp. 699–710. ACM, New York (1992)
15. Gennaro, R., Rabin, M.O., Rabin, T.: Simplified VSS and fast-track multiparty computations with applications to threshold cryptography. In: PODC 1998: Proceedings of the seventeenth annual ACM symposium on Principles of distributed computing, pp. 101–111. ACM, New York (1998)
16. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC 1909: Proceedings of the 41st annual ACM symposium on Theory of computing, pp. 169–178. ACM, New York (2009)
17. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: STOC, pp. 218–229. ACM, New York (1987)
18. Hirt, M., Maurer, U.M.: Player simulation and general adversary structures in perfect multiparty computation. J. Cryptology 13(1), 31–60 (2000)
19. Hirt, M., Maurer, U.M., Przydatek, B.: Efficient secure multi-party computation. In: ASIACRYPT 2000: Proceedings of the 6th International Conference on the Theory and Application of Cryptology and Information Security, London, UK, pp. 143–161. Springer, Heidelberg (2000)
20. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-knowledge from secure multiparty computation. In: STOC 2007: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing, pp. 21–30. ACM, New York (2007)
21. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Cryptography with constant computational overhead. In: STOC 2008: Proceedings of the 40th annual ACM symposium on Theory of computing, pp. 433–442. ACM, New York (2008)
22. Ishai, Y., Prabhakaran, M., Sahai, A.: Founding cryptography on oblivious transfer — efficiently. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 572–591. Springer, Heidelberg (2008)
23. Kilian, J.: A note on efficient zero-knowledge proofs and arguments (extended abstract). In: STOC 1992: Proceedings of the twenty-fourth annual ACM symposium on Theory of computing, pp. 723–732. ACM, New York (1992)
24. 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)
25. Micali, S.: Computationally sound proofs. SIAM J. Comput. 30(4), 1253–1298 (2000)

26. 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)
27. Rabin, T., Ben-Or, M.: Verifiable secret sharing and multiparty protocols with honest majority. In: STOC 1989: Proceedings of the twenty-first annual ACM symposium on Theory of computing, pp. 73–85. ACM, New York (1989)
28. Shamir, A.: How to share a secret. Commun. ACM 22(11), 612–613 (1979)
29. Waksman, A.: A permutation network. J. ACM 15(1), 159–163 (1968)

Adaptively Secure Broadcast

Martin Hirt and Vassilis Zikas

Department of Computer Science, ETH Zurich
{hirt,vzikas}@inf.ethz.ch

Abstract. A broadcast protocol allows a sender to distribute a message through a point-to-point network to a set of parties, such that (i) all parties receive the same message, even if the sender is corrupted, and (ii) this is the sender's message, if he is honest. Broadcast protocols satisfying these properties are known to exist if and only if $t < n/3$, where n denotes the total number of parties, and t denotes the maximal number of corruptions. When a setup allowing signatures is available to the parties, then such protocols exist even for $t < n$.

Since its invention in [LSP82], broadcast has been used as a primitive in numerous multi-party protocols making it one of the fundamental primitives in the distributed-protocols literature. The security of these protocols is analyzed in a model where a broadcast primitive which behaves in an ideal way is assumed. Clearly, a definition of broadcast should allow for secure composition, namely, it should be secure to replace an assumed broadcast primitive by a protocol satisfying this definition. Following recent cryptographic reasoning, to allow secure composition the ideal behavior of broadcast can be described as an ideal functionality, and a simulation-based definition can be used.

In this work, we show that the property-based definition of broadcast does not imply the simulation-based definition for the natural broadcast functionality. In fact, most broadcast protocols in the literature do not securely realize this functionality, which raises a composability issue for these broadcast protocols. In particular, we do not know of any broadcast protocol which could be securely invoked in a multi-party computation protocol in the secure-channels model. The problem is that existing protocols for broadcast do not preserve the secrecy of the message while being broadcasted, and in particular allow the adversary to corrupt the sender (and change the message), depending on the message being broadcasted. For example, when every party should broadcast a random bit, the adversary could corrupt those parties who intend to broadcast 0, and make them broadcast 1.

More concretely, we show that simulatable broadcast in a model with secure channels is possible if and only if $t < n/3$, respectively $t \leq n/2$ when a signature setup is available. The positive results are proven by constructing secure broadcast protocols.

1 Introduction

Broadcast is one of the most fundamental primitives in distributed cryptography. It is used in almost any task that involves multiple players, like, e.g., voting, bidding, secure function evaluation, threshold key generation, multi-party computation, etc — just to mention a few. The security of these protocols inherently relies on the security of the

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 466–485, 2010.

underlying broadcast protocol. Informally, broadcast allows a sender to distribute his input among a set of players, such that every player gets the same value, even if the sender is dishonest.

1.1 Summary of Known Results

Broadcast was introduced by Pease, Shostak, and Lamport [LSP82] who showed that an adversary who can corrupt up to t players can be tolerated for perfectly secure Broadcast if and only if $3t < n$. This model has been extensively studied [DFF$^+$82, TPS87, FM88, CW89, BGP89, BDDS92, GM93] and protocols with optimal resiliency and complexity (communication and computation) polynomial in the number of players were suggested.[1] Other solutions [DS82, PW92] considered a setting where a setup allowing digital signatures is available, and showed that Broadcast tolerating an arbitrary number of cheaters ($t < n$) is possible. The suggested protocols are polynomial in the number of players and are as secure as the underlying signature scheme.[2]

Recently, Lindell, Lysyanskaya, and Rabin [LLR02] proved that, unless unique session identifiers are available, the bound $t < n/3$ is necessary for feasibility of concurrently composable Broadcast, even when a setup allowing digital signatures is given. To the positive side, they showed that when unique session IDs are available, then the protocols which achieve Broadcast and use signatures for authentication, e.g., [DS82, PW92] can be be transformed to concurrently composable Broadcast protocols.

1.2 Property-Based vs. Simulation-Based Definition

Intuitively, one could think of broadcast as a megaphone given to the sender, which every player can hear. More formally, this megaphone can be modeled as a functionality (in the sense of [Can00, Can01]), which receives an arbitrary message from the sender, and forwards this message to all players. The goal of a broadcast protocol is to realize this functionality, in the sense that in any context the abstract broadcast functionality can safely be replaced by the broadcast protocol. However, in the big body of broadcast literature, protocols are not proven to securely realize the above functionality; rather, they are shown to satisfy the following properties:

- Consistency: There exists some y such that every player outputs y.
- Validity: If the sender is honest and has input x then $y = x$.
- Termination: For every honest player the protocol terminates after a finite number of rounds.

Of course, the hope is that these properties imply security of a broadcast protocol in a simulation-based sense (i.e., any protocol satisfying these properties is expected

[1] Many of these protocols are actually Consensus protocols, from which a Broadcast protocol can be build by having the sender send his input to everybody and then invoke Consensus on the received values.

[2] In fact, feasibility of Broadcast for $t < n$ when a setup is available was also proved in [LSP82] but the suggested protocol has exponential communication complexity.

to securely realize the above broadcast functionality). However, this is not the case, as the property-based definition has a major flaw: the validity condition does not take into account the point in time when the sender gets corrupted. In particular, the definition does not rule out that the adversary can corrupt the sender *depending on the message which the sender intends to broadcast.* In fact, a broadcast protocol can satisfy the above three properties, and still allow the adversary to *first* learn the sender's message, and *then* to decide whether or not to corrupt the sender and make him broadcast a different message. This clearly contradicts the simulation-based definition, as well as the intuition with the megaphone. We stress that it is perfectly legal that the adversary can change the broadcasted message by corrupting the sender, and also it is perfectly legal that she learns the broadcasted message; however, it is counter-intuitive that she can first learn the message, without corrupting the sender, and then still be able to corrupt the sender and change it.

We give two examples to demonstrate the relevance of this problem: First, consider the following process for 10 players: Each player p_i ($i = 1, \dots, 10$) in turn chooses a bit $b_i \in_R \{0, 1\}$ uniformly at random and announces it using ideal broadcast (e.g., using a megaphone), i.e., first p_1 selects $b_1 \in_R \{0, 1\}$ and announces it, subsequently p_2 selects $b_2 \in_R \{0, 1\}$ and announces it, etc. Consider an adversary who can corrupt at most three of the players, and her goal is to have only 1's broadcasted. Clearly, the probability that the output sequence consists only of 1's is at most 2^{-7}, as each of the seven bits chosen by the honest players are 1 with probability $1/2$. However, when we replace the ideal broadcast with some broadcast protocol satisfying the above three properties, then the adversary might be able to bring this probability to $46 \cdot 2^{-9}$, which is more than ten times bigger. She can achieve this by only corrupting those players p_i who intend to broadcast 0. With the mentioned probability, there are at most three such players, and the adversary can corrupt each of them and make them broadcast 1.

A more cryptography-related example is the following: Consider a prover p who uses the Fiat-Shamir (interactive) protocol to publicly prove to n players (verifiers) that he knows the square root of some publicly known y (in an RSA group). In order to do that, p executes one round of the Fiat-Shamir protocol with each verifier p_i in sequence. All executions are public, in the sense that all the messages are exchanged using ideal broadcast. Each verifier accepts if all rounds are accepting. Assume that the adversary can corrupt up to $t = n/2$ of the verifiers. Then in this protocol the probability that a malicious prover can make the players accept when he does not know the square root is negligible in n. However, along the lines of the above example, when the ideal broadcast is replaced by a broadcast protocol satisfying the above properties, a malicious prover might be able to corrupt only those verifiers who intend to challenge the bit the prover is not prepared to, which allows a malicious prover to cheat with probability $1/2$.

1.3 Broadcast in the Literature

As mentioned in the previous section, the big body of broadcast protocols in the literature are proven secure with respect to the mentioned properties, rather than with respect to a broadcast functionality. This would be only a minor issue if these protocols would securely realize the broadcast functionality. However, in the following we show that (at least most of them) fail to do so.

Most broadcast protocols in the literature [LSP82, DS82, BPW91, PW92, BHR07] proceed as follows:[3] First the sender sends the message to the players, possibly along with a signature; then, the players try to establish a consistent view on the sender's input. Obviously, any protocol following this approach cannot be secure against an adaptive adversary: Unless some kind of simultaneous multi-send assumption on the communication channels is made (see below), some corrupted player can happen to be the first to receive the message from the sender, and depending on this message, the adversary can decide whether or not to corrupt the sender (and change the message to be broadcast). Clearly, this behavior is not allowed when the above mentioned broadcast functionality is used, because as soon as some corrupted player receives (from the functionality) the broadcasted value, it is guaranteed that the honest players will also receive it (the functionality also sends it to them). Note that we do not need to assume a fully rushing adversary for the above behavior; we simply do not exclude that some corrupted player might get the message first, before it is sent to other players.

Note that many broadcast protocols can apparently be turned secure when the network offers a *simultaneous multi-send* operation. Such an operation is *atomic* and allows the sender to distribute an n-ary vector such that every player p_i receives the i-th component of the vector. More precisely, the operation is atomic in the sense that as soon as some player obtains some information about his component, then all other player must be guaranteed to receive their respective component as well. Such a network-operation is of course quite a strong assumption. Indeed, assuming such an operation implies that a player who honestly behaves at a specific point in time can broadcast a message (by multi-sending it) which seems to be closer to a broadcast channel than to a point-to-point communication network. In fact, the Universal Composition framework [Can01] which is the most widely accepted framework for arguing about the security of protocols, explicitly excludes such a simultaneous multi-send assumption. Furthermore, for broadcast protocols using signatures and tolerating $t \geq n/3$ [DS82, PW92], even this assumption does not help, as still the adversary can learn the message in the first phase of the protocol, and make the broadcast fail afterwards by corrupting the sender and introducing signatures for different messages.[4]

The major problem is that in the broadcast literature, protocols are proven secure with respect to properties, but in the cryptographic protocols literature (VSS, MPC, etc), protocols are proven secure in a hybrid world with access to an ideal broadcast functionality (e.g., "secure-channels model with broadcast"). The security of these cryptographic protocols, when the broadcast functionality is instantiated with some broadcast protocol from the literature, is doubtful.

1.4 Contributions

We show that the property-based definition of broadcast does not imply simulation-based security with the natural functionality, not even in a stand-alone setting, not even

[3] This also includes any broadcast protocol which first has the sender send his input to everybody and then invokes a Consensus protocol, e.g. [DFF$^+$82, TPS87, FM88, CW89, BGP89, BDDS92, GM93], on the received values.

[4] This would essentially correspond to a Broadcast functionality with partial fairness and unanimous abort [GL02].

in the secure-channels model with perfect security. We also describe a weaker functionality which is realized by the known broadcast protocols, and, under certain conditions, can instantiate a broadcast primitive within a high level protocol. These conditions are, for example, satisfied by the VSS protocol from [BGW88]. Note however, that in many of the known protocols which assume broadcast, e.g., [CDD$^+$99], these conditions are not guaranteed. Hence, if one would be willing to make a compromise and accept the weaker functionality as the ideal functionality for broadcast, then he would need to (re-)prove the security of such protocols with this functionality in mind.

Furthermore, we give broadcast protocols with simulation-based security in the secure-channels model that tolerate $t < n/3$ (with perfect security, without further assumptions), respectively $t \leq n/2$ (with statistical resp. cryptographic security, when a secure signature functionality is available). Both bounds are tight. We stress that in the secure channels model, no protocol exists that securely realizes the natural broadcast functionality when $t > n/2$ (although property-based security is possible for $t < n$ [DS82, PW92]).

The negative result can easily be illustrated in the following broadcast protocol: First, the sender transmits the message to all players. Then, the players run a perfectly secure consensus protocol on the received values [BGP89]. The resulting broadcast protocol satisfies the consistency and validity property with perfect security when $t < n/3$. However, it is **not** a secure realization of the above natural functionality for broadcast. The main problem is that an adaptive adversary might first learn the message to be broadcasted, and then, depending on the learned message, still can corrupt the sender and make him broadcast a different message. The authors are not aware of any broadcast protocol in the literature that does not suffer from this problem (but see related work below).

The positive result for perfect security with $t < n/3$ is rather straight-forward: First, the sender secret-shares the message among the players. Then, the sharing is reconstructed. The only issue is how to do a secret-sharing without having a composable broadcast primitive. The second positive result, namely statistical and computational security for $t \leq n/2$, it more involved, a verifiable secret-sharing exists only for $t < n/2$ (but not for $t = n/2$).

The tightness of the bound for perfect security ($t < n/3$) follows directly from the impossibility of property-based broadcast. The tightness of $t \leq n/2$ is proven indirectly: we show that in any "broadcast protocol" for $2t = n + 1$, there exists a round in which the adversary (not corrupting the sender) obtains noticeable (i.e., not negligible) information about the message, but still can corrupt the sender and change the message.

1.5 Comparison with Previous Work

The idea to use VSS to get more out of a broadcast protocol was used in the context of simultaneous broadcast [CGMA85, CR87, Gen95, Gen00, HM05]. However, the goal of these works is to satisfy an additional property, namely to allow different parties to broadcast values in parallel while guaranteeing mutual independence of the broadcast values. This does not imply simulation-based composable security. Recently, Hevia [Hev06] proposed a simultaneous broadcast protocol which he proved to be universally composable. However, as all previous protocols in this line of research, also this protocol uses "normal" broadcast as sub-protocol, and the security analysis relies on the hope that this securely composes, which, as we show here, in general is not the case. In fact, the protocol in [Hev06] employs the verifiable secret-sharing scheme from [CDD+99], which in turn employs some broadcast primitive which (hopefully) securely composes in the secure-channels model for $t < n/2$. To our knowledge, our work is the first to present such a composable broadcast protocol.

In [LLR02] a broadcast protocol for $t < n$ was described, which is concurrently composable when unique session IDs are available. This result does not contradict ours, as it implicitly assumes that the players can simultaneously multi-send messages (c.f. [LLR02, Sect. 2.1]). In fact, this protocol is a "transformation from almost any Broadcast protocol to a protocol that concurrently composes". Because all known broadcast protocols have the above mentioned problem, also this construction has it when run in a model without simultaneous multi-send.

2 The Model

We consider the well-known secure channels model introduced in [BGW88, CCD88], where the players in $\mathcal{P} = \{p_1, \ldots, p_n\}$ are connected by a complete network of bilateral secure channels. In such a network the only way that the adversary can get information on a sent message is by corrupting the sender or the receiver.

2.1 Synchronous Communication (No Multi-send)

The communication is synchronous, i.e., all players have synchronized clocks and there is a known upper bound on the delivery time of any sent message. In such a synchronous model, the protocols proceed in rounds, where in each round every player can send a message to every other player.

There are several variations of the synchronous channels model suggested in the literature. In some works [Nie03, LLR02] it is implicitly assume that honest players can simultaneously multi-send messages, i.e, simultaneously send messages to several recipients. Such a multi-send operation is atomic and guarantees that for a sender who is honest upon sending, if one of the messages is delivered to its recipient then all the messages will be delivered (unchanged) to the corresponding recipients. As we already pointed out, such a simultaneous multi-send operation is a quite strong assumption on the communication network.

In this work we only assume bilateral communication and, in particular, *we do not assume simultaneous multi-send*: a player who is instructed to send a message to more than one players can do so one player at a time. This is consistent with the formulation of [Can01] where it is required that the processes are activated in turns, where at any point only a single process can be active, and it can send a message to one other process which becomes now the active process, and so on.

2.2 The Adversary

We consider a threshold adversary who can actively corrupt up to t players (we refer to this adversary as *t-adversary*). When some player p_i is corrupted then the adversary has full control on p_i. A player who is not corrupted is called *uncorrupted* or *honest*. Analogously, the corrupted players are also called *dishonest*.

There are several adversarial models in the literature which restrict the power of the adversary. For example a *static* adversary is one who chooses the players to corrupt at the beginning of the protocol.

In this work we do not put any such restrictions on the adversary's corruption power. In particular, the assumed adversary is *adaptive*, i.e., in contrast to a static adversary, she can corrupt additional players in the flow of the protocol depending on messages seen so far, with the only restriction that the total number of players she corrupts has to be at most t. Because no simultaneous multi-send is assumed, it might happen that some corrupted player receives his message from an honest player p in some round, before p has finished sending all his messages for this round.[5] If this happens, the adversary can corrupt p after learning the message which was sent to the corrupted player, and force him change the remaining messages which he intended to send in that round.

2.3 Security Definition

Following the [Can00, Can01] methodology security of protocols is argued via the ideal-world/real-world paradigm. In the real-world the players execute the protocol. The ideal-world is a specification of the task which we want the protocol to implement. More concretely, in the ideal-world the players can invoke a fully trusted party, called the *functionality*, denoted as $\mathcal{F}$, in the following way: the player sends their input(s) to $\mathcal{F}$; $\mathcal{F}$ runs its program on the received inputs (while running the program, $\mathcal{F}$ might receive additional inputs from the players or the adversary or send values to the adversary), and returns to the players their specified outputs. The specification of $\mathcal{F}$ is such that this ideal-evaluation captures, as good as possible, the goals of the designed protocol.

Intuitively, a protocol securely realizes a functionality $\mathcal{F}$, when the adversary cannot achieve more in the protocol than what she could achieve in an ideal-evaluation of $\mathcal{F}$. To formalize this statement, we assume an environment $\mathcal{Z}$ which decides the inputs of all players, and also sees their outputs. $\mathcal{Z}$ also sees the full view of the adversary $\mathcal{A}$ who is attacking the protocol. We denote the view of $\mathcal{Z}$ for an invocation of protocol π with adversary $\mathcal{A}$ as $\text{EXEC}_{\pi,\mathcal{A},\mathcal{Z}}$. A protocol π *t-securely realizes functionality* $\mathcal{F}$ when for any t-adversary $\mathcal{A}$ attacking protocol π, there exists an ideal-world adversary $\mathcal{S}$ (also called the *simulator*) such that no environment $\mathcal{Z}$ cannot tell whether it is interacting with $\mathcal{A}$ and the players running π or with $\mathcal{S}$ and the players running the ideal-world protocol (we denote the view of $\mathcal{Z}$ in an ideal-evaluation of $\mathcal{F}$ as $\text{EXEC}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$).

The three typical security notions are: *perfect security* ($\mathcal{A}$ is computationally unbounded, and the random variables $\text{EXEC}_{\pi,\mathcal{A},\mathcal{Z}}$ and $\text{EXEC}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$ are identically distributed), *statistical security* ($\mathcal{A}$ is computationally unbounded, and $\text{EXEC}_{\pi,\mathcal{A},\mathcal{Z}}$ and $\text{EXEC}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$ are statistically close), and *computational security* ($\mathcal{A}$ is efficient, and $\text{EXEC}_{\pi,\mathcal{A},\mathcal{Z}}$ and $\text{EXEC}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$ are computationally indistinguishable).

[5] Note that if one would assume a rushing adversary then this would be the case "by definition".

The $\mathcal{F}$-hybrid model. The power of the simulation-based definition is that it allows to argue about security of protocols in a composable way. In particular, let π_1 be a protocol which securely realizes a functionality $\mathcal{F}_1$. If we can prove that π_2 securely realizes a functionality $\mathcal{F}_2$ using *ideal-calls* to $\mathcal{F}_1$, then it follows automatically that the protocol which results by replacing, in π_2, the calls to $\mathcal{F}_1$ by invocations of π_1 also securely realizes $\mathcal{F}_2$. Therefore we only need to prove the security of π_2 in the so-called $\mathcal{F}_1$-*hybrid* model, where the players run π_2 and are allowed to make ideal-calls to $\mathcal{F}_1$. For more details on composability of protocols and a formal handling of both sequential and parallel composition (and also of universal composability), the reader is referred to [Can00, Can01].

3 Perfect Security (No Setup)

In this section we consider the case of perfect security, i.e., information theoretic (i.t.) with no error probability. We show that perfectly secure broadcast tolerating a t-adversary is possible if and only if $t < n/3$. Although this bound already appears in the literature, to the best of our knowledge, none of the suggested synchronous broadcast protocols for perfect security satisfies the simulation-based definition when secure-channels and an adaptive adversary are considered. Also, in addition to handling the perfect security case, this section serves as a good way to introduce some of our ideas.

The ideal functionality for broadcast $\mathcal{F}_{\text{BC}}$ when synchronous secure channels are assumed is quite intuitive; nevertheless, to keep our analysis complete, in the following we give a description. For simplicity we describe the functionality in terms of an ideal-world protocol. A UC-type version of this functionality can be found in the full version of this paper.

Functionality $\mathcal{F}_{\text{BC}}$

1. p_s sends his input x_s to the functionality $\mathcal{F}_{\text{BC}}$.
2. $\mathcal{F}_{\text{BC}}$ sends x_s to every $p \in \mathcal{P}$.

To show that the above functionality is not realized by known protocols, we observe that known broadcast protocols have the following pattern: At the beginning of the protocol the sender p_s sends his input x_s to the players in $\mathcal{P} \setminus \{p_s\}$; in a second phase the players try to establish a consistent view on the sender's input. Clearly all protocols which start by the sender sending his input to everybody and then invoke a consensus protocol on the received value, e.g., [CW89, BGP89, BDDS92], are of the above type. However, even the protocols where the second phase is not a self-contained consensus protocol, e.g., the broadcast protocols from [DS82, PW92], also follow the above paradigm.

The fact that any protocol following the above paradigm is insecure against an adaptive adversary can be seen as follows: In any such protocol, there is a good probability that a corrupted player is the first to receive the input x_s from p_s and the adversary can, depending on the received value, decide whether or not to corrupt the sender p_s (and possibly change the broadcasted value).[6] However, this behavior cannot be simulated, as by the time the simulator learns x_s from the functionality it is already too late to change it (the functionality also sends it to all honest players).

A direct way to deal with the above problem is to make sure that before any player (or the adversary) learns any information on x_s, the value x_s is secret-shared in a robustly reconstructible way. More concretely, when a secure Verifiable Secret Sharing (VSS) scheme is given, then one can easily construct a secure broadcast protocol (i.e., a protocol realizing $\mathcal{F}_{\text{BC}}$) by having p_s share his input x_s, and, subsequently, having the players publicly reconstruct the sharing.

It might look that we are done, as one could use the perfectly secure VSS from [BGW88] to achieve broadcast. But this is not quite true. The reason is that [BGW88] (and all other known VSS schemes with perfect security) use broadcast as a primitive. If we instantiate this primitive by one of the known broadcast protocols then we can no longer argue about the security of the full construction using composition. Nevertheless, we show in the following that replacing all broadcast invocations in the [BGW88] VSS scheme by executions of the [BGP89] broadcast protocol[7] does not cause any loss of security; we denote this VSS scheme by $\text{VSS}_{\text{BGW}}^{(\text{BGP})}$.

The security of $\text{VSS}_{\text{BGW}}^{(\text{BGP})}$ is argued in two steps. In a first step, we show that although the [BGP89] broadcast protocol, denoted in the following as BC_{BGP}, does not securely realize $\mathcal{F}_{\text{BC}}$, it does realize a weaker functionality, denoted as $\mathcal{F}_{\text{UBC}}$ (we refer to this functionality as *unfair broadcast*). In a second step, we show that under certain conditions (which are satisfied by the [BGW88] VSS protocol) , we can replace $\mathcal{F}_{\text{BC}}$ by $\mathcal{F}_{\text{UBC}}$ without loosing security.

The functionality $\mathcal{F}_{\text{UBC}}$ is described in the following. Intuitively, the difference to the functionality $\mathcal{F}_{\text{BC}}$ is that $\mathcal{F}_{\text{UBC}}$ allows the adversary to first receive the sender's p_s input (even without corrupting p_s) and then, depending on the received value, decide whether or not she wants to corrupt p_s and possibly modify the broadcasted value.

Functionality $\mathcal{F}_{\text{UBC}}$

1. p_s sends his input x_s to the functionality $\mathcal{F}_{\text{UBC}}$.
2. $\mathcal{F}_{\text{UBC}}$ sends x_s to the adversary.
3. If p_s is corrupted then the adversary sends a value to $\mathcal{F}_{\text{UBC}}$; $\mathcal{F}_{\text{UBC}}$ denotes the received value by x'_s (if p_s is not corrupted then $\mathcal{F}_{\text{UBC}}$ sets $x'_s := x_s$).
4. $\mathcal{F}_{\text{UBC}}$ sends x'_s to every $p \in \mathcal{P}$

Lemma 1. *Protocol BC_{BGP} perfectly t-securely realizes the functionality $\mathcal{F}_{UBC}$ for $t < n/3$.*

Proof. (sketch) As shown in [BGP89], the protocol BC_{BGP} satisfies the property-based definition of broadcast (i.e., it satisfies validity, consistency, and termination). We show

[6] In fact, if one assumes a rushing adversary then she can, by definition, always perform such an attack, as she first learns the messages sent to corrupted recipients.

[7] In fact [BGP89] describes a consensus protocol. A protocol for broadcast can be constructed by having the sender send his value to everybody and then invoke consensus on the received values.

that it perfectly securely realizes $\mathcal{F}_{\text{UBC}}$. Let $\mathcal{A}$ be an adversary attacking BC_{BGP}; a corresponding simulator $\mathcal{S}$ can be built as follows: First S waits for the input x_s from $\mathcal{F}_{\text{UBC}}$. Note that, because BC_{BGP} is fully deterministic and the players in $\mathcal{P} \setminus \{p_s\}$ have no input, knowing x_s allows $\mathcal{S}$ to perfectly simulate all the messages sent by honest players.[8] $\mathcal{S}$ invokes $\mathcal{A}$ and does the following:

1. $\mathcal{S}$ simulates all the players in the computation.
2. Whenever $\mathcal{A}$ requests to corrupt some $p_i \in \mathcal{P}$, $\mathcal{S}$ corrupts p_i and sends (the simulated) internal state of p_i to $\mathcal{A}$. From that point on, $\mathcal{S}$ has (the simulated) p_i follow $\mathcal{A}$'s instruction.
3. Whenever $\mathcal{A}$ sends a message to the environment $\mathcal{Z}$, $\mathcal{S}$ forwards this message to $\mathcal{Z}$.
4. At the end of the simulation, if some (simulated) uncorrupted player p_i outputs $x_i = x_s$, then in the ideal evaluation p_s sends x_s to $\mathcal{F}_{\text{UBC}}$ (even when he is corrupted). Otherwise, i.e., if $x_i \neq x_s$, $\mathcal{S}$ instructs p_s to send x_i to the functionality $\mathcal{F}_{\text{UBC}}$ in Step 3 (the correctness property of BC_{BGP} implies $x_i \neq x_s$ only when p_s is actively corrupted.).

It is easy to verify that $\text{EXEC}_{\mathcal{F}_{\text{UBC}},\mathcal{S},\mathcal{Z}} \equiv \text{EXEC}_{\pi,\mathcal{A},\mathcal{Z}}$, i.e., the protocol perfectly securely realizes $\mathcal{F}_{\text{UBC}}$. □

Remark 1. One can verify that most broadcast protocols in the literature, including those that assume a setup and tolerate $t < n$ corrupted players, securely realize the ideal functionality $\mathcal{F}_{\text{UBC}}$. The proof is along the lines of the above proof. One might even be willing to make a compromise and accept this functionality as a tight description of what one would expect from broadcast. However, we point out that this functionality allows the counter-intuitive behavior explained in the introduction. Furthermore, the security of protocols which assume broadcast should be (re-)analyzed with this functionality in mind.

For the second step, we show that if a protocol Π (which assumes broadcast) satisfies an appropriate pre-condition, then it is safe to instantiate broadcast in Π by calls to $\mathcal{F}_{\text{UBC}}$. The pre-condition is the following: For any value v which is supposed to be broadcasted, the adversary *"knows v in advance"*, i.e., there exists a *deterministic* strategy for this adversary to compute v based on the contents of her view *before* the call to the broadcast primitive. We formalize this in the following lemma. Note that the lemma holds for any security level and is not restricted to perfect security.[9]

Lemma 2. *Let Π be an $\mathcal{F}_{BC}$-hybrid protocol which securely realizes a given functionality $\mathcal{F}$, and let Π' denote the protocol which results by replacing in Π all the calls to $\mathcal{F}_{BC}$ with calls to $\mathcal{F}_{UBC}$. If Π uses calls to $\mathcal{F}_{BC}$ only to broadcast values which the adversary knows in advance, then Π' securely realizes $\mathcal{F}$.*

Proof. (sketch) Let $\mathcal{A}'$ be an adversary attacking Π' in the $\mathcal{F}_{\text{UBC}}$-hybrid model. We show how to construct an adversary $\mathcal{A}$ attacking Π in the $\mathcal{F}_{\text{BC}}$-hybrid model such that

[8] In fact, for any adversary $\mathcal{A}$, $\mathcal{S}$ can generate exactly the same messages as the uncorrupted players would if the protocol would be run with this adversary.

[9] However, for the case of computational security we will have to require that the strategy of the adversary to compute the value which is to be broadcasted is efficient.

$\mathrm{Exec}_{\mathcal{Z},\mathcal{A}',\Pi'} \equiv \mathrm{Exec}_{\mathcal{Z},\mathcal{A},\Pi}$. This is sufficient as then we can use the simulator for $\mathcal{A}$ (which is guaranteed to exist by the security of Π) as a simulator for $\mathcal{A}'$. $\mathcal{A}$ behaves exactly as $\mathcal{A}'$ except in the invocations of $\mathcal{F}_{\mathrm{UBC}}$: when $\mathcal{F}_{\mathrm{UBC}}$ is to be called, in order to simulate the first message of $\mathcal{F}_{\mathrm{UBC}}$ towards $\mathcal{A}'$ (corresponding to the broadcasted value) $\mathcal{A}$ computes the value to be broadcasted (using the deterministic strategy on his view which is guaranteed to exist by the fact that she knows the broadcasted value in advance)[10] and sends this value to $\mathcal{A}'$. $\mathcal{A}'$ is now allowed to corrupt the sender and (possibly) change the value he is supposed to broadcast. $\mathcal{A}$ acts accordingly and then invokes $\mathcal{F}_{\mathrm{BC}}$. It is straightforward to verify that $\mathrm{Exec}_{\mathcal{Z},\mathcal{A}',\Pi'} \equiv \mathrm{Exec}_{\mathcal{Z},\mathcal{A},\Pi}$. □

We point out that the [BGW88] VSS protocol satisfies the pre-condition of Lemma 2. Indeed, the protocol uses broadcast only for the complaints and the accusations issued by players and for the dealer to reply to them. By careful inspection of the protocol one can verify that all these broadcasted values can be computed from the view of the adversary before they are broadcasted. Because this VSS is secure for $t < n/3$, combining Lemmas 1 and 2 we get the following corollary.

Corollary 1. *Protocol $\mathit{VSS}_{BGW}^{(BGP)}$ perfectly t-securely realizes the functionality $\mathcal{F}_{BC}$ for $t < n/3$.*

Remark 2. Although replacing $\mathcal{F}_{\mathrm{BC}}$ by $\mathcal{F}_{\mathrm{UBC}}$ did not affect the security of [BGW88] VSS, this is not necessarily true for other protocols using broadcast. For example, the VSS in [CDD$^+$99] does not satisfy the pre-condition of Lemma 2. In fact, in [CDD$^+$99] uniformly random values are broadcasted. As demonstrated in the examples given in the introduction, broadcasting random values by a protocol which only securely realizes $\mathcal{F}_{\mathrm{UBC}}$ can have unexpected results. In fact, it is unclear whether or not [CDD$^+$99] is secure if we instantiate the assumed broadcast-channel by calls to $\mathcal{F}_{\mathrm{UBC}}$.

To complete this section we show that $t < n/3$ is tight for perfectly secure broadcast. We use the following impossibility result from [LSP82, KY84, FLM86] (for a nice proof see also [Fit03]).

Lemma 3 ([LSP82, KY84, FLM86, Fit03]). *For $t \geq n/3$ there exists no protocol which simultaneously satisfies correctness, consistency, and termination, even in the presence of a non-adaptive adversary.*

The impossibility proof for the functionality $\mathcal{F}_{\mathrm{BC}}$ follows directly from the above lemma and the fact that any protocol securely realizing $\mathcal{F}_{\mathrm{BC}}$ satisfies the given three properties.

Corollary 2. *For $t \geq n/3$ there exists no protocol which perfectly t-securely realizes the functionality $\mathcal{F}_{BC}$.*

We point out that Lemma 3, hence also the impossibility for $\mathcal{F}_{\mathrm{BC}}$, holds even for the cases of computational and statistical security *when no setup is available*. This implies the following:

Corollary 3. *When $t \geq n/3$ and no setup is available then there exists no protocol which computationally t-securely realizes the functionality $\mathcal{F}_{BC}$. The statement holds also for statistical security.*

[10] Wlog we can assume that $\mathcal{A}'$ forwards his entire view to $\mathcal{A}$ [Can00, Can01].

4 Statistical and Computational Security (with a Trusted Setup)

In this section we consider the cases of statistical security, i.e., information theoretic with negligible error-probability, and computational security. For these security notions, it is widely believed that when a setup allowing digital signatures is assumed, then broadcast is possible for an arbitrary number of cheaters (i.e., $t < n$), e.g., by using the Dolev-Strong broadcast protocol [DS82] for computational security or using [PW92] for statistical security. We show that this folklore belief is wrong when an adaptive adversary is considered. We already argued in the previous section that the Dolev-Strong broadcast protocol, denoted in the following as Π_{DS}, is not adaptively secure. In this section we show that the condition $t \leq n/2$ is necessary and sufficient for broadcast both for computational and statistical security.

We start by proving the sufficiency of the condition $t \leq n/2$; this is done by providing a protocol which securely realizes $\mathcal{F}_{\mathrm{BC}}$. We handle the two security notions, i.e., computational and statistical, in parallel. In our protocol, the players will need to digitally sign messages they send. This is modeled by assuming that the protocol has access to an ideal functionality for digital signatures $\mathcal{F}_{\mathrm{SIG}}$ (for definition and properties of such a functionality see [Can03]).

Analogously to the case of perfect security, our approach proceeds in two steps, namely we first show that there exists a secure realization of $\mathcal{F}_{\mathrm{UBC}}$ for $t \leq n/2$, and then use Lemma 2 to derive a protocol for $\mathcal{F}_{\mathrm{BC}}$ from an $\mathcal{F}_{\mathrm{UBC}}$-hybrid protocol. However, this last step is more involved than simply using a statistically secure VSS protocol satisfying the preconditions of Lemma 2. Indeed, on the one hand, all known protocols for statistical VSS are only secure for $t < n/2$ which is stronger than $t \leq n/2$. On the other hand, these protocols do not satisfy the pre-condition of Lemma 2. Before describing how to overcome these difficulties we state the following lemma which will allow us to use Π_{DS} as a secure realization of $\mathcal{F}_{\mathrm{UBC}}$. The proof is along the lines of the proof of Lemma 1; the only difference is that the simulator needs also to simulate the digital signatures of honest players in a run of the protocol, which is guaranteed to be possible by the definition of $\mathcal{F}_{\mathrm{SIG}}$.[11]

Lemma 4. *Protocol Π_{DS} perfectly t-securely realizes $\mathcal{F}_{UBC}$ for $t < n$ in the $\mathcal{F}_{SIG}$-hybrid model, where the signatures are replaced by calls to an ideal signature functionality $\mathcal{F}_{SIG}$.*

To implement the second step, namely construct the $\mathcal{F}_{\mathrm{UBC}}$-hybrid protocol realizing $\mathcal{F}_{\mathrm{BC}}$, we use as starting point the VSS from [CDD$^+$99]. In [CDD$^+$99] IC-signatures are used to ensure that some p_j who receives a value v from some p_i can, at a later point, publicly prove that p_i indeed send him v. Because IC-signatures are secure only when $t < n/2$, in this work we use digital signatures for the same purpose. The signatures are generated and verified by calls to the assumed digital signatures functionality $\mathcal{F}_{\mathrm{SIG}}$. We point out that the signed message should include enough information to uniquely identify for which message in the flow of the protocol the signature was issued (e.g., a unique message ID associated with every message sent in the protocol). Depending

[11] The idea of using [DS82] with i.t. secure signatures to get an i.t. secure broadcast protocol appears also in [PW92, Fit03].

on whether the calls to $\mathcal{F}_{\text{SIG}}$ are instantiated by a computationally or an i.t. secure signature-scheme, our broadcast protocol will achieve computational or i.t. security, respectively.

In the following, we first describe our sharing, which is along the lines of [CDD$^+$99], and specify some useful security properties, and then we describe and analyze our broadcast protocol.

Secret Sharing. Following the terminology of [CDD$^+$99], we say that a vector $v = (v_1, \ldots, v_m) \in \mathbb{F}^m$ is d-consistent, if there exists a polynomial $p(\cdot)$ of degree d such that $p(i) = v_i$ for $i = 1, \ldots, m$. A value s is said to be d-*shared* among the players in $\mathcal{P}$ when every (honest) player $p_i \in \mathcal{P}$ holds a degree-d polynomial $g_i(\cdot)$ and for each $p_j \in \mathcal{P}$ p_i also holds p_j's signature on $g_i(j)$, where the following condition holds: there exists a degree-d polynomial $q(\cdot)$ with $q(0) = s$ and $g_i(0) = q(i)$ for all p_i. The polynomials $g_1(\cdot), \ldots, g_n(\cdot)$ along with the corresponding signatures constitute a d-*sharing* of s.

We describe the protocols HD-Share (the HD stands for Honest Dealer) and Reconstruct which allow for a dealer p_D to d-share a value s, and for public reconstruction of a shared value, respectively.

The protocol HD-Share is along the lines of the sharing protocol from [CDD$^+$99]. The main difference from a standard sharing protocol is that the correctness of the output-sharing is guaranteed only when the dealer is honest until the end of the protocol. We describe HD-Share (see next page) in the $\{\mathcal{F}_{\text{SIG}}, \mathcal{F}_{\text{BC}}\}$-hybrid model, i.e., HD-Share uses calls to $\mathcal{F}_{\text{BC}}$ for broadcasting and calls to $\mathcal{F}_{\text{SIG}}$ for signature generation and verification. To ensure that the output of HD-Share matches the form of our sharing, i.e., every honest p_i holds a degree-d polynomial $g_i(\cdot)$ and signatures from all other players, we do the following: for every message transmission, the receiver p_j confirms when he receives a well-formed message from p_i or, otherwise, p_j complains and p_i is expected to answer the complaint by broadcasting the message. If some p_i is publicly caught to misbehave, e.g., by broadcasting a malformed message, then p_i is disqualified. Because dishonest players cannot be forced to sign the messages they send, we make the following convention: when p_i is disqualified, then every player takes a default value, denoted as $\perp$, to be p_i's signature on any message ($\perp$ will always be accepted as valid signature of disqualified players on any message).

Lemma 5. *Protocol* HD-Share *invoked in the* $\{\mathcal{F}_{BC}, \mathcal{F}_{SIG}\}$*-hybrid model achieves the following: The view of any* d*-adversary attacking the protocol can be perfectly simulated (privacy);*[12] *When the dealer is honest until the end of* HD-Share *then the output is a* d*-sharing of* s *(honest-dealer correctness). Furthermore, in all calls to* $\mathcal{F}_{BC}$*, the adversary "knows in advance" the value to be broadcasted.*

Proof. (sketch) Clearly the adversary "knows in advance" all the values to be broadcasted, as these are accusations and replies which might only occur if at least one of the disputing players is corrupted. The privacy of HD-Share can be argued along the lines of [CDD$^+$99]. Nevertheless we sketch how the simulator $\mathcal{S}$ simulates the view of the adversary $\mathcal{A}$: The simulation of the signatures is trivial, as the functionality $\mathcal{F}_{\text{SIG}}$

[12] This ensures that no information about s leaks to a d-adversary.

Protocol HD-Share$_d$ (p_D, s)

1. The dealer p_D chooses a uniformly random bivariate polynomial $f(\cdot,\cdot)$ of degree d in each variable, such that $f(0,0) = s$. For each $p_i \in \mathcal{P}$:
   (a) For $j = 1, \ldots, n$: p_D sends p_i the values $s_{i,j} = f(i,j)$ and $s_{j,i} = f(j,i)$ along with his signature on them; p_i denotes the received values as $s^{(i)}_{i,j}, s^{(i)}_{j,i}, sig_{p_D}(s^{(i)}_{i,j})$, and $sig_{p_D}(s^{(i)}_{j,i})$.
   (b) p_i broadcasts a complaint if any of the vectors $\left(s^{(i)}_{i,1}, \ldots, s^{(i)}_{i,n}\right)$ and $\left(s^{(i)}_{1,i}, \ldots, s^{(i)}_{n,i}\right)$ is not d-consistent or if for some value no valid signature was received.
   (c) p_D answers each complaint by broadcasting the values he sent to p_i in Step 1a. If p_D broadcasts a message of the wrong form or invalid signatures then p_D is disqualified; otherwise p_i adopts the broadcasted messages as the messages he should have received in Step 1a.
2. For each $p_i \in \mathcal{P}$:
   (a) For $j = 1, \ldots, n$: p_i sends $s^{(i)}_{i,j}$ to p_j along with his signature $sig_{p_i}(s^{(i)}_{i,j})$ and the dealer's signature $sig_{p_D}(s^{(i)}_{i,j})$.
   (b) Each $p_j \in \mathcal{P}$ broadcasts a complaint if he did not receive a message along with valid signatures from p_i and p_D in Step 2a.
   (c) p_i answers each complaint by broadcasting $(s^{(i)}_{i,j}, sig_{p_D}(s^{(i)}_{i,j}), sig_{p_i}(s^{(i)}_{i,j}))$. If p_i does not broadcast a message or any of the signatures is invalid then p_i is disqualified, every player replaces all p_i's signatures by $\perp$, and p_j adopts $s^{(j)}_{i,j}$ as the value he should have received in Step 2a; otherwise p_j adopts the broadcasted messages as the messages he should have received in Step 2a.
3. Every p_i checks if he received a $s^{(j)}_{i,j}$ from some p_j in Step 2 which is inconsistent with his own view of $s_{i,j}$, i.e., $s^{(j)}_{i,j} \neq s^{(i)}_{i,j}$, and if so, broadcasts $\left(s^{(i)}_{i,j}, s^{(j)}_{i,j}, sig_{p_D}(s^{(i)}_{i,j}), sig_{p_D}(s^{(j)}_{i,j})\right)$; every player verifies that $s^{(j)}_{i,j} \neq s^{(i)}_{i,j}$ and that the signatures are valid and if so p_D is disqualified.[a]

[a] Recall that the signature includes information to uniquely identify for which message in the flow of the protocol it was generated, e.g. a unique message ID, and also includes a unique session ID.

allows $\mathcal{S}$ to choose the actual signature. As long as $\mathcal{A}$ does not corrupt p_D, the simulator proceeds as follows: whenever $\mathcal{A}$ requests to corrupt some p_i, $\mathcal{S}$ creates a simulated view for p_i (up to the current simulated round) by choosing all the values in p_i's view uniformly at random except those that have already appeared in the view of $\mathcal{A}$ (e.g., if p_j has been already corrupted then the values $s^{(i)}_{i,j} = s^{(j)}_{i,j}$ and $s^{(i)}_{j,i} = s^{(j)}_{j,i}$ have been already given to the adversary). Note that, because the sharing polynomial is of degree d, from the point of view of the d-adversary $\mathcal{A}$ the simulated views are distributed as in a real run of the protocol HD-Share. If at some point $\mathcal{A}$ requests to corrupt p_D, then at that point $\mathcal{S}$ learns p_D's input s and, can simulate the view of all the remaining players while making sure that all simulated values are consistent with some degree-d polynomial $f'(\cdot,\cdot)$ with $f'(0,0) = s$. Honest-dealer correctness is proved as follows:

When the dealer is honest at the end of HD-Share, then only values which lie on the actual polynomial $f(\cdot,\cdot)$ appear in the output of honest players. Moreover, every honest p_i holds all the signatures he should hold, as otherwise p_i would have complained in Step 5 and exposed the inconsistency. Therefore, the output will be a d-sharing of the dealer's value. □

To reconstruct a sharing the protocol Reconstruct is invoked (see below). The idea is the following: every p_i broadcasts his share and the corresponding signatures from the players in $\mathcal{P}$; if some signature is invalid or the the announced share is not d-consistent, then p_i is excluded from the reconstruction, otherwise his share-polynomial is interpolated; The zero-coefficients of the share-polynomials of the players that have not been excluded are used to reconstruct the shared value. Depending on the actual choice of d and the number of corrupted parties, the sharing might not uniquely define a value. In any case the players adopt the value which is output by the interpolation algorithm. The consistency of the output is guaranteed as it is decided on publicly seen values.

Protocol Reconstruct

1. Each $p_i \in \mathcal{P}$ broadcasts $(s_{i,1},\ldots,s_{i,n})$ along with the corresponding signatures $sig_{p_1}(s_{i,1}),\ldots,sig_{p_n}(s_{i,n})$; if any of the broadcasted signatures is invalid or if the broadcasted vector is not d-consistent, then p_i is disqualified. Otherwise a polynomial $g_i(\cdot)$ is defined by interpolating the components of the vector.
2. Let $P_{\text{``ok''}} = \{p_{i_1},\ldots,p_{i_\ell}\}$ denote the set of non-disqualified players. The values $g_{i_1}(0),\ldots,g_{i_\ell}(0)$ are used to interpolate a polynomial $g'(\cdot)$ and every player outputs $g'(0)$.

Lemma 6. *Protocol* Reconstruct *invoked to the* $\{\mathcal{F}_{BC},\mathcal{F}_{SIG}\}$*-hybrid model outputs (the same)* $y \in \mathbb{F}$ *towards every player. Furthermore, if* $d < n - t$ *(where* t *is the number of corrupted players) and the input is a* d*-consistent sharing of some* s*, then* $y = s$*.*

Proof. (sketch) As the output is decided based on values which are agreed upon using $\mathcal{F}_{\text{BC}}$, all players output the same value y. Furthermore, when $d < n-t$ then there are at least $t+1$ honest players. When additionally the input is a d-consistent sharing then the values which the honest players have signed uniquely define all the share-polynomials $g_i(\cdot)$. Because $\mathcal{F}_{\text{SIG}}$ never verifies as valid a signature on a value which was not signed by the corresponding player, the adversary cannot announce a polynomial other than $g_i(\cdot)$ for any $p_i \in \mathcal{P}$. Hence, every corrupted p_i either announces the correct value or is disqualified. However, the honest players always announce the correct values, hence the correct polynomials are interpolated. Because there are at least $d+1$ honest players there will always be at least $d+1$ values to interpolate the correct $g'(\cdot)$ and recover the shared value. □

We next describe our broadcast protocol for $t \leq n/2$. The idea is to have the sender p_s share his input x_s by a degree-$(t-1)$ sharing using HD-Share with $d = t-1$ and subsequently invoke Reconstruct on the output of HD-Share. The intuition is the following: When p_s is honest until the end of HD-Share, then HD-Share outputs a $(t-1)$-sharing of x_s (Lemma 5: honest-dealer correctness); as $t \leq n/2$ implies $d =$

$t-1 < n-t$, Lemma 6 guarantees that Reconstruct will output x_s. Hence, the only way the adversary can change the output to some $s' \neq s$ is by corrupting p_s during (or before) protocol HD-Share. As there are at most t corrupted players, if the adversary wishes to corrupt p_s then she can corrupt at most $t-1$ of the remaining players; as a $(t-1)$-adversary gets no information on x_s (Lemma 5: privacy), the decision whether or not to corrupt p_s has to be taken independently of x_s, which is a behavior that can be easily simulated.

In the above, we managed to tweak the VSS protocol from [CDD$^+$99] (which is secure if and only if $t < n/2$) so that we can use it for broadcast when $t \leq n/2$. However, as already mentioned, both HD-Share and Reconstruct use calls to $\mathcal{F}_{\text{BC}}$ for broadcasting. In order to replace $\mathcal{F}_{\text{BC}}$ by $\mathcal{F}_{\text{UBC}}$, we need to make sure that the precondition of Lemma 2 is satisfied (i.e., the adversary "knows in advance" all broadcasted values). For protocol HD-Share this is guaranteed by Lemma 5. However, the values which are broadcasted in Reconstruct are not necessarily known to the adversary in advance. We resolve this by a technical trick, namely we introduce a *dummy* step between HD-Share and Reconstruct where every player sends to every other player his output from protocol HD-Share. Observe that such a modification could potentially give an advantage to the adversary. But this might only happen in case the adversary has not corrupted p_s by the end of HD-Share, as otherwise she knows all the outputs by then. However, even in this bad case, because p_s is honest until the end of HD-Share, by the time the dummy step is executed the output is already fixed to x_s, and the adversary cannot change it even with access to the full transcript. For completeness we include a description of our broadcast protocol and state its achieved security. The proof of the lemma can be found in the appendix.

```
Protocol Broadcast $(p_s, x_s)$
  1. Invoke HD-Share$_{t-1}(p_s, x_s)$; if $p_s$ is disqualified then every player outputs a
     default value, e.g., 0 and halts.
  2. Every $p_i \in \mathcal{P}$ sends his output from HD-Share to every $p_j \in \mathcal{P}$.
  3. Invoke Reconstruct on the output of HD-Share.
```

Lemma 7. *Protocol* Broadcast *perfectly t-securely realizes the functionality $\mathcal{F}_{BC}$ in the $\{\mathcal{F}_{UBC}, \mathcal{F}_{SIG}\}$-hybrid model, for $t \leq n/2$.*

Proof. (sketch) By inspection of the protocol one can verify that the pre-conditions of Lemma 2 are satisfied for every value which is broadcasted, hence using $\mathcal{F}_{\text{UBC}}$ for broadcasting values in protocol Broadcast is as secure as using $\mathcal{F}_{\text{BC}}$. Therefore, it suffices to argue the security of Broadcast in the $\{\mathcal{F}_{\text{BC}}, \mathcal{F}_{\text{SIG}}\}$-hybrid model. We sketch the simulator $\mathcal{S}$ for a given adversary $\mathcal{A}$. During the execution of HD-Share, the simulator behaves as the simulator in the proof of Lemma 5. In the subsequent steps, if $\mathcal{A}$ has already corrupted p_s before the end of the simulated run of HD-Share, then at that point $\mathcal{S}$ has learned the sender's input x_s and can simulate the remaining transcript as in the proof of Lemma 5 ($\mathcal{S}$ can clearly simulate all the messages exchanged in Steps 2 and 3 as they appear in the transcript of HD-Share). Otherwise, i.e., if by the end of the simulated run of HD-Share the adversary $\mathcal{A}$ has not requested to corrupt p_s, then

$\mathcal{S}$ allows for the invocation of $\mathcal{F}_{\mathrm{BC}}$, where p_s gives his input x_s; $\mathcal{S}$ learns x_s from $\mathcal{F}_{\mathrm{BC}}$ (as the output of any corrupted player) and can, same as before, simulate the remaining transcript. □

Combining the above lemma with Lemma 4 we get the following.

Corollary 4. *If $t \leq n/2$ and a statistically (resp. computationally) secure signature scheme is available then the above protocol statistically (resp. computationally) t-securely realizes the functionality $\mathcal{F}_{BC}$.*

To complete this section, we show that the condition $t \leq n/2$ is necessary for adaptively secure synchronous broadcast both for i.t. and for computational security. The idea of the proof is the following: Because the adversary can corrupt half of the players in $\mathcal{P} \setminus \{p_s\}$, she can be the first to learn noticeable information on the dealers input, before the honest players in $\mathcal{P} \setminus \{p_s\}$ jointly learn noticeable information. Depending on this information the adversary can corrupt the sender and, with overwhelming probability, change the output to some other value. However this behavior cannot be simulated.

Lemma 8. *There exists no protocol which computationally t-securely realizes the functionality $\mathcal{F}_{BC}$ for $t > n/2$, not even in the $\{\mathcal{F}_{SIG}\}$-hybrid model. The statement holds also for statistical security.*

Proof. To arrive at a contradiction, assume that there exists a computationally (resp. statistically) t-secure Broadcast protocol Π. Wlog, assume that p_s uses Π to broadcast a uniformly random $x_s \in_R \mathbb{F}$. For every round i, protocol Π implicitly assigns to every set $\mathcal{P}' \subseteq \mathcal{P}$ a probability $\Pr_{\mathcal{P}',x_s,\Pi,i}$, which is the probability of the best efficient adversary corrupting $\mathcal{P}'$ to output x_s based only on her view in Π up to round i. For all $\mathcal{P}' \subseteq \mathcal{P} \setminus \{p_s\}$ this probability is negligible if i is the first round of Π and overwhelming if i is the last round of Π. As the total number of rounds in Π is polynomial, for each $\mathcal{P}' \subseteq \mathcal{P} \setminus \{p_s\}$ there exists a round, denoted as $i_{\mathcal{P}'}$, where this probability from negligible becomes noticeable, i.e., not negligible. The adversary corrupts the set $A \subseteq \mathcal{P} \setminus \{p_s\}$ with $|A| = t - 1$ such that $i_A = \min\{i_{\mathcal{P}'} \mid \mathcal{P}' \subseteq \mathcal{P} \land |\mathcal{P}'| \leq t - 1\}$. In round i_A, the adversary gets the values which are sent to corrupted players and runs the best (efficient) strategy to compute x_s on input the view of the players in A; denote by x' the output of this protocol (by our assumption, $x' = x_s$ with noticeable probability). Let $\mathbb{F}_{1/2}$ denote the set of first $|\mathbb{F}|/2$ (in any ordering) elements in $\mathbb{F}$. If $x' \in \mathbb{F}_{1/2}$ then the adversary acts as a passive adversary (i.e., all corrupted players are instructed to correctly execute their protocol). Otherwise, i.e., if $x' \in \mathbb{F} \setminus \mathbb{F}_{1/2}$, then the adversary actively corrupts p_s and forces all the actively corrupted players to crash before sending any message in round i; as $|\mathcal{P} \setminus A| \leq t - 1$ we know that $i_{\mathcal{P} \setminus A} \geq i_A$, hence, because the players in $\mathcal{P} \setminus A$ do not get the messages from round i_A, with overwhelming probability the output of the honest players will be in $\mathbb{F} \setminus \{x_s\}$. With this strategy the adversary achieves that when $x_s \in \mathbb{F} \setminus \mathbb{F}_{1/2}$, then the output of the honest players in Π is different than x_s with noticeable probability. However the simulator cannot simulate this behavior as he has to decide whether or not to corrupt p_s and change the output independent of x_s. □

5 Other Models

We presented our solutions in the secure-channels model, because in this model it is clear that the only way the adversary can learn a transmitted message is by corrupting the sender or the receiver (after the message has been received). In particular, this implies that for a message sent to the trusted party/functionality, as long as the sender is honest, the simulator cannot learn the sent message before the functionality learns it.

In the authenticated-channels model (without privacy), the same composability issue appears as the one we deal with in this work. However, as it is typically the case, one could pretend to solve this issue by giving the simulator additional power on the communication network. For example, if the simulator is allowed to read the sent message, delete it from the channel, and then corrupt the sender and re-send the message, then the above problem disappears "by definition". It is arguable, however, how consistent such a model is with the synchronicity assumption on the communication network. Furthermore, when defining such an authenticated communication model which eliminates the composability issue presented in this work, one has to keep in mind that the described protocols typically are not composable when the authenticated-channels are replaced by secure-channels.

Also, in the case of asynchronous communication, the same problem appears. Take for example the asynchronous secure-channels model as defined in [BCG93, BKR94]. As in the synchronous case, unless we assume some kind of asynchronous atomic multi-send, when a player p is instructed to send a message to several other players,[13] then it might happen that the adversary first learns the message by corrupting one of the receivers, and still is able to corrupt p and change it. In fact, as already mentioned, this is the case in the UC framework [Can01]. Clearly this behavior cannot be simulated in the ideal world. As in the synchronous case, one might be willing to make a compromise and accept an asynchronous version of $\mathcal{F}_{\text{UBC}}$ to be the desired ideal functionality for broadcast.

6 Conclusions

We considered the problem of securely realizing broadcast in the secure-channels model. In this model, it has been shown that there exist protocols satisfying the property-based definition of broadcast and tolerating a t-adversary, if and only if $t < n/3$ when perfect security is considered. For unconditional and computational security, when a setup allowing digital signatures is given, the corresponding bound is $t < n$.

We showed that the property-based definition of broadcast does not imply the simulation-based definition for the natural broadcast functionality. Furthermore, we showed that most known broadcast protocols do not realize this functionality in the secure-channels model. As a result, if one replaces the broadcast invocations in any of the known multi-party protocols for the secure-channels model, e.g., [BGW88, RB89, CDD$^+$99], by one of the known broadcast protocols, the security of the resulting protocol cannot be argued using the composition theorems.

[13] Observe that this is the way in which most asynchronous broadcast protocols start, e.g., [Bra84].

We described protocols which securely realize the (natural) ideal functionality for broadcast for each of the three security notions. For the case of perfect security, we showed that the tight bound matches the corresponding bound for the property-based definition, i.e., $t < n/3$. However, for the cases of statistical and computational security (with setup assumptions) the necessary and sufficient bound is $t \leq n/2$. Furthermore, we described a weaker ideal functionality for broadcast which is securely realized by the known protocols but achieves less than what one expects from a broadcast protocol. Of course, one might be willing to make a compromise and accept this as the desired ideal functionality for broadcast. But in that case, all known protocols should be (re-)analyze with this weaker ideal functionality in mind.

References

[BCG93] Ben-Or, M., Canetti, R., Goldreich, O.: Asynchronous secure computation. In: STOC 1993, pp. 52–61 (1993)

[BDDS92] Bar-Noy, A., Dolev, D., Dwork, C., Strong, H.R.: Shifting gears: Changing algorithms on the fly to expedite Byzantine agreement. Information and Computation 97(2), 205–233 (1992)

[BGP89] Berman, P.J., Garray, J., Perry, J.: Towards optimal distributed consensus. In: FOCS 1989, pp. 410–415 (1989); Full version in Computer Science Research (1992)

[BGW88] Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: STOC 1988, pp. 1–10 (1988)

[BHR07] Beerliova-Trubiniova, Z., Hirt, M., Riser, M.: Efficient Byzantine agreement with faulty minority. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 393–409. Springer, Heidelberg (2007)

[BKR94] Ben-Or, M., Kelmer, B., Rabin, T.: Asynchronous secure computations with optimal resilience (extended abstract). In: PODC 1994, pp. 183–192. ACM, New York (1994)

[BPW91] Baum-Waidner, B., Pfitzmann, B., Waidner, M.: Unconditional Byzantine agreement with good majority. In: Jantzen, M., Choffrut, C. (eds.) STACS 1991. LNCS, vol. 480, pp. 285–295. Springer, Heidelberg (1991)

[Bra84] Bracha, G.: An asynchronou [(n-1)/3]-resilient consensus protocol. In: PODC 1984, pp. 154–162 (1984)

[Can00] Canetti, R.: Security and composition of multiparty cryptographic protocols. Journal of Cryptology 13(1), 143–202 (2000)

[Can01] Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: FOCS 2001, pp. 136–145 (2001)

[Can03] Canetti, R.: Universally composable signatures, certification and authentication. Cryptology ePrint Archive, Report 2003/239 (2003), http://eprint.iacr.org/

[CCD88] Chaum, D., Crépeau, C., Damgård, I.: Multiparty unconditionally secure protocols (extended abstract). In: STOC 1988, pp. 11–19 (1988)

[CDD$^+$99] Cramer, R., Damgård, I., Dziembowski, S., Hirt, M., Rabin, T.: Efficient multiparty computations secure against an adaptive adversary. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 311–326. Springer, Heidelberg (1999)

[CGMA85] Chor, B., Goldwasser, S., Micali, S., Awerbuch, B.: Verifiable secret sharing and achieving simultaneity in the presence of faults. In: FOCS 1985, pp. 383–395 (1985)

[CR87] Chor, B., Rabin, M.O.: Achieving independence in logarithmic number of rounds. In: PODC 1987, pp. 260–268 (1987)

[CW89] Coan, B.A., Welch, J.L.: Modular construction of nearly optimal Byzantine agreement protocols. In: PODC 1989, pp. 295–305 (1989); Full version in Information and Computation (1992)

[DFF$^+$82] Dolev, D., Fischer, M.J., Fowler, R., Lynch, N.A., Strong, H.R.: An efficient algorithm for Byzantine agreement without authentication. Information and Control 52(3), 257–274 (1982)

[DS82] Dolev, D., Strong, H.R.: Polynomial algorithms for multiple processor agreement. In: STOC 1982, pp. 401–407 (1982); Full version in *SIAM Journal on Computing* 12(4), 656–666 (1983)

[Fit03] Fitzi, M.: Generalized Communication and Security Models in Byzantine Agreement. PhD thesis, ETH Zurich (2003)

[FLM86] Fischer, M.J., Lynch, N.A., Merritt, M.: Easy impossibility proofs for distributed consensus problems. Distributed Computing 1, 26–39 (1986)

[FM88] Feldman, P., Micali, S.: Optimal algorithms for Byzantine agreement. In: STOC 1988, pp. 148–161 (1988)

[Gen95] Gennaro, R.: Achieving independence efficiently and securely. In: PODC 1995, pp. 130–136 (1995)

[Gen00] Gennaro, R.: A protocol to achieve independence in constant rounds. IEEE Trans. Parallel Distrib. Syst. 11(7), 636–647 (2000)

[GL02] Goldwasser, S., Lindell, Y.: Secure computation without agreement. In: Malkhi, D. (ed.) DISC 2002. LNCS, vol. 2508, pp. 17–32. Springer, Heidelberg (2002)

[GM93] Garay, J.A., Moses, Y.: Fully polynomial Byzantine agreement in t+1 rounds. In: STOC 1993, pp. 31–41 (1993)

[Hev06] Hevia, A.: Universally composable simultaneous broadcast. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 18–33. Springer, Heidelberg (2006)

[HM05] Hevia, A., Micciancio, D.: Simultaneous broadcast revisited. In: PODC 2005, pp. 324–333 (2005)

[KY84] Karlin, A., Yao, A.C.: Manuscript (1984)

[LLR02] Lindell, Y., Lysyanskaya, A., Rabin, T.: On the composition of authenticated Byzantine agreement. In: STOC 2002, pp. 514–523 (2002)

[LSP82] Lamport, L., Shostak, R., Pease, M.: The Byzantine generals problem. ACM Transactions on Programming Languages and Systems 4(3), 382–401 (1982)

[Nie03] Nielsen, J.B.: On Protocol Security in the Cryptographic Model. PhD thesis, BRICS (2003)

[PW92] Pfitzmann, B., Waidner, M.: Unconditional Byzantine agreement for any number of faulty processors. In: STACS 1992. LNCS, vol. 577, pp. 339–350 (1992)

[RB89] Rabin, T., Ben-Or, M.: Verifiable secret sharing and multiparty protocols with honest majority. In: STOC 1989, pp. 73–85 (1989)

[TPS87] Toueg, S., Perry, K.J., Srikanth, T.K.: Fast distributed agreement. SIAM J. Comput. 16(3), 445–457 (1987)

Universally Composable Quantum Multi-party Computation[⋆]

Dominique Unruh

Saarland University

Abstract. The Universal Composability model (UC) by Canetti (FOCS 2001) allows for secure composition of arbitrary protocols. We present a quantum version of the UC model which enjoys the same compositionality guarantees. We prove that in this model statistically secure oblivious transfer protocols can be constructed from commitments. Furthermore, we show that every statistically classically UC secure protocol is also statistically quantum UC secure. Such implications are not known for other quantum security definitions. As a corollary, we get that quantum UC secure protocols for general multi-party computation can be constructed from commitments.

1 Introduction

Since the inception of quantum key distribution by Bennett and Brassard [4], it has been known that quantum communication permits to achieve protocol tasks that are impossible given only a classical channel. For example, a quantum key distribution scheme [4] permits to agree on a secret key that is statistically secret, using only an authenticated but not secret channel. (By statistical security we mean security against computationally unbounded adversaries, also known as information-theoretical security.) In contrast, when using only classical communication, it is easy to see that such a secret key can always be extracted by a computationally sufficiently powerful adversary. Similarly, based on an idea by Wiesner [25], Bennett, Brassard, Crépeau, and Skubiszewska [5] presented a protocol that was supposed to construct a statistically secure oblivious transfer[1] protocol from a commitment, another feat that is easily seen to be impossible classically.[2] Oblivious transfer, on the other hand, has been recognized by Kilian [15] to securely evaluate arbitrary functions. Unfortunately, the protocol of Bennett et al. could, at the time, not be proven secure, and the first complete proof

[⋆] Funded by the Cluster of Excellence "Multimodal Computing and Interaction".

[1] In an oblivious transfer protocol, Alice holds two bitstrings m_0, m_1, and Bob a bit c. Bob is supposed to get m_c but not m_{1-c}, and Alice should not learn c.

[2] We remark that, on the other hand, Mayers [16] shows that also in the quantum case, constructing a statistically secure commitment scheme *without any additional assumption* is impossible. However, under additional assumptions like in the quantum bounded storage model by Damgård, Fehr, Salvail, and Schaffner [10], statistically secure bit commitment is possible. See Section 1.1 for a discussion of the implications of Mayers' impossibility result for our result.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 486–505, 2010.

of (a variant of) that protocol was given almost two decades later by Damgård, Fehr, Lunemann, Salvail, and Schaffner [9].

Yet, although the oblivious transfer protocol satisfies the intuitive secrecy requirements of oblivious transfer, in certain cases the protocol might lose its security when used in a larger context. In other words, there are limitations on how the protocol can be composed. For example, no security guarantee is given when several instances of the protocol are executed concurrently (see the full version [21] for a more detailed explanations of the various restrictions).

The problem of composability has been intensively studied by the classical cryptography community (here and in the following, we use the word classical as opposed to quantum). To deal with this problem in a general way, Canetti [7] introduced the notion of Universal Composability, UC for short (Pfitzmann and Waidner [19] independently introduced the equivalent Reactive Simulatability framework). The UC framework allows to express the security of a multitude of protocol tasks in a unified way, and any UC-secure protocol automatically enjoys strong composability guarantees (so-called universal composability). In particular, such a protocol can be run concurrently with others, and it can be used as a subprotocol of other protocols in a general way. Ben-Or and Mayers [3] and Unruh [20] have shown that the idea of UC-security can be easily adapted to the quantum setting and have independently presented quantum variants of the UC notion. These notions enjoy the same strong compositionality guarantees. Shortly afterwards, Ben-Or, Horodecki, Leung, Mayers, and Oppenheim [2] showed that many quantum key distribution protocols are quantum-UC-secure.

Our contribution. In this work, we use the UC framework to show the existence of a statistically secure and universally composable oblivious transfer protocol that uses only a commitment scheme. Towards this goal, we first present a new definition of quantum-UC-security. In our opinion, our notion is technically simpler than the notions of Ben-Or and Mayers [3] and Unruh [20]. We believe that this may also help to increase the popularity of this notion in the quantum cryptography community and to show the potential for using UC-security in the design of quantum protocols. Second, we show that a variant of the protocol by Bennett et al. [5] is indeed a UC-secure oblivious transfer protocol. By composing this protocol with a UC-secure protocol for general multi-party computations by Ishai, Prabhakaran, and Sahai [13], we get UC-secure protocols for general multi-party computations using only commitments and a quantum channel – this is easily seen to be impossible in a purely classical setting.

UC-secure quantum oblivious transfer. The oblivious transfer (OT) protocol used in this paper is essentially the same as the protocol proposed by Damgård et al. [9] which in turn is based on a protocol by Bennett et al. [5]. The basic idea of the protocol is that Alice encodes a random sequence $\tilde{x}$ of bits as a quantum state, each bit randomly either in the computational basis or in the diagonal basis.[3] Then Bob is supposed to measure all qubits, this time in random bases of his choosing. Then Alice sends the bases she used to Bob. Let $I_=$ denote the set of indices of the bits $\tilde{x}_i$ where Alice and Bob chose the same basis, and $I_{\neq}$ the set of indices of the bits where Alice and Bob chose different bases. Assume that Bob wants to receive the message m_c out of Alice's messages m_0, m_1. Then Bob sets $I_c := I_=$ and $I_{1-c} := I_{\neq}$ and sends (I_0, I_1) to Alice. Alice will not know which of these two sets is which and hence does not learn c. Bob will know the bits $\tilde{x}_i$ at indices $i \in I_c$. But even a dishonest Bob, assuming that he measured the whole quantum state, will not know the bits at indices $i \in I_{1-c}$ since he used the wrong bases for these bits. Thus Alice uses the bits at I_0 to mask her message m_0, and the bits at I_1 to mask her message m_1. Then Bob can recover m_c but not m_{1-c}. (To deal with the fact that a malicious Bob might have partial knowledge about the bits at I_{1-c}, we use so-called privacy amplification to extract a near uniformly mask from these bits.)

The problem with this analysis is that we have assumed that a malicious Bob measures the whole quantum state upon reception. But instead, Bob could store the quantum state until he learns the bases that Alice used, and then use these bases to measure all bits $\tilde{x}_i$ accurately. Hence, we need to force a dishonest Bob to measure all bits before Alice sends the bases. The idea of Bennett et al. [5] is to introduce the following test: Bob has to commit to the bases he used and to his measurement outcomes. Then Alice picks a random subset of the bits, and Bob opens the commitments on his bases and outcomes corresponding to this subset of bits. Alice then checks whether Bob's measurement outcomes are consistent with what Alice sent. If Bob does not measure enough bits, then he will commit to the wrong values in many of the commitments, and there will be a high probability that Alice detects this.

It was a long-standing open problem what kind of a commitment needs to be used in order for this protocol to be secure. Damgård et al. [9] give criteria for the commitment scheme under which the OT protocol can be proven to have so-called stand-alone security; stand-alone security, however, does not give as powerful compositionality guarantees as UC-security. In order to achieve UC-security, we assume that the commitment is given as an ideal functionality. Then we have to show UC-security in the case of a corrupted Alice, and UC-security in the case of a corrupted Bob. The case of a corrupted Alice is simple, as one can easily see that no information flows from Bob to Alice (the commitment functionality does, by definition, not leak any information about the committed values). The case of a corrupted Bob is more complex and requires a careful analysis about the amount of information that Bob can retrieve about Alice's bits. Such an analysis has already been performed by Damgård et al. [9] in their setting. Fortunately, we do not need to repeat the analysis. We show that under certain special conditions, stand-alone security already implies UC-security. Since in the case of a corrupted Bob,

[3] If we were to use photons for transmission, in the computational basis we might encode the bit 0 as a vertically polarized photon and the bits 1 as a horizontally polarized photon. In the diagonal basis we might encode the bit 0 as a 45°-polarized photon, and the bit 1 as a 135°-polarized photon.

these conditions are fulfilled, we get the security in the case of a corrupted Bob as a corollary from the work by Damgård et al. [9].

In Section 4, we show that the OT protocol by Damgård et al. [9], when using an ideal functionality for the commitment, is statistically quantum-UC-secure. Furthermore, the universal composition theorem guarantees that we can replace the commitment functionality by any quantum-UC-secure commitment protocol.

Quantum lifting and multi-party computation. We are now equipped with a statistically quantum-UC-secure OT protocol π_{QOT} in the commitment-hybrid model. As noted first by Kilian [15], OT can be used for securely evaluating arbitrary functions, short, OT is complete for multi-party computation. Furthermore, Ishai, Prabhakaran, and Sahai [13] showed that for any functionality $\mathcal{G}$ (even interactive functionalities that proceed in several rounds), there is a classical protocol $\rho^{\mathcal{F}_{\mathrm{OT}}}$ in the OT-hybrid model that statistically classical-UC-emulates $\mathcal{G}$. Thus, to get a protocol for $\mathcal{G}$ in the commitment-hybrid model, we simply replace all invocations to $\mathcal{F}_{\mathrm{OT}}$ by invocations of the subprotocol π_{QOT}, resulting in a protocol $\rho^{\pi_{\mathrm{QOT}}}$. We then expect that the security of $\rho^{\pi_{\mathrm{QOT}}}$ follows directly using the universal composition theorem (in its quantum variant). There is, however, one difficulty: To show that $\rho^{\pi_{\mathrm{QOT}}}$ statistically quantum-UC-emulates $\mathcal{G}$, the universal composition theorem requires that the following premises are fulfilled: π_{QOT} statistically quantum-UC-emulates $\mathcal{F}_{\mathrm{OT}}$, and $\rho^{\mathcal{F}_{\mathrm{OT}}}$ statistically quantum-UC-emulates $\mathcal{G}$. But from the result of Ishai et al. [13] we only have that $\rho^{\mathcal{F}_{\mathrm{OT}}}$ statistically *classical*-UC-emulates $\mathcal{G}$. Hence, we first have to show that the same result also holds with respect to quantum-UC-security. Fortunately, we do not have to revisit the proof of Ishai et al., because we show the following general fact:

Theorem 1 (Quantum lifting theorem – informal). *If the protocols π and ρ are classical protocols, and π statistically classical-UC-emulates ρ, then π statistically quantum-UC-emulates ρ.*

Combining this theorem with the universal composition theorem, we immediately get that $\rho^{\pi_{\mathrm{QOT}}}$ statistically quantum-UC-emulates $\mathcal{G}$. In other words, any multi-party computation can be performed securely using only a commitment and a quantum-channel. In contrast, we show that in the classical setting a commitment is not even sufficient to compute the AND-function.

We stress that a property like the quantum lifting theorem should not be taken for granted. For example, for the so-called stand-alone model as considered by Fehr and Schaffner [11], no corresponding property is known. A special case of security in the stand-alone model is the zero-knowledge property: The question whether protocols that are statistical zero-knowledge with respect to classical adversaries are also zero-knowledge with respect to quantum adversaries has been answered positively by Watrous [23] for particular protocols, but is still open in the general case.

1.1 How to Interpret Our Result

We show that we can perform arbitrary statistically UC-secure multi-party computations, given a quantum channel and a commitment. However, Mayers [16] has shown that, even in the quantum setting, statistically secure commitment schemes do not exist, not even with respect to security notions much weaker than quantum-UC-security. In the light of this result, the reader may wonder whether our result is not vacuous. To illustrate why our result is useful even in the light of Mayers' impossibility result, we present four possible application scenarios.

Weaker computational assumptions. The first application of our result would be to combine our protocols with a commitment scheme that is only *computationally* quantum-UC-secure. Of course, the resulting multi-party computation protocol would then not be *statistically* secure any more. However, since commitment intuitively seems to be a simpler task than oblivious transfer, constructing a computationally quantum-UC-secure commitment scheme might be possible using simpler computational assumptions, and our result then implies that the same computational assumptions can be used for general multi-party computation.

Physical setup. One might seek a direct physical implementation of a commitment, such as a locked strongbox (or an equivalent but technologically more advanced construct). With our result, such a physical implementation would be sufficient for general multi-party computation. In contrast, in a classical setting one would be forced to try to find physical implementations of OT. It seems that a commitment might be a simpler physical assumption than OT (or at least an incomparable one). So our result reduces the necessary assumptions when implementing general multi-party computation protocols based on physical assumptions. Also, Kent [14] proposes to build commitments based on the fact that the speed of light is bounded. Although it is not clear whether his schemes are UC-secure (and in particular, how to model his physical assumptions in the UC framework), his ideas might lead to a UC-secure commitment scheme that then, using our result, gives general UC-secure multi-party computation based on the limitation of the speed of light.

Theoretical separation. Our result can also be seen from the purely theoretical point of view. It gives a separation between the quantum and the classical setting by showing that in the quantum setting, commitment is complete for general statistically secure multi-party computation, while in the classical world it is not. Such separations – even without practical applications – may increase our understanding of the relationship between the classical and the quantum setting and are therefore arguably interesting in their own right.

Long-term security. Müller-Quade and Unruh [17] introduce the concept of long-term UC-security. In a nutshell, long-term UC-security is a strengthening of computational UC-security that guarantees that a protocol stays secure even if the adversary gets unlimited computational power after the protocol execution. This captures the fact that, while we might confidently judge today's technology, we cannot easily make predictions about which computational problems will be hard in the future. Müller-Quade and Unruh show that (classically) long-term UC-secure commitment protocols exist given certain practical infrastructure assumptions, so-called signature cards. It is, however, likely that their results cannot be extended to achieve general multi-party computation. Our result, on the other hand, might allow to overcome this limitation: Assume that we show that the commitment protocol of Müller-Quade and Unruh is also secure in a quantum variant of long-term UC-security. Then we could compose that commitment protocol with the protocols presented here, leading to long-term UC-secure general multi-party protocols from signature cards.

1.2 Related Work

Security models. General quantum security models based on the stand-alone model have first been proposed by van de Graaf [22]. His model comes without a composition theorem. The notion has been refined by Wehner and Wullschleger [24] and by Fehr and Schaffner [11] who also prove sequential composition theorems. Quantum security models in the style of the UC model have been proposed by Ben-Or and Mayers [3] and by Unruh [20]. The original idea behind the UC framework in the classical setting was independently discovered by Canetti [7] and by Pfitzmann and Waidner [19] (the notion is called Reactive Simulatability in the latter paper).

Quantum protocols. The idea of using quantum communication for cryptographic purposes seems to originate from Wiesner [25]. The idea gained widespread recognition with the BB84 quantum key-exchange protocol by Bennett and Brassard [4]. A statistically hiding and binding commitment scheme was proposed by Brassard, Crépeau, Jozsa, and Langlois [6]. Unfortunately, the scheme was later found to be insecure; in fact, Mayers [16] showed that statistically hiding and binding quantum commitments are impossible without using additional assumptions. Kent [14] circumvents this impossibility result by proposing a statistically hiding and binding commitment scheme that is based on the limitation of the speed of light. Bennett, Brassard, Crépeau, and Skubiszewska [5] present a protocol for statistically secure oblivious transfer in the quantum setting. They prove their protocol secure under the assumption that the adversary cannot store qubits and measures each qubit individually. They also sketch an extension that uses a commitment scheme to make their OT protocol secure against adversaries that can store and compute on quantum states. The protocol analyzed in the present paper is, in its basic idea, that extension. Yao [26] gave a partial proof of the extended OT protocol. His proof, however, is incomplete and refers to a future complete paper which, to the best of our knowledge, never appeared. As far as we know, the first complete proof of a variant of that OT protocol has been given by Damgård, Fehr, Lunemann, Salvail, and Schaffner [9]; their protocol is secure in the stand-alone model. Hofheinz and Müller-Quade [12] conjectured that the extended OT protocol by Bennett et al. [5] is indeed UC-secure; in the present paper we prove this claim. Damgård, Fehr, Salvail, and Schaffner [10] have presented OT and commitment protocols which are statistically secure under the assumption that the adversary has a bounded quantum storage capacity. [1] (extended abstract only) give a protocol for performing quantum-UC multi-party computation given an honest majority. Their protocol even allows to compute functions which have quantum output.

Classical vs. quantum security. To the best of our knowledge, van de Graaf [22] was the first to notice that even statistically secure classical protocols are not necessarily secure in a quantum setting. The reason is that the powerful technique of rewinding the adversary is not available in the quantum setting. Watrous [23] showed that in particular cases, a technique similar to classical rewinding can be used. He uses this technique to construct quantum zero-knowledge proofs. No general technique relating classical and quantum security is known; to the best of our knowledge, our quantum lifting theorem is the first such result (although restricted to the statistical UC model).

Miscellaneous. Kilian [15] first noted that OT is complete for general multi-party computation. Ishai, Prabhakaran, and Sahai [13] prove that this also holds in the UC setting. Computationally secure UC commitment schemes have been presented by Canetti and Fischlin [8].

1.3 Preliminaries

General. A nonnegative function μ is called negligible if for all $c > 0$ and all sufficiently large k, $\mu(k) < k^{-c}$. A nonnegative function f is called overwhelming if $f \geq 1 - \mu$ for some negligible μ. Keywords in typewriter font (e.g., `environment`) are assumed to be fixed but arbitrary distinct non-empty words in $\{0,1\}^*$. $\varepsilon \in \{0,1\}^*$ denotes the empty word. Given a sequence $x = x_1, \dots, x_n$, and a set $I \subseteq \{1, \dots, n\}$, $x_{|I}$ denote the sequence x restricted to the indices $i \in I$.

Quantum systems. We can only give a terse overview over the formalism used in quantum computing. For a thorough introduction, we recommend the textbook by Nielsen and Chuang [18, Chap. 1–2]. A (pure) state in a quantum system is described by a vector $|\psi\rangle$ in some Hilbert space $\mathcal{H}$. In this work, we only use Hilbert spaces of the form $\mathcal{H} = \mathbb{C}^N$ for some countable set N, usually $N = \{0,1\}$ for qubits or $N = \{0,1\}^*$ for bitstrings. We always assume a designated orthonormal basis $\{|x\rangle : x \in N\}$ for each Hilbert space, called the computational basis. The basis states $|x\rangle$ represent classical states (i.e., states without superposition). Given several separate subsystems $\mathcal{H}_1 = \mathbb{C}^{N_1}, \dots, \mathcal{H}_n = \mathbb{C}^{N_n}$, we describe the joint system by the tensor product $\mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_n = \mathbb{C}^{N_1 \times \cdots \times N_n}$. We write $\langle \Psi|$ for the linear transformation mapping $|\Phi\rangle$ to the scalar product $\langle \Psi|\Phi\rangle$. Consequently, $|\Psi\rangle\langle\Psi|$ denotes the orthogonal projector on $|\Psi\rangle$. We set $|0\rangle_+ := |0\rangle$, $|1\rangle_+ := |1\rangle$, $|0\rangle_\times := \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and $|1\rangle_\times := \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. For $x \in \{0,1\}^n$ and $\theta \in \{+,\times\}^n$, we define $|x\rangle_\theta := |x_1\rangle_{\theta_1} \otimes \cdots \otimes |x_n\rangle_{\theta_n}$.

Mixed states. If a system is not in a single pure state, but instead is in the pure state $|\Psi_i\rangle \in \mathcal{H}$ with probability p_i (i.e., it is in a mixed state), we describe the system by a density operator $\rho = \sum_i p_i |\Psi_i\rangle\langle\Psi_i|$ over $\mathcal{H}$. This representation contains all physically observable information about the distribution of states, but some distributions are not distinguishable by any measurement and thus are represented by the same mixed state. The set of all density operators is the set of all positive[4] operators $\mathcal{H}$ with trace 1, and is denoted $\mathcal{P}(\mathcal{H})$. Composed systems

[4] We call an operator positive if it is Hermitean and has only nonnegative eigenvalues.

are descibed by operators in $\mathcal{P}(\mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_n)$. In the following, when speaking about (quantum) states, we always mean mixed states in the density operator representation. A mapping $\mathcal{E} : \mathcal{P}(\mathcal{H}_1) \to \mathcal{P}(\mathcal{H}_2)$ represents a physically possible operation (realizable by a sequence of unitary transformations, measurements, and initializations and removals of qubits) iff it is a completely positive trace preserving map.[5] We call such mappings superoperators. The superoperator $\mathcal{E}^m_{init}$ on $\mathcal{P}(\mathcal{H})$ with $\mathcal{H} := \mathbb{C}^{\{0,1\}^*}$ and $m \in \{0,1\}^*$ is defined by $\mathcal{E}^m_{init}(\rho) := |m\rangle\langle m|$ for all ρ.

Composed systems. Given a superoperator $\mathcal{E}$ on $\mathcal{P}(\mathcal{H}_1)$, the superoperator $\mathcal{E} \otimes id$ operates on $\mathcal{P}(\mathcal{H}_1 \otimes \mathcal{H}_2)$. Instead of saying "we apply $\mathcal{E} \otimes id$", we say "we apply $\mathcal{E}$ to $\mathcal{H}_1$". If we say "we initialize $\mathcal{H}$ with m", we mean "we apply $\mathcal{E}^m_{init}$ to $\mathcal{H}$". Given a state $\rho \in \mathcal{P}(\mathcal{H}_1 \otimes \mathcal{H}_2)$, let $\rho_x := (|x\rangle\langle x| \otimes id)\rho(|x\rangle\langle x| \otimes id)$. Then the outcome of measuring $\mathcal{H}_1$ in the computational basis is x with probability $\operatorname{tr}\rho_x$, and after measuring x, the quantum state is $\frac{\rho_x}{\operatorname{tr}\rho_x}$. Since we will only perform measurements in the computational basis in this work, we will omit the qualification "in the computational basis". The terminology in this paragraph generalizes to systems composed of more than two subsystems.

Classical states. Classical probability distributions $P : N \to [0,1]$ over a countable set N are represented by density operators $\rho \in \mathcal{P}(\mathbb{C}^N)$ with $\rho = \sum_{x \in N} P(x)|x\rangle\langle x|$ where $\{|x\rangle\}$ is the computational basis. We call a state classical if it is of this form. We thus have a canonical isomorphism between the classical states over $\mathbb{C}^N$ and the probability distributions over N. We call a superoperator $\mathcal{E} : \mathcal{P}(\mathbb{C}^{N_1}) \to \mathcal{P}(\mathbb{C}^{N_2})$ classical iff if there is a randomized function $F : N_1 \to N_2$ such that $\mathcal{E}(\rho) = \sum_{x \in N_1, y \in N_2} \Pr[F(x) = y] \cdot \langle x|\rho|x\rangle \cdot |y\rangle\langle y|$. Classical superoperators describe what can be realized with classical computations. An example of a classical superoperator on $\mathcal{P}(\mathbb{C}^N)$ is $\mathcal{E}_{class} : \rho \mapsto \sum_x \langle x|\rho|x\rangle \cdot |x\rangle\langle x|$. Intuitively, $\mathcal{E}_{class}$ measures ρ in the computational basis and then discards the outcome, thus removing all superpositions from ρ.

2 Quantum Universal Composability

We now present our quantum-UC-framework. The basic idea of our definition is the same as that underlying Canetti's UC-framework [7]. The main change is that we allow all machines to perform quantum computations and to send quantum states as messages. For a gentler introduction into the ideas and intuitions underlying the UC-framework, we refer to [7].

Machine model. A machine M is described by an identity id_M in $\{0,1\}^*$ and a sequence of superoperators $\mathcal{E}^{(k)}_M$ ($k \in \mathbb{N}$) on $\mathcal{H}^{state} \otimes \mathcal{H}^{class} \otimes \mathcal{H}^{quant}$ with $\mathcal{H}^{state}, \mathcal{H}^{class}, \mathcal{H}^{quant} := \mathbb{C}^{\{0,1\}^*}$ (the *state transition operators*). The index k in $\mathcal{E}^{(k)}_M$ denotes the security parameter. The Hilbert space $\mathcal{H}^{state}$ represents the state kept by the machine between invocations, and $\mathcal{H}^{class}$ and $\mathcal{H}^{quant}$ are used

[5] A map $\mathcal{E}$ is completely positive iff for all Hilbert spaces $\mathcal{H}'$, and all positive operators ρ on $\mathcal{H}_1 \otimes \mathcal{H}'$, $(\mathcal{E} \otimes id)(\rho)$ is positive.

both for incoming and outgoing messages. Any message consists of a classical part stored in $\mathcal{H}^{class}$ and a quantum part stored in $\mathcal{H}^{quant}$. If a machine id_{sender} wishes to send a message with classical part m and quantum part $|\Psi\rangle$ to a machine id_{rcpt}, the machine id_{sender} initializes $\mathcal{H}^{class}$ with $(id_{sender}, id_{rcpt}, m)$ and $\mathcal{H}^{quant}$ with $|\Psi\rangle$. (See the definition of the network execution below for details.) The separation of messages into a classical and a quantum part is for clarity only, all information could also be encoded directly in a single register. If a machine does not wish to send a message, it initializes $\mathcal{H}^{class}$ and $\mathcal{H}^{quant}$ with ε.

A network $\mathbf{N}$ is a set of machines with pairwise distinct identities containing a machine $\mathcal{Z}$ with $id_{\mathcal{Z}} = \texttt{environment}$. We write $ids_{\mathbf{N}}$ for the set of the identities of the machines in $\mathbf{N}$.

We call a machine M quantum-polynomial-time if there is a uniform[6] sequence of quantum circuits C_k such that for all k, the circuit C_k implements the superoperator $\mathcal{E}_M^{(k)}$.

Network execution. The state space $\mathcal{H}_{\mathbf{N}}$ of a network $\mathbf{N}$ is defined as $\mathcal{H}_{\mathbf{N}} := \mathcal{H}^{class} \otimes \mathcal{H}^{quant} \otimes \bigotimes_{id \in ids_{\mathbf{N}}} \mathcal{H}_{id}^{state}$ with $\mathcal{H}_{id}^{state}, \mathcal{H}^{class}, \mathcal{H}^{quant} := \mathbb{C}^{\{0,1\}^*}$. Here $\mathcal{H}_{id}^{state}$ represents the local state of the machine with identity id and $\mathcal{H}^{class}$ and $\mathcal{H}^{quant}$ represent the state spaces used for communication. ($\mathcal{H}^{class}$ and $\mathcal{H}^{quant}$ are shared between all machines. Since only one machine is active at a time, no conflicts occur.)

A step in the execution of $\mathbf{N}$ is defined by a superoperator $\mathcal{E} := \mathcal{E}_{\mathbf{N}}^{(k)}$ operating on $\mathcal{H}_{\mathbf{N}}$. This superoperator performs the following steps: First, $\mathcal{E}$ measures $\mathcal{H}^{class}$ in the computational basis and parses the outcome as $(id_{sender}, id_{rcpt}, m)$. Let M be the machine in $\mathbf{N}$ with identity id_{rcpt}. Then $\mathcal{E}$ applies $\mathcal{E}_M^{(k)}$ to $\mathcal{H}_{id_{rcpt}}^{state} \otimes \mathcal{H}^{class} \otimes \mathcal{H}^{quant}$. Then $\mathcal{E}$ measures $\mathcal{H}^{class}$ and parses the outcome as $(id'_{sender}, id'_{rcpt}, m')$. If the outcome could not be parsed, or if $id'_{sender} \neq id_{rcpt}$, initialize $\mathcal{H}^{class}$ with $(\varepsilon, \texttt{environment}, \varepsilon)$ and $\mathcal{H}^{quant}$ with ε. (This ensures that the environment is activated if a machine sends no or an ill-formed message.)

The output of the network $\mathbf{N}$ on input z and security parameter k is described by the following algorithm: Let $\rho \in \mathcal{P}(\mathcal{H}_{\mathbf{N}})$ be the state that is initialized to $(\varepsilon, \texttt{environment}, z)$ in $\mathcal{H}^{class}$, and to the empty word ε in all other registers. Then repeat the following indefinitely: Apply $\mathcal{E}_{\mathbf{N}}^{(k)}$ to ρ. Measure $\mathcal{H}^{class}$. If the outcome is of the form $(\texttt{environment}, \varepsilon, out)$, return out and terminate. Otherwise, continue the loop. The probability distribution of the return value out is denoted by $\mathrm{Exec}_{\mathbf{N}}(k, z)$.

Corruptions. To model corruptions, we introduce *corruption parties*, special machines that follow the instructions given by the adversary. When invoked, the corruption party P_{id}^C with identity id measures $\mathcal{H}^{class}$ and parses the outcome as $(id_{sender}, id_{rcpt}, m)$. If $id_{sender} = \texttt{adversary}$, $\mathcal{H}^{class}$ is initialized with m. (In this case, m specifies both the message and the sender/recipient. Thus the

[6] A sequence of circuits C_k is uniform if a deterministic Turing machine can output the description of C_k in time polynomial in k.

adversary can instruct a corruption party to send to arbitrary recipients.) Otherwise, $\mathcal{H}^{class}$ is initialized with $(id, \texttt{adversary}, (id_{sender}, id_{rcpt}, m))$. (The message is forwarded to the adversary.) Note that, since P^C_{id} does not touch the $\mathcal{H}^{quant}$, the quantum part of the message is forwarded. Given a network $\mathbf{N}$, and a set of identities C, we write $\mathbf{N}^C$ for the set resulting from replacing each machine $M \in \mathbf{N}$ with identity $id \in C$ by P^C_{id}.

Security model. A protocol π is a set of machines with $\texttt{environment}$, $\texttt{adversary} \notin \mathit{ids}(\pi)$. We assume a set of identities $\mathit{parties}_\pi \subseteq \mathit{ids}(\pi)$ to be associated with π. $\mathit{parties}_\pi$ denotes which of the machines in the protocol are actually protocol parties (as opposed to incorruptible entities such as ideal functionalities).

An *environment* is a machine with identity $\texttt{environment}$, an *adversary* or a *simulator* is a machine with identity $\texttt{adversary}$ (there is no formal distinction between adversaries and simulators, the terms refer to different intended roles of a machine). We call two networks $\mathbf{N}, \mathbf{N}'$ *indistinguishable* if there is a negligible function μ such that for all $z \in \{0,1\}^*$ and $k \in \mathbb{N}$, $|\Pr[\mathrm{Exec}_{\mathbf{N}}(k,z) = 1] - \Pr[\mathrm{Exec}_{\mathbf{N}'}(k,z) = 1]| \leq \mu(k)$. We speak of *perfect indistinguishability* if $\mu = 0$.

Definition 2 (Statistical quantum-UC-security). *Let protocols π and ρ be given. We say π* statistically quantum-UC-emulates *ρ iff for every set $C \subseteq \mathit{parties}_\pi$ and for every adversary* Adv *there is a simulator* Sim *such that for every environment $\mathcal{Z}$, the networks $\pi^C \cup \{\mathrm{Adv}, \mathcal{Z}\}$ (called the real model) and $\rho^C \cup \{\mathrm{Sim}, \mathcal{Z}\}$ (called the ideal model) are indistinguishable. We furthermore require that if* Adv *is quantum-polynomial-time, so is* Sim*.*

Definition 3 (Computational quantum-UC-security). *Let protocols π and ρ be given. We say π* computationally quantum-UC-emulates *ρ iff for every set $C \subseteq \mathit{parties}_\pi$ and for every quantum-polynomial-time adversary* Adv *there is a quantum-polynomial-time simulator* Sim *such that for every quantum-polynomial-time environment $\mathcal{Z}$, the networks $\pi^C \cup \{\mathrm{Adv}, \mathcal{Z}\}$ and $\rho^C \cup \{\mathrm{Sim}, \mathcal{Z}\}$ are indistinguishable.*

Note that although $\mathrm{Exec}_{\pi^C \cup \{\mathrm{Adv}, \mathcal{Z}\}}(k,z)$ may return arbitrary bitstrings, we only compare whether the return value of $\mathcal{Z}$ is 1 or not. This effectively restricts $\mathcal{Z}$ to returning a single bit. This can be done without loss of generality (see [7] for a discussion of this issue; their arguments also apply to the quantum case) and simplifies the definition.

In our framework, any communication between two parties is perfectly secure since the network model guarantees that they are delivered to the right party and not leaked to the adversary. To model a protocol with insecure channels instead, one would explicitly instruct the protocol parties to send all messages through the adversary. Authenticated channels can be realized by introducing an ideal functionality (see the next section) that realizes an authenticated channel. For simplicity, we only consider protocols with secure channels in this work.

Ideal functionalities. In most cases, the behavior of the ideal model is described by a single machine $\mathcal{F}$, the so-called ideal functionality. We can think of this functionality as a trusted third party that perfectly implements the desired protocol behavior. For example, the functionality $\mathcal{F}_{\mathrm{OT}}$ for oblivious transfer would take as input from Alice two bitstrings m_0, m_1, and from Bob a bit c, and send to Bob the bitstring m_c. Obviously, such a functionality constitutes a secure oblivious transfer. We can thus define a protocol π to be a secure OT protocol if π quantum-UC-emulates $\mathcal{F}_{\mathrm{OT}}$ where $\mathcal{F}_{\mathrm{OT}}$ denotes the protocol consisting only of one machine, the functionality $\mathcal{F}_{\mathrm{OT}}$ itself. There is, however, one technical difficulty here. In the real protocol π, the bitstring m_c is sent to the environment $\mathcal{Z}$ by Bob, while in the ideal model, m_c is sent by the functionality. Since every message is tagged with the sender of that message, $\mathcal{Z}$ can distinguish between the real and the ideal model merely by looking at the sender of m_c. To solve this issue, we need to ensure that $\mathcal{F}$ sends the message m_c in the name of Bob (and for analogous reasons, that $\mathcal{F}$ receives messages sent by $\mathcal{Z}$ to Alice or Bob). To achieve this, we use so-called dummy-parties [7] in the ideal model. These are parties with the identities of Alice and Bob that just forward messages between the functionality and the environment.

Definition 4 (Dummy-party). *Let a machine P and a functionality $\mathcal{F}$ be given. The dummy-party $\tilde{P}$ for P and $\mathcal{F}$ is a machine that has the same identity as P and has the following state transition operator: Let $id_{\mathcal{F}}$ be the identity of $\mathcal{F}$. When activated, measure $\mathcal{H}^{class}$. If the outcome of the measurement is of the form $(\texttt{environment}, id_P, m)$, initialize $\mathcal{H}^{class}$ with $(id_P, id_{\mathcal{F}}, m)$. If the outcome is of the form $(id_{\mathcal{F}}, id_P, m)$, initialize $\mathcal{H}^{class}$ with $(id_P, \texttt{environment}, m)$. In all cases, the quantum communication register is not modified (i.e., the message in that register is forwarded).*

Note the strong analogy to the corruption parties (page 494).

Thus, if we write π quantum-UC-emulates $\mathcal{F}$, we mean that π quantum-UC-emulates $\rho_{\mathcal{F}}$ where $\rho_{\mathcal{F}}$ consists of the functionality $\mathcal{F}$ and the dummy-parties corresponding to the parties in π. More precisely:

Definition 5. *Let π be a protocol and $\mathcal{F}$ be a functionality. We say that π statistically/computationally quantum-UC-emulates $\mathcal{F}$ if π statistically/computationally quantum-UC-emulates $\rho_{\mathcal{F}}$ where $\rho_{\mathcal{F}} := \{\tilde{P} : P \in parties_\pi\} \cup \{\mathcal{F}\}$.*

For more discussion of dummy-parties and functionalities, see [7].

Using the concept of an ideal functionality, we can specify a range of protocol tasks by simply defining the corresponding functionality. Below, we give the definitions of various functionalities. All these functionalities are classical, we therefore do not explicitly describe when the registers $\mathcal{H}^{class}$ and $\mathcal{H}^{quant}$ are measured/initialized but instead describe the functionality in terms of the messages sent and received.

Definition 6 (Commitment). *Let A and B be two parties. The functionality $\mathcal{F}_{\mathrm{COM}}^{B\to A,\ell}$ behaves as follows: Upon (the first) input $(\texttt{commit}, x)$ with $x \in \{0,1\}^{\ell(k)}$ from B, send $\texttt{committed}$ to A. Upon input $\texttt{open}$ from B send $(\texttt{open}, x)$ to A. All communication/input/output is classical. We call B the sender and A the recipient.*

Definition 7 (Oblivious transfer (OT)). *Let A and B be two parties. The functionality $\mathcal{F}_{\mathrm{OT}}^{A\to B,\ell}$ behaves as follows: When receiving input (s_0, s_1) from A with $s_0, s_1 \in \{0,1\}^{\ell(k)}$ and $c \in \{0,1\}$ from B, send $s := s_c$ to B. All communication/input/output is classical. We call A the sender and B the recipient.*[7]

Definition 8 (Randomized oblivious transfer (ROT)). *Let A and B be two parties. The functionality $\mathcal{F}_{\mathrm{ROT}}^{A\to B,\ell}$ behaves as follows: If A is uncorrupted, when receiving input $c \in \{0,1\}$ from B, choose $s_0, s_1 \in \{0,1\}^{\ell(k)}$ uniformly and send (s_0, s_1) to A and $s := s_c$ to B. If A is corrupted, when receiving input (s_0, s_1) from A with $s_0, s_1 \in \{0,1\}^{\ell(k)}$ and $c \in \{0,1\}$ from B, send $s := s_c$ to B. All communication/input/output is classical.*

Dummy-adversary. In the definition of UC-security, we have three entities interacting with the protocol: the adversary, the simulator, and the environment. Both the adversary and the environment are all-quantified, hence we would expect that they do, in some sense, work together. This intuition is backed by the following fact which was first noted by Canetti [7]: Without loss of generality, we can assume an adversary that is completely controlled by the environment. This so-called dummy-adversary only forwards messages between the environment and the protocol. The actual attack is then executed by the environment.

Definition 9 (Dummy-adversary Adv_{dummy}). *When activated, the dummy-adversary Adv_{dummy} measures $\mathcal{H}^{class}$; call the outcome m. If m is of the form $(\mathtt{environment}, \mathtt{adversary}, m')$, initialize $\mathcal{H}^{class}$ with m'. Otherwise initialize $\mathcal{H}^{class}$ with $(\mathtt{adversary}, \mathtt{environment}, m)$. In all cases, the quantum communication register is not modified (i.e., the message in that register is forwarded).*

Note the strong analogy to the dummy-parties (Definition 4) and the corruption parties (page 494).

Lemma 10 (Completeness of the dummy-adversary). *Assume that π quantum-UC-emulates ρ with respect to the dummy-adversary (i.e., instead of quantifying over all adversaries Adv, we fix $\mathrm{Adv} := \mathrm{Adv}_{dummy}$). Then π quantum-UC-emulates ρ. This holds both for statistical and computational quantum-UC-security.*

The proof of Lemma 10 is very similar to that given in [7] and given in the full version [21].

Universal composition. For some protocol σ, and some protocol π, by σ^π we denote the protocol where σ invokes (up to polynomially many) instances of π. That is, in σ^π the machines from σ and from π run together in one network, and the machines from σ access the inputs and outputs of π. (That is, σ plays the role of the environment from the point of view of π. In particular, $\mathcal{Z}$ then

[7] We used A as the sender in the description of the OT functionality, and as the recipient in the description of the commitment functionality. We do so to simplify notation later; our protocol for OT from A to B will use a commitment from B to A.

talks only to σ and not to the subprotocol π directly.) A typical situation would be that $\sigma^{\mathcal{F}}$ is some protocol that makes use of some ideal functionality $\mathcal{F}$, say a commitment functionality, and then σ^{π} would be the protocol resulting from implementing that functionality with some protocol π, say a commitment protocol. (We say that $\sigma^{\mathcal{F}}$ is a protocol in the $\mathcal{F}$-hybrid model.) One would hope that such an implementation results in a secure protocol σ^{π}. That is, we hope that if π quantum-UC-emulates $\mathcal{F}$ and $\sigma^{\mathcal{F}}$ quantum-UC-emulates $\mathcal{G}$, then σ^{π} quantum-UC-emulates $\mathcal{G}$. Fortunately, this is the case:

Theorem 11 (Universal Composition Theorem). *Let π, ρ, and σ be quantum-polynomial-time protocols. Assume that π quantum-UC-emulates ρ. Then σ^{π} quantum-UC-emulates σ^{ρ}. This holds both for statistical and computational quantum-UC-security.*

If we additionally have that σ quantum-UC-emulates $\mathcal{G}$, from the transitivity of quantum-UC-emulation (shown in the full version [21]), it immediately follows that σ^{π} quantum-UC-emulates $\mathcal{G}$.

The proof of Theorem 11 is very similar to that given in [7] and given in the full version [21].

3 Relating Classical and Quantum-UC

We call a machine classical if its state transition operator is classical. A protocol is classical if all its machines are classical.

Using this definition we can reformulate the definition of statistical classical UC in our framework.

Definition 12 (Statistical classical-UC-security). *Let protocols π and ρ be given. We say π* statistically classical-UC-emulates *ρ iff for every set $C \subseteq$ parties$_{\pi}$ and for every classical adversary* Adv *there is a classical simulator* Sim *such that for every classical environment $\mathcal{Z}$, $\pi^{C} \cup \{\mathrm{Adv}, \mathcal{Z}\}$ and $\rho^{C} \cup \{\mathrm{Sim}, \mathcal{Z}\}$ are indistinguishable. We furthermore require that if* Adv *is probabilistic-polynomial-time, so is* Sim*.*

Note that classical statistical UC is essentially the same as the notion of statistical UC-security defined by Canetti [7]. Thus, known results for statistical UC-security carry over to the setting of Definition 12.

The next theorem guarantees that if a classical protocol is statistically classical UC-secure, then it is also statistically quantum-UC-secure. This allows, e.g., to first prove the security of a protocol in the (usually much simpler) classical setting, and then to compose it with quantum protocols using the universal composition theorem (Theorem 11).

Theorem 13 (Quantum lifting theorem). *Let π and ρ be classical protocols. Assume that π statistically classical-UC-emulates ρ. Then π statistically quantum-UC-emulates ρ.*

Proof. Given a machine M, let $\mathcal{C}(M)$ denote the machine which behaves like M, but measures incoming messages in the computational basis before processing them, and measures outgoing messages in the computational basis. More precisely, the superoperator $\mathcal{E}^{(k)}_{\mathcal{C}(M)}$ first invokes $\mathcal{E}_{class}$ on $\mathcal{H}^{class} \otimes \mathcal{H}^{quant}$, then invokes $\mathcal{E}^{(k)}_{M}$ on $\mathcal{H}^{state} \otimes \mathcal{H}^{class} \otimes \mathcal{H}^{quant}$, and then again invokes $\mathcal{E}_{class}$ on $\mathcal{H}^{class} \otimes \mathcal{H}^{quant}$. Since it is possible to simulate quantum Turing machines on classical Turing machines (with an exponential overhead), for every machine M, there exists a classical machine M' such that $\mathcal{C}(M)$ and M' are perfectly indistinguishable.[8]

We define the classical dummy-adversary $\mathrm{Adv}^{class}_{dummy}$ to be the classical machine that is defined like Adv_{dummy} (Definition 9), except that in each invocation, it first measures $\mathcal{H}^{class}$, $\mathcal{H}^{quant}$, and $\mathcal{H}^{state}$ in the computational basis (i.e., it applies $\mathcal{E}_{class}$ to $\mathcal{H}^{state} \otimes \mathcal{H}^{class} \otimes \mathcal{H}^{quant}$) and then proceeds as does Adv_{dummy}. Note that $\mathrm{Adv}^{class}_{dummy}$ is probabilistic-polynomial-time.

By Lemma 10, we only need to show that for any set C of corrupted parties, there exists a quantum-polynomial-time machine Sim such that for every machine $\mathcal{Z}$ the real model $\pi^C \cup \{\mathcal{Z}, \mathrm{Adv}_{dummy}\}$ and the ideal model $\rho^C \cup \{\mathcal{Z}, \mathrm{Sim}\}$ are indistinguishable.

The protocol π is classical, thus π^C is classical, too, and thus all messages forwarded by Adv_{dummy} from π^C to $\mathcal{Z}$ have been measured in the computational basis by π^C, and all messages forwarded by Adv_{dummy} from $\mathcal{Z}$ to π^C will be measured by π^C before being used. Thus, if Adv would additionally measure all messages it forwards in the computational basis, the view of $\mathcal{Z}$ would not be modified. More formally, $\pi^C \cup \{\mathcal{Z}, \mathrm{Adv}_{dummy}\}$ and $\pi^C \cup \{\mathcal{Z}, \mathrm{Adv}^{class}_{dummy}\}$ are perfectly indistinguishable. Furthermore, since both π^C and $\mathrm{Adv}^{class}_{dummy}$ measure all messages upon sending and receiving, $\pi^C \cup \{\mathcal{Z}, \mathrm{Adv}^{class}_{dummy}\}$ and $\pi^C \cup \{\mathcal{C}(\mathcal{Z}), \mathrm{Adv}^{class}_{dummy}\}$ are perfectly indistinguishable. Since it is possible to simulate quantum machines on classical machines (with an exponential overhead), there exists a classical machine $\mathcal{Z}'$ that is perfectly indistinguishable from $\mathcal{C}(\mathcal{Z})$. Then $\pi^C \cup \{\mathcal{C}(\mathcal{Z}), \mathrm{Adv}^{class}_{dummy}\}$ and $\pi^C \cup \{\mathcal{Z}', \mathrm{Adv}^{class}_{dummy}\}$ are perfectly indistinguishable. Since $\mathrm{Adv}^{class}_{dummy}$ and $\mathcal{Z}'$ are classical and $\mathrm{Adv}^{class}_{dummy}$ is polynomial-time, there exists a classical probabilistic-polynomial-time simulator Sim (whose construction is independent of $\mathcal{Z}'$) such that $\pi^C \cup \{\mathcal{Z}', \mathrm{Adv}^{class}_{dummy}\}$ and $\rho^C \cup \{\mathcal{Z}', \mathrm{Sim}\}$ are indistinguishable.

Then $\rho^C \cup \{\mathcal{Z}', \mathrm{Sim}\}$ and $\rho^C \cup \{\mathcal{C}(\mathcal{Z}), \mathrm{Sim}\}$ are perfectly indistinguishable by construction of $\mathcal{Z}'$. And since both ρ^C and Sim measure all messages they send and receive, $\rho^C \cup \{\mathcal{C}(\mathcal{Z}), \mathrm{Sim}\}$ and $\rho^C \cup \{\mathcal{Z}, \mathrm{Sim}\}$ are perfectly indistinguishable.

Summarizing, we have that $\pi^C \cup \{\mathcal{Z}, \mathrm{Adv}_{dummy}\}$ and $\rho^C \cup \{\mathcal{Z}, \mathrm{Sim}\}$ are indistinguishable for all quantum-polynomial-time environments $\mathcal{Z}$. Furthermore, Sim is classical probabilistic-polynomial-time and hence quantum-polynomial-time and its construction does not depend on the choice of $\mathcal{Z}$. Thus π statistically quantum-UC-emulates ρ. □

[8] More precisely, for any set of machines N, the networks $N \cup \{M\}$ and $N \cup \{\mathcal{C}(M)\}$ are perfectly indistinguishable.

Parameters: Integers n, $m > n$, ℓ, a family $\mathbf{F}$ of universal hash functions.
Parties: The sender Alice and the recipient Bob.
Inputs: Alice gets no input, Bob gets a bit c.

1. Alice chooses $\tilde{x}^A \in \{0,1\}^m$ and $\tilde{\theta}^A \in \{+,\times\}^m$ and sends $|\tilde{x}^A\rangle_{\tilde{\theta}^A}$ to Bob.
2. Bob receives the state $|\Psi\rangle$ sent by the sender. Then Bob chooses $\tilde{\theta}^B \in \{+,\times\}^m$ and measures the qubits of $|\Psi\rangle$ in the bases $\tilde{\theta}^B$. Call the result $\tilde{x}^B$.
3. For each i, Bob commits to $\tilde{\theta}_i^B$ and $\tilde{x}_i^B$ using one instance of $\mathcal{F}_{\mathrm{COM}}^{B\to A,1}$ each.
4. Alice chooses a set $T \subseteq \{1,\dots,m\}$ of size $m-n$ and sends T to Bob.
5. Bob opens the commitments of $\tilde{\theta}_i^B$ and $\tilde{x}_i^B$ for all $i \in T$.
6. Alice checks $\tilde{x}_i^A = \tilde{x}_i^B$ for all i with $i \in T$ and $\tilde{\theta}_i^A = \tilde{\theta}_i^B$. If this test fails, Alice aborts.
7. Let x^A be the n-bit string resulting from removing the bits at positions $i \in T$ from $\tilde{x}^A$. Define θ^A, x^B, and θ^B analogously.
8. Alice sends θ^A to Bob.
9. Bob sets $I_c := \{i : \theta_i^A = \theta_i^B\}$ and $I_{1-c} := \{i : \theta_i^A \neq \theta_i^B\}$. Then Bob sends (I_0, I_1) to Alice.
10. Alice chooses $s_0, s_1 \in \{0,1\}^{\ell(k)}$ and $f_0, f_1 \in \mathbf{F}$, output (s_0, s_1), and computes $m_j := s_j \oplus f_j(x^A|_{I_j})$ for $j = 0, 1$. Then Alice sends f_0, f_1, m_0, m_1 to Bob.
11. Bob outputs $s := m_c \oplus f_c(x^B|_{I_c})$.

Fig. 1. Protocol π_{QROT} for randomized oblivious transfer

4 Oblivious Transfer

Definition 14 (OT protocols). *The protocol π_{QROT} is defined in Figure 1. Fix a commitment scheme* com. *The protocol $\pi_{\mathrm{QROT}}^{\mathrm{com}}$ is defined like π_{QROT}, but instead of using the functionality $\mathcal{F}_{\mathrm{COM}}$, the commitment scheme* com *is used. The protocol π_{QOT} is defined like π_{QROT}, with the following modifications: Alice takes as input two $\ell(k)$-bit strings v_0, v_1. In Step 10, Alice additionally sends t_0, t_1 with $t_i := s_i \oplus v_i$. Bob outputs $s \oplus t_c$ instead of s in Step 11.*

We first analyze π_{QROT} and will then deduce the security of π_{QOT} from that of π_{QROT}.

4.1 Corrupted Alice

Lemma 15. *The protocol π_{QROT} statistically quantum-UC-emulates $\mathcal{F}_{\mathrm{ROT}}^{A\to B,\ell}$ in the case of corrupted Alice.*

Proof. First, we describe the structure of the real and ideal model in the case that the party A (Alice) is corrupted:

In the real model, we have the environment $\mathcal{Z}$, the adversary Adv, the corruption party A^C, the honest party B (Bob), and the $2m$ instances of the commitment functionality $\mathcal{F}_{\mathrm{COM}}$. The adversary controls the corruption party A^C, so effectively he controls the communication with Bob and the inputs of $\mathcal{F}_{\mathrm{COM}}$. Bob's input (a choice bit c) is chosen by the environment, and the environment also gets Bob's output (a bitstring $s \in \{0,1\}^\ell$). See Figure 2(a).

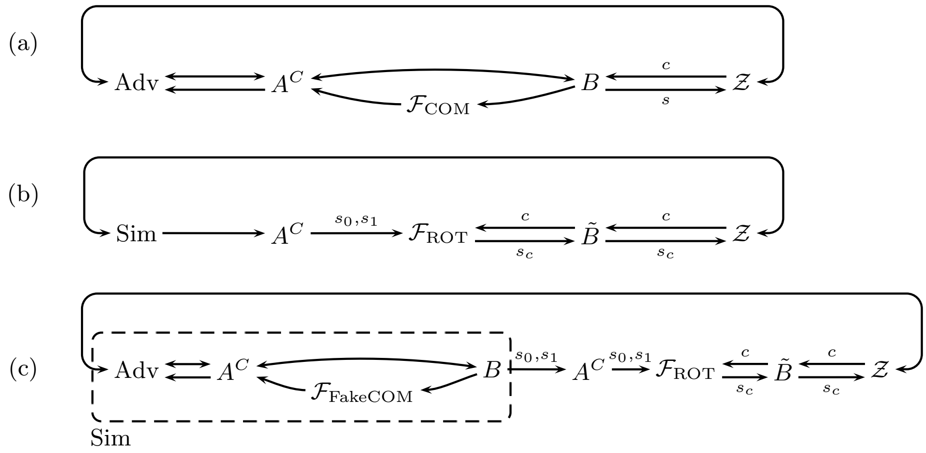


Fig. 2. Networks occurring in the proof of Lemma 15. The dashed box represents the machine Sim that internally simulates Adv, A^C, $\mathcal{F}_{\mathrm{FakeCOM}}$ and B.

In the ideal model, we have the environment $\mathcal{Z}$, the simulator Sim (to be defined below), the corruption party A^C, the dummy-party $\tilde{B}$, and the randomized OT functionality $\mathcal{F}_{\mathrm{ROT}}$. The simulator Sim controls the corruption party A^C and hence effectively chooses the inputs s_0, s_1 of $\mathcal{F}_{\mathrm{ROT}}$.[9] The input c of $\mathcal{F}_{\mathrm{ROT}}$ is chosen by the dummy-party $\tilde{B}$ and thus effectively by the environment $\mathcal{Z}$. The output $s := s_c$ of $\mathcal{F}_{\mathrm{ROT}}$ is given to the dummy-party $\tilde{B}$ and thus effectively to the environment $\mathcal{Z}$. See Figure 2(b).

To show Lemma 15, we need to find a simulator Sim such that, for any environment $\mathcal{Z}$, the real model and the ideal model are indistinguishable. To do so, we start with the real model, and change the machines in the real model step-by-step until we end up with the ideal model containing a suitable simulator Sim (which we define below in the description of Game 6). In each step, we show that network before and after the step are perfectly indistinguishable.

Game 1. We replace $\mathcal{F}_{\mathrm{COM}}$ by a commitment functionality $\mathcal{F}_{\mathrm{FakeCOM}}$ in which Bob (the sender) can cheat. That is, in the commit phase, $\mathcal{F}_{\mathrm{FakeCOM}}$ expects a message `commit` from B (instead of $(\texttt{commit}, x)$), and in the open phase, $\mathcal{F}_{\mathrm{FakeCOM}}$ expects a message $(\texttt{open}, x)$ (instead of `open`) and then sends $(\texttt{open}, x)$ to Alice. We also change Bob's implementation accordingly, i.e., when Bob should commit to a bit b, he stores that bit b and gives it to $\mathcal{F}_{\mathrm{FakeCOM}}$ when opening the commitment. Obviously, this change leads to a perfectly indistinguishable network (since Bob still opens the commitment in the same way).

Game 2. Since Bob uses $\mathcal{F}_{\mathrm{FakeCOM}}$ instead of $\mathcal{F}_{\mathrm{COM}}$, he does not use the outcomes $\tilde{x}_i^B$ of his measurements before Step 5 (for $i \in T$) or Step 11 (for $i \notin T$) of the protocol. Thus, we modify Bob so that he performs the measurements

[9] Remember that, if Alice is corrupted, $\mathcal{F}_{\mathrm{ROT}}$ behaves like $\mathcal{F}_{\mathrm{OT}}$ and takes inputs s_0, s_1 from Alice.

with outcomes $\tilde{x}_i^B$ $(i \in T)$ in Step 5 (in particular, after learning T), and the measurements with outcomes x_i^B in Step 11. Delaying the measurements leads to a perfectly indistinguishable network.

Game 3. The bits x_i^B with $i \in I_{1-c}$ are never used by Bob. Thus we can modify Bob to use the bases θ_i^A instead of θ_i^B for these bits without changing the output of $\mathcal{Z}$. Furthermore, since $\theta_i^A = \theta_i^B$ for $i \in I_c$, we can modify Bob to also use the bases θ_i^A instead of θ_i^B when measuring x_i^B with $i \in I_c$. Summarizing, we modify Bob to use θ^A instead of θ^B, and we get a perfectly indistinguishable network.

Game 4. The bases θ^B are chosen randomly by Bob, and they are only used to compute the sets I_0 and I_1. We change Bob to instead pick (I_0, I_1) as a random partition of $\{1, \dots, n\}$. Since this leads to the same distribution of (I_0, I_1) and since θ^B is not used elsewhere, this leads to a perfectly indistinguishable network.

Game 5. In Step 11, we change Bob to compute $s_i := m_i \oplus f_i(x^B|_{I_i})$ for $i = 0, 1$ and to output $s := s_c$. This leads to the same value of s as the original computation $s := m_c \oplus f_c(x^B|_{I_c})$, hence the resulting network is perfectly indistinguishable from the previous one. Note that now, Bob only uses the choice bit c to pick which of the two values s_0, s_1 to output.

Game 6. We now construct a machine Sim that internally simulates the machines Adv, A^C, $\mathcal{F}_{\text{FakeCOM}}$, and Bob. We let Sim run with an (external) corruption party A^C, and when (the simulated) Bob computes s_0, s_1 in Step 11, Sim instructs the (external) corruption party A^C to input s_0, s_1 into $\mathcal{F}_{\text{ROT}}$ (instead of letting Bob output $s = s_c$). Then $\mathcal{F}_{\text{ROT}}$ will, given input c from the dummy-party $\tilde{B}$, output s_c to the dummy-party $\tilde{B}$. The dummy-party $\tilde{B}$ then forwards s_c to the environment $\mathcal{Z}$. See Figure 2(c). The only difference with respect to the previous network (besides a regrouping of machines) is that now s_c is computed by $\mathcal{F}_{\text{ROT}}$ from s_0, s_1. However, $\mathcal{F}_{\text{ROT}}$ computes s_c in the same way as Bob would have done. Thus, the resulting network is perfectly indistinguishable from the previous one.

Since the network from Game 6 (Figure 2(c)) is identical to the ideal model (Figure 2(b)), and since the real model is perfectly indistinguishable from the network from Game 6, we have that the real and the ideal network are perfectly indistinguishable.

Furthermore, Sim is quantum-polynomial-time if Adv is, and the construction of Sim does not depend on the choice of the environment $\mathcal{Z}$. Thus the protocol π_{QROT} statistically quantum-UC-emulates $\mathcal{F}_{\text{ROT}}^{A \to B, \ell}$ in the case of corrupted Alice. □

Theorem 16. *Fix constants $0 < \alpha < 1$ and $0 < \lambda < \frac{1}{4}$. Let $m := \lceil n/(1-\alpha) \rceil$ and $\ell := \lfloor \lambda n \rfloor$ and assume that n grows at least linearly in the security parameter. Then the protocol π_{QROT} statistically quantum-UC-emulates $\mathcal{F}_{\text{ROT}}^{A \to B, \ell}$.*

For the case of corrupted Alice, this is shown in Lemma 15. The cases where both parties are honest or both parties are corrupted are trivial. Thus for Theorem 16 we are left to analyze the case where Bob is corrupted. This case needs a considerably more involved analysis than the case of corrupted Alice because we have to consider the fact that Bob may succeed in Step 6 of π_{QROT} but still have a certain amount of information about the bits $x^A|_{I_{1-c}}$. A very similar analysis has already been performed by Damgård, Fehr, Lunemann, Salvail, and Schaffner [9] in the so-called stand-alone model. Fortunately, we do not need to redo their analysis; it turns out that – although the stand-alone model is weaker than the quantum-UC-model – the particular simulator constructed by Damgård et al. is already strong enough to be used as a simulator in the quantum-UC-model. Thus we can reuse the result of Damgård et al. in our setting and get Theorem 16 without re-analyzing π_{QROT}.[10]

The full proof of Theorem 16 is given in the full version [21].

Theorem 17. *Let $0 < \alpha < 1$ and $0 < \lambda < \frac{1}{4}$ be constants. Assume $m = \lceil n/(1-\alpha)\rceil$ and $\ell = \lfloor \lambda n \rfloor$ and that n grows at least linearly in the security parameter. Then the protocol π_{QOT} (Def. 14) statistically quantum-UC-emulates $\mathcal{F}_{\mathrm{OT}}^{A\to B,\ell}$.*

Proof. Consider the following protocol π'_{QOT} in the $\mathcal{F}_{\mathrm{ROT}}$-hybrid model. Given inputs $v_0, v_1 \in \{0,1\}^{\ell(k)}$ for Alice and a bit c for Bob, Bob invokes $\mathcal{F}_{\mathrm{ROT}}$ with input c. Then Alice gets random $s_0, s_1 \in \{0,1\}^{\ell(k)}$, and Bob gets $s = s_c$. Then Alice sends t_0, t_1 with $t_i := v_i \oplus s_i$ to Bob. And Bob outputs $s \oplus t_c$. It is easy to see that π'_{QOT} statistically classical-UC-emulates $\mathcal{F}_{\mathrm{OT}}$. Hence, by the quantum lifting theorem (Theorem 13), π'_{QOT} statistically quantum-UC-emulates $\mathcal{F}_{\mathrm{OT}}$. Note that the protocol π_{QOT} is the protocol resulting from replacing, in π'_{QOT}, calls to $\mathcal{F}_{\mathrm{ROT}}$ by calls to the subprotocol π_{QROT}. Furthermore, π_{QROT} statistically quantum-UC-emulates $\mathcal{F}_{\mathrm{ROT}}$ by Theorem 16. Hence, by the composition theorem (Theorem 11), π_{QOT} statistically quantum-UC-emulates $\mathcal{F}_{\mathrm{OT}}$. □

5 Multi-party Computation

Theorem 18. *Let $\mathcal{F}$ be a classical probabilistic-polynomial-time functionality.*[11] *Then there exists a protocol π in the $\mathcal{F}_{\mathrm{COM}}$-hybrid model that statistically quantum-UC-emulates $\mathcal{F}$. (Assuming the number of protocol parties does not depend on the security parameter.)*

Proof. Ishai, Prabhakaran, and Sahai [13] prove the existence of a protocol $\rho^{\mathcal{F}_{\mathrm{OT}}}$ in the $\mathcal{F}_{\mathrm{OT}}$-hybrid model that statistically classical-UC-emulates $\mathcal{F}$ (assuming

[10] One major difference between the UC-model and the stand-alone model is that in the first, the honest parties' inputs may depend on messages the adversary intercepts during the protocol run. A simulator constructed for the stand-alone model usually is not able to cope with such dependencies. Thus, it turns out to be important that we first considered the randomized OT protocol π_{QROT} and not immediately the OT protocol π_{QOT}. In π_{QROT}, Alice gets no input, and in particular her inputs may not depend on messages intercepted by the adversary.

[11] Subject to certain technical restrictions stemming from the proof by Ishai et al. [13]: Whenever the functionality gets an input, the adversary is informed about the length of that input. Whenever the functionality makes an output, the adversary is informed about the length of that output and may decide when this output is to be scheduled.

a constant number of parties). By the quantum lifting theorem (Theorem 13), $\rho^{\mathcal{F}_{\mathrm{OT}}}$ statistically quantum-UC-emulates $\mathcal{F}$. By Theorem 17, π_{QOT} statistically quantum-UC-emulates $\mathcal{F}_{\mathrm{OT}}$. Let $\pi := \rho^{\pi_{\mathrm{QOT}}}$ be the result of replacing invocations to $\mathcal{F}_{\mathrm{OT}}$ in $\rho^{\mathcal{F}_{\mathrm{OT}}}$ by invocations of the subprotocol π_{QOT} (as described before Theorem 11). Then by the universal composition theorem (Theorem 11), π statistically quantum-UC-emulates $\rho^{\mathcal{F}_{\mathrm{OT}}}$. Using the fact that quantum-UC-emulation is transitive (shown in the full version [21]), it follows that π statistically quantum-UC-emulates $\mathcal{F}$. □

We proceed to show that the result from Theorem 18 is possible only in the quantum setting. That is, we show that there is a natural functionality that cannot be statistically classical-UC-emulated in the commitment-hybrid model.

Definition 19 (AND). *The functionality $\mathcal{F}_{\mathrm{AND}}$ expects an input $a \in \{0,1\}$ from Alice and $b \in \{0,1\}$ from Bob. Then it sends $a \cdot b$ to Alice and Bob.*

Theorem 20 (Impossibility of classical multi-party computation). *There is no classical probabilistic-polynomial-time protocol π in the $\mathcal{F}_{\mathrm{COM}}$-hybrid model such that π statistically classical-UC-emulates $\mathcal{F}_{\mathrm{AND}}$.*

The proof is given in the full version [21].

Acknowledgements. I thank Jörn Müller-Quade for the original inspiration for this work and Christian Schaffner for valuable discussions.

References

1. Ben-Or, M., Crépeau, C., Gottesman, D., Hassidim, A., Smith, A.: Secure multi-party quantum computation with (only) a strict honest majority. In: FOCS 2006, pp. 249–260. IEEE Computer Society, Los Alamitos (2006)
2. Ben-Or, M., Horodecki, M., Leung, D.W., Mayers, D., Oppenheim, J.: The universal composable security of quantum key distribution. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 386–406. Springer, Heidelberg (2005); Preprint at arXiv:quant-ph/0409078v1
3. Ben-Or, M., Mayers, D.: General security definition and composability for quantum & classical protocols. arXiv:quant-ph/0409062v2 (September 2004)
4. Bennett, C.H., Brassard, G.: Quantum cryptography: Public-key distribution and coin tossing. In: IEEE International Conference on Computers, Systems and Signal Processing 1984, pp. 175–179. IEEE Computer Society, Los Alamitos (1984)
5. Bennett, C.H., Brassard, G., Crépeau, C., Skubiszewska, M.-H.: Practical quantum oblivious transfer. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 351–366. Springer, Heidelberg (1992)
6. Brassard, G., Crépeau, C., Jozsa, R., Langlois, D.: A quantum bit commitment scheme provably unbreakable by both parties. In: FOCS 1993, Los Alamitos, CA, USA, pp. 362–371. IEEE Computer Society, Los Alamitos (1993)
7. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: FOCS 2001, pp. 136–145. IEEE Computer Society, Los Alamitos (2001); Full and revised version is IACR ePrint 2000/067

8. Canetti, R., Fischlin, M.: Universally composable commitments. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 19–40. Springer, Heidelberg (2001); Full version is IACR ePrint 2001/055
9. Damgård, I., Fehr, S., Lunemann, C., Salvail, L., Schaffner, C.: Improving the security of quantum protocols. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 408–427. Springer, Heidelberg (2009)
10. Damgård, I., Fehr, S., Salvail, L., Schaffner, C.: Cryptography in the bounded quantum-storage model. In: FOCS 2005, pp. 449–458 (2005); Full version is arXiv:quant-ph/0508222v2
11. Fehr, S., Schaffner, C.: Composing quantum protocols in a classical environment. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 350–367. Springer, Heidelberg (2009)
12. Hofheinz, D., Müller-Quade, J.: A paradox of quantum universal composability. In: 4th European QIPC Workshop, poster (2003), http://www.qipprocone.org/Hot%20Topics%20posters/muellerquade_poster.pdf
13. Ishai, Y., Prabhakaran, M., Sahai, A.: Founding cryptography on oblivious transfer – efficiently. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 572–591. Springer, Heidelberg (2008)
14. Kent, A.: Unconditionally secure bit commitment. PRL 83(7), 1447–1450 (1999)
15. Kilian, J.: Founding cryptography on oblivious transfer. In: STOC 1988, pp. 20–31. ACM Press, New York (1988)
16. Mayers, D.: Unconditionally Secure Quantum Bit Commitment is Impossible. Physical Review Letters 78(17), 3414–3417 (1997); Preprint at arXiv:quant-ph/9605044v2
17. Müller-Quade, J., Unruh, D.: Long-term security and universal composability. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 41–60. Springer, Heidelberg (2007)
18. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000)
19. Pfitzmann, B., Waidner, M.: A model for asynchronous reactive systems and its application to secure message transmission. In: 22nd IEEE Symposium on Security & Privacy, pp. 184–200 (2001)
20. Unruh, D.: Simulatable security for quantum protocols (September 2004), arXiv:quant-ph/0409125v2
21. Unruh, D.: Universally composable quantum multi-party computation (October 2009), arXiv:0910.2912 [quant-ph], Full version of this paper
22. van de Graaf, J.: Towards a formal definition of security for quantum protocols. PhD thesis, Départment d'informatique et de r.o., Université de Montréal (1998), http://www.cs.mcgill.ca/~crepeau/PS/these-jeroen.ps
23. Watrous, J.: Zero-knowledge against quantum attacks. In: STOC 2006, pp. 296–305. ACM, New York (2006)
24. Wehner, S., Wullschleger, J.: Composable security in the bounded-quantum-storage model. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 604–615. Springer, Heidelberg (2008); Full version is arXiv:0709.0492v1 [quant-ph]
25. Wiesner, S.: Conjugate coding. SIGACT News 15(1), 78–88 (1983) (manuscript written ca. 1970)
26. Yao, A.C.-C.: Security of quantum protocols against coherent measurements. In: STOC 1995, pp. 67–75. ACM, New York (1995)

A Simple BGN-Type Cryptosystem from LWE

Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan

IBM Research

Abstract. We construct a simple public-key encryption scheme that supports polynomially many additions and one multiplication, similar to the cryptosystem of Boneh, Goh, and Nissim (BGN). Security is based on the hardness of the learning with errors (LWE) problem, which is known to be as hard as certain worst-case lattice problems.

Some features of our cryptosystem include support for large message space, an easy way of achieving formula-privacy, a better message-to-ciphertext expansion ratio than BGN, and an easy way of multiplying two encrypted polynomials. Also, the scheme can be made identity-based and leakage-resilient (at the cost of a higher message-to-ciphertext expansion ratio).

1 Introduction

In this work we describe an encryption scheme which is additively homomorphic, and in addition also supports one multiplication. Our scheme is based on the trapdoor function proposed by Gentry, Peikert and Vaikuntanathan [10] (henceforth referred to as the GPV trapdoor function). Recall that the "public key" in the GPV trapdoor function is a matrix $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ (for parameters p and $m > n$), and the corresponding trapdoor is a full rank integer matrix with small entries $\mathbf{T} \in \mathbb{Z}^{m\times m}$ such that $\mathbf{TA} = 0 \pmod q$. The public and secret keys in our cryptosystem are exactly the same as in the GPV trapdoor function. We encrypt a square binary matrix $\mathbf{B} \in \mathbb{Z}_2^{m\times m}$ by setting

$$\mathbf{C} = \mathbf{AS} + 2\mathbf{X} + \mathbf{B} \bmod q$$

where $\mathbf{S}$ is a random "coefficient matrix" $\mathbf{S} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{X}$ is a "noise matrix" with small entries $\mathbf{X} \in Z^{m\times m}$.

Ciphertext matrices can be added, and a single matrix multiplication $\mathbf{C}' = \mathbf{C}_1 \cdot \mathbf{C}_2^t \bmod q$ is also supported. ($\mathbf{C}^t$ is the transpose of $\mathbf{C}$.) To decrypt, we set

$$\mathbf{B} = \mathbf{T}^{-1} \cdot (\mathbf{TCT}^t \bmod q) \cdot (\mathbf{T}^t)^{-1} \bmod 2$$

The security of our scheme is equivalent to the hardness of learning with errors (LWE). This problem, which is related to the well-known "learning parity with noise", has become standard in the study of lattice-based cryptography. The problem was first proposed by Regev [14], and shown by Regev [14] and Peikert [13] to be as hard as *worst-case instances* of various problems in integer lattices.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 506–522, 2010.

1.1 An Abridged History of Homomorphic Encryption

Encryption schemes that support operations on encrypted data (aka homomorphic encryption) are very useful for secure computation. Many public-key cryptosystems supports either addition or multiplication of encrypted data, but obtaining both at the same time seems harder.

It is known that computing arbitrary functions on encrypted data can be implemented, e.g., using Yao's "garbled circuit" technique [16,12], but the size of the ciphertext and complexity of decryption grow at least linearly with the number of gates in the circuit being computed. Also, Sander, Young and Yung [15] described a technique that permits evaluation of arbitrary circuits, but the ciphertext size grows exponentially with the circuit depth. Both of these methods can be implemented using only "general hardness assumptions" (e.g., the existence of two-flow Oblivious-Transfer protocols etc.)

Boneh, Goh, and Nissim described a cryptosystem that permitted arbitrary number of additions and one multiplication, without growing the ciphertext size [5]. Below we refer to this scheme as the BGN cryptosystem. Security of the BGN cryptosystem is based on the subgroup-membership problem in composite-order groups that admit bilinear maps. This cryptosystem immediately implies an efficient protocol for evaluating 2DNF formula (or more generally bilinear forms). Boneh et al. also described applications of the BGN cryptosystem to improving the efficiency of private information retrieval schemes (PIR) and for a voting protocol.

More recently, Aguilar Melchor, Gaborit, and Herranz described in [2] a "template" for converting additively homomorphic encryption into a cryptosystem that permits both additions and multiplications. They show how to use this template to combine the BGN cryptosystem with the cryptosystem of Kawachi et al. [11], thus obtaining a cryptosystem that supports two multiplications and arbitrary additions, based on the hardness of both the subgroup membership problem and the unique-shortest vector problem in lattices. They also show how to use this template with the cryptosystem of Aguilar Melchor et al. [1] in order to obtain unlimited multiplication depth, where the ciphertext size grows exponentially with the multiplication depth but additions are supported without increasing the size. (Security of this last realization is based on a relatively unstudied hardness assumption, called the "Differential Knapsack Vector Problem.")

Very recently, Gentry described a fully homomorphic cryptosystem [9], supporting polynomially many additions and multiplications without increasing the ciphertext size, with security based on the hardness of finding short vectors in ideal lattices [8].

1.2 Our Contributions

Even given the great advances in homomorphic encryption over the last year, our scheme still offers some advantages over prior schemes in the literature. Below we list some of these advantages, mostly in comparison to the BGN cryptosystem.

Perhaps the main difference between our scheme and previous work is the underlying hardness assumption. In particular, ours is the first reported cryptosystem based on LWE that has more than just additive homomorphism. Also, our scheme is very efficient: it can encrypt a matrix of m^2 elements in time $\tilde{O}(m^3)$, and decryption takes comparable time.

One important difference between our scheme and the BGN cryptosystem is that the BGN cryptosystem can only encrypt messages from a small space (since on decryption one only recovers a group element g^m, and then need to search for the message m). In our scheme, we can replace the binary matrices by matrices over $\mathbb{Z}_p$ for any p, as long as the ciphertext is defined over $\mathbb{Z}_q$ where q is sufficiently larger than p. A related advantage is that by choosing a large modulus p, our scheme can be made to have ciphertext expansion of O(1) (whereas the BGN cryptosystem expands $O(\log n)$ bits of plaintext to $O(n)$ ciphertext bits).[1]

We also note that the modulus p that defines the message space in our scheme can be chosen dynamically by the encryptor: the same public/secret key pair can be used to encrypt/decrypt messages modulo many different fields (or rings). Our scheme also support ciphertext blinding (a given ciphertext is converted into a random ciphertext that encrypts the same thing), and also the stronger property of modular blinding: Given a ciphertext that encrypts a matrix $\mathbf{B} \in \mathbb{Z}_p^{m\times n}$, and given some divisor p' of p, we can produce a random ciphertext that encrypts $\mathbf{B}$ mod p'. For example, if the original plaintext matrix had numbers in $\mathbb{Z}_{2^n}$, we can blind the ciphertext so as to erase all but the least-significant bits of the entries in $\mathbf{B}$.

One consequence of the (standard) blinding property and the flexibility of choosing the message space is that our system provide a very simple procedure for formula-private secure computation. Namely, it is very easy to compute a 2DNF formulas (or a general bilinear form) on ciphertexts, while at the same time hiding from the holder of the secret key everything about the formula itself (other than the result of applying it on the given inputs).

Finally, our scheme inherits much of the flexibility that comes with LWE-based cryptosystems. In particular, it can be made identity-based (in the random-oracle model) using the construction of Gentry et al. [10], and it can be made leakage resilient using a recent result of Dodis et al. [6]. Both of these applications follow from the observation that the "dual Regev cryptosystem" from [10] can be described as a special case of our cryptosystem.

Relation to the AMGH transformation. It turns out that our cryptosystem fits "right out of the box" in the template of Aguilar Melchor et al. [2]. Their transformation apply to any additively homomorphic cryptosystem for which you can embed the ciphertexts back into the plaintext space while maintaining the semantics of addition, which is easy in our case. See the appendix for a brief description of their transformation and how it applies to our cryptosystem.

[1] To achieve such bandwidth efficient encryption, an application would have to encode its input as a matrix. Although this can always be done, it is not clear that such encoding will maintain the semantics of multiplication that the application needs. See some examples of this point in Section 5.

Combining our cryptosystem with the AMGH transformation yields a homomorphic encryption scheme for circuits of logarithmic multiplication depth (with arbitrary additions), whose security is based on the hardness of LWE.[2] We point out that even in this context, using our native multiplication operation will be advantageous, since it does not increase the ciphertext size (or the decryption time). Thus we can get either one more multiplication level for a given complexity bound, or a more efficient scheme for the same circuit depth.

Applications. Clearly, our scheme can be used as a drop-in replacement in the applications to voting and PIR that were discussed in the paper of Boneh et al. [5]. In addition, since out scheme encrypts matrices natively, it is a good match for applications that can benefit from batching, or when efficient linear algebra is important. Some examples of batching include applications that need to multiply polynomials (whose coefficients are to be encoded in the entries of the plaintext matrix) or large integers (whose bit representation is to be encoded in the entries of the plaintext matrix). In Section 5.3 we describe how these can be encoded in a matrix so that a single multiplication of $m \times m$ matrices can be used to multiply two degree-$(m-1)$ polynomials (or two m-bit integers), so that the result does not leak anything about the inputs other than their product.

2 Preliminaries

Notations. We denote scalars by lower-case letters $(a, b, \ldots)$, vectors by lower-case bold letters $(\mathbf{a}, \mathbf{b}, \ldots)$, and matrices by upper-case bold letters $(\mathbf{A}, \mathbf{B}, \ldots)$. We denote the Euclidean norm of a vector $\mathbf{v}$ by $\|\mathbf{v}\|$, and the largest entry in a vector or a matrix is denoted $\|\mathbf{v}\|_\infty$ or $\|\mathbf{M}\|_\infty$, respectively. We consider the operation $(a \bmod q)$ as mapping the integer a into the interval $(-q/2, +q/2]$.

2.1 Learning with Errors (LWE)

The LWE problem was introduced by Regev [14] as a generalization of "learning parity with noise". For positive integers n and $q \geq 2$, a vector $\mathbf{s} \in \mathbb{Z}_q^n$, and a probability distribution χ on $\mathbb{Z}_q$, let $A_{\mathbf{s},\chi}$ be the distribution obtained by choosing a vector $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random and a noise term $x \leftarrow \chi$, and outputting $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + x) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.

Definition 1 (LWE). *For an integer $q = q(n)$ and an error distribution $\chi = \chi(n)$ over $\mathbb{Z}_q$, the learning with errors problem* $\mathsf{LWE}_{n,m,q,\chi}$ *is defined as follows: Given m independent samples from $A_{\mathbf{s},\chi}$ (for some $\mathbf{s} \in \mathbb{Z}_q^n$), output $\mathbf{s}$ with noticeable probability.*

The decision variant of the LWE problem, denoted $\mathsf{distLWE}_{n,m,q,\chi}$, *is to distinguish (with non-negligible advantage) m samples chosen according to $A_{\mathbf{s},\chi}$ (for*

[2] We comment that the AMGH transformation appears to be inherently "non private", in that the holder of the secret key can deduce the multiplication structure of the circuit that was used to generate a given ciphertext. This can be addressed using generic techniques such as Yao's garbled circuits.

uniformly random $\mathbf{s} \in_R \mathbb{Z}_q^n$*), from* m *samples chosen according to the uniform distribution over* $\mathbb{Z}_q^n \times \mathbb{Z}_q$*.*

For cryptographic applications we are primarily interested in the decision problem $\mathsf{distLWE}$. Regev [14] showed that for a prime modulus q, $\mathsf{distLWE}$ can be reduced to worst-case LWE, with a loss of up to a $q \cdot \mathsf{poly}(n)$ factor in the parameter m.

At times, we find it convenient to describe the LWE problem $\mathsf{LWE}_{n,m,q,\chi}$ using a compact matrix notation: given $(\mathbf{A}, \mathbf{As} + \mathbf{x})$ where $\mathbf{A} \leftarrow \mathbb{Z}_q^{m \times n}$ is uniformly random, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ is the LWE secret, and $\mathbf{x} \leftarrow \chi^m$, find $\mathbf{s}$. We also use similar matrix notation for the decision version $\mathsf{distLWE}$.

Gaussian error distributions $\overline{\mathbf{\Psi}}_\beta$*.* We are primarily interested in the LWE and $\mathsf{distLWE}$ problems where the error distribution χ over $\mathbb{Z}_q$ is derived from a Gaussian. For any $\beta > 0$, the density function of a Gaussian distribution over the reals is give by $D_\beta(x) = 1/\beta \cdot \mathsf{exp}(-\pi(x/\beta)^2)$. For an integer $q \geq 2$, define $\overline{\Psi}_\beta(q)$ to be the distribution on $\mathbb{Z}_q$ obtained by drawing $y \leftarrow D_\beta$ and outputting $\lfloor q \cdot y \rceil$ (mod q). We write $\mathsf{LWE}_{n,m,q,\beta}$ as an abbreviation for $\mathsf{LWE}_{n,m,q,\overline{\Psi}_\beta(q)}$.

Here we state some basic facts about Gaussians (tailored to the error distribution $\overline{\Psi}_\beta$); see, e.g. [7]. (In what follows overwhelming probability means probability $1 - \delta$ for δ which is negligible in n.)

Fact 1. *Let* $\beta > 0$ *and* $q \in \mathbb{Z}$*, and let the vector* $\mathbf{x}$ *be chosen as* $\mathbf{x} \leftarrow \overline{\Psi}_\beta(q)^n$*. Also let* $\mathbf{y} \in \mathbb{Z}^n$ *be an arbitrary vector and let* $g = \omega(\sqrt{\log n})$*. Then with overwhelming probability* $|\langle \mathbf{x}, \mathbf{y} \rangle| \leq \beta q \cdot g \cdot \|\mathbf{y}\|$*.*

Fact 2. *Let* $y \in \mathbb{R}$ *be arbitrary. The statistical distance between the distributions* $\overline{\Psi}_\beta$ *and* $\overline{\Psi}_\beta + y$ *is at most* $|y|/(\beta q)$*.*

Evidence for the hardness of $\mathsf{LWE}_{n,m,q,\beta}$ follows from results of Regev [14], who gave a *quantum* reduction from approximating certain problems on n-dimensional lattices in the worst case to within $\widetilde{O}(n/\beta)$ factors to solving $\mathsf{LWE}_{n,m,q,\beta}$ for any desired $m = \mathsf{poly}(n)$, when $\beta \cdot q \geq 2\sqrt{n}$. Recently, Peikert [13] also gave a related *classical* reduction for some other problems with similar parameters.

2.2 Trapdoor Sampling

The basis of our encryption scheme is a trapdoor sampling algorithm first constructed by Ajtai [3], and later improved by Alwen and Peikert [4]. The trapdoor sampling procedure generates an (almost) uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, together with a matrix $\mathbf{T} \in \mathbb{Z}^{m \times m}$ such that (a) $\mathbf{T} \cdot \mathbf{A} = 0 \pmod q$, (b) $\mathbf{T}$ is invertible, and (c) the entries of $\mathbf{T}$ are small (say, of size $O(n \log q)$).

The trapdoor $\mathbf{T}$ can be used to solve the LWE problem relative to $\mathbf{A}$, i.e., given $\mathbf{y} = \mathbf{As} + \mathbf{x}$ where $\mathbf{x}$ is any "sufficiently short" vector, it can be used to recover $\mathbf{s}$. This is done as follows: compute

$$\mathbf{Ty} \; = \; \mathbf{T}(\mathbf{As} + \mathbf{x}) = \mathbf{TAs} + \mathbf{Tx} = \mathbf{Tx} \pmod q$$

where the last equality follows since the rows of $\mathbf{T}$ belong to lattice $\Lambda^\perp(\mathbf{A})$. Now, since both $\mathbf{T}$ and $\mathbf{x}$ contain small entries, each entry of the vector $\mathbf{Tx}$ is smaller than q, and thus $\mathbf{Tx} \bmod q$ is $\mathbf{Tx}$ itself! Finally, multiplying by $\mathbf{T}^{-1}$ (which is well-defined since $\mathbf{T}$ is a basis and therefore has full rank) gives us $\mathbf{x}$. The LWE secret $\mathbf{s}$ can then be recovered by Gaussian elimination. We state the result of Alwen and Peikert [4] below.

Lemma 1 ([3,4]). *There is a probabilistic polynomial-time algorithm* TrapSamp *that, on input 1^n, a positive integer $q \geq 2$, and a* $\mathsf{poly}(n)$*-bounded positive integer $m \geq 8n \log q$, outputs matrices $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and $\mathbf{T} \in \mathbb{Z}^{m \times m}$ such that:*

- *$\mathbf{A}$ is statistically close to uniform over $\mathbb{Z}_q^{m \times n}$,*
- *the rows of $\mathbf{T}$ form a basis of the lattice $\Lambda^\perp(\mathbf{A}) \overset{\text{def}}{=} \{\mathbf{w} \in \mathbb{Z}^m \ : \ \mathbf{w} \cdot \mathbf{A} = 0 \pmod q\}$,*
- *the Euclidean norm of all the rows is $\mathbf{T}$ (and therefore also $\|\mathbf{T}\|_\infty$) is bounded by $O(n \log q)$. (Alwen and Peikert assert that the constant hidden in the $O(\cdot)$ is no more than 20.)*

We note that since the rows of $\mathbf{T}$ span the lattice $\Lambda^\perp(\mathbf{A})$, it follows that $\det(\mathbf{T}) = q^n$, hence for odd q we know that $\mathbf{T}$ is invertible mod 2.

3 The Encryption Scheme

For ease of presentation, we focus below on the case of encrypting binary matrices. The extension for encrypting matrices mod p for $p > 2$ is straightforward, and is discussed in Section 5.1.

Below we let n denote the security parameter. Other parameters of the system are two numbers $m, q = \mathsf{poly}(n)$ (with q an odd prime), and a Gaussian error parameter $\beta = 1/\mathsf{poly}(n)$. (See Section 3.2 for concrete instantiations of these parameters.) For these parameters, the message space is the set of binary m-by-m matrices, i.e., $\mathbf{B} \in \mathbb{Z}_2^{m \times m}$. Public keys are matrices $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, secret key are matrices $\mathbf{T} \in \mathbb{Z}_q^{m \times m}$, and ciphertexts are matrices $\mathbf{C} \in \mathbb{Z}_q^{m \times m}$.

$\mathsf{KeyGen}(1^n)$: Run the trapdoor sampling algorithm TrapSamp of Lemma 1 to obtain a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ together with the trapdoor matrix $\mathbf{T} \in \mathbb{Z}^{m \times m}$, $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{TrapSamp}(1^n, q, m)$. The public key is $\mathbf{A}$ and the secret key is $\mathbf{T}$.

$\mathsf{Enc}(\mathbf{A}, \mathbf{B} \in \{0,1\}^{m \times m})$: Choose a uniformly random matrix $\mathbf{S} \overset{\$}{\leftarrow} \mathbb{Z}_q^{n \times m}$ and an "error matrix" $\mathbf{X} \overset{\$}{\leftarrow} \overline{\Psi}_\beta(q)^{m \times m}$. Output the ciphertext

$$\mathbf{C} \leftarrow \mathbf{AS} + 2\mathbf{X} + \mathbf{B} \pmod q$$

(Here, $2\mathbf{X}$ means multiplying each entry of the matrix $\mathbf{X}$ by 2.)

$\mathsf{Dec}(\mathbf{T}, \mathbf{C})$: Set $\mathbf{E} \leftarrow \mathbf{TCT}^t \bmod q$, and then output $\mathbf{B} \leftarrow \mathbf{T}^{-1}\mathbf{E}(\mathbf{T}^t)^{-1} \bmod 2$.

To see that decryption works, recall that $\mathbf{T} \cdot \mathbf{A} = 0 \pmod q$ and therefore $\mathbf{TCT}^t = \mathbf{T}(2\mathbf{X}+\mathbf{B})\mathbf{T}^t \pmod q$. If in addition all the entries of $\mathbf{T}(2\mathbf{X}+\mathbf{B})\mathbf{T}^t$ are smaller than q then we also have the equality over the integers $\mathbf{E} = (\mathbf{TCT}^t \bmod q) = \mathbf{T}(2\mathbf{X} + \mathbf{B})\mathbf{T}^t$, and hence $\mathbf{T}^{-1}\mathbf{E}(\mathbf{T}^t)^{-1} = \mathbf{B} \pmod 2$. This means that we have correct decryption as long as we set the parameter β small enough so that with high probability all the entries of $\mathbf{T}(2\mathbf{X} + \mathbf{B})\mathbf{T}^t$ are smaller than $q/2$.

Remark 1. Note that the right-multiplication by $\mathbf{T}^t$ and $(\mathbf{T}^t)^{-1}$ on decryption are redundant here, we can instead just compute $\mathbf{B} \leftarrow \mathbf{T}^{-1}(\mathbf{TC} \bmod q) \bmod 2$. The right-multiplication is needed to decrypt product ciphertexts, as described below. As opposed to the BGN cryptosystem, in our scheme the "normal ciphertexts" and "product ciphertexts" live in the same space, and we can use the same decryption procedure to decrypt both.

Also, we can optimize away the need to multiply by $\mathbf{T}^{-1}$ and $(\mathbf{T}^t)^{-1}$ by using the modified trapdoor $\mathbf{T}' = (\mathbf{T}^{-1} \bmod 2) \cdot \mathbf{T}$ (product over the integers). Clearly we have $\mathbf{T}'\mathbf{A} = 0 \pmod q$, and the entries of $\mathbf{T}'$ are not much larger than those of $\mathbf{T}$ (since $(\mathbf{T}^{-1} \bmod 2)$ is a 0-1 matrix).

3.1 Homomorphic Operations

Addition. Given two ciphertexts $\mathbf{C}_1, \mathbf{C}_2$ that decrypt to $\mathbf{B}_1, \mathbf{B}_2$, respectively, it is easy to see that the matrix $\mathbf{C} = \mathbf{C}_1 + \mathbf{C}_2 \bmod q$ would be decrypted to $\mathbf{B}_1 + \mathbf{B}_2 \bmod 2$, as long as there is no "overflow" in any entry. Specifically, if we have $\mathbf{C}_1 = \mathbf{AS}_1 + 2\mathbf{X}_1 + \mathbf{B}_1$ and $\mathbf{C}_1 = \mathbf{AS}_2 + 2\mathbf{X}_2 + \mathbf{B}_2$ then

$$\mathbf{C} = \mathbf{C}_1 + \mathbf{C}_2 = \mathbf{A}(\mathbf{S}_1 + \mathbf{S}_2) + 2(\mathbf{X}_1 + \mathbf{X}_2) + (\mathbf{B}_1 + \mathbf{B}_2)$$

which would be decrypted as $\mathbf{B}_1 + \mathbf{B}_2$ as long as all the entries in $\mathbf{T}(2(\mathbf{X}_1 + \mathbf{X}_2) + \mathbf{B}_1 + \mathbf{B}_2)\mathbf{T}^t$ are smaller than $q/2$. See Section 3.2 for the exact parameters.

Multiplication. Given two ciphertexts $\mathbf{C}_1, \mathbf{C}_2$ that encrypt $\mathbf{B}_1, \mathbf{B}_2$, respectively, we compute the product ciphertext as $\mathbf{C} = \mathbf{C}_1 \cdot \mathbf{C}_2^t \bmod q$. If we have $\mathbf{C}_1 = \mathbf{AS}_1 + 2\mathbf{X}_1 + \mathbf{B}_1$ and $\mathbf{C}_2 = \mathbf{AS}_2 + 2\mathbf{X}_2 + \mathbf{B}_2$ then

$$\begin{aligned}\mathbf{C} = \mathbf{C}_1 \cdot \mathbf{C}_2^t &= (\mathbf{AS}_1 + 2\mathbf{X}_1 + \mathbf{B}_1)(\mathbf{AS}_2 + 2\mathbf{X}_2 + \mathbf{B}_2)^t \\ = \mathbf{A} \cdot \underbrace{(\mathbf{S}_1\mathbf{C}_2^t)}_{\mathbf{S}} &+ 2\underbrace{(\mathbf{X}_1(2\mathbf{X}_2 + \mathbf{B}_2) + \mathbf{B}_1\mathbf{X}_2^t)}_{\mathbf{X}} + \underbrace{\mathbf{B}_1\mathbf{B}_2^t}_{\mathbf{B}} + \underbrace{(2\mathbf{X}_1 + \mathbf{B}_1)\mathbf{S}_2^t}_{\mathbf{S}'} \cdot \mathbf{A}^t \pmod q.\end{aligned}$$

Hence the product ciphertext has the form $\mathbf{AS} + 2\mathbf{X} + \mathbf{B} + \mathbf{S}'\mathbf{A}^t$.

As before, we see that $\mathbf{TCT}^t = \mathbf{T}(2\mathbf{X} + \mathbf{B})\mathbf{T}^t \pmod q$, and if all the entries of $\mathbf{T}(2\mathbf{X} + \mathbf{B})\mathbf{T}^t$ are smaller than $q/2$ then we have $\mathbf{E} = (\mathbf{TCT}^t \bmod q) = \mathbf{T}(2\mathbf{X} + \mathbf{B})\mathbf{T}^t$ over the integers, and therefore $\mathbf{T}^{-1}\mathbf{E}(\mathbf{T}^t)^{-1} = \mathbf{B} \pmod 2$. Below we establish the parameters that we need for this to work.

Remark 2. We remark that the $AS + 2X + B$ format for ciphertexts, which is borrowed from [9], seem particularly conducive for homomorphic encryption. When applied in a commutative ring as in [9], it supports a large number of additions and multiplications on ciphertexts. In our case we use it in the ring of matrices, which is not commutative, but multiplying by the transpose offers a partial workaround, supporting one level of multiplication.

3.2 Setting the Parameters

Theorem 1. *Fix the security parameter n and any $c = c(n) > 0$. Let q, m, β be set as*

$$\begin{aligned} q &> 2^{20}(c+4)^3 n^{3c+4} \log^5 n, \quad q \textit{ is a prime} \\ m &= \lfloor 8n \log q \rfloor \\ \beta &= \frac{1}{27n^{1+(3c/2)} \log n \log q \sqrt{qm}} \end{aligned}$$

Then the encryption scheme from above with parameters n, m, q, β supports n^c additions and one multiplication (in any order) over the matrix ring $\mathbb{Z}_2^{m \times m}$.

Remark 3. Note that in Theorem 1 we can allow n^c additions for a non-constant c. The reason that this may be needed is for taking linear combinations of ciphertexts with large coefficients. Specifically, if we have ciphertext matrices $\mathbf{C}_1, \mathbf{C}_2, \ldots$, we can homomorphically compute $\sum \alpha_i \mathbf{C}_i$ as long as $|\sum \alpha_i| < n^c$.

Proof. First, let $\mathbf{C}$ be a matrix that was obtained by adding $\ell \leq n^c$ ciphertexts, $\mathbf{C} = \sum_{i=1}^{\ell}(\mathbf{AS}_i + 2\mathbf{X}_i + \mathbf{B}_i)$. Denote $\mathbf{X} = \sum_{i=1}^{\ell} \mathbf{X}_i$, and $\mathbf{B} = \sum_{i=1}^{\ell} \mathbf{B}_i$, and we analyze the size of the entries in the matrix $\mathbf{T}(2\mathbf{X} + \mathbf{B})$. Recall from Lemma 1 that every row of T has Euclidean norm at most $20n \log q$. Applying Fact 1 (with $g = \log n - 1$), with overwhelming probability every entry of $\mathbf{TX}_i$ is at most $20\beta q(\log n - 1)n \log q$, hence every entry of $\mathbf{TX}$ is at most $20\ell\beta q(\log n - 1)n \log q$. At the same time, all the $\mathbf{B}_i$'s are binary so each entry of $\mathbf{TB}$ is at most $20\ell n \log q$. Hence the absolute value of each entry in $\mathbf{T}(2\mathbf{X} + \mathbf{B})$ is bounded by

$$\begin{aligned} & 20\ell n \log q \cdot (2\beta q(\log n - 1) + 1) \;<\; 20\ell n \log q \cdot 2\beta q \log n \\ = \;& \frac{40\ell \cdot n \log n \cdot q \log q}{27n^{1+(3c/2)} \log n \cdot \log q \sqrt{qm}} \;=\; \frac{40\ell\sqrt{q}}{27n^{3c/2}\sqrt{m}} \;\overset{(\star)}{\ll}\; \sqrt{q/m} \end{aligned}$$

where inequality $(\star)$ uses the fact that $\ell \leq n^c$. This in particular means that each entry in $\mathbf{T}(2\mathbf{X}+\mathbf{B})\mathbf{T}^t$ is bounded by $m \cdot 20n \log q \cdot \sqrt{q/m} = 20n \log q \sqrt{qm} \ll q/2$. Since $\mathbf{TA} = 0 \pmod q$ then $\mathbf{TCT}^t = \mathbf{T}(2\mathbf{X} + \mathbf{B})\mathbf{T}^t \pmod q$, and as all the entries in $\mathbf{T}(2\mathbf{X}+\mathbf{B})$ are less than $q/2$ in absolute value, we have the equality over the integers $(\mathbf{TCT}^t \bmod q) = \mathbf{T}(2\mathbf{X}+\mathbf{B})\mathbf{T}^t$, hence $\mathbf{T}^{-1}(\mathbf{TCT}^t \bmod q)(\mathbf{T}^t)^{-1} = \mathbf{B} \pmod 2$.

Next, consider a circuit with one ℓ_1-fan-in addition layer, followed by a multiplication layer of fan-in two, and another ℓ_2-fan-in layer of addition, where $\ell_1 + \ell_2 \leq n^c$. We have shown above that when multiplying two matrices of the form $\mathbf{AS}_i + 2\mathbf{X}_i + \mathbf{B}_i$ $(i = 1, 2)$, the result is of the form $\mathbf{AS} + 2\mathbf{X} + \mathbf{B} + \mathbf{S}'\mathbf{A}^t$. Hence all the matrices at the output of the multiplication layer are of this form, and therefore so is the output ciphertext that results from adding them all together. We now proceed to show that for that final ciphertext $\mathbf{C} = \mathbf{AS} + 2\mathbf{X} + \mathbf{B} + \mathbf{S}'\mathbf{A}^t$, it holds that every entry in $\mathbf{T}(2\mathbf{X} + \mathbf{B})\mathbf{T}^t$ is less that $q/2$ in absolute value.

Consider one particular ciphertext at the output of the multiplication layer, this ciphertext is of the form $\mathbf{C}_i = \mathbf{AS} + (2\mathbf{X}_1 + \mathbf{B}_1)(2\mathbf{X}_2^t + \mathbf{B}_2^t) + \mathbf{S}'\mathbf{A}^t$, and the matrices $(2\mathbf{X}_i + \mathbf{B}_i)$ were obtained from adding upto ℓ_1 encryptions. By the analysis from above, each entry in $\mathbf{T}(2\mathbf{X}_1 + \mathbf{B}_1)$ is bounded by $\frac{40\ell_1\sqrt{q}}{27n^{3c/2}\sqrt{m}}$, and the same bound apply also to each entry in $(2\mathbf{X}_2^t + \mathbf{B}_2^t)\mathbf{T}^t$. Hence each entry in the product $\mathbf{T}(2\mathbf{X}_1 + \mathbf{B}_1)(2\mathbf{X}_2^t + \mathbf{B}_2^t)\mathbf{T}^t$ is bounded by

$$m \cdot \left(\frac{40\ell_1\sqrt{q}}{27n^{3c/2}\sqrt{m}}\right)^2 \;=\; \left(\frac{40}{27}\right)^2 \cdot \frac{\ell_1^2}{n^{3c}} \cdot q$$

Adding $\ell_2 \leq n^c - \ell_1$ such matrices, the entry in the result is bounded by

$$\left(\frac{40}{27}\right)^2 \cdot \frac{\ell_1^2(n^c - \ell_1)}{n^{3c}} \cdot q \;\overset{(\star)}{\leq}\; \left(\frac{40}{27}\right)^2 \cdot \frac{2}{9} \cdot q \;<\; q/2$$

where the inequality $(\star)$ follows since the function $f(x) = x^2(a - x)$ obtains its maximum at $x = 2a/3$, where $f(x) = 2a^3/9$.

Once again, since each entry in $\mathbf{T}(2\mathbf{X}+\mathbf{B})\mathbf{T}^t$ is less than $q/2$ in absolute value, and since $\mathbf{TCT}^t = \mathbf{T}(2\mathbf{X}+\mathbf{B})\mathbf{T}^t \pmod q$, we have the equality over the integers $(\mathbf{TCT}^t \bmod q) = \mathbf{T}(2\mathbf{X}+\mathbf{B})\mathbf{T}^t$, which means that $\mathbf{T}^{-1}(\mathbf{TCT}^t \bmod q)(\mathbf{T}^t)^{-1} = \mathbf{B} \pmod 2$.

4 Security

The CPA security of the encryption scheme follows directly from the hardness of the decision LWE problem, as we now prove.

Theorem 2. *Any distinguishing algorithm with advantage ϵ against the CPA security of the scheme with parameters n, m, q, β, can be converted to a distinguisher against $\mathsf{distLWE}_{n,m,q,\beta}$ with roughly the same running time and advantage at least $\epsilon/2m$.*

Proof. Let $\mathcal{A}$ be a CPA-adversary that distinguishes between encryptions of messages of its choice with advantage ϵ. We first construct a distinguisher $\mathcal{D}$ with advantage at least $\epsilon/2$ between the two distributions

$$\left\{(\mathbf{A}, \mathbf{AS}+\mathbf{X}) : \mathbf{A} \leftarrow \mathbb{Z}_q^{m\times n}, \mathbf{S} \leftarrow \mathbb{Z}_q^{n\times m}, \mathbf{X} \leftarrow \overline{\Psi}_\beta(q)^{m\times m}\right\} \text{ and } \left\{\mathsf{Unif}(\mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^{m\times m})\right\}$$

The distinguisher $\mathcal{D}$ takes as input a pair of matrices $(\mathbf{A} \in \mathbb{Z}_q^{m\times n}, \mathbf{C} \in \mathbb{Z}_q^{m\times m})$, and runs the adversary $\mathcal{A}$ with $\mathbf{A}$ as the public key. Upon receiving message $\mathbf{B}_0, \mathbf{B}_1$ from the adversary, $\mathcal{D}$ chooses at random $i \in_R \{0, 1\}$, returns the challenge ciphertext $2\mathbf{C} + \mathbf{B}_i \bmod q$, then outputs 1 if the adversary $\mathcal{A}$ guesses the right i, and 0 otherwise.

On the one hand, if $\mathbf{C}$ is a uniformly random matrix then the challenge ciphertext is also uniformly random, regardless of the choice of i. Hence in this case $\mathcal{D}$ outputs 1 with probability at most 1/2. On the other hand, if $\mathbf{C} = \mathbf{AS}+\mathbf{X} \bmod q$, then the challenge ciphertext is $2\mathbf{C} + \mathbf{B} = \mathbf{AS}' + 2\mathbf{X} + \mathbf{B} \bmod q$, where $\mathbf{S}' = 2\mathbf{S} \bmod q$ is uniformly distributed (since q and 2 are relatively prime). This is identical to the output distribution of $\mathsf{Enc}(PK, \mathbf{B}_i)$, hence by assumption $\mathcal{A}$ will guess the right i with probability $(1+\epsilon)/2$, which means that $\mathcal{D}$ outputs 1 with the same probability. Hence $\mathcal{D}$ has advantage at least $\epsilon/2$.

Finally, a standard hybrid argument can be used to convert the distinguisher $\mathcal{D}$ from above to a $\mathsf{distLWE}_{n,m,q,\beta}$ distinguisher with advantage $\epsilon/2m$.

Worst-case Connection. Regev [14] showed that if there is a PPT algorithm that solves $\mathsf{distLWE}_{n,m,q,\beta}$, then there is an $O(q \cdot m)$-time *quantum* algorithm that approximates various lattice problems on n-dimensional lattices in the worst case to within $\widetilde{O}(n/\beta)$ factors, when $\beta \cdot q \geq 2\sqrt{n}$. Recently, Peikert [13] also gave a related *classical* reduction with similar parameters.

Observe that for $n \geq \max\{140,\ 10\sqrt{c+4}\}$, the conditions on q, m, β imply that $\beta q > 2\sqrt{n}$. Plugging in our parameters m, q and β for the scheme that supports n^c additions, we get that breaking semantic security of the scheme is at least as hard as solving worst-case lattice problems to within a factor of $\tilde{O}(n^{3c+7/2})$.

5 Extensions and Applications

5.1 Encrypting Matrices over Larger Rings

As we said in the introduction, we can use the same scheme to encrypt matrices over larger rings and still enjoy the same homomorphic properties, just by working with a larger modulus q. Specifically, we can encrypt matrices over $\mathbb{Z}_p$ for any p by setting $q = \omega(p^2 n^{3c+1} \log^5 n)$ while keeping all the other parameters intact. We then encrypt a matrix $\mathbf{B} \in Z_p^{m \times m}$ as $\mathbf{C} = \mathbf{AS} + p\mathbf{X} + \mathbf{B}$, and decrypt it as $\mathbf{T}^{-1} \cdot (\mathbf{TCT}^t \bmod q) \cdot (\mathbf{T}^t)^{-1} \bmod p$. (We recall again that the determinant of $\mathbf{T}$ is q^n, so $\mathbf{T}$ is invertible mod p.) Using the above with $p \geq n^{3c+1} \log^5 n$, we have $q \leq p^3$ which means that our ciphertext expansion ratio is only three. (The plaintext has $m^2 \log p$ bits while the ciphertext has $m^2 \log q$ bits.)

We comment that once we fix these larger parameters, the choice of the underlying ring can be made adaptively by the encryptor. Namely, with the same public key $\mathbf{A}$ and secret key $\mathbf{T}$, the encryptor can choose the underlying ring as $\mathbb{Z}_r$ for any $r \leq p$ (thereby computing the ciphertext as $\mathbf{C} = \mathbf{AS} + r\mathbf{X} + \mathbf{B}$), and the decryptor can decrypt accordingly.

5.2 Formula Privacy

As described so far, the scheme does not ensure "formula privacy" against the holder of the secret key. For example, given a ciphertext matrix $\mathbf{C}$, the decryptor may be able to distinguish the case where this ciphertext was obtained by multiplying an encryption of the identity with an encryption of the zero matrix from the case where it was obtained by multiplying two encryptions of the zero matrix.

This deficiency can be remedied by standard techniques. We first need to increase the size of the modulus somewhat: switching from q as specified in Theorem 1 to $q' \geq q \cdot 2^{\omega(\log n)}$. Then given a ciphertext matrix $\mathbf{C}^*$, encrypting some plaintext matrix mod p, we blind it by setting

$$\mathbf{C} \;\leftarrow\; \mathbf{C}^* + \mathbf{A}\mathbf{S}_1 + p\mathbf{X} + \mathbf{S}_2^t\mathbf{A}^t,$$

where $\mathbf{S}, \mathbf{S}'$ are uniform in $\mathbb{Z}_{q'}^{n\times m}$ and each entry of $\mathbf{X}^*$ is chosen from $\overline{\Psi}_{\beta'}(q)$ with β' super-polynomially larger than the parameter β that is used in the scheme.

Using Fact 2 we can then show that the noise in the added $\mathbf{X}^*$ "drowns" all traces of the origin of this ciphertext. Namely, the resulting ciphertext is of the form $\mathbf{C} = \mathbf{A}\mathbf{S}_1' + p\mathbf{X}' + \mathbf{B} + (\mathbf{S}_2')^t\mathbf{A}^t$, where $\mathbf{S}_1', \mathbf{S}_2'$ are uniformly random, $\mathbf{B}$ is the corresponding plaintext, and the distribution of $\mathbf{X}'$ is nearly independent of the provenance of this ciphertext matrix.

We note that the same blinding technique can be used even if the encrypted plaintext matrix was chosen in a larger ring $\mathbb{Z}_{p'}$, as long as the parameter p that is used in the blinding procedure divides the original p'.

5.3 Encrypting Polynomials and Large Integers

To encrypt polynomials or large numbers, we need to encode them as matrices, in a way that would let us exploit the matrix operations that are supported natively by our scheme to do operations over the these polynomials or numbers.

We begin with polynomials: it is well known how to embed the coefficients of two polynomials in two matrices, so that multiplying these matrices we get all the coefficients of the resulting product polynomial. For example, for two polynomials $\hat{a}(x) = \sum a_i x^i$ and $\hat{b}(x) = \sum b_i x^i$, we can use

$$\mathbf{A} = \begin{pmatrix} a_3 & a_2 & a_1 \\ & a_3 & a_2 \\ & & a_3 \end{pmatrix} \mathbf{B} = \begin{pmatrix} b_1 & b_2 & b_3 \\ & b_1 & b_2 \\ & & b_1 \end{pmatrix} \;\Rightarrow\; \mathbf{A}\mathbf{B}^t = \begin{pmatrix} a_1b_3 + a_2b_2 + a3b_1 & a_1b_2 + a_2b_1 & a_1b_1 \\ a_2b_3 + a_3b_2 & \star & \star \\ a_3b_3 & \star & \star \end{pmatrix}$$

Note that the product matrix above is not private, in that it reveals more than just the coefficients of the product polynomial. This can be fixed easily by adding an encryption of a matrix with zero first column and first row and random entries everywhere else. Also, this simple embedding is "wasteful" in that it results in ciphertext expansion ratio of $O(m)$ (we encrypt degree-$(m-1)$ polynomials using $m \times m$ matrices). We do not know if more economical embeddings are possible.

Moving to integer multiplication, an obvious way of multiplying two m-bit integers is to just set the plaintext space to $\mathbb{Z}_p$ for some $p \geq 2^{2m}$, but working with such large plaintext space may be inconvenient. We thus seek a method for implementing large integer multiplication with a small input space. One possibility is to use the same technique as we did for polynomials, viewing the integer with binary representation $a = \sum a_i 2^i$ as a binary polynomial $\hat{a}(x)$ evaluated at $x = 2$. Given two integers a, b, we encrypt the binary coefficients of the corresponding polynomials $\hat{a}, \hat{b}$ *over plaintext space* $\mathbb{Z}_p$ *for some* $p \geq m$. Reading out the coefficients of the product polynomial, we then compute $a \cdot b = (\hat{a} \cdot \hat{b})(2)$ over the integers.

This solution is not private however, it leaks more information about a, b than just their integer product. One approach for making it private is to add random elements $r_i \in \mathbb{Z}_p$ to the first row and column of the product matrix such that $\sum_i 2^i r_i = 0 \pmod p$. This will make it possible for the secret key holder to recover $a \cdot b \pmod p$. Repeating it several times with different p's, we can then use Chinese remaindering to recover $a \cdot b$ completely.

5.4 Two-Out-of-Two Decryption

We point out a peculiar property of our cryptosystem, which so far we were not able to find applications for. Namely, if we have encryptions of two matrices under *two different public keys*, we can multiply these two ciphertexts, thus obtaining an "ciphertext" corresponding to the product of the two plaintext matrices". This "ciphertext" can then be decrypted by pulling together the two secret keys.

In more details, suppose that we have two public keys $\mathbf{A}_1, \mathbf{A}_2$ and the corresponding two secret keys $\mathbf{T}_1, \mathbf{T}_2$, with both pairs defined modulo the same prime number q. (We also assume for simplicity that both pairs use the same parameters n and m, but this assumption is not really needed). Then, given two ciphertexts

$$\mathbf{C}_1 = \mathbf{A}_1\mathbf{S}_1 + 2\mathbf{X}_1 + \mathbf{B}_1 \text{ and } \mathbf{C}_2 = \mathbf{A}_2\mathbf{S}_2 + 2\mathbf{X}_2 + \mathbf{B}_2,$$

we can compute the "product ciphertext" $\mathbf{C} = \mathbf{C}_1\mathbf{C}_2^t \pmod q$, corresponding to the plaintext $\mathbf{B}_1\mathbf{B}_2^t \pmod 2$. This plaintext can be recovered if we know both $\mathbf{T}_1$ and $\mathbf{T}_2$, by setting

$$B \leftarrow \mathbf{T}_1^{-1} \cdot (\mathbf{T}_1\mathbf{C}\mathbf{T}_2^t \bmod q) \cdot (\mathbf{T}_2^t)^{-1} \bmod 2$$

5.5 Identity-Based and Leakage-Resilient BGN-Type Encryption

Next we show how to extend the one-multiplication homomorphism beyond just standard public-key encryption, to get more "advanced features" such as identity-based encryption and leakage-resilience. This follows from the simple observations that the "dual Regev cryptosystem" from [10] (with a different input encoding) can be viewed as a special case of our encryption scheme (for a particular form of matrices), and hence it supports the same homomorphic operations. IBE (in the random-oracle model) follows directly since Gentry et al. showed in [10] how to derive dual Regev keys from a master key, and leakage-resilience follows since Dodis et al. proved in [6] that the dual Regev cryptosystem is leakage resilient.

Recall the "dual Regev cryptosystem" from [10]: The public key is a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, and the secret key is one short vector in the dual, namely a short $\boldsymbol{u} \in \mathbb{Z}_q^m$ such that $\boldsymbol{u}\mathbf{A} = 0 \pmod q$. Moreover, the last entry in $\boldsymbol{u}$ is always -1.

In the cryptosystem as described in [10], a bit b is encrypted by choosing a uniform vector $\boldsymbol{s} \in \mathbb{Z}_q^n$ and a small error vector $\boldsymbol{x} \in \mathbb{Z}_q^m$, and then encoding the bit b in the "most significant bit" of one entry of the ciphertext vector, namely $\boldsymbol{c} \leftarrow \mathbf{A}\boldsymbol{s} + \boldsymbol{x} + \langle 0 \dots 0\ 1 \rangle^t \cdot \lceil q/2 \rceil \bmod q$. But to get homomorphism, we want to encode the input in the least significant bit, setting instead $\boldsymbol{c} \leftarrow \mathbf{A}\boldsymbol{s} + 2\boldsymbol{x} + \langle 0 \dots 0\ b \rangle^t \bmod q$. With this input encoding, one can view the dual Regev cryptosystem as a special case of our cryptosystem, where the public key is the same matrix $\mathbf{A}$, but the secret key is not a full rank matrix but instead a rank-1 matrix. The matrices $\mathbf{T}, \mathbf{S}, \mathbf{X}, \mathbf{B}$ are defined as

$$\mathbf{T} = \begin{pmatrix} -\boldsymbol{u}- \\ 0 \end{pmatrix}, \ \mathbf{S} = \begin{pmatrix} 0 & \boldsymbol{s} \end{pmatrix}, \ \mathbf{X} = \begin{pmatrix} 0 & \boldsymbol{x} \end{pmatrix}, \ \mathbf{B} = \begin{pmatrix} 0 & \\ & b \end{pmatrix}.$$

(That is, all but the top row of T are zero, all but the rightmost columns of $\mathbf{S}, \mathbf{X}$ are zero, and all but the bottom-right element of $\mathbf{B}$ are zero.)

Although this choice does not follow our input distribution for these matrices, it is nonetheless easy to show that semantic security follows from LWE. Since the key is just the dual Regev key, then the same proof as in [6] shows that it remains secure even in the face of partial leakage of the secret key. Also, it was shown in [10] how this secret key can be computed from a master secret key in an identity-based setting (in the random-oracle model).

With these choices, most of the "ciphertext matrix" is zero, so all we need to output as the ciphertext is indeed the one vector $\boldsymbol{c} \leftarrow \mathbf{A}\boldsymbol{s} + 2\boldsymbol{x} + \langle 0 \dots 0\ b \rangle^t \bmod q$, which implicitly encodes the matrix $\mathbf{C} = \begin{pmatrix} 0 & \boldsymbol{c} \end{pmatrix}$. The homomorphic operations are then applied to the implicit matrices, namely addition is just element-wise addition modulo q and multiplication of two vectors is an outer-product operation.

To decrypt a ciphertext matrix, we multiply if from left and right by the secret key vector $\boldsymbol{c}$, reducing the result first modulo q and then modulo 2. Due to the special form of the plaintext matrix $\mathbf{B}$, this is the same as multiplying by $\mathbf{T}$ on the left and and $\mathbf{T}^t$ on the right, and then taking only the bottom right element of the result.

Although the matrix $\mathbf{T}$ no longer has an inverse, we can still recover the hidden bit b. This is done simply by setting $b \leftarrow \boldsymbol{u}\mathbf{C}\boldsymbol{u}^t \bmod q \bmod 2$, without needing to multiply by the inverse, since we have

$$(\boldsymbol{u}\mathbf{C}\boldsymbol{u}^t \bmod q) \quad = \quad \boldsymbol{u} \begin{pmatrix} 0 & \\ & b \end{pmatrix} \boldsymbol{u}^t \quad = \quad b \cdot u_m^2 \pmod 2$$

Recalling that $u_m = -1$ in the dual Regev cryptosystem, this procedure indeed gives the right answer.

Acknowledgments. We thank the anonymous Eurocrypt 2010 reviewers for their valuable comments.

References

1. Aguilar Melchor, C., Castagnos, G., Gaborit, P.: Lattice-based homomorphic encryption of vector spaces. In: IEEE International Symposium on Information Theory, ISIT 2008, pp. 1858–1862 (2008)
2. Aguilar Melchor, C., Gaborit, P., Javier, H.: Additive Homomorphic Encryption with t-Operand Multiplications. Technical Report 2008/378, IACR ePrint archive (2008), http://eprint.iacr.org/2008/378/
3. 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)
4. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. In: STACS, pp. 75–86 (2009)
5. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts, pp. 325–341 (2005)
6. 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)
7. Feller, W.: An Introduction to Probability Theory and Its Applications, vol. 1. Wiley, Chichester (1968)
8. Gentry, C.: A fully homomorphic encryption scheme. PhD thesis, Stanford University (2009), http://crypto.stanford.edu/craig
9. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC 2009, pp. 169–178. ACM, New York (2009)
10. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC, pp. 197–206 (2008)
11. Kawachi, A., Tanaka, K., Xagawa, K.: Multi-bit Cryptosystems Based on Lattice Problems. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 315–329. Springer, Heidelberg (2007)
12. Lindell, Y., Pinkas, B.: A proof of security of yao's protocol for two-party computation. J. Cryptology 22(2) (2009)
13. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem. In: STOC 2009, pp. 333–342. ACM, New York (2009)
14. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM 56(6) (2009); Preliminiary version in STOC 2005
15. Sander, T., Young, A., Yung, M.: Non-interactive CryptoComputing for NC1. In: 40th Annual Symposium on Foundations of Computer Science, pp. 554–567. IEEE, Los Alamitos (1999)
16. Yao, A.C.: Protocols for secure computations (extended abstract). In: 23rd Annual Symposium on Foundations of Computer Science – FOCS 1982, pp. 160–164. IEEE, Los Alamitos (1982)

A The Aguilar Melchor-Gaborit-Herranz Transformation

Below is a brief description of the Aguilar Melchor-Gaborit-Herranz transformation [2] of an additively-homomorphic cryptosystem to one that supports evaluation of d-degree polynomials with upto m terms (where d, m are parameters). Here we only describe the basic approach, exemplified for the special case of $d = 3$ (since indexing becomes unwieldy for larger d). Aguilar Melchor et al. also describe in [2] some extensions and optimizations.

To evaluate d-degree binary polynomials with upto m terms, we need an encryption scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathbb{Z}_p$ for $p \geq m+1$, that satisfies the following properties:

- The ciphertext is a vector of integers in $[0, q-1]$ (for some parameter q), which we denote by Greek letters, $\mathsf{Enc}(a) = \langle \alpha[1], \dots, \alpha[n] \rangle \in \mathbb{Z}_q^n$. (We identify Z_q with the set of integers in $[0, q-1]$.) Denote the bit-length of the parameter q by $t \overset{\text{def}}{=} \lceil \log(q+1) \rceil$, so the total size of the ciphertext is nt bits. (Note that to support message space $\mathbb{Z}_p$ for $p > m$, we need $nt \geq \Omega(\kappa + \log m)$ for security parameter κ.)
- $\mathcal{E}$ is additively homomorphic, via mod-q addition of the ciphertext vectors. Specifically, what we need is that for any $m' \leq m$ plaintext bits $a_1, \dots, a_{m'} \in \{0, 1\}$ and their encryption $\boldsymbol{\alpha_j} \leftarrow \mathsf{Enc}(a_j)$, the vector

$$\boldsymbol{\alpha} = \sum_j \boldsymbol{\alpha_j} \bmod q$$

  whole elements are integers in $[0, q-1]$, is decrypted (with probability one) to the integer $\sum_j a_j$. (Note that since $m < p$ and all the a_j's are bits, then the sum of the a_j's is less than p, and hence addition modulo p is the same as the sum over the integers.)

Consider now a vector of m triples of plaintext bits $\langle (a_1, b_1, c_1), \dots, (a_m, b_m, c_m) \rangle \in (\{0,1\}^3)^m$, and their encryption $\boldsymbol{\alpha_j} \leftarrow \mathsf{Enc}(a_j)$, $\boldsymbol{\beta_j} \leftarrow \mathsf{Enc}(b_j)$, $\boldsymbol{\gamma_j} \leftarrow \mathsf{Enc}(c_j)$. We now show how to generate a "ciphertext" of size $(nt)^3$ that can be decrypted to the bit $\sum_{j=1}^m a_j b_j c_j \bmod 2$. (More generally, for degree-d polynomials the size of the ciphertext is at most $(nt)^d$. If the underlying scheme has $nt = O(\kappa + \log m)$ then this would give ciphertext size of $O(\kappa^d + \log^d m)$ for degree-d, m-term polynomials with security parameter κ.)

$\underline{\textsc{PolyEval}\Big(\langle (\boldsymbol{\alpha_j}, \boldsymbol{\beta_j}, \boldsymbol{\gamma_j}) \rangle_{j=1,\dots,m}\Big)}$:

For $j = 1, \dots, m$, denote the integers in the ciphertext vectors $\boldsymbol{\beta_j}$, $\boldsymbol{\gamma_j}$ by

$$\boldsymbol{\beta_j} = \langle \beta_j[1], \dots, \beta_j[n] \rangle, \quad \boldsymbol{\gamma_j} = \langle \gamma_j[1], \dots, \gamma_j[n] \rangle$$

Recall that these are all non-negative t-bit integers, and we denote their bit representations by

$$\beta_j[i] = \sum_{k=0}^{t-1} 2^k \cdot \beta_j^{(k)}[i], \quad \gamma_j[i'] = \sum_{k'=0}^{t-1} 2^{k'} \cdot \gamma_j^{(k')}[i'] \quad \text{(integer addition)}$$

where each $\beta_j^{(k)}[i]$ and $\gamma_j^{(k')}[i']$ is a bit. The "compound ciphertext" consists of the $(nt)^2$ vectors

$$\boldsymbol{\delta}^{(i,k,i',k')} \overset{\text{def}}{=} \sum_{j=1}^m \underbrace{\boldsymbol{\alpha_j}}_{\text{ctxt}} \cdot \underbrace{\beta_j^{(k)}[i]}_{\text{bit}} \cdot \underbrace{\gamma_j^{(k')}[i']}_{\text{bit}} \bmod q \qquad (1)$$

In other words, each vector $\boldsymbol{\delta}^{(i,k,i',k')}$ is computed as a subset-sum (over Z_q) of the m ciphertext vectors $\boldsymbol{\alpha_j}$. Since we have $(nt)^2$ such vectors, the total size of the compound ciphertext is $(nt)^3$, as claimed.

<u>$\textsc{Decrypt}\Big(\Big\{\boldsymbol{\delta}^{(i,k,i',k')} : \ k, k' \in [0, t-1], i, i' \in [1, n]\Big\}\Big)$:</u>

1. For all k, k', i, i' decrypt $\boldsymbol{\delta}^{(i,k,i',k')}$ to get an integer $\lambda^{(i,k,i',k')} \leftarrow \mathsf{Dec}\Big(\boldsymbol{\delta}^{(i,k,i',k')}\Big)$. Due to the additive homomorphism of the underlying scheme, we have that

$$\lambda^{(i,k,i',k')} \quad = \quad \sum_j \underbrace{a_j}_{\text{bit}} \cdot \underbrace{\beta_j^{(k)}[i]}_{\text{bit}} \cdot \underbrace{\gamma_j^{(k')}[i']}_{\text{bit}} \qquad \text{(equality over the integers)}$$

   Note that this equality is over the integers since this is a sum of m bits, and hence must be less than p.

2. For all k', i, i' compute the integer $\lambda^{(i',k')}[i] \leftarrow \sum_{k=0}^{t-1} 2^k \cdot \lambda^{(i,k,i',k')} \bmod q$. By construction we have $\lambda^{(i',k')}[i] \in Z_q$, and by changing the order of summation we see that

$$\begin{aligned}\lambda^{(i',k')}[i] &= \sum_{k=0}^{t-1} 2^k \cdot \sum_j a_j \cdot \beta_j^{(k)}[i] \cdot \gamma_j^{(k')}[i'] \qquad (2)\\ &= \sum_j a_j \cdot \gamma_j^{(k')}[i'] \cdot \sum_{k=0}^{t-1} 2^k \cdot \beta_j^{(k)}[i] = \sum_j \underbrace{a_j}_{\text{bit}} \cdot \underbrace{\gamma_j^{(k')}[i']}_{\text{bit}} \cdot \underbrace{\beta_j[i]}_{\text{integer}} \pmod q\end{aligned}$$

3. For all k', i' denote

$$\boldsymbol{\lambda}^{(i',k')} \overset{\text{def}}{=} \Big\langle \lambda^{(i',k')}[1], \ldots, \lambda^{(i',k')}[n] \Big\rangle \quad = \quad \sum_j \underbrace{a_j}_{\text{bit}} \cdot \underbrace{\gamma_j^{(k')}[i']}_{\text{bit}} \cdot \underbrace{\boldsymbol{\beta_j}}_{\text{ctxt}} \pmod q$$

   Again, each $\boldsymbol{\lambda}^{(i',k')}$ is equal to a subset-sum over Z_q of the $\boldsymbol{\beta_j}$ ciphertext vectors.

4. For all k', i' decrypt $\boldsymbol{\lambda}^{(i',k')}$ to get an integer $\mu^{(i',k')} \leftarrow \mathsf{Dec}\Big(\boldsymbol{\lambda}^{(i',k')}\Big)$. As before, due to the additive homomorphism of the underlying scheme we have that

$$\mu^{(i',k')} \quad = \quad \sum_j \underbrace{a_j}_{\text{bit}} \cdot \underbrace{b_j}_{\text{bit}} \cdot \underbrace{\gamma_j^{(k')}[i']}_{\text{bit}} \qquad \text{(equality over the integers)}$$

5. For all i' compute the integer $\mu[i'] \leftarrow \sum_{k'=0}^{t-1} 2^{k'} \cdot \mu^{(i',k')} \bmod q$. Again, we have $\mu[i'] \in Z_q$ and by changing the order of summation we see that

$$\mu[i'] = \sum_{k'=0}^{t-1} 2^{k'} \cdot \sum_j a_j \cdot b_j \cdot \gamma_j^{(k')}[i'] \ = \ \sum_j a_j \cdot b_j \cdot \sum_{k'=0}^{t-1} 2^{k'} \cdot \gamma_j^{(k')}[i'] \ = \ \sum_j \underbrace{a_j}_{\text{bit}} \cdot \underbrace{b_j}_{\text{bit}} \cdot \underbrace{\gamma_j[i']}_{\text{integer}}$$

6. Denote $\boldsymbol{\mu} \overset{\text{def}}{=} \langle \mu[1], \ldots, \mu[n] \rangle = \sum_j \underbrace{a_j}_{\text{bit}} \cdot \underbrace{b_j}_{\text{bit}} \cdot \underbrace{\boldsymbol{\gamma_j}}_{\text{ctxt}}$
7. Decrypt $\boldsymbol{\mu}$ to get the integer $\nu \leftarrow \mathsf{Dec}(\boldsymbol{\mu})$, and once again due to the additive homomorphism we have the equality $\nu = \sum_j a_j b_j c_j$ holding over the integers. Finally output (ν mod 2) as the decrypted bit.

Bonsai Trees, or How to Delegate a Lattice Basis

David Cash[1,⋆], Dennis Hofheinz[2,⋆⋆], Eike Kiltz[3,⋆⋆⋆], and Chris Peikert[4,†]

[1] University of California, San Diego
cdc@ucsd.edu
[2] Karlsruhe Institute of Technology
Dennis.Hofheinz@kit.edu
[3] Cryptology & Information Security Group, CWI, Amsterdam, The Netherlands
kiltz@cwi.nl
[4] Georgia Institute of Technology
cpeikert@cc.gatech.edu

Abstract. We introduce a new *lattice-based* cryptographic structure called a *bonsai tree*, and use it to resolve some important open problems in the area. Applications of bonsai trees include:

- An efficient, stateless 'hash-and-sign' signature scheme in the *standard model* (i.e., no random oracles), and
- The first *hierarchical* identity-based encryption (HIBE) scheme (also in the standard model) that does not rely on bilinear pairings.

Interestingly, the abstract properties of bonsai trees seem to have no known realization in conventional number-theoretic cryptography.

1 Introduction

Lattice-based cryptographic schemes have undergone rapid development in recent years, and are attractive due to their low asymptotic complexity and potential resistance to quantum-computing attacks. One notable recent work in this area is due to Gentry, Peikert, and Vaikuntanathan [25], who constructed an efficient 'hash-and-sign' signature scheme and an identity-based encryption (IBE) scheme. (IBE is a powerful cryptographic primitive in which *any string* can serve as a public key [53].)

Abstractly, the GPV schemes are structurally quite similar to Rabin/Rabin-Williams signatures [50] (based on integer factorization) and the Cocks/Boneh-Gentry-Hamburg IBEs [18, 13] (based on the quadratic residuosity problem), in that they all employ a so-called "preimage sampleable" trapdoor function as a basic primitive. As a result,

[⋆] Part of work performed while at Georgia Institute of Technology.

[⋆⋆] Part of work performed while at CWI and supported by an NWO Veni grant.

[⋆⋆⋆] Supported by the research program Sentinels.

[†] This material is based upon work supported by the National Science Foundation under Grants CNS-0716786 and CNS-0749931, and by the US Department of Homeland Security under Contract Number HSHQDC-07-C-00006. 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 National Science Foundation or the US Department of Homeland Security.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 523–552, 2010.

they have so far required the random oracle model (or similar heuristics) for their security analysis. This is both a theoretical drawback and also a practical concern (see, e.g., [35]), so avoiding such heuristics is an important goal.

Another intriguing open question is whether any of these IBE schemes can be extended to deliver richer levels of functionality, as has been done in pairing-based cryptography since the work of Boneh and Franklin [12]. For example, the more general notion of *hierarchical* IBE [33, 26] permits multiple levels of secret-key authorities. This notion is more appropriate than standard IBE for large organizations, can isolate damage in the case of secret-key exposure, and has further applications such as forward-secure encryption [16] and broadcast encryption [21, 58].

1.1 Our Results

We put forward a new cryptographic notion called a *bonsai tree*, and give a realization based on hard lattice problems. (Section 1.2 gives an intuitive overview of bonsai trees, and Section 1.4 discusses their relation to other primitives and techniques.) We then show that bonsai trees resolve some central open questions in lattice-based cryptography: to summarize, they remove the need for random oracles in many important applications, and facilitate delegation for purposes such as hierarchical IBE.

Our first application of bonsai trees is an efficient, stateless signature scheme that is secure in the *standard model* (no random oracles) under conventional lattice assumptions. Our scheme has a 'hash-and-sign' flavor that does not use the key-refresh/authentication-tree paradigm of many prior constructions (both generic [28, 43] and specialized to lattice assumptions [37]), and in particular it does not require the signer to keep any state. (Statelessness is a crucial property in many real-world scenarios, where distinct systems may sign relative to the same public key.) In our scheme, the verification key, signature length, and verification time are all an $O(k)$ factor larger than in the random-oracle scheme of [25], where k is the output length of a *chameleon* hash function, and the $O(\cdot)$ notation hides only a 1 or 2 factor. The signing algorithm is essentially as efficient as the one from [25].[1] The underlying hard problem is the standard *short integer solution* (SIS) problem dating back to the seminal work of Ajtai [5], which is known to be as hard as several worst-case approximation problems on lattices (see also [41, 25]). Via SIS, the security of our signature scheme rests upon the hardness of approximating worst-case problems on n-dimensional lattices to within an $\tilde{O}(\sqrt{k} \cdot n^{3/2})$ factor; this is only a $\sqrt{k}$ factor looser than that of [25].

Our second application is a collection of various *hierarchical* identity-based encryption (HIBE) schemes, which are the first HIBEs that do not rely on bilinear pairings. Our main scheme works in the standard model, also making it the first non-pairing-based IBE (hierarchical or not) that does not use random oracles or qualitatively similar heuristics. The underlying hard problem is the standard *learning with errors* (LWE) problem as defined by Regev, which may be seen as the 'dual' of SIS and is also as hard as certain worst-case lattice problems [51, 45]; LWE is also the foundation for the plain IBE of [25], among many other recent cryptographic schemes.

[1] Our signing algorithm performs about k *forward* computations of a trapdoor function, plus one inversion (which dominates the running time).

Additionally, our HIBE is *anonymous* across all levels of the hierarchy, i.e., a ciphertext conceals (computationally) the identity to which is was encrypted. Anonymity is a useful property in many applications, such as fully private communication [7] and searching on encrypted data [11, 1]. While there are a few anonymous (non-hierarchical) IBEs [12, 20, 13, 25], only one other HIBE is known to be anonymous [15].

1.2 Overview of Bonsai Trees and Applications

The ancient art of bonsai is centered around a *tree* and the selective *control* thereof by an arborist, the tree's cultivator and caretaker. By combining natural, *undirected* growth with *controlled* propagation techniques such as wiring and pruning, arborists cultivate trees according to a variety of aesthetic forms.

Similarly, cryptographic bonsai is not so much a precise definition as a collection of principles and techniques, which can be employed in a variety of ways. (The informal description here is developed technically in Section 3.) The first principle is the tree itself, which in our setting is a *hierarchy of trapdoor functions* having certain properties. The arborist can be any of several entities in the system — e.g., the signer in a signature scheme or a simulator in a security proof — and it can exploit both kinds of growth, undirected and controlled. Briefly stated, *undirected* growth of a branch means that the arborist has no privileged information about the associated function, whereas the arborist *controls* a branch if it knows a trapdoor for the function. Moreover, control automatically extends down the hierarchy, i.e., knowing a trapdoor for a parent function implies knowing a trapdoor for any of its children.

In our concrete lattice-based instantiation, the functions in the tree are indexed by a hierarchy of public lattices chosen at random from a certain 'hard' family (i.e., one having a connection to worst-case problems). The lattices may be specified by a variety of means, e.g., a public key, interaction via a protocol, a random oracle, etc. Their key property is that they naturally form a hierarchy as follows: every lattice in the tree (excepting the root) is a *higher-dimensional superlattice* of its parent. Specifically, a parent lattice in $\mathbb{R}^m$ is simply the restriction of its child(ren) in $\mathbb{R}^{m'}$ (where $m' > m$) to the first m dimensions. As we shall see shortly, this hierarchical relationship means that a parent lattice naturally 'subsumes' its children (and more generally, all its descendants).

Undirected growth in our realization is technically straightforward, emerging naturally from the underlying hard average-case lattice problems (SIS and LWE). This growth is useful primarily for letting a simulator embed a challenge problem into one or more branches of the tree (but it may have other uses as well).

To explain *controlled growth*, we first need a small amount of technical background. As explored in prior works on lattice-based cryptography (e.g., [27, 30, 29, 25, 49, 45]), a lattice has a 'master trapdoor' in the form of a *short basis*, i.e., a basis made up of relatively short lattice vectors. Knowledge of such a trapdoor makes it easy to solve a host of seemingly hard problems relative to the lattice, such as decoding within a bounded distance, or randomly sampling short lattice vectors. The reader may view a short basis for a lattice as roughly analogous to the factorization of an integer, though we emphasize that there are in general *many distinct* short bases that convey roughly 'equal power' with respect to the lattice.

In light of the above, we say that an arborist *controls* a branch of a bonsai tree if it knows a short basis for the associated lattice. The hierarchy of lattices is specially designed so that any short basis of a parent lattice can be easily *extended* to a short basis of any higher-dimensional child lattice, with no loss in quality. This means that control of a branch implicitly comes with control over all its offshoots. In a typical application, the privileged entity in the system (e.g., the signer in a signature scheme) will know a short basis for the root lattice, thus giving it control over the entire tree. Other entities, such as an attacker, will generally have less power, though in some applications they might even be given control over entire subtrees.

So far, we have deliberately avoided the question of *how* an arborist comes to control a (sub)tree by acquiring a short basis for the associated lattice. A similar issue arises in other recent cryptographic schemes [25, 49, 45], but in a simpler setting involving only a single lattice and short basis (not a hierarchy). In these schemes, one directly applies a special algorithm, originally conceived by Ajtai [4] and recently improved by Alwen and Peikert [6], which generates a hard random lattice together with a short basis 'from scratch.' At first glance, the algorithms of [4, 6] seem useful only for controlling a new tree entirely by its root, which is not helpful if we need finer-grained control. Fortunately, we observe that the same technique used for extending an already-controlled lattice also allows us to 'graft' a solitary controlled lattice onto an *uncontrolled* branch.[2]

This whole collection of techniques, therefore, allows an arborist to achieve a primary bonsai aesthetic: a carefully controlled tree that nonetheless gives the appearance of having grown without any outside intervention. As we shall see next, bonsai techniques can reduce the construction of complex cryptographic schemes to the design of simple *combinatorial* games between an arborist and an adversary.

Application 1: Hash-and-Sign without Random Oracles. Our end goal is a signature scheme that meets the *de facto* notion of security, namely, existential unforgeability under adaptive chosen-message attack [28]. By a standard, efficient transformation using *chameleon hashes* [34] (which have efficient realizations under conventional lattice assumptions, as we show), it suffices to construct a *weakly secure* scheme, namely, one that is existentially unforgeable under a static attack in which the adversary non-adaptively makes all its queries before seeing the public key.

Our weakly secure scheme signs messages of length k, the output length of the chameleon hash. The public key represents a *binary* bonsai tree T of depth k in a compact way, which we describe in a moment. The secret key is a short basis for the lattice Λ_ε at the root of the tree, which gives the signer control over all of T. To sign a string $\mu \in \{0,1\}^k$ (which is the chameleon hash of the 'true' message m), the signer first derives the lattice Λ_μ from T by walking the root-to-leaf path specified by μ. The signature is simply a short nonzero vector $\mathbf{v} \in \Lambda_\mu$, chosen at random from the 'canonical' Gaussian distribution (which can be sampled efficiently using the signer's control over Λ_μ).

[2] It is worth noting that in [4, 6], even the simple goal of generating a solitary lattice together with a short basis actually proceeds in two steps: first start with a sufficient amount of random undirected growth, then produce a single controlled offshoot by way of a certain linear algebraic technique. Fittingly, this is analogous to the traditional bonsai practice of growing a new specimen from a cutting of an existing tree, which is generally preferred to growing a new plant 'from scratch' with seeds.

A verifier can check the signature $\mathbf{v}$ simply by deriving Λ_μ itself from the public key, and checking that $\mathbf{v}$ is a sufficiently short nonzero vector in Λ_μ.

The bonsai tree T is represented compactly by the public key in the following way. First, the root lattice Λ_ε is specified completely. Then, for each level $i = 0, \dots, k-1$, the public key includes two blocks of randomness that specify how a parent lattice at level i branches into its two child lattices. We emphasize that all nodes at a given depth use the *same* two blocks of randomness to derive their children.

The proof of security is at heart a combinatorial game on the tree between the simulator $\mathcal{S}$ and forger $\mathcal{F}$, which goes roughly as follows. The forger gives the simulator a set $M = \{\mu_1, \dots, \mu_Q\}$ of messages, and $\mathcal{S}$ needs to cultivate a bonsai tree (represented by pk) so that it controls some set of subtrees that *cover* all of M, yet is unlikely to control the leaf of whatever arbitrary message $\mu^* \notin M$ that $\mathcal{F}$ eventually produces as a forgery. If the latter condition happens to hold true, then the forger has found a short nonzero vector in an uncontrolled lattice, in violation of the underlying assumption.

To satisfy these conflicting constraints, $\mathcal{S}$ colors red all the edges on the root-to-leaf paths of the messages in M, and lets all the other edges implicitly be colored blue. The result is a forest of at most $Q \cdot k$ distinct blue subtrees $\{B_\ell\}$, each growing off of some red path by a single blue edge. The simulator chooses one of these subtrees B_ℓ uniformly at random (without regard to its size), guessing that the eventual forgery will lie in B_ℓ. It then cultivates a bonsai tree so that all the growth on the path up to and throughout B_ℓ is *undirected* (by embedding its given challenge instance as usual), while all the remaining growth in $T \setminus B_\ell$ is controlled. This goal can be achieved within the confines of the public key by controlling one branch at each level leading up to B_ℓ (namely, the branch growing off of the path to B_ℓ), and none thereafter.

Application 2: Hierarchical Identity-Based Encryption. Bonsai trees also provide a very natural and flexible approach for realizing HIBE. For simplicity, consider an authority hierarchy that is a *binary tree*, which suffices for forward-secure encryption and general HIBE itself [16]. The master public key of the scheme describes a binary bonsai tree, which mirrors the authority hierarchy. The root authority starts out by controlling the entire tree, i.e., it knows a trapdoor short basis for the lattice at the root. Each authority is entitled to control its corresponding branch of the tree. Any entity in the hierarchy can delegate control over an offshoot branch to the corresponding sub-authority, simply by computing and revealing a short basis of the associated child lattice. In this framework, encryption and decryption algorithms based on the LWE problem are standard.

For the security proof, the simulator again prepares a bonsai tree so that it controls certain branches (which should cover the adversary's queries), while allowing the undirected growth of others (corresponding to the adversary's target identity). This can be accomplished in a few ways, with different advantages and drawbacks in terms of the security notion achieved and the tightness of the reduction. One notion is security against a *selective-identity* attack, where the adversary must declare its target identity before seeing the public key, but may adaptively query secret keys afterward. In this model, the simulator can cultivate a bonsai tree whose growth toward the (known) target identity is undirected, while controlling each branch off of that path; this setup makes it easy for the simulator to answer any legal secret-key query.

A stronger notion is a *fully adaptive* attack, where the adversary may choose its target identity after making its secret-key queries. There are generic combinatorial techniques for converting selective-identity-secure (H)IBE schemes into fully secure ones; we show how to apply and optimize these techniques to our HIBE. First, we use the techniques of Boneh and Boyen [8] construct a fully secure HIBE scheme in the random oracle model. The basic idea is to hash all identities; this way, the target identity can be dynamically embedded as the answer to a random oracle query. Secondly, we demonstrate that other tools of Boneh and Boyen [9] can be adapted to our setting to yield a fully secure HIBE scheme *without* random oracles. This works by hashing identities to branches of a bonsai tree, where a probabilistic argument guarantees that any given identity hashes to a controlled branch with a certain probability. We can adjust this probability in the right way, so that with non-negligible probability, all queried identities hash to controlled branches, while the target identity hashes to an uncontrolled branch. In our probabilistic argument, we employ *admissible hash functions (AHFs)*, as introduced by [9]. However, as we will explain in Section 5.4, their original AHF definition and proof strategy do not take into consideration the statistical dependence of certain crucial events. We circumvent this with a different AHF definition and a different proof.

Based on the above description, the reader may still wonder whether secret-key delegation is actually secure, i.e., whether the real and simulated bases are drawn from the same probability distribution. In fact, they may not be! For example, under the most straightforward method of extending a basis, the child basis actually contains the parent basis as a submatrix, so it is clearly insecure to reveal the child. We address this issue with an additional bonsai principle of *randomizing control*, using the 'oblivious' Gaussian sampling algorithm of [25]. This produces a new basis under a 'canonical' distribution, regardless of the original input basis, which ensures that the real system and simulation coincide. The randomization increases the length of the basis by a small factor — which accumulates geometrically with each delegation from parent to child — but for reasonable depths, the resulting bases are still short enough to be useful when all the parameters are set appropriately. (See Section 1.3 for more details.)

For achieving security under chosen-ciphertext attacks (CCA security), a transformation due to Boneh, Canetti, Halevi, and Katz [10] gives a CCA-secure HIBE for depth d from any chosen plaintext-secure HIBE for depth $d + 1$. Alternatively, we observe that the public and secret keys in our HIBE scheme are of exactly the same 'type' as those in the recent CCA-secure cryptosystem of [45], so we can simply plug that scheme into our bonsai tree/HIBE framework. Interestingly, the two approaches result in essentially identical schemes.

Variations. This paper focuses almost entirely on bonsai trees that are related, via worst- to average-case reductions, to *general* lattices. Probably the main drawback is that the resulting public and secret keys are rather large. For example, the public key in our signature scheme is larger by a factor of k (the output length of a chameleon hash function) than that of its random-oracle analogue [25], which is already at least quadratic in the security parameter. Fortunately, the principles of bonsai trees may be applied equally well using analogous hard problems and tools for *cyclic/ideal lattices* (developed in, e.g., [39, 47, 36, 48, 55, 38]). This approach can 'miniaturize' the bonsai trees and most of their associated operations by about a linear factor in the security parameter. The resulting schemes are still not suitable for practice, but their asymptotic behavior is attractive.

1.3 Complexity and Open Problems

Here we discuss some additional quantitative details of our schemes, and describe some areas for further research.

Several important quantities in our bonsai tree constructions and applications depend upon the depth of the tree. The dimension of a lattice in the tree grows linearly with its depth, and the size of the trapdoor basis grows roughly quadratically with the dimension.

Accordingly, in our HIBE schemes, the dimension of a ciphertext vector grows (at least) linearly with the depth of the identity to which it is encrypted. Moreover, the (Euclidean) length of an user's trapdoor basis increases *geometrically* with its depth in the tree (more precisely, with the length of the delegation chain), due to the basis randomization that is performed with each delegation. To ensure correct decryption, the inverse noise parameter $1/\alpha$ in the associated LWE problem, and hence the approximation factor of the underlying worst-case lattice problems, must grow with the basis length. In particular, a hierarchy of depth d corresponds (roughly) to an $n^{d/2}$ approximation factor for worst-case lattice problems, where n is the dimension. Because lattice problems are conjectured to be hard to approximate to within even subexponential factors, the scheme may remain secure for depths as large as $d = n^c$, where $c < 1$.

Our HIBE scheme that enjoys security under a full *adaptive-identity* attack requires large keys and has a somewhat loose security reduction. In particular, the attack simulation partitions an (implicit) bonsai tree into controlled and undirected branches. This is done in the hope that all user secret key queries refer to controlled branches (so the simulation can derive the corresponding secret key), and that the target identity refers to an undirected branch (so the attack can be converted into one on the LWE problem). This simulation approach (dubbed 'partitioning strategy' in [57]) involves, to a certain extent, guessing the adversary's user secret key and challenge queries. The result is a rather loose security reduction.

In contrast, recent works have achieved tight reductions (and even small keys, in some cases) for pairing-based (H)IBEs under various assumptions [23, 24, 57], and a variant of the GPV IBE (in the random oracle model) also has a tight reduction, but their approaches do not seem to translate to our setting. The issue, essentially, is that our simulator is required to produce a 'master trapdoor' for each queried identity, which makes it difficult to embed the challenge problem into the adversary's view. In prior systems with tight reductions, secret keys are less 'powerful,' so the simulator can embed a challenge while still producing secret keys for any identity (even the targeted one).

A final very interesting (and challenging) question is whether bonsai trees can be instantiated based on other mathematical foundations, e.g., integer factorization. At a very fundamental level, our lattice-based construction seems to rely upon a kind of random self-reducibility that the factorization problem is not known to enjoy.

1.4 Related Techniques and Works

This paper represents a combination of two concurrent and independent works by the first three authors [17] and the fourth author [44], which contained some overlapping results and were accepted to Eurocrypt 2010 under the condition that they be merged.

The abstract properties of bonsai trees appear to have no known realization in conventional number-theoretic cryptography. However, our applications use combinatorial techniques that are similar to those from prior works.

The analysis of our signature scheme is reminiscent of (and influenced by) the recent RSA-based signatures of Hohenberger and Waters [32], but there are some notable structural differences. Most significantly, our scheme does not implicitly 'sign' every prefix of the message as in [32]. Additionally, in contrast with prior hash-and-sign schemes based on RSA [22, 19, 31, 32], our simulator cannot use an 'accumulator' to produce signatures for *exactly* the queried messages, but instead sets up the public key so that it knows enough trapdoors to *cover* all the messages (and potentially many others). This requires the simulator to cultivate a tree whose structure crucially depends on the *global* properties of the entire query set, thus inducing the forest of subtrees as described in Section 1.2.

The structure of our HIBE is also similar, at a combinatorial level at least, to that of prior pairing-based HIBEs, in that the simulator can 'control' certain edges of an (implicit) tree by choosing certain random exponents itself. However, there are no *trapdoor functions* per se in pairing-based constructions; instead, the pairing is used to facilitate secret agreement between the encrypter and decrypter. Our approach, therefore, may be seen as a blending of pairing-based techniques and the trapdoor techniques found in [18, 13, 25].

Following the initial dissemination of our results in [17, 44], several extensions and additional applications have been found. Rückert [52] modified our signature scheme to make it *strongly* unforgeable, and constructed hierarchical identity-based signatures. Agrawal and Boyen [3] constructed a standard-model IBE based on LWE, which is secure under a selective-identity attack; their construction has structure similar to ours, but it does not address delegation, nor does it give an efficient signature scheme. Agrawal, Boneh, and Boyen [2] improved the efficiency of our (H)IBE schemes (under a somewhat stronger LWE assumption), and Boyen [14] used similar techniques to obtain shorter signatures (under a stronger SIS assumption).

2 Preliminaries

2.1 Notation

For a positive integer k, $[k]$ denotes the set $\{1, \ldots, k\}$; $[0]$ is the empty set. We denote the set of integers modulo an integer $q \geq 1$ by $\mathbb{Z}_q$. For a string x over some alphabet, $|x|$ denotes the length of x. We say that a function in n is *negligible*, written $\mathrm{negl}(n)$, if it vanishes faster than the inverse of any polynomial in n.

The *statistical distance* between two distributions $\mathcal{X}$ and $\mathcal{Y}$ (or two random variables having those distributions), viewed as functions over a countable domain D, is defined as $\max_{A \subseteq D} |\mathcal{X}(A) - \mathcal{Y}(A)|$.

Column vectors are named by lower-case bold letters (e.g., $\mathbf{x}$) and matrices by upper-case bold letters (e.g., $\mathbf{X}$). We identify a matrix $\mathbf{X}$ with the ordered set $\{\mathbf{x}_j\}$ of its column vectors, and let $\mathbf{X}\|\mathbf{X}'$ denote the (ordered) concatenation of the sets $\mathbf{X}, \mathbf{X}'$. For a set $\mathbf{X}$ of real vectors, we define $\|\mathbf{X}\| = \max_j \|\mathbf{x}_j\|$, where $\|\cdot\|$ denotes the Euclidean norm.

For any (ordered) set $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_k\} \subset \mathbb{R}^m$ of linearly independent vectors, let $\widetilde{\mathbf{S}} = \{\widetilde{\mathbf{s}_1}, \ldots, \widetilde{\mathbf{s}_k}\}$ denote its *Gram-Schmidt orthogonalization*, defined iteratively as follows: $\widetilde{\mathbf{s}_1} = \mathbf{s}_1$, and for each $i = 2, \ldots, k$, the vector $\widetilde{\mathbf{s}_i}$ is the component of $\mathbf{s}_i$ orthogonal to $\mathrm{span}(\mathbf{s}_1, \ldots, \mathbf{s}_{i-1})$. In matrix notation, there is a unique QR decomposition $\mathbf{S} = \mathbf{QR}$ where the columns of $\mathbf{Q} \in \mathbb{R}^{m \times k}$ are orthonormal (i.e., $\mathbf{Q}^t\mathbf{Q} = \mathbf{I} \in \mathbb{R}^{k \times k}$) and $\mathbf{R} \in \mathbb{R}^{k \times k}$ is right-triangular with positive diagonal entries; the Gram-Schmidt orthogonalization is $\widetilde{\mathbf{S}} = \mathbf{Q} \cdot \mathrm{diag}(r_{1,1}, \ldots, r_{k,k})$. Clearly, $\|\widetilde{\mathbf{s}_i}\| \leq \|\mathbf{s}_i\|$ for all i.

2.2 Cryptographic Definitions

The main cryptographic security parameter through the paper is n, and all algorithms (including the adversary) are implicitly given the security parameter n in unary.

For a (possibly interactive) algorithm $\mathcal{A}$, we define its *distinguishing advantage* between two distributions $\mathcal{X}$ and $\mathcal{Y}$ to be $|\Pr[\mathcal{A}(\mathcal{X}) = 1] - \Pr[\mathcal{A}(\mathcal{Y}) = 1]|$. We use the general notation $\mathbf{Adv}^{\text{atk}}_{\text{SCH}}(\mathcal{A})$ to describe the advantage of an adversary $\mathcal{A}$ mounting an atk attack on a cryptographic scheme SCH, where the definition of advantage is specified as part of the attack. Similarly, we write $\mathbf{Adv}_{\text{PROB}}(\mathcal{A})$ for the advantage of an adversary $\mathcal{A}$ against a computational problem PROB (where again the meaning of advantage is part of the problem definition).

Chameleon hash functions. Chameleon hashing was introduced by Krawczyk and Rabin [34]. For our purposes, we need a slight generalization in the spirit of "preimage sampleable" (trapdoor) functions [25].

A family of chameleon hash functions is a collection $\mathcal{H} = \{h_i : \mathcal{M} \times \mathcal{R} \to \mathcal{Y}\}$ of functions h_i mapping a message $m \in \mathcal{M}$ and randomness $r \in \mathcal{R}$ to a range $\mathcal{Y}$. The randomness space $\mathcal{R}$ is endowed with some efficiently sampleable distribution (which may not be uniform). A function h_i is efficiently computable given its description, and the family has the property that for any $m \in \mathcal{M}$, for $h_i \leftarrow \mathcal{H}$ and $r \leftarrow \mathcal{R}$, the pair $(h_i, h_i(m, r))$ is uniform over $(\mathcal{H}, \mathcal{Y})$ (up to negligible statistical distance). The chameleon property is that a random $h_i \leftarrow \mathcal{H}$ may be generated together with a trapdoor t, such that for any output $y \in \mathcal{Y}$ and message $m \in \mathcal{M}$, it is possible (using t) to efficiently sample $r \in \mathcal{R}$ (under the $\mathcal{R}$'s distribution) conditioned on the requirement that $h_i(m, r) = y$. Finally, the family has the standard collision-resistance property, i.e., given $h_i \leftarrow \mathcal{H}$ it should be hard for an adversary to find distinct $(m, r), (m', r') \in \mathcal{M} \times \mathcal{R}$ such that $h_i(m, r) = h_i(m', r')$.

A realization under conventional lattice assumptions of chameleon hash functions (in the above sense) for $\mathcal{M} = \{0, 1\}^\ell$ is straightforward, using the particular preimage sampleable functions (PSFs) from [25]. Briefly, the chameleon hash function is simply a PSF applied to $m\|r$, which may also be viewed as the sum of two independent PSFs applied to m and r, respectively. We omit the details.

Signatures. A signature scheme SIG for a message space $\mathcal{M}$ is a tuple of PPT algorithms as follows:

- Gen outputs a verification key vk and a signing key sk.
- $\mathsf{Sign}(sk, \mu)$, given a signing key sk and a message $\mu \in \mathcal{M}$, outputs a signature $\sigma \in \{0,1\}^*$.
- $\mathsf{Ver}(vk, \mu, \sigma)$, given a verification key vk, a message μ, and a signature σ, either accepts or rejects.

The correctness requirement is: for all $\mu \in \mathcal{M}$, all possible $(vk, sk) \leftarrow \mathsf{Gen}$ and $\sigma \leftarrow \mathsf{Sign}(sk, \mu)$, we have that $\mathsf{Ver}(vk, \mu, \sigma)$ accepts with overwhelming probability (over all the randomness in the experiment).

We recall two standard notions of security for signatures. The first, existential unforgeability under *static* chosen-message attack, or eu-scma security, is defined as follows: first, the forger $\mathcal{F}$ outputs a list of query messages $\mu_1, \ldots, \mu_Q$ for some Q. Next, $(vk, sk) \leftarrow \mathsf{Gen}$ and $\sigma_i \leftarrow \mathsf{Sign}(sk, \mu_i)$ are generated for each $i \in [Q]$, then vk and σ_i (for each $i \in [Q]$) are given to $\mathcal{F}$. Finally, $\mathcal{F}$ outputs an attempted forgery (μ^*, σ^*). The advantage $\mathcal{A}_{\text{SIG}}^{\text{eu-scma}}(\mathcal{F})$ of $\mathcal{F}$ is the probability that $\mathsf{Ver}(vk, \mu^*, \sigma^*)$ accepts and $\mu^* \neq \mu_i$ for all $i \in [Q]$, taken over all the randomness of the experiment.

Another notion, called existential unforgeability under *adaptive* chosen-message attack, or eu-acma security, is defined similarly, except that $\mathcal{F}$ is first given vk and may adaptively choose the messages μ_i.

Using a family of chameleon hash functions (as defined above), there is a generic construction of eu-acma-secure signatures from eu-scma-secure signatures; e.g., [34]. Furthermore, the construction results in an *online/offline* signature scheme; see [54]. The basic idea behind the construction is that the signer chameleon hashes the message to be signed, then signs the hashed message using the eu-scma-secure scheme (and includes the randomness used in the chameleon hash with the final signature).

Key-Encapsulation Mechanism (KEM). We present all of our encryption schemes in the framework of *key encapsulation*, which simplifies the definitions and leads to more modular constructions. A KEM for keys of length $\ell = \ell(n)$ is a triple of PPT algorithms as follows:

- $\mathsf{KEM.Gen}$ outputs a public key pk and a secret key sk.
- $\mathsf{KEM.Encaps}(pk)$ outputs a key $\kappa \in \{0,1\}^\ell$ and its encapsulation as $\sigma \in \{0,1\}^*$.
- $\mathsf{KEM.Decaps}(sk, \sigma)$ outputs a key κ.

The correctness requirement is: for all possible $(pk, sk) \leftarrow \mathsf{KEM.Gen}$ and $(\kappa, \sigma) \leftarrow \mathsf{KEM.Encaps}(pk)$, we have that $\mathsf{KEM.Decaps}(sk, \sigma)$ outputs κ with all but $\operatorname{negl}(n)$ probability.

In this work we are mainly concerned with indistinguishability under chosen-plaintext attack, or ind-cpa security. The attack is defined as: generate $(pk, sk) \leftarrow \mathsf{KEM.Gen}$, $(\kappa^*, \sigma^*) \leftarrow \mathsf{KEM.Encaps}(pk)$, and $\kappa' \leftarrow \{0,1\}^\ell$ (chosen uniformly and independently of the other values). The advantage $\mathbf{Adv}_{\text{KEM}}^{\text{ind-cpa}}(\mathcal{A})$ of an adversary $\mathcal{A}$ is its distinguishing advantage between (pk, σ^*, κ^*) and (pk, σ^*, κ').

Hierarchical Identity-Based Encryption (HIBE) and Binary Tree Encryption (BTE). In HIBE, identities are strings over some alphabet $\mathcal{ID}$; BTE is the special case of HIBE with identity space $\mathcal{ID} = \{0,1\}$. A HIBE is a tuple of PPT algorithms as follows:

- $\mathsf{Setup}(1^d)$ outputs a master public key mpk and root-level user secret key usk_ε. (In the following, 1^d and mpk are implicit parameters to every algorithm, and every usk_{id} is assumed to include id itself.)
- $\mathsf{Extract}(usk_{id}, id')$, given an user secret key for identity $id \in \mathcal{ID}^{<d}$ that is a prefix of $id' \in \mathcal{ID}^{\leq d}$, outputs a user secret key $usk_{id'}$ for identity id'.
- $\mathsf{Encaps}(id)$ outputs a key $\kappa \in \{0,1\}^\ell$ and its encapsulation as $\sigma \in \{0,1\}^*$, to identity id.
- $\mathsf{Decaps}(usk_{id}, \sigma)$ outputs a key κ.

The correctness requirement is: for any identity $id \in \mathcal{ID}^{\leq d}$, generate $(mpk, usk_\varepsilon) \leftarrow \mathsf{Setup}(1^d)$, usk_{id} via any legal sequence of calls to Extract starting from usk_ε, and $(\kappa, \sigma) \leftarrow \mathsf{Encaps}(id)$. Then $\mathsf{Decaps}(usk_{id}, \sigma)$ should output κ with all but $\operatorname{negl}(n)$ probability (over all the randomness in the experiment).

There are several attack notions for HIBE. We are mainly concerned with the simple notion of indistinguishability under a chosen-plaintext, *selective-identity* attack, or sid-ind-cpa security. The attack is defined as follows: first, the adversary $\mathcal{A}$ is given 1^d and names a target identity $id^* \in \mathcal{ID}^{\leq d}$. Next, $(mpk, msk) \leftarrow \mathsf{Setup}(1^d)$, $(\kappa, \sigma^*) \leftarrow \mathsf{Encaps}(id^*)$, and $\kappa' \leftarrow \{0,1\}^\ell$ are generated. Then $\mathcal{A}$ is given $(mpk, \kappa^*, \sigma^*)$, where κ^* is either κ or κ'. Finally, $\mathcal{A}$ may make extraction queries, i.e., it is given oracle access to $\mathsf{Extract}(sk_\varepsilon, \cdot)$, subject to the constraint that it may not query any identity that is a prefix of (or equal to) the target identity id^*. The advantage $\mathbf{Adv}_{\text{HIBE}}^{\text{sid-ind-cpa}}(\mathcal{A})$ of $\mathcal{A}$ is its distinguishing advantage between the two cases $\kappa^* = \kappa$ and $\kappa^* = \kappa'$.

Another notion is an *adaptive-identity* attack, in which the adversary is first given mpk and oracle access to $\mathsf{Extract}(sk_\varepsilon, \cdot)$ before choosing its target identity id^* (as before, under the constraint that no query identity be a prefix of id^*). Finally, both notions may be extended to *chosen-ciphertext* attacks in the natural way; we omit precise definitions.

2.3 Lattices

In this work, we use m-dimensional *full-rank integer* lattices, which are discrete additive subgroups of $\mathbb{Z}^m$ having finite index, i.e., the quotient group $\mathbb{Z}^m/\Lambda$ is finite. A lattice $\Lambda \subseteq \mathbb{Z}^m$ can equivalently be defined as the set of all integer linear combinations of m linearly independent *basis* vectors $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_m\} \subset \mathbb{Z}^m$:

$$\Lambda = \mathcal{L}(\mathbf{B}) = \left\{ \mathbf{B}\mathbf{c} = \sum\nolimits_{i \in [m]} c_i \mathbf{b}_i \ : \ \mathbf{c} \in \mathbb{Z}^m \right\}.$$

When $m \geq 2$, there are infinitely many bases that generate the same lattice.

Every lattice $\Lambda \subseteq \mathbb{Z}^m$ has a *unique* canonical basis $\mathbf{H} = \mathrm{HNF}(\Lambda) \in \mathbb{Z}^{m \times m}$ called its *Hermite normal form* (HNF). The only facts about the HNF that we require are that it is unique, and that it may be computed efficiently given an arbitrary basis $\mathbf{B}$ of the lattice (see [42] and references therein). We write $\mathrm{HNF}(\mathbf{B})$ to denote the Hermite normal form of the lattice generated by basis $\mathbf{B}$.

The following lemma will be useful in our constructions.

Lemma 1 ([40, Lemma 7.1, page 129]). *There is a deterministic poly-time algorithm* $\mathsf{ToBasis}(\mathbf{S}, \mathbf{B})$ *that, given a full-rank set (not necessarily a basis) of lattice vectors* $\mathbf{S} \subset \Lambda = \mathcal{L}(\mathbf{B})$*, outputs a basis* $\mathbf{T}$ *of* Λ *such that* $\|\widetilde{\mathbf{t}_i}\| \leq \|\widetilde{\mathbf{s}_i}\|$ *for all* i*.*

Hard Lattices and Problems. We will work with an certain family of integer lattices whose importance in cryptography was first demonstrated by Ajtai [5]. Let $n \geq 1$ and modulus $q \geq 2$ be integers; the dimension n is the main cryptographic security parameter throughout this work, and all other parameters are implicitly functions of n. An m-dimensional lattice from the family is specified relative to the additive group $\mathbb{Z}_q^n$ by a *parity check* (more accurately, "arity check") matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. The associated lattice is defined as

$$\Lambda^{\perp}(\mathbf{A}) = \left\{ \mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \sum\nolimits_{j \in [m]} x_j \cdot \mathbf{a}_j = \mathbf{0} \in \mathbb{Z}_q^n \right\} \subseteq \mathbb{Z}^m.$$

One may check that $\Lambda^{\perp}(\mathbf{A})$ contains $q\mathbb{Z}^m$ (and in particular, the identity $\mathbf{0} \in \mathbb{Z}^m$) and is closed under addition, hence it is a full-rank subgroup of (and lattice in) $\mathbb{Z}^m$. For any $\mathbf{y}$ in the subgroup of $\mathbb{Z}_q^n$ generated by the columns of $\mathbf{A}$, we also define the coset

$$\Lambda_{\mathbf{y}}^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{y}\} = \Lambda^{\perp}(\mathbf{A}) + \bar{\mathbf{x}},$$

where $\bar{\mathbf{x}} \in \mathbb{Z}^m$ is an arbitrary element of $\Lambda_{\bar{\mathbf{x}}}^{\perp}$.

It is known (see, e.g., [51, Claim 5.3]) that for any fixed constant $C > 1$ and any $m \geq Cn \lg q$, the columns of a *uniformly random* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ generate all of $\mathbb{Z}_q^n$, except with $2^{-\Omega(n)} = \operatorname{negl}(n)$ probability. (Moreover, the subgroup generated by $\mathbf{A}$ can be computed efficiently.) Therefore, throughout the paper we sometimes implicitly assume that such a uniform $\mathbf{A}$ generates $\mathbb{Z}_q^n$.

We recall the *short integer solution* (SIS) and *learning with errors* (LWE) problems, which may be seen as average-case problems related to the family of lattices described above.

Definition 1 (Short Integer Solution). *An instance of the* $\mathsf{SIS}_{q,\beta}$ *problem (in the* ℓ_2 *norm) is a uniformly random matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *for any desired* $m = \operatorname{poly}(n)$*. The goal is to find a* nonzero *integer vector* $\mathbf{v} \in \mathbb{Z}^m$ *such that* $\|\mathbf{v}\|_2 \leq \beta$ *and* $\mathbf{A}\mathbf{v} = \mathbf{0} \in \mathbb{Z}_q^n$*, i.e.,* $\mathbf{v} \in \Lambda^{\perp}(\mathbf{A})$*.*

Let χ be some distribution over $\mathbb{Z}_q$. For a vector $\mathbf{v} \in \mathbb{Z}_q^{\ell}$ of any dimension $\ell \geq 1$, $\mathsf{Noisy}_{\chi}(\mathbf{v}) \in \mathbb{Z}_q^{\ell}$ denotes the vector obtained by adding (modulo q) independent samples drawn from χ to each entry of $\mathbf{v}$ (one sample per entry). For a vector $\mathbf{s} \in \mathbb{Z}_q^n$, $A_{\mathbf{s},\chi}$ is the distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by choosing a vector $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random and outputting $(\mathbf{a}, \mathsf{Noisy}_{\chi}(\langle \mathbf{a}, \mathbf{s} \rangle))$. In this work (and most others relating to LWE), χ is always a discretized normal error distribution parameterized by some $\alpha \in (0,1)$, which is obtained by drawing $x \in \mathbb{R}$ from the Gaussian distribution of width α (i.e., x is chosen with probability proportional to $\exp(-\pi x^2/\alpha^2)$) and outputting $\lfloor q \cdot x \rceil \bmod q$.

Definition 2 (Learning with Errors). *The* $\mathsf{LWE}_{q,\chi}$ *problem is to distinguish, given oracle access to any desired* $m = \operatorname{poly}(n)$ *samples, between the distribution* $A_{\mathbf{s},\chi}$ *(for uniformly random and secret* $\mathbf{s} \in \mathbb{Z}_q^n$*) and the uniform distribution over* $\mathbb{Z}_q^n \times \mathbb{Z}_q$*.*

We write $\mathbf{Adv}_{\mathsf{SIS}_{q,\beta}}(\mathcal{A})$ and $\mathbf{Adv}_{\mathsf{LWE}_{q,\chi}}(\mathcal{A})$ to denote the success probability and distinguishing advantage of an algorithm $\mathcal{A}$ for the SIS and LWE problems, respectively.

For appropriate parameters, solving SIS and LWE (on the average, with non-negligible advantage) is known to be as hard as approximating certain lattice problems, such as the (decision) shortest vector problem, in the worst case. Specifically, for $q \geq \beta \cdot \omega(\sqrt{n \log n})$, solving $\mathsf{SIS}_{q,\beta}$ yields approximation factors of $\tilde{O}(\beta \cdot \sqrt{n})$ [41, 25]. For $q \geq (1/\alpha) \cdot \omega(\sqrt{n \log n})$, solving $\mathsf{LWE}_{q,\chi}$ yields approximation factors of $\tilde{O}(n/\alpha)$ (in some cases, via a quantum reduction); see [51, 45] for precise statements.

Gaussians over Lattices. We briefly recall Gaussian distributions over lattices, specialized to the family described above; for more details see [41, 25]. For any $s > 0$ and dimension $m \geq 1$, the Gaussian function $\rho_s \colon \mathbb{R}^m \to (0, 1]$ is defined as $\rho_s(\mathbf{x}) = \exp(-\pi \|\mathbf{x}\|^2/s^2)$. For any coset $\Lambda^{\perp}_{\mathbf{y}}(\mathbf{A})$, the *discrete Gaussian distribution* $D_{\Lambda^{\perp}_{\mathbf{y}}(\mathbf{A}),s}$ (centered at zero) over the coset assigns probability proportional to $\rho_s(\mathbf{x})$ to each $\mathbf{x} \in \Lambda^{\perp}_{\mathbf{y}}(\mathbf{A})$, and probability zero elsewhere.

We summarize several standard facts from the literature about discrete Gaussians over lattices, again specialized to our family of interest.

Lemma 2. *Let* $\mathbf{S}$ *be any basis of* $\Lambda^{\perp}(\mathbf{A})$ *for some* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *whose columns generate* $\mathbb{Z}_q^n$*, let* $\mathbf{y} \in \mathbb{Z}_q^n$ *be arbitrary, and let* $s \geq \|\widetilde{\mathbf{S}}\| \cdot \omega(\sqrt{\log n})$.

1. *[41, Lemma 4.4]:* $\Pr_{\mathbf{x} \leftarrow D_{\Lambda^{\perp}_{\mathbf{y}}(\mathbf{A}),s}}[\|\mathbf{x}\| > s \cdot \sqrt{m}] \leq \operatorname{negl}(n)$.
2. *[47, Lemma 2.11]:* $\Pr_{\mathbf{x} \leftarrow D_{\Lambda^{\perp}(\mathbf{A}),s}}[\mathbf{x} = \mathbf{0}] \leq \operatorname{negl}(n)$.
3. *[51, Corollary 3.16]: a set of* $O(m^2)$ *independent samples from* $D_{\Lambda^{\perp}(\mathbf{A}),s}$ *contains a set of* m *linearly independent vectors, except with* $\operatorname{negl}(n)$ *probability.*
4. *[25, Theorem 3.1]: For* $\mathbf{x} \leftarrow D_{\mathbb{Z}^m,s}$*, the marginal distribution of* $\mathbf{y} = \mathbf{A}\mathbf{x} \in \mathbb{Z}_q^n$ *is uniform (up to* $\operatorname{negl}(n)$ *statistical distance), and the conditional distribution of* $\mathbf{x}$ *given* $\mathbf{y}$ *is* $D_{\Lambda^{\perp}_{\mathbf{y}}(\mathbf{A}),s}$.
5. *[25, Theorem 4.1]: there is a PPT algorithm* $\mathsf{SampleD}(\mathbf{S}, \mathbf{y}, s)$ *that generates a sample from* $D_{\Lambda^{\perp}_{\mathbf{y}}(\mathbf{A}),s}$ *(up to* $\operatorname{negl}(n)$ *statistical distance).*

For Item 5 above, a recent work [46] gives an alternative SampleD algorithm that is more efficient and fully parallelizable; it works for any $s \geq \sigma_1(\mathbf{S}) \cdot \omega(\sqrt{\log n})$, where $\sigma_1(\mathbf{S})$ is the largest *singular value* of $\mathbf{S}$ (which is never less than $\|\widetilde{\mathbf{S}}\|$, but is also not much larger in most important cases; see [46] for details).

3 Principles of Bonsai Trees

In this section we lay out the framework and main techniques for the cultivation of bonsai trees. There are four basic principles: undirected growth, controlled growth, extending control over arbitrary new growth, and randomizing control.

3.1 Undirected Growth

Undirected growth is useful primarily for allowing a simulator to embed an underlying challenge problem (i.e., SIS or LWE) into a tree. This is done simply by drawing fresh uniformly random and independent samples $\mathbf{a}_i \in \mathbb{Z}_q^n$ from the problem distribution, and grouping them into (or appending them onto) a parity-check matrix $\mathbf{A}$.

More formally, let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be arbitrary for some $m \geq 0$, and let $\mathbf{A}' = \mathbf{A} \| \bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times m'}$ for some $m' > m$ be an arbitrary extension of $\mathbf{A}$. Then it is easy to see that $\Lambda^{\perp}(\mathbf{A}') \subseteq \mathbb{Z}^{m'}$ is a *higher-dimensional superlattice* of $\Lambda^{\perp}(\mathbf{A}) \subseteq \mathbb{Z}^m$, when the latter is lifted to $\mathbb{Z}^{m'}$. Specifically, for any $\mathbf{v} \in \Lambda^{\perp}(\mathbf{A})$, the vector $\mathbf{v}' = \mathbf{v} \| \mathbf{0} \in \mathbb{Z}^{m'}$ is in $\Lambda^{\perp}(\mathbf{A}')$ because $\mathbf{A}'\mathbf{v}' = \mathbf{A}\mathbf{v} = \mathbf{0} \in \mathbb{Z}_q^n$.

In fact, the columns of $\mathbf{A}'$ may be ordered arbitrarily (e.g., the columns of $\bar{\mathbf{A}}$ may be both appended and *prepended* to $\mathbf{A}$), which simply results in the entries of the vectors in $\Lambda^{\perp}(\mathbf{A}')$ being permuted in the corresponding manner. That is, $\Lambda^{\perp}(\mathbf{A}'\mathbf{P}) = \mathbf{P} \cdot \Lambda^{\perp}(\mathbf{A}')$ for any permutation matrix $\mathbf{P} \in \{0,1\}^{m' \times m'}$, because $(\mathbf{A}'\mathbf{P})\mathbf{x} = \mathbf{A}'(\mathbf{P}\mathbf{x}) \in \mathbb{Z}_q^n$ for all $\mathbf{x} = \mathbb{Z}^{m'}$.

3.2 Controlled Growth

We say that an arborist *controls* a lattice if it knows a relatively good (i.e., short) basis for the lattice. The following lemma says that a random lattice from our family of interest can be generated under control.[3]

Proposition 1 ([6]). *There is a fixed constant $C > 1$ and a probabilistic polynomial-time algorithm* $\mathsf{GenBasis}(1^n, 1^m, q)$ *that, for* $\mathrm{poly}(n)$*-bounded $m \geq Cn \lg q$, outputs $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{S} \in \mathbb{Z}^{m \times m}$ such that:*

- *the distribution of $\mathbf{A}$ is within* $\mathrm{negl}(n)$ *statistical distance of uniform,*
- *$\mathbf{S}$ is a basis of $\Lambda^{\perp}(\mathbf{A})$, and*
- $\|\widetilde{\mathbf{S}}\| \leq \widetilde{L} = O(\sqrt{n \log q})$.

3.3 Extending Control

Here we describe how an arborist may extend its control of a lattice to an arbitrary higher-dimensional extension, without any loss of quality in the resulting basis.

Lemma 3. *Let $\mathbf{S} \in \mathbb{Z}^{m \times m}$ be an arbitrary basis of $\Lambda^{\perp}(\mathbf{A})$ for some $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ whose columns generate the entire group $\mathbb{Z}_q^n$, and let $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times \bar{m}}$ be arbitrary. There is a deterministic polynomial-time algorithm* $\mathsf{ExtBasis}(\mathbf{S}, \mathbf{A}' = \mathbf{A} \| \bar{\mathbf{A}})$ *that outputs a basis $\mathbf{S}'$ of $\Lambda^{\perp}(\mathbf{A}') \subseteq \mathbb{Z}^{m+\bar{m}}$ such that $\|\widetilde{\mathbf{S}'}\| = \|\widetilde{\mathbf{S}}\|$. Moreover, the statement holds even if the columns of $\mathbf{A}'$ are permuted arbitrarily (e.g., if columns of $\bar{\mathbf{A}}$ are both appended and prepended to $\mathbf{A}$).*

Proof. The $\mathsf{ExtBasis}(\mathbf{S}, \mathbf{A}')$ algorithm computes and outputs an $\mathbf{S}'$ of the form

$$\mathbf{S}' = \begin{pmatrix} \mathbf{S} & \mathbf{W} \\ \mathbf{0} & \mathbf{I} \end{pmatrix},$$

[3] An earlier version of this paper [44] used an underlying lemma from [6] to *directly* extend a random parity-check matrix $\mathbf{A}$ (*without* known good basis) into a random $\mathbf{A}' = \mathbf{A} \| \bar{\mathbf{A}}$ *with* known good basis. While that method saves a small constant factor in key sizes, the applications become somewhat more cumbersome to describe; moreover, our present approach is more general.

where $\mathbf{I} \in \mathbb{Z}^{\bar{m}\times\bar{m}}$ is the identity matrix, and $\mathbf{W} \in \mathbb{Z}^{m\times\bar{m}}$ is an *arbitrary* (not necessarily short) solution to $\mathbf{AW} = -\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times\bar{m}}$. Note that $\mathbf{W}$ exists by hypothesis on $\mathbf{A}$, and may be computed efficiently using Gaussian elimination (for example).

We analyze $\mathbf{S}'$. First, $\mathbf{A}'\mathbf{S}' = \mathbf{0}$ by assumption on $\mathbf{S}$ and by construction, so $\mathbf{S}' \subset \Lambda^{\perp}(\mathbf{A}')$. Moreover, $\mathbf{S}'$ is a *basis* of $\Lambda^{\perp}(\mathbf{A}')$: let $\mathbf{v}' = \mathbf{v}\|\bar{\mathbf{v}} \in \Lambda^{\perp}(\mathbf{A}')$ be arbitrary, where $\mathbf{v} \in \mathbb{Z}^m$, $\bar{\mathbf{v}} \in \mathbb{Z}^{\bar{m}}$. Then we have

$$\mathbf{0} = \mathbf{A}'\mathbf{v}' = \mathbf{A}\mathbf{v} + \bar{\mathbf{A}}\bar{\mathbf{v}} = \mathbf{A}\mathbf{v} - (\mathbf{AW})\bar{\mathbf{v}} = \mathbf{A}(\mathbf{v} - \mathbf{W}\bar{\mathbf{v}}) \in \mathbb{Z}_q^n.$$

Thus $\mathbf{v} - \mathbf{W}\bar{\mathbf{v}} \in \Lambda^{\perp}(\mathbf{A})$, so by assumption on $\mathbf{S}$ there exists some $\mathbf{z} \in \mathbb{Z}^m$ such that $\mathbf{Sz} = \mathbf{v} - \mathbf{W}\bar{\mathbf{v}}$. Now let $\mathbf{z}' = \mathbf{z}\|\bar{\mathbf{v}} \in \mathbb{Z}^{m+\bar{m}}$. By construction, we have

$$\mathbf{S}'\mathbf{z}' = (\mathbf{Sz} + \mathbf{W}\bar{\mathbf{v}})\|\bar{\mathbf{v}} = \mathbf{v}\|\bar{\mathbf{v}} = \mathbf{v}'.$$

Because $\mathbf{v}' \in \Lambda^{\perp}(\mathbf{A}')$ was arbitrary, $\mathbf{S}'$ is therefore a basis of $\Lambda^{\perp}(\mathbf{A}')$.

We next confirm that $\|\widetilde{\mathbf{S}'}\| = \|\widetilde{\mathbf{S}}\|$. For every $i \in [m]$, we clearly have $\|\widetilde{\mathbf{s}'_i}\| = \|\widetilde{\mathbf{s}_i}\|$. Now because $\mathbf{S}$ is full-rank, we have $\mathrm{span}(\mathbf{S}) = \mathrm{span}(\mathbf{e}_1, \ldots, \mathbf{e}_m) \subseteq \mathbb{R}^{m+\bar{m}}$. Therefore, for $i = m+1, \ldots, m+\bar{m}$ we have $\widetilde{\mathbf{s}'_i} = \mathbf{e}_i \in \mathbb{R}^{m+\bar{m}}$, so $\|\widetilde{\mathbf{s}'_i}\| = 1 \leq \|\widetilde{\mathbf{s}'_1}\|$, as desired.

For the final part of the lemma, we simply compute $\mathbf{S}'$ for $\mathbf{A}' = \mathbf{A}\|\bar{\mathbf{A}}$ as described above, and output $\mathbf{S}'' = \mathbf{PS}'$ as a basis for $\Lambda^{\perp}(\mathbf{A}'\mathbf{P})$, where $\mathbf{P}$ is the desired permutation matrix. The Gram-Schmidt lengths remain unchanged, i.e., $\|\widetilde{\mathbf{s}''_i}\| = \|\widetilde{\mathbf{s}'_i}\|$, because $\mathbf{P}$ is orthogonal and hence the right-triangular matrices are exactly the same in the QR decompositions of $\mathbf{S}'$ and $\mathbf{PS}'$.

An Optimization. In many of our cryptographic applications, a common design pattern is to extend a basis $\mathbf{S}$ of an m-dimensional lattice $\Lambda^{\perp}(\mathbf{A})$ to a basis $\mathbf{S}'$ of a dimension-m' superlattice $\Lambda^{\perp}(\mathbf{A}')$, and then immediately sample (one or more times) from a discrete Gaussian over the superlattice. For the construction and analysis of our schemes, it is more convenient and modular to treat these operations separately; however, a naive implementation would be rather inefficient, requiring at least $(m')^2$ space and time (where m' can be substantially larger than m). Fortunately, the special structure of the extended basis $\mathbf{S}'$, together with the recursive "nearest-plane" operation of the $\mathsf{SampleD}$ algorithm from [25], can be exploited to avoid any *explicit* computation of $\mathbf{S}'$, thus saving a significant amount of time and space over the naive approach.

Let $\mathbf{S} \in \mathbb{Z}^{m\times m}$ be a basis of $\Lambda^{\perp}(\mathbf{A})$, and let $\mathbf{A}' = \mathbf{A}\|\bar{\mathbf{A}}$ for some $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times\bar{m}}$, where $m' = m + \bar{m}$. Consider a hypothetical execution of $\mathsf{SampleD}(\mathbf{S}', \mathbf{y}', s)$, where $\mathbf{S}' = \left(\begin{smallmatrix}\mathbf{S} & \mathbf{W}\\ \mathbf{0} & \mathbf{I}\end{smallmatrix}\right)$ is the extended basis as described in the proof of Lemma 3. Recall that for all $i = m+1, \ldots, m'$, the vectors $\mathbf{s}'_i$ are integral and have unit Gram-Schmidt vectors $\widetilde{\mathbf{s}'_i} = \mathbf{e}_i$. By inspection, it can be verified that a recursive execution of $\mathsf{SampleD}(\mathbf{S}', \mathbf{y}', s)$ simply ends up choosing all the entries of $\bar{\mathbf{v}} \in \mathbb{Z}^{\bar{m}}$ *independently* from $D_{\mathbb{Z},s}$, then choosing $\mathbf{v} \leftarrow \mathsf{SampleD}(\mathbf{S}, \mathbf{y}' - \bar{\mathbf{A}}\bar{\mathbf{v}}, s)$, and outputting $\mathbf{v}' = \mathbf{v}\|\bar{\mathbf{v}}$. Therefore, the optimized algorithm can perform exactly the same steps, thus avoiding any need to compute and store $\mathbf{W}$ itself. A similar optimization also works for any permutation of the columns of $\mathbf{A}'$.

In the language of the "preimage sampleable" function $f_{\mathbf{A}}(\mathbf{v}) = \mathbf{A}\mathbf{v} \in \mathbb{Z}_q^n$ defined in [25], the process described above corresponds to sampling a preimage from $f_{\mathbf{A}'}^{-1}(\mathbf{y}')$ by first computing $\bar{\mathbf{y}} = f_{\bar{\mathbf{A}}}(\bar{\mathbf{v}}) = \bar{\mathbf{A}}\bar{\mathbf{v}} \in \mathbb{Z}_q^n$ in the "forward" direction (for random $\bar{\mathbf{v}} \leftarrow \mathbb{D}_{\mathbb{Z}^{\bar{m}},s}$), then choosing a random preimage $\mathbf{v} \leftarrow f_{\mathbf{A}}^{-1}(\mathbf{y}' - \bar{\mathbf{y}})$ under the appropriate distribution, and outputting $\mathbf{v}' = \mathbf{v} \| \bar{\mathbf{v}}$.[4]

3.4 Randomizing Control

Finally, we show how an arborist can randomize its lattice basis, with a slight loss in quality. This operation is useful for securely delegating control to another entity, because the resulting basis is still short, but is statistically independent (essentially) of the original basis.

The probabilistic polynomial-time algorithm $\mathsf{RandBasis}(\mathbf{S}, s)$ takes a basis $\mathbf{S}$ of an m-dimensional integer lattice Λ and a parameter $s \geq \|\widetilde{\mathbf{S}}\| \cdot \omega(\sqrt{\log n})$, and outputs a basis $\mathbf{S}'$ of Λ, generated as follows.

1. Let $i \leftarrow 0$. While $i < m$,
   (a) Choose $\mathbf{v} \leftarrow \mathsf{SampleD}(\mathbf{S}, s)$. If $\mathbf{v}$ is linearly independent of $\{\mathbf{v}_1, \ldots, \mathbf{v}_i\}$, then let $i \leftarrow i + 1$ and let $\mathbf{v}_i = \mathbf{v}$.
2. Output $\mathbf{S}' = \mathsf{ToBasis}(\mathbf{V}, \mathrm{HNF}(\mathbf{S}))$.

In the final step, the (unique) Hermite normal form basis $\mathrm{HNF}(\mathbf{S})$ of Λ is used to ensure that no information particular to $\mathbf{S}$ is leaked by the output; any other publicly available (or publicly computable) basis of the lattice could also be used in its place.

Lemma 4. *With overwhelming probability,* $\mathbf{S}' \leftarrow \mathsf{RandBasis}(\mathbf{S}, s)$ *repeats Step 1a at most* $O(m^2)$ *times, and* $\|\mathbf{S}'\| \leq s \cdot \sqrt{m}$*. Moreover, for any two bases* $\mathbf{S}_0, \mathbf{S}_1$ *of the same lattice and any* $s \geq \max\{\|\widetilde{\mathbf{S}_0}\|, \|\widetilde{\mathbf{S}_1}\|\} \cdot \omega(\sqrt{\log n})$*, the outputs of* $\mathsf{RandBasis}(\mathbf{S}_0, s)$ *and* $\mathsf{RandBasis}(\mathbf{S}_1, s)$ *are within* $\mathrm{negl}(n)$ *statistical distance.*

Proof. The bound on $\|\mathbf{S}'\|$ and on the number of iterations follow immediately from Lemma 2, items 1 and 3, respectively. The claim on the statistical distance follows from the fact that each sample $\mathbf{v}$ drawn in Step 1a has the same distribution (up to $\mathrm{negl}(n)$ statistical distance) whether $\mathbf{S}_0$ or $\mathbf{S}_1$ is used, and the fact that $\mathrm{HNF}(\mathbf{S}_0) = \mathrm{HNF}(\mathbf{S}_1)$ because the Hermite normal form of a lattice is unique.

4 Signatures

Here we use bonsai tree principles to construct a signature scheme that is existentially unforgeable under a *static* chosen-message attack (i.e., eu-scma-secure). As discussed in Section 2.2, this suffices (using chameleon hashing) for the construction of an (offline / online) signature scheme that is unforgeable under adaptive chosen-message attack (eu-acma-secure).

[4] An earlier version of this paper [17] explicitly defined a sampling procedure using this perspective, and gave a (somewhat involved) proof that it correctly samples from a discrete Gaussian over $\Lambda^{\perp}(\mathbf{A}')$. Here, correctness follows directly by examining the operation of $\mathsf{SampleD}$ on the structured basis $\mathbf{S}'$.

Our scheme involves a few parameters:

- a dimension $m = O(n \lg q)$ and a bound $\widetilde{L} = O(\sqrt{n \lg q})$, as per Proposition 1;
- a (hashed) message length k, which induces a 'total dimension' $m' = m \cdot (k+1)$;
- a Gaussian parameter $s = \widetilde{L} \cdot \omega(\sqrt{\log n})$.

The scheme SIG is defined as follows.

- Gen: using Proposition 1, generate $\mathbf{A}_0 \in \mathbb{Z}_q^{n \times m}$ that is (negligibly close to) uniform, together with a basis $\mathbf{S}_0$ of $\Lambda^\perp(\mathbf{A}_0)$ such that $\|\widetilde{\mathbf{S}_0}\| \leq \widetilde{L}$. (Recall that the columns of $\mathbf{A}_0$ generate all of $\mathbb{Z}_q^n$, with overwhelming probability.)

  Then for each $(b, j) \in \{0,1\} \times [k]$, choose uniformly random and independent $\mathbf{A}_j^{(b)} \in \mathbb{Z}_q^{n \times m}$. Output $vk = (\mathbf{A}_0, \{\mathbf{A}_j^{(b)}\})$ and $sk = (\mathbf{S}_0, vk)$.
- $\mathsf{Sign}(sk, \mu \in \{0,1\}^k)$: let $\mathbf{A}_\mu = \mathbf{A}_0 \| \mathbf{A}_1^{(\mu_1)} \| \cdots \| \mathbf{A}_k^{(\mu_k)} \in \mathbb{Z}_q^{n \times m'}$. Output $\mathbf{v} \leftarrow D_{\Lambda^\perp(\mathbf{A}_\mu), s}$, via
$$\mathbf{v} \leftarrow \mathsf{SampleD}(\mathsf{ExtBasis}(\mathbf{S}_0, \mathbf{A}_\mu), \mathbf{0}, s).$$
  (In the rare event that $\mathbf{v} = \mathbf{0}$ or $\|\mathbf{v}\| > s \cdot \sqrt{m'}$ (Lemma 2, items 2 and 2), resample $\mathbf{v}$. Note also that the optimization of Section 3.3 applies here.)
- $\mathsf{Ver}(vk, \mu, \mathbf{v})$: let $\mathbf{A}_\mu$ be as above. Accept if $\mathbf{v} \neq \mathbf{0}$, $\|\mathbf{v}\| \leq s \cdot \sqrt{m'}$, and $\mathbf{v} \in \Lambda^\perp(\mathbf{A}_\mu)$; else, reject.

Completeness is by inspection. Note that the matrix $\mathbf{A}_0$ can be omitted from the above scheme (thus making the total dimension $m \cdot k$), at the expense of a secret key that contains *two* short bases $\mathbf{S}_1^{(b)}$ of $\Lambda^\perp(\mathbf{A}_1^{(b)})$, for $b = 0, 1$. The scheme's algorithms and security proof are easy to modify accordingly.

4.1 Security

Theorem 1. *There exists a PPT oracle algorithm (a reduction) $\mathcal{S}$ attacking the $\mathsf{SIS}_{q,\beta}$ problem for $\beta = s \cdot \sqrt{m'}$ such that, for any adversary $\mathcal{F}$ mounting an eu-scma attack on SIG and making at most Q queries,*

$$\mathbf{Adv}_{\mathsf{SIS}_{q,\beta}}(\mathcal{S}^{\mathcal{F}}) \geq \mathbf{Adv}_{SIG}^{eu\text{-}scma}(\mathcal{F})/(k \cdot Q) - \mathrm{negl}(n).$$

Proof. Let $\mathcal{F}$ be an adversary mounting an eu-scma attack on SIG. We construct a reduction $\mathcal{S}$ attacking $\mathsf{SIS}_{q,\beta}$. The reduction $\mathcal{S}$ takes as input $m'' = m \cdot (2k+1)$ uniformly random and independent samples from $\mathbb{Z}_q^n$ in the form of a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m''}$, parsing $\mathbf{A}$ as

$$\mathbf{A} = \mathbf{A}_0 \| \mathbf{U}_1^{(0)} \| \mathbf{U}_1^{(1)} \| \cdots \| \mathbf{U}_k^{(0)} \| \mathbf{U}_k^{(1)}$$

for matrices $\mathbf{A}_0, \mathbf{U}_i^{(b)} \in \mathbb{Z}_q^{n \times m}$.

$\mathcal{S}$ simulates the static chosen-message attack to $\mathcal{F}$ as follows. First, $\mathcal{S}$ invokes $\mathcal{F}$ to receive Q messages $\mu^{(1)}, \ldots, \mu^{(Q)} \in \{0,1\}^k$. (We may assume without loss of generality that $\mathcal{F}$ makes exactly Q queries.) Then $\mathcal{S}$ computes the set P of all strings $p \in \{0,1\}^{\leq k}$ having the property that p is a shortest string for which no $\mu^{(j)}$ has p as a prefix. In brief, each p corresponds to a maximal subtree of $\{0,1\}^{\leq k}$ (viewed as a tree) that does not contain any of the queried messages. The set P may be computed efficiently via a breadth-first pruned search of $\{0,1\}^{\leq k}$. Namely, starting from a queue initialized to $\{\varepsilon\}$, repeat the following until the queue is empty: remove the next string p from the queue and test whether it is the prefix of any $\mu^{(j)}$; if not, add p to P, else if $|p| < k$, add $p\|0, p\|1 \in \{0,1\}^{\leq k}$ to the queue. Note that this algorithm runs in polynomial time because the only strings ever placed in the queue are prefixes of $\mu^{(j)}$, and hence there are at most $k \cdot Q$ strings in the set.

Next, $\mathcal{S}$ chooses some p from P uniformly at random, letting $t = |p|$. It then provides an SIG verification key $vk = (\mathbf{A}_0, \{\mathbf{A}_j^{(b)}\})$ to $\mathcal{F}$, generated as follows:

- *Uncontrolled growth:* for each $i \in [t]$, let $\mathbf{A}_i^{(p_i)} = \mathbf{U}_i^{(0)}$. For $i = t+1, \ldots, k$, and $b \in \{0,1\}$, let $\mathbf{A}_i^{(b)} = \mathbf{U}_i^{(b)}$.
- *Controlled growth:* for each $i \in [t]$, invoke Proposition 1 to generate $\mathbf{A}_i^{(1-p_i)}$ and basis $\mathbf{S}_i$ of $\Lambda^{\perp}(\mathbf{A}_i^{1-p_i})$ such that $\|\widetilde{\mathbf{S}}_i\| \leq \widetilde{L}$.

Next, $\mathcal{S}$ generates signatures for each queried message $\mu = \mu^{(j)}$ as follows: let $i \in [t]$ be the first position at which $\mu_i \neq p_i$ (such i exists by construction of p). Then $\mathcal{S}$ generates the signature $\mathbf{v} \leftarrow D_{\Lambda^{\perp}(\mathbf{A}_\mu),s}$ as

$$\mathbf{v} \leftarrow \mathsf{SampleD}(\mathsf{ExtBasis}(\mathbf{S}_i, \mathbf{A}_\mu), \mathbf{0}, s),$$

where $\mathbf{A}_\mu = \mathbf{A}_L \| \mathbf{A}_i^{(1-p_i)} \| \mathbf{A}_R$ (for some matrices $\mathbf{A}_L, \mathbf{A}_R$) is as in the signature scheme, and has the form required by $\mathsf{ExtBasis}$. (In the event that $\mathbf{v} = \mathbf{0}$ or $\|\mathbf{v}\| > \beta = s \cdot \sqrt{m'}$, resample $\mathbf{v}$.)

Finally, if $\mathcal{F}$ produces a valid forgery $(\mu^*, \mathbf{v}^* \neq \mathbf{0})$, then we have $\mathbf{v}^* \in \Lambda^{\perp}(\mathbf{A}_{\mu^*})$, for $\mathbf{A}_{\mu^*}$ as defined in the scheme. First, $\mathcal{S}$ checks whether p is a prefix of μ^*. If not, $\mathcal{S}$ aborts; otherwise, note that $\mathbf{A}_{\mu^*}$ is the concatenation of $\mathbf{A}_0$ and k blocks $\mathbf{U}_i^{(b)}$. Therefore, by inserting zeros into $\mathbf{v}^*$, $\mathcal{S}$ can generate a nonzero $\mathbf{v} \in \mathbb{Z}^{m''}$ so that $\mathbf{A}\mathbf{v} = \mathbf{0} \in \mathbb{Z}_q^n$. Finally, $\mathcal{S}$ outputs $\mathbf{v}$ as a solution to SIS.

We now analyze the reduction. First observe that conditioned on any choice of $p \in P$, the verification key vk given to $\mathcal{F}$ is negligibly close to uniform, and the signatures given to $\mathcal{F}$ are distributed exactly as in the real attack (up to negligible statistical distance), by Lemma 2 and the fact that $s \geq \|\widetilde{\mathbf{S}}_i\| \cdot \omega(\sqrt{\log n})$. Therefore, $\mathcal{F}$ outputs a valid forgery $(\mu^*, \mathbf{v}^* \neq \mathbf{0})$ with probability at least $\mathbf{Adv}_{\text{SIG}}^{\text{eu-scma}}(\mathcal{F}) - \text{negl}(n)$. Finally, conditioned on the forgery, the choice of $p \in P$ is still negligibly close to uniform, so p is a prefix of μ^* with probability at least $1/(k \cdot Q) - \text{negl}(n)$. In such a case, $\mathbf{A}\mathbf{v} = \mathbf{0}$ and $\|\mathbf{v}\| = \|\mathbf{v}^*\| \leq \beta$ by construction, hence $\mathbf{v}$ is a valid solution to the given SIS instance, as desired.

5 Hierarchical ID-Based Encryption

5.1 Key Encapsulation Mechanism

For our HIBE schemes, it is convenient and more modular to abstract away the encryption and decryption processes into a key-encapsulation mechanism (KEM). The following LWE-based KEM from [25] (which is dual to the scheme of Regev [51]) is now standard. The reader need not be concerned with the details in order to progress to the HIBE schemes; it is enough simply to understand the KEM interface (i.e., the public/secret keys and ciphertext).

KEM is parametrized by a modulus q, dimension m, key length ℓ, and Gaussian parameter s that determines the error distribution χ used for encapsulation. As usual, all these parameters are functions of the LWE dimension n, and are instantiated based on the particular context in which the KEM is used.

- KEM.Gen: Choose $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ uniformly at random, $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,s}$ and set $\mathbf{y} = \mathbf{Ae} \in \mathbb{Z}_q^n$. Output public key $pk = (\mathbf{A}, \mathbf{y}) \in \mathbb{Z}_q^{n \times (m+1)}$ and secret key $sk = \mathbf{e}$.
- KEM.Encaps($pk = (\mathbf{A}, \mathbf{y})$): Choose $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ and let
$$\mathbf{b} \leftarrow \mathsf{Noisy}_\chi(\mathbf{A}^t\mathbf{s}) \quad \text{and} \quad p \leftarrow \mathsf{Noisy}_\chi(\mathbf{y}^t\mathbf{s} + k \cdot \lfloor q/2 \rfloor),$$
where $k \in \{0,1\}$ is a random bit. Output the key bit k and ciphertext $(\mathbf{b}, p) \in \mathbb{Z}_q^{m+1}$.
- KEM.Decaps($sk = \mathbf{e}, (\mathbf{b}, p)$): Compute $p - \mathbf{e}^t\mathbf{b} \bmod q$ and output 0 if the result is closer to 0 than $\lfloor q/2 \rfloor$ modulo q, and 1 otherwise.

As explained in [25], the basic scheme can be amortized to allow for KEM keys of length $\ell = \text{poly}(n)$ bits, with ciphertexts in $\mathbb{Z}_q^{m+\ell}$ and public keys in $\mathbb{Z}_q^{n \times (m+\ell)}$. This is done by including ℓ syndromes $\mathbf{y}_1, \ldots, \mathbf{y}_\ell$ (where $\mathbf{y}_i = \mathbf{Ae}_i$ for independent $\mathbf{e}_i \leftarrow D_{\mathbb{Z}^m,s}$) in the public key, and concealing one KEM bit with each of them using the same $\mathbf{s}$ and $\mathbf{b} \leftarrow \mathsf{Noisy}_\chi(\mathbf{A}^t\mathbf{s})$. Furthermore, it is also possible to conceal $\Omega(\log n)$ KEM bits per syndrome, which yields an amortized expansion factor of $O(1)$. For simplicity, in this work we deal only with the case of single-bit encapsulation, but all of our schemes can be amortized in a manner similar to the above.

We point out one nice property of KEM, which is convenient for the security proof of our BTE/HIBE schemes: for any dimensions $m \leq m'$ (and leaving all other parameters the same), the adversary's view for dimension m may be produced by taking a view for dimension m', and truncating the values $\mathbf{A} \in \mathbb{Z}_q^{n \times m'}$ and $\mathbf{b} \in \mathbb{Z}_q^{m'}$ to their first m (out of m') components.

The following lemma is standard from prior work.

Lemma 5 (Correctness and Security). *Let $m \geq Cn \lg q$ for any fixed constant $C > 1$, let $q \geq 4s(m+1)$, and let χ be the discretized Gaussian of parameter α for $1/\alpha \geq s\sqrt{m+1} \cdot \omega(\sqrt{\log n})$. Then* KEM.Decaps *is correct with overwhelming probability over all the randomness of* KEM.Gen *and* KEM.Encaps. *Moreover, there exists a PPT oracle algorithm (a reduction) $\mathcal{S}$ attacking the* $\mathsf{LWE}_{q,\chi}$ *problem such that, for any adversary $\mathcal{A}$ mounting an ind-cpa attack on KEM,*

$$\mathbf{Adv}_{\mathsf{LWE}_{q,\chi}}(\mathcal{S}^{\mathcal{A}}) \geq \mathbf{Adv}_{\mathit{KEM}}^{\mathit{ind\text{-}cpa}}(\mathcal{A}) - \text{negl}(n).$$

5.2 BTE and HIBE Scheme

Our main construction in this section is a binary tree encryption (BTE) scheme, which suffices for full HIBE by hashing the components of the identities with a universal one-way or collision-resistant hash function [16]. We mainly focus on the case of *selective-identity*, *chosen-plaintext* attacks, i.e., sid-ind-cpa security.

The BTE scheme is parametrized by a dimension $m = O(n \lg q)$ as per Proposition 1, as well as a few quantities that are indexed by depth within the hierarchy. For an identity at depth $i \geq 0$ (where $i = 0$ corresponds to the root),

- $(i+1)m$ is the dimension of a lattice associated with the identity;
- $\widetilde{L_i}$ is an upper bound on the Gram-Schmidt lengths of its secret short basis;
- for $i \geq 1$, s_i is the Gaussian parameter used to generate that secret basis, which must exceed $\widetilde{L_j} \cdot \omega(\sqrt{\log n})$ for all $j < i$.

These parameters, along with the total depth d of the hierarchy (or more accurately, the maximum number of delegations down any chain of authority), determine the modulus q and error distribution χ used in the cryptosystem. We instantiate all the parameters after describing the scheme.

- $\mathsf{BTE.Setup}(d)$: Generate (via Proposition 1) $\mathbf{A}_0 \in \mathbb{Z}_q^{n \times m}$ that is (negligibly close to) uniform with a basis $\mathbf{S}_0$ of $\Lambda^{\perp}(\mathbf{A}_0)$ such that $\|\widetilde{\mathbf{S}}\| \leq \widetilde{L_0}$. For each $(b,j) \in \{0,1\} \times [d]$, generate uniform and independent $\mathbf{A}_j^{(b)} \in \mathbb{Z}_q^{n \times m}$. Choose $\mathbf{y} \in \mathbb{Z}_q^n$ uniformly at random. Output $mpk = (\mathbf{A}_0, \{\mathbf{A}_j^{(b)}\}, \mathbf{y}, d)$ and $msk = \mathbf{S}_0$.

  All remaining algorithms implicitly take the master public key mpk as input. For an identity $id = (id_1, \ldots, id_t)$ of length $t = |id| \leq d$, let

$$\mathbf{A}_{id} = \mathbf{A}_0 \| \mathbf{A}_1^{(id_1)} \| \cdots \| \mathbf{A}_t^{(id_t)} \in \mathbb{Z}_q^{n \times (t+1)m},$$

  and let $pk_{id} = (\mathbf{A}_{id}, \mathbf{y})$ denote the KEM public key associated with identity id.
- $\mathsf{BTE.Extract}(sk_{id} = (\mathbf{S}_{id}, \mathbf{e}_{id}), id' = id \| \bar{id})$: if $t' = |id'| > d$, output $\perp$. Else, let $t = |id|$ and $\bar{t} = |\bar{id}|$, and choose

$$\mathbf{S}_{id'} \leftarrow \mathsf{RandBasis}(\mathsf{ExtBasis}(\mathbf{S}_{id}, \mathbf{A}_{id'}), s_{t'}).$$

  (Note that $s_{t'} \geq \widetilde{L_t} \cdot \omega(\sqrt{\log n}) \geq \|\widetilde{\mathbf{S}_{id}}\| \cdot \omega(\sqrt{\log n})$, as required by $\mathsf{RandBasis}$.) Sample $\mathbf{e}_{id'} \leftarrow D_{\Lambda_{\mathbf{y}}^{\perp}(\mathbf{A}_{id'}), s_{t'}}$ using $\mathsf{SampleD}(\mathsf{ExtBasis}(\mathbf{S}_{id}, \mathbf{A}_{id'}), \mathbf{y}_{id'}, s_{t'})$ and output $sk_{id'} = (\mathbf{S}_{id'}, \mathbf{e}_{id'})$.
- $\mathsf{BTE.Encaps}(id)$: Output $(k, C) \leftarrow \mathsf{KEM.Encaps}(pk_{id})$.
- $\mathsf{BTE.Decaps}(sk_{id} = (\mathbf{S}_{id}, \mathbf{e}_{id}), C)$: Output $k \leftarrow \mathsf{KEM.Decaps}(\mathbf{e}_{id}, C)$.

A multi-bit BTE follows in the same way from the multi-bit KEM scheme by using multiple uniform syndromes $\mathbf{y}_i \in \mathbb{Z}_q^n$, one for each bit of the KEM key.

Instantiating the parameters. Suppose that BTE is employed in a setting in which $\mathsf{BTE.Extract}(sk_{id}, id')$ is invoked only on identities id' whose lengths are a multiple of some $k \geq 1$. For example, consider the two main applications of [16]: in the forward-secure encryption scheme we have $k = 1$, while in the generic BTE-to-HIBE transformation, k is the output length of some UOWHF.

It is enough to define s_i and $\widetilde{L_i}$ for i that are multiples of k. Let

$$\widetilde{L_i} = s_i \cdot \sqrt{(i+1)m} = s_i \cdot O(\sqrt{d \cdot n \lg q})$$

be the bound on the Gram-Schmidt lengths of the secret bases (and note that this bound is satisfied with overwhelming probability by Lemma 2). Define $s_i = \widetilde{L_{i-k}} \cdot \omega(\sqrt{\log n})$, and unwind the recurrence to obtain

$$\widetilde{L_t} = \widetilde{L_0} \cdot O(\sqrt{d \cdot n \lg q})^{t/k} \cdot \omega(\sqrt{\log n})^{t/k},$$

with $\widetilde{L_0} = O(\sqrt{n \lg q})$ by Proposition 1.

Finally, to ensure that the underlying KEM is complete (Lemma 5), we let $q \geq 4s_d \cdot (d+2)m$ and $1/\alpha = s_d \cdot \sqrt{(d+2)m} \cdot \omega(\sqrt{\log n})$. (It is also possible to use a different noise parameter for each level of the hierarchy.) For any $d = \text{poly}(n)$, invoking known worst-case to average-case reductions for LWE yields an underlying approximation factor of $\tilde{O}(n/\alpha) = n \cdot \tilde{O}((d/k) \cdot \sqrt{nk}))^{d/k}$ for worst-case lattice problems.

Extensions: Anonymity and chosen-ciphertext security. With a small modification, BTE may be made *anonymous* across all depths of the hierarchy. That is, a ciphertext hides (computationally) the particular identity to which it was encrypted. The modification is simply to extend the $\mathbf{b}$ component of the KEM ciphertext to have length exactly $(d+1)m$, by padding it with enough uniformly random and independent elements of $\mathbb{Z}_q$. (The decryption algorithm simply ignores the padding.) Anonymity then follows immediately by the pseudorandomness of the LWE distribution.

Security under chosen-ciphertext attack (sid-ind-cca or aid-ind-cca) follows directly by a transformation of [10], from ind-cpa-secure HIBE for depth $d+1$ to ind-cca-secure HIBE for depth d.

Theorem 2 (Security of BTE). *There exists a PPT oracle algorithm (a reduction) $\mathcal{S}$ attacking KEM (instantiated with dimension $(d+1)m$ and q, χ as in BTE) such that, for any adversary $\mathcal{A}$ mounting an atk attack on BTE,*

$$\mathbf{Adv}_{\mathit{KEM}}(\mathcal{S}^{\mathcal{A}}) \geq \mathbf{Adv}_{\mathit{BTE}}^{\mathit{sid\text{-}ind\text{-}cpa}}(\mathcal{A}) - \text{negl}(n).$$

Proof. Let $\mathcal{A}$ be an adversary mounting a sid-ind-cpa-attack on BTE. We construct a reduction $\mathcal{S}$ attacking KEM. It is given a uniformly random public key $pk = (\mathbf{A}, \mathbf{y}) \in \mathbb{Z}_q^{n \times (d+1)m} \times \mathbb{Z}_q^n$, an encapsulation $(\mathbf{b}, p) \in \mathbb{Z}_q^{(d+1)m} \times \mathbb{Z}_q$, and a bit k which either is encapsulated by $(\mathbf{b}, p)$ or is uniform and independent; the goal of $\mathcal{S}$ is to determine which is the case.

$\mathcal{S}$ simulates the (selective-identity) attack on BTE to $\mathcal{A}$ as follows. First, $\mathcal{S}$ invokes $\mathcal{A}$ on 1^d to receive its challenge identity id^* of length $t^* = |id^*| \in [d]$. Then $\mathcal{S}$ produces a master public key mpk, encapsulated key, and some secret internal state as follows:

- *Parsing the KEM inputs.* Parse $\mathbf{A}$ as $\mathbf{A} = \mathbf{A}_0 \| \mathbf{A}_1 \| \cdots \| \mathbf{A}_d \in \mathbb{Z}_q^{n \times (d+1)m}$ for $\mathbf{A}_i \in \mathbb{Z}_q^{n \times m}$ for all $i \in \{0, \ldots, d\}$. Similarly, truncate $\mathbf{b}$ to $\mathbf{b}^* \in \mathbb{Z}_q^{(t^*+1)m}$.
- *Undirected growth.* For each $i \in [t^*]$, let $\mathbf{A}_i^{(id_i^*)} = \mathbf{A}_i$.
- *Controlled growth.* For each $i \in [t^*]$, generate $\mathbf{A}_i^{(1-id_i^*)} \in \mathbb{Z}_q^{n \times m}$ and basis $\mathbf{S}_i$ by invoking $\mathsf{GenBasis}(1^n, 1^m, q)$. If $t^* < d$, for each $b \in \{0,1\}$ generate $\mathbf{A}_{t^*+1}^{(b)}$ and basis $\mathbf{S}_{t^*+1}^{(b)}$ by two independent invocations of $\mathsf{GenBasis}(1^n, 1^m, q)$. For each $i > t^* + 1$ (if any) and $b \in \{0,1\}$, generate $\mathbf{A}_i^{(b)} \in \mathbb{Z}_q^{n \times m}$ uniformly at random.

$\mathcal{S}$ gives to $\mathcal{A}$ the master public key $mpk = (\mathbf{A}_0, \{\mathbf{A}_j^{(b)}\}, \mathbf{y}, d)$, the encapsulation $(\mathbf{b}^*, p)$, and the key bit k.

Then $\mathcal{S}$ answers each secret-key query on an identity id that is not a prefix of (or equal to) id^* as follows:

- If $t = |id| \leq t^*$, then let $i \geq 1$ be the first position at which $id_i \neq id_i^*$. Answer the query with $(\mathbf{S}_{id}, \mathbf{e}_{id})$, which are computed by

$$\mathbf{S}_{id} \leftarrow \mathsf{RandBasis}(\mathsf{ExtBasis}(\mathbf{S}_i, \mathbf{A}_{id}), s_t)$$
$$\mathbf{e}_{id} \leftarrow \mathsf{SampleD}(\mathsf{ExtBasis}(\mathbf{S}_i, \mathbf{A}_{id}), \mathbf{y}_{id}, s_t).$$

- If $t = |id| > t^*$, answer the query $(\mathbf{S}_{id}, \mathbf{e}_{id})$, which are computed by

$$\mathbf{S}_{id} \leftarrow \mathsf{RandBasis}(\mathsf{ExtBasis}(\mathbf{S}_{t^*+1}^{(id_{t^*+1})}, \mathbf{A}_{id}), s_t)$$
$$\mathbf{e}_{id} \leftarrow \mathsf{SampleD}(\mathsf{ExtBasis}(\mathbf{S}_{t^*+1}^{(id_{t^*+1})}, \mathbf{A}_{id}), \mathbf{y}_{id}, s_t).$$

Finally, $\mathcal{S}$ outputs whatever bit $\mathcal{A}$ outputs.

We now analyze the reduction. First, observe that the master public key given to $\mathcal{A}$ is negligibly close to uniform (hence properly distributed), by hypothesis on KEM and by Proposition 1. Next, one can check that secret-key queries are distributed as in the real attack (to within $\mathrm{negl}(n)$ statistical distance), by Lemma 4 (note that the Gram-Schmidt vectors of each basis $\mathbf{S}_i, \mathbf{S}_{t^*+1}^{(b)}$ are sufficiently short to invoke RandBasis and SampleD). Finally, the encapsulation $(\mathbf{b}^*, p)$ (for identity id^*) and key bit k are distributed as in the real attack, by the truncation property of KEM. Therefore, $\mathcal{S}$'s overall advantage is within $\mathrm{negl}(n)$ of $\mathcal{A}$'s advantage, as desired.

5.3 Full Security in the Random Oracle Model

To obtain a fully secure HIBE in the random oracle model we can use a generic transformation by Boneh and Boyen [8]. It starts from a selective-id secure HIBE and applies hash functions to the identities. The resulting HIBE is fully secure, in the random oracle model, losing roughly a factor of Q_H^d in security, where Q_H is the number of random oracle queries. Furthermore, the $\{\mathbf{A}_j^{(b)}\}$ component of the master public key may be omitted, because each $\mathbf{A}_{id}$ can instead be constructed by querying the random oracle on, say, each prefix of the identity id.

We now give a more efficient fully-secure HIBE scheme, ROHIBE, in the random oracle model. It can be seen as a generalization of the GPV IBE scheme [25]. Compared to the fully-secure scheme obtained by the generic transformation, the efficiency improvement stems from the fact that $\mathbf{y}$ from pk_{id} now also depends on the identity id (via a hash function G). This way the dimension of the lattice associated to id can be decreased. The scheme is again parametrized by a dimension $m = O(n \lg q)$ and the following parameters. For an identity at depth $i \geq 1$,

- $i \cdot m$ is the dimension of a lattice associated with the identity;
- $\widetilde{L_i}$ is an upper bound on the Gram-Schmidt lengths of its secret short basis;
- for $i \geq 1$, s_i is the Gaussian parameter used to generate that secret basis, which must exceed $\widetilde{L_j} \cdot \omega(\sqrt{\log n})$ for all $j < i$.

These parameters, along with the total depth d of the hierarchy, determine the modulus q and error distribution χ used in the cryptosystem. As before, we instantiate all the parameters after describing the scheme. Let $H : \{0,1\}^* \to \mathbb{Z}_q^{n \times m}$ and $G : \{0,1\}^* \to \mathbb{Z}_q^n$ be hash functions.

- $\mathsf{ROHIBE.Setup}(d)$: This is the same as $\mathsf{BTE.Setup}(d)$, except that we only generate $\mathbf{A}_0$ and $\mathbf{S}_0$. More precisely, using Proposition 1, select $\mathbf{A}_0 \in \mathbb{Z}_q^{n \times m}$ that is (negligibly close to) uniform and a basis $\mathbf{S}_0$ of $\Lambda^\perp(\mathbf{A}_0)$ such that $\|\widetilde{\mathbf{S}}\| \leq \widetilde{L_0}$. Output $mpk = (\mathbf{A}_0, d)$ and $msk = \mathbf{S}_0$.

  All the remaining algorithms implicitly take the master public key mpk as an input. For an identity vector id of length $t \leq d$, we let

  $$\mathbf{A}_{id} = \mathbf{A}_0 \| \mathbf{A}_1 \| \cdots \| \mathbf{A}_{t-1} \in \mathbb{Z}_q^{n \times tm}, \quad \mathbf{y}_{id} = G(id) \in \mathbb{Z}_q^n,$$

  where $\mathbf{A}_i = H(id_1, \ldots, id_i) \in \mathbb{Z}_q^{n \times m}$. We let $pk_{id} = (\mathbf{A}_{id}, \mathbf{y}_{id})$ denote the KEM public key associated with identity vector id.
- $\mathsf{ROHIBE.Extract}(\mathbf{S}_{id}, id' = id \| \bar{id})$: if $t' = |id'| > d$, output $\perp$. Else, let $t = |id|$ and $\bar{t} = |\bar{id}|$, and choose

  $$\mathbf{S}_{id'} \leftarrow \mathsf{RandBasis}(\mathsf{ExtBasis}(\mathbf{S}_{id}, \mathbf{A}_{id'}), s_{t'}).$$

  Sample $\mathbf{e}_{id'} \leftarrow D_{\Lambda^\perp_{\mathbf{y}_{id'}}(\mathbf{A}_{id'}), s_{t'}}$ using $\mathsf{SampleD}(\mathsf{ExtBasis}(\mathbf{S}_{id}, \mathbf{A}_{id'}), \mathbf{y}_{id'}, s_{t'})$ and output $sk_{id'} = (\mathbf{S}_{id'}, \mathbf{e}_{id'})$.

  For technical reasons, we assume that the same $\mathbf{e}_{id'}$ is drawn every time this identity is used. This means that the actual algorithm should be stateless or use standard techniques like PRFs to get repeated randomness.
- $\mathsf{ROHIBE.Encaps}(id)$: Output $(k, C) \leftarrow \mathsf{KEM.Encaps}(pk_{id})$.
- $\mathsf{ROHIBE.Decaps}(sk_{id} = (\mathbf{S}_{id}, \mathbf{e}_{id}), C)$: Output $k \leftarrow \mathsf{KEM.Decaps}(\mathbf{e}_{id}, C)$.

Instantiating the parameters. A similar computation as in the last subsection shows that we can set

$$\widetilde{L_t} = \widetilde{L_0} \cdot O(\sqrt{d \cdot n \lg q})^{t-1} \cdot \omega(\sqrt{\log n})^{t-1},$$

with $\widetilde{L_0} = O(\sqrt{n \lg q})$. To ensure that the underlying KEM is complete (Lemma 5), we let $q \geq 4 s_d \cdot (d+1)m$ and $1/\alpha = s_d \cdot \sqrt{(d+1)m} \cdot \omega(\sqrt{\log n})$. For any $d = \mathrm{poly}(n)$, invoking the worst-case to average-case reduction for LWE yields an underlying approximation factor of $n \cdot \tilde{O}(d \cdot \sqrt{n})^d$.

Theorem 3 (Security of ROHIBE). *There exists a PPT oracle algorithm (a reduction) $\mathcal{S}$ attacking KEM (instantiated with dimension dm and q, χ as in ROHIBE) such that, for any adversary $\mathcal{A}$ mounting an aid-ind-cpa attack on ROHIBE making Q_H queries to the random oracle H and Q_G queries to the random oracle G,*

$$\mathbf{Adv}_{\mathit{KEM}}(\mathcal{S}^{\mathcal{A}}) \geq \mathbf{Adv}^{\mathit{aid\text{-}ind\text{-}cpa}}_{\mathit{ROHIBE}}(\mathcal{A}) / (d Q_H^{d-1} Q_G) - \mathrm{negl}(n).$$

Proof. Let $\mathcal{A}$ be an adversary mounting a aid-ind-cpa-attack on ROHIBE. We construct a reduction $\mathcal{S}$ attacking KEM. It is given a uniformly random public key $pk = (\mathbf{A}, \mathbf{y}) \in$

$\mathbb{Z}_q^{n \times dm} \times \mathbb{Z}_q^n$, an encapsulation $(\mathbf{b}, p) \in \mathbb{Z}_q^{dm} \times \mathbb{Z}_q$, and a bit k which either is encapsulated by $(\mathbf{b}, p)$ or is uniform and independent; the goal of $\mathcal{S}$ is to determine which is the case.

Let Q_G and Q_H be the number or queries that $\mathcal{A}$ issues to H and G, respectively. In our analysis, we will actually be more generous and let the adversary issue at most $d \cdot Q_H$ total queries, where it is allowed Q_H queries to H at each input length. To simplify the analysis, we also assume without loss of generality that (1) whenever $\mathcal{A}$ queries $H(id_1, \ldots, id_i)$, it has already issued the queries $H(id_1, \ldots, id_j)$ for $j < i$, and (2) that when $\mathcal{A}$ asks for sk_{id}, it has already queried $H(id)$ and $G(id)$.

$\mathcal{S}$ simulates the attack on ROHIBE to $\mathcal{A}$ as follows. First, $\mathcal{S}$ produces a master public key mpk, encapsulated key, and some secret internal state as follows:

- *Guess length of challenge identity and random oracle queries.* Choose $t^* \leftarrow [d]$, a guess for the length of the challenge identity. Pick a vector $\mathbf{j}^* = (j_1^*, \ldots, j_{t^*-1}^*) \leftarrow \{1, \ldots, Q_H\}^{t^*-1}$ and index $j \leftarrow \{1, \ldots, Q_G\}$.
- *Parsing the KEM inputs.* Parse $\mathbf{A}$ as $\mathbf{A} = \mathbf{A}_0 \| \mathbf{A}_1 \| \cdots \| \mathbf{A}_{d-1} \in \mathbb{Z}_q^{n \times dm}$ for $\mathbf{A}_0 \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{A}_i \in \mathbb{Z}_q^{n \times m}$ for all $i \in [d-1]$. Similarly, truncate $\mathbf{b}$ to $\mathbf{b}^* \in \mathbb{Z}_q^{t^* m}$.

$\mathcal{S}$ gives to $\mathcal{A}$ the master public key $mpk = (\mathbf{A}_0, d)$. To simulate the attack game for $\mathcal{A}$, $\mathcal{S}$ must simulate oracle queries to H and G, queries for user secret keys, and it must also generate the challenge encapsulation. To do this, it will maintain two lists, denoted $\mathcal{H}$ and $\mathcal{G}$, which are initialized to be empty and will store tuples of values. $\mathcal{S}$ processes queries as follows.

Queries to $H(\cdot)$. On $\mathcal{A}$'s j_i-th distinct query $(id_{j_i,1}, \ldots, id_{j_i,i})$ to $H(\cdot)$ of length i, do the following: if $i \leq t^* - 1$ and $j_i = j_i^*$, then return $\mathbf{A}_i$ (i.e., this branch undergoes *undirected growth*). Otherwise, if $i \geq t^*$ or $j_i \neq j_i^*$, run $\mathsf{GenBasis}(1^n, 1^m, q)$ to generate $\mathbf{A}_{i,j_i} \in \mathbb{Z}_q^{n \times m}$ with corresponding short basis $\mathbf{S}_{i,j_i}$ (i.e., this branch undergoes *controlled growth*). Store the tuple $((id_{j_i,1}, \ldots, id_{j_i,i}), \mathbf{A}_{i,j_i}, \mathbf{S}_{i,j_i})$ in list $\mathcal{H}$, and return $\mathbf{A}_{i,j_i}$.

Queries to $G(\cdot)$. On $\mathcal{A}$'s j-th distinct query id_j to $G(\cdot)$, do the following: if $j = j^*$ then return $\mathbf{y}$. (Recall that $\mathbf{y}$ was obtained from the KEM input.) Otherwise for $j \neq j^*$, sample $\mathbf{e}_j \leftarrow D_{\Lambda^\perp(\mathbb{Z}^m), s_t}$ (where t is the depth of id_j) and set $\mathbf{y}_j := \mathbf{A}_{(id_{j,1}, \ldots, id_{j,t-1})} \mathbf{e}_j \in \mathbb{Z}_q^n$. (Recall that we assumed $\mathcal{A}$ has already made all relevant queries to H that in particular define $\mathbf{A}_{(id_{j,1}, \ldots, id_{j,t-1})} = H(id_{j,1}, \ldots, id_{j,t-1})$.) Return $\mathbf{y}_j$ and store $(id_j, \mathbf{y}_j, \mathbf{e}_j)$ in list $\mathcal{G}$.

Queries to ROHIBE.Extract. When $\mathcal{A}$ asks for a user secret key for $id = (id_1, \ldots, id_t)$, we again assume that $\mathcal{A}$ has already made all relevant queries to G and H that define $\mathbf{y}_{id}$ and $\mathbf{A}_{id}$. If, for one $i \in [t-1]$, $\mathbf{A}_{id_i} = H(id_1, \ldots, id_i)$ is contained in list $\mathcal{H}$, then $\mathcal{B}$ can compute a properly distributed short basis $\mathbf{S}_{id}$ for $\mathbf{A}_{id}$ by running $\mathsf{RandBasis}(\mathsf{ExtBasis}(\mathbf{S}_{i,id_i}, \mathbf{A}_{id}), s_t)$, where $\mathbf{S}_{i,id_i}$ is obtained from $\mathcal{H}$. If $\mathbf{y}_{id}$ is contained in list $\mathcal{G}$, then $\mathcal{B}$ can retrieve a properly distributed vector $\mathbf{e}_{id}$ from $\mathcal{G}$ satisfying $\mathbf{A}_{(id_1, \ldots, id_{t-1})} \mathbf{e}_{id} = \mathbf{y}_{id}$. If the generation of $sk_{id} = (\mathbf{B}_{id}, \mathbf{e}_{id})$ was successful, then $\mathcal{B}$ returns sk_{id}. In all other cases, $\mathcal{B}$ aborts (and returns a random bit).

Challenge query for id^*. Let t be the depth of id^*. If $t \neq t^*$, or $G(id^*) \neq \mathbf{y}$, or $H(id_1^*, \ldots, id_i^*) \neq \mathbf{A}_i$ (for one $1 \leq i \leq t^* - 1$), then abort. Otherwise, return the encapsulation $(\mathbf{b}^*, p)$, and the key-bit k.

$\mathcal{S}$ runs until $\mathcal{A}$ halts, and it outputs whatever bit $\mathcal{A}$ outputs.

It remains to analyze the reduction. The master public key given to $\mathcal{A}$ is negligibly close to uniform by the construction of KEM and Proposition 1. By the same proposition, we have that oracle queries to H are properly simulated, up to negligible statistical distance. Oracle responses for G are negligibly far from uniform by Lemma 2 (4). It can also be verified using the truncation property of KEM that the challenge encapsulation is properly distributed, conditioned on the event that $\mathcal{S}$ does not abort. Thus all that remains to check is the probability that $\mathcal{S}$ aborts; this is at most $1/(dQ_G Q_H^{d-1})$. This completes the proof.

5.4 Full Security in the Standard Model

To achieve aid-ind-cca security in the standard model, we will essentially try to implement the random oracle from our scheme ROHIBE with a suitable hash function. We will employ a probabilistic argument, along the lines of [9]. Concretely, we will set up the public key such that in the simulation, we will know a short basis for $\mathbf{A}_{id}$ with a certain probability. A sophisticated construction of the hash function will ensure that, to a certain degree, these probabilities (resp. the corresponding events) are independent. That is, even an adversary that adaptively asks for user secret keys will not manage to produce an identity id for which the simulation is *guaranteed* to know a trapdoor. In this case, a successful simulation will be possible.

Of course, we will have to take care that the event of a successful simulation is (at least approximately) independent of the adversary's view. To achieve independence, we will employ an "artificial abort" strategy similar to the one from [56].

Admissible hash functions. We give a variant of the definition from [9]. Let $\mathcal{H} = \{\mathcal{H}_n\}$ be a collection of distributions of functions $H : \mathcal{C}_n \to \mathcal{D}_n = \{0,1\}^\lambda$. For $H \in \mathcal{H}_n$, $K \in \{0,1,\perp\}^\lambda$, and $x \in \mathcal{C}_n$, define

$$F_{K,H}(x) = \begin{cases} \mathtt{B} & \text{if } \exists u \in \{1,\dots,\lambda\} : t_u = K_i \\ \mathtt{R} & \text{if } \forall u \in \{1,\dots,\lambda\} : t_u \neq K_i \end{cases} \quad \text{for } (t_1,\dots,t_\lambda) = H(x).$$

For $\mu \in \{0,\dots,\lambda\}$, denote by $\mathcal{K}_\mu$ the uniform distribution on all keys $K \in \{0,1,\perp\}^\lambda$ with exactly μ non-$\perp$ components.

We say that $\mathcal{H}$ is *Δ-admissible* (for $\Delta : \mathbb{N}^2 \to \mathbb{R}$) if for every polynomial $Q = Q(n)$, there exists an efficiently computable function $\mu = \mu(n)$, and efficiently recognizable sets $\mathsf{bad}_H \subseteq (\mathcal{C}_n)^*$ ($H \in \mathcal{H}_n$), so that the following holds:

- For every PPT algorithm $\mathcal{C}$ that, on input a function $H \in \mathcal{H}_n$, outputs a vector $\mathbf{x} \in \mathcal{C}_n^{Q+1}$, the function

$$\mathbf{Adv}^{adm}_{\mathcal{H}}(\mathcal{C}) := \Pr[\mathbf{x} \in \mathsf{bad}_H \mid H \leftarrow \mathcal{H}_n \ ; \ \mathbf{x} \leftarrow \mathcal{C}(H)]$$

  is negligible in n.
- For every $H \in \mathcal{H}_n$ and every $\mathbf{x} = (x_0,\dots,x_Q) \in \mathcal{C}_n^{Q+1} \setminus \mathsf{bad}_H$, we have that

$$\Pr[F_{K,H}(x_0) = \mathtt{R} \wedge F_{K,H}(x_1) = \cdots = F_{K,H}(x_Q) = \mathtt{B}] \geq \Delta(n,Q),$$

  where the probability is over uniform $K \in \mathcal{K}_{\mu(n,Q)}$.

We say that $\mathcal{H}$ is *admissible* if $\mathcal{H}$ is admissible for some Δ, such that $\Delta(n,Q)$ is significant for every polynomial $Q = Q(n)$.

Difference to the definition of [9]. Note that our definition of admissibility is conceptually different from that of [9]. The reason for our change is that our definition is better suited for our purposes. Concretely, their definition is based upon indistinguishability from a (biased) random function. However, their construction only achieves asymptotic indistinguishability (i.e., negligible distinguishing success) when the "target" random function is constant. (In their notation, this corresponds to the case when γ is negligible, so that $\Pr[F_{K,H}(x) = 1] = 1 - \mathrm{negl}(n)$.) Such a function is not very useful for aymptotic purposes. In an asymptotic sense, their construction becomes only useful with parameters that cause the distinguishing advantage to become significant (but smaller than the inverse of a given polynomial). With that parameter choice, our definition allows for a conceptually simpler analysis. Namely, it separates certain negligible error probabilities (of $\mathbf{x} \in \mathsf{bad}_H$) from significant, but purely combinatorial bounds on the probability of the "simulation-enabling" event

$$\mathsf{good} := [F_{K,H}(x_0) = \mathtt{R} \wedge F_{K,H}(x_1) = \cdots = F_{K,H}(x_Q) = \mathtt{B}].$$

Specifically, we can bound $\Pr[\mathsf{good}]$ for *every* $\mathbf{x} \notin \mathsf{bad}_H$, which simplifies the artificial abort step below. Note that while the original analysis from [9] does not incorporate an artificial abort step, this actually would have been necessary to guarantee sufficient independence of (their version of) event good. This becomes an issue in [9, Claim 2], when the success probability of an adversary conditioned on good is related to its original (unconditioned) success probability.

Constructions. [9] show how to construct admissible hash functions from a given collision-resistant hash function family. Since collision-resistant hash functions can be built from the LWE problem (e.g., [5]), this does not entail extra assumptions in the encryption context. Specifically, for parameter choices as in [9, Section 5.3], we get a single hash function with output length $\lambda = O(k^{2+\varepsilon})$ (for arbitrary $\varepsilon > 0$) that is Δ-admissible with $\Delta = \Theta(1/Q^2)$.[5]

The scheme SMHIBE. Let $d \in \mathbb{N}$ denote the maximal depth of the HIBE, and fix a dimension m, as well as $\widetilde{L_i}$, s_i. Let $\mathcal{H} = (\mathcal{H}_n)_n$ be an admissible family of hash functions $H : \{0,1\}^n \to \{0,1\}^\lambda$.

$\mathsf{SMHIBE.Setup}(d)$. Using Proposition 1, generate $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a corresponding short basis $\mathbf{S} \in \mathbb{Z}^{m\times m}$ with $\|\widetilde{\mathbf{S}}\| \leq \widetilde{L_0}$. Furthermore, sample uniformly and independently matrices $\mathbf{B}_{i,u,b} \in \mathbb{Z}_q^{n\times m}$ (for $1 \leq i \leq d, 1 \leq u \leq \lambda$ and $0 \leq b \leq 1$) and a vector $\mathbf{y} \in \mathbb{Z}_q^n$. Finally, choose $H_1, \ldots, H_d \leftarrow \mathcal{H}_n$. Return

$$mpk = (\mathbf{A}, \mathbf{y}, (\mathbf{B}_{i,u,b})_{(i,u,b)\in[d]\times[\lambda]\times\{0,1\}}, (H_i)_{i=1}^d), \qquad msk = (mpk, \mathbf{S}).$$

For an identity $id = (id_1, \ldots, id_\ell)$ we define

$$\begin{aligned} \mathbf{A}_{id} &:= \mathbf{A}||\mathbf{A}_{1,id_1}||\ldots||\mathbf{A}_{\ell,id_\ell} \in \mathbb{Z}_q^{n\times(\lambda\ell+1)m} \\ \text{for} \quad \mathbf{A}_{i,id_i} &:= \mathbf{B}_{i,1,t_1}||\ldots||\mathbf{B}_{i,\lambda,t_\lambda} \in \mathbb{Z}_q^{n\times\lambda m}, \end{aligned} \tag{5.1}$$

[5] In the notation of [9], we replace the output length β_H of the original hash function with k, and bound the number Q of hash function queries by $2^{k^{\varepsilon/2}}$. Note that Q will later correspond to the number of (online) user secret key queries, so we bound Q by a comparatively small exponential function.

where $(t_1, \dots, t_\lambda) := H_i(id_i) \in \{0,1\}^\lambda$. The user secret keys for an identity id will consist of a basis part $\mathbf{S}_{id}$ for $\Lambda^\perp(\mathbf{A}_{id})$ and a syndrome part $\mathbf{e}_{id}$ satisfying $\mathbf{A}_{id}\mathbf{e}_{id} = \mathbf{y}$. For brevity, we will write $id|\ell := (id_1, \dots, id_\ell)$ for $\ell \le |id|$.

$\mathsf{SMHIBE.Extract}(msk, id)$: This algorithm computes a user secret key $(\mathbf{S}_{id}, \mathbf{e}_{id})$ for $id = (id_1, \dots, id_\ell)$, where $\mathbf{S}_{id} \leftarrow \mathsf{RandBasis}(\mathsf{ExtBasis}(\mathbf{A}_{id}, \mathbf{S}_\varepsilon), s_\ell)$ is a basis for $\Lambda^\perp(\mathbf{A}_{id})$ and $\mathbf{e}_{id} \leftarrow \mathsf{SampleD}(\mathsf{ExtBasis}(\mathbf{A}_{id}, \mathbf{S}_\varepsilon), \mathbf{y}_{id}, s_\ell)$ is distributed according to $D_{\mathbb{Z}^{(\lambda\ell+1)m}, s_\ell}$ conditioned on $\mathbf{A}_{id}\mathbf{e}_{id} = \mathbf{y}_{id}$.

$\mathsf{SMHIBE.HIBEDel}(usk_{id|\ell-1}, id)$: The delegation algorithm derives a user secret key for an identity $id = (id_1, \dots, id_\ell)$ $(1 \le \ell \le d)$ given a user secret key for $id|\ell - 1$ which contains a basis $\mathbf{S}_{id|\ell-1}$ for $\Lambda^\perp(\mathbf{A}_{id|\ell-1})$ with $\|\widetilde{\mathbf{S}}_{id|\ell-1}\| \le L(\ell-1)$. (We note that the short vector $\mathbf{e}_{id|\ell-1}$ is not needed for delegation.) Note that $\mathbf{A}_{id} = \mathbf{A}||\mathbf{A}_{1,id_1}|| \cdots ||\mathbf{A}_{\ell,id_\ell} = \mathbf{A}_{1,id|\ell-1}||\mathbf{A}_{\ell,id_\ell} \in \mathbb{Z}_q^{n\times(\lambda\ell+1)m}$. To compute the basis part, run $\mathbf{S}_{id} \leftarrow \mathsf{RandBasis}(\mathsf{ExtBasis}(\mathbf{A}_{id}, \mathbf{S}_{id|\ell-1}), s_\ell)$. Note that since ℓ is constant,

$$L(\ell) = L(\ell-1)\cdot\sqrt{\lambda m}\cdot\omega(\sqrt{\log \lambda m}) \ge \|\widetilde{\mathbf{S}}_{id|\ell-1}\| \cdot \sqrt{(\lambda\ell+1)m}\cdot\omega(\sqrt{\log{(\lambda\ell+1)m}}).$$

The syndrome part of the user secret key is computed as

$$\mathbf{e}_{id} \leftarrow \mathsf{SampleD}(\mathsf{ExtBasis}(\mathbf{A}_{id}, \mathbf{S}_{id|\ell-1}), \mathbf{y}, s_\ell).$$

By Lemma 4, the user secret key $usk_{id} = (\mathbf{S}_{id}, \mathbf{e}_{id})$ has a distribution that is statistically close to the one computed by Extract.

$\mathsf{SMHIBE.Encaps}(id, b)$: Output $C = (k, \mathbf{p}) \leftarrow \mathsf{KEM.Encaps}(pk = (\mathbf{A}_{id}, \mathbf{y}))$.

$\mathsf{SMHIBE.Decaps}(sk_{id}, (\mathbf{S}_{id}, \mathbf{e}_{id}), C)$: Output $k \leftarrow \mathsf{KEM.Decaps}(\mathbf{e}_{id}, C)$.

The scheme's correctness is inherited by that of KEM.

Security of SMHIBE. We now formally state security of our construction. If the hash function $\mathcal{H}$ is admissible, then we can prove the scheme aid-ind-cpa secure. Unfortunately, we only know constructions of admissible hash functions that require $\lambda = n^{2+\varepsilon}$ so the resulting scheme is not very practical.

Theorem 4. *Assume an adversary $\mathcal{A}$ on* SM-HIBE*'s aid-ind-cpa security that makes at most $Q(n)$ user secret key queries. Then, for every polynomial $S = S(n)$, there exists an $\mathsf{LWE}_{q,\chi}$-distinguisher $\mathcal{D}$ and an adversary $\mathcal{C}$ on $\mathcal{H}$'s admissibility such that*

$$\mathbf{Adv}^{\text{aid-ind-cpa}}_{\text{SM-HIBE}}(\mathcal{A}) \le d \cdot \mathbf{Adv}^{adm}_{\mathcal{H}}(\mathcal{C}) + \frac{\mathbf{Adv}_{\mathsf{LWE}_{q,\chi}}(\mathcal{D})}{\Delta(n,Q)^d} + \frac{1}{S(n)} + \text{negl}(n). \quad (5.2)$$

Here, the running time of $\mathcal{C}$ is roughly that of the aid-ind-cpa experiment with $\mathcal{A}$, and the running time of $\mathcal{D}$ is roughly that of the aid-ind-cpa experiment with $\mathcal{A}$ plus $O(n^2QS/\Delta^d)$ steps.

Note that for the admissible hash function from [9], $\Delta(n,Q)^d = \Theta(1/Q^{2d})$ is significant. Since S in Theorem 4 is arbitrary, we obtain:

Corollary 1 (SM-HIBE **is aid-ind-cpa secure).** *If $\mathcal{H}$ is admissible, and if the* $\mathsf{LWE}_{q,\chi}$ *problem is hard, then* SM-HIBE *is CPA secure.*

We defer a proof of Theorem 4 to the full version.

Acknowledgments

We thank the anonymous Eurocrypt reviewers for their helpful comments, and for pointing out a small error in an earlier formulation of RandBasis.

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); Preliminary version in CRYPTO 2005

[2] Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: EUROCRYPT 2010 (to appear, 2010)

[3] Agrawal, S., Boyen, X.: Identity-based encryption from lattices in the standard model (July 2009) (manuscript)

[4] 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)

[5] Ajtai, M.: Generating hard instances of lattice problems. Quaderni di Matematica 13, 1–32 (2004); Preliminary version in STOC 1996

[6] Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. In: STACS, pp. 75–86 (2009)

[7] 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)

[8] 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)

[9] 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)

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

[11] Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004)

[12] Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. SIAM J. Comput. 32(3), 586–615 (2003); Preliminary version in CRYPTO 2001

[13] Boneh, D., Gentry, C., Hamburg, M.: Space-efficient identity based encryption without pairings. In: FOCS, pp. 647–657 (2007)

[14] Boyen, X., Niçoises, L., Trapdoors, V.: A Framework for Fully Secure Short Signatures and more. In: PKC 2010 (to appear, 2010)

[15] 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)

[16] 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

[17] Cash, D., Hofheinz, D., Kiltz, E.: How to delegate a lattice basis. Cryptology ePrint Archive, Report 2009/351 (July 2009), http://eprint.iacr.org/

[18] Cocks, C.: An identity based encryption scheme based on quadratic residues. In: IMA Int. Conf., pp. 360–363 (2001)

[19] Cramer, R., Shoup, V.: Signature schemes based on the strong RSA assumption. ACM Trans. Inf. Syst. Secur. 3(3), 161–185 (2000); Preliminary version in CCS 1999

[20] Di Crescenzo, G., Saraswat, V.: Public key encryption with searchable keywords based on Jacobi symbols. In: Srinathan, K., Rangan, C.P., Yung, M. (eds.) INDOCRYPT 2007. LNCS, vol. 4859, pp. 282–296. Springer, Heidelberg (2007)

[21] Dodis, Y., Fazio, N.: Public key broadcast encryption for stateless receivers. In: ACM Workshop on Digital Rights Management, pp. 61–80 (2002)

[22] 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)

[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., 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)

[25] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC, pp. 197–206 (2008)

[26] Gentry, C., Silverberg, A.: Hierarchical ID-based cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002)

[27] Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reduction problems. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 112–131. Springer, Heidelberg (1997)

[28] 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); Preliminary version in FOCS 1984

[29] 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)

[30] Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: Cohen, H. (ed.) ANTS 1996. LNCS, vol. 1122, pp. 267–288. Springer, Heidelberg (1996)

[31] 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)

[32] 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)

[33] Horwitz, J., Lynn, B.: Toward hierarchical identity-based encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 466–481. Springer, Heidelberg (2002)

[34] Krawczyk, H., Rabin, T.: Chameleon signatures. In: NDSS (2000)

[35] Leurent, G., Nguyen, P.Q.: How risky is the random-oracle model? In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 445–464. Springer, Heidelberg (2009)

[36] 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)

[37] 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)

[38] Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: EUROCRYPT (to appear, 2010)

[39] Micciancio, D.: Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. Computational Complexity 16(4), 365–411 (2007); Preliminary version in FOCS 2002

[40] Micciancio, D., Goldwasser, S.: Complexity of Lattice Problems: a cryptographic perspective. The Kluwer International Series in Engineering and Computer Science, vol. 671. Kluwer Academic Publishers, Boston (2002)

[41] Micciancio, D., Regev, O.: Worst-case to average-case reductions based on Gaussian measures. SIAM J. Comput. 37(1), 267–302 (2007); Preliminary version in FOCS 2004

[42] Micciancio, D., Warinschi, B.: A linear space algorithm for computing the Hermite normal form. In: ISSAC, pp. 231–236 (2001)

[43] Naor, M., Yung, M.: Universal one-way hash functions and their cryptographic applications. In: STOC, pp. 33–43 (1989)

[44] Peikert, C.: Bonsai trees (or, arboriculture in lattice-based cryptography). Cryptology ePrint Archive, Report 2009/359 (July 2009), http://eprint.iacr.org/

[45] Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem. In: STOC, pp. 333–342 (2009)

[46] Peikert, C.: An efficient and parallel Gaussian sampler for lattices (2010) (manuscript)

[47] 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)

[48] Peikert, C., Rosen, A.: Lattices that admit logarithmic worst-case to average-case connection factors. In: STOC, pp. 478–487 (2007)

[49] 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)

[50] Rabin, M.O.: Digitalized signatures and public-key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, MIT Laboratory for Computer Science (1979)

[51] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM 56(6) (2009); Preliminary version in STOC 2005

[52] Rückert, M.: Strongly unforgeable signatures and hierarchical identity-based signatures from lattices without random oracles. In: PQCrypto (to appear, 2010)

[53] 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)

[54] Shamir, A., Tauman, Y.: Improved online/offline signature schemes. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 355–367. Springer, Heidelberg (2001)

[55] 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)

[56] Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)

[57] 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)

[58] Yao, D., Fazio, N., Dodis, Y., Lysyanskaya, A.: ID-based encryption for complex hierarchies with applications to forward security and broadcast encryption. In: ACM Conference on Computer and Communications Security, pp. 354–363 (2004)

Efficient Lattice (H)IBE in the Standard Model*

Shweta Agrawal[1], Dan Boneh[2,**], and Xavier Boyen[3]

[1] University of Texas, Austin
[2] Stanford University
[3] Université de Liège, Belgium

Abstract. We construct an efficient identity based encryption system based on the standard learning with errors (LWE) problem. Our security proof holds in the standard model. The key step in the construction is a family of lattices for which there are two distinct trapdoors for finding short vectors. One trapdoor enables the real system to generate short vectors in all lattices in the family. The other trapdoor enables the simulator to generate short vectors for all lattices in the family except for one. We extend this basic technique to an adaptively-secure IBE and a Hierarchical IBE.

1 Introduction

Identity-Based Encryption (IBE) provides a public-key encryption mechanism where a public key is an arbitrary string such as an email address or a telephone number. The corresponding private key can only be generated by a Private-Key Generator (PKG) who has knowledge of a master secret. Identity-based encryption was first proposed by Shamir [28], however, it is only recently that practical implementations were proposed. Boneh and Franklin [8] define a security model for identity-based encryption and give a construction based on the Bilinear Diffie-Hellman (BDH) problem. Cocks [13] describes a construction using quadratic residues modulo a composite (see also [9]) and Gentry et al. [16] give a construction using lattices. The security of all these systems requires cryptographic hash functions that are modeled as random oracles.

For pairing-based systems, the structure of pairing groups enabled several secure IBE systems in the standard model [11,6,7,31,17,32]. For systems based on quadratic residuosity it is still not known how to build a secure IBE in the standard model.

In this paper we focus on lattice-based IBE. Cash et al. [12], Peikert [24] and Agrawal et al. [3] recently showed how to construct secure IBE in the standard model from the learning with errors (LWE) problem [27]. Their constructions view an identity as a sequence of bits and then assign a matrix to each bit. The resulting systems, while quite elegant, are considerably less efficient than the underlying random-oracle system of [16] on which they are built.

[*] A full version of this paper is available at [1].

[**] Supported by NSF and the Packard Foundation.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 553–572, 2010.

1.1 Our Results

We construct a lattice-based IBE in the standard model whose performance is comparable to the performance of the random-oracle system from [16]. In particular, we process identities as one chunk rather than bit-by-bit resulting in lattices whose dimension is similar to those in the random oracle system.

Lattices in our system are built from two parts called "right" and "left" lattices. A trapdoor for the left lattice is used as the master secret in the real system and enables one to generate private keys for all identities. A trapdoor for the right lattice is only used in the proof of selective security and enables the simulator to generate private keys for all identities except for one. We use a "low norm" randomization matrix R to ensure that an attacker cannot distinguish between the real world and a simulation.

In pairing-based IBE systems one uses large groups G and therefore identities can be encoded as integers in the range $1 \dots |G|$. In contrast, lattice systems are typically defined over a relatively small field $\mathbb{Z}_q$ and consequently encoding identities as integers in $1 \dots q$ would result in too few identities for the system. Instead, we represent identities as matrices in $\mathbb{Z}_q^{n \times n}$ for some n. More precisely, we represent identities as elements in $\mathbb{Z}_q^n$ (for a total of q^n identities) and then use an encoding function $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$ to map identities to matrices. Our security proof requires that for all $\mathsf{id}_1 \neq \mathsf{id}_2$ the matrix $H(\mathsf{id}_1) - H(\mathsf{id}_2) \in \mathbb{Z}_q^{n \times n}$ is invertible. We present an encoding function H that has this property and expect this encoding to be useful in other lattice-based constructions. A similar function H was developed by Cramer and Damgard [14] in an entirely different context.

Full IBE. In the full version of the paper [1] we show that our base construction extends to an adaptively-secure IBE using a lattice analog of the Waters IBE [31]. Our base construction requires that the underlying field $\mathbb{Z}_q$ satisfy $q > Q$ where Q is the number of private key queries issued by the adversary. This requirement can be relaxed using the framework of Boyen [10].

Hierarchical IBE (HIBE). In the full version of the paper [1] we show how to extend our base IBE to an HIBE using the basis delegation technique from [12,24]. The construction assigns a matrix to each level of the hierarchy and the resulting lattice dimension is linear in the recipient identity's depth. Since we do not process identities bit-by-bit we obtain an efficient HIBE where the lattice dimension is much smaller than in [12,24]. We note that a recent result of [2] uses a different basis delegation mechanism to construct an improved HIBE where the lattice dimension is fixed for the entire hierarchy.

2 Preliminaries

Notation. Throughout the paper we say that a function $\epsilon : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is negligible if $\epsilon(n)$ is smaller than all polynomial fractions for sufficiently large n. We say that an event happens with overwhelming probability if it happens with probability at least $1 - \epsilon(n)$ for some negligible function ϵ. We say that integer vectors $v_1, \ldots, v_n \in \mathbb{Z}^m$ are $\mathbb{Z}_q$-linearly independent if they are linearly independent when reduced modulo q.

2.1 IBE and Hierarchical IBE

Recall that an Identity-Based Encryption system (IBE) consists of four algorithms [28,8]: Setup, Extract, Encrypt, Decrypt. The Setup algorithm generates system parameters, denoted by PP, and a master key MK. The Extract algorithm uses the master key to extract a private key corresponding to a given identity. The encryption algorithm encrypts messages for a given identity (using the system parameters) and the decryption algorithm decrypts ciphertexts using the private key.

In a Hierarchical IBE [20,18], identities are vectors, and there is a fifth algorithm called Derive. A vector of dimension ℓ represents an identity at depth ℓ. Algorithm Derive takes as input an identity $\mathsf{id} = (\mathsf{I}_1, \ldots, \mathsf{I}_\ell)$ at depth ℓ and the private key $\mathsf{SK}_{\mathsf{id}|\ell-1}$ of the parent identity $\mathsf{id}_{|\ell-1} = (\mathsf{I}_1, \ldots, \mathsf{I}_{\ell-1})$ at depth $\ell - 1 \geq 0$. It outputs the private key $\mathsf{SK}_{\mathsf{id}}$ for identity id. We sometimes refer to the master key as the private key at depth 0, given which the algorithm Derive performs the same function as Extract. The Setup algorithm in an HIBE scheme takes the maximum depth of the hierarchy as input.

Selective and Adaptive ID Security. The standard IBE security model of [8] defines the indistinguishability of ciphertexts under an adaptive chosen-ciphertext and chosen-identity attack (IND-ID-CCA2). A weaker notion of IBE security given by Canetti, Halevi, and Katz [11] forces the adversary to announce ahead of time the public key it will target, which is known as a selective-identity attack (IND-sID-CCA2).

As with regular public-key encryption, we can deny the adversary the ability to ask decryption queries (for the target identity), which leads to the weaker notions of indistinguishability of ciphertexts under an adaptive chosen-identity chosen-plaintext attack (IND-ID-CPA) and under a selective-identity chosen-plaintext attack (IND-sID-CPA) respectively.

Security Game. We define IBE and HIBE selective security using a game that captures a strong privacy property called *indistinguishable from random* which means that the challenge ciphertext is indistinguishable from a random element in the ciphertext space. This property implies both semantic security and recipient anonymity, and also implies that the ciphertext hides the public parameters (PP) used to create it. This can make the IBE more resistant to subpoenas since an observer cannot tell from the ciphertext which authority holds the corresponding master secret. For a security parameter λ, we let $\mathcal{M}_\lambda$ denote the message space and let $\mathcal{C}_\lambda$ denote the ciphertext space. The game, for a hierarchy of maximum depth d, proceeds as follows.

Init: The adversary is given the maximum depth of the hierarchy d and outputs a target identity $\mathsf{id}^* = (\mathsf{I}_1^*, \ldots, \mathsf{I}_k^*), k \leq d$.

Setup: The challenger runs $\mathsf{Setup}(1^\lambda, 1^d)$ (where $d = 1$ for IBE) and gives the adversary the resulting system parameters PP. It keeps the master key MK to itself.

Phase 1: The adversary issues queries $q_1, \ldots, q_m$ where the i-th query q_i is a query on id_i, where $\mathsf{id}_i = (\mathsf{I}_1, \ldots, \mathsf{I}_u)$ for some $u \le d$. We require that id_i is not a prefix of id^*, (i.e., it is not the case that $u \le k$ and $\mathsf{I}_i = \mathsf{I}_i^*$ for all $i = 1, \ldots, u$). The challenger responds by running algorithm $\mathsf{Extract}$ to obtain a private key d_i for the public key id_i. It sends d_i to the adversary. All queries may be made adaptively, that is, the adversary may ask q_i with knowledge of the challenger's responses to $q_1, \ldots, q_{i-1}$.

Challenge: Once the adversary decides that Phase 1 is over it outputs a plaintext $M \in \mathcal{M}_\lambda$ on which it wishes to be challenged. The challenger picks a random bit $r \in \{0, 1\}$ and a random ciphertext $C \in \mathcal{C}_\lambda$. If $r = 0$ it sets the challenge ciphertext to $C^* := \mathsf{Encrypt}(\mathsf{PP}, \mathsf{id}^*, M)$. If $r = 1$ it sets the challenge ciphertext to $C^* := C$. It sends C^* as the challenge to the adversary.

Phase 2: The adversary issues additional adaptive queries $q_{m+1}, \ldots, q_n$ where q_i is a private-key extraction query on id_i, where id_i is not a prefix of id^*. The challenger responds as in Phase 1.

Guess: Finally, the adversary outputs a guess $r' \in \{0, 1\}$ and wins if $r = r'$.

We refer to such an adversary $\mathcal{A}$ as an $\mathsf{INDr\text{–}sID\text{-}CPA}$ adversary. We define the advantage of the adversary $\mathcal{A}$ in attacking an IBE or HIBE scheme $\mathcal{E}$ as

$$\mathrm{Adv}_{d,\mathcal{E},\mathcal{A}}(\lambda) = \left| \Pr[r = r'] - 1/2 \right|$$

The probability is over the random bits used by the challenger and the adversary.

Definition 1. *We say that an IBE or a depth d HIBE system $\mathcal{E}$ is selective-identity, indistinguishable from random if for all $\mathsf{INDr\text{–}sID\text{-}CPA}$ PPT adversaries $\mathcal{A}$ we have that $Adv_{d,\mathcal{E},\mathcal{A}}(\lambda)$ is a negligible function. We abbreviate this by saying that $\mathcal{E}$ is $\mathsf{INDr\text{–}sID\text{-}CPA}$ secure for depth d.*

2.2 Statistical Distance

Let X and Y be two random variables taking values in some finite set Ω. Define the *statistical distance*, denoted $\Delta(X; Y)$, as

$$\Delta(X; Y) := \frac{1}{2} \sum_{s \in \Omega} \left| \Pr[X = s] - \Pr[Y = s] \right|$$

We say that X is δ-uniform over Ω if $\Delta(X; U_\Omega) \le \delta$ where U_Ω is a uniform random variable over Ω.

Let $X(\lambda)$ and $Y(\lambda)$ be ensembles of random variables. We say that X and Y are statistically close if $d(\lambda) := \Delta(X(\lambda); Y(\lambda))$ is a negligible function of λ.

2.3 Integer Lattices

Let $B = \left[\, b_1 \,\middle|\, \ldots \,\middle|\, b_m \,\right] \in \mathbb{R}^{m \times m}$ be an $m \times m$ matrix whose columns are linearly independent vectors $b_1, \ldots, b_m \in \mathbb{R}^m$. The m-dimensional full-rank lattice Λ generated by B is the set,

$$\Lambda = \mathcal{L}(B) = \left\{ \; y \in \mathbb{R}^m \quad \text{s.t.} \quad \exists s \in \mathbb{Z}^m \;, \;\; y = B\, s = \sum_{i=1}^{m} s_i\, b_i \; \right\}$$

Here, we are interested in integer lattices, i.e, when L is contained in $\mathbb{Z}^m$.

Definition 2. *For q prime, $A \in \mathbb{Z}_q^{n \times m}$ and $u \in \mathbb{Z}_q^n$, define:*

$$\begin{aligned} \Lambda_q(A) &:= \left\{ \; e \in \mathbb{Z}^m \quad s.t. \quad \exists s \in \mathbb{Z}_q^n \;\; where \;\; A^\top s = e \pmod q \; \right\} \\ \Lambda_q^\perp(A) &:= \left\{ \; e \in \mathbb{Z}^m \quad s.t. \quad A\, e = 0 \pmod q \; \right\} \\ \Lambda_q^u(A) &:= \left\{ \; e \in \mathbb{Z}^m \quad s.t. \quad A\, e = u \pmod q \; \right\} \end{aligned}$$

Observe that if $t \in \Lambda_q^u(A)$ then $\Lambda_q^u(A) = \Lambda_q^\perp(A) + t$ and hence $\Lambda_q^u(A)$ is a shift of $\Lambda_q^\perp(A)$.

2.4 The Gram-Schmidt Norm of a Basis

Let S be a set of vectors $S = \{s_1, \ldots, s_k\}$ in $\mathbb{R}^m$. We use the following notation:

- $\|S\|$ denotes the L_2 length of the longest vector in S, i.e. $\|S\| := \max_i \|s_i\|$ for $1 \le i \le k$.
- $\tilde{S} := \{\tilde{s}_1, \ldots, \tilde{s}_k\} \subset \mathbb{R}^m$ denotes the Gram-Schmidt orthogonalization of the vectors $s_1, \ldots, s_k$ taken in that order.

We refer to $\|\widetilde{S}\|$ as the Gram-Schmidt norm of S.

Micciancio and Goldwassser [22] showed that a full-rank set S in a lattice Λ can be converted into a basis T for Λ with an equally low Gram-Schmidt norm.

Lemma 1 ([22, Lemma 7.1]). *Let Λ be an m-dimensional lattice. There is a deterministic polynomial-time algorithm that, given an arbitrary basis of Λ and a full-rank set $S = \{s_1, \ldots, s_m\}$ in Λ, returns a basis T of Λ satisfying*

$$\|\widetilde{T}\| \le \|\widetilde{S}\| \quad and \quad \|T\| \le \|S\|\sqrt{m}/2$$

Ajtai [4] showed how to sample an essentially uniform matrix $A \in \mathbb{Z}_q^{n \times m}$ with an associated basis S_A of $\Lambda_q^\perp(A)$ with low Gram-Schmidt norm. We use an improved sampling algorithm from Alwen and Peikert [5]. The following follows from Theorem 3.2 of [5] taking $\delta := 1/3$.

Theorem 1. *Let $q \geq 3$ be odd and $m := \lceil 6n \log q \rceil$.*
There is a probabilistic polynomial-time algorithm $\mathsf{TrapGen}(q, n)$ that outputs a pair ($A \in \mathbb{Z}_q^{n \times m}$, $S \in \mathbb{Z}^{m \times m}$) such that A is statistically close to a uniform matrix in $\mathbb{Z}_q^{n \times m}$ and S is a basis for $\Lambda_q^{\perp}(A)$ satisfying

$$\|\widetilde{S}\| \leq O(\sqrt{n \log q}\,) \quad \textit{and} \quad \|S\| \leq O(n \log q)$$

with all but negligible probability in n.

We will also need the following simple lemma about the effect of matrix multiplication on the Gram-Schmidt norm.

Lemma 2. *Let R be a matrix in $\mathbb{R}^{\ell \times m}$ and $S = \{s_1, \ldots, s_k\} \subset \mathbb{R}^m$ a linearly independent set. Let $S_R := \{Rs_1, \ldots, Rs_k\}$. Then*

$$\|\widetilde{S_R}\| \leq \max_{1 \leq i \leq k} \|R\tilde{s}_i\|$$

Proof. We show that for all $i = 1, \ldots, k$ the i-th Gram-Schmidt vector of S_R has L_2 norm less than $\|R\tilde{s}_i\|$. This will prove the lemma.

For $i \in \{1, \ldots, k\}$ let $V := \mathrm{span}_{\mathbb{R}}(Rs_1, \ldots, Rs_{i-1})$. Set $v := s_i - \tilde{s}_i$. Then $v \in \mathrm{span}_{\mathbb{R}}(s_1, \ldots, s_{i-1})$ and therefore $Rv \in V$. Let u be the projection of $R\tilde{s}_i$ on V and let $z := R\tilde{s}_i - u$. Then z is orthogonal to V and

$$Rs_i = Rv + R\tilde{s}_i = Rv + u + z = (Rv + u) + z \ .$$

By construction, $Rv + u \in V$ and hence, since z is orthogonal to V, this z must be the i-th Gram-Schmidt vector of S_R. Since z is the projection of $R\tilde{s}_i$ on $V^{\perp}$ we obtain that $\|z\| \leq \|R\tilde{s}_i\|$. Hence, for all $i = 1, \ldots, k$ the i-th Gram-Schmidt vector of S_R has L_2 norm less than $\|R\tilde{s}_i\|$ which proves the lemma. □

2.5 Discrete Gaussians

Let L be a subset of $\mathbb{Z}^m$. For any vector $c \in \mathbb{R}^m$ and any positive parameter $\sigma \in \mathbb{R}_{>0}$, define:

$\rho_{\sigma,c}(x) \ = \exp\left(-\pi \frac{\|x-c\|^2}{\sigma^2}\right)$: a Gaussian-shaped function on $\mathbb{R}^m$ with center c and parameter σ,

$\rho_{\sigma,c}(L) \ = \sum_{x \in L} \rho_{\sigma,c}(x)$: the (always converging) sum of $\rho_{\sigma,c}$ over L,

$\mathcal{D}_{L,\sigma,c}$: the discrete Gaussian distribution over L with parameters σ and c,

$$\forall y \in L \quad , \quad \mathcal{D}_{L,\sigma,c}(y) = \frac{\rho_{\sigma,c}(y)}{\rho_{\sigma,c}(L)}$$

We abbreviate $\rho_{\sigma,0}$ and $\mathcal{D}_{L,\sigma,0}$ as ρ_σ and $\mathcal{D}_{L,\sigma}$. We write ρ to denote ρ_1. The distribution $\mathcal{D}_{L,\sigma,c}$ will most often be defined over the lattice $L = \Lambda_q^{\perp}(A)$ for a matrix $A \in \mathbb{Z}_q^{n \times m}$ or over a coset $L = t + \Lambda_q^{\perp}(A)$ where $t \in \mathbb{Z}^m$.

Properties. The following lemma from [24] captures standard properties of these distributions. The first two properties follow from Lemma 4.4 of [23] and Corollary 3.16 of [27] respectively (using Lemma 3.1 from [16] to bound the smoothing parameter). We state in property (2) a stronger version of Regev's Corollary 3.16 found in [2]. The last two properties are algorithms from [16].

Lemma 3. *Let $q \geq 2$ and let A be a matrix in $\mathbb{Z}_q^{n \times m}$ with $m > n$. Let T_A be a basis for $\Lambda_q^{\perp}(A)$ and $\sigma \geq \|\widetilde{T_A}\|\, \omega(\sqrt{\log m}\,)$. Then for $c \in \mathbb{R}^m$ and $u \in \mathbb{Z}_q^n$:*

1. $\Pr\left[\, x \sim \mathcal{D}_{\Lambda_q^{\perp}(A),\sigma} \;:\; \|x\| > \sqrt{m}\,\sigma \,\right] \;\leq\; \mathrm{negl}(n)$.
2. *A set of $O(m \log m)$ samples from $\mathcal{D}_{\Lambda_q^{\perp}(A),\sigma}$ contains a full rank set in $\mathbb{Z}^m$, except with negligible probability.*
3. *There is a PPT algorithm $\mathsf{SampleGaussian}(A, T_A, \sigma, c)$ that returns $x \in \Lambda_q^{\perp}(A)$ drawn from a distribution statistically close to $\mathcal{D}_{\Lambda,\sigma,c}$.*
4. *There is a PPT algorithm $\mathsf{SamplePre}(A, T_A, u, \sigma)$ that returns $x \in \Lambda_q^{u}(A)$ sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^{u}(A),\sigma}$.*

Recall that if $\Lambda_q^u(A)$ is not empty then $\Lambda_q^u(A) = t + \Lambda_q^{\perp}(A)$ for some $t \in \Lambda_q^u(A)$. Algorithm $\mathsf{SamplePre}(A, T_A, u, \sigma)$ works by calling $\mathsf{SampleGaussian}(A, T_A, \sigma, t)$ and subtracts t from the result.

2.6 The LWE Hardness Assumption

Security of all our constructions reduces to the LWE (learning with errors) problem, a classic hard problem on lattices defined by Regev [27].

Definition 3. *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 $s \in \mathbb{Z}_q^n$, or, a truly random sampler $\mathcal{O}_\$$, whose behaviors are respectively as follows:*

$\mathcal{O}_s$: *outputs samples of the form $(u_i, v_i) = \big(u_i,\ u_i^T\, s + x_i\big) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where, $s \in \mathbb{Z}_q^n$ is a uniformly distributed persistent value invariant across invocations, $x_i \in \mathbb{Z}_q$ is a fresh sample from χ, and u_i is uniform in $\mathbb{Z}_q^n$.*

$\mathcal{O}_\$$: *outputs truly uniform random samples from $\mathbb{Z}_q^n \times \mathbb{Z}_q$.*

The $(\mathbb{Z}_q, n, \chi)$-LWE problem allows repeated queries to the challenge oracle $\mathcal{O}$. We say that an algorithm $\mathcal{A}$ decides the $(\mathbb{Z}_q, n, \chi)$-LWE problem if $\big|\Pr[\mathcal{A}^{\mathcal{O}_s} = 1] - \Pr[\mathcal{A}^{\mathcal{O}_\$} = 1]\big|$ is non-negligible for a random $s \in \mathbb{Z}_q^n$.

Regev [27] shows that for certain noise distributions χ, denoted $\overline{\Psi}_\alpha$, the LWE problem is as hard as the worst-case SIVP and GapSVP under a quantum reduction (see also [25]).

Definition 4. *Consider a real parameter $\alpha = \alpha(n) \in (0, 1)$ and a prime q. 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 $x \in \mathbb{R}$. We denote by $\overline{\Psi}_\alpha$ the discrete distribution over $\mathbb{Z}_q$ of the random variable $\lfloor q\,X \rceil \bmod q$ where the random variable $X \in \mathbb{T}$ has distribution Ψ_α.*

Theorem 2 ([27]). *If there exists an efficient, possibly quantum, algorithm for deciding the $(\mathbb{Z}_q, n, \overline{\Psi}_\alpha)$-LWE problem for $q > 2\sqrt{n}/\alpha$ then there exists an efficient quantum algorithm for approximating the SIVP and GapSVP problems, to within $\tilde{O}(n/\alpha)$ factors in the ℓ_2 norm, in the worst case.*

If we assume the hardness of approximating the SIVP or GapSVP problems in lattices of dimension n to within approximation factors that are polynomial in n, then it follows from Lemma 2 that deciding the LWE problem is hard when n/α is polynomial in n.

3 Randomness Extraction

We will need the following lemma which follows directly from a generalization of the left over hash lemma due to Dodis et al. [15].

Lemma 4. *Suppose that $m > (n+1)\log_2 q + \omega(\log n)$ and that q is prime. Let A, B be matrices chosen uniformly in $\mathbb{Z}_q^{n \times m}$ and let R be an $m \times m$ matrix chosen uniformly in $\{1, -1\}^{m \times m} \bmod q$. Then, for all vectors w in $\mathbb{Z}_q^m$, the distribution $(A,\ AR,\ R^\top w)$ is statistically close to the distribution $(A,\ B,\ R^\top w)$.*

To prove the lemma recall that for a prime q the family of hash functions $h_A : \mathbb{Z}_q^m \to \mathbb{Z}_q^n$ for $A \in \mathbb{Z}_q^{n \times m}$ defined by $h_A(x) = A\,x$ is universal. Therefore, when the columns of R are sampled independently and have sufficient entropy, the left over hash lemma (e.g. as stated in [29, Theorem 8.38]) shows that the distributions $(A,\ AR)$ and $(A,\ B)$ are statistically close. A generalization by Dodis et al. [15] (Lemma 2.2b and 2.4) shows that the same holds even if some small amount of information about R is leaked. In our case $R^\top w$ is leaked which is precisely the settings of Dodis et al. We provide the complete proof of Lemma 4 in the full version of the paper [1].

3.1 Random Subset Sums

We will also need the following simple lemma.

Lemma 5. *Let R be an $m \times m$ matrix chosen at random from $\{-1, 1\}^{m \times m}$. Then for all vectors $u \in \mathbb{R}^m$ we have*

$$\Pr\left[\ \|Ru\| > \|u\|\,\sqrt{m} \cdot \omega(\sqrt{\log m})\ \right] < \text{negl}(m)\ .$$

Proof. Let $r \in \{-1,1\}^m$ be a row vector of the matrix R. Then $r \cdot u$ can be written as $r^\top u = \sum_{i=1}^m x_i$ where $x_i = r_i u_i$. We know that $E[x_i] = 0$ and that $x_i \in [-u_i, u_i]$ for all $i = 1, \ldots, m$. Then, by the Hoeffding bound [19, Theorem 2] we obtain that

$$\Pr\left[\ |r \cdot u| > \|u\|\ \omega(\sqrt{\log m})\ \right] < \operatorname{negl}(m)$$

The lemma now follows since an m-vector whose entries are less than some bound B has L_2 norm less than $\sqrt{m}B$. □

4 Sampling Algorithms

Let A and B be matrices in $\mathbb{Z}_q^{n\times m}$ and let R be a matrix in $\{-1,1\}^{m\times m}$. Our construction makes use of matrices of the form $F = (A \mid AR + B) \in \mathbb{Z}_q^{n\times 2m}$ and we will need to sample short vectors in $\Lambda_q^u(F)$ for some u in $\mathbb{Z}_q^n$. We show that this can be done using either a trapdoor for $\Lambda_q^\perp(A)$ or a trapdoor $\Lambda_q^\perp(B)$. More precisely, we define two algorithms:

1. $\mathsf{SampleLeft}$ takes a basis for $\Lambda_q^\perp(A)$ (the left side of F) and outputs a short vector $e \in \Lambda_q^u(F)$.
2. $\mathsf{SampleRight}$ takes a basis for $\Lambda_q^\perp(B)$ (the right side of F) and outputs a short vector $e \in \Lambda_q^u(F)$.

We will show that, with appropriate parameters, the distributions on e produced by these two algorithms are statistically indistinguishable.

4.1 Algorithm SampleLeft

Algorithm $\mathsf{SampleLeft}(A, M_1, T_A, u, \sigma)$:

Inputs:

$$\begin{array}{l} \text{a rank } n \text{ matrix } A \text{ in } \mathbb{Z}_q^{n\times m} \text{ and a matrix } M_1 \text{ in } \mathbb{Z}_q^{n\times m_1}, \\ \text{a ``short'' basis } T_A \text{ of } \Lambda_q^\perp(A) \text{ and a vector } u \in \mathbb{Z}_q^n, \\ \text{a gaussian parameter } \sigma > \|\widetilde{T_A}\| \cdot \omega(\sqrt{\log(m+m_1)}). \end{array} \qquad (1)$$

Output: Let $F_1 := (A \mid M_1)$. The algorithm outputs a vector $e \in \mathbb{Z}^{m+m_1}$ sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^u(F_1),\sigma}$. In particular, $e \in \Lambda_q^u(F_1)$.

The algorithm appears in Theorem 3.4 in [12] and also in the signing algorithm in [24]. For completeness, we briefly review the algorithm.

1. sample a random vector $e_2 \in \mathbb{Z}^{m_1}$ distributed statistically close to $\mathcal{D}_{\mathbb{Z}^{m_1},\sigma}$,
2. run $e_1 \xleftarrow{R} \mathsf{SamplePre}(A, T_A, y, \sigma)$ where $y = u - (M_1 \cdot e_2) \in \mathbb{Z}_q^n$, note that $\Lambda_q^y(A)$ is not empty since A is rank n,
3. output $e \leftarrow (e_1, e_2) \in \mathbb{Z}^{m+m_1}$

Clearly $(A \mid M_1) \cdot e = u \bmod q$ and hence $e \in \Lambda_q^u(F_1)$. Theorem 3.4 in [12] shows that the vector e is sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^u(F_1),\sigma}$.

Peikert's basis extension method [24] gives an alternate way to view this. Given the basis T_A of $\Lambda_q^\perp(A)$ Peikert shows how to build a basis T_{F_1} of $\Lambda_q^\perp(F_1)$ with the same Gram-Schmidt norm as T_A. Then calling $\mathsf{SamplePre}(F_1, T_{F_1}, u, \sigma)$ generates a vector e sampled from a distribution close to $\mathcal{D}_{\Lambda_q^u(F_1),\sigma}$. We summarize this in the following theorem.

Theorem 3. *Let $q > 2,\ m > 2n \log q$ and $\sigma > \|\widetilde{T_A}\| \cdot \omega(\sqrt{\log(m + m_1)})$. Then Algorithm $\mathsf{SampleLeft}(A, M_1, T_A, u, \sigma)$ taking inputs as in (1), outputs a vector $e \in \mathbb{Z}^{m+m_1}$ distributed statistically close to $\mathcal{D}_{\Lambda_q^u(F_1),\sigma}$ where $F_1 := (A \mid M_1)$.*

4.2 Algorithm SampleRight

Algorithm $\mathsf{SampleRight}(A, B, R, T_B, u, \sigma)$.

Inputs:

$$\begin{array}{l}\text{matrices } A, B \text{ in } \mathbb{Z}_q^{n\times m} \text{ where } B \text{ is rank } n,\\ \text{a uniform random matrix } R \in \{-1,1\}^{m\times m},\\ \text{a basis } T_B \text{ of } \Lambda_q^\perp(B) \text{ and a vector } u \in \mathbb{Z}_q^n,\\ \text{a parameter } \sigma > \|\widetilde{T_B}\| \cdot \sqrt{m} \cdot \omega(\log m).\end{array} \tag{2}$$

Output: Let $F_2 := (A \mid AR{+}B)$. The algorithm outputs a vector $e \in \mathbb{Z}^{2m}$ sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^u(F_2),\sigma}$. In particular, $e \in \Lambda_q^u(F_2)$.

The algorithm uses the basis growth method of Peikert [24, Sec. 3.3] and works in three steps:

1. First, it constructs a set T_{F_2} of $2m$ linearly independent vectors in $\Lambda_q^\perp(F_2)$ such that
$$\|\widetilde{T_{F_2}}\| < \|\widetilde{T_B}\| \cdot \sqrt{m} \cdot \omega(\sqrt{\log m}) < \sigma/\omega(\sqrt{\log m})$$
with overwhelming probability over the choice of R.
2. Next, if needed it uses Lemma 1 to convert T_{F_2} into a basis T'_{F_2} of $\Lambda_q^\perp(F_2)$ with the same Gram-Schmidt norm as T_{F_2}.
3. Finally, it invokes $\mathsf{SamplePre}(F_2, T'_{F_2}, u, \sigma)$ to generate a vector $e \in \Lambda_q^u(F_2)$. Since $\sigma > \|\widetilde{T_{F_2}}\|\, \omega(\sqrt{\log m})$ w.h.p, this e is distributed as $\mathcal{D}_{\Lambda_q^u(F_2),\sigma}$, as required.

Step 1 is the only step that needs explaining. Let $T_B = \{b_1, \ldots, b_m\} \in \mathbb{Z}^{m\times m}$ be the given basis of $\Lambda_q^\perp(B)$. We construct the $2m$ vectors in $\Lambda_q^\perp(F_2)$ as follows:

1. for $i = 1, \ldots, m$ set $t_i := (-Rb_i \mid b_i) \in \mathbb{Z}^{2m}$ and view it as a column vector; then clearly $F_2 \cdot t_i = B\, b_i = 0 \bmod q$ and therefore t_i is in $\Lambda_q^\perp(F_2)$.
2. for $i = 1, \ldots, m$ let w_i be the i-th column of the identity matrix I_m. Let u_i be an arbitrary vector in $\mathbb{Z}^m$ satisfying $Aw_i + Bu_i = 0 \bmod q$. This u_i exists since B is rank n. Set t_{i+m} to be
$$t_{i+m} := \begin{bmatrix} w_i - Ru_i \\ u_i \end{bmatrix} \in \mathbb{Z}^{2m}$$
Then $F_2 \cdot t_{i+m} = Aw_i + Bu_i = 0 \bmod q$ and hence, $t_{i+m} \in \Lambda_q^\perp(F_2)$.

We show that $T_{F_2} := \{t_1, \ldots, t_{2m}\}$ are linearly independent in $\mathbb{Z}^{2m}$. First, observe that the first m vectors are linearly independent and span the linear space V of vectors of the form $(-Rx \mid x)$ where $x \in \mathbb{Z}_q^m$. For all $i > m$, the vector t_i is the sum of the unit vector $(w_i \mid 0^m)$ plus a vector in V. It follows that T_{F_2} is a linearly independent set. This also means that for $i > m$ the i-th Gram-Schmidt vector of T_{F_2} cannot be longer than $(w_i \mid 0^m)$ and therefore has norm at most 1. Hence, to bound $\|\widetilde{T_{F_2}}\|$ it suffices to bound the Gram-Schmidt norm of the first m vectors $\{t_1, \ldots, t_m\}$.

Let $W \in \mathbb{Z}^{2m \times m}$ be the matrix $(-R^\top \mid I_m)^\top$. Then $t_i = Wb_i$ for $i = 1, \ldots, m$. Since R is uniform in $\{-1, 1\}^{m \times m}$ we know by Lemma 5 that for all vectors $x \in \mathbb{R}^m$ we have w.h.p

$$\|Wx\| \leq \|Rx\| + \|x\| \leq \|x\|\sqrt{m} \cdot \omega(\sqrt{\log m}) + \|x\| \leq \|x\|\sqrt{m} \cdot \omega(\sqrt{\log m})$$

Now, since $t_i = Wb_i$ for $i = 1, \ldots, m$, applying Lemma 2 to the matrix W gives a bound on the Gram-Schmidt norm of $\{t_1, \ldots, t_m\}$ (and hence also on $\|\widetilde{T_{F_2}}\|$):

$$\begin{aligned}\|\widetilde{T_{F_2}}\| &\leq \max_{1 \leq i \leq m} \|W\tilde{b}_i\| \leq \max_{1 \leq i \leq m} \|\tilde{b}_i\| \cdot \sqrt{m} \cdot \omega(\sqrt{\log m}) \\ &\leq \|\widetilde{T_B}\| \cdot \sqrt{m} \cdot \omega(\sqrt{\log m})\end{aligned}$$

Thus, we built $2m$ linearly independent vectors in $\Lambda_q^\perp(F_2)$ that w.h.p. have a short Gram-Schmidt norm as required for Step 1. This completes the description of algorithm SampleRight. We summarize this in the following theorem.

Theorem 4. *Let $q > 2, m > n$ and $\sigma > \|\widetilde{T_B}\| \cdot \sqrt{m} \cdot \omega(\log m)$. Then Algorithm, SampleRight$(A, B, R, T_B, u, \sigma)$ taking inputs as in (2), with R uniform in $\{1, -1\}^{m \times m}$, outputs a vector $e \in \mathbb{Z}^{2m}$ distributed statistically close to $\mathcal{D}_{\Lambda_q^u(F_2), \sigma}$ where $F_2 := (A \mid AR + B)$.*

5 Encoding Identities as Matrices

Our construction uses an encoding function $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$ to map identities in $\mathbb{Z}_q^n$ to matrices in $\mathbb{Z}_q^{n \times n}$. Our proof of security requires that the map H satisfy a strong notion of injectivity, namely that, for any two distinct inputs id_1 and id_2, the difference between the outputs $H(\mathsf{id}_1)$ and $H(\mathsf{id}_2)$ is never singular, i.e., $\det(H(\mathsf{id}_1) - H(\mathsf{id}_2)) \neq 0$.

Definition 5. *Let q be a prime and n a positive integer. We say that a function $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$ is an* **encoding with full-rank differences** *(FRD) if:*

1. *for all distinct $u, v \in \mathbb{Z}_q^n$, the matrix $H(u) - H(v) \in \mathbb{Z}_q^{n \times n}$ is full rank; and*
2. *H is computable in polynomial time (in $n \log q$).*

Clearly the function H must be injective since otherwise, if $u \neq v$ satisfies $H(u) = H(v)$, then $H(u) - H(v)$ is not full-rank and hence H cannot be FRD.

The function H in Definition 5 has domain of size q^n which is the largest possible for a function satisfying condition 1 of Definition 5. Indeed, if H had domain larger than q^n then its image is also larger than q^n. But then, by pigeonhole, there are two distinct inputs u, v such that the matrices $H(u)$ and $H(v)$ have the same first row and therefore $H(u) - H(v)$ is not full rank. It follows that our definition of FRD, which has domain of size of q^n, is the largest possible.

An Explicit FRD Construction. We construct an injective FRD encoding for the exponential-size domain $\mathsf{id} \in \mathbb{Z}_q^n$. A similar construction is described in [14]. Our strategy is to construct an additive subgroup $\mathbb{G}$ of $\mathbb{Z}_q^{n\times n}$ of size q^n such that all non-zero matrices in $\mathbb{G}$ are full-rank. Since for all distinct $A, B \in \mathbb{G}$ the difference $A - B$ is also in $\mathbb{G}$, it follows that $A - B$ is full-rank.

While our primary interest is the finite field $\mathbb{Z}_q$ we describe the construction for an arbitrary field $\mathbb{F}$. For a polynomial $g \in \mathbb{F}[X]$ of degree less than n define $\text{coeffs}(g) \in \mathbb{F}^n$ to be the n-vector of coefficients of g (written as a row-vector). If g is of degree less than $n-1$ we pad the coefficients vector with zeroes on the right to make it an n-vector. For example, for $n = 6$ we have $\text{coeffs}(x^3 + 2x + 3) = (3, 2, 0, 1, 0, 0) \in \mathbb{F}^6$. Let f be some polynomial of degree n in $\mathbb{F}[X]$ that is irreducible. Recall that for a polynomial $g \in \mathbb{F}[X]$ the polynomial $g \bmod f$ has degree less than n and therefore $\text{coeffs}(g \bmod f)$ is a vector in $\mathbb{F}^n$.

Now, for an input $u = (u_0, u_1, \ldots, u_{n-1}) \in \mathbb{F}^n$ define the polynomial $g_u(X) = \sum_{i=0}^{n-1} u_i x^i \in \mathbb{F}[X]$. Define $H(u)$ as

$$H(u) := \begin{pmatrix} \text{coeffs}(\ g_u\ ) \\ \text{coeffs}(\ X \cdot g_u \bmod f\ ) \\ \text{coeffs}(\ X^2 \cdot g_u \bmod f\ ) \\ \vdots \\ \text{coeffs}(\ X^{n-1} \cdot g_u \bmod f\ ) \end{pmatrix} \in \mathbb{F}^{n\times n} \tag{3}$$

This completes the construction. Since for all primes q and integers $n > 1$ there are (many) irreducible polynomials in $\mathbb{Z}_q[X]$ of degree n, the construction can accommodate any pair of q and n.

The following theorem proves that the function H in (3) is an FRD. The proof, given in [14], is based on the observation that the matrix $H(u)^\top$ corresponds to multiplication by a constant in the number field $K = \mathbb{F}[X]/(f)$ and is therefore invertible when the matrix is non-zero. We note that similar matrix encodings of ring multiplication were previously used in [26,21].

Theorem 5. *Let $\mathbb{F}$ be a field and f a polynomial in $\mathbb{F}[X]$. If f is irreducible in $\mathbb{F}[X]$ then the function H defined in (3) is an encoding with full-rank differences (or FRD encoding).*

An example. Let $n = 4$ and $f(X) = x^4 + x - 1$. The function H works as follows:

$$H\big(u = (u_0, u_1, u_2, u_3)\big) \;:=\; \begin{pmatrix} u_0 & u_1 & u_2 & u_3 \\ u_3 & u_0 - u_3 & u_1 & u_2 \\ u_2 & u_3 - u_2 & u_0 - u_3 & u_1 \\ u_1 & u_2 - u_1 & u_3 - u_2 & u_0 - u_3 \end{pmatrix}$$

Theorem 5 shows that the map H is FRD for all primes q where $x^4 + x - 1$ is irreducible in $\mathbb{Z}_q[X]$ (e.g. $q = 19, 31, 43, 47$).

6 The Main Construction: An Efficient IBE

The system uses parameters q, n, m, σ, α specified in Section 6.3. Throughout the section, the function H refers to the FRD map $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$ defined in Section 5. We assume identities are elements in $\mathbb{Z}_q^n$. The set of identities can be expanded to $\{0,1\}^*$ by hashing identities into $\mathbb{Z}_q^n$ using a collision resistant hash.

6.1 Intuition

The public parameters in our system consist of three random $n \times m$ matrices over $\mathbb{Z}_q$ denoted by A_0, A_1 and B as well as a vector $u \in \mathbb{Z}_q^n$. The master secret is a trapdoor T_{A_0} (i.e. a basis with a low Gram-Schmidt norm) for the lattice $\Lambda_q^{\perp}(A_0)$.

The secret key for an identity id is a short vector $e \in \mathbb{Z}^{2m}$ satisfying $F_{\mathsf{id}} \cdot e = u$ in $\mathbb{Z}_q$ where

$$F_{\mathsf{id}} := (A_0 \mid A_1 + H(\mathsf{id})\, B) \quad \in \mathbb{Z}_q^{n \times 2m}$$

The vector e is generated using algorithm $\mathsf{SampleLeft}$ (Theorem 3) and the trapdoor T_{A_0}.

In a selective IBE security game the attacker announces an identity id^* that it plans to attack. We need a simulator that can respond to private key queries for $\mathsf{id} \neq \mathsf{id}^*$, but knows nothing about the private key for id^*. We do so by choosing the public parameters A_0 and B at random as before, but choosing A_1 as

$$A_1 := A_0\, R - H(\mathsf{id}^*)\, B$$

where R is a random matrix in $\{1, -1\}^{m \times m}$. We show that $A_0\, R$ is uniform and independent in $\mathbb{Z}_q^{n \times m}$ so that A_1 is distributed as required. We provide the simulator with a trapdoor T_B for $\Lambda_q^{\perp}(B)$, but no trapdoor for $\Lambda_q^{\perp}(A_0)$.

Now, to respond to a private key query for an identity id, the simulator must produce a short vector e satisfying $F_{\mathsf{id}} \cdot e = u$ in $\mathbb{Z}_q$ where

$$F_{\mathsf{id}} := \big(A_0 \mid A_0 \cdot R + B'\big) \in \mathbb{Z}_q^{n \times 2m} \quad \text{and} \quad B' := \big(H(\mathsf{id}) - H(\mathsf{id}^*)\big) \cdot B\ .$$

When $\mathsf{id} \neq \mathsf{id}^*$ we know that $H(\mathsf{id}) - H(\mathsf{id}^*)$ is full rank by construction and therefore T_B is also a trapdoor for the lattice $\Lambda_q^{\perp}(B')$. The simulator can now generate e using algorithm $\mathsf{SampleRight}$ and the basis T_B.

When $\mathsf{id} = \mathsf{id}^*$ the matrix F_{id} no longer depends on B and the simulator's trapdoor disappears. Consequently, the simulator can generate private keys for all identities other than id^*. As we will see, for id^* the simulator can produce a challenge ciphertext that helps it solve the given LWE challenge.

6.2 The Basic IBE Construction

Setup(λ)**:** On input a security parameter λ, set the parameters q, n, m, σ, α as specified in Section 6.3 below. Next do:

1. Use algorithm $\mathsf{TrapGen}(q, n)$ to select a uniformly random $n \times m$-matrix $A_0 \in \mathbb{Z}_q^{n\times m}$ with a basis T_{A_0} for $\Lambda_q^\perp(A_0)$ such that $\|\widetilde{T_{A_0}}\| \le O(\sqrt{n \log q})$
2. Select two uniformly random $n \times m$ matrices A_1 and B in $\mathbb{Z}_q^{n\times m}$.
3. Select a uniformly random n-vector $u \xleftarrow{R} \mathbb{Z}_q^n$.
4. Output the public parameters and master key,

$$\mathsf{PP} = \Big(\ A_0, A_1, B, u\ \Big) \quad ; \quad \mathsf{MK} = \Big(\ T_{A_0}\ \Big) \ \in \mathbb{Z}^{m\times m}$$

Extract($\mathsf{PP}, \mathsf{MK}, \mathsf{id}$)**:** On input public parameters PP, a master key MK, and an identity $\mathsf{id} \in \mathbb{Z}_q^n$, do:

1. Sample $e \in \mathbb{Z}^{2m}$ as $e \leftarrow \mathsf{SampleLeft}(A_0,\ \ A_1 + H(\mathsf{id})\, B,\ \ T_{A_0},\ \ u,\ \ \sigma)$ where H is an FRD map as defined in Section 5.
   Note that A_0 is rank n w.h.p as explained in Section 6.3.
2. Output $\mathsf{SK}_{\mathsf{id}} := e\ \in \mathbb{Z}^{2m}$

Let $F_{\mathsf{id}} := \big(A_0 \mid A_1 + H(\mathsf{id})\, B\big)$, then $F_{\mathsf{id}} \cdot e = u$ in $\mathbb{Z}_q$ and e is distributed as $D_{\Lambda_q^u(F_{\mathsf{id}}),\sigma}$ by Theorem 3.

Encrypt($\mathsf{PP}, \mathsf{id}, b$)**:** On input public parameters PP, an identity id, and a message $b \in \{0, 1\}$, do:

1. Set $F_{\mathsf{id}} \leftarrow \big(A_0 \mid A_1 + H(\mathsf{id}) \cdot B\big) \in \mathbb{Z}_q^{n\times 2m}$
2. Choose a uniformly random $s \xleftarrow{R} \mathbb{Z}_q^n$
3. Choose a uniformly random $m \times m$ matrix $R \xleftarrow{R} \{-1, 1\}^{m\times m}$
4. Choose noise vectors $x \xleftarrow{\bar{\Psi}_\alpha} \mathbb{Z}_q$ and $y \xleftarrow{\bar{\Psi}_\alpha^m} \mathbb{Z}_q^m$, and set $z \leftarrow R^\top y \in \mathbb{Z}_q^m$ (the distribution $\bar{\Psi}_\alpha$ is as in Definition 4),
5. Set $c_0 \leftarrow u^\top s + x + b\lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q$ and $c_1 \leftarrow F_{\mathsf{id}}^\top s + \begin{bmatrix} y \\ z \end{bmatrix} \in \mathbb{Z}_q^{2m}$
6. Output the ciphertext $\mathsf{CT} := (c_0, c_1)\ \in \mathbb{Z}_q \times \mathbb{Z}_q^{2m}$.

Decrypt($\mathsf{PP}, \mathsf{SK}_{\mathsf{id}}, \mathsf{CT}$)**:** On input public parameters PP, a private key $\mathsf{SK}_{\mathsf{id}} := e_{\mathsf{id}}$, and a ciphertext $\mathsf{CT} = (c_0, c_1)$, do:

1. Compute $w \leftarrow c_0 - e_{\mathsf{id}}^\top c_1 \in \mathbb{Z}_q$.
2. Compare w and $\lfloor \frac{q}{2} \rfloor$ treating them as integers in $\mathbb{Z}$. If they are close, i.e., if $\Big| w - \lfloor \frac{q}{2} \rfloor \Big| < \lfloor \frac{q}{4} \rfloor$ in $\mathbb{Z}$, output 1, otherwise output 0.

The matrix R. The matrix R used in encryption plays an important role in the security proof. Note that the matrix is only used as a tool to sample the noise vector (y, z) from a specific distribution needed in the simulation.

6.3 Parameters and Correctness

When the cryptosystem is operated as specified, we have,

$$w \ = c_0 - e_{\mathsf{id}}^\top c_1 = b \lfloor \frac{q}{2} \rfloor + \underbrace{x - e_{\mathsf{id}}^\top \begin{bmatrix} y \\ z \end{bmatrix}}_{\text{error term}}$$

In the full paper we show that the error term is bounded by $\tilde{O}(q\alpha\sigma m)$ w.h.p. This follows from the same analysis as in [16, Lemma 8.2] plus Lemma 5 to bound $\|z\|$.

To ensure that the error term is less than $q/5$, that σ is sufficiently large for SampleLeft and SampleRight, that TrapGen can operate (i.e. $m > 6n \log q$), and that Regev's reduction applies (i.e. $q > 2\sqrt{n}/\alpha$), we set the parameters (q, m, σ, α) as follows, taking n to be the security parameter:

$$\begin{aligned} m &= 6\, n^{1+\delta} \quad , \quad q = m^2 \sqrt{n} \cdot \omega(\log n) \\ \sigma &= m \cdot \omega(\log n) \quad , \quad \alpha = [m^2 \cdot \omega(\log n)]^{-1} \end{aligned} \tag{4}$$

and round up m to the nearest larger integer and q to the nearest larger prime. Here we assume that δ is such that $n^\delta > \lceil \log q \rceil = O(\log n)$.

Since the matrices A_0, B are random in $\mathbb{Z}_q^{n \times m}$ and $m > n \log q$, with overwhelming probability both matrices will have rank n. Hence, calling SampleLeft in algorithm Extract succeeds w.h.p.

6.4 Security Reduction

We show that the basic IBE construction is indistinguishable from random under a selective identity attack as in Definition 1. Recall that indistinguishable from random means that the challenge ciphertext is indistinguishable from a random element in the ciphertext space. This property implies both semantic security and recipient anonymity.

Theorem 6. *The basic IBE system with parameters $(q, n, m, \sigma, \alpha)$ as in (4) is INDr–sID-CPA secure provided that the $(\mathbb{Z}_q, n, \bar{\Psi}_\alpha)$-LWE assumption holds.*

Proof. The proof proceeds in a sequence of games where the first game is identical to the INDr–sID-CPA game from Definition 1. In the last game in the sequence the adversary has advantage zero. We show that a PPT adversary cannot distinguish between the games which will prove that the adversary has negligible advantage in winning the original INDr–sID-CPA game. The LWE problem is used in proving that Games 2 and 3 are indistinguishable.

Game 0. This is the original INDr–sID-CPA game from Definition 1 between an attacker $\mathcal{A}$ against our scheme and an INDr–sID-CPA challenger.

Game 1. Recall that in Game 0 the challenger generates the public parameters PP by choosing three random matrices A_0, A_1, B in $\mathbb{Z}_q^{n \times m}$ such that a trapdoor

T_{A_0} is known for $\Lambda_q^{\perp}(A_0)$. At the challenge phase the challenger generates a challenge ciphertext CT^*. We let $R^* \in \{-1,1\}^{m\times m}$ denote the random matrix generated for the creation of CT^* (in step 3 of $\mathsf{Encrypt}$).

In Game 1 we slightly change the way that the challenger generates A_1 in the public parameters. Let id^* be the identity that $\mathcal{A}$ intends to attack. The Game 1 challenger chooses R^* at the setup phase and constructs A_1 as

$$A_1 \leftarrow A_0\, R^* - H(\mathsf{id}^*)\, B \tag{5}$$

The remainder of the game is unchanged.

We show that Game 0 is statistically indistinguishable from Game 1 by Lemma 4. Observe that in Game 1 the matrix R^* is used only in the construction of A_1 and in the construction of the challenge ciphertext where $z \leftarrow (R^*)^{\top} y$. By Lemma 4 the distribution $(A_0,\ A_0\, R^*,\ z)$ is statistically close to the distribution $(A_0,\ A_1',\ z)$ where A_1' is a uniform $\mathbb{Z}_q^{n\times m}$ matrix. It follows that in the adversary's view, the matrix $A_0\, R^*$ is statistically close to uniform and therefore A_1 as defined in (5) is close to uniform. Hence, A_1 in Games 0 and 1 are indistinguishable.

Game 2. We now change how A_0 and B in PP are chosen. In Game 2 we generate A_0 as a random matrix in $\mathbb{Z}_q^{n\times m}$, but generate B using algorithm $\mathsf{TrapGen}$ so that B is a random matrix in $\mathbb{Z}_q^{n\times m}$, but the challenger has a trapdoor T_B for $\Lambda_q^{\perp}(B)$. The choice of A_1 remains as in Game 1, i.e. $A_1 = A_0 \cdot R^* - H(\mathsf{id}^*) \cdot B$.

The challenger responds to private key queries using the trapdoor T_B. To respond to a private key query for $\mathsf{id} \neq \mathsf{id}^*$ the challenger needs a short $e \in \Lambda_q^{u}(F_{\mathsf{id}})$ where

$$F_{\mathsf{id}} := (A_0 \mid A_1 + H(\mathsf{id}) \cdot B) = \big(A_0 \mid A_0 R^* + \big(H(\mathsf{id}) - H(\mathsf{id}^*)\big)B\big)\ .$$

By construction, $[H(\mathsf{id}) - H(\mathsf{id}^*)]$ is non-singular and therefore T_B is also a trapdoor for $\Lambda_q^{\perp}(B')$ where $B' := \big(H(\mathsf{id}) - H(\mathsf{id}^*)\big)B$. Moreover, since B is rank n w.h.p, so is B'. The challenger can now respond to the private key query by running

$$e \leftarrow \mathsf{SampleRight}\big(A_0,\ \ \big(H(\mathsf{id}) - H(\mathsf{id}^*)\big)B,\ \ R^*,\ T_B,\ \ u,\ \ \sigma\big) \in \mathbb{Z}_q^{2m}$$

and sending $\mathsf{SK}_{\mathsf{id}} := e$ to $\mathcal{A}$. Since the σ used in the system is sufficiently large, this e is distributed close to $D_{\Lambda_q^{u}(F_{\mathsf{id}}),\sigma}$, as in Game 1 by Theorem 4.

Game 2 is otherwise the same as Game 1. Since A_0, B and responses to private key queries are statistically close to those in Game 1, the adversary's advantage in Game 2 is at most negligibly different from its advantage in Game 1.

Game 3. Game 3 is identical to Game 2 except that the challenge ciphertext (c_0^*, c_1^*) is *always* chosen as a random independent element in $\mathbb{Z}_q \times \mathbb{Z}_q^{2m}$. Since the challenge ciphertext is always a fresh random element in the ciphertext space, $\mathcal{A}$'s advantage in this game is zero.

It remains to show that Game 2 and Game 3 are computationally indistinguishable for a PPT adversary, which we do by giving a reduction from the LWE problem.

Reduction from LWE. Suppose $\mathcal{A}$ has non-negligible advantage in distinguishing Games 2 and 3. We use $\mathcal{A}$ to construct an LWE algorithm $\mathcal{B}$.

Recall from Definition 3 that an LWE problem instance is provided as a sampling oracle $\mathcal{O}$ which can be either truly random $\mathcal{O}_\$$ or a noisy pseudorandom $\mathcal{O}_s$ for some secret $s \in \mathbb{Z}_q^n$. 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, for each $i = 0, \ldots, m$, a fresh pair $(u_i, v_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.

Targeting. $\mathcal{A}$ announces to $\mathcal{B}$ the identity id^* that it intends to attack.

Setup. $\mathcal{B}$ constructs the system's public parameters PP as follows:

1. Assemble the random matrix $A_0 \in \mathbb{Z}_q^{n \times m}$ from m of the previously given LWE samples by letting the i-th column of A_0 be the n-vector u_i for all $i = 1, \ldots, m$.
2. Assign the zeroth LWE sample (so far unused) to become the public random n-vector $u_0 \in \mathbb{Z}_q^n$.
3. The remainder of the public parameters, namely A_1 and B, are constructed as in Game 2 using id^* and R^*.

Queries. $\mathcal{B}$ answers each private-key extraction query as in Game 2.

Challenge. $\mathcal{B}$ prepares, when prompted by $\mathcal{A}$ with a message bit $b^* \in \{0, 1\}$, a challenge ciphertext for the target identity id^*, as follows:

1. Let $v_0, \ldots, v_m$ be entries from the LWE instance. Set $v^* = \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} \in \mathbb{Z}_q^m$.
2. Blind the message bit by letting $c_0^* = v_0 + b^* \lfloor \frac{q}{2} \rceil \in \mathbb{Z}_q$.
3. Set $c_1^* = \begin{bmatrix} v^* \\ (R^*)^\top v^* \end{bmatrix} \in \mathbb{Z}_q^{2m}$.
4. Choose a random bit $r \xleftarrow{R} \{0, 1\}$. If $r = 0$ send $\mathsf{CT}^* = (c_0^*, c_1^*)$ to the adversary. If $r = 1$ choose a random $(c_0, c_1) \in \mathbb{Z}_q \times \mathbb{Z}_q^{2m}$ and send (c_0, c_1) to the adversary.

We argue that when the LWE oracle is pseudorandom (i.e. $\mathcal{O} = \mathcal{O}_s$) then CT^* is distributed exactly as in Game 2. First, observe that $F_{\mathsf{id}^*} = (A_0 \mid A_0 R^*)$. Second, by definition of $\mathcal{O}_s$ we know that $v^* = A_0^\top s + y$ for some random noise vector $y \in \mathbb{Z}_q^m$ distributed as $\bar{\Psi}_\alpha^m$. Therefore, c_1^* defined in step (3) above satisfies

$$c_1^* = \begin{bmatrix} A_0^\top s + y \\ (R^*)^\top A_0^\top s + (R^*)^\top y \end{bmatrix} = \begin{bmatrix} A_0^\top s + y \\ (A_0 R^*)^\top s + (R^*)^\top y \end{bmatrix} = (F_{\mathsf{id}^*})^\top s + \begin{bmatrix} y \\ (R^*)^\top y \end{bmatrix}$$

and the quantity on the right is precisely the c_1 part of a valid challenge ciphertext in Game 2. Also note that $v_0 = u_0^\top s + x$, just as the c_0 part of the challenge ciphertext in Game 2.

When $\mathcal{O} = \mathcal{O}_\$$ we have that v_0 is uniform in $\mathbb{Z}_q$ and v^* is uniform in $\mathbb{Z}_q^m$. Therefore c_1^* as defined in step (3) above is uniform and independent in $\mathbb{Z}_q^{2m}$ by the standard left over hash lemma (e.g. Theorem 8.38 of [29]) where the hash function is defined by the matrix $(A_0^\top | v^*)$. Consequently, the challenge ciphertext is always uniform in $\mathbb{Z}_q \times \mathbb{Z}_q^{2m}$, as in Game 3.

Guess. After being allowed to make additional queries, $\mathcal{A}$ guesses if it is interacting with a Game 2 or Game 3 challenger. Our simulator outputs $\mathcal{A}$'s guess as the answer to the LWE challenge it is trying to solve.

We already argued that when $\mathcal{O} = \mathcal{O}_s$ the adversary's view is as in Game 2. When $\mathcal{O} = \mathcal{O}_\$$ the adversary's view is as in Game 3. Hence, $\mathcal{B}$'s advantage in solving LWE is the same as $\mathcal{A}$'s advantage in distinguishing Games 2 and 3, as required. This completes the description of algorithm $\mathcal{B}$ and completes the proof.

6.5 Multi-bit Encryption

We briefly note that, as in [16], it is possible to reuse the same ephemeral encryption randomness s to encrypt multiple message bits. An N-bit message can thus be encrypted as N components c_0 plus a single component c_1, where the same ephemeral $s \in \mathbb{Z}_q^n$ is used throughout. The total ciphertext size with this technique is 1 element of $\mathbb{Z}_q$ for each bit of the message, plus a constant $2\,m$ elements of $\mathbb{Z}_q$ regardless of the message length. The ciphertext size is thus $(N + 2\,m)$ elements of $\mathbb{Z}_q$.

7 Extensions: HIBE and Adaptively-Secure IBE

In the full version of the paper [1] we show two extensions of the basic IBE construction from Section 6.2.

Adaptively secure IBE. Recall that Waters [31] showed how to convert the selectively-secure IBE in [6] to an adaptively secure IBE. We show that a similar technique, also used in Boyen [10], can convert our basic IBE construction to an adaptively secure IBE. We treat an identity id as a sequence of ℓ bits $\mathsf{id} = (b_1, \dots, b_\ell)$ in $\{1, -1\}^\ell$. Then during encryption we use the matrix

$$F_{\mathsf{id}} := \big(\ A_0 \mid C + \sum_{i=1}^{\ell} b_i\, A_i\ \big) \quad \in \mathbb{Z}_q^{n \times 2m}$$

where $A_0, A_1, \dots, A_\ell, C$ are matrices in the public parameters. The result is an adaptively secure lattice IBE, simpler and with shorter ciphertexts than the recent construction of Cash et al. [12].

Hierarchical IBE. We show how the basis delegation techniques from [12,24] can convert the basic IBE construction to an HIBE. For an identity $\mathsf{id} = (\mathsf{id}_1, \dots, \mathsf{id}_\ell)$ at depth ℓ the matrix F_{id} used in encryption is defined as follows:

$$F_{\mathsf{id}} := \big(\ A_0 \mid A_1 + H(\mathsf{id}_1)B \mid \ \dots\ \mid A_\ell + H(\mathsf{id}_\ell)B\ \big) \quad \in \mathbb{Z}_q^{n \times (\ell+1)m}$$

where $A_0, A_1, \dots, A_\ell, B$ are matrices in the public parameters. We note that a recent HIBE construction in [2] gives a lattice-based HIBE where the lattice dimension does not grow with the identity's depth in the hierarchy.

8 Conclusion and Open Problems

We constructed an efficient identity-based encryption scheme and proven its security in the standard model from the LWE assumption (which is itself implied by worst-case lattice assumptions). In the full paper [1] we extend the basic selective-ID secure scheme to provide full adaptive-ID security, and to support a delegation mechanism to make it hierarchical.

It would be interesting to improve these constructions by adapting them to ideal lattices [30]. Another open problem is to construct an adaptively secure lattice-based IBE in the standard model where all the data is short (including the public parameters).

Acknowledgments

We are grateful to Chris Peikert for suggesting that we use the basis extension method from [24] to simplify the analysis of algorithm SampleLeft. This suggestion also let us to remove the matrix R from the master secret. We also thank Ron Rivest for pointing out that indistinguishability from random can help IBE systems resist subpoenas.

References

1. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model (2010); Full version of this paper. Available on the authors' web page
2. Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE (2010) (manuscript)
3. Agrawal, S., Boyen, X.: Identity-based encryption from lattices in the standard model (2009) (manuscript), http://www.cs.stanford.edu/~xb/ab09/
4. 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)
5. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. In: Proc. of STACS 2009, pp. 75–86 (2009)
6. 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)
7. 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)
8. 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)
9. Boneh, D., Gentry, C., Hamburg, M.: Space-efficient identity based encryption without pairings. In: Proc. of FOCS 2007, pp. 647–657 (2007)
10. Boyen, X.: Lattices niçoises and vanishing trapdoors: A framework for fully secure short signatures and more. In: PKC 2010. LNCS. Springer, Heidelberg (to appear, 2010)
11. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. J. Cryptol. 20(3), 265–294 (2007)
12. Cash, D., Hofheinz, D., Kiltz, E.: How to delegate a lattice basis. Cryptology ePrint Archive, Report 2009/351 (2009), http://eprint.iacr.org/

13. Cocks, C.: An identity based encryption scheme based on quadratic residues. In: Proceedings of the 8th IMA Conference, pp. 26–28 (2001)
14. Cramer, R., Damgard, I.: On the amortized complexity of zero-knowledge protocols. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 177–191. Springer, Heidelberg (2009)
15. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM Journal on Computing 38(1), 97–139 (2008)
16. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proc. of STOC 2008, pp. 197–206 (2008)
17. Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
18. Gentry, C., Silverberg, A.: Hierarchical id-based cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002)
19. Hoeffding, W.: Probability inequalities for sums of bounded random variables. Journal of the American Statistical Association 58(301), 13–30 (1963)
20. Horwitz, J., Lynn, B.: Toward hierarchical identity-based encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 466–481. Springer, Heidelberg (2002)
21. Lyubashevsky, V., Micciancio, D.: Generalized compact knapsacks are collision resistant. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 144–155. Springer, Heidelberg (2006)
22. Micciancio, D., Goldwasser, S.: Complexity of Lattice Problems: a cryptographic perspective, vol. 671. Kluwer Academic Publishers, Dordrecht (2002)
23. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. In: Proc. of FOCS 2004, pp. 372–381 (2004)
24. Peikert, C.: Bonsai trees (or, arboriculture in lattice-based cryptography). Cryptology ePrint Archive, Report 2009/359 (2009), http://eprint.iacr.org/
25. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Proc. of STOC 2009, pp. 333–342. ACM, New York (2009)
26. 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)
27. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proc. of STOC 2005, pp. 84–93 (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. Shoup, V.: A Computational Introduction to Number Theory and Algebra, 2nd edn. Cambridge University Press, Cambridge (2008)
30. Stehle, 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)
31. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
32. Waters, B.: Dual key encryption: Realizing fully secure IBE and HIBE under simple assumption. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)

Multi-property-preserving Domain Extension Using Polynomial-Based Modes of Operation

Jooyoung Lee[1] and John Steinberger[2,⋆]

[1] The Attached Institute of Electronics and Telecommunications Research Institute,
Daejeon, Korea
jlee05@ensec.re.kr

[2] Institute of Theoretical Computer Science, Tsinghua University, Beijing, China
jpsteinb@gmail.com

Abstract. In this paper, we propose a new double-piped mode of operation for multi-property-preserving domain extension of MACs (message authentication codes), PRFs (pseudorandom functions) and PROs (pseudorandom oracles). Our mode of operation performs twice as fast as the original double-piped mode of operation of Lucks [15] while providing comparable security. Our construction, which uses a class of polynomial-based compression functions proposed by Stam [22,23], makes a single call to a $3n$-bit to n-bit primitive at each iteration and uses a finalization function f_2 at the last iteration, producing an n-bit hash function $H[f_1, f_2]$ satisfying the following properties.

1. $H[f_1, f_2]$ is unforgeable up to $O(2^n/n)$ query complexity as long as f_1 and f_2 are unforgeable.
2. $H[f_1, f_2]$ is pseudorandom up to $O(2^n/n)$ query complexity as long as f_1 is unforgeable and f_2 is pseudorandom.
3. $H[f_1, f_2]$ is indifferentiable from a random oracle up to $O(2^{2n/3})$ query complexity as long as f_1 and f_2 are public random functions.

To our knowledge, our result constitutes the first time $O(2^n/n)$ unforgeability has been achieved using only an unforgeable primitive of n-bit output length. (Yasuda showed unforgeability of $O(2^{5n/6})$ for Lucks' construction assuming an unforgeable primitive, but the analysis is suboptimal; in the appendix, we show how Yasuda's bound can be improved to $O(2^n)$.)

In related work, we strengthen Stam's collision resistance analysis of polynomial-based compression functions (showing that unforgeability of the primitive suffices) and discuss how to implement our mode by replacing f_1 with a $2n$-bit key blockcipher in Davies-Meyer mode or by replacing f_1 with the cascade of two $2n$-bit to n-bit compression functions.

1 Introduction

The Merkle-Damgård transform has been the most popular method to build a cryptographic hash function from a fixed-size compression function. A major

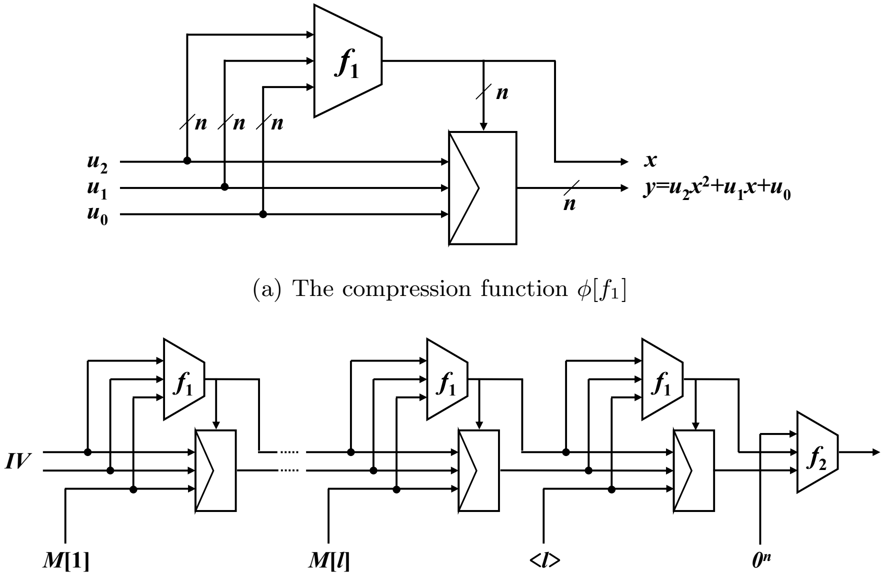
[⋆] Supported by the National Natural Science Foundation of China Grant 60553001 and by the National Basic Research Program of China Grant 2007CB807900, 2007CB807901.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 573–596, 2010.

advantage of this construction is that it preserves collision resistance with an appropriate padding algorithm, allowing one to focus on the construction of secure compression functions. However, Joux showed that if computing collisions becomes somehow feasible for the underlying compression function, then the hash function may fail worse than expected: for a hash function based on a compression function of n-bit output, one can find a t-multicollision only with $O(2^{n/2} \log t)$ complexity, which is much smaller than $O(2^{(t-1)n/t})$ required for an ideal random function. This observation led to several generic attacks such as long-message second preimage attacks [13] and herding attacks [12]. Lucks observed that these weaknesses can be mitigated by increasing the size of the internal state and claimed that the internal state size should be seen as a security parameter of its own right [15]. Since a secure compression function of a larger output size might be harder to construct than the hash function itself, Lucks proposed to use a "narrow" compression function in a double-piped mode. In a subsequent paper [24], Yasuda rigorously analyzed the security of the double-piped mode of operation as a multi-property-preserving domain extension. Specifically, he showed that Lucks' double-piped mode of operation preserves unforgeability up to $O(2^{5n/6})$ query complexity, and indistinguishability and indifferentiability both up to $O(2^n)$ query complexity. Moreover it was later noticed by several researchers that Yasuda's unforgeability bound could be increased to $O(2^n)$ with a slightly modified proof. (See appendix B.)

As such Lucks' construction turned out to provide nearly optimal security. However, the fact that Lucks' compression function uses two applications of a (fairly strong) primitive remains a drawback. Stam [22,23] recently proposed a class of wide-pipe compression functions making a *single* call to an equal primitive (we call these *polynomial-based* compression functions). In this paper we analyze the security properties of double-piped modes using Stam's polynomial-based compression functions, focusing on MAC-preservation, PRF-preservation and PRO-preservation. Except for PRO-preservation (where we only achieve $O(2^{2n/3})$ security), our bounds are comparable to those found by Yasuda for Lucks' original construction (and even better for unforgeability, given the suboptimality of Yasuda's bound in that case, though the "corrected" unforgeability bound exceeds ours by a factor of n) even though our construction has twice the rate.

Besides performance, a second concern that arises for Lucks' double-pipe construction is the rather strong primitive it assumes: a $3n$-bit to n-bit function (note that careful consideration is typically already given for the construction of $2n$-bit to n-bit compression functions from smaller or more available primitives). Here we also tackle this problem and show our double-piped polynomial-based mode can be implemented with a blockcipher of $2n$-bit key in Davies-Meyer mode, in either the ideal-cipher model or the weaker "unpredictable cipher" model (see Section 5) without significant loss of security. We also prove MAC-preservation and PRF-preservation for a compression function obtained by replacing the $3n$-bit to n-bit primitive with the cascade of two $2n$-bit to n-bit primitives. This latter result potentially opens the door to implementing the



(a) The compression function $\phi[f_1]$

(b) The hash function $H[f_1, f_2]$

Fig. 1. The polynomial-based mode of operation for $c = n$

compression function with two calls to an n-bit key blockcipher in Davies-Meyer mode (which would be the first time, to our knowledge, that a $3n$-bit to $2n$-bit compression function using two calls to an n-bit key blockcipher is proved secure nearly up to the birthday bound).

Construction and Results. To keep our construction comparably general to Lucks' [15] and Yasuda's [24], we discuss a hash function obtained by iterating a $(2n + c)$-bit to $2n$-bit compression function $\phi[f_1]$ where the primitive f_1 used by the compression function is a $2n + c$-bit to n-bit compression function (the "expected" setting of the parameters is $c = n$).

The compression function $\phi[f_1]$ is illustrated in Fig. 1(a) for the case $c = n$ of a $3n$-bit to $2n$-bit compression function. Let $u \in \{0, 1\}^{2n+c}$ and let $u_d|| \ldots ||u_0$ be the segmentation of u into n-bit blocks u_0, ..., u_{d-1} and a block u_d of no more than n bits (so $d = \lceil \frac{2n+c}{n} \rceil - 1$). Then $\phi[f_1](u)$ is defined by

$$\phi[f_1](u) = x||y,$$

where

$$\begin{aligned} x &= f_1(u), \\ y &= u_d x^d + u_{d-1} x^{d-1} + \cdots + u_1 x + u_0, \end{aligned}$$

with all field operations taking place in $\mathbb{F}_{2^n}$ (and u_d being viewed as embedded in $\{0, 1\}^n$). We call $\phi[f_1]$ a *polynomial-based compression function.* This design is due to Stam [22,23].

Given an independent compression function $f_2 : \{0,1\}^{2n+c} \rightarrow \{0,1\}^n$, we define a hash function

$$\begin{aligned} H[f_1, f_2] : \{0,1\}^* &\longrightarrow \{0,1\}^n \\ M &\longmapsto f_2\left(0^c || v\right), \end{aligned}$$

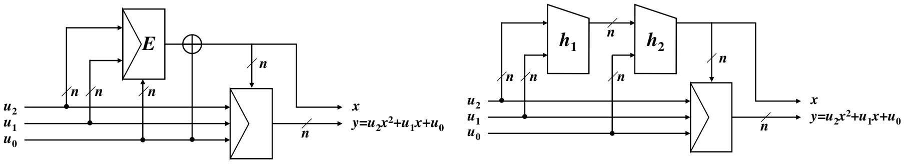
where $v = MD[\phi[f_1]](M)$, the Merkle-Damgård iteration of $\phi[f_1]$ on message M (with the usual "strengthened" padding for M that appends the length of the message—see Section 2 for details). The scheme is pictured for $c = n$ in Figure 1(b).

We comment at this point that our mode of operation uses two distinct primitives instead of a single primitive f_1 as do Lucks and Yasuda. As such, our construction explicitly follows the framework of An and Bellare [1] for proving unforgeability whereas Yasuda adopts it implicitly: with some extra work, one can use $f_1 = f_2$ because the f_2-queries are (with very high likelihood) all independent from f_1-queries, due to the presence of the 0^c input segment. (This technique for reducing key material was first used by Maurer and Sjödin [19].) We opt for using two primitives because it simplifies the proofs and allows separation of the security properties required by f_1 and f_2 (the security requirements for f_1 being often much less than those for f_2).

The following points summarize our results on $\phi[f_1]$ and $H[f_1, f_2]$. For this summary we say that f_i is *unforgeable* to mean that a computationally bounded adversary with oracle access to f_i has low probability of predicting the output of f_i on an unqueried value when f_i is sampled from a keyed function family (as low as for a random function of the same range). The *query complexity* of an attack on a variable input length (VIL) function is the number of queries to the underlying primitive necessary to compute the answers to the adversary's queries.

1. We prove that $\phi[f_1]$ is collision resistant up to $O(2^n/n)$ queries to f_1 as long as f_1 is unforgeable. This result also implies the collision resistance of $\phi[f_1]$ against an information-theoretic adversary if f_1 is a random function. It also implies $H[f_1, f_2]$ is unforgeable up to $O(2^n/n)$ query complexity as long as f_1 and f_2 are unforgeable, and that $H[f_1, f_2]$ is weakly collision resistant up to $O(2^n/n)$ query complexity as long as f_1 is unforgeable and f_2 is weakly collision resistant.
2. We prove that $H[f_1, f_2]$ is pseudorandom up to $O(2^n/n)$ query complexity as long as f_1 and f_2 are pseudorandom. In the complexity-theoretic model, we can weaken the assumption so that f_1 is unforgeable.
3. We prove that $\phi[f_1]$ is preimage aware[1] up to $O(2^{2n/3})$ query complexity as long as f_1 is a public random function. By the results of [6], this implies $H[f_1, f_2]$ is indifferentiable from a random oracle up to $O(2^{2n/3})$ query complexity as long as f_1 and f_2 are public random functions.

[1] *Preimage awareness* is a security notion introduced by Dodis, Ristenpart and Shrimpton [6]. The Merkle-Damgård iteration of a preimage aware compression function composed with a random function results in a construction that is indifferentiable from a random oracle, up to the preimage awareness security of the compression function and the maximum message length queried to the iterated construction.



(a) The quadratic blockcipher-based function

(b) The quadratic cascade function

Fig. 2. Variants of the quadratic compression function

Refinements. As mentioned, we also investigate two variants of the $3n$-to-$2n$ bit polynomial-based compression function (a.k.a. the "quadratic" compression function) with a view towards concrete implementations of the mode. These alternate constructions are shown in Figure 2. The first variant replaces f_1 by a blockcipher E of $2n$-bit key in Davies-Meyer mode. We show this compression function $\psi[E]$ is collision resistant up to $O(2^n/n)$ queries as long as E is "unpredictable", a notion we discuss in Section 5. Similar corollaries follow on the security of the hash function obtained by iterating $\psi[E]$.

The second is obtained by replacing f_1 with the cascade of two $2n$-bit to n-bit compression functions h_1 and h_2. We show this compression function, denoted $\tau[h_1, h_2]$, is collision resistant up to $O(2^n/n^3)$ queries as long as h_1 and h_2 are unforgeable. It follows that the hash function $G[h_1, h_2, f_2]$ obtained by iterating $\tau[h_1, h_2]$ (defined like $H[f_1, f_2]$ but substituting $\tau[h_1, h_2]$ for $\phi[f_1]$) has unforgeability security up to $O(2^n/n^3)$ query complexity when h_1, h_2 and f_2 are unforgeable, has collision security up to $O(2^n/n^3)$ query complexity when h_1, h_2 are unforgeable and f_2 is collision resistant, and is indistinguishable from a PRF up to $O(2^n/n^3)$ query complexity when h_1, h_2 are unforgeable and f_2 is pseudorandom.

Related Work. All the compression functions discussed in this paper, including the cascaded and blockcipher variants, were proposed by Stam [22,23]. In [23] Stam proves the collision resistance of polynomial-based compression functions of degrees two and three in the random function model, and also proves the collision security of the quadratic blockcipher mode in the ideal cipher model. Here our contribution is that we weaken the model by showing collision resistance is already assured when f_1 and E are unforgeable/unpredictable rather than random. (It is this weakening of the model that allows us to prove MAC-preservation results for the resulting hash functions.) Regarding the quadratic cascade compression function, Stam proves collision resistance for a special class of non-adaptive adversaries assuming random primitives. Our analysis supports fully adaptive adversaries and once again weakens the model to unforgeable primitives.

Lucks [16] recently proposed a double-pipe hash function iterating a $3n$-bit to $2n$-bit compression function which, like the quadratic blockcipher-based mode, uses a single call to a blockcipher of $2n$-bit key. However, by contrast to the quadratic blockcipher compression function, Lucks' compression function is neither collision resistant nor preimage resistant. As a consequence, collision and preimage security can only be proved in the iteration (higher security notions like indifferentiability are unaddressed). On the other hand, for $n = 128$ Lucks gives a better explicit collision security bound than we do for the quadratic blockcipher compression function: 2^{122} versus 2^{119} queries, respectively.

This paper can be seen as an extension of Yasuda's work [24] since our main achievement is to double the rate of that construction while maintaining comparable MAC-preservation and PRF-preservation properties. However, from a technical standpoint we owe most to Dodis and Steinberger [7], whose "multicollision-to-forgery" (MTF) balls-in-bins games are used in nearly all of our analyses (the sole exception being the preimage awareness bound for polynomial-based compression functions). Indeed, the main "message" of this paper may well be the versatility and power of MTF games.

2 Preliminaries

$\mathbb{F}_{2^n}$ denotes a finite field of order 2^n. Throughout our work, we will identify $\mathbb{F}_{2^n}$ and $\{0,1\}^n$, assuming a fixed mapping between the two sets. For $u \in \{0,1\}^*$, $|u|$ is the length in bits of u. For two bitstrings u and v, $u||v$ denotes the concatenation of u and v. For a set U, we write $u \overset{\$}{\leftarrow} U$ to denote uniform sampling from U and assignment to u.

Let $M \in \{0,1\}^*$ and let $c \geq 1$ be message block length (as c will denote throughout the paper). Then $\mathsf{pad}(M) := M||10^b||\langle l \rangle$ where b is the least integer such that $|M||10^b|$ is a multiple of c and where l is the number of c-bit blocks in $M||10^b$. (This representation is possible as long as $l < 2^c$, but this is not a restriction for most applications.) The main property of $\mathsf{pad}(\cdot)$ is that it gives a suffix-free encoding of messages.

The (strengthened) *Merkle-Damgård transform* produces a VIL-function $MD[F] : \{0,1\}^* \to \{0,1\}^n$ from a FIL-function $F : \{0,1\}^{n+c} \to \{0,1\}^n$. Given a predetermined constant $IV \in \{0,1\}^n$, the function $MD[F]$ is defined as follows.

```
Function $MD[F](M)$
  $z[0] \leftarrow IV$
  Break $\mathsf{pad}(M)$ into $c$-bit blocks, $\mathsf{pad}(M) = M[1]||\ldots||M[l+1]$
  for $i \leftarrow 1$ to $l+1$ do
      $z[i] \leftarrow F(z[i-1]||M[i])$
  return $z[l+1]$
```

3 Security Definitions

Unforgeability and Weak Collision Resistance. A *function family* is a map $f : \{0,1\}^\kappa \times \mathsf{Dom}(f) \to \{0,1\}^n$ where $\mathsf{Dom}(f) \subset \{0,1\}^*$. The bitstrings in $\{0,1\}^\kappa$ are the *keys* of f and we write $f_k(M)$ for $f(k,M)$ for $k \in \{0,1\}^\kappa$ and

$M \in \mathsf{Dom}(f)$. The function f_k is called a *member* of f. The unforgeability of f as a secure message authentication code (MAC) is estimated by the experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{mac}}$ described in Figure 3(a). In the experiment, an adversary $\mathcal{A}$ has oracle access to $f_k(\cdot)$ and tries to produce a valid tag z for a new message M. Here we call a message M "new" if it has not been queried to oracle $f_k(\cdot)$. The *forgery advantage* of $\mathcal{A}$ is defined by

$$\mathbf{Adv}_{f}^{\mathsf{mac}}(\mathcal{A}) = \mathbf{Pr}\left[\mathbf{Exp}_{\mathcal{A}}^{\mathsf{mac}} = 1\right]. \tag{1}$$

The probability is taken over the random choice of k and $\mathcal{A}$'s coins (if any). We define $\mathbf{Adv}_{f}^{\mathsf{mac}}(t, q, \mu)$ as the maximum of $\mathbf{Adv}_{f}^{\mathsf{mac}}(\mathcal{A})$ over all adversaries $\mathcal{A}$ making at most q queries whose total combined length is at most μ bits (including the forgery produced by $\mathcal{A}$) and of "running time" at most t. The "running time" is defined to be the total running time of the experiment, including the time required to compute the answers to $\mathcal{A}$'s queries. We write $\mathbf{Adv}_{f}^{\mathsf{mac}}(t, q)$ for $\mathbf{Adv}_{f}^{\mathsf{mac}}(t, q, \mu)$ if f is a family of fixed input length functions, as in this case μ is automatically determined by q.

The weak collision resistance (WCR) of f is estimated by the experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{wcr}}$ described in Figure 3(b). In contrast to the definition of collision resistance (in the dedicated-key setting) where $\mathcal{A}$ is provided key k, $\mathcal{A}$ is allowed only oracle access to $f_k(\cdot)$. Let

$$\mathbf{Adv}_{f}^{\mathsf{wcr}}(\mathcal{A}) = \mathbf{Pr}\left[\mathbf{Exp}_{\mathcal{A}}^{\mathsf{wcr}} = 1\right]. \tag{2}$$

Then the weak collision resistance of f, denoted $\mathbf{Adv}_{f}^{\mathsf{wcr}}(t, q, \mu)$, is defined to be the maximum of $\mathbf{Adv}_{f}^{\mathsf{wcr}}(\mathcal{A})$ over all adversaries $\mathcal{A}$ making at most q queries whose total combined length is at most μ bits and of running time at most t. When f is a family of fixed input length functions we likewise write $\mathbf{Adv}_{f}^{\mathsf{wcr}}(t, q)$ instead of $\mathbf{Adv}_{f}^{\mathsf{wcr}}(t, q, \mu)$.

Our security proof for unforgeability will follow the approach developed by An and Bellare [1]. One of their results is that $f_2 \circ MD[f_1]$ is a VIL-MAC if f_1 is a FIL-WCR and f_2 is a FIL-MAC. With a slight modification, we summarize Lemma 4.2 and Lemma 4.3 in [1] as the following lemma.

Lemma 1. *Let $f^1 : \{0,1\}^{\kappa} \times \{0,1\}^{n+c} \to \{0,1\}^{n}$ and $f^2 : \{0,1\}^{\kappa'} \times \{0,1\}^{n} \to \{0,1\}^{m}$ be function families. Then,*

$$\mathbf{Adv}_{f^2 \circ MD[f^1]}^{\mathsf{mac}}(t, \tilde{q}, \mu) \leq \mathbf{Adv}_{f^2}^{\mathsf{mac}}(t, \tilde{q}) + \mathbf{Adv}_{f^1}^{\mathsf{wcr}}\left(t, \left\lfloor \frac{\mu}{c} \right\rfloor + 2\tilde{q}\right).$$

Remark 1. $\left\lfloor \frac{\mu}{c} \right\rfloor + 2\tilde{q}$ is the maximum number of queries to f_1 required to compute $MD[f_1](x_i)$ for $x_1, \ldots, x_{\tilde{q}}$ such that $|x_1| + \ldots + |x_{\tilde{q}}| \leq \mu$.

Indifferentiability and Indistinguishability. In the indifferentiability framework, a distinguisher is given two systems $(F[\mathcal{P}], \mathcal{P})$ and $(\mathcal{H}, \mathcal{S}[\mathcal{H}])$. Here $\mathcal{P}$ is an ideal primitive used as a building block for the construction of $F[\mathcal{P}]$. An ideal primitive $\mathcal{H}$ and a probabilistic Turing machine $\mathcal{S}[\mathcal{H}]$ with oracle access

```
Experiment $\mathbf{Exp}^{\mathsf{mac}}_{\mathcal{A}}$
    $k \xleftarrow{\$} \{0,1\}^{\kappa}$
    $(M, z) \leftarrow \mathcal{A}^{f_k(\cdot)}$
    if $M$ is new and $f_k(M) = z$ then
        output 1
    else
        output 0
```

(a) Quantification of unforgeability

```
Experiment $\mathbf{Exp}^{\mathsf{wcr}}_{\mathcal{A}}$
    $k \xleftarrow{\$} \{0,1\}^{\kappa}$
    $(M, M') \leftarrow \mathcal{A}^{f_k(\cdot)}$
    if $f_k(M) = f_k(M')$ then
        output 1
    else
        output 0
```

(b) Quantification of weak collision resistance

Fig. 3. Experiments for quantification of unforgeability and weak collision resistance

to $\mathcal{H}$ have the same interfaces as $F[\mathcal{P}]$ and $\mathcal{P}$, respectively. The *simulator* $\mathcal{S}[\mathcal{H}]$ tries to emulate the ideal primitive $\mathcal{P}$ so that no distinguisher can tell apart the two systems $(\mathcal{H}, \mathcal{S}[\mathcal{H}])$ and $(F[\mathcal{P}], \mathcal{P})$ with non-negligible probability, based on their responses to queries that the distinguisher may send. We say that the construction $F[\mathcal{P}]$ is indifferentiable from $\mathcal{H}$ if the existence of such a simulator is proved. The indifferentiability implies the absence of a generic attack against $F[\mathcal{P}]$ that regards $\mathcal{P}$ merely as a black-box. Now we give a formal definition of indifferentiability in the information-theoretic model. For a more comprehensive introduction of the indifferentiability framework, we refer to [3,18].

Definition 1. *A Turing machine F with oracle access to an ideal primitive $\mathcal{P}$ is said to be* (q, ϵ, t)-indifferentiable from an ideal primitive $\mathcal{H}$ *if there exists a simulator $\mathcal{S}$ of running time at most t with oracle access to $\mathcal{H}$ such that for any distinguisher $\mathcal{A}$ making at most q queries, it holds that*

$$\left|\mathbf{Pr}\left[\mathcal{A}^{F[\mathcal{P}],\mathcal{P}} = 1\right] - \mathbf{Pr}\left[\mathcal{A}^{\mathcal{H},\mathcal{S}[\mathcal{H}]} = 1\right]\right| < \epsilon.$$

If $\mathcal{H}$ is a public random function, then $F[\mathcal{P}]$ is called a (q, ϵ, t)-pseudorandom oracle (PRO)*.*

If $\mathcal{A}$ is not allowed to make queries for the underlying primitive, we obtain the definition of indistinguishability.

Definition 2. *A Turing machine F with oracle access to an ideal primitive $\mathcal{P}$ is said to be* (q, ϵ)-indistinguishable from an ideal primitive $\mathcal{H}$ *if for any distinguisher $\mathcal{A}$ making at most q queries, it holds that*

$$\left|\mathbf{Pr}\left[\mathcal{A}^{F[\mathcal{P}]} = 1\right] - \mathbf{Pr}\left[\mathcal{A}^{\mathcal{H}} = 1\right]\right| < \epsilon.$$

If $\mathcal{H}$ is a public random function, then $F[\mathcal{P}]$ is called a (q, ϵ)-pseudorandom function (PRF)*.*

Collision Resistance and Adaptive Preimage Resistance. First, we review the definition of collision resistance *in the information-theoretic model.*

```
Experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{coll}}$
    $\mathcal{A}^{\mathcal{P}}$ updates $\mathcal{Q}$
    if $\exists\ u \neq u',\ z$ s.t. $z = F_{\mathcal{Q}}(u) = F_{\mathcal{Q}}(u')$ then
        output 1
    else
        output 0
```

(a) Quantification of collision resistance

```
Experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{adpr}}$
    $\mathcal{A}^{\mathcal{P}}$ updates $\mathcal{Q}$ and $\mathcal{L}$
    if $\exists\ u$ s.t. $z = F_{\mathcal{Q}}(u)$ for some $z \in \mathcal{L}$
    then
        output 1
    else
        output 0
```

(b) Quantification of adaptive preimage resistance

Fig. 4. Experiments for quantification of collision resistance and adaptive preimage resistance

Given a function $F = F[\mathcal{P}]$ and an information-theoretic adversary $\mathcal{A}$ both with oracle access to an ideal primitive $\mathcal{P}$, the collision resistance of F against $\mathcal{A}$ is estimated by the experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{coll}}$ described in Figure 4(a). The experiment records every query-response pair that $\mathcal{A}$ obtains by oracle queries into a *query history* $\mathcal{Q}$. We write $z = F_{\mathcal{Q}}(u)$ if $\mathcal{Q}$ contains all the query-response pairs required to compute $z = F(u)$. At the end of the experiment, $\mathcal{A}$ would like to find two distinct evaluations yielding a collision. The *collision-finding advantage* of $\mathcal{A}$ is defined to be

$$\mathbf{Adv}_{F}^{\mathsf{coll}}(\mathcal{A}) = \mathbf{Pr}\left[\mathbf{Exp}_{\mathcal{A}}^{\mathsf{coll}} = 1\right]. \tag{3}$$

The probability is taken over the random choice of $\mathcal{P}$ and $\mathcal{A}$'s coins (if any). For $q > 0$, we define $\mathbf{Adv}_{F}^{\mathsf{coll}}(q)$ as the maximum of $\mathbf{Adv}_{F}^{\mathsf{coll}}(\mathcal{A})$ over all adversaries $\mathcal{A}$ making at most q queries.

In this section, we also present a new notion of security, called *adaptive preimage resistance*. This notion will be useful for the proof of preimage awareness. Given a function $F = F[\mathcal{P}]$ and an information-theoretic adversary $\mathcal{A}$ both with oracle access to an ideal primitive $\mathcal{P}$, the adaptive preimage resistance of F against $\mathcal{A}$ is estimated by the experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{adpr}}$ described in Figure 4(b). At any point during the experiment, the adversary $\mathcal{A}$ is allowed to choose a commitment element z in the range of F such that the query history $\mathcal{Q}$ has not determined any preimage of z. The experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{adpr}}$ records the element z into a *commitment list* $\mathcal{L}$. Queries and commitments are made in an arbitrarily interleaved order. At the end of the experiment, $\mathcal{A}$ would like to succeed in finding a preimage of some element in the commitment list. The *adaptive preimage-finding advantage* of $\mathcal{A}$ is defined to be

$$\mathbf{Adv}_{F}^{\mathsf{adpr}}(\mathcal{A}) = \mathbf{Pr}\left[\mathbf{Exp}_{\mathcal{A}}^{\mathsf{adpr}} = 1\right]. \tag{4}$$

For $q_p, q_e > 0$, we define $\mathbf{Adv}_{F}^{\mathsf{adpr}}(q_p, q_e)$ as the maximum of $\mathbf{Adv}_{F}^{\mathsf{adpr}}(\mathcal{A})$ over all adversaries $\mathcal{A}$ that make at most q_p queries and at most q_e commitments.

Preimage Awareness. The notion of preimage awareness was first introduced by Dodis, Ristenpart and Shrimpton [6]. This notion is useful for the proof of indifferentiability of "NMAC" type constructions. Let $F = F[\mathcal{P}]$ be a function with oracle access to an ideal primitive $\mathcal{P}$. In order to estimate the preimage awareness of F, we use the experiment described in Figure 5. Here an adversary $\mathcal{A}$ is provided two oracles P and Ex. The oracle P provides access to the ideal primitive $\mathcal{P}$ and records a query history $\mathcal{Q}$. Note that this oracle is implicitly used in the experiments for collision resistance and adaptive preimage resistance. The *extraction oracle* Ex provides an interface to an *extractor* $\mathcal{E}$, which is a deterministic algorithm that takes as input an element z in the range of F and the query history $\mathcal{Q}$, and returns either $\perp$ or an element in the domain of F. With respect to the extractor $\mathcal{E}$, we define the advantage of $\mathcal{A}$ to be

$$\mathbf{Adv}_F^{\mathsf{pra}}(\mathcal{A}, \mathcal{E}) = \mathbf{Pr}\left[\mathbf{Exp}_{\mathcal{A},\mathcal{E}}^{\mathsf{pra}} = 1\right]. \tag{5}$$

Let $\mathcal{E}^*$ be an algorithm that on input $(z, \mathcal{Q})$ returns an element u if there exists u such that $F_{\mathcal{Q}}(u) = z$ and $\perp$ otherwise. Let

$$\mathbf{Adv}_F^{\mathsf{pra}}(\mathcal{A}, \mathcal{E}^*) = \mathsf{P}_1 + \mathsf{P}_2,$$

where P_1 is the probability that $u \neq \mathsf{V}[z] \neq \perp$ and $\mathsf{L}[z] = 1$ at the end of the experiment and P_2 is the probability that $u \neq \mathsf{V}[z] = \perp$ and $\mathsf{L}[z] = 1$ at the end of the experiment. Then $\mathcal{A}$ can be regarded as a collision-finding adversary such that $\mathbf{Adv}_F^{\mathsf{coll}}(\mathcal{A}) = \mathsf{P}_1$. Furthermore, $\mathcal{A}$ can be transformed into an adaptive preimage-finding adversary $\mathcal{B}$ such that $\mathbf{Adv}_F^{\mathsf{adpr}}(\mathcal{B}) = \mathsf{P}_2$: $\mathcal{B}$ runs $\mathcal{A}$ as a subroutine, asks the same primitive queries as $\mathcal{A}$, and makes commitments z if $\mathcal{A}$ makes a query for $\mathsf{Ex}(z)$ and $\mathcal{Q}$ has not determined any preimage of z. If $\mathcal{A}$ makes at most q_p primitive queries and q_e extraction queries, then it follows that $\mathsf{P}_1 \leq \mathbf{Adv}_F^{\mathsf{coll}}(q_p)$ and $\mathsf{P}_2 \leq \mathbf{Adv}_F^{\mathsf{adpr}}(q_p, q_e)$. We record this observation as the following lemma.

Lemma 2. *Let $F = F[\mathcal{P}]$ be a function with oracle access to an ideal primitive $\mathcal{P}$. Then there exists an extractor $\mathcal{E}^*$ such that for any adversary $\mathcal{A}$ it holds that*

$$\mathbf{Adv}_F^{\mathsf{pra}}(\mathcal{A}, \mathcal{E}^*) \leq \mathbf{Adv}_F^{\mathsf{coll}}(q_p) + \mathbf{Adv}_F^{\mathsf{adpr}}(q_p, q_e).$$

The main application of preimage awareness lies in the construction of pseudorandom oracles. In the following lemma which is a combination of Theorem 4.1 and Theorem 4.2 in [6], unpad is an algorithm such that $\mathsf{unpad}(y) = x$ if $y = \mathsf{pad}(x)$ is a valid output of pad and $\mathsf{unpad}(y) = \perp$ otherwise. For any algorithm $\mathcal{F}$, we write $\mathsf{Time}(\mathcal{F}, l)$ for the maximum time required to compute $\mathcal{F}(x)$ for any input x such that $|x| \leq l$. If $\mathcal{F}$ is an algorithm with oracle access to an ideal primitive $\mathcal{P}$, then $\mathsf{NQ}(F, l)$ is the maximum number of queries to $\mathcal{P}$ required to compute $F(x)$ for any input x such that $|x| \leq l$. Without any constraint on the input length, we just write $\mathsf{Time}(\mathcal{F})$ and $\mathsf{NQ}(F)$.

Lemma 3. *Let $F : \{0,1\}^{n+c} \rightarrow \{0,1\}^n$ be a function with oracle access to an ideal primitive $\mathcal{P}$ and let $g : \{0,1\}^n \rightarrow \{0,1\}^m$ and $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^m$ be public random functions for $m \leq n$. For an arbitrary extractor $\mathcal{E}$ with respect to F,*

```
Experiment $\mathbf{Exp}^{\mathsf{pra}}_{\mathcal{A},\mathcal{E}}$
    $u \leftarrow \mathcal{A}^{\mathsf{P},\mathsf{Ex}}$
    $z \leftarrow F[\mathcal{P}](u)$
    output 1 if $u \neq \mathsf{V}[z]$ and $\mathsf{L}[z] = 1$
```

```
Oracle $\mathsf{P}(x)$
    $y \leftarrow \mathcal{P}(x)$
    $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(x, y)\}$
    return $y$
```

```
Oracle $\mathsf{Ex}(z)$
    $\mathsf{L}[z] \leftarrow 1$
    $\mathsf{V}[z] \leftarrow \mathcal{E}(z, \mathcal{Q})$
    return $\mathsf{V}[z]$
```

Fig. 5. Experiments for quantification of preimage awareness. Arrays L and V are global, and respectively initialized empty and $\perp$ everywhere.

there exists a simulator $\mathcal{S}$ such that for any distinguisher $\mathcal{A}$ making at most (q_0, q_1, q_2) queries to the three oracle interfaces associated with $(\mathcal{H}, \mathcal{P}, g)$, there exists an adversary $\mathcal{B}$ such that

$$\left|\mathbf{Pr}\left[\mathcal{A}^{g \circ MD[F[\mathcal{P}]],(\mathcal{P},g)} = 1\right] - \mathbf{Pr}\left[\mathcal{A}^{\mathcal{H},\mathcal{S}[\mathcal{H}]} = 1\right]\right| \leq \mathbf{Adv}^{\mathsf{pra}}_{F}(\mathcal{B}, \mathcal{E}).$$

Let l_{max} be the length in bits of the longest query made by $\mathcal{A}$ to its first oracle, and let $L = \left\lceil \frac{l_{max}+1}{c} + 1 \right\rceil$. Then, simulator $\mathcal{S}$ runs in time $O\left(q_1 + Lq_2\mathsf{Time}(\mathcal{E}) + Lq_2\mathsf{Time}(\mathsf{unpad})\right)$. Adversary $\mathcal{B}$ runs in time $O\left(\mathsf{Time}(\mathcal{A}) + q_0\mathsf{Time}(MD[F], l_{max}) + q_1 + (L+1)q_2\right)$, makes at most $L\mathsf{NQ}(F)(q_0+1)+q_1$ primitive queries, and makes at most Lq_2 extraction queries.

4 Security of the Polynomial-Based Mode of Operation

For this section and the rest of the paper, $\phi[f_1]$ and $H[f_1, f_2]$ refer to the compression function and hash function defined in Section 1. We use "log" to denote the logarithm base 2. For simplicity of notation, we assume that $\log q$ is an integer for the number of queries q.

4.1 Weak Collision Resistance and Unforgeability

We begin with the proof of weak collision resistance for $\phi[f_1]$ such that f_1 is randomly chosen from a function family f.

Theorem 1. *Let ϕ be a function family defined by $f : \{0,1\}^{\kappa} \times \{0,1\}^{2n+c} \rightarrow \{0,1\}^{n}$. Then,*

$$\mathbf{Adv}^{\mathsf{wcr}}_{\phi}\,(t, q) \leq 2q \max(d + \log q, d2^{d+2})\mathbf{Adv}^{\mathsf{mac}}_{f}\left(t + O(d^2 n^2 q^{d+2}) + \mathsf{Time}_d, q\right),$$

where $d = \left\lceil \frac{2n+c}{n} \right\rceil - 1$ and Time_d is the time required to solve a univariate polynomial equation of degree d over $\mathbb{F}_{2^n}$.

Remark 2. For a univariate polynomial of degree d over $\mathbb{F}_{2^n}$, there is a deterministic algorithm to find zeros using $O(d^{3/2}n)$ field operations (ignoring log factors). See [8,9].

In order to prove Theorem 1, we use a generalization of the multicollision-to-forgery (MTF) balls-in-bins game first introduced in [7].

MTF game. This game is played by two players $\mathcal{A}$ and $\mathcal{B}$ according to the following rules. Parameters are integers $q > 0$ and $m_2 > m_1 \geq 0$.

1. The game consists of q rounds.
2. At each round, $\mathcal{A}$ publicly places a set of balls into a set of bins such that
   (a) balls placed at the same round go into distinct bins,
   (b) the number of balls that are placed into bins already containing m_1 balls at the moment of placement is finite,
   (c) some bin eventually contains more than m_2 balls.
3. Before each round, $\mathcal{B}$ can secretly "pass" or "guess" a bin that will receive a ball in the next round. $\mathcal{B}$ makes exactly one guess throughout the game.
4. If $\mathcal{B}$ makes a correct guess, then $\mathcal{B}$ wins. Otherwise, $\mathcal{B}$ loses.

Note there is no constraint on the total number of balls or bins.

Lemma 4. *Irrespective of $\mathcal{A}$'s strategy, there exists a strategy for $\mathcal{B}$ to win the above game with probability at least $1/q \cdot 1/(c_{m_1})^{1/(m_2-m_1)}$, where c_{m_1} is the number of balls that are placed into bins already containing m_1 balls at the moment of placement.*

Proof. $\mathcal{B}$'s strategy is simple, as follows:

1. Choose a round $i \in \{1, \ldots, q\}$ and a level $j \in \{m_1 + 1, \ldots, m_2\}$ uniformly at random.
2. Before the i-th round of the game, guess a bin uniformly at random from all bins containing at least j balls already.

Let c_j be the total number of balls that are placed into bins that already have at least j balls in them right before the round when the ball is placed. For a given value of j, each ball placed into a bin with at least j balls in it already gives $\mathcal{B}$ chance at least $1/(qc_{j-1})$ of winning since c_{j-1} is an upper bound for the number of bins that have j balls in them at the end of the game. Therefore $\mathcal{B}$'s chance of winning is at least $c_j/(qc_{j-1})$ for a given value of j, and hence $\mathcal{B}$'s overall chance of winning is at least

$$\begin{aligned} \frac{1}{m_2 - m_1} \sum_{j=m_1+1}^{m_2} \frac{c_j}{qc_{j-1}} &= \frac{1}{q} \mathsf{ArithmeticMean}\left(\frac{c_{m_1+1}}{c_{m_1}}, \ldots, \frac{c_{m_2}}{c_{m_2-1}}\right) \\ &\geq \frac{1}{q} \mathsf{GeometricMean}\left(\frac{c_{m_1+1}}{c_{m_1}}, \ldots, \frac{c_{m_2}}{c_{m_2-1}}\right) \\ &\geq \frac{1}{q}\left(\frac{1}{c_{m_1}}\right)^{1/(m_2-m_1)} . \end{aligned}$$ □

Note that Lemma 4 asserts nothing about $\mathcal{B}$'s efficiency—in fact, if a huge number of balls are thrown, it may be difficult to keep track of all bins that have received at least j balls already (which is necessary for sampling uniformly among the bins).

In our case, bins will often be curves defined by polynomials of degree $\leq d$ over $\mathbb{F}_{2^n}$ and balls points in $\mathbb{F}_{2^n} \times \mathbb{F}_{2^n}$, where a ball (x, y) goes into bin $\mathcal{C}$ if $(x, y) \in \mathcal{C}$ (a ball is thus "cloned" into many different bins). In this setting, it becomes easier to keep track of which bins have at least j balls in them when $j \geq d+1$, as $d+1$ points uniquely determine a polynomial of degree $\leq d$. Thus for such a game it may be helpful to set $m_1 = d$, in order to keep the complexity of sampling under control. (Dodis and Steinberger do not have games in which the number of bins containing balls is so large that sampling for small values of j is an issue, and always use MTF games with $m_1 = 0$.) The value of m_2 is then set large enough to make the term $(1/c_{m_1})^{1/(m_2-m_1)}$ small.

We typically upper bound c_{m_1} by qM where M is an upper bound on *the total number of bins with at least* $m_1 + 1$ *balls at the end of the game*. Indeed, because balls are thrown into distinct bins at each round, this definition of M implies that at each round at most M balls are thrown into bins with m_1 balls in them already. We thus have the following corollary:

Corollary 1. *If the number of bins that contain at least* $m_1 + 1$ *balls at the end of the MTF game is at most* M*, then* $\mathcal{B}$ *can win the MTF game with probability at least* $1/q \cdot 1/(qM)^{1/(m_2-m_1)}$.

We note that Corollary 1 is a bit wasteful, in the sense that it is possible to give a better bound for $\mathcal{B}$'s chance of success as a function of m_1, m_2 and M from the relationship $M = c_{m_1} - c_{m_1+1}$ and the fact that $\mathcal{B}$'s chance of success is also lower bounded by $\frac{\alpha}{q(m_2-m_1)}$ where $\alpha = \max\left(\frac{c_{m_1+1}}{c_{m_1}}, \ldots, \frac{c_{m_2}}{c_{m_2-1}}\right)$. However this gain leads to a more complicated statement and is minor enough for us to ignore [2].

Proof (Theorem 1). Let $\mathcal{A}$ be a weak collision-finding adversary such that

$$\mathbf{Adv}_{\phi}^{\mathsf{wcr}}(\mathcal{A}) = \mathbf{Adv}_{\phi}^{\mathsf{wcr}}(t, q) = \epsilon.$$

We write $u[i] = u_d[i] || \cdots || u_0[i]$ for the i-query that $\mathcal{A}$ makes to $f_1(\cdot)$ and $\phi(u[i]) = (x[i], y[i])$, where $x[i] = f_1(u[i])$ and $y[i] = u_d[i]x[i]^d + \cdots + u_1[i]x[i] + u_0[i]$. The i-th query is associated with a curve

$$\mathcal{C}_i = \{(x, y) \in \mathbb{F}_{2^n}^2 : y = u_d[i]x^d + \cdots + u_1[i]x + u_0[i]\}.$$

Let $\Gamma_i = \{1 \leq j \leq i : (x[j], y[j]) \in \mathcal{C}_i\}$ and let $\gamma = \max_i |\Gamma_i|$. By assumption that $\mathcal{A}$ succeeds to find a collision with probability ϵ, one of the following two events occurs with probability at least $\epsilon/2$.

Case 1: $\mathcal{A}$ finds a collision and $\gamma \leq d + \log q$. For this case, we can construct a forger $\mathcal{B}_1$ for f_1 as follows.

[2] More precisely, we have $M = c_{m_1} - c_{m_1+1} \geq c_{m_1}(1-\alpha)$ or $c_{m_1} \leq M/(1-\alpha)$. Then $\mathcal{B}$'s chance of success is at least $\frac{1}{q}\max\left(\frac{\alpha}{m_2-m_1}, \left(\frac{1-\alpha}{M}\right)^{1/(m_2-m_1)}\right)$ where we know $0 < \alpha \leq 1$.

1. $\mathcal{B}_1$ chooses $i \in \{1, \ldots, q\}$ and $s \in \{1, \ldots, d + \log q\}$ uniformly at random.
2. $\mathcal{B}_1$ runs $\mathcal{A}$ as a subroutine and faithfully answers the queries made by $\mathcal{A}$ until the $(i-1)$-th query.
3. On the i-query $u[i]$, $\mathcal{B}_1$ presents a forgery $(u[i], x[j_s])$ without making a query to $f_1(\cdot)$, where j_s is the s-th element of $\mathit{\Gamma}_i$. (If $|\mathit{\Gamma}_i| < s$, then $\mathcal{B}_1$ presents a random value.)

Note that if there exists a collision $(x[j], y[j]) = (x[i], y[i])$ for $j < i$, then $(x[j], y[j]) \in \mathcal{C}_i$ or equivalently $j \in \mathit{\Gamma}_i$. With this observation, we obtain

$$\mathbf{Adv}_f^{\mathsf{mac}}(\mathcal{B}_1) \geq \frac{\epsilon}{2q(d + \log q)}. \tag{6}$$

Case 2: $\mathcal{A}$ produces $\gamma > d + \log q$. This is the case where we construct a forger $\mathcal{B}_2$ for f_1 using the MTF game: The bins are $(d+1)$-tuples $(u_d, \ldots, u_0) \in \mathbb{F}_{2^n}^{d+1}$ (regarded as a curve in the plane $\mathbb{F}_{2^n}^2$) and the balls are points $(x, y) \in \mathbb{F}_{2^n}^2$. A query $f_1(u_d || \cdots || u_0)$ results in a new ball $(x, y) = (f_1(u_d || \cdots || u_0), u_d x^d + \cdots + u_1 x + u_0)$ being placed into all bins $(v_d, \ldots, v_0)$ such that $v_d x^d + \cdots + v_1 x + v_0 = y$, namely all bins giving the coefficients of a polynomial curve fitting the point (x, y), *except the bin* $(u_d, \ldots, u_0)$ *itself.* Thus one ball is replicated in $2^{dn} - 1$ different bins. We assume that the i-th query $u[i]$ is known to $\mathcal{B}_2$ before the i-th round of the game. Then, when $\mathcal{B}_2$ correctly guesses a bin $(v_d, \ldots, v_0)$ that will receive the new ball $(x[i], y[i])$, $\mathcal{B}_2$ has a chance to forge f_1 with probability $1/d$ since the intersection of the curves associated with $(u_d, \ldots, u_0)$ and $(v_d, \ldots, v_0)$ contains at most d elements. Here we assume the existence of an algorithm of running time Time_d to find zeros of a univariate polynomial of degree d. Let $m_1 = d$, $m_2 = d + \log q$ and $M = \binom{q}{d+1}$. Since $d + 1$ points determine a unique polynomial of degree d fitting the points, we can apply Corollary 1 to obtain a forger $\mathcal{B}_2$ of success probability

$$\mathbf{Adv}_f^{\mathsf{mac}}(\mathcal{B}_2) \geq \frac{\epsilon}{2} \cdot \frac{1}{d} \cdot \frac{1}{q} \left(\frac{1}{qM}\right)^{1/(m_2 - m_1)} \geq \frac{\epsilon}{2} \cdot \frac{1}{d} \cdot \frac{1}{q} \left(\frac{1}{q^{d+2}}\right)^{1/\log q} = \frac{\epsilon}{dq2^{d+3}}, \tag{7}$$

and of running time $O(d^2 n^2 q^{d+2}) + \mathsf{Time}_d$. From (6) and (7), it follows that

$$\mathbf{Adv}_\phi^{\mathsf{wcr}}(t, q) \leq 2q \max(d + \log q, d2^{d+2}) \mathbf{Adv}_f^{\mathsf{mac}}\left(t + O(d^2 n^2 q^{d+2}) + \mathsf{Time}_d, q\right).$$

□

The following theorem is immediate from Lemma 1 and Theorem 1.

Theorem 2. *Let $H = H[f_1, f_2]$ be a function family where f_1 and f_2 are independently chosen from two function families f^1 and f^2, respectively. Then for $q = \lfloor \mu/c \rfloor + 2\tilde{q}$,*

$$\mathbf{Adv}_H^{\mathsf{mac}}(t, \tilde{q}, \mu) \leq \epsilon,$$

where

$$\epsilon = \mathbf{Adv}_{f^2}^{\mathsf{mac}}(t, \tilde{q}) + 2q \max\left(d + \log q, d2^{d+2}\right) \mathbf{Adv}_{f^1}^{\mathsf{mac}}\left(t + O(d^2 n^2 q^{d+2}) + \mathsf{Time}_d, q\right),$$

and Time_d *is the time required to solve a univariate polynomial equation of degree* d *over* $\mathbb{F}_{2^n}$*. If* $f_1 = f_2$ *are chosen from the same function family* f*, then*

$$\mathbf{Adv}_{H^*}^{\mathsf{mac}}(t, \tilde{q}, \mu) \leq \epsilon + q2^{\max\{0,n-c\}}\mathbf{Adv}_f^{\mathsf{mac}}(t, q),$$

where ϵ *is as above with* f *replacing* f^1 *and* f^2*.*

In the single-key setting, we assume that $IV_1 \neq 0^n$ for two n-bit blocks IV_1 and IV_2 such that $IV = IV_1||IV_2$. Then we can use the techniques employed in the CS construction [19]. The term $q2^{\max\{0,n-c\}}\mathbf{Adv}_f^{\mathsf{mac}}(t, q)$ comes from the case where $f_1(M[i]) = 0^{\min\{n,c\}}||*$ for some message block $M[i]$ during the Merkle-Damgård iteration.

4.2 Collision Resistance and Indistinguishability

Let Φ_{2n+c}^n be the set of all functions from $\{0,1\}^{2n+c}$ to $\{0,1\}^n$. Then Φ_{2n+c}^n can be regarded as a function family $f^* : \{0,1\}^\kappa \times \{0,1\}^{2n+c} \rightarrow \{0,1\}^n$ by identifying Φ_{2n+c}^n and $\{0,1\}^\kappa$ for $\kappa = n2^{2n+c}$. The weak collision resistance of ϕ defined by f^* against a computationally unbounded adversary implies its collision resistance in the information-theoretic model (due to the equivalence of oracle access to either ϕ or f^*). Since

$$\mathbf{Adv}_{f^*}^{\mathsf{mac}}(t, q) = \frac{1}{2^n}$$

for any q and t, the following theorem is immediate from Theorem 1.

Theorem 3. *If* $f_1 : \{0,1\}^{2n+c} \rightarrow \{0,1\}^n$ *is a random function, then*

$$\mathbf{Adv}_{\phi[f_1]}^{\mathsf{coll}}(q) \leq \frac{\max\left(d + \log q, d2^{d+2}\right) q}{2^{n-1}}.$$

When we construct NMAC type pseudorandom oracles based on preimage aware functions, adaptive preimage resistance is only needed for the case where a distinguisher makes a query to the finalization function. If there is no interface to access the inner primitive, we do not need to worry about adaptive preimage finding. The following lemma shows that any collision resistant function can be combined with a random function producing a pseudorandom function. We give the proof in Appendix A.

Lemma 5. *Let* $F : \{0,1\}^* \rightarrow \{0,1\}^n$ *be a function with oracle access to an ideal primitive* $\mathcal{P}$ *and let* $g : \{0,1\}^n \rightarrow \{0,1\}^m$ *and* $H : \{0,1\}^* \rightarrow \{0,1\}^m$ *be random functions. Then the composite function* $g \circ F$ *is* $(\tilde{q}, \epsilon)$*-indistinguishable from* H*, where* $\epsilon = \mathbf{Adv}_F^{\mathsf{coll}}(q)$*,* $q = \mathsf{NQ}(F, l_{max})\tilde{q}$ *and* l_{max} *is the length in bits of the longest query made by a distinguisher.*

Since the strengthened Merkle-Damgård transform preserves collision resistance, we obtain the following theorem.

Theorem 4. *If $f_1, f_2 : \{0,1\}^{2n+c} \to \{0,1\}^n$ are random functions, then $H[f_1, f_2]$ is $(\tilde{q}, \epsilon)$-indis-tinguishable from a random function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^n$, where*

$$\epsilon = \frac{\max\left(d + \log q, d2^{d+2}\right) q}{2^{n-1}},$$

and

$$q = \mathsf{NQ}(MD[\phi[f_1]], l_{max})\tilde{q} = \left\lceil \frac{l_{max}+1}{c} + 1 \right\rceil \tilde{q},$$

for the length in bits l_{max} of the longest query made by a distinguisher. In the single-key setting, $H[f_1, f_1]$ is $(\tilde{q}, \epsilon + \frac{q}{2^c})$-indistinguishable from $\mathcal{H}$.

Lemma 5 holds with $\epsilon = \mathbf{Adv}_F^{\mathsf{wcr}}(t, \tilde{q}, l_{max}\tilde{q})$ when F is a keyed function family (in the complexity-theoretic model). This implies that $H[f_1, f_2]$ is pseudorandom up to $O(2^n/n)$ query complexity as long as f_1 is unforgeable and f_2 is pseudorandom.

4.3 Preimage Awareness and Indifferentiability

We begin with the proof of adaptive preimage resistance for $\phi[f_1]$ where f_1 is a public random function. Let $\mathcal{A}$ be an "optimal" adaptive preimage-finding adversary that makes at most q_p queries and at most q_e commitments. That is, $\mathbf{Adv}_{\phi[f_1]}^{\mathsf{adpr}}(\mathcal{A}) = \mathbf{Adv}_{\phi[f_1]}^{\mathsf{adpr}}(q_p, q_e)$. During the experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{adpr}}$, $\mathcal{A}$ makes queries and commitments in an arbitrarily interleaved order based on a deterministic strategy. Here we can assume that the strategy does not depend on the responses of oracle $f_1(\cdot)$ to queries that $\mathcal{A}$ sends since the probability distribution of the response to a certain query is independent of the previous query-response pairs (as long as $\mathcal{A}$ does not make a redundant query). Therefore, in order to estimate $\mathbf{Adv}_{\phi[f_1]}^{\mathsf{adpr}}(\mathcal{A})$, we can use the following game.

1. $\mathcal{A}$ makes q_p queries $\{\mathcal{C}_1, \ldots, \mathcal{C}_{q_p}\}$ and q_e commitments $\mathcal{L} = \{(x_1, y_1), \ldots, (x_{q_e}, y_{q_e})\}$ based on the optimal strategy. (Here each query is represented by a curve in $\mathbb{F}_{2^n}^2$ as in the analysis of unforgeability.)
2. One point (x_i^*, y_i^*) is chosen from each curve $\mathcal{C}_i$ uniformly at random.
3. If $(x_i^*, y_i^*) \in \mathcal{L}$ for some $i = 1, \ldots, q_e$, then $\mathcal{A}$ wins. Otherwise, $\mathcal{A}$ loses.

The winning probability of $\mathcal{A}$ for the above game is equal to the adaptive preimage-finding advantage of $\mathcal{A}$. Let $\Gamma_i = \mathcal{C}_i \cap \mathcal{L}$ for $i = 1, \ldots, q_p$, and let

$$\Delta_\theta = \{1 \le i \le q_p : |\Gamma_i| \ge \theta\},$$

for a parameter θ. Then for $\Delta \subset \Delta_\theta$ and $\delta = |\Delta|$, we have

$$q_e \ge \left| \bigcup_{i \in \Delta} \Gamma_i \right| \ge \sum_{i \in \Delta} |\Gamma_i| - \sum_{i \ne j \in \Delta} |\Gamma_i \cap \Gamma_j| \ge \delta\theta - \binom{\delta}{2} \cdot d.$$

Therefore we conclude that $|\Delta_{\theta^*}| < \delta^*$ for any (θ^*, δ^*) such that

$$\delta^*\theta^* - \binom{\delta^*}{2} \cdot d > q_e. \tag{8}$$

This implies that the number of curves that intersect with $\mathcal{L}$ at $\geq \theta^*$ points is less than δ^*. Thus the winning probability of $\mathcal{A}$ is upper-bounded by

$$\mathbf{Adv}^{\mathsf{adpr}}_{\phi[f_1]}(\mathcal{A}) \leq \delta^* \frac{q_e}{2^n} + (q_p - \delta^*)\frac{\theta^*}{2^n}.$$

By Lemma 2 and Theorem 3, we obtain the following theorem.

Theorem 5. *Let (θ^*, δ^*) satisfy inequality (8). Then for a random function f_1, there exists an extractor $\mathcal{E}^*$ such that for any adversary $\mathcal{A}$ it holds that*

$$\mathbf{Adv}^{\mathsf{pra}}_{\phi[f_1]}(\mathcal{A}, \mathcal{E}^*) \leq \frac{\max\left(d + \log q_p, d2^{d+2}\right) q_p}{2^{n-1}} + \delta^* \frac{q_e}{2^n} + (q_p - \delta^*)\frac{\theta^*}{2^n}.$$

We can use Lemma 3 and Theorem 5 with $(\theta^*, \delta^*) = (dq_e^{1/2}, q_e^{1/2})$ to obtain the following theorem.

Theorem 6. *Let $f_1, f_2 : \{0,1\}^{2n+c} \to \{0,1\}^n$ and $\mathcal{H} : \{0,1\}^* \to \{0,1\}^n$ be public random functions. Then there exists a simulator $\mathcal{S}$ such that for any distinguisher $\mathcal{A}$ making at most (q_0, q_1, q_2) queries to the three oracle interfaces associated with $(\mathcal{H}, f_1, f_2)$,*

$$\left|\mathbf{Pr}\left[\mathcal{A}^{H[f_1,f_2],(f_1,f_2)} = 1\right] - \mathbf{Pr}\left[\mathcal{A}^{\mathcal{H},\mathcal{S}[\mathcal{H}]} = 1\right]\right| \leq \epsilon,$$

where

$$\epsilon = \frac{\max\left(d + \log\left(Lq_0 + q_1 + L\right), d2^{d+2}\right)}{2^{n-1}}\left(Lq_0 + q_1 + L\right) + \frac{L^{1/2}q_2^{1/2}}{2^n}\left(dLq_0 + dq_1 + Lq_2 + dL\right),$$

l_{max} is the length in bits of the longest query made by $\mathcal{A}$ to its first oracle and $L = \left\lceil \frac{l_{max}+1}{c} + 1 \right\rceil$. Simulator $\mathcal{S}$ runs in time $O\left(q_1 + Lq_2\mathsf{Time}(\mathcal{E}^*) + Lq_2\mathsf{Time}(\mathsf{unpad})\right)$, where $\mathcal{E}^*$ is the obvious extractor used in Lemma 2. In the single-key setting, we have

$$\left|\mathbf{Pr}\left[\mathcal{A}^{H[f_1,f_1],f_1} = 1\right] - \mathbf{Pr}\left[\mathcal{A}^{\mathcal{H},\mathcal{S}'[\mathcal{H}]} = 1\right]\right| \leq \epsilon + \frac{Lq_0}{2^c},$$

where simulator $\mathcal{S}'$ is obtained by a slight modification of $\mathcal{S}$: On input x, $\mathcal{S}'$ returns $f_2(x)$ if $x = 0^c||*$ for some $* \in \{0,1\}^{2n}$ and returns $f_1(x)$ otherwise, by using the interfaces f_1 and f_2 of $\mathcal{S}$.

Tightness of Indifferentiability. The preservation of indifferentiability is guaranteed only up to $O(2^{2n/3})$ query complexity which is beyond the birthday bound but still far from optimal. This bound is dominated by the adaptive preimage resistance, depending on a configuration that consists of q curves in $\mathbb{F}^2_{2^n}$ and q points on the curves (assuming $q = q_p = q_e$). If there exists a subfield

$\mathbb{F}'$ of $\mathbb{F}_{2^n}$ such that $|\mathbb{F}'| = \sqrt{q}$, then we have a configuration that provides tight adaptive preimage resistance: The set of points is $\mathbb{F}' \times \mathbb{F}' \subset \mathbb{F}_{2^n}^2$ and the set of curves consists of q polynomials of degree d with coefficients in $\mathbb{F}'$. However, for the case where $\mathbb{F}_{2^n}$ admits no proper subfield (e.g. with prime n), there remains a question whether a similar construction exists or the bound can be qualitatively improved. We pose this as an open problem.

5 The Quadratic Blockcipher-Based Compression Function

In this section and the next, we discuss how to instantiate $\phi[f_1]$ for $c = n$ (the "quadratic" polynomial mode) by replacing the $3n$-bit to n-bit compression function f_1 with a smaller primitive. First, we discuss a concrete instantiation of the quadratic compression function using a blockcipher with $2n$-bit keys. Given $f_1 : \{0,1\}^{3n} \to \{0,1\}^n$, the compression function $\phi[f_1] : \{0,1\}^{3n} \to \{0,1\}^{2n}$ is defined by $\phi[f_1](u_2||u_1||u_0) = x||y$, where $x = f_1(u_2||u_1||u_0)$ and $y = u_2x^2 + u_1x + u_0 \in \mathbb{F}_{2^n}$ for u_2, u_1, $u_0 \in \{0,1\}^n$. In the *quadratic blockcipher-based compression function*, f_1 is implemented using a blockcipher $E : \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^n$, $E(k,x) = E_k(x)$ by letting $f_1(u_2||u_1||u_0) = E_{u_2||u_1}(u_0) + u_0$ as described in Figure 2(a). We write $\psi[E]$ for the resulting compression function. Thus $\psi[E] : \{0,1\}^{3n} \to \{0,1\}^n$ and $\psi[E](u_2||u_1||u_0) = x||y$ where

$$\begin{aligned} x &= E_{u_2||u_1}(u_0) + u_0, \\ y &= u_2x^2 + u_1x + u_0. \end{aligned}$$

We can prove that $\psi[E]$ provide similar security as the quadratic mode $\phi[f_1]$ when instantiated with an ideal cipher E, in terms of unforgeability, collision resistance and pseudorandomness. In fact, our results do not actually necessitate an ideal cipher E (which is a set of independent random permutations with one permutation per key) but only an "unpredictable" blockcipher E. For the latter, all that is assumed is that it is difficult for an adversary to fully predict the output of an unqueried value. We call this the *unpredictability* of the blockcipher (which is similar to the unforgeability of a keyed function family, except no keys are involved) and we quantify it by the advantage $\mathbf{Adv}_E^{\mathsf{unp}}(t,q)$ which is the maximum over all adversaries A running in time t and making at most q queries to E of the probability that A can output a tuple $(u_2||u_1, v, w)$ such that $E_{u_2||u_1}(v) = w$ without making queries for $E_{u_2||u_1}(v)$ or $E^{-1}_{u_2||u_1}(w)$.

Implicitly $\mathbf{Adv}_E^{\mathsf{unp}}(t,q)$ depends on a sampling procedure for E. In the ideal cipher model, E is sampled uniformly at random among all n-bit blockciphers with $2n$-bit keys. Here we allow any sampling procedure for E. Note that $\mathbf{Adv}_E^{\mathsf{unp}}(t,q) \leq 1/(2^n - q)$ if E is an ideal cipher, so we can always revert to that bound by assuming an ideal cipher. Our use of unpredictable blockciphers is somewhat similar to that of [7], with the significant difference that the blockciphers of [7] use fixed keys, and that they are sampled by sampling the fixed keys. The unpredictability then corresponds to the unforgeability of a keyed function family (which happens to be a family of permutations).

We show that the collision resistance of $\psi[E]$ can be effectively bounded in terms of $\mathbf{Adv}_E^{\mathsf{unp}}(t,q)$. Let $\mathbf{Adv}_{\psi[E]}^{\mathsf{coll}}(t,q)$ be the maximum probability that an adversary A of running time t with oracle access to E and E^{-1} outputs a collision (M,M') for $\psi[E]$ which it has verified (i.e. has made the queries necessary to compute $\psi[E](M)$ and $\psi[E](M')$). The following theorems are proved in the full version.

Theorem 7. *Let $\psi[E]$ be the quadratic blockcipher-based compression function, where E is sampled from the set of all n-bit blockciphers with $2n$-bit keys according to an arbitrary fixed distribution. Then,*

$$\mathbf{Adv}_{\psi[E]}^{\mathsf{coll}}(t,q) \leq 2q(\log q+3)\mathbf{Adv}_E^{\mathsf{unp}}(t+O(n^2q^4)+\mathsf{Time}_2,q).$$

Furthermore, let $G=G[E,f_2]=f_2\circ MD\,[\psi[E]]$ be a function family where f_2 is chosen from a function family $f:\{0,1\}^\kappa\times\{0,1\}^{2n}\rightarrow\{0,1\}^n$. Then for $q=\lfloor\mu/c\rfloor+2\tilde{q}$,

$$\mathbf{Adv}_G^{\mathsf{mac}}(t,\tilde{q},\mu) \leq \mathbf{Adv}_f^{\mathsf{mac}}(t,\tilde{q})+2q(\log q+3)\mathbf{Adv}_E^{\mathsf{unp}}(t+O(n^2q^4)+\mathsf{Time}_2,q).$$

Remark 3. The term "Time_2", which represents the time necessary to select a root of a quadratic polynomial over $\mathbb{F}_{2^n}$, is mainly kept to facilitate comparison with Theorem 1.

Theorem 8. *Let $E:\{0,1\}^{2n}\times\{0,1\}^n\rightarrow\{0,1\}^n$ be an ideal blockcipher and let $f_2:\{0,1\}^{2n}\rightarrow\{0,1\}^n$ be a random functions. Then, $\mathbf{Adv}_{\psi[E]}^{\mathsf{coll}}(q)\leq\epsilon(q)$ for*

$$\epsilon(q)=\frac{2q(\log q+3)}{2^n-q},$$

and $G[E,f_2]=f_2\circ MD\,[\psi[E]]$ is $(\tilde{q},\epsilon(\bar{q}))$-indistinguishable from a random function $\mathcal{H}:\{0,1\}^*\rightarrow\{0,1\}^n$, where

$$\bar{q}=\mathsf{NQ}(MD[\psi[E]],l_{max})\tilde{q}=\left\lceil\frac{l_{max}+1}{n}+1\right\rceil\tilde{q},$$

for the length in bits l_{max} of the longest query made by a distinguisher.

6 The Quadratic Cascade Compression Function

In this section, we discuss a concrete instantiation of the quadratic compression function (polynomial-based compression function of degree $d=2$) using the cascade of two $2n\rightarrow n$ functions. In the *quadratic cascade compression function*, the compression function f_1 is implemented by the cascade of two compression functions $h_1,h_2:\{0,1\}^{2n}\times\{0,1\}^n\rightarrow\{0,1\}^n$ by letting $f_1(u_2||u_1||u_0)=h_2(h_1(u_2||u_1)||u_0)$ as described in Figure 2(b). We write $\tau[h_1,h_2]$ for the resulting compression function. Thus $\tau[h_1,h_2]:\{0,1\}^{3n}\rightarrow\{0,1\}^n$ and $\tau[h_1,h_2](u_2||u_1||u_0)=x||y$, where

$$x=h_2(h_1(u_2||u_1)||u_0),$$
$$y=u_2x^2+u_1x+u_0.$$

Now we have the following theorems. Due to lack of space, the proof will be given in the full version of this paper. (The proof has similar—but slightly worse—complexity as the security proof for SS-NMAC in [7]. It involves 13 different cases for the forger, 8 of which use MTF games.)

Theorem 9. *Let $\tau = \tau[h_1, h_2]$ be a function family where h_1 and h_2 are independently chosen from a function family h, respectively. Then,*

$$\mathbf{Adv}_{\tau}^{\mathsf{wcr}}(t, q) \leq 26q \log q (\log q + 1)^2 \, \mathbf{Adv}_{h}^{\mathsf{mac}}\left(t + O(n^2 q^4 \log^4 q) + \mathsf{Time}_2, q\right).$$

Furthermore, let $G = G[h_1, h_2, f_2] = f_2 \circ MD[\tau[h_1, h_2]]$ be a function family where f_2 is chosen from a function family f. Then for $q = \lfloor \mu/c \rfloor + 2\tilde{q}$,

$$\mathbf{Adv}_{G}^{\mathsf{mac}}(t, \tilde{q}, \mu) \leq \mathbf{Adv}_{f}^{\mathsf{mac}}(t, \tilde{q}) + 26q \log q (\log q + 1)^2 \, \mathbf{Adv}_{h}^{\mathsf{mac}}\left(t + O(n^2 q^4 \log^4 q) + \mathsf{Time}_2, q\right).$$

Theorem 10. *If $h_1, h_2, f_2 : \{0,1\}^{2n} \rightarrow \{0,1\}^n$ are random functions, then $\mathbf{Adv}_{\tau[h_1,h_2]}^{\mathsf{coll}}(q) \leq \epsilon(q)$ for*

$$\epsilon(q) = \frac{26q \log q (\log q + 1)^2}{2^n},$$

and $G[h_1, h_2, f_2] = f_2 \circ MD[\tau[h_1, h_2]]$ is $(\tilde{q}, \epsilon(\bar{q}))$-indistinguishable from a random function $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^n$, where

$$\bar{q} = \mathsf{NQ}(MD[\tau[h_1, h_2]], l_{max})\tilde{q} = 2\left\lceil \frac{l_{max} + 1}{n} + 1 \right\rceil \tilde{q},$$

for the length in bits l_{max} of the longest query made by a distinguisher.

References

1. An, J.H., Bellare, M.: Constructing VIL-MACs from FIL-MACs: Message authentication under weakened assumptions. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 252–269. Springer, Heidelberg (1999)
2. Bellare, M., Ristenpart, T.: Multi-property-preserving Hash Domain Extension and the EMD Transform. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 299–314. Springer, Heidelberg (2006)
3. Coron, J., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-Damgård revisited: How to construct a hash function. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 430–448. Springer, Heidelberg (2005)
4. Damgård, I.: A design principle for hash functions. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 416–427. Springer, Heidelberg (1990)
5. Dodis, Y., Pietrzak, K., Puniya, P.: A new mode of operation for block ciphers and length-preserving MACs. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 198–219. Springer, Heidelberg (2008)
6. Dodis, Y., Ristenpart, T., Shrimpton, T.: Salvaging Merkle-Damgård for practical applications. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 371–388. Springer, Heidelberg (2009)
7. Dodis, Y., Steinberger, J.: Message authentication codes from unpredictable block ciphers. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 267–285. Springer, Heidelberg (2009)

8. von zur Gathen, J., Panario, D.: Factoring polynomials over finite fields: A survey. J. Symbolic computation 31, 3–17 (2001)
9. von zur Gathen, J., Shoup, V.: Computing Frobenius maps and factoring polynomials. Computational complexity 2, 187–224 (1992)
10. Hirose, S.: Some plausible constructions of double length hash functions. In: Robshaw, M.J.B. (ed.) FSE 2006. LNCS, vol. 4047, pp. 210–225. Springer, Heidelberg (2006)
11. Joux, A.: Multicollisions in iterated hash functions. Application to cascaded constructions. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 306–316. Springer, Heidelberg (2004)
12. Kelsey, J., Kohno, T.: Herding hash functions and the Nostradmus attack. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 183–200. Springer, Heidelberg (2006)
13. Kelsey, J., Schneier, B.: Second preimages on n-bit hash functions for much less than 2^n work. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 474–490. Springer, Heidelberg (2005)
14. Lai, X., Massey, J.: Hash function based on block ciphers. In: Rueppel, R.A. (ed.) EUROCRYPT 1992. LNCS, vol. 658, pp. 55–70. Springer, Heidelberg (1993)
15. Lucks, S.: A failure-freindly design principle for hash functions. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 474–494. Springer, Heidelberg (2005)
16. Lucks, S.: A collision-resistant rate-1 double-block-length hash function. In: Symmetric Cryptography, Dagstuhl Seminar Proceedings 07021 (2007)
17. Merkle, R.: One way hash functions and DES. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 428–446. Springer, Heidelberg (1990)
18. Maurer, U., Renner, R., Holenstein, R.: Indifferentiability, impossibility results on reductions, and apllications to the random oracle methodology. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 21–39. Springer, Heidelberg (2004)
19. Maurer, U., Sjödin, J.: Single-key AIL-MACs from any FIL-MAC. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 472–484. Springer, Heidelberg (2005)
20. Özen, O., Stam, M.: Another glance at double length hashing. In: Parker, M.G. (ed.) Cryptography and Coding 2009. LNCS, vol. 5921, pp. 176–201. Springer, Heidelberg (2009)
21. Rogaway, P., Steinberger, J.: Constructing cryptographic hash functions from fixed-key blockciphers. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 433–450. Springer, Heidelberg (2008)
22. Stam, M.: Beyond uniformity: Security/efficiency tradeoffs for compression functions. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 397–412. Springer, Heidelberg (2008)
23. Stam, M.: Blockcipher based hashing revisited. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 67–83. Springer, Heidelberg (2009)
24. Yasuda, K.: A double-piped mode of operation for MACs, PRFs and PROs: Security beyond the birthday barrier. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 242–259. Springer, Heidelberg (2009)

A The Proof of Lemma 5

Let $\mathcal{G}_0$ and $\mathcal{G}_1$ be games with a single interface, as defined in Figure 6. Assume that a distinguisher $\mathcal{A}$ makes no redundant query. Then whenever $\mathcal{A}$ makes

Games $\mathcal{G}_0$ $\boxed{\textbf{Games } \mathcal{G}_1}$

```
Interface $H(v)$
    $w \leftarrow F[\mathsf{Sample}\text{-}\mathcal{P}](v)$
    $z \leftarrow \mathsf{Sample}\text{-}g(v, w)$
    return $z$

Subroutine $\mathsf{Sample}\text{-}\mathcal{P}(x)$
    return $\mathcal{P}(x)$

Subroutine $\mathsf{Sample}\text{-}g(v, w)$
    $g(v, w) \xleftarrow{\$} \{0,1\}^m$
    if $g(v', w)$ was previously queried for $v' \neq v$ then
        $\mathsf{bad} \leftarrow$ true
        $\boxed{g(v, w) \leftarrow g(v', w)}$
    return $g(v, w)$
```

Fig. 6. Games $\mathcal{G}_0$ and $\mathcal{G}_1$. $\mathcal{G}_1$ includes the boxed statement

a query to $H(\cdot)$ in game $\mathcal{G}_0$, it will receive an independent random value in $\{0,1\}^m$. It means that the interface $H(\cdot)$ faithfully implements a random function $H : \{0,1\}^* \rightarrow \{0,1\}^m$ in game $\mathcal{G}_0$. On the other hand, the interface $H(\cdot)$ in game $\mathcal{G}_1$ faithfully implements $g \circ F[\mathcal{P}] : \{0,1\}^* \rightarrow \{0,1\}^m$ for a random function $g : \{0,1\}^n \rightarrow \{0,1\}^m$ since $\mathsf{Sample}\text{-}g(v, w)$ only depends on the value v. Flag bad sets to true only when $\mathcal{A}$ makes a collision in $F[\mathcal{P}]$. Therefore, for any distinguisher $\mathcal{A}$ that makes at most $\tilde{q}$ queries to $H(\cdot)$, we have

$$\left|\mathbf{Pr}\left[\mathcal{A}^{g \circ F[\mathcal{P}]} = 1\right] - \mathbf{Pr}\left[\mathcal{A}^{H} = 1\right]\right| \leq \mathbf{Pr}\left(\mathcal{A}^{\mathcal{G}_1} \text{ sets } \mathsf{bad}\right) \leq \mathbf{Adv}_F^{\mathsf{coll}}(q),$$

where $q = \mathsf{NQ}(F, l_{max})\tilde{q}$ and l_{max} is the length in bits of the longest query made by a distinguisher.

B Improved Analysis of Lucks' Double-Piped Mode of Operation

We begin with the definition of Lucks' double-piped mode of operation (with a slight modification). First, two $2n + c \rightarrow n$ bit functions f_1 and f_2 are concatenated, yielding the following compression function.

$$\begin{aligned} F : \{0,1\}^{2n+c} &\longrightarrow \{0,1\}^{2n} \\ u &\longmapsto f_1(u)||f_2(u). \end{aligned}$$

The pictorial description of F for $c = n$ is shown in Figure 7. Given $F = F[f_1, f_2]$ and an independent compression function $f_3 : \{0,1\}^{2n+c} \rightarrow \{0,1\}^n$, the Lucks' mode of operation defines a hash function

$$\begin{aligned} G[f_1, f_2, f_3] : \{0,1\}^* &\longrightarrow \{0,1\}^n \\ M &\longmapsto f_3(0^c||v), \end{aligned}$$

where $v = MD[F](M)$.

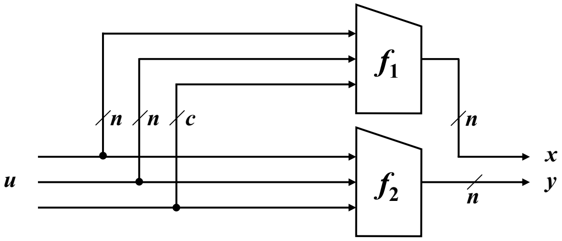


Fig. 7. Lucks' double-piped compression function

Now we prove the weak collision resistance of $F = F[f_1, f_2]$ where f_1 and f_2 are independently chosen from a function family f.

Theorem 11. *Let F be a function family defined by $f : \{0,1\}^\kappa \times \{0,1\}^{2n+c} \to \{0,1\}^n$ as described above. Then,*

$$\mathbf{Adv}_F^{\mathsf{wcr}}(t,q) \leq \sqrt{2}q\mathbf{Adv}_f^{\mathsf{mac}}(t,q)\,.$$

Proof. Let $\mathcal{A}$ be a weak collision-finding adversary such that

$$\mathbf{Adv}_F^{\mathsf{wcr}}(\mathcal{A}) = \mathbf{Adv}_F^{\mathsf{wcr}}(t,q) = \epsilon.$$

We write $u[i]$ for the i-query that $\mathcal{A}$ makes to F and $F(u[i]) = (x[i], y[i])$, where $x[i] = f_1(u[i])$ and $y[i] = f_2(u[i])$. Let

$$\Gamma = \left\{(j,i) \in \{1,\ldots,q\}^2 : j < i \text{ and } x[j] = x[i]\right\},$$

and let $\gamma = |\Gamma|$ be the number of "upper half-collisions". Here we fix an ordering in Γ. By assumption that $\mathcal{A}$ succeeds to find a collision with probability ϵ, at least one of the following two events happens with probability $\geq \epsilon/2$, where θ is a parameter to be optimized later.

Case 1: $\mathcal{A}$ finds a collision and $\gamma > \theta$. For this case, we can construct a forger $\mathcal{B}_1$ for f_1 as follows.

1. $\mathcal{B}_1$ chooses $(j,i) \in \{1,\ldots,q\}^2$ such that $j < i$ uniformly at random.
2. $\mathcal{B}_1$ runs $\mathcal{A}$ as a subroutine and faithfully answers the queries made by $\mathcal{A}$ until the $(i-1)$-th query. Here $\mathcal{B}_1$ simulates f_2 by choosing a key for f_2 uniformly at random.
3. On the i-query $u[i]$, $\mathcal{B}_1$ presents a forgery $(u[i], y[j])$ without making a query to $f_1(\cdot)$.

Since the probability that $(j,i) \in \Gamma$ is $\theta/\binom{q}{2}$, we have

$$\mathbf{Adv}_f^{\mathsf{mac}}(\mathcal{B}_1) \geq \frac{\epsilon\theta}{2\binom{q}{2}} \geq \frac{\epsilon\theta}{q^2}. \tag{9}$$

Case 2: $\mathcal{A}$ finds a collision and $\gamma \leq \theta$. For this case, we can construct a forger $\mathcal{B}_2$ for f_2 as follows.

1. $\mathcal{B}_2$ chooses $s \in \{1, \dots, \theta\}$ uniformly at random.
2. $\mathcal{B}_2$ runs $\mathcal{A}$ as a subroutine: To each query made by $\mathcal{A}$, $\mathcal{B}_2$ faithfully respond by simulating f_1 and making queries to $f_2(\cdot)$.
3. $\mathcal{B}_2$ counts the number of collisions in f_1. At the s-th collision $(j, i) \in \Gamma$, $\mathcal{B}_2$ stops without making a query to $f_2(\cdot)$ and presents a forgery $(u[i], y[j])$.

If there exists a collision $(x[j], y[j]) = (x[i], y[i])$ for $j < i$, then obviously $(j, i) \in \Gamma$. Therefore we have

$$\mathbf{Adv}_f^{\mathsf{mac}}(\mathcal{B}_2) \geq \frac{\epsilon}{2\theta}. \tag{10}$$

From (9) and (10), it follows that

$$\mathbf{Adv}_F^{\mathsf{wcr}}(t, q) \leq \max\left\{\frac{q^2}{\theta}, 2\theta\right\} \mathbf{Adv}_f^{\mathsf{mac}}(t, q).$$

By setting $q^2/\theta = 2\theta$ or $\theta = q/\sqrt{2}$, we obtain

$$\mathbf{Adv}_F^{\mathsf{wcr}}(t, q) \leq \sqrt{2} q \mathbf{Adv}_f^{\mathsf{mac}}(t, q). \qquad \square$$

By Lemma 1 and Theorem 11, we obtain the following theorem.

Theorem 12. *Let $G = G[f_1, f_2, f_3]$ be a function family such that f_1, f_2 and f_3 are independently chosen from a function family f. Then for $q = \lfloor \mu/c \rfloor + 2\tilde{q}$,*

$$\mathbf{Adv}_G^{\mathsf{mac}}(t, \tilde{q}, \mu) \leq \mathbf{Adv}_f^{\mathsf{mac}}(t, \tilde{q}) + \sqrt{2} q \mathbf{Adv}_f^{\mathsf{mac}}(t, q).$$

With a slight modification of the above argument, we can prove that the Lucks' mode of operation *using a single key* also preserves unforgeability up to $O(2^n)$ query complexity, improving the bound $O(2^{5n/6})$ proved by Yasuda [24].

Stam's Collision Resistance Conjecture

John Steinberger[⋆]

Institute of Theoretical Computer Science, Tsinghua University, Beijing
jpsteinb@gmail.com

Abstract. At CRYPTO 2008 Stam [7] made the following conjecture: if an $m + s$-bit to s-bit compression function F makes r calls to a primitive f of n-bit input, then a collision for F can be obtained (with high probability) using $r2^{(nr-m)/(r+1)}$ queries to f. For example, a $2n$-bit to n-bit compression function making two calls to a random function of n-bit input cannot have collision security exceeding $2^{n/3}$. We prove this conjecture up to a constant multiplicative factor and under the condition $m' := (2m - n(r-1))/(r+1) \geq \log_2(17)$. This covers nearly all cases $r = 1$ of the conjecture and the aforementioned example of a $2n$-bit to n-bit compression function making two calls to a primitive of n-bit input.

1 Introduction

A popular paradigm for security proofs in the field of hash function design is to assume that some primitive used by the hash function, such as a blockcipher, is "ideal", namely perfectly random subject to the constraints of the type of primitive concerned, and then to bound the chance of success of some adversary given oracle access to this primitive in terms of the number of queries allowed to the adversary. In this "ideal primitive" model (or IPM, as we will call it) adversaries are usually information-theoretic: their only obstacle to achieving an attack is the randomness of the query responses.

Because the IPM considers information-theoretic adversaries certain limitations naturally arise as to what kind of security can be achieved for a certain functionality using a certain primitive a certain number of times. For example, consider the task of constructing a $2n$-bit to n-bit compression function F using a random n-bit to n-bit permutation f as a primitive. There are 2^{2n} inputs to F but only 2^n inputs to f. Thus each input to f corresponds on average to 2^n inputs to F, so with just two calls to f we can learn to evaluate F on at least $2 \cdot 2^n$ inputs. But this is more than the number of outputs of F, so a collision can be obtained with probability 1 in just two queries. (Note that determining which two f-queries to make is no problem for an information-theoretic adversary, nor is "finding the collision" among the $2 \cdot 2^n$ mapped values.) Thus it is not possible to design a compression function with these parameters that is collision resistant in the IPM.

In the same vein as the above argument, this paper pursues the task of determining the limits of IPM security. Specifically, we tackle the following question: given $m, n, r, s \geq 1$, what is the maximum collision security of a compression function $F : \{0,1\}^{m+s} \to$

[⋆] Supported by the National Natural Science Foundation of China Grant 60553001 and by the National Basic Research Program of China Grant 2007CB807900, 2007CB807901.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 597–615, 2010.

$\{0,1\}^s$ that makes r calls to an ideal primitive f of domain $\{0,1\}^n$? (The range of f is not specified because it turns out to be immaterial[1].) Here "collision security" means the largest number of f-queries the best information-theoretic adversary can ask before achieving probability $\frac{1}{2}$ of obtaining a collision.

Since it costs at most r queries to evaluate any point in the domain, a birthday attack implies that collision security cannot exceed $q = 2\sqrt{2}r2^{s/2}$ queries (cf. Proposition 1 Section 5). However other attacks may be more constraining than birthday attacks. In particular Stam [7] conjectured[2] that

$$q = r\lceil 2^{(nr-m)/(r+1)}\rceil + 1 \tag{1}$$

queries should always suffice for finding a collision with probability at least $\frac{1}{2}$. This bound becomes more constraining than the birthday attack when $s/2 > (nr-m)/(r+1)$. This occurs for example when $(m,n,r,s) = (n,n,2,n)$, the case of a $2n$-bit to n-bit compression function making two calls to a primitive of n-bit input, for which Stam's bound forecasts a maximum collision resistance of $2^{n/3}$ whereas a birthday attack caps the collision resistance at $2^{n/2}$. It is noteworthy that Stam's bound is independent of s. We explain later the intuition behind the exponent $(nr-m)/(r+1)$.

Stam's conjecture is particularly appealing because it apparently constitutes the *optimal* upper bound on collision resistance for all cases for which it beats the birthday bound, while the birthday bound can apparently be achieved in all other cases. In other words, to the best of current understanding, it seems that the maximum collision resistance of a compression function $F : \{0,1\}^{m+s} \to \{0,1\}^s$ making r calls to a random function f of n-bit input in fact equals

$$\min(r2^{s/2}, r\lceil 2^{(nr-m)/(r+1)}\rceil)$$

up to possible lower order terms. This thesis is supported by a number of constructions [4,7,6].

So far, however, Stam's bound has not been proved for any case of interest (cases of "non-interest" being those for which $s/2 \leq (nr-m)/(r+1)$ or $nr - m \leq 0$; see Section 2). Here we try to remedy this situation. We show there is an absolute constant $C \geq 1$ such that if

$$m' := (2m - n(r-1))/(r+1) \geq 4.09 \tag{2}$$

then

$$q = Cr\lceil 2^{(nr-m)/(r+1)}\rceil \tag{3}$$

queries suffice in order to obtain a collision with probability at least $\frac{1}{2}$ (see Corollary 3 in Section 5 for a tighter statement). In other words, we prove Stam's conjecture up

[1] Immaterial to proving our upper bound; better upper bounds on security should be provable if f has sufficiently small range, see comments by Stam [7].

[2] Stam's wording is not quite as precise, as he omits the ceiling brackets, the '+1' term, and the fact that a collision can only be found with "sufficient" probability, but it is easy to see these changes are necessary for correctness of the conjecture.

to a constant multiplicative factor as long as (2) holds. To get a better handle on the restriction (2) note that it reduces to $m = m' \geq 4.09$ for $r = 1$ and to $\frac{2}{3}m - \frac{1}{3}n \geq 4.09$ for $r = 2$. For $r = 2$ setting $m = n$ reduces the condition to $n \geq 12.27$. Our result is partly based on the observation that Stam's conjecture reduces to the case $r = 1$ when $m' \geq 1$; see Section 4 for details.

We emphasize that our result holds for arbitrary primitives f. That is, if f has range $\{0,1\}^b$, then f may be sampled with any distribution from all functions of domain $\{0,1\}^n$ and range $\{0,1\}^b$. Thus our result covers not only perfectly random primitives but also random permutations and ideal ciphers[3]. Moreover, in the case where $r > 1$, F may call r distinct primitives (of potentially different distributions) rather than the same primitive r times.

PROBLEM HISTORY. The first authors to consider the limits of IPM security in the information-theoretic setting were Black, Cochran and Shrimpton [2], who showed that any iterated hash function using a $2n$-bit to n-bit compression function F making a single call to one of r different ideal n-bit permutations would have (unacceptably low) collision security of $r(n + \log(n))$ queries. Rogaway and Steinberger [5] generalized this result by showing that collisions could be found with probability 1 in $1.89s2^{n(1-\alpha)}$ queries for any permutation-based hash function of rate α and output length s (the rate being the number of n-bit message blocks processed per application of the n-bit primitive). The latter result is somewhat noteworthy because it does not make any assumption on the structure (iterated, etc) of the hash function, and does not even restrict the number of different independent permutations used by the hash function—moreover the result more generally holds (with the same proof) if the permutations are replaced by any primitives of domain $\{0,1\}^n$.

Rogaway and Steinberger also considered the IPM security of compression functions instantiated from n-bit random permutations (like above, their proofs in fact apply for any primitive of domain $\{0,1\}^n$). They showed that with $r(2^{n-m/r} + 1)$ queries an adversary could find a collision with probability 1 for any compression function $F : \{0,1\}^{m+s} \to \{0,1\}^s$ that makes r calls to an n-bit permutation. They also noted that, for compression functions F meeting a certain reasonable-looking heuristic assumption dubbed "collision uniformity", $r2^{n-(m+\frac{s}{2})/r}$ queries suffice for finding a collision with probability $\frac{1}{2}$. Stam [7] subsequently found examples of non-collision-uniform compression functions having higher collision security than $r2^{n-(m+\frac{s}{2})/r}$ and posited that collision security could not exceed $r\lceil 2^{(nr-m)/(r+1)} \rceil$ independently of any heuristic assumption. This is the bound we discuss in this paper.

ON 'OPTIMALITY'. Security upper bounds are useful as benchmarks for designers. In this area, though, the situation isn't so simple: when $2^{(nr-m)/(r+1)} < 2^{s/2}$ (namely when Stam's bound becomes more constraining than the birthday attack upper bound) then the only constructions which can achieve the best-possible collision security are non-uniform constructions, implying a questionable non-random behavior. The "better"

[3] A blockcipher of k-bit key and l-bit word is modeled as a primitive of $l + k$-bit input; note the absence of inverse queries does typically not affect the task of proving *upper bounds* on security, though if desired one may even emulate bidirectional blockcipher queries with an extra bit of input specifying forward or inverse queries.

construction may then be a uniform construction of lower collision security. On the other hand, some non-collision-uniform constructions have been proposed, for example the JHash compression function [9]. The non-uniformity of these compression functions is usually belied by the fact that many collisions are obtained whenever a single collision is obtained. (Uniformity is explained in more detail in Section 3.)

Regarding this issue, Stam has suggested that when $(nr - m)/(r + 1) < s/2$ one should consider lowering the state size s until $s/2 = (nr - m)/(r + 1)$, so that one may (at least theoretically) achieve the optimum collision resistance with a uniform construction, as opposed to achieving the same collision resistance with a non-uniform construction or a lower collision resistance with a uniform construction. This makes sense from the point of view of compression function design, though designers should bear in mind the hash function obtained by iterating the compression function will probably be weakened by lowering the state size at the same time the compression function is strengthened (the collision resistance of the hash function being typically higher than that of the compression function); for example, while a uniform $2n$-bit to n-bit compression function F_1 making two calls to an n-bit input random function f may have only $2^{n/4}$ collision security against $2^{n/3}$ collision security for a uniform $\frac{5}{3}n$-bit to $\frac{2}{3}n$-bit compression function F_2 also making two calls to a random n-bit input function f, the iteration of F_1 may have $2^{n/2}$ collision security[4] whereas the iteration of F_2 will be "stuck" at $2^{n/3}$ collision security.

Finally, the usual caveats regarding the ideal primitive model apply to this paper: as the IPM considers information-theoretic adversaries, our results do not imply security upper bounds with respect to real-world, computationally bounded adversaries.

ORGANIZATION. In the next section we give some background of results of Rogaway and Steinberger. Section 3 is an optional section giving some intuition about Stam's conjecture for $r > 1$. Section 4 examines the case $r = 1$ and how certain cases of Stam's conjecture with $r > 1$ reduce to the case $r = 1$. Section 5 contains the main proof and the formal statement of our result, which is summarized by Corollary 3. Appendix A discusses an alternate approach to our main result for the special case of random primitives.

2 Basic Results

We first formalize the notion of a compression function F making r calls to a primitive f. In fact we allow F to call potentially distinct primitives $f_1, \ldots, f_r$ in *fixed order mode*, meaning f_i is called before f_j for $i < j$.

Let $f_1, \ldots, f_r$ be (not necessarily distinct) functions of domain $\{0,1\}^n$ and range $\{0,1\}^b$, where b is arbitrary. The compression function $F : \{0,1\}^{m+s} \to \{0,1\}^s$ is defined by r functions $g_1, \ldots, g_r$ where $g_i : \{0,1\}^{m+s} \times \{0,1\}^{b(i-1)} \to \{0,1\}^n$ and a function $h : \{0,1\}^{m+s} \times \{0,1\}^{br} \to \{0,1\}^s$. We then define $F(v) = h(v, y_1, \ldots, y_r)$ where $y_j = f_j(g_j(v, y_1, \ldots, y_{j-1}))$ for $j = 1 \ldots r$. We call the values $y_1, \ldots, y_r$ "intermediate chaining variables".

[4] This is indeed conjectured for a number of two-call constructions, such as the Grøstl compression function [3].

We say an adversary A with oracle access to $f_1, \ldots, f_r$ "knows the first k chaining variables" for some input $v \in \{0,1\}^{m+s}$ when A has made the queries $f_1(g_1(v)) = y_1$, $f_2(g_2(v, y_1)) = y_2, \ldots, f_k(g_k(v, y_1, \ldots, y_{k-1})) = y_k$, where $0 \leq k \leq r$. We start with the following basic observation of Rogaway and Steinberger [5]:

Lemma 1. *Let $F : \{0,1\}^{m+s} \to \{0,1\}^s$ be a compression function calling primitives $f_1, \ldots, f_r : \{0,1\}^n \to \{0,1\}^b$ in fixed-order mode and let $0 \leq k \leq r$. Then with at most q queries to each of the functions $f_1, \ldots, f_k$ an adversary can learn the first k chaining variables for at least*

$$2^{m+s} \left(\frac{q}{2^n}\right)^k$$

inputs.

Proof. We proceed by induction on k, with the result obviously holding for $k = 0$. Now assume $1 \leq k \leq r$. By the induction hypothesis, the adversary can make q queries to each of $f_1, \ldots, f_{k-1}$ so that it knows the first $k-1$ chaining variables for at least

$$2^{m+s} \left(\frac{q}{2^n}\right)^{k-1}$$

inputs. Let X be the set of these inputs, and for each $z \in \{0,1\}^n$ let X_z be the set of inputs $v \in X$ such that $g_k(v, y_1, \ldots, y_{k-1}) = z$ where $y_1, \ldots, y_{k-1}$ are the first $k-1$ chaining variables for v. Because $\{X_z : z \in \{0,1\}^n\}$ are disjoint and have union X there exist distinct values $z_1, \ldots, z_q \in \{0,1\}^n$ such that $\sum_{i=1}^q |X_{z_i}| \geq q|X|/2^n$. By querying $f_k(z_1), \ldots, f_k(z_q)$ the adversary thus learns the first k intermediate variables for at least

$$q|X|/2^n \geq 2^{m+s} \left(\frac{q}{2^n}\right)^k$$

inputs. □

Rogaway and Steinberger originally stated this observation for primitives $f_1, \ldots, f_r : \{0,1\}^n \to \{0,1\}^n$, but the output length of the f_i's does not in fact play any role. Stam [7] subsequently generalized Lemma 1 to the case of compressing primitives $f_i : \{0,1\}^{n+c} \to \{0,1\}^n$, but this generalization is equivalent to Lemma 1 for the same reason (namely it can be obtained by substituting $n+c$ for n and n for b, the latter with no effect).

As a direct corollary of Lemma 1, we have the following:

Corollary 1. *Let $F : \{0,1\}^{m+s} \to \{0,1\}^s$ be a compression function calling primitives $f_1, \ldots, f_r : \{0,1\}^n \to \{0,1\}^b$ in fixed-order mode. Then with q queries to each f_i, an adversary can learn to evaluate F on at least*

$$2^{m+s} \left(\frac{q}{2^n}\right)^r$$

inputs.

In particular, if

$$2^{m+s} \left(\frac{q}{2^n}\right)^r > 2^s$$

then an adversary can obtain a collision for F with probability 1 in rq queries. Solving this inequality for q gives

$$q > 2^{n-m/r}$$

so that

$$r(\lfloor 2^{n-m/r} \rfloor + 1)$$

queries suffice to find a collision with probability 1 (when $n - m/r = 0$ one can improve this bound to $r + 1$ queries). This proves Stam's conjecture for the case $nr - m \leq 0$. (In fact (1) is one more query than needed when $nr - m < 0$.)

3 Intuition for Stam's Bound: The Case $r > 1$

In this section we explain where Stam's bound "comes from". We assume $r > 1$; the case $r = 1$, which has certain peculiarities, is discussed in the next section. Our account of the intuition behind the conjecture gives a different viewpoint than Stam's own, so readers will find an additional perspective by consulting [7]. The rest of the paper does not rely on this section's discussion.

We keep the definitions of F, $f_1, \ldots, f_r$ as in Section 2. Let

$$\mathsf{Yield}(q) = 2^{m+s} \left(\frac{q}{2^n} \right)^r .$$

Thus $\mathsf{Yield}(q)$ is a lower bound for the number of F-inputs an adversary can learn to evaluate with q queries to each primitive f_i (Corollary 1). However, $\mathsf{Yield}(q)$ may badly underestimate this number of inputs. For example an adversary can always learn to evaluate at least q inputs in q queries to each of the f_i's, whereas $\mathsf{Yield}(q)$ goes to zero for large r as long as (say) $q < 2^{n-1}$. A better (and in fact fairly accurate) lower bound is

$$\mathsf{BYield}(q) = \max(q, \mathsf{Yield}(q))$$

where 'B' is for 'better'. Since

$$\mathsf{Yield}(q) \geq q \iff 2^{m+s} \left(\frac{q}{2^n} \right)^r \geq q \iff q \geq 2^{(nr-m-s)/(r-1)}$$

(where we use $r > 1$) we have more exactly that

$$\mathsf{BYield}(q) = \begin{cases} q & \text{if } q \leq 2^{(nr-m-s)/(r-1)}, \\ \mathsf{Yield}(q) & \text{if } q \geq 2^{(nr-m-s)/(r-1)}. \end{cases}$$

Notice[5] that as long as $q < 2^{(nr-m-s)/(r-1)}$ one may increase m or s without affecting $\mathsf{BYield}(q)$, whereas if $q \geq 2^{(nr-m-s)/(r-1)}$ increasing 2^{m+s} by a factor c increases

[5] It is also instructive to note that the threshold $q = 2^{(nr-m-s)/(r-1)}$ occurs when the adversary of Lemma 1 learns on average the value of exactly one input with each query it makes to f_r. Indeed,

$$2^{m+s} \left(\frac{q}{2^n} \right)^r = q \iff 2^{m+s} \left(\frac{q}{2^n} \right)^{r-1} = 2^n$$

meaning that with $q = 2^{(nr-m-s)/(r-1)}$ queries to $f_1, \ldots, f_{r-1}$ the adversary will have 2^n "surviving inputs" for which it knows the first $r-1$ intermediate chaining values, or on average one input for each point in the domain of f_r.

$\mathsf{BYield}(q)$ by that much; for example increasing s by 1, which doubles the size of the range, also doubles the size of $\mathsf{BYield}(q)$.

Empirically, one might estimate that the chance of finding a collision for a given value of q is lower bounded by

$$\mathsf{BYield}^2(q)/2^s$$

since a birthday attack which learns t outputs in a range of size 2^s has chance approximately $t^2/2^s$ of yielding a collision. This is correct when $\mathsf{BYield}(q) = q$, since then the adversary can independently sample each input point for which it chooses to learn the output, but when $\mathsf{BYield}(q) > q$ the inputs for which the adversary learns the output are not independently sampled, and, hence, it is not clear the attack works (indeed it is in fact easy to construct an artificial compression function F that will fool the deterministic adversary of Lemma 1 in this regard). Roughly speaking, Rogaway and Steinberger [5] say that a compression function F is *collision uniform* if learning to evaluate F on any t inputs gives chance $\approx t^2/2^s$ of obtaining a collision. Since a random F has this property, they argue that so should most cryptographically good constructions (i.e. constructions of interest). So far this thesis seems to bear out for all real-world constructions with $r > 1$. The $1.5n$-bit to n-bit JHash compression function (Fig. 1) is a nice example of a non-collision-uniform compression function with $r = 1$: a single query to the underlying permutation already allows the evaluation of $t = 2^{n/2}$ inputs, but one must actually make $q = 2^{n/4}$ queries on average to the permutation before finding a collision (at which point $2^{n/2}$ different collisions are found at once). One can also note the JHash compression function is quite "non-random", as $2^{n/2}$ input-output pairs can be deduced from any single input-output pair.

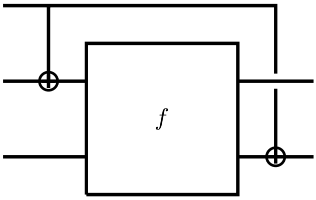


Fig. 1. The JH compression function from $\{0,1\}^{1.5n}$ to $\{0,1\}^n$. All wires carry $n/2$-bit values.

In any case, let us momentarily (and heuristically) assume that adversaries have chance $\mathsf{BYield}^2(q)/2^s$ of obtaining a collision in q queries. If so, the collision resistance of F will be (r times) the least q such that $\mathsf{BYield}(q) = 2^{s/2}$. If $2^{s/2} \le 2^{(nr-m-s)/(r-1)}$ this is $2^{s/2}$, otherwise it is the solution to

$$2^{m+s}\left(\frac{q}{2^n}\right)^r = 2^{s/2}$$

which is $q = 2^{(nr-m-\frac{s}{2})/r}$. Thus, noting $\mathsf{HeuristicSec}(m,n,r,s)$ this "heuristic maximum collision security", we have

$$\mathsf{HeuristicSec}(m,n,r,s) = \begin{cases} r2^{s/2} & \text{if } 2^{s/2} \le 2^{(nr-m-s)/(r-1)}, \\ r2^{(nr-m-\frac{s}{2})/r} & \text{if } 2^{s/2} \ge 2^{(nr-m-s)/(r-1)}. \end{cases}$$

Now consider m, n, r as fixed and s as variable. Note that for sufficiently large s we will be in the second case, $2^{s/2} \geq 2^{(nr-m-s)/(r-1)}$. Also note that if we increase s while in the second case, HeuristicSec *decreases*[6]. However, as noted by Stam, increasing the state size s should never decrease the best-possible collision security of a compression function, as additional input bits can always be forwarded to the output as the identity without affecting collision security. This shows that HeuristicSec is *provably not* the correct maximum collision security for the range $2^{s/2} \geq 2^{(nr-m-s)/(r-1)}$.

This leaves us with the question of determining the "real" collision security when $2^{s/2} \geq 2^{(nr-m-s)/(r-1)}$. Still thinking of m, n, r as fixed and s as variable, Stam conjectured that as s increases collision security simply "tops off" when $2^{s/2}$ reaches $2^{(nr-m-s)/(r-1)}$ and remains constant afterwards. We have $2^{(nr-m-s)/(r-1)} = 2^{s/2}$ when $s = s_0 = 2(nr-m)/(r+1)$, meaning that collision security can never exceed $r2^{(nr-m)/(r+1)}$ according to this conjecture (or more precisely, since q must be kept integer, that collision security can never exceed $r\lceil 2^{(nr-m)/(r+1)}\rceil$). Succinctly put, while the heuristic attack gives an incorrect bound, it still manages to "freeze" collision security at the point where the attack comes into effect.

Summarizing, Stam's conjecture for $r > 1$ stipulates the "true maximum collision security" $\mathsf{TrueSec}(m, n, r, s)$ is

$$\begin{aligned}\mathsf{TrueSec}(m, n, r, s) &= \begin{cases} r2^{s/2} & \text{if } 2^{s/2} \leq 2^{(nr-m-s)/(r-1)} \\ r\lceil 2^{(nr-m)/(r+1)}\rceil & \text{if } 2^{s/2} \geq 2^{(nr-m-s)/(r-1)} \end{cases} \\ &= \min(r2^{s/2}, r\lceil 2^{(nr-m)/(r+1)}\rceil)\end{aligned}$$

up to some small multiplicative constant. Since $r2^{s/2}$ queries obviously do suffice for finding a collision with probability $\frac{1}{2}$ (up to said small multiplicative constant), the problem reduces to showing that $r\lceil 2^{(nr-m)/(r+1)}\rceil$ queries also always suffice.

4 Intuition for $r = 1$ and Reduction to $r = 1$

For $r = 1$ the conjectured maximum collision security is again

$$\min(r2^{s/2}, r\lceil 2^{(nr-m)/(r+1)}\rceil) = \min(2^{s/2}, \lceil 2^{(n-m)/2}\rceil)$$

but a separate explanation is required. Note that when $r = 1$ an adversary can learn to evaluate F on at least $2^{m+s-n}q$ inputs in $q \leq 2^n$ queries to the (unique) primitive f_1. If $m \geq n$ this gives a 2-query attack, so we may assume $m \leq n$. If $n \geq m + s$ then $2^{(n-m)/2} \geq 2^{s/2}$ is more than the cost of a birthday attack, so we may also assume $n \leq m + s$.

We now argue the bound of $2^{(n-m)/2}$ queries "by example" for the case $m \leq n \leq m + s$ by showing a construction collision secure up to that many queries. As each input to f_1 corresponds on average to 2^{m+s-n} inputs from the domain $\{0,1\}^{m+s}$, it is natural to write the domain as $\{0,1\}^{m+s-n} \times \{0,1\}^n$, and to have $g_1(x||y) = y$ for

[6] This can be seen as a consequence of the fact that $\mathsf{BYield}(q)$ is proportional to 2^s when $q \geq 2^{(nr-m-s)/(r-1)}$, and that the chance of obtaining a collision is estimated as $\mathsf{BYield}^2(q)/2^s$, so that increasing s actually increases this ratio.

any $x \in \{0,1\}^{m+s-n}$ and $y \in \{0,1\}^n$ (this at least "balances" g_1 across the domain). Since we do not want the adversary to obtain a collision from a single query $f_1(y)$, we "reserve" $m+s-n$ output bits for the portion of the domain which does not affect y; namely we set $F(x||y) = x||z$ where z is the truncation to $s-(m+s-n) = n-m$ bits of $f_1(y)$, where we can assume f_1 has output length $b \geq n-m$. To find a collision the adversary only needs to find a collision in the last $n-m$ bits of output (and can then adjust the first $m+s-n$ bits as it wants), leading to collision resistance of $2^{(n-m)/2}$.

Crucially to the results of this paper, certain cases $r > 1$ of the conjecture reduce to the case $r = 1$. Assume $r > 1$. By Lemma 1, an adversary making $q = 2^{(nr-m)/(r+1)}$ queries to each $f_1, \ldots, f_{r-1}$ can learn the first $r-1$ chaining variables for at least

$$\begin{aligned} 2^{m+s}\left(\frac{q}{2^n}\right)^{r-1} &= 2^{m+s}(2^{(nr-m)/(r+1)-n})^{r-1} \\ &= 2^{m+s}(2^{-(n+m)(r-1)/(r+1)}) \\ &= 2^{s+(2m-n(r-1))/(r+1)} \end{aligned}$$

inputs to F. Let A be the set of these inputs. Consider the compression function $F' : A \to \{0,1\}^s$ defined by $F'(v) = F(v)$. Let $m' = (2m - n(r-1))/(r+1)$. If $m' \geq 1$ then we may view F' as a compression function from $\{0,1\}^{m'+s}$ bits to $\{0,1\}^s$ making a single call to a primitive of n-bit input, namely f_r (when m' is non-integral we simply mean that F' has domain of size at least $2^{m'+s}$). According to the case $r = 1$ of Stam's conjecture, $2^{(n-m')/2}$ queries to f_r should suffice for finding a collision in F'. However,

$$2^{(n-m')/2} = 2^{(n-\frac{2m-n(r-1)}{r+1})/2} = 2^{(nr-m)/(r+1)} = q,$$

the number of queries allotted to $f_1, \ldots, f_{r-1}$. Thus if Stam's conjecture holds for $r = 1$ and for non-integral $m \geq 1$ (to allow non-integral m') then it more generally holds whenever $(2m - n(r-1))/(r+1) \geq 1$. We make this idea more formal in the next section.

5 Main Result

We first prove Stam's conjecture for $r = 1$ and $m \geq \log_2(17) \approx 4.09$. The more general result will follow as a corollary via the reduction outlined at the end of the previous section.

Clearly the fact that the compression function F manipulates bit strings is unimportant: the determining factors are the size of the domain, the size of the range, and the size of f's domain. We let the size of F's domain and range be MS and S, respectively, where S is a positive integer and $M \geq 2$. If MS is non-integral then our meaning is that F has domain of size *at least* MS (so $\lceil MS \rceil$ or more). The size of f's domain will be N. Thus under our original notation, $M = 2^m$, $S = 2^s$ and $N = 2^n$. For $r = 1$, the object is to show that $\approx 2^{(n-m)/2} = \sqrt{N/M}$ queries to f suffice for finding a collision in F.

Our collision attack ultimately reduces to a birthday attack. To make fully precise what we mean by a "birthday attack" let $B : D_B \to R_B$ be any fixed function of finite domain D_B and finite range R_B. Then performing a q-query birthday attack on B

means evaluating B at q points of D_B sampled uniformly without replacement, halting when a collision is found. We use the following proposition due to Wiener [8] lower bounding the probability of success of a birthday attack:

Proposition 1. (cf. [8] Theorem 7) *Let $B : D_B \to R_B$ such that D_B, R_B are finite and $D_B \geq 2R_B$. Then a q-query birthday attack on B has chance at least* $1 - 3e^{-2} > 0.5$ *of success when $q \geq 2\sqrt{2R_B} + 1$.*

We can now state and prove our main technical result:

Theorem 1. *Let S, N be positive integers and let $M \geq 17$ be a real number such that $N/M \geq 128$. Let F be a compression function of domain of size at least MS and range of size S making a single call to a primitive f of domain of size N. Then a collision can be found for F with probability at least* 0.5 *in $q = \lceil 4\sqrt{8N/M}\rceil$ queries to f.*

Proof. Let $w = MS/2N$ and let $b = \lceil 4S/w \rceil = \lceil 8N/M \rceil$.

Let D_F, R_F denote the domain and range of F and let D_f denote the domain of f. For each $x \in D_f$ let $T_x = \{y \in D_F : g_1(y) = x\}$ (namely T_x is the set of F-inputs that can be evaluated once f is queried at x). Let $W = \{x \in D_f : |T_x| \geq w\}$. Note the adversary can compute W.

For each $x \in W$ the adversary divides T_x into sets $T_x^1, \ldots, T_x^{j_x}$ such that each $w \leq |T_x^i| < 2w$ for $i = 1 \ldots j_x$. Let U be the set of all these sets, namely $U : \{T_x^i : x \in W, 1 \leq i \leq j_x\}$. The adversary's attack will consist in repeatedly choosing without replacement a random element T_x^i from U uniformly among the elements of U that have not yet been chosen and querying f at x if f has not yet been queried at that point, until either q queries have been made or until no elements are left in U.

We lower bound the adversary's chance of finding a collision with this attack. In fact, we will only give the adversary credit if it finds a collision for inputs that belong to sets that it chose from U, so we more precisely lower bound the probability of the latter event happening.

Let $U_1 = \{T_x^i \in U : |F(T_x^i)| = |T_x^i|\}$ and let $U_2 = U \backslash U_1 = \{T_x^i \in U : |F(T_x^i)| < |T_x^i|\}$. Thus U is the disjoint union of U_1 and U_2. For $T_x^i \in U_1$ consider the event $A_{T_x^i}$ that b random elements of R_F chosen uniformly with replacement do not intersect $F(T_x^i)$. Since $|F(T_x^i)| = |T_x^i| \geq w$ and $|R_F| = S$, we have

$$\Pr[A_{T_x^i}] \leq \left(1 - \frac{w}{S}\right)^b \leq e^{-4} \leq 0.02.$$

Thus there exists some set of b values $\{r_1, \ldots, r_b\} \subseteq R_F$ such that at least 0.98 of the sets in U_1 contain one of the values $r_1, \ldots, r_b$.

Let $D'_F = \bigcup_{x \in W} T_x$. Since $\sum_{x \notin W} |T_x| \leq Nw$ we have $|D'_F| \geq MS - Nw$. Since each element of U is a set of size at most $2w$ and since $D'_F = \bigcup_{T_x^i \in U} T_x^i$, we have

$$|U| \geq \frac{|D'_F|}{2w} \geq \frac{MS - Nw}{2w} = \frac{1}{2}\left(\frac{MS}{w} - N\right)$$

and so $0.98|U| \geq 2b$, since

$$\begin{aligned} 0.98|U| \geq 2b \quad &\Longleftarrow \quad \frac{0.98}{2}\left(\frac{MS}{w} - N\right) \geq 4S/w + 2 \\ &\Longleftrightarrow \quad 0.49N \geq 8N/M + 2 \\ &\Longleftrightarrow \quad 0.49M \geq 8 + 2M/N \\ &\Longleftrightarrow \quad M(0.98 - \frac{4}{N}) \geq 16 \\ &\Longleftarrow \quad M \geq 17 \end{aligned}$$

using $N \geq 128M \geq 128 \cdot 17$ for the last implication.

We say that a set T_x^i chosen by the adversary during its attack (as described above) is "lost" if $T_x^i \in U_1$ and $T_x^i \cap \{r_1, \ldots, r_b\} = \emptyset$. Since $|U| \geq 2b$ and $q \leq 4\sqrt{b} + 1$, any set chosen by the adversary has probability at most

$$\begin{aligned} \frac{0.02|U|}{|U| - q} &\leq \frac{0.02(2b)}{2b - 4\sqrt{b} - 1} \\ &= \frac{0.04}{2 - 4/\sqrt{b} - 1/b} \\ &= \frac{0.04}{2 - 4/32 - 1/1024} \\ &\leq 0.0214 \end{aligned}$$

of being lost independently of the result of previous choices, using $b \geq 8N/M \geq 1024$. By a multiplicative Chernoff bound, the probability that total number of non-lost sets is less than $0.8(1 - 0.0214)q = 0.8 \cdot 0.9786q = 0.78288q$ is therefore at most

$$e^{-\frac{0.9786q0.2^2}{2}} \leq e^{-2.505}$$

using $q \geq 4\sqrt{8N/M} \geq 128$. Thus with chance at least $1 - e^{-2.505} \geq 0.918$, the adversary chooses at least $0.78288q \geq 3\sqrt{b}$ non-lost sets.

The theorem follows by ascribing to each non-lost element of U_1 an element of $\{r_1, \ldots, r_b\}$ that it contains and to each element of U_2 an arbitrary element of $\{r_1, \ldots, r_b\}$, and noting that the adversary wins if it ever chooses two (non-lost) elements of U that are ascribed the same element of $\{r_1, \ldots, r_b\}$. (Indeed, if the adversary ever chooses an element of U_2, it finds a collision automatically.) Thus the adversary's attack becomes a birthday attack on a function of domain at least $0.98|U| \geq 2b$ and range b, in which the adversary queries at least $3\sqrt{b} \geq 2\sqrt{2b} + 1$ independent domain points of the function with probability at least 0.918. By Proposition 1 the latter number of queries is sufficient to find a collision with probability at least $1 - 3e^{-2} \geq 0.5/0.918$, thus concluding the proof. □

Corollary 2. *Let S, N be positive integers and let $M \geq 17$. Let F be a compression function of domain of size at least MS and range of size S making a single call to a primitive f of domain of size N. Then a collision can be found for F with probability at least* 0.5 *in*

$$q = \begin{cases} 2175 & \textit{if } N/17 < 128 \\ 128 & \textit{if } N/17 \geq 128 \textit{ and } N/M < 128 \\ \lceil 4\sqrt{8N/M} \rceil & \textit{if } N/M \geq 128 \end{cases}$$

queries to f.

Proof. The last case is Theorem 1 and the first case is obvious since $N < 17 \cdot 128 = 2176$ when $N/17 < 128$, and f has domain of size N. For the second case, it suffices to observe that we can apply Theorem 1 to a restricted version F' of F, where F' is the restriction of F to a domain $D'_{F'} \subseteq D_F$, $|D'_{F'}| = M'S$ where $M' = N/128 \geq 17$. In the latter case, the cost of the Theorem 1 attack on F' is $q = 4\lceil \sqrt{8N/M'} \rceil = 128$. □

The next corollary is the paper's main result:

Corollary 3. *Let $F : \{0,1\}^{m+s} \rightarrow \{0,1\}^s$ be a compression function calling primitives $f_1, \ldots, f_r : \{0,1\}^n \rightarrow \{0,1\}^b$ in fixed-order mode. Then if $m' = (2m - n(r-1))/(r+1) \geq \log_2(17)$, an adversary making*

$$q = (r-1)\lceil 2^{(nr-m)/(r+1)} \rceil + \begin{cases} 2175 & \textit{if } 2^n/17 < 128 \\ 128 & \textit{if } 2^n/17 \geq 128 \textit{ and } n - m' < 7 \\ \lceil 8\sqrt{2} \cdot 2^{(nr-m)/(r+1)} \rceil & \textit{if } n - m' \geq 7 \end{cases}$$

queries to the f_i's can find a collision for F with probability > 0.5.

Proof. As shown at the end of section 4, an adversary making $q_0 = \lceil 2^{(nr-m)/(r+1)} \rceil$ queries to each of the functions $f_1, \ldots, f_{r-1}$ can learn the intermediate chaining values $y_1, \ldots, y_{r-1}$ for at least $2^{s+m'}$ inputs. We then consider the restriction F' of F to those inputs as a single-call compression function. F' has a domain of size MS and a range of size S where $S = 2^s$, $M = 2^{m'} \geq 17$, and uses a primitive of domain $N = 2^n$. The result then follows from Corollary 2 by noting that $\sqrt{N/M} = 2^{(n-m')/2} = 2^{(nr-m)/(r+1)}$. □

Acknowledgements

The author would like to thank the referees for their careful read and Greg Kuperberg for helpful discussions.

References

1. Bellare, M., Kohno, T.: Hash function imbalance and its impact on birthday attacks. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 401–418. Springer, Heidelberg (2004)
2. Black, J., Cochran, M., Shrimpton, T.: On the Impossibility of Highly-Efficient Blockcipher-Based Hash Functions. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 526–541. Springer, Heidelberg (2005)
3. Gauravaram, P., Knudsen, L., Matusiewicz, K., Mendel, F., Rechberger, C., Schläffer, M., Thomsen, S.: Grøstl, a SHA-3 candidate, NIST SHA-3 competition submission (October 2008)

4. Rogaway, P., Steinberger, J.: Constructing cryptographic hash functions from fixed-key blockciphers. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 433–450. Springer, Heidelberg (2008)
5. Rogaway, P., Steinberger, J.: Security/Efficiency Tradeoffs for Permutation-Based Hashing. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 220–236. Springer, Heidelberg (2008)
6. Shrimpton, T., Stam, M.: Building a Collision-Resistant Compression Function from Non-Compressing Primitives. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 643–654. Springer, Heidelberg (2008), Cryptology ePrint Archive: Report 2007/409
7. Stam, M.: Beyond uniformity: Better Security/Efficiency Tradeoffs for Compression Functions. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 397–412. Springer, Heidelberg (2008)
8. Wiener, M.: Bounds on birthday attack times. Cryptology ePrint archive (2005)
9. Wu, H.: The JH hash function, NIST SHA-3 competition submission (October 2008)

Appendix

A An Alternate Approach for Random Primitives

In this section we give an alternate proof of (a version of) Theorem 1 when the primitive $f = f_1$ of the compression function F is random, or more exactly when its outputs are independently distributed from each other (though not necessarily uniformly distributed across the range of f). This alternate version implies corollaries similar to corollaries 2 and 3, which we do not list. We present this alternate proof partly because some may find it more intuitive than the proof of Theorem 1 and partly because of the intrinsic interest of a supporting lemma, whose content and proof technique are of independent interest from the rest of the paper.

We start by stating this lemma, which we dub the 'MECMAC' lemma for 'Many Expected Collisions Means A Collision'.

Lemma 2 (MECMAC). *Let S be a set and let $c \leq |S|$ be a positive integer. Let $X_1, \ldots, X_n$ be independent random variables whose values are subsets of S of size at most c. Let $X = \sum_{i<j} |X_i \cap X_j|$ and let $\mu = E[X]$. Then*

$$\Pr[X = 0] \leq e^{-\mu/4c} + e^{-\sqrt{\mu/2c}} + \sqrt{\mu/2c}\, e^{1-\frac{3}{28}\sqrt{\mu/2c}}.$$

We do not believe the bound of Lemma 2 is sharp; we expect the optimal upper bound for $\Pr[X = 0]$ to be closer to $(1 + \sqrt{2\mu/c})e^{-\sqrt{2\mu/c}}$, but we could not achieve this bound with our current proof technique. Note Lemma 2 has a statement of the form: "Let $X_1, \ldots, X_n$ be independent random variables, and let $\mu = \sum_{i<j} f_{ij}(X_i, X_j)$ where $f_{ij} : \mathit{Range}(X_i) \times \mathit{Range}(X_j) \to [0, c]$. Then if μ/c is large, $\Pr[\sum f_{ij}(X_i, X_j) = 0]$ is small". However, this more general type of statement is not true, as can be seen from easily-constructed counterexamples. Thus Lemma 2 crucially relies on structural properties of set intersections (and in particular on the fact that if many sets intersect a single one, these are also likely to intersect each other).

Our alternate version of Theorem 1 for random primitives is the following:

Theorem 2. *Let S, N be positive integers and let $M \geq 16$ be a real number. Let F be a compression function of domain of size at least MS and range of size S making a single call to a primitive f of domain of size N whose outputs are independently distributed. Let $E \geq 16$ be such that $q = 1 + E\sqrt{N/M}$ is an integer and let $\psi = E/4$. Then if $1 + \lceil \log\log N \rceil < \frac{3}{8}M$ a collision can be found for F with probability at least $1 - g(\psi)$ where*

$$g(\psi) = e^{-\psi/4} + e^{-\sqrt{\psi/2}} + \sqrt{\psi/2}e^{1-\frac{3}{28}\sqrt{\psi/2}}$$

by using q queries to f.

Note: The constraint $\lceil \log\log N \rceil < \frac{3}{8}M$ does not correspond to any constraint in Theorem 1. In practice N is around 2^{128}, say, in which case $\lceil \log\log N \rceil < \frac{3}{8}M$ becomes $M > 64/3$, which is not much more restrictive than $M \geq 16$.

Proof of Theorem 2. Let D_F, R_F be the domain and range of F, and let g, h be the deterministic functions such that $F(v) = h(v, f(g(v)))$. Also let D_f be the domain of f. For each $x \in D_f$ let $T_x \subseteq T$ be the set of inputs $v \in D_F$ such that $g(v) = x$. Thus if the adversary makes the query $f(x)$ it learns to evaluate $F(v)$ for all $v \in T_x$.

For each $x \in D_f$ we let $\overline{X}_x = F(T_x)$. Note $\overline{X}_x$ is a random variable that depends on $f(x)$ and whose value is a subset of R_F. Then $\{\overline{X}_x : x \in D_f\}$ is an independent set of random variables. Let Coll_x be the event that a collision occurs among the inputs in T_x, namely that $|\overline{X}_x| < |F(T_x)|$. If $\Pr[\mathsf{Coll}_x] = 1$ for some x the adversary can simply query $f(x)$, so we may assume $\Pr[\mathsf{Coll}_x] < 1$ for all $x \in D_f$. (This poses the question of how the adversary "knows" the existence of such an x; however since the adversary is chosen after the parameters m, n, r, s and the distribution for f is fixed, the value of x may be hardcoded. Similar remarks apply to further points in the proof.) Let $X_x = \overline{X}_x | \neg\mathsf{Coll}_x$ be the modified random variable whose distribution is conditioned on the event $\neg\mathsf{Coll}_x$. Thus $|X_x| = |F(T_x)|$ and $\{X_x : x \in D_f\}$ is an independent set of random variables. We will exhibit a set $Z \subseteq D_f$ of size q such that

$$\Pr[X_x \cap X_y = \emptyset \text{ for all } x, y \in Z, x \neq y] \leq g(\psi).$$

This will prove the theorem since the adversary can query f at all the points in Z, and since the adversary obtains a collision anyway if Coll_x occurs for some $x \in Z$.

Define a sequence $\beta_0, \beta_1, \beta_2, \ldots$ by

$$\beta_k = E^{2^{k-1}-1}MS/N$$

for $k \geq 0$. Note that $\beta_{k+1} = \beta_k^2 EN/MS$. Let $U_k = \{x \in D_f : \beta_k < |T_x| \leq \beta_{k+1}\}$ and let $\Sigma_k = \sum_{x \in U_k} |T_x|$ for all $k \geq 0$.

Let $t \geq 0$ be the least integer such that $\beta_{t+1} \geq S$. Then

$$\begin{aligned} t &\leq \lceil \log(1 + \log(N/M)/\log(E)) \rceil \\ &\leq \lceil \log\log N \rceil \end{aligned}$$

using $M, E \geq 16$. Note we cannot have $|U_k| > 0$ for $k > t$, or else $\Pr[\mathsf{Coll}_x] = 1$ for $x \in U_k$. If $|\Sigma_k| \leq 2S$ for all $k \geq 0$ then because $E^{-\frac{1}{2}} \leq \frac{1}{4}$ and $1 + \lceil \log\log N \rceil < \frac{3}{8}M$,

$$\begin{aligned}|T| &= \sum_{x \in D_f} |T_x| \\ &\leq N\beta_0 + \sum_{k=0}^{t} |\Sigma_k| \\ &\leq N\beta_0 + (t+1)2S \\ &\leq E^{-\frac{1}{2}} MS + 2(1 + \lceil \log \log N \rceil)S \\ &< S\left(\frac{1}{4}M + \frac{3}{4}M\right) \\ &= MS\end{aligned}$$

a contradiction. Thus there must exist a value k_0 such that $\Sigma_{k_0} \geq 2S$.

If $|U_{k_0}| \leq q$ then the adversary can query f at all points in U_{k_0} and obtain a collision with probability 1, so we may assume $|U_{k_0}| \geq q$. Now consider the following two experiments:

(1) query f at all points in U_{k_0}, resulting in values of X_x for $x \in U_{k_0}$, then select q distinct sets $X_{x_1}, \dots, X_{x_q}$ uniformly at random from $\{X_x : x \in U_{k_0}\}$, and remove the other sets
(2) query f at q distinct random points $x_1, \dots, x_q$ in U_{k_0}, resulting in q known sets $X_{x_1}, \dots, X_{x_q}$

Clearly these two experiments have identical outcomes. For each experiment, let a "collision" be a triple (i, j, t) with $i < j$ such that $t \in X_{x_i} \cap X_{x_j}$. We will show that in experiment (1) the expected number of collisions is at least $\psi\beta_{k_0+1}$ and hence that there exists some set Z of q distinct values $x_1, \dots, x_q \in U_{k_0}$ such that the expected number of collisions among $X_{x_1}, \dots, X_{x_q}$ is at least $\psi\beta_{k_0+1}$.

Let $\Sigma_{k_0} = aS$ where $a \geq 2$. After the first stage of experiment (1) it is easy to see (even when a is not an integer) that there are at least $\frac{a(a-1)}{2}S \geq a^2S/4$ collisions among the sets $\{X_x : x \in U_{k_0}\}$. When selecting q distinct sets at random from the set of $|U_{k_0}|$ sets, each collision remains selected with probability at least $\frac{q(q-1)}{|U_{k_0}|(|U_{k_0}|-1)} \geq (q-1)^2/|U_{k_0}|^2$, so by linearity of expectation the expected number of collisions in experiment (1) is at least $a^2S(q-1)^2/4|U_{k_0}|^2$. Since $|U_{k_0}|\beta_{k_0} \leq \Sigma_{k_0} = aS$, we have $|U_{k_0}| \leq aS/\beta_{k_0}$, so we have

$$\begin{aligned}\frac{a^2S(q-1)^2}{4|U_{k_0}|^2} &\geq \frac{a^2S(q-1)^2}{4a^2S^2/\beta_{k_0}^2} \\ &= \frac{\beta_{k_0}^2(q-1)^2}{4S} \\ &= \frac{\beta_{k_0}^2 E^2 N}{4MS} \\ &= \psi\beta_{k_0+1}\end{aligned}$$

where we used $\beta_{k+1} = \beta_k^2 EN/MS$ and $\psi = E/4$.

By the probabilistic argument outlined earlier, there therefore exist a set Z of q distinct points $x_1, \dots, x_q$ such that the expected number of collisions among

$X_{x_1}, \ldots, X_{x_q}$ is at least $\psi\beta_{k_0+1}$. However by the definition of U_{k_0} we have $|X_{x_i}| \leq \beta_{k_0+1}$ for $i = 1 \ldots q$, so, because $X_{x_1}, \ldots, X_{x_q}$ are independent, Lemma 2 applied with $\mu = \psi\beta_{k_0+1}$ and $c = \beta_{k_0+1}$ implies the probability of no collisions among them is at most $g(\psi)$, as desired. □

Proof of the MECMAC Lemma. Because the bound is void for $\mu \leq 2c$ we can assume $\mu \geq 2c$. For any partition $\mathcal{C}, \mathcal{D}$ of $[n] = \{1, 2, \ldots, n\}$ let

$$X_{\mathcal{C},\mathcal{D}} = |\{(i, j, s) : s \in X_i \cap X_j \text{ and } (i, j) \in (\mathcal{C} \times \mathcal{D}) \cup (\mathcal{D} \times \mathcal{C})\}|$$

and let $\mu_{\mathcal{C},\mathcal{D}} = E[X_{\mathcal{C},\mathcal{D}}]$. If $\mathcal{C}, \mathcal{D}$ are selected at random by independently placing each element of $[n]$ in $\mathcal{C}$ or $\mathcal{D}$ with probability $\frac{1}{2}$ then

$$E[\mu_{\mathcal{C},\mathcal{D}}] = \frac{1}{2}\mu$$

since for each triplet (i, j, s) such that $s \in X_i \cap X_j$ and $i \neq j$ there is chance $\frac{1}{2}$ that $(i, j) \in (\mathcal{C} \times \mathcal{D}) \cup (\mathcal{D} \times \mathcal{C})$. Therefore there must exist a partition $\mathcal{A}$, $\mathcal{B}$ of $[n]$ such that $\mu_{\mathcal{A},\mathcal{B}} \geq \frac{1}{2}\mu$.

Let $k = |\mathcal{A}|$, $\ell = |\mathcal{B}|$. We rename $X_1, \ldots, X_n$ as two lists $A_1, \ldots, A_k$ and $B_1, \ldots, B_\ell$ such that $\{A_1, \ldots, A_k\} = \{X_i : i \in \mathcal{A}\}$ and $\{B_1, \ldots, B_\ell\} = \{X_j : j \in \mathcal{B}\}$. For $1 \leq i \leq k$ let $Y_i = |\{(j, s) : s \in A_i \cap B_j\}|$ and let $\mu_i = E[Y_i]$. Then

$$\sum_{i=1}^{k} \mu_i = \mu_{\mathcal{A},\mathcal{B}}.$$

For all $U \subseteq S$ let

$$\beta_U = \sum_{j=1}^{\ell} E\big[|B_j \cap U|\big].$$

We have

$$\mu_i = \sum_{U \subseteq S} \beta_U \Pr[A_i = U] = E[\beta_{A_i}].$$

Let $M = \sqrt{\mu_{\mathcal{A},\mathcal{B}}/c} \geq \sqrt{\mu/2c} \geq 1$. Assume first there is some $s \in S$ such that $\beta_s > M$. Then letting $\alpha_j = \Pr[B_j = s] = E[|B_j \cap \{s\}|]$ we have $\alpha_1 + \cdots + \alpha_\ell > M$ and

$$\begin{aligned}
\Pr[X = 0] &\leq \prod_{j=1}^{\ell}(1 - \alpha_j) + \sum_{j=1}^{\ell} \alpha_j \prod_{h=1, h \neq j}^{\ell} (1 - \alpha_h) \\
&\leq e^{-\alpha_1 - \ldots - \alpha_\ell} + \sum_{j=1}^{\ell} \alpha_j e^{\alpha_j - \alpha_1 - \ldots - \alpha_\ell} \\
&\leq e^{-M} + e^{1 - \alpha_1 - \ldots - \alpha_\ell} \sum_{j=1}^{\ell} \alpha_j \\
&\leq e^{-M} + M e^{1-M} \\
&\leq e^{-\sqrt{\mu/2c}} + \sqrt{\mu/2c}\, e^{1-\sqrt{\mu/2c}}
\end{aligned}$$

where the last two inequalities use the fact that ye^{-y} is a decreasing function of y for $y \geq 1$.

Now assume instead that $\beta_s \leq M$ for all $s \in S$. Since

$$\beta_{A_i} = \sum_{s \in A_i} \beta_s \leq Mc,$$

β_{A_i} is a nonnegative r.v. bounded by Mc of mean μ_i for $1 \leq i \leq k$, so

$$\mathrm{Var}(\beta_{A_i}) \leq \mu_i(Mc - \mu_i)$$

for $1 \leq i \leq k$ and

$$\begin{aligned}\sum_{i=1}^{k} \mathrm{Var}(\beta_{A_i}) &\leq \sum_{i=1}^{k} \mu_i(Mc - \mu_i)\\ &\leq Mc \sum_{i=1}^{k} \mu_i\\ &= c^{\frac{1}{2}} \mu_{\mathcal{A},\mathcal{B}}^{\frac{3}{2}}.\end{aligned}$$

Because β_{A_1}, …, β_{A_k} are independent and uniformly bounded by Mc, Bernstein's inequality (see notes at bottom) then implies

$$\begin{aligned}\Pr\left[\sum_{i=1}^{k} \beta_{A_i} \leq \mu_{\mathcal{A},\mathcal{B}}/2\right] &\leq \exp\left(-\frac{(\mu_{\mathcal{A},\mathcal{B}}/2)^2/2}{\sum_{i=1}^{k} \mathrm{Var}(\beta_{A_i}) + Mc\mu_{\mathcal{A},\mathcal{B}}/6}\right)\\ &\leq \exp\left(-\frac{\mu_{\mathcal{A},\mathcal{B}}^2/8}{c^{\frac{1}{2}}\mu_{\mathcal{A},\mathcal{B}}^{\frac{3}{2}} + c^{\frac{1}{2}}\mu_{\mathcal{A},\mathcal{B}}^{\frac{3}{2}}/6}\right)\\ &\leq e^{-\frac{3}{28}(\mu_{\mathcal{A},\mathcal{B}}/c)^{\frac{1}{2}}}\\ &\leq e^{-\frac{3}{28}\sqrt{\mu/c}}.\end{aligned}$$

Let "$\Sigma_{\geq}$" be the event that $\sum_{i=1}^{k} \beta_{A_i} \geq \mu_{\mathcal{A},\mathcal{B}}/2$ and let "$A_{\neq}$" be the event that $A_i \cap A_j = \emptyset$ for $i \neq j$. We have

$$\begin{aligned}\Pr[X = 0] &= \Pr[X = 0 \,|\, \Sigma_{\geq}] \Pr[\Sigma_{\geq}] + \Pr[X = 0 \,|\, \neg\Sigma_{\geq}] \Pr[\neg\Sigma_{\geq}]\\ &\leq \Pr[X = 0 \,|\, \Sigma_{\geq}] + \Pr[\neg\Sigma_{\geq}]\\ &\leq \Pr[X = 0 \,|\, \Sigma_{\geq}] + e^{-\frac{3}{28}\sqrt{\mu/c}}\end{aligned}$$

and, since $\neg A_{\neq} \implies X \geq 1$,

$$\begin{aligned}\Pr[X = 0 \,|\, \Sigma_{\geq}] = {} & \Pr[X = 0 \,|\, \Sigma_{\geq} \wedge A_{\neq}] \Pr[A_{\neq} \,|\, \Sigma_{\geq}] + {}\\ & \Pr[X = 0 \,|\, \Sigma_{\geq} \wedge \neg A_{\neq}] \Pr[\neg A_{\neq} \,|\, \Sigma_{\geq}]\\ = {} & \Pr[X = 0 \,|\, \Sigma_{\geq} \wedge A_{\neq}] \Pr[A_{\neq} \,|\, \Sigma_{\geq}]\\ \leq {} & \Pr[X = 0 \,|\, \Sigma_{\geq} \wedge A_{\neq}].\end{aligned}$$

Moreover

$$\Pr[X = 0 \,|\, \Sigma_{\geq} \wedge A_{\neq}] \leq \prod_{j=1}^{\ell} \Pr[B_j \cap (A_1 \cup \cdots \cup A_k) = \emptyset \,|\, \Sigma_{\geq} \wedge A_{\neq}].$$

To upper bound the latter probability, fix any values of $A_1, \ldots, A_k$ such that $\Sigma_{\geq} \wedge A_{\neq}$. For $1 \leq j \leq \ell$ let B'_j be a new random variable that selects uniformly at random an element from B_j. Then

$$\begin{aligned}\prod_{j=1}^{\ell} \Pr[B_j \cap (A_1 \cup \cdots \cup A_k) = \emptyset] &\leq \prod_{j=1}^{\ell} \Pr[B'_j \notin A_1 \cup \cdots \cup A_k] \\ &= \prod_{j=1}^{\ell} \Big(1 - \sum_{i=1}^{k} \Pr[B'_j \in A_i]\Big) \\ &\leq e^{-\sum_{j=1}^{\ell}\sum_{i=1}^{k} \Pr[B'_j \in A_i]} \\ &= e^{-\sum_{j=1}^{\ell}\sum_{i=1}^{k} E[|B'_j \cap A_i|]} \\ &\leq e^{-\sum_{i=1}^{k} \beta_{A_i}/c} \\ &\leq e^{-\mu_{\mathcal{A},\mathcal{B}}/2c} \\ &\leq e^{-\mu/4c}\end{aligned}$$

where $A_{\neq}$ is used going to the second line and $\Sigma_{\geq}$ is used in the next-to-last inequality. Thus

$$\Pr[X = 0 \,|\, \Sigma_{\geq} \wedge A_{\neq}] \leq e^{-\mu/4c}.$$

Combining these results we have

$$\Pr[X = 0] \;\leq\; e^{-\mu/4c} + e^{-\frac{3}{28}\sqrt{\mu/c}}$$

if $\beta_s \leq M$ for all $s \in S$, and

$$\Pr[X = 0] \;\leq\; e^{-\sqrt{\mu/2c}} + \sqrt{\mu/2c}\, e^{1-\sqrt{\mu/2c}}$$

if $\beta_s \geq M$ for some $s \in S$, so we can conclude that

$$\Pr[X = 0] \;\leq\; e^{-\mu/4c} + e^{-\sqrt{\mu/2c}} + \sqrt{\mu/2c}\, e^{1-\frac{3}{28}\sqrt{\mu/2c}}$$

in all cases. □

Bernstein's Inequality

Let $Z_1, \ldots, Z_n$ be independent random variables of mean zero such that $|Z_i| \leq M$ almost surely for $1 \leq i \leq n$. Bernstein's inequality states that

$$\Pr\left[\sum_{i=1}^{n} Z_i \geq t\right] \leq \exp\Big(-\frac{t^2/2}{\sum_{i=1}^{n} E[Z_i^2] + Mt/3}\Big).$$

for all $t > 0$. Now let $T_1, \ldots, T_n$ be independent random variables of nonzero mean such that $T_i \in [0, M]$ almost surely, and let $\mu = E[T_1 + \cdots + T_n]$. By Bernstein's inequality applied to $Z_1 = -(T_1 - E[T_1])$, …, $Z_n = -(T_n - E[T_n])$ (so $|Z_i| \le M$ a.s.) we have

$$\begin{aligned}
\Pr\left[\sum_{i=1}^{n} T_i \le \mu/2\right] &= \Pr\left[\sum_{i=1}^{n}(T_i - E[T_i]) \le -\mu/2\right] \\
&= \Pr\left[\sum_{i=1}^{n} Z_i \ge \mu/2\right] \\
&\le \exp\Big(-\frac{(\mu/2)^2/2}{\sum_{i=1}^{n} E[Z_i^2] + M\mu/6}\Big) \\
&= \exp\Big(-\frac{(\mu/2)^2/2}{\sum_{i=1}^{n} \mathrm{Var}(T_i) + M\mu/6}\Big).
\end{aligned}$$

This is the form used in the proof of the MECMAC lemma.

Universal One-Way Hash Functions via Inaccessible Entropy

Iftach Haitner[1], Thomas Holenstein[2], Omer Reingold[3,⋆],
Salil Vadhan[4,⋆⋆], and Hoeteck Wee[5,⋆⋆⋆]

[1] Microsoft Research, New England
iftach@microsoft.com
[2] Department of Computer Science, ETH Zurich
thomas.holenstein@inf.ethz.ch
[3] Microsoft Research, Silicon Valley and Weizmann Institute of Science
omreing@microsoft.com
[4] School of Engineering & Applied Sci. and Center for Research on Computation & Society,
Harvard University
salil@seas.harvard.edu
[5] Queens College, CUNY
hoeteck@cs.qc.cuny.edu

Abstract. This paper revisits the construction of Universal One-Way Hash Functions (UOWHFs) from any one-way function due to Rompel (STOC 1990). We give a simpler construction of UOWHFs, which also obtains better efficiency and security. The construction exploits a strong connection to the recently introduced notion of *inaccessible entropy* (Haitner et al. STOC 2009). With this perspective, we observe that a small tweak of any one-way function f is already a weak form of a UOWHF: Consider $F(x, i)$ that outputs the i-bit long prefix of $f(x)$. If F were a UOWHF then given a random x and i it would be hard to come up with $x' \neq x$ such that $F(x, i) = F(x', i)$. While this may not be the case, we show (rather easily) that it is hard to sample x' with almost full entropy among all the possible such values of x'. The rest of our construction simply amplifies and exploits this basic property.

With this and other recent works, we have that the constructions of three fundamental cryptographic primitives (Pseudorandom Generators, Statistically Hiding Commitments and UOWHFs) out of one-way functions are to a large extent unified. In particular, all three constructions rely on and manipulate computational notions of entropy in similar ways. Pseudorandom Generators rely on the well-established notion of pseudoentropy, whereas Statistically Hiding Commitments and UOWHFs rely on the newer notion of inaccessible entropy.

Keywords: computational complexity, cryptography, hashing, target collision-resistance, one-way functions.

[⋆] Supported by US-Israel BSF grant 2006060.
[⋆⋆] Supported by NSF grant CNS-0831289 and US-Israel BSF grant 2006060.
[⋆⋆⋆] Supported in part by PSC-CUNY Award #6014939 40.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 616–637, 2010.

1 Introduction

Universal one-way hash functions (UOWHFs), as introduced by Naor and Yung [10], are a weaker form of collision-resistant hash functions. The standard notion of collision resistance requires that given a randomly chosen function $f \overset{\mathrm{R}}{\leftarrow} \mathcal{F}$ from the hash family, it is infeasible to find any pair of distinct inputs x, x' such that $f(x) = f(x')$. UOWHFs only require *target collision resistance*, where the adversary must specify one of the inputs x before seeing the description of the function f. Formally:

Definition 1. *A family of functions $\mathcal{F}_k = \{F_z : \{0,1\}^{n(k)} \to \{0,1\}^{m(k)}\}_{z \in \{0,1\}^k}$ is a family of* universal one-way hash functions (UOWHFs) *if it satisfies:*

1. *Efficiency: Given $z \in \{0,1\}^k$ and $x \in \{0,1\}^{n(k)}$, $F_z(x)$ can be evaluated in time* $\mathrm{poly}(n(k), k)$.
2. *Shrinking: $m(k) < n(k)$.*
3. *Target Collision Resistance: For every probabilistic polynomial-time adversary A, the probability that A succeeds in the following game is negligible in k:*
   (a) Let $(x, \mathsf{state}) \leftarrow A(1^k) \in \{0,1\}^{n(k)} \times \{0,1\}^*$.
   (b) Choose $z \overset{R}{\leftarrow} \{0,1\}^k$.
   (c) Let $x' \overset{R}{\leftarrow} A(\mathsf{state}, z) \in \{0,1\}^{n(k)}$.
   (d) A succeeds *if $x \neq x'$ and $F_z(x) = F_z(x')$.*

It turns out that this weaker security property suffices for many applications. The most immediate application given in [10] is *secure fingerprinting*, whereby the pair $(f, f(x))$ can taken as a compact "fingerprint" of a large file x, such that it is infeasible for an adversary, seeing the fingerprint, to change the file x to x' without being detected. More dramatically, Naor and Yung [10] also showed that UOWHFs can be used to construct secure digital signature schemes, whereas all previous constructions (with proofs of security in the standard model) were based on trapdoor functions (as might have been expected to be necessary due to the public-key nature of signature schemes). More recently, UOWHFs have been used in the Cramer–Shoup encryption scheme [3] and in the construction of statistically hiding commitment schemes from one-way functions [4, 5].

Naor and Yung [10] gave a simple and elegant construction of UOWHFs from any one-way *permutation*. Subsequently, Rompel [11] gave a much more involved construction to prove that UOWHFs can be constructed from an arbitrary one-way function, thereby resolving the complexity of UOWHFs (as one-way functions are the minimal complexity assumption for complexity-based cryptography, and are easily implied by UOWHFs).[1] While complications may be expected for constructions from arbitrary one-way functions (due to their lack of structure), Rompel's analysis also feels quite ad hoc. In contrast, the construction of pseudorandom generators from one-way functions of [7], while also somewhat complex, involves natural abstractions (e.g., pseudoentropy) that allow for modularity and measure for what is being achieved at each stage of the construction.

In this paper, we give simpler constructions of UOWHFs from one-way functions, based on (a variant of) the recently introduced notion of *inaccessible entropy* [5]. In addition, one of the constructions obtains slightly better efficiency and security.

[1] More details of Rompel's proof are worked out, with some corrections, in [12, 9].

1.1 Inaccessible Entropy

For describing our construction, it will be cleaner to work with a variant of UOWHFs where there is a *single* shrinking function $F : \{0,1\}^n \to \{0,1\}^m$ (for each setting of the security parameter k) such that it is infeasible to find collisions with *random inputs*. That is, an adversary A is given a uniformly random $x \overset{R}{\leftarrow} \{0,1\}^n$, outputs an x' such that $F(x') = F(x)$, and succeeds if $x' \neq x$.[2] Note that we can assume without loss of generality that $x' = A(x)$ is always a preimage of $F(x)$ (A has the option of outputting x in case it does not find a different preimage); we refer to an algorithm A with this property as an *F-collision finder*.

Our construction is based on an entropy-theoretic view of UOWHFs. The fact that F is shrinking implies that there are many preimages x' available to A. Indeed, if we consider an (inefficient) adversary $A(x)$ that outputs $x' \overset{R}{\leftarrow} F^{-1}(F(x))$ and let X be a random variable uniformly distributed on $\{0,1\}^n$, then

$$\mathrm{H}(A(X)|X) = \mathrm{H}(X|F(X)) \geq n - m,$$

where $\mathrm{H}(\cdot|\cdot)$ denotes conditional Shannon entropy. (See Section 2 for more definitional details.) We refer to the quantity $\mathrm{H}(X|F(X))$ as the *real entropy of F^{-1}*.

On the other hand, the target collision resistance means that effectively only one of the preimages is accessible to A. That is for every probabilistic polynomial-time F-collision finder A, we have $\Pr[A(X) \neq X] = \mathrm{neg}(n)$, which is equivalent to requiring that:

$$\mathrm{H}(A(X)|X) = \mathrm{neg}(n)$$

for all probabilistic polynomial-time F-collision finders A. (If A can find a collision X' with nonnegligible probability, then it can achieve nonnnegligible conditional entropy by outputting X' with probability 1/2 and outputting X with probability $1/2$.) We refer to the maximum of $\mathrm{H}(A(X)|X)$ over all efficient F-collision finders as the *accessible entropy of F^{-1}*. We stress that accessible entropy refers to an *upper bound* on a form of computational entropy, in contrast to the Håstad et al.'s notion of *pseudoentropy* [7].

Thus, a natural weakening of the UOWHF property is to simply require a noticeable gap between the real and accessible entropies of F^{-1}. That is, for every probabilistic polynomial-time F-collision finder A, we have $\mathrm{H}(A(X)|X) < \mathrm{H}(X|F(X)) - \Delta$, for some noticeable Δ, which we refer to as the *inaccessible entropy of F*.

1.2 Our Constructions

Our constructions of UOWHFs have two parts. First, we show how to obtain a function with noticeable inaccessible entropy from any one-way function. Second, we show how to build a UOWHF from any function with inaccessible entropy.

OWFs ⇒ Inaccessible Entropy. Given a one-way function $f : \{0,1\}^n \to \{0,1\}^m$, we show that a random truncation of f has inaccessible entropy. Specifically, we define $F(x, i)$ to be the first i bits of $f(x)$.

[2] It is easy to convert any such function F into a standard UOWHF family by defining $F_z(x) = F(z + x)$.

To see that this works, suppose for contradiction that F does not have noticeable inaccessible entropy. That is, we have an efficient adversary A that on input (x, i) can sample from the set $S(x, i) = \{x' : f(x')_{1\ldots i} = f(x)_{1\ldots i}\}$ with almost-maximal entropy, which is equivalent to sampling according to a distribution that is statistically close to the uniform distribution on $S(x, i)$. We can now use A to construct an inverter *Inv* for f that works as follows on input y: choose $x_0 \overset{\text{R}}{\leftarrow} \{0,1\}^n$, and then for $i = 1, \ldots, n$ generate a random $x_i \overset{\text{R}}{\leftarrow} A(x_{i-1}, i-1)$ subject to the constraint that $f(x_i)_{1,\cdots,i} = y_{1,\cdots,i}$. The latter step is feasible, since we are guaranteed that $f(x_i)_{1,\ldots,i-1} = y_{1,\cdots,i-1}$ by the fact that A is an F-collision finder, and the expected number of trials needed get agreement with y_i is at most 2 (since $y_i \in \{0,1\}$, and y and $f(x_i)$ are statistically close). It is not difficult to show that when run on a random output Y of f, *Inv* produces an almost-uniform preimage of Y. This contradicts the one-wayness of f. Indeed, we only need f to be a *distributional* one-way function [8], whereby it is infeasible to generate almost-uniform preimages under f.

Inaccessible Entropy $\Rightarrow$ *UOWHFs.* Once we have a non-negligible amount of inaccessible entropy, we can construct a UOWHF via a series of standard transformations.

1. **Repetition:** By evaluating F on many inputs, we can increase the amount of inaccessible entropy from $1/\operatorname{poly}(n)$ to $\operatorname{poly}(n)$. Specifically, we take $F^t(x_1, \ldots, x_t) = (F(x_1), \ldots, F(x_t))$ where $t = \operatorname{poly}(n)$. This transformation also has the useful effect of converting the real entropy of F^{-1} to *min-entropy*.
2. **Hashing Inputs:** By hashing the input to F (namely taking $F'(x, g) = (F(x), g(x))$ for a universal hash function g), we can reduce both the real (min-)entropy and the accessible entropy so that $(F')^{-1}$ still has a significant amount of real entropy, but has (weak) target collision resistance (on random inputs).
3. **Hashing Outputs:** By hashing the output to F (namely taking $F'(x, g) = g(F(x))$), we can reduce the output length of F to obtain a shrinking function that still has (weak) target collision resistance.

There are two technicalities that occur in the above steps. First, hashing the inputs only yields *weak* target collision resistance; this is due to the fact that accessible Shannon entropy is an average-case measure and thus allows for the possibility that the adversary can achieve high accessible entropy most of the time. Fortunately, this weak form of target collision resistance can be amplified to full target collision resistance using another application of repetition and hashing (similar to [1]).

Second, the hashing steps require having a fairly accurate estimate of the real entropy. This can be handled similarly to [7, 11], by trying all (polynomially many) possibilities and concatenating the resulting UOWHFs, at least one of which will be target collision resistant.

A More Efficient Construction. We obtain a more efficient construction of UOWHFs by hashing the output of the one-way function f before truncating. That is, we define $F(x, g, i) = (g, g(f(x))_{1\cdots i})$. This function is in the spirit of the function that Rompel [11] uses as a first step, but our function uses three-wise independent hash function instead of n-wise independent one, and enjoys a much simpler structure.[3] Our analysis of this function is significantly simpler than Rompel's and can be viewed as providing a clean abstraction of what it achieves (namely, inaccessible entropy) that makes the subsequent transformation to a UOWHF much easier.

We obtain improved UOWHF parameters over our first construction for two reasons. First, we obtain a larger amount of inaccessible entropy: $(\log n)/n$ bits instead of roughly $1/n^4$ bits. Second, we obtain a bound on a stronger form of accessible entropy, which enables us to get full target collision resistance when we hash the inputs, avoiding the second amplification step.

This construction yields better parameters than Rompel's original construction. A one-way function of input length n yields a UOWHF with output length $\tilde{O}(n^7)$, improving Rompel's bound of $\tilde{O}(n^8)$. Additionally, we are able to reduce the key length needed: Rompel's original construction uses a key of length $\tilde{O}(n^{12})$, whereas our construction only needs a key of length $\tilde{O}(n^7)$. If we allow the construction to utilize some nonuniform information (namely an estimate of the real entropy of F^{-1}), then we obtain output length $\tilde{O}(n^5)$, improving Rompel's bound of $\tilde{O}(n^6)$. For the key length, the improvement in this case is from $\tilde{O}(n^7)$ to $\tilde{O}(n^5)$. Of course, these bounds are still far from practical, but they illustrate the utility of inaccessible entropy in reasoning about UOWHFs, which may prove useful in future constructions (whether based on one-way functions or other building blocks).

1.3 Perspective

The idea of inaccessible entropy was introduced in [5] for the purpose of constructing statistically hiding commitment schemes from one-way functions and from zero-knowledge proofs. There, the nature of statistically hiding commitments necessitated more involved notions of inaccessible entropy than we present here — inaccessible entropy was defined in [5] for interactive protocols and for "generators" that output many blocks, where one considers adversaries that try to generate next-messages or next-blocks of high entropy. In such a setting, it is necessary to have the adversary privately "justify" that it is behaving consistently with the honest party, and to appropriately discount the entropy in case the adversary outputs an invalid justification.

Here, we are able to work with a much simpler form of inaccessible entropy. The simplicity comes from the noninteractive nature of UOWHFs (so we only need to measure the entropy of a single string output by the adversary), and the fact that we can assume without loss of generality that the adversary behaves consistently with the honest party. Thus, the definitions here can serve as a gentler introduction to the concept of inaccessible entropy. On the other hand, the many-round notions from [5] allow for a useful "entropy equalization" transformation that avoids the need to try all possible guesses for the entropy. We do not know an analogous transformation for constructing UOWHFs. We also note that our simple construction of a function with inaccessible entropy by randomly truncating a one-way function (and its analysis) is inspired by the the construction of an "inaccessible entropy generator" from a one-way function in [5].

[3] Rompel started with the function $f'(z, g_1, g_2) := (g_2(f_0(g_1(z))), g_1, g_2)$, where g_1 and g_2 are n-wise independent hash-functions, and f_0 is defined as $f_0(x, y, i) = (f(x), y^{n-i}, 0^i)$.

Finally, with our constructions, the proof that one-way functions imply UOWHFs now parallels those of pseudorandom generators [7, 6] and statistically hiding commitments [4, 5], with UOWHFs and statistically hiding commitments using dual notions of entropy (high real entropy, low accessible entropy) to pseudorandom generators (low real entropy, high pseudoentropy).

2 Inaccessible Entropy for Inversion Problems

We will refer to several measures of entropy in this work (proofs are omitted due to the lack of space). For a random variable X and $x \in \operatorname{Supp}(X)$, we define the *sample-entropy* of x with respect to X to be the quantity

$$\mathrm{H}_X(x) := \log(1/\Pr[X = x]).$$

Using this notion, we can define the *Shannon entropy* $\mathrm{H}(X)$ and *min-entropy* $\mathrm{H}_\infty(X)$ as follows:

$$\mathrm{H}(X) := \mathrm{E}_{x \overset{\mathrm{R}}{\leftarrow} X}[\mathrm{H}_X(x)] \quad \text{and} \quad \mathrm{H}_\infty(X) := \min_{x \in \operatorname{Supp}(X)} \mathrm{H}_X(x)$$

We will also discuss the *max-entropy* $\mathrm{H}_0(X) := \log(1/|\operatorname{Supp}(X)|)$. It can be shown that $\mathrm{H}_\infty(X) \leq \mathrm{H}(X) \leq \mathrm{H}_0(X)$ with equality if and only if X is flat.

As discussed in the introduction, for a function F, we define the *real entropy* of F^{-1} to be the amount of entropy left in the input after revealing the output.

Definition 2. *Let n be a security parameter, and $F : \{0,1\}^n \to \{0,1\}^m$ a function. We say that F^{-1} has* real Shannon entropy *k if*

$$\mathrm{H}(X|F(X)) = k,$$

where X is uniformly distributed on $\{0,1\}^n$. We say that F^{-1} has real min-entropy *at least k if there is a negligible function $\varepsilon = \varepsilon(n)$ such that*

$$\Pr_{x \overset{R}{\leftarrow} X}\left[\mathrm{H}_{X|F(X)}(x|F(x)) \geq k\right] \geq 1 - \varepsilon(n).$$

We say that F^{-1} has real max-entropy *at most k if there is a negligible function $\varepsilon = \varepsilon(n)$ such that*

$$\Pr_{x \overset{R}{\leftarrow} X}\left[\mathrm{H}_{X|F(X)}(x|F(x)) \leq k\right] \geq 1 - \varepsilon(n).$$

Note that more concrete formulas for the entropies above are:

$$\mathrm{H}_{X|F(X)}(x|F(x)) = \log|F^{-1}(F(x))| \text{ and } \mathrm{H}(X|F(X)) = \mathrm{E}\left[\log|F^{-1}(F(X))|\right].$$

As our goal is to construct UOWHFs that are shrinking, achieving high real entropy is a natural intermediate step. Indeed, the amount by which F shrinks is a lower bound on the real entropy of F^{-1}:

Proposition 1. *If $F : \{0,1\}^n \to \{0,1\}^m$, then the real Shannon entropy of F^{-1} is at least $n - m$, and the real min-entropy of F^{-1} is at least $n - m - s$ for any $s = \omega(\log n)$.*

To motivate the definition of accessible entropy, we now present an alternative formulation of real entropy in terms of the entropy that computationally unbounded "collision-finding" adversaries can generate.

Definition 3. *For a function $F : \{0,1\}^n \to \{0,1\}^m$, an F*-collision-finder *is a randomized algorithm A such that for every $x \in \{0,1\}^n$ and coin tosses r for A, we have $A(x;r) \in F^{-1}(F(x))$.*

Note that A is required to *always* produce an input $x' \in \{0,1\}^n$ such that $F(x) = F(x')$. This is a reasonable constraint because A has the option of outputting $x' = x$ if it does not find true collision. We consider A's goal to be maximizing the entropy of its output $x' = A(x)$, given a random input x. If we let A be computationally unbounded, then the optimum turns out to equal exactly the real entropy:

Proposition 2. *Let $F : \{0,1\}^n \to \{0,1\}^m$. Then the real Shannon entropy of F^{-1} equals the maximum of $\mathrm{H}(A(X;R)|X)$ over all (computationally unbounded) F-collision finders A, where random variable X is uniformly distributed in $\{0,1\}^n$ and R is uniformly random coin tosses for A. That is,*

$$\mathrm{H}(X|F(X)) = \max_A \mathrm{H}(A(X;R)|X),$$

where the maximum is taken over all F-collision finders A.

The notion of *accessible entropy* simply restricts the above to efficient adversaries, e.g. those that run in probabilistic polynomial time (PPT for short):

Definition 4. *Let n be a security parameter and $F : \{0,1\}^n \to \{0,1\}^m$ a function. We say that F^{-1} has* accessible Shannon entropy *at most k if for every* PPT *F-collision-finder A, we have*

$$\mathrm{H}(A(X;R)|X) \leq k$$

for all sufficiently large n, where random variable X is uniformly distributed on $\{0,1\}^n$ and R is uniformly random coin tosses for A.

As usual, it is often useful to have an upper bound not only on Shannon entropy, but on the max-entropy (up to some negligible statistical distance). Recall that a random variable Z has max-entropy at most k iff the support of Z is contained in a set of size 2^k. Thus, we require that $A(X;R)$ is contained in a set $L(X)$ of size at most 2^k, except with negligible probability:

Definition 5. *Let n be a security parameter and $F : \{0,1\}^n \to \{0,1\}^m$ a function. For $p = p(n) \in [0,1]$, we say that F^{-1} has p*-accessible max-entropy *at most k if for every* PPT *F-collision-finder A, there exists a family of sets $\{L(x)\}_{x \in \mathrm{Supp}(X)}$ each of size at most 2^k such that $x \in L(x)$ for all $x \in \mathrm{Supp}(X)$ and*

$$\Pr\left[A(X;R) \in L(X)\right] \geq 1 - p$$

for all sufficiently large n, where random variable X is uniformly distributed on $\{0,1\}^n$ and R is uniformly random coin tosses for A. In addition, if $p = \varepsilon(n)$ for some negligible function $\varepsilon(\cdot)$, then we simply say that F^{-1} has accessible max-entropy *at most k.*

The reason that having an upper bound on accessible entropy is useful as an intermediate step towards constructing UOWHFs is that accessible max-entropy 0 is equivalent to target collision resistance (on random inputs):

Definition 6. *Let $F : \{0,1\}^n \to \{0,1\}^m$ be a function. For $q = q(n) \in [0,1]$, we say that F is q-*collision-resistant on random inputs *if for every* PPT *F-collision-finder A,*

$$\Pr[A(X;R) \ = X] \geq q,$$

for all sufficiently large n, where random variable X is uniformly distributed on $\{0,1\}^n$ and R is uniformly random coin tosses for A. In addition, if $q = 1 - \varepsilon(n)$ for some negligible function $\varepsilon(\cdot)$, we say that F is collision-resistant on random inputs.

Lemma 1. *Let n be a security parameter and $F : \{0,1\}^n \to \{0,1\}^m$ be a function. Then, for any $p = p(n) \in (0,1)$, the following statements are equivalent:*

(1) *F^{-1} has p-accessible max-entropy* 0.
(2) *F is $(1-p)$-collision-resistant on random inputs.*

In particular, F^{-1} has accessible max-entropy 0 *iff F is collision-resistant on random inputs.*

While bounding p-accessible max-entropy with negligible p is our ultimate goal, one of our constructions will work by first giving a bound on accessible Shannon entropy, and then deducing a bound p-accessible max-entropy for a value of $p < 1$ using the following lemma:

Lemma 2. *Let n be a security parameter and $F : \{0,1\}^n \to \{0,1\}^m$ be a function. If F^{-1} has accessible Shannon entropy at most k, then F^{-1} has p-accessible max-entropy at most $k/p + O(2^{-k/p})$ for any $p = p(n) \in (0,1)$.*

Once we have a bound on p-accessible max-entropy for some $p < 1$, we need to apply several transformations to obtain a function with a good bound on $\operatorname{neg}(n)$-accessible max-entropy.

Our second construction (which achieves better parameters), starts with a bound on a different average-case form of accessible entropy, which is stronger than bounding the accessible Shannon entropy. The benefit of this notion it that it can be converted more efficiently to $\operatorname{neg}(n)$-accessible max-entropy, by simply taking repetitions.

To motivate the definition, recall that a bound on accessible Shannon entropy means that the sample entropy $\mathrm{H}_{A(X;R)|X}(x'|x)$ is small on average over $x \overset{\mathrm{R}}{\leftarrow} X$ and $x' \overset{\mathrm{R}}{\leftarrow} A(x;R)$. This sample entropy may depend on both the input x and the x' output by the adversary (which in turn may depend on its coin tosses). A stronger requirement is to say that we have upper bounds $k(x)$ on the sample entropy that depend *only on x*. The following definition captures this idea, thinking of $k(x) = \log |L(x)|$. (We work with sets rather than sample entropy to avoid paying a $\log(1/\varepsilon)$ loss.)

Definition 7. *Let n be a security parameter and $F : \{0,1\}^n \to \{0,1\}^m$ a function. We say that F^{-1} has* accessible average max-entropy *at most k if for every* PPT *F-collision-finder A, there exists a family of sets $\{L(x)\}_{x \in \mathrm{Supp}(X)}$ and a negligible* *function* $\varepsilon = \varepsilon(n)$ *such that* $x \in L(x)$ *for all* $x \in \mathrm{Supp}(X)$*,* $\mathrm{E}[\log |L(X)|] \leq k$ *and*

$$\Pr\left[A(X;R) \in L(X)\right] \geq 1 - \varepsilon(n),$$

for all sufficiently large n*, where random variable* X *is uniformly distributed on* $\{0,1\}^n$ *and* R *is uniformly random coin tosses for* A*.*

We observe that bounding accessible average max-entropy is indeed stronger than bounding accessible Shannon entropy:

Proposition 3. *If* F^{-1} *has accessible average max-entropy at most* k*, then for every constant* c*,* F^{-1} *has accessible Shannon entropy at most* $k + 1/n^c$*.*

3 Inaccessible Entropy from One-Way Functions

In Section 3.1 we show that *any* one-way function can be very slightly altered into a function with inaccessible entropy. In Section 3.2 we show that an additional hashing step implies a stronger form of inaccessible entropy (which we can then use for a more efficient construction of UOWHF). Still, we find the more direct construction of Section 3.1 and its analysis to be striking in its simplicity.

3.1 A Direct Construction

Theorem 1 (Inaccessible Shannon entropy from one-way functions). *Let* $f\colon \{0,1\}^n \mapsto \{0,1\}^n$ *be a one-way function and define* F *over* $\{0,1\}^n \times [n]$ *as* $F(x,i) = f(x)_{1,\dots,i}$*. Then* F^{-1} *has accessible Shannon entropy at most* $\mathrm{H}(Z|F(Z)) - 1/(2^9 \cdot n^4 \cdot \log^2 n)$*, where* $Z = (X,I)$ *is uniformly distributed over* $\{0,1\}^n \times [n]$*.*[4]

Proof. Suppose on the contrary that there exists a PPT F-collision-finder A such that

$$\mathrm{H}(Z|F(Z)) - \mathrm{H}(A(Z;R)|Z) < \varepsilon = 1/(2^9 \cdot n^4 \cdot \log^2 n)$$

for infinitely many n's, and R is uniformly distributed over the random coins of A. Since I is determined by $F(Z)$, and since Z also determines the second part of A's output (since A is an F-collision-finder), it follows that

$$\mathrm{H}(X|F(Z)) - \mathrm{H}(A'(Z;R)|Z) < \varepsilon$$

where A' is the algorithm that on input $(z;r)$ outputs the first component of $A(z;r)$'s output. In the following use A' to construct an efficient algorithm that inverts f with constant probability. We do so in two steps: 1. Constructing such an inverter under the assumption that we have access to an (inefficient) oracle Sam_{Ideal} defined shortly, and 2. Showing how to efficiently approximate Sam_{Ideal} using A'.

[4] We believe that the actual gap between the real and accessible entropy of F^{-1} is $\Omega(1/n^2)$, or possibly even $\Omega(1/n)$, and not $\Omega(1/n^4)$ as stated. Since even the optimistic $\Omega(1/n)$ bound does not yield as efficient overall construction as the one resulting from Section 3.2, we defer a tighter analysis to the final version of the paper.

Algorithm 1 (Sam_{Ideal})
Input: $x \in \{0,1\}^n$, $i \in [n]$ *and* $b \in \{0,1\}$.

Return a random $x' \in F^{-1}(F(x, i-1))_1$ *such that* $f(x')_i = b$ *(return an arbitrarily value if no such* x' *exists), where* $F^{-1}(F(x,j))_1 = \{x' \in \{0,1\}^n \colon F(x',j) = F(x,j)\}$.

That is, Sam_{Ideal} outputs uniformly at random x' such that $f(x')_{1,\ldots,i} = (f(x)_{1,\ldots,i-1}, b)$. We define an algorithm *Inv* with access to an oracle Sam. When $Sam = Sam_{\mathsf{Ideal}}$, it will be easy to argue that *Inv* inverts f with probability one.

Algorithm 2 (Inv^{Sam})
Input: $y \in \{0,1\}^n$.
Oracle: Sam.

```
For $i = 1$ to $n$ do:
    let $x^i = Sam(x^{i-1}, i, y_i)$ (where $x^0$ is chosen arbitrarily)
Output $x^n$.
```

It is immediate that $Inv^{Sam_{\mathsf{Ideal}}}$ inverts f with probability one. We now turn to showing that A' can be used to efficiently approximate Sam_{Ideal}. The resulting algorithm Sam_δ will be sufficiently similar to Sam_{Ideal} and as a result Inv^{Sam_δ} will still invert f with high probability. A property of *Inv* that will come handy is that, on a uniform value $y = f(x)$, the first coordinate of each individual query that the inverter $Inv^{Sam_{\mathsf{Ideal}}}$ makes (i.e., x^i) is uniform in $\{0,1\}^n$ (the queries are correlated of course).

Recall that the output of A' has high Shannon entropy - almost as high as the uniform distribution over its set of prescribed outputs. Claim 3.1 (which is rather standard), shows that this also implies that the distribution of A''s output is statistically close to this uniform distribution.

Definition 8. *For* $\delta \in [0,1]$ *let* $\mathcal{A}_\delta$ *be the family of efficient* F*-collision-finders with the following guarantee: for every* $A'' \in \mathcal{A}_\delta$ *there exist infinitely many* n*'s such that* $\| \, (Z, A''(Z;R)) - (Z, \mathrm{F}^{-1}(F(Z))_1) \, \| \le \delta$*, where* R *is uniformly distributed over the random-coins of* A'' *and* $\mathrm{F}^{-1}(F(x,i))_1$ *is uniformly distributed over* $F^{-1}(F(x,i))_1$.

Showing that the output of A' is statistically close to uniform can therefore be formalized by showing the following claim:

Claim. $A' \in \mathcal{A}_{\sqrt{\varepsilon}}$.

Proof.

$$\begin{aligned}
\| \, (Z, \mathrm{F}^{-1}(F(Z))_1) - (Z, A'(Z;R)) \, \| &= \mathop{\mathrm{E}}_{z \leftarrow Z}\left[\|\mathrm{F}^{-1}(F(z))_1 - A'(z;R))\|\right] \\
&\le \mathop{\mathrm{E}}_{z \leftarrow Z}\left[\sqrt{\mathrm{H}(\mathrm{F}^{-1}(F(z))_1) - \mathrm{H}(A'(z;R))}\right] \\
&\le \sqrt{\mathop{\mathrm{E}}_{z \leftarrow Z}\left[\mathrm{H}(\mathrm{F}^{-1}(F(z))_1) - \mathrm{H}(A'(z;R))\right]} \\
&= \sqrt{\mathrm{H}(X|F(Z)) - \mathrm{H}(A'(Z;R))} \\
&\le \sqrt{\varepsilon},
\end{aligned}$$

where the first inequality uses the fact that if W is a random variable whose support is contained in a set S and U is the uniform distribution on S, then $\|U - W\| \leq \sqrt{\mathrm{H}(U) - \mathrm{H}(W)}$. (See [2, Lemma 11.6.1].)

As we just shown that $A' \in \mathcal{A}_{\sqrt{\varepsilon}}$ it is enough to show how to use an algorithm $A'' \in \mathcal{A}_\delta$ to approximate Sam_{Ideal} (with error which depends on δ). In order to keep notation simple, we abuse notation and denote by $\mathcal{A}_\delta$ some $A'' \in \mathcal{A}_\delta$. Fix $\delta \in [0,1]$ and consider the following efficient approximation of Sam_{Ideal}:

Algorithm 3 (Sam_δ)
Input: $x \in \{0,1\}^n$, $i \in [n]$ *and* $b \in \{0,1\}$.
Oracle: $\mathcal{A}_\delta$.

```
Repeat $16n \cdot \log n$ times:
    1. Let $x' = \mathcal{A}_\delta(x, i)$
    2. If $f(x')_i = b$, return $x'$.
Abort.
```

Let Inv_δ denote Inv^{Sam_δ} and Inv_{Ideal} denote $Inv^{Sam_{\mathsf{Ideal}}}$. We will show that the output of Inv_δ (on a random value $f(x)$) is statistically close to that of Inv_{Ideal}. As Inv_{Ideal} inverts f with probability one, we will conclude that Inv_δ inverts f with high probability as well. To analyze the statistical distance between the outputs of the two inverters, we consider hybrid inverters that use the ideal Sam_{Ideal} in the first queries and use Sam_δ in the rest of the queries: For $i \in [n+1]$ let Inv_δ^i be the variant of Inv that uses Sam_{Ideal} in the first $i-1$ queries and Sam_δ for the rest of the queries. The next claim will allow us to easily bound the difference between the output distribution of any two neighboring hybrid inverters:

Claim. Let $i \in [n]$ and let $\delta_i = \left\|(X, \mathcal{A}_\delta(X, i; R)) - (X, \mathrm{F}^{-1}(F(X,i))_1)\right\|$, then $\|(X, Sam_{\mathsf{Ideal}}(X, i, f(X)_i)) - (X, Sam_\delta(X, i, f(X)_i))\| \leq 1/2n + 16 \cdot n \cdot \log n \cdot \delta_i$.

Proof. Sam_δ is imperfect for two reasons (which our analysis handles separately). The first reason is that Sam_δ relies on the output of $\mathcal{A}_\delta$ that returns an inverse that is only close to uniform (rather than fully uniform). The error accumulated in each query to $\mathcal{A}_\delta$ is δ_i and there are only $16 \cdot n \cdot \log n$ such queries, which altogether contributes $16 \cdot n \cdot \log n \cdot \delta_i$ to the statistical distance bounded by the claim. The second source of error is that after $16 \cdot n \cdot \log n$ unsuccessful repetitions, Sam_δ aborts without retrieving a correct inverse x'. As we now argue, such failure will only happen with small probability (contributing $\frac{1}{2n}$ to the bound in the claim).

To separate our analysis of the two sources of error, we start by considering the case that $\delta_i = 0$. Note that in this case $\mathcal{A}_\delta(x, i; R) = \mathcal{A}_0(x, i; R)$ is identical to $\mathrm{F}^{-1}(F(x,i))_1$. For $x \in \{0,1\}^n$, $i \in [m]$ and $b \in \{0,1\}$, let $\alpha(x, i, b) := \Pr_{y \xleftarrow{\mathrm{R}} f(X)}[y_i = b \mid y_{1,\ldots,i-1} = f(x)_{1,\ldots,i-1}]$. Note that for every i, $\Pr[\alpha(X, i, f(X)_i)) < \beta] < \beta$ for every $\beta > 0$. We also note that $Sam_\delta(x, i, f(x)_i)$ aborts with probability at most $(1 - \frac{1}{4n})^{16n \cdot \log n} < \frac{1}{4n}$ in the case that $\alpha(x, i, f(x)_i) \geq$

$\frac{1}{4n}$, and that in case it does not abort, (since we assume that $\delta_i = 0$) it returns the same distribution as $Sam_{\mathsf{Ideal}}(x, i, b)$ does. Hence, for the case that $\delta_i = 0$ we have that

$$\begin{aligned}
&\|(X, Sam_{\mathsf{Ideal}}(X, i, f(X)_i)) - (X, Sam_\delta(X, i, f(X)_i))\| \\
&\le \Pr[\alpha(X, i, f(X)_i) < \frac{1}{4n}] + \Pr[Sam_\delta(X, i, f(X)_i) \text{ aborts} \mid \alpha(X, i, f(X)_i) \ge \frac{1}{4n}] \\
&< \frac{1}{4n} + \frac{1}{4n} \\
&\le \frac{1}{2n}.
\end{aligned}$$

We now want to analyze the general case where δ_i may be larger than zero. The statistical distance between the output distribution of $Sam_\delta(X, i, f(X)_i)$ in the case $\delta_i = 0$ and in the general case is at most the maximal number of calls to $\mathcal{A}_\delta$ made by Sam_δ times $\|(X, \mathcal{A}_\delta(X, i)) - (X, \mathcal{A}_0(X, i))\|$, we therefore have that

$$\begin{aligned}
&\|(X, Sam_{\mathsf{Ideal}}(X, i, f(X)_i) - (X, Sam_\delta(X, i, f(X)_i)))\| \\
&\le \frac{1}{2n} + 16n \cdot \log n \cdot \|(X, \mathcal{A}_\delta(X, i) - (X, \mathcal{A}_0(X, i)\| \\
&= \frac{1}{2n} + 16n \cdot \log n \cdot \left\|(X, \mathcal{A}_\delta(X, i) - (X, \mathrm{F}^{-1}(F(X, i))_1))\right\| \\
&= \frac{1}{2n} + 16 \cdot n \cdot \log n \cdot \delta_i.
\end{aligned}$$

Now note that the i'th query of $Inv_\delta^i(f(X))$ (a query to Sam_δ) and the i'th query of $Inv_\delta^{i+1}(f(X))$ (a query to Sam_{Ideal}) are both distributed as $(X, i, f(X)_i)$. Therefore Claim 3.1 yields that for every $i \in [n]$,

$$\left\|Inv_\delta^{i+1}(f(X)) - Inv_\delta^i\right\| \le \frac{1}{2n} + 16 \cdot n \cdot \log n \cdot \delta_i.$$

Hence,

$$\begin{aligned}
&\Pr[Inv_\delta(f(X)) \in f^{-1}(f(X))] \\
&\ge 1 - \sum_{i=1}^{n} \left\|Inv_\delta^{i+1}(f(X)) - Inv_\delta^i(f(X))\right\| \\
&\ge 1 - \sum_{i=1}^{n} \frac{1}{2n} + 16 \cdot n \cdot \log n \cdot \|(X, Sam_{\mathsf{Ideal}}(X, i, f(X)_i) - (X, Sam_\delta(X, i, f(X)_i)\| \\
&\ge \frac{1}{2} - 16 \cdot n^2 \cdot \log n \cdot \delta.
\end{aligned}$$

Let Inv be the instantiation of Inv_δ obtained when we implement Sam_δ using A'. Claim 3.1 yields that $\Pr[Inv(f(X)) \in f^{-1}(f(X))] \ge \Pr[Inv_{\sqrt{\varepsilon}}(f(X)) \in f^{-1}(f(X))] \ge 1/2 - 16 \cdot n^2 \cdot \log n \cdot \sqrt{\varepsilon} > 1/4$.

3.2 A More Efficient Construction

The following theorem shows that a simplified variant of the first step of [11] (which is also the first step of [9]) yields inaccessible entropy with much stronger guarantees than those obtained in Section 3.1. The function we construct is $F(x, g, i) = (g(f(x))_{1,...,i}, g)$, where $g : \{0,1\}^n \rightarrow \{0,1\}^n$ is a three-wise independent function. Since the composition of g and f is still a one-way function then Theorem 1 already implies that F^{-1} has inaccessible entropy. The benefits of the additional hashing step are that 1. we get more inaccessible entropy ($\tilde{\Theta}(1/n)$ bits rather than $\tilde{\Theta}(1/n^4)$ bits), and 2. we get a bound on accessible average max-entropy rather than accessible Shannon entropy. These allow for a simpler and more efficient transformation of F into a UOWHF.

Theorem 2 (Inaccessible average max-entropy from one-way functions). *Let $f\colon \{0,1\}^n \mapsto \{0,1\}^n$ be a one-way function and let $\mathcal{G} = \{g : \{0,1\}^n \rightarrow \{0,1\}^n\}$ be a family of constructible,[5] three-wise independent hash functions. Define F with domain* $\mathrm{Dom}(F) := \{0,1\}^n \times \mathcal{G} \times [n]$ *by*

$$F(x, g, i) = (g(f(x))_{1,...,i}, g).$$

Then, for every constant d, F^{-1} has accessible average max-entropy at most $\mathrm{H}(Z|F(Z)) - (d \log n)/n$ *for every $d > 0$, where $Z = (X, G, I)$ is uniformly distributed over* $\mathrm{Dom}(F)$.

Proof. Let c be a sufficiently large constant (whose value we determine later, depending on the constant d in the theorem statement) and define for every $y \in \{0,1\}^n$ and $i \in [n]$:

$$L(y, i) = \left\{y' \in \{0,1\}^n \colon \ \mathrm{H}_{f(X)}(y') \geq (i + c \cdot \log n) \vee y' = y\right\}.$$

(Recall that the sample entropy is defined as $\mathrm{H}_{f(X)}(y) = \log(1/\Pr[f(X) = y]) = n - \log\left|f^{-1}(y)\right|$, so the “heavy” images, where $f^{-1}(y)$ is large, have low sample entropy.) Namely, $L(y, i)$ consists, in addition to y itself, of “i-light” images with respect to f.

We later show that the sets $L'(x, g, i) = f^{-1}(L(f(x), i)) \times \{(g, i)\}$ satisfy the properties required to show that the accessible max-entropy of F^{-1} is as stated in the theorem.[6] Towards this goal, we first show that the only accessible inputs of F come from preimages of $L(y, i)$.

Claim. For every PPT F-collision-finder A and every constant $c > 0$, it holds that

$$\Pr[A_1(X, G, I; R) \notin f^{-1}(L(f(X), I))] \leq \mathrm{neg}(n),$$

where (X, G, I) is uniformly distributed over $\mathrm{Dom}(F)$, R is uniformly distributed over the random coins of A, and A_1 denotes the first component of A’s output.

[5] $\mathcal{G}$ is *constructible* if given the description of a function $g \in \mathcal{G}$ and $x \in \{0,1\}^n$, $g(x)$ can be computed in time $\mathrm{poly}(n)$, and there is a probabilistic polynomial-time algorithm that given $x \in \{0,1\}^n$, and $y \in \{0,1\}^n$, outputs a random $g \overset{\mathrm{R}}{\leftarrow} \mathcal{G}$ such that $g(x) = y$.

[6] We are working with the set L, and not with L', as it significantly simplifies notations. Note that the sets L' are independent of the adversary, even though the definition of accessible average max-entropy allows the sets to depend on the adversary. Further, note that the sets L are independent of $\mathcal{G}$.

Note that the above immediately yields that $\Pr[A(X,G,I;R) \notin L'(X,G,I)] \leq \mathrm{neg}(n)$, since the other two output components of A are required to equal (g,i), due to the fact that $F(x,g,i)$ determines (g,i).

Proof. Suppose on the contrary that there exist an efficient F-collision-finder A, $c > 0$ and a non-negligible function $\varepsilon = \varepsilon(n)$ such that $\Pr[A_1(X,G,I;R) \notin f^{-1}(L(f(X),I))] \geq \varepsilon$. Fix a triple (x,i,r) and let

$$\varepsilon_{x,i,r} = \Pr[A_1(x,G;r) \notin f^{-1}(L(f(x),i))].$$

Define $A'(g) = A_1(x,g;r)$. We will show how to use any such A' to invert f with probability at least $\varepsilon_{x,i,r}/n^c$. By picking (x,i,r) at random, we will invert f with probability at least $\mathrm{E}_{x,i,r}[\varepsilon_{x,i,r}/n^c] = \varepsilon/n^c$, which contradicts the one-wayness of f. Our inverter works as follows, on input $y \in \{0,1\}^n$.

$Inv(y)$: choose g uniformly at random from $\mathcal{G}$ subject to the constraint $g(y)_{1\cdots i} = g(f(x))_{1\cdots i}$,[7] and output $A'(g)$.

To analyze the success probability Inv, we first rewrite the success probability of A' as follows:

$$\begin{aligned}
\varepsilon_{x,i,r} &\leq \Pr[A'(G) \notin f^{-1}(L(f(x),i)) \\
&= \sum_{y \notin L(f(x),i)} \Pr[A'(G) \in f^{-1}(y)] \\
&= \sum_{y \notin L(f(x),i)} \Pr[G(y)_{1\cdots i} = G(f(x))_{1\cdots i}] \\
&\qquad \cdot \Pr[A'(G) \in f^{-1}(y) | G(y)_{1\cdots i} = G(f(x))_{1\cdots i}] \\
&= 2^{-i} \cdot \sum_{y \notin L(f(x),i)} \Pr[A'(G) \in f^{-1}(y) | G(y)_{1\cdots i} = G(f(x))_{1\cdots i}].
\end{aligned}$$

Above the second equality follows because A is an F-collision finder (so it is always the case that $x' = A'(g) = A(x,g,i)_1$ has the property that $g(f(x'))_{1\cdots i} = g(f(x))_{1\cdots i}$), and the third inequality follows by the two-wise independence of $\mathcal{G}$ ($y \notin L(f(x),i)$ implies that $y \neq f(x)$). Now, we can bound the success probability of Inv in finding a preimage of $Y = f(X)$ by:

$$\begin{aligned}
&\Pr[Inv(Y) \in f^{-1}(Y)] \\
&= \sum_y \Pr[Y = y] \cdot \Pr[A'(G) \in f^{-1}(y) | G(y)_{1\cdots i} = f(x)_{1\cdots i}] \\
&\geq \sum_{y \notin L(f(x),i)} \Pr[Y = y] \cdot \Pr[A'(G) \in f^{-1}(y) | G(y)_{1\cdots i} = f(x)_{1\cdots i}] \\
&\geq \frac{1}{2^{i + c\log n}} \cdot \sum_{y \notin L(f(x),i)} \Pr[A'(G) \in f^{-1}(y) | G(y)_{1\cdots i} = f(x)_{1\cdots i}] \\
&\geq \varepsilon_{x,i,r}/n^c,
\end{aligned}$$

[7] This can be done by first choosing $z \xleftarrow{\mathrm{R}} \{0,1\}^{n-i}$ and then using the constructibility of $\mathcal{G}$ to generate a random g such that $g(y) = (g(f(x))_{1\cdots i}, z)$.

where the penultimate inequality holds because every $y \notin L(f(x), i)$ satisfies $\mathrm{H}_{f(X)}(y) < (i + c \cdot \log n)$.

We have seen that sets $f^{-1}(L(y, i))$ capture the accessible inputs of F; now it remains to show that the expected logarithm of their size is sufficiently smaller than the real entropy $\mathrm{H}(Z|F(Z)) = \mathrm{E}[\log |F^{-1}(F(Z))|]$ (again, this property immediately propagates to L').

Claim. For every constant $c > 8$, it holds that

$$\mathrm{E}\left[\log \left|f^{-1}(L(f(X), I))\right|\right] \leq \mathrm{E}\left[\log \left|F^{-1}(F(Z))\right|\right] - \Omega\left(\frac{c \log n}{n}\right),$$

where $Z = (X, G, I)$ is uniformly distributed in $\mathrm{Dom}(F)$.

Proof. We assume for simplicity that n is a power of 2 (otherwise, we "pad" f) and that c is a power of 2, and let $c' = c/2$. For ease of notation, we will work in entropy units of $c' \log n$. Namely, for $i \in \{0, \cdots, m = n/(c' \log n)\}$ and $y \in \{0,1\}^n$, let $y_{\{1\},\ldots,\{i\}}$ be the first $i \cdot c' \log n$ bits of y, define

$$\mathrm{H}_f(y) := \frac{\mathrm{H}_{f(X)}(y)}{c' \log n}.$$

and let

$$q_i = \Pr[\mathrm{H}_f(f(X)) \in [i, i+1)].$$

Recall that (X, G, I) is uniformly distributed in $\mathrm{Dom}(F)$. We define additional random variables that categorize the "non trivial collisions" induced by F into two separate categories:

$$\begin{aligned}\text{Light} := |\{x' \in \{0,1\}^n \colon f(x') \neq f(X) \wedge G(f(x'))_{\{1\},\ldots,\{I\}} = G(f(X))_{\{1\},\cdots,\{I\}} \\ \wedge\, \mathrm{H}_f(f(x')) \geq I + 2\}|.\end{aligned}$$

Namely, Light consists of the preimages that collide with $f(X)$, different from $f(X)$, and "light" — have few preimages. Similarly, let

$$\begin{aligned}\text{Heavy} := |\{x' \in \{0,1\}^n \colon f(x') \neq f(X) \wedge G(f(x'))_{\{1\},\ldots,\{I\}} = G(f(X))_{\{1\},\cdots,\{I\}} \\ \wedge\, \mathrm{H}_f(f(x')) < I + 2\}|.\end{aligned}$$

Namely, Heavy consists of the preimages that collide with $f(X)$, different from $f(X)$, and "heavy" — have many preimages. Note that

$$\left|F^{-1}(F(Z))\right| = \text{Light} + \text{Heavy} + |f^{-1}(f(X))|$$

(recall that the all elements $F^{-1}(F(x, g, i))$ are of the form (x', g, i)) and

$$\left|f^{-1}(L(f(X), I))\right| \leq \text{Light} + |f^{-1}(f(X))|.$$

Thus, we have

$$\begin{aligned}&\mathrm{E}[\log|F^{-1}(F(Z))|] - \mathrm{E}[\log|f^{-1}(L(f(X),I))|] \\ &\geq \mathrm{E}\left[\log\frac{\mathrm{Light}+\mathrm{Heavy}+|f^{-1}(f(X))|}{\mathrm{Light}+|f^{-1}(f(X))|}\right]\end{aligned} \tag{1}$$

We manipulate this as follows:

$$\begin{aligned}&\mathrm{E}\left[\log\frac{\mathrm{Light}+\mathrm{Heavy}+|f^{-1}(f(X))|}{|f^{-1}(f(X))|+\mathrm{Light}}\right] \\ &\geq \mathrm{E}\left[\log\left(1+\frac{\mathrm{Heavy}}{|f^{-1}(f(X))|+\mathrm{Light}+\mathrm{Heavy}}\right)\right] \\ &\geq \mathrm{E}\left[\frac{\mathrm{Heavy}}{|f^{-1}(f(X))|+\mathrm{Light}+\mathrm{Heavy}}\right],\end{aligned} \tag{2}$$

where the last inequality uses the fact that $\log(1+\alpha) \geq \alpha$ for $\alpha \leq 1$. The proof of Claim 3.2 easily follows from the next claim, which yields that with constant probability, Heavy is a significant term in $(|f^{-1}(f(X))| + \mathrm{Light} + \mathrm{Heavy})$.

Claim. Let $\alpha \geq 1,\ i \in \{0,\ldots,m-1\}$ and $x \in \{0,1\}^n$. Condition on $I = i$ and $X = x$, and define the following events (over the random variable G):

$$\begin{aligned}E_i^1 &: (\mathrm{Light}+\mathrm{Heavy}) \leq 3\cdot 2^{n-i\cdot(c'\log n)} \\ E_i^2 &: \mathrm{Heavy} \geq (q_{i+1} - \alpha\cdot\sqrt{1/n^{c'}})\cdot 2^{n-i\cdot c'\log n-1}\end{aligned}$$

Then $\Pr[E_i^1] \geq 2/3$, and $\Pr[E_i^2] \geq 1-4/\alpha^2$.

Proof. For E_i^1, we note that $\mathrm{E}[\mathrm{Light}+\mathrm{Heavy}] \leq 2^{n-i\cdot(c'\log n)}$ by two-universality of $\mathcal{G}$, and apply Markov's Inequality.

For E_i^2, let

$$S := \{x' \in \{0,1\}^n \colon f(x') \neq f(x) \wedge \mathrm{H}_f(f(x')) \in [i+1, i+2)\}.$$

Note that $|S| \geq (q_{i+1} - \mathrm{neg}(n))\cdot 2^n$, where we subtract $\mathrm{neg}(n)$ for not taking into account the preimages of $f(x)$. For $g \in \mathcal{G}$, let

$$\begin{aligned}S_g := \{x' \in \{0,1\}^n \colon & f(x') \neq f(x) \wedge \mathrm{H}_f(f(x')) \in [i+1,i+2) \\ & \wedge g(f(x'))_{\{1\},\ldots,\{i\}} = g(f(x))_{\{1\},\cdots,\{i\}}\},\end{aligned}$$

note that (conditioned on $I = i$ and $X = x$) $\mathrm{Heavy} \geq |S_G|$. We write $|S_g| = \sum_{y\in f(S)} 1_{g,y}\cdot|f^{-1}(y)|$, where $1_{g,y}$ is the indicator for $g(y)_{\{1\},\ldots,\{i\}} = g(f(x))_{\{1\},\cdots,\{i\}}$. By the three-wise independence of $\mathcal{G}$, the $1_{G,y}$'s are pairwise independent Bernoulli random variables, each with expectation $2^{-i\cdot c'\log n}$. Thus, $\mathrm{E}[|S_G|] \geq (q_{i+1} - \mathrm{neg}(n))\cdot 2^{n-i\cdot c'\log n}$. Assuming that $q_{i+1} > \alpha\cdot\sqrt{1/n^{c'}} \geq \sqrt{1/n^{c'}}$ (as otherwise the claim about E_i^2 holds trivially), it follows that

$$\mathrm{E}[|S_G|] > q_{i+1}\cdot 2^{n-i\cdot c'\log n-1}$$

By the pairwise independence of $1_{G,y}$'s, we also have

$$\begin{aligned}
\mathrm{Var}[|S_G|] &= \sum_{y\in f(S)} \mathrm{Var}[1_{G,y}\cdot |f^{-1}(y)|] \\
&\leq 2^{-i\cdot c'\log n}\cdot \sum_{y\in f(S)} \left|f^{-1}(y)\right|^2 \\
&\leq 2^{-i\cdot c'\log n}\cdot |S|\cdot \max_{y\in f(S)} |f^{-1}(y)| \leq 2^{-i\cdot c'\log n}\cdot 2^n\cdot 2^{n-(i+1)\cdot c'\log n} \\
&= \left(\frac{2}{\sqrt{n^{c'}}}\cdot 2^{n-i\cdot c'\log n-1}\right)^2,
\end{aligned}$$

and thus by Chebyshev inequality

$$\begin{aligned}
\Pr[E_i^2] &\geq \Pr\left[|S_G| \geq (q_{i+1} - \alpha\cdot\sqrt{1/n^{c'}})\cdot 2^{n-i\cdot c'\log n-1}\right] \\
&\geq 1 - \Pr\left[\big||S_G| - \mathrm{E}[|S_G|]\big| \geq \frac{\alpha}{2}\cdot\sqrt{\mathrm{Var}[|S_G|]}\right] \geq 1 - \frac{4}{\alpha^2}.
\end{aligned}$$

Noting that $\mathrm{H}_f(f(X)) \geq i$ means $|f^{-1}(f(X))| \leq 2^{n-i\cdot(c'\log n)}$, and applying Claim 3.2 with $\alpha = 4$, we have

$$\begin{aligned}
&\mathrm{E}[\log|F^{-1}(F(Z))|] - \mathrm{E}[\log\left|f^{-1}(L(Y,I))\right|] \\
&\geq \mathrm{E}\left[\frac{\mathrm{Heavy}}{|f^{-1}(f(X))| + \mathrm{Light} + \mathrm{Heavy}}\right] \\
&\geq \frac{1}{m}\cdot\sum_{i=0}^{m-1} \Pr[\mathrm{H}_f(f(X)) \geq i]\cdot \Pr[E_i^1 \wedge E_i^2 \mid \mathrm{H}_f(f(X)) \geq i] \\
&\quad\cdot\ \frac{\left(q_{i+1} - \frac{4}{n^{c'/2}}\right)\cdot 2^{n-i\cdot c'\log n-1}}{2^{n+2-i\cdot(c'\log n)}} \\
&\geq \frac{1}{m}\cdot\sum_{i=0}^{m-1}(q_{i+1}+\cdots+q_m)\cdot\left(1-\frac{1}{3}-\frac{1}{4}\right)\cdot\left(\frac{q_{i+1}-4/n^{c'/2}}{8}\right) \\
&\geq \frac{1}{48m}\cdot\left(\sum_{j,i\in\{0,\ldots,m\}} q_i\cdot q_j\right) - O\left(\frac{m}{n^{c'/2}}\right) \\
&\geq \frac{1}{48m} - O\left(\frac{m}{n^{c'/2}}\right),
\end{aligned}$$

where the first inequality is by Equation 2, and the third inequality holds since $q_0 = 0$ for every one-way function, which implies that $\sum_{1\leq i\leq j\leq m} q_i\cdot q_j = \sum_{0\leq i\leq j\leq m} q_i\cdot q_j \geq \frac{1}{2}\cdot\sum_{j,i\in\{0,\ldots,m\}} q_i\cdot q_j$. Thus, Claim 3.2 holds with respect to any $c = 2c' \geq 8$.

By Claims 3.2 and 3.2 and the fact that $F(x,g,i)$ determines g and i, the sets $L'(x,g,i) = f^{-1}(L(f(x),i)) \times \{(g,i)\}$ satisfy the properties required to show that the accessible max-entropy of F^{-1} is at most $\mathrm{H}(Z|F(Z)) - \Omega(c(\log n)/n)$. Taking c to be a sufficiently large constant times d, completes the proof.

4 UOWHFs from Inaccessible Entropy

In this section we show how to construct a UOWHF from any efficiently computable function with a noticeable gap between real Shannon entropy and either accessible average max-entropy or accesssible Shannon entropy. Recall that the more efficient construction from Section 3.2 satisfies the former, and the more direct construction from Section 3.1 satisfies the latter. Combined with these constructions, we obtain two new constructions of UOWHFs from any one-way function.

In both cases, we first transform the entropy gap into a noticeable gap between real Shannon entropy and accessible *max*-entropy. We begin with the construction that starts from a gap between real Shannon entropy and accessible average max-entropy because the transformation involves fewer steps (and is also more efficient).

4.1 The More Efficient UOWHF

Starting with any one-way function $f : \{0,1\}^n \to \{0,1\}^n$, the final UOWHF has output length $O(n^7)$ and key length $\tilde{O}(n^7)$ (or $O(n^5)$ for both output and key lengths for a non-uniform construction). Throughout, let $s \in \omega(\log n)$ denote any super-logarithmic function.

STEP 0 **(basic construction):** Let F_0 denote the function from Section 3.2. That is, F_0 is defined on domain $\{0,1\}^n \times \mathcal{G} \times [n]$ as $F_0(x,g,i) = (g(f(x))_{1,\dots,i}, g)$, where $f\colon \{0,1\}^n \mapsto \{0,1\}^n$ is a one-way function and $\mathcal{G}$ is a family of constructible, 3-wise independent hash functions over $\{0,1\}^n$.

- $F_0 : \{0,1\}^{\ell_0^{\text{IN}}} \to \{0,1\}^{\ell_0^{\text{OUT}}}$ where $\ell_0^{\text{IN}} = \ell_0^{\text{IN}}(n) = O(n)$ and $\ell_0^{\text{OUT}} = \ell_0^{\text{OUT}}(n) = O(n)$.
- Let k_{REAL} denote the real Shannon entropy of F_0^{-1}. Theorem 2 yields that the accessible average max-entropy of F_0^{-1} is bounded by $k_{\text{REAL}} - \Delta$ for $\Delta = (\log n)/n$.

STEP 1 **(gap amplification):** Let F_1 be the t-fold direct product of F_0. That is, $F_1(x_1,\dots,x_t) = (F_0(x_1),\dots,F_0(x_t))$ where $t \in O(n^2 s/\Delta^2)$. Specifically, we require that

$$tk_{\text{REAL}} - \ell_0^{\text{IN}} \cdot \sqrt{st} \geq t \cdot (k_{\text{REAL}} - \Delta/2) + \ell_0^{\text{IN}} \cdot \sqrt{st} + 3s.$$

This repetition increases both the real and accessible entropies of F_1 by a factor of t (comparing to F_0). In addition, this repetition converts real Shannon entropy to real min-entropy and accessible average max-entropy to accessible max-entropy (up to additive terms that are sub-linear in t). More precisely:

- $F_1 : \{0,1\}^{\ell_1^{\text{IN}}} \to \{0,1\}^{\ell_1^{\text{OUT}}}$ where $\ell_1^{\text{IN}}(n) = t \cdot \ell_0^{\text{IN}} = O(tn)$ and $\ell_1^{\text{OUT}}(n) = t \cdot \ell_0^{\text{OUT}} = O(tn)$.
- F_1^{-1} has real min-entropy at least $tk_{\text{REAL}} - \ell_0^{\text{OUT}} \cdot \sqrt{st}$, which by our choice of t is at least $t \cdot (k_{\text{REAL}} - \Delta/2) + \ell_0^{\text{IN}} \cdot \sqrt{st} + 3s$.
- F_1^{-1} has accessible max-entropy at most $t \cdot (k_{\text{REAL}} - \Delta) + \ell_0^{\text{IN}} \cdot \sqrt{st}$.

From the next step on, the construction is given an additional parameter k (a "good" estimate of k_{REAL}) such that $k \in [k_{\text{REAL}}, k_{\text{REAL}} + \varDelta/2]$. This means that:

- F_1^{-1} has real min-entropy at least $t \cdot (k - \varDelta) + \ell_0^{\text{OUT}} \cdot \sqrt{st} + 3s$.
- F_1^{-1} has accessible max-entropy at most $t \cdot (k - \varDelta) + \ell_0^{\text{OUT}} \cdot \sqrt{st}$.

That is, there is a gap of $3s$ between real min-entropy and accessible max-entropy, and moreover, we "know" where the gap is (given k).

STEP 2 **(entropy reduction):** Apply entropy reduction to F_1 to obtain F_2. That is, $F_2(x, g) = (F_1(x), g, g(x))$, where $g : \{0,1\}^{\ell_1^{\text{IN}}} \to \{0,1\}^{\ell}$ is selected from a family of 2-universal hash functions, where $\ell = \ell(n,k) = t \cdot (k - \varDelta) + \ell_0^{\text{IN}} \cdot \sqrt{st} + s = O(tn)$. This additional hashing reduces the real min-entropy and accessible max-entropy by ℓ (up to an additive term of s). More precisely,

- $F_2 : \{0,1\}^{\ell_2^{\text{IN}}} \to \{0,1\}^{\ell_2^{\text{OUT}}}$ where $\ell_2^{\text{IN}}(n,k) = O(tn)$ and $\ell_2^{\text{OUT}}(n,k) = O(tn)$.
- F_2^{-1} has real min-entropy at least s.
- F_2^{-1} has accessible max-entropy at most 0. Hence, F_2 is collision-resistant on random inputs (by Lemma 1).

STEP 3 **(reducing the output length):** First reduce the output length of F_2 by hashing the output to $\ell_2^{\text{IN}} - \log n$ bits. That is, $F_3(x, g) = (g, g(F_3(x)))$ where $g : \{0,1\}^{\ell_2^{\text{OUT}}} \to \{0,1\}^{\ell_2^{\text{IN}} - \log n}$ is selected from a family of pairwise-independent hash functions.

- $F_3 : \{0,1\}^{\ell_3^{\text{IN}}} \to \{0,1\}^{\ell_3^{\text{OUT}}}$ where $\ell_3^{\text{IN}}(n,k) = O(tn)$ and $\ell_3^{\text{OUT}}(n,k) = \ell_3^{\text{IN}} - \log n$.
- F_3 remains collision-resistant on random inputs.

Next, transform F_3 into a family $\{F_y\}$ of target collision-resistant hash functions via a random shift. That is, $F_y(x) = F_3(y + x)$.

- This yields a non-uniform construction $\{F_y\}$ with input length and key length $\ell_3^{\text{IN}}(n,k) = O(tn) = O(n \cdot n^2 s/\varDelta^2) = O(n^5)$, where the non-uniformity corresponds to choice of the parameter $k \in [k_{\text{REAL}}, k_{\text{REAL}} + \varDelta/2]$.

STEP 4 **(removing non-uniformity):** To remove the non-uniform advice k, we "try all possibilities" from 0 to $\ell_0^{\text{IN}}(n)$ in steps of size $\varDelta/2$, similar to the approach used in [11] (see also [9, Section 3.6]) :

i. First, we construct $m = \ell_0^{\text{IN}}(n) \cdot 2/\varDelta$ families of functions $\{F_y^i\}$ for $i = 1, 2, \ldots, m$, where $\{F_y^i\}$ is the family of functions obtained by instantiating Steps 1 through 3 with the parameter k set to the value $i\varDelta/2$. This m families of functions satisfy the following properties:
   - Each of $F_y^1, \ldots, F_y^m$ is length-decreasing; in particular, F_y^i has input length $\ell_3^{\text{IN}}(n, i\varDelta/2)$ and output length $\ell_3^{\text{IN}}(n, i\varDelta/2) - \log n$. Note that $\ell_3^{\text{IN}}(n, i\varDelta/2) \leq \ell_3^{\text{IN}}(n, \ell_0^{\text{IN}}(n))$ for all i because $\ell(n,k)$ increases as a function of k. We may then assume that all m functions $F_y^1, \ldots, F_y^m$ have the same input length $\ell_3^{\text{IN}}(n, \ell_0^{\text{IN}}(n))$ and the same output length $\ell_3^{\text{IN}}(n, \ell_0^{\text{IN}}(n)) - \log n$ by padding "extra part" of the input to the output.
   - At least one of $\{F_y^1\}, \ldots, \{F_y^m\}$ is target collision-resistant; this is because $k_{\text{REAL}} \in [0, \ell_0^{\text{IN}}(n)]$ so there exists some i for which $i\varDelta/2$ lies between k_{REAL} and $k_{\text{REAL}} + \varDelta/2$.

ii. Next, for each $i = 1, 2, \ldots, m$, we construct a family of functions $\{\tilde{F}^i_{\tilde{y}}\}$ from $\{F^i_y\}$ with input length $m \cdot \ell^{\text{IN}}_3(n, \ell^{\text{IN}}_0(n))$, key length $O(\ell^{\text{IN}}_3(n, \ell^{\text{IN}}_0(n)) \cdot \log n)$ and output length $\ell^{\text{IN}}_3(n, \ell^{\text{IN}}_0(n)) - \log n$, by following the construction given in [13]. Again, at least one of $\{\tilde{F}^1_{\tilde{y}_1}\}, \ldots, \{\tilde{F}^m_{\tilde{y}_m}\}$ is target collision-resistant.

iii. Finally, we define a family of functions $\{F_{\tilde{y}_1,\ldots,\tilde{y}_m}\}$ to be the concatenation of $\tilde{F}^1_{\tilde{y}_1}, \ldots, \tilde{F}^m_{\tilde{y}_m}$ on the same input. That is, $F_{\tilde{y}_1,\ldots,\tilde{y}_m}(x) = \tilde{F}^1_{\tilde{y}_1}(x) \circ \cdots \circ \tilde{F}^m_{\tilde{y}_m}(x)$.
   - Note that F has input length $m \cdot \ell^{\text{IN}}_3(n, \ell^{\text{IN}}_0(n))$ and output length $m \cdot (\ell^{\text{IN}}_3(n, \ell^{\text{IN}}_0(n)) - \log n)$, so F is length-decreasing.
   - Moreover, since at least one of $\{\tilde{F}^1_{\tilde{y}_1}\}, \ldots, \{\tilde{F}^m_{\tilde{y}_m}\}$ is target collision-resistant, $\{F_{\tilde{y}_1,\ldots,\tilde{y}_m}\}$ must also be target collision-resistant. This is because a collision for $F_{\tilde{y}_1,\ldots,\tilde{y}_m}$ is a collision for each of $\tilde{F}^1_{\tilde{y}_1}, \ldots, \tilde{F}^m_{\tilde{y}_m}$.

This yields a uniform construction of a UOWHF with output length length $O(n/\Delta \cdot n \cdot n^2 s/\Delta^2) = O(n^7)$. and key length $\tilde{O}(n/\Delta \cdot n \cdot n^2 s/\Delta^2 \cdot \log n) = \tilde{O}(n^7)$.

4.2 UOWHF via a Direct Construction

Here, the final construction has output length $\tilde{O}(n^{36})$ (or $\tilde{O}(n^{26})$ for a non-uniform construction).

STEP 0 **(basic construction):** Let F_0 denote the function from Section 3.1. That is, F_0 is defined over $\{0,1\}^n \times [n]$ as $F_0(x, i) = (f(x)_{1,\ldots,i})$, where $f\colon \{0,1\}^n \mapsto \{0,1\}^n$ is a one-way function.
   - $F_0 : \{0,1\}^{\ell^{\text{IN}}_0} \to \{0,1\}^{\ell^{\text{OUT}}_0}$ where $\ell^{\text{IN}}_0 = \ell^{\text{IN}}_0(n) \leq 2n$ and $\ell^{\text{OUT}}_0 = \ell^{\text{OUT}}_0(n) \leq 2n$.
   - Let k_{REAL} denote the real Shannon entropy of F_0^{-1}. Theorem 1 yields that the accessible Shannon entropy of F_0^{-1} is at most $k_{\text{REAL}} - \Delta$, where $\Delta \in \Omega(1/n^4 \cdot \log^2 n)$

STEP 1 **(gap amplification):** Let F_1 be the t-fold direct product of F_0 for a sufficiently large t to be determined later. That is, $F_1(x_1, \ldots, x_t) = (F_0(x_1), \ldots, F_0(x_t))$.
This repetition increases both the real and accessible entropies of F_1 by a factor of t. In addition, the repetition converts real Shannon entropy to real min-entropy and real max-entropy (up to an additive $o(t)$ term). More precisely:
   - $F_1 : \{0,1\}^{\ell^{\text{IN}}_1} \to \{0,1\}^{\ell^{\text{OUT}}_1}$ where $\ell^{\text{IN}}_1(n) = t \cdot \ell^{\text{IN}}_0 = O(tn)$ and $\ell^{\text{OUT}}_1(n) = t \cdot \ell^{\text{OUT}}_0 = O(tn)$.
   - F_1^{-1} has real min-entropy at least $t \cdot k_{\text{REAL}} - 2n\sqrt{st}$ and real max-entropy at most $t \cdot k_{\text{REAL}} + 2n\sqrt{st}$.
   - F_1^{-1} has accessible Shannon entropy at most $t \cdot k_{\text{REAL}} - t\Delta$.

From the next step on, the construction is given an additional parameter k (a “good” estimate of k_{REAL}) such that $k \in [k_{\text{REAL}}, k_{\text{REAL}} + \Delta^2/256n]$. This means that:
   - F_1^{-1} has accessible Shannon entropy at most $tk - t\Delta$. This means F_1^{-1} has $(1 - \Delta/4k)$-accessible max-entropy at most $tk - t\Delta/2$.

STEP 2 **(entropy reduction):** Apply entropy reduction to F_1 with $\ell = \ell(n,k) = tk - t\Delta/2 + s$ to obtain F_2. That is, $F_2(x,g) = (F_1(x), g, g(x))$, where $g : \{0,1\}^{\ell_1^{\text{IN}}} \to \{0,1\}^{\ell}$ is selected from a family of 2-universal hash functions.

This reduces the accessible max-entropy to 0, which allows us to deduce that F_2 is weakly collision-resistant on random inputs.

- $F_2 : \{0,1\}^{\ell_2^{\text{IN}}} \to \{0,1\}^{\ell_2^{\text{OUT}}}$ where $\ell_2^{\text{IN}}(n,k) = O(tn + \ell(n,k)) = O(tn)$ and $\ell_2^{\text{OUT}}(n,k) = O(tn + \ell(n,k)) = O(tn)$.
- F_2^{-1} has real min-entropy at least $t(k_{\text{REAL}} - k + \Delta/2) - 2n\sqrt{st} - 2s$ and real max-entropy at most $t(k_{\text{REAL}} - k + \Delta/2) + 2n\sqrt{st}$.
- F_2^{-1} has $(1 - \Delta/4k + 2^{-\Omega(s)})$-accessible max-entropy at most 0. Thus, F_2 is q-collision-resistant on random inputsfor $q = \Delta/4k - 2^{-\Omega(s)}$.

STEP 3 **(gap amplification):** F_3 is t'-fold direct product of F_2, where $t' = s/q = O(ks/\Delta) = \tilde{O}(n^5)$. That is, $F_3(x_1, \ldots, x_{t'}) = (F_2(x_1), \ldots, F_2(x_{t'}))$.

This allows us to amplify the weak collision-resistance property of F_2 to obtain a gap between real min-entropy and accessible max-entropy in F_3.

- F_3^{-1} has real min-entropy at least $t' \cdot \big(t(k_{\text{REAL}} - k + \Delta/2) - 2n\sqrt{st} - 2s\big)$, which is at least:
$$t' \cdot \big(t(\Delta/2 - \Delta^2/256n) - 2n\sqrt{st} - 2s\big).$$
- F_3^{-1} has accessible max-entropy at most $t' \cdot \big((1 - q/8)(t(k_{\text{REAL}} - k + \Delta/2) + 2n\sqrt{st}) + 1\big)$, which is at most:
$$t' \cdot \big(t(\Delta/2 - \Delta q/16) + 2n\sqrt{st}) + 1\big).$$
Now, $k \le \ell_0^{\text{IN}}(n) \le 2n$, so $q = \Delta/4k - 2^{-\Omega(s)} \ge \Delta/8n - 2^{-\Omega(s)}$. This means F_3^{-1} has accessible max-entropy at most:
$$t' \cdot \big(t(\Delta/2 - \Delta^2/128n + 2^{-\Omega(s)}) + 2n\sqrt{st}) + 1\big).$$

Note that the gap is at least $t' \cdot \big(t \cdot \Delta^2/256n - 2^{-\Omega(s)} - (4n\sqrt{st} + 2s + 1)\big)$, which is at least $3s$ as long as:
$$t \cdot \Delta^2/256n \ge 2^{-\Omega(s)} + 4n\sqrt{st} + 2s + 1 + 3s/t'$$
Since $3s/t' = 3q \le 3\Delta$, we can set $t = O(n/\Delta + ns/\Delta^2 + n^4 s/\Delta^4) = \tilde{O}(n^{20})$ so that F_3^{-1} has a gap of $3s$ between real min-entropy and accessible max-entropy, and moreover, we know where this gap is (given k).

STEPS 4/5/6: We follow steps 2, 3, and 4 in the previous construction, with the following modifications in the parameters:

- We apply entropy reduction first, with
$$\ell = t' \cdot \big(t(\Delta/2 - \Delta q/16) + 2n\sqrt{st}) + 1\big) + s.$$
- To remove the non-uniform advice k, we "try all possibilities" from 0 to $\ell_0^{\text{IN}}(n) \le 2n$ in steps of size $\Delta^2/256n$.

We then obtain a non-uniform construction of UOWHFs with output length $O(n \cdot t \cdot t') = \tilde{O}(n^{26})$ and a uniform construction with output length $O(n/(\Delta^2/n) \cdot n \cdot t \cdot t' \cdot \log n) = \tilde{O}(n^{36})$.

Acknowledgements

We are thankful to Ran Raz and Chiu-Yuen Koo for useful conversations.

References

[1] Canetti, R., Rivest, R.L., Sudan, M., Trevisan, L., Vadhan, S.P., Wee, H.: Amplifying collision resistance: A complexity-theoretic treatment. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 264–283. Springer, Heidelberg (2007)

[2] Cover, T.M., Thomas, J.A.: Elements of information theory, 2nd edn. Wiley-Interscience, New York (2006)

[3] Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing 33(1), 167–226 (2003) (electronic)

[4] Haitner, I., Nguyen, M., Ong, S.J., Reingold, O., Vadhan, S.: Statistically hiding commitments and statistical zero-knowledge arguments from any one-way function. SIAM Journal on Computing 39(3), 1153–1218 (2009)

[5] Haitner, I., Reingold, O., Vadhan, S., Wee, H.: Inaccessible entropy. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC). ACM Press, New York (2009)

[6] Haitner, I., Reingold, O., Vadhan, S.: Efficiency improvements in constructions of pseudorandom generators. In: Proceedings of the 42th Annual ACM Symposium on Theory of Computing (STOC). ACM Press, New York (2010)

[7] Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing 28(4), 1364–1396 (1999); Preliminary versions in STOC 1989 and STOC 1990

[8] Impagliazzo, R., Luby, M.: One-way functions are essential for complexity based cryptography. In: Proceedings of the 30th Annual Symposium on Foundations of Computer Science (FOCS), pp. 230–235 (1989)

[9] Katz, J., Koo, C.: On constructing universal one-way hash functions from arbitrary one-way functions. Technical Report 2005/328, Cryptology ePrint Archive (2005)

[10] Naor, M., Yung, M.: Universal one-way hash functions and their cryptographic applications. In: Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC), pp. 33–43. ACM Press, New York (1989)

[11] Rompel, J.: One-way functions are necessary and sufficient for secure signatures. In: Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (STOC), pp. 387–394 (1990)

[12] Rompel, J.: Techniques for computing with low-independence randomness. PhD thesis, Massachusetts Institute of Technology (1990), http://dspace.mit.edu/handle/1721.1/7582

[13] Shoup, V.: A composition theorem for universal one-way hash functions. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 445–452. Springer, Heidelberg (2000)

Constant-Round Non-malleable Commitments from Sub-exponential One-Way Functions

Rafael Pass[1,⋆] and Hoeteck Wee[2,⋆⋆]

[1] Cornell University
[2] Queens College, CUNY

Abstract. We present a constant-round non-malleable commitment scheme based on the existence of sub-exponential one-way functions and using a black-box proof of security. As far as we know, this is the first construction of a constant-round non-malleable protocol based on only one-wayness, or to admit a black-box proof of security under any standard-type assumption.

Keywords: commitment schemes, non-malleability, complexity leveraging.

1 Introduction

We consider the execution of two-party protocols in the presence of an adversary that has full control of the communication channel between the parties. The adversary may omit, insert, or modify messages at will. The honest parties are not necessarily aware of the existence of the adversary, and not use any kind of trusted set-up (such as a common reference string). The above kind of attack is often referred to as a *man-in-the-middle* attack. Protocols that are secure against such attacks are said to be *non-malleable*, and were first studied in the seminal work of Dolev, Dwork and Naor [6]. Due to the hostile environment in which they operate, the design and analysis of non-malleable protocols is a notoriously difficult task. The task becomes even more challenging if the honest parties are not allowed to use any kind of trusted set-up. Indeed, only a handful of such protocols have been constructed so far.

In their paper, Dolev et al. presented non-malleable protocols for the tasks of commitment and zero-knowledge. The protocols rely on the existence of one-way functions, and require $O(\log n)$ rounds of interaction, where n is a security parameter. More recently, Barak [2] presented the first constant-round non-malleable protocols for commitment and zero-knowledge whose security relies on the existence of trapdoor permutations and collision-resistant hash functions with sub-exponential hardness. The result was subsequently improved by Pass and Rosen [24], who obtained constant-round protocols assuming only collision-resistant hash functions with standard hardness.

[⋆] Supported in part by a Microsoft New Faculty Fellowship, NSF CAREER Award CCF-0746990, AFOSR Award FA9550-08-1-0197, and BSF Grant 2006317.

[⋆⋆] Most of this work was done while a post-doc at Columbia University; supported in part by NSF Grants CNS-0716245, SBE-0245014, NSF CAREER Award CNS-0953626 and PSC-CUNY Award # 6014939 40.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 638–655, 2010.

There has been a series of follow-up work on non-malleable commitments [25, 16, 21, 20, 15], but none of which reduces the assumptions in [24] for constant-round non-malleable commitments. This raises the following natural question:

> *What are the minimal assumptions under which we can construct constant-round non-malleable commitment schemes? Specifically, is one-wayness alone sufficient to construct constant-round non-malleable commitment schemes?*

1.1 Our Results

In this work, we address the above question. Our main result is that one-wayness alone—with sub-exponential hardness—suffices for constructing constant-round non-malleable commitments.

> **Main Theorem (informal):** Suppose there exists one-way functions secure against sub-exponential size circuits. Then, there exists a constant-round non-malleable commitment scheme.

We note that while all known candidates of one-way functions are conceivably also secure against sub-exponential size circuits, there are several natural candidates which do not appear to yield collision-resistant hash functions. Our result should be compared with the very recent work of Lin and Pass [15], which gave a $O(1)^{\log^* n}$-round non-malleable commitment schemes under the minimal assumption of one-way functions with standard (super-polynomial) hardness. Comparing the two, our result may be viewed as offering a new trade-off between round complexity and quantitative hardness assumptions. As with [15, 25, 16, 21], our commitment scheme achieves a very strong notion of non-malleability—that of concurrent non-malleability—which guarantees independence of the committed values even when multiple executions of the commitment schemes are executed at the same time. Before providing further details about our construction, we provide some additional context and applications.

On black-box proofs of security. While the original [6] construction only relies on "elementary" techniques and has a black-box proof of security, basically all constant-round non-malleable commitment schemes rely on *non-black-box* simulation techniques [1] and inherit the sophisticated machinery (e.g. the PCP theorem) associated with them, along with the need for qualitatively stronger assumptions (that of collision-resistant hash functions). As such, the problem of reducing the cryptographic assumptions for constant-round non-malleable commitment schemes appears to be intimately related to the question of whether non-black-box techniques are necessary for constructing constant-round non-malleable protocols, without resorting to non-standard assumptions (c.f. [21][1]).

[1] Pandey, Pass and Vaikuntanathan constructed non-interactive non-malleable commitment schemes assuming the existence of, so called, adaptive one-way permutations – namely permutations which remain one-way even when the adversary has access to an inversion oracle. Note that this assumption has a strong non-malleability flavor and as such provide limited insight into realizing non-malleability "from scratch".

Understanding the power and limitations of black-box techniques has been an important goal in the foundations of cryptography, starting from [13]. For the usage of a primitive in cryptographic constructions, a recent line of work has narrowed the gap between what can be achieved using black-box and non-black-box techniques. On the other hand, for usage of the adversary's code in the proof of security, we do know for a fact that non-black-box techniques are inherently more powerful, as evidenced by the works on constant-round public-coin zero-knowledge protocols [1, 9]. A natural question is whether such a separation extends beyond the realm of zero knowledge. Given the state-of-the-art for non-malleability, it is tempting to conjecture that such a separation extends also to constant-round non-malleable commitment schemes. Our construction refutes such a conjecture since it admits a black-box proof of security (which is to be expected since we do not require collision-resistant hash functions).

On constant-round secure multi-party computation. The early work of Goldreich, Micali and Wigderson [10] showed that we may realize secure multi-party computation in the presence of a dishonest majority assuming the existence of enhanced trapdoor permutations, where the round complexity of the protocol grows linearly with the number of parties.[2] Subsequent improvements by Katz, Ostrovsky and Smith [14] (relying on [2]) and Pass [22] culminated a constant-round protocol, assuming in addition the existence of collision-resistant hash functions. As with previous constant-round non-malleable protocols, both of these constructions exploit non-black-box techniques in the proof of security.

More recently, Lin, Pass and Venkitasubramaniam [17] showed that constant-round protocols for secure multi-party computation may be based on enhanced trapdoor permutations and any "natural" constant-round non-malleable commitment scheme. Combining their construction with our commitment scheme yields the following corollary:

> **Corollary (informal):** Suppose there exists one-way functions secure against sub-exponential size circuits and standard enhanced trapdoor permutations. Then, there exists a constant-round protocol that secure computes any multi-party functionality against a malicious adversary corrupting any number of parties.

As with our non-malleable commitment scheme, the ensuing protocol for secure multi-party computation admits a black-box proof of security.

Perspective. Prior to this work, the trade-offs between computational assumptions and round complexity for non-malleable commitments and secure computation looked fairly similar to those for (computational) zero-knowledge *proofs* for NP (c.f. [11, 8]): we have constant-round protocols based on collision-resistant hash functions whereas those based on the minimal assumption of one-way functions require at least a

[2] In the protocol, each player takes turns to sequentially commit to its input (along with a "proof of knowledge"); any non-trivial improvement in round complexity will require interweaving these input commitments, which could potentially allow an adversary to violate input independence via a man-in-the-middle attack.

super-constant number of rounds (for secure computation, we also require oblivious transfer). An interesting open problem is whether we can also base constant-round zero-knowledge proofs for NP on one-way functions with sub-exponential hardness.

1.2 Our Techniques

Our construction of the non-malleable commitment scheme proceeds in two steps:

Step 1: Short identities from sub-exponential hardness. First, we construct a constant-round concurrent non-malleable commitment scheme for identities of length $\log\log\log n + O(1)$ (again, n here refers to the security parameter). Our main technical contribution lies in this step. The starting point of this construction is "two-slot message length" technique from [22] underlying the recent constructions of constant-round non-malleable protocols in [24, 25].[3] The basic (and very much simplified) idea is to let the receiver *sequentially* send two challenges—one "long" and one "short"; the length of the challenges are determined by the identity of the sender. Intuitively, the protocol is designed to have the property that the response to a shorter challenge does not help an adversary to provide a response to a longer challenge. If done appropriately, this guarantees that the man-in-the-middle adversary needs to act independently. Our key conceptual insight is to rely on the complexity leveraging technique from [4] to construct these challenges.[4] More precisely, for one-way functions with sub-exponential hardness, an oracle for inverting challenges of length $n^{o(1)}$ (the "short" challenge) does not help invert invert random challenges of length n (the "long" challenge), since we may simulate such an oracle by brute force in time $2^{n^{o(1)}}$.

Step 2: Non-malleability amplification. Next, we transform the initial construction into a constant-round concurrent non-malleable commitment scheme for identities of length $\text{poly}(n)$. This relies on *non-malleability amplification* techniques of Lin and Pass [15]. This is a transformation of so-called "natural" commitment schemes that are non-malleable for identities of length t into ones for identities of length $\Omega(2^t)$ while incurring only a constant multiplicative blow-up in round complexity. We modify our initial construction to satisfy naturality by using the "multiple slots" approach from [22] (introduced in the context of handling longer identities) to boost the number of rewinding slots. Applying the [15] transformation to the modified construction a constant number of times yields the final construction.[5]

Our final protocol has a conceptually simple and "elementary" proof of security. This is a welcome respite from the technical subtleties and complexity and/or heavy technical machinery that arise in much of the previous literature on non-malleability. We also

[3] Our protocol, like that in [24, 25], also has a "commit and prove" structure.

[4] This appears to be the first work to exploit complexity leveraging with a super-constant levels of challenges.

[5] In [15], the transformation is applied to the [6] protocol for constant-length identities (for which the protocol is constant-round) a total of $O(\log^* n)$ times.

Primitive	Hardness	Rounds	Black-box?	Reference
one-way functions	standard	$O(\log n)$	yes	[6]
one-way functions	standard	$O(1)^{\log^* n}$	yes	[15]
one-way functions	sub-exp	$O(1)$	yes	this work
collision-resistance	standard	$O(1)$	no	[24]
collision-resistance, TDP	sub-exp	$O(1)$	no	[2, 5]
adaptive OWP	standard	1	yes	[21]

Fig. 1. Summary of non-malleable statistically binding commitments

point out that complexity leveraging has been previously used in [18, 23] - as in Step 1 - to achieve similar but weaker notions of "independence". The constructions therein use a single challenge slot and achieve only "uni-directional" independence as they require that the challenge in the left interaction be shorter than that on the right. This appears a prori to be an inherent limitation of the complexity leveraging approach[6], because with two challenge slots, the long challenge in the left interaction may be longer than both challenges on the right, so that solving the challenge on the left via brute force violates soundness for both challenges on the right. We show precisely how to overcome this difficulty in our construction and in the analysis.

Organization. We present our construction for short identities in Section 3. For simplicity, we first present the construction assuming one-way *permutations* secure against circuits of size 2^{n^δ} for some constant $\delta < 1$. In Section 4, we apply non-malleability amplification to handle identities of length $\mathrm{poly}(n)$. In Section 5, we modify our constructions to work with general one-way *functions* (as opposed to permutations).

2 Concurrent Non-malleable Commitments

We recall the definition of concurrent non-malleability from [16], which builds upon those in [6, 25]. Let $(\mathcal{C}, \mathcal{R})$ be a commitment scheme with identities, and 1^n be the security parameter.

The man-in-the-middle execution. In the man-in-the-middle execution, the adversary $\mathcal{A}$ is participating m left interactions and m right interactions. In the left interactions, $\mathcal{A}$ interacts with $\mathcal{C}$ receiving a commitment to m values $v_1, \ldots, v_m$, using identities $\mathsf{id}_1, \ldots, \mathsf{id}_m$ of its choice. In the right interactions, $\mathcal{A}$ interacts with $\mathcal{R}$ attempting to commit to a sequence of m related values $\tilde{v}_1, \ldots, \tilde{v}_m$, again using identities $\tilde{\mathsf{id}}_1, \ldots, \tilde{\mathsf{id}}_m$ of its choice. $\mathcal{A}$ also receives an auxiliary z. If any of the

[6] And indeed, [23] —the pre-cursor to [22]— handles the "opposite direction" via non-black-box techniques.

right commitments (as determined by the transcript) are invalid or undefined, its value is set to $\bot$. For any i such that $\tilde{\mathsf{id}}_i = \tilde{\mathsf{id}}_j$ for some j, the value $\tilde{v}_i$ is also set to $\bot$ (that is, any commitment where adversary uses the same identity as that in one of the left interactions is considered invalid). We write $\mathsf{mim}^{\mathcal{A}}(v_1, \ldots, v_m, z)$ to denote a random variable comprising the view of $\mathcal{A}$ along with the m-tuple of values $(\tilde{v}_1, \ldots, \tilde{v}_m)$.

The simulated execution. In the simulated execution, a simulator $\mathcal{S}$ receives the auxiliary input z and interacts directly with $\mathcal{R}$ in m right interactions. We write $\mathsf{sta}^{\mathcal{S}}(1^n, z)$ to denote a random variable comprising the output of $\mathcal{S}$ along with the m-tuple of values $(\tilde{v}_1, \ldots, \tilde{v}_m)$ that the simulator has committed to as determined by the transcript.

Definition 1 ([16, 6, 25]). *A commitment scheme* $(\mathcal{C}, \mathcal{R})$ *is* concurrent non-malleable *if for every* PPT $\mathcal{A}$ *and every polynomial* $m = m(n)$*, there exists a* PPT $\mathcal{S}$ *such that*

$$\left\{\mathsf{mim}^{\mathcal{A}}(v_1, \ldots, v_m, z)\right\}_{v_1,\ldots,v_m \in \{0,1\}^n, z \in \{0,1\}^*} \text{ and}$$

$$\left\{\mathsf{sta}^{\mathcal{S}}(1^n, z)\right\}_{v_1,\ldots,v_m \in \{0,1\}^n, z \in \{0,1\}^*, \mathsf{id} \in \{0,1\}^m}$$

are computationally indistinguishable.

We will also consider a restricted notion of concurrent non-malleability where in the left and right interactions, the adversary $\mathcal{A}$ may only use identities of length at most d. In addition, we will refer to relaxed notions of concurrent non-malleability: one-many and one-one non-malleability. In the former, the adversary participates in one interaction on the left and m interactions on the right, and in the latter, the adversary participates in one interaction on the left and one interaction on the right. As shown in [16], any commitment scheme that is one-many non-malleable is also concurrent non-malleable.

3 Short Identities from Sub-exponential Hardness

3.1 Overview of Our Construction

We construct a family of $d = \Theta(\log \log n)$ protocols (corresponding to d different identities) as follows. Let $n^{\omega(1)} = T_0 \ll T_1 \ll \cdots \ll T_{d-1}$ be a hierarchy of running times. The ith protocol in the family, $i = 0, 1, \ldots, d-1$ is as follows: to commit to a string v (with identity i),

- Commit to v using a statistically binding commitment Com that is hiding against adversaries of size T_d.
- **Slot 1:** prove knowledge of v using a zero-knowledge argument of knowledge that is computationally sound against adversaries of size T_i and can be (straight-line) simulated in time $o(T_{i+1})$.
- **Slot 2:** prove knowledge of v using a zero-knowledge argument of knowledge that is computationally sound against adversaries of size T_{d-1-i} and can be (straight-line) simulated in time $o(T_{d-i})$.

The intuition is that for one of the two slots, the man-in-middle adversary must prove knowledge of the string $\tilde{v}$ committed to in the right interaction without getting much help from the left interaction. Roughly speaking, we will then "extract" from that slot on the right (by rewinding) while simulating on the left (c.f. [6]). To guarantee that the extraction succeeds we need to ensure that the simulation does not violate the soundness of the right interaction; this property is often called *simulation soundness* [26].

For concreteness, consider a synchronizing adversary participating in the ith protocol on the left and the jth protocol on the right. If $i < j$, we may extract the string committed to on the right as follows: run the knowledge extractor for first slot on the right while simulating the first slot on the left. This works because we may simulate on the left in time $o(T_{i+1}) \ll T_j$ without rewinding, without knowing the string committed to on the left, and without violating soundness for the first slot on the right. Similarly, if $i > j$, we can extract the string committed to on the right by running the knowledge extractor for the second slot in the right while simulating the second slot on the left in time $o(T_{d-i+1}) \ll T_{d-j}$. In either case, we may achieve strict polynomial-time simulation by running the man-in-the-middle adversary and committing to 0^n on the left (cf. [25, 16]).

We point out several technical difficulties that arise in turning the above intuition into a proof (indeed, the actual analysis is quite different from that suggested by the above line of reasoning).

- *Simulation may violate soundness.* Consider the case $i > j$, where we need to extract from the second slot on the right. To reach the second slot, we will still need to simulate the first slot on the left, and simply running the straight-line simulator may violate soundness for the second slot on the right. Roughly speaking, we get around this specific problem by using non-uniformity.
- *Which slot should we extract from?* In the analysis, we need to know which slot to simulate and which one to extract from. This is problematic because we allow the identity on the right to be adaptively chosen, and because we do not know the message schedule in advance. To make things worse, the messages may be adaptively and dynamically scheduled.

The key insight in the analysis is to decouple the issue of extraction and the issue of simulation-soundness (this is similar to the approach in [21]). Specifically, we will always simulate both slots on the left and extract from both slots on the right, no matter what the scheduling is. We will then carefully argue that extraction succeeds in at least one slot even though we may be violating soundness while simulating on the left. This is where we reason about the scheduling of messages. For technical reasons, we will also require that we can break Com via brute force in less time than it takes to break the zero-knowledge property.

3.2 Handling Identities of Length $\log \log \log n + O(1)$

Let π denote a one-way permutation secure against circuits of size 2^{n^δ} (where $\delta < 1$) and let Com be a statistically binding commitment scheme. In addition, let

$\langle \mathcal{P}_{\text{WIPOK}}, \mathcal{V}_{\text{WIPOK}} \rangle$ denote the 3-round public-coin witness-indistinguishable proof of knowledge based on the Feige-Shamir protocol from [7], which satisfies the following properties:

- The first two messages depend only on the length of the instance and the security parameter and can be computed efficiently without knowing the instance or the witness.
- The third message can be computed efficiently given the instance, the witness, and the randomness used to generate the first message.
- The protocol is *special-sound*—namely, given any two accepting proofs of x, $(\alpha, \beta, \gamma), (\alpha, \beta', \gamma')$ such that $\beta \neq \beta'$. a witness to x can be efficiently recovered.

We consider a hierarchy of security levels for the one-wayness of the d permutations $\pi_0, \ldots, \pi_{d-1}$ and the hiding properties of Com and $\langle \mathcal{P}_{\text{WIPOK}}, \mathcal{V}_{\text{WIPOK}} \rangle$, that is given by:

$$\pi_0 \ll \pi_1 \ll \cdots \ll \pi_{d-1} \ll \mathsf{Com} \ll \langle \mathcal{P}_{\text{WIPOK}}, \mathcal{V}_{\text{WIPOK}} \rangle$$

- For each $i = 0, 1, \ldots, d-1$: π_i is T_i-one-way but can be broken in time $T_{i+1}^{1/2}$.
- Com is T_d-hiding but can be broken in time $T_{d+1}^{1/2}$.
- $\langle \mathcal{P}_{\text{WIPOK}}, \mathcal{V}_{\text{WIPOK}} \rangle$ is T_{d+1}-witness-indistinguishable (by using a T_{d+1}-hiding commitment). We denote the messages of the protocol by (α, β, γ).

Specifically, we pick π_i to be π restricted to $\{0,1\}^{\ell_i}$, where $\ell_i = (\log n)^{(4/\delta)^{i+1}}$ so that $\text{poly}(n) \cdot 2^{\ell_i} \ll 2^{\ell_{i+1}^{\delta}}$. Taking $\ell_{d-1} = \text{poly}(n)$, yields $(4/\delta)^d = \Theta(\log n / \log\log n)$ and thus $d = \Theta(\log\log n)$. We will instantiate Com from π on $(\log n)^{(4/\delta)^{d+2}}$ bits and $\langle \mathcal{P}_{\text{WIPOK}}, \mathcal{V}_{\text{WIPOK}} \rangle$ from π on $(\log n)^{(4/\delta)^{d+3}}$ bits. We present the protocol in Fig 2.

Lemma 1. *The protocol* nmCom *is a statistically binding commitment scheme.*

Proof. The binding property follows readily from the fact that Com is itself statistically binding. To establish hiding, we construct a simulator $\mathcal{C}'$ that plays the role of the sender in nmCom. $\mathcal{C}'$ on input a commitment c to a string under Com and an identity id interacts with $\mathcal{R}$ as follows:

Stage 0: Sends c.

Stage 1: Computes $s_1 = \pi^{-1}(\sigma_1)$ and proves the statement (c, σ_1) using the witness $(\perp, \perp, s_1)$.

Stage 2: Computes $s_2 = \pi^{-1}(\sigma_2)$ and proves the statement (c, σ_2) using the witness $(\perp, \perp, s_2)$.

We allow $\mathcal{C}'$ to run in time $o(T_d)$ so that it can invert π on σ_1, σ_2. Then, witness indistinguishability of $\langle \mathcal{P}_{\text{WIPOK}}, \mathcal{V}_{\text{WIPOK}} \rangle$ implies that for all v:

$$\mathsf{view}_{\mathcal{R}^*} \langle \mathcal{C}(v), \mathcal{R}^* \rangle \cong_c \mathsf{view}_{\mathcal{R}^*} \langle \mathcal{C}'(\mathsf{Com}(v)), \mathcal{R}^* \rangle$$

On the other hand, Com is T_d-hiding and $\mathcal{C}'$ runs in time $o(T_d)$, so we have

$$\mathsf{view}_{\mathcal{R}^*} \langle \mathcal{C}'(\mathsf{Com}(v)), \mathcal{R}^* \rangle \cong_c \mathsf{view}_{\mathcal{R}^*} \langle \mathcal{C}'(\mathsf{Com}(0^n)), \mathcal{R}^* \rangle$$

Common input: security parameter 1^n and an identity $\mathsf{id} \in \{0, 1, \ldots, d-1\}$.

Sender's input: a value $v \in \{0,1\}^n$.

Commit Phase:

Stage 0:

$\mathcal{C} \to \mathcal{R}$: Pick uniformly $r \in \{0,1\}^{\mathrm{poly}(n)}$ and send $c = \mathsf{Com}(v; r)$.

Stage 1 (Slot 1):

$\mathcal{R} \to \mathcal{C}$: Pick uniformly $\sigma_1 \in \{0,1\}^{\ell_{\mathsf{id}}}$.

$\mathcal{C} \Leftrightarrow \mathcal{R}$: Prove statement (c, σ_1) using $\langle \mathcal{P}_{\mathrm{WIPOK}}, \mathcal{V}_{\mathrm{WIPOK}} \rangle$ and witness $(v, r, \perp)$ w.r.t. the relation

$$\Lambda_{\mathsf{Com}} = \{((c, \sigma), (v, r, s)), |s| = |\sigma| \mid c = \mathsf{Com}(v; r) \text{ OR } \pi(s) = \sigma\}$$

Stage 2 (Slot 2):

$\mathcal{R} \to \mathcal{C}$: Pick uniformly $\sigma_2 \in \{0,1\}^{\ell_{d-1-\mathsf{id}}}$.

$\mathcal{C} \Leftrightarrow \mathcal{R}$: Prove statement (c, σ_2) using $\langle \mathcal{P}_{\mathrm{WIPOK}}, \mathcal{V}_{\mathrm{WIPOK}} \rangle$ and witness $(v, r, \perp)$ w.r.t. Λ_{Com}.

Reveal Phase:

$\mathcal{C} \to \mathcal{R}$: Send v, r.

$\mathcal{R}$: Verify that $c = \mathsf{Com}(v; r)$.

Fig. 2. The commitment scheme $\mathsf{nmCom} = (\mathcal{C}, \mathcal{R})$ for short identities. We denote the 4 messages exchanged in stage b by $\sigma_b, \alpha_b, \beta_b, \gamma_b$, for $b = 1, 2$. The values $\ell_0, \ldots, \ell_d$ are specified in Section 3.2.

Combining, we obtain $\mathsf{view}_{\mathcal{R}^*}\langle \mathcal{C}(v), \mathcal{R}^* \rangle \cong_c \mathsf{view}_{\mathcal{R}^*}\langle \mathcal{C}(0^n), \mathcal{R}^* \rangle$, from which hiding follows. □

Lemma 2. *The protocol* nmCom *is one-one non-malleable for identities of length* $\log\log\log n + O(1)$.

Proof. Consider a man-in-the-middle adversary $\mathcal{A}$. We assume WLOG that $\mathcal{A}$ is deterministic. Following [25, 16], the stand-alone adversary $\mathcal{S}$ uses $\mathcal{A}$ as a black box and emulates the left interaction by honestly committing to the string 0^n. Messages from the right interaction are forwarded externally. As such, it suffices to show that for all v:

$$\mathsf{mim}^{\mathcal{A}}_{\mathsf{nmCom}}(v) \cong_c \mathsf{mim}^{\mathcal{A}}_{\mathsf{nmCom}}(0^n) \qquad (*)$$

On a high level, the proof consists of a series of hybrid arguments:

STEP 1: *Simulate the left interaction using $\mathcal{C}'$ instead of $\mathcal{C}$.*

Specifically, let $\mathcal{S}'$ denote the stand-alone adversary that like $\mathcal{S}$, uses $\mathcal{A}$ as a black box and forwards message from the right interaction externally; the difference is that it emulates the left interaction by running $\mathcal{C}'$ (on input $\mathsf{Com}(v)$) instead of $\mathcal{C}$. We denote the output of this experiment by $\mathsf{sta}^{\mathcal{S}'}(\mathsf{Com}(v))$. By T_{d+1}-witness-indistinguishability, the transcripts of the right interaction when we use $\mathcal{C}$ and when we use $\mathcal{C}'$ on the left will be T_{d+1}-indistinguishable; in particular, the commitments in Stage 0 on the right are T_{d+1}-indistinguishable. Recall that we can extract the values in these Stage 0 commitments in time $o(T_{d+1})$. This implies:

$$\mathsf{mim}^{\mathcal{A}}_{\mathsf{nmCom}}(v) \cong_c \mathsf{sta}^{\mathcal{S}'}(\mathsf{Com}(v))$$

STEP 2: *Extract $\tilde{v}$ on the right.*

Using the knowledge extractor for $\langle \mathcal{P}_{\text{WIPOK}}, \mathcal{V}_{\text{WIPOK}} \rangle$ on the right, we may extract the witnesses for both slots on the right in the experiment $\mathsf{sta}^{\mathcal{S}'}(\mathsf{Com}(v))$, the idea being one of the two witnesses should contain the witness $(\tilde{v}, \tilde{r})$ for the commitment on the right. More precisely, let $\mathsf{ext}\text{-}\mathsf{sta}^{\mathcal{S}'}(c)$ denote the output of the following experiment (a pictorial representation is provided in Fig 4):

1. Fix a random tape for $\mathcal{S}'(c)$ by fixing one for $\mathcal{C}'(c)$. This allows us to treat $\mathcal{S}'(c)$ as a single deterministic entity.
2. Fix a random tape for $\mathcal{R}$ and compute $\tau = \langle \mathcal{S}'(c), \mathcal{R} \rangle$. Let $\widetilde{\mathsf{id}}$ denote the tag of the right interaction and τ denote the transcript $(\tilde{c}, \tilde{\sigma}_1, \tilde{\alpha}_1, \tilde{\beta}_1, \tilde{\gamma}_1, \tilde{\sigma}_2, \tilde{\alpha}_2, \tilde{\beta}_2, \tilde{\gamma}_2)$. If $\widetilde{\mathsf{id}} = \mathsf{id}$ or $\mathcal{S}'$ aborts or if $\mathcal{R}$ rejects, output the view of $\mathcal{A}$ and $\perp$ and halt.
3. Rewind and attempt to extract witnesses $\tilde{w}_1, \tilde{w}_2$ for the respective statements $(\tilde{c}, \tilde{\sigma}_1)$ and $(\tilde{c}, \tilde{\sigma}_2)$ w.r.t. Λ_{Com}, relying on the special-soundness of $\langle \mathcal{P}_{\text{WIPOK}}, \mathcal{V}_{\text{WIPOK}} \rangle$. This is done as usual, by sending new random messages $\tilde{\beta}_1'$, $\tilde{\beta}_2'$, but with the following important exception: if $\mathcal{A}$ schedules messages in a different way than in τ (or if $\mathcal{R}$ rejects), the rewinding is aborted, and restarted. Let Γ denote the set of all possible scheduling; clearly, $|\Gamma| = O(1)$ since the protocol is constant-round. We will show that the expected number of rewindings for Slot 1 is given by $|\Gamma| = O(1)$; the same argument applies to Slot 2. Let τ_1 denote the prefix of τ up to Slot 1. For each schedule $\rho \in \Gamma$, let $\Pr[\rho \mid \tau_1]$ denote the probability that $\mathcal{R}$ accepts (i.e. obtaining convincing proof both slots) using the scheduling ρ conditioned on the prefix being τ_1. For a fixed τ_1, ρ, the expected number of rewindings is given by $\frac{1}{\Pr[\rho|\tau_1]}$. Therefore, the total expected number of rewindings for Slot 1 is given by:

$$\sum_{\tau_1} \Pr[\tau_1] \sum_{\rho \in \Gamma} \Pr[\rho \mid \tau_1] \cdot \frac{1}{\Pr[\rho \mid \tau_1]} = \sum_{\tau_1} \Pr[\tau_1] \cdot |\Gamma| = |\Gamma|$$

By linearity of expectations, the total expected number of rewindings for both slots is also $O(1)$. We now only need to make sure that we indeed extracted a valid opening: if either $\tilde{w}_1$ or $\tilde{w}_2$ is a valid opening $(\tilde{v}, \tilde{r})$ for $\tilde{c}$, output $(\tau, \tilde{v})$, else output `fail`.

We know that whenever $\mathsf{ext\text{-}sta}^{\mathcal{S}'}(\mathsf{Com}(v))$ does not output $\texttt{fail}$, its output contains the correct value $\tilde{v}$ and therefore the distributions

$$\mathsf{sta}^{\mathcal{S}'}(\mathsf{Com}(v)) \quad \text{and} \quad \mathsf{ext\text{-}sta}^{\mathcal{S}'}(\mathsf{Com}(v))$$

are identical. In the next subsection, we will establish the following claim:

Claim (simulation-soundness). For all v, $\Pr[\mathsf{ext\text{-}sta}^{\mathcal{S}'}(\mathsf{Com}(v)) = \texttt{fail}] = \mathrm{neg}(n)$

For now, we hint that the proof of the claim exploits the two-slot structure in an essential way to transform a non-negligible failure probability in extraction into non-negligible success probability at inverting π. Assuming that the claim holds, it follows readily that

$$\mathsf{sta}^{\mathcal{S}'}(\mathsf{Com}(v)) \cong_c \mathsf{ext\text{-}sta}^{\mathcal{S}'}(\mathsf{Com}(v))$$

STEP 3: *Replace the input to* $\mathcal{S}'$ *with* $\mathsf{Com}(0^n)$.

Now, we observe that $\mathcal{S}'$ combined with the knowledge extractor on the right runs in expected time $o(T_d)$. This is less than the time it takes to break Com, and thus its output will be indistinguishable whether the input to $\mathcal{S}'$ is $\mathsf{Com}(v)$ or $\mathsf{Com}(0^n)$. In particular, we have

$$\mathsf{ext\text{-}sta}^{\mathcal{S}'}(\mathsf{Com}(v)) \cong_c \mathsf{ext\text{-}sta}^{\mathcal{S}'}(\mathsf{Com}(0^n))$$

Combining steps 1 and 2, we have

$$\mathsf{mim}^{\mathcal{A}}_{\mathsf{nmCom}}(v) \cong_c \mathsf{ext\text{-}sta}^{\mathcal{S}'}(\mathsf{Com}(v))$$
$$\mathsf{mim}^{\mathcal{A}}_{\mathsf{nmCom}}(0^n) \cong_c \mathsf{ext\text{-}sta}^{\mathcal{S}'}(\mathsf{Com}(0^n))$$

Combining with Step 3 yields $(*)$. □

3.3 Proof of Simulation-Soundness

We complete the proof of Lemma 2 by establishing the main technical claim. Suppose towards a contradiction that the claim is false, i.e., there is some non-negligible function ϵ such that for all sufficiently large n, there exists some v satisfying

$$\Pr[\mathsf{ext\text{-}sta}^{\mathcal{S}'}(\mathsf{Com}(v)) = \texttt{fail}] > \epsilon(n)$$

Fix one such n, along with an associated v and identity id. In addition, we may also fix the coin tosses of $\mathcal{S}'$ and some specific $c = \mathsf{Com}(v)$, along some $\tilde{\mathsf{id}}$ on the right, while losing a factor d in the probability $\mathsf{ext\text{-}sta}$ outputs $\texttt{fail}$. That is, with probability at least $\frac{\epsilon(n)}{d}$ (over the coin tosses of $\mathcal{R}$), the tag on the right is $\tilde{\mathsf{id}}$ and the knowledge extractor outputs witnesses $\pi^{-1}(\tilde{\sigma}_1)$ and $\pi^{-1}(\tilde{\sigma}_2)$. We then construct an adversary $\tilde{\mathcal{A}}$ that for some $j \in \{\tilde{\mathsf{id}}, d-1-\tilde{\mathsf{id}}\}$, inverts π on $\{0,1\}^{\ell_j}$ with probability $\Omega(\frac{\epsilon(n)}{d})$ in time $o(T_j)$, which contradicts the one-wayness of π. Roughly speaking, $\tilde{\mathcal{A}}$ works as follows: on input a challenge $\sigma \in \{0,1\}^{\ell_j}$, simulate the experiment $\mathsf{ext\text{-}sta}^{\mathcal{S}'}(\mathsf{Com}(v))$, and

- if $j = \tilde{\mathsf{id}}$, set $\tilde{\sigma}_1 = \sigma$ and compute $\pi^{-1}(\sigma)$ by extracting the witness from Slot 1; and
- if $j = d - 1 - \tilde{\mathsf{id}}$, set $\tilde{\sigma}_2 = \sigma$ and compute $\pi^{-1}(\sigma)$ by extracting the witness from Slot 2.

Recall that $\mathcal{S}'$ is simply $\mathcal{A}$ with a left execution of $\mathcal{C}'(\mathsf{Com}(v))$ and thus a naive simulation of $\mathcal{S}'$ takes time roughly $T_d \gg T_j$. The bottleneck to an efficient simulation lies in computing each of the messages γ_1, γ_2 in stages 1 and 2 in the computation of $\mathcal{C}'$. We adopt one of three strategies to accomplish this in time $o(T_j)$: compute the message by computing a witness, hardwire the message into the reduction, or argue that we do not need to compute the message for extraction on the right. We consider three representative schedulings of the messages γ_1, γ_2 in relation to the two slots in the right execution. In our analysis we crucially rely on the fact that ext-sta aborts all rewindings that use a different schedule than ext-sta saw in the first simulation τ. Given this property it is sufficient to consider a *static* scheduling. In particular, as the number of possible scheduling is constant, we can WLOG consider a particular fixed scheduling (again at the cost of only a constant loss).

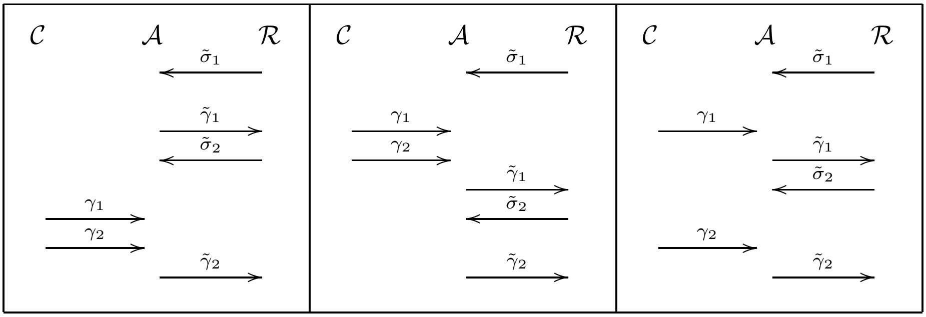


Fig. 3. Three scheduling strategies

Both γ_1, γ_2 are sent after Slot 1 ends. In this case, we construct an adversary that breaks π on $\{0,1\}^{\ell_{\tilde{\mathsf{id}}}}$ by extracting from Slot 1 on the right. Here, we observe that we do not need to compute γ_1 or γ_2. Specifically, we just need to simulate the interaction on the left up to the end of Slot 1. Since both γ_1 and γ_2 are sent after Slot 1, we can complete the simulation on the left in polynomial time.

Both γ_1, γ_2 are sent before Slot 2 begins. In this case, we construct an adversary that breaks π on $\{0,1\}^{\ell_{d-1-\tilde{\mathsf{id}}}}$ by extracting from Slot 2 on the right. Here, we argue that we can fix γ_1 and γ_2. Specifically, we non-uniformly fix a partial transcript of the execution up to the point just before Slot 2 on the right begins, i.e. just before $\tilde{\sigma_2}$ is sent. This fixes all of the left interaction, so there is nothing left to simulate on the left.

γ_1 **is contained in Slot 1 and** γ_2 **in Slot 2.** More generally, this covers the case γ_1 is sent before Slot 2 begins and γ_2 is sent after Slot 1 ends. We need to consider two sub-cases:

- $\mathsf{id} < \tilde{\mathsf{id}}$. We construct an adversary that breaks π on $\{0,1\}^{\ell_{\tilde{\mathsf{id}}}}$ by extracting from Slot 1 on the right. Here, we compute γ_1 (by inverting σ_1) and observe that we do not need to compute γ_2. Specifically, we just need to simulate the interaction on the left up to the end of Slot 1 on the right, i.e. up to the point $\tilde{\gamma_1}$ is sent. It suffices to compute $\pi^{-1}(\sigma_1)$ and thus γ_1, which can be done in time $o(T_{\mathsf{id}+1}) \ll T_{\tilde{\mathsf{id}}}$.
- $\mathsf{id} > \tilde{\mathsf{id}}$. We construct an adversary that breaks π on $\{0,1\}^{\ell_{d-\tilde{\mathsf{id}}-1}}$ by extracting from Slot 2 on the right. Here, we fix γ_1 and then compute γ_2 (by inverting σ_2). Specifically, we non-uniformly fix a partial transcript of the execution up to the point just before Slot 2 on the right begins, i.e. just before $\tilde{\sigma_2}$ is sent. This fixes all of the first slot on the left (including γ_1) which we may then hardwire into the reduction. To complete the simulation on the left for Slot 2, it suffices to compute $\pi^{-1}(\sigma_2)$ and thus γ_2, which can be done in time $o(T_{d-\mathsf{id}}) \ll T_{d-\tilde{\mathsf{id}}-1}$.

With probability $\Omega(\frac{\epsilon(n)}{d})$, $\mathsf{ext\text{-}sta}$ outputs `fail` for one of these schedulings. We may then use $\tilde{\mathcal{A}}$ for that scheduling to derive a contradiction to the one-wayness of π.

Remark 1. We highlight two subtleties in the analysis:

- It is important that in $\tilde{\mathcal{A}}$'s simulation of $\mathsf{ext\text{-}sta}^{\mathcal{S}'}(\mathsf{Com}(v))$, it uses the same witnesses for the relation Λ_{Com} as $\mathcal{C}'$; otherwise, we could have easily solved the problem of efficient simulation by having $\tilde{A}$ use the witness (v,r) for the commitment $\mathsf{Com}(v)$. We cannot appeal to witness-indistinguishability here because we rewind the $\langle \mathcal{P}_{\mathrm{WIPOK}}, \mathcal{V}_{\mathrm{WIPOK}} \rangle$ protocols.
- It is also important for simulation-extractability that σ_2 is sent *after* the completion of Stage 1 in nmCom. This way, we can fix a partial transcript up to the end of Slot 1 while allowing the verifier's challenge σ_2 in Slot 2 to remain undetermined.

4 Non-malleability Amplification

In order to apply the non-malleable amplification theorem from [15] to our construction, we first need to modify our construction to satisfy an additional technical requirement, that of non-malleability w.r.t 4-round protocols (to be formalized shortly), which they coin *natural*. [15] also requires that the commitment scheme be *initial-binding*, that is, the first message sent by the sender already determines the value committed to; our commitment scheme clearly satisfies this.

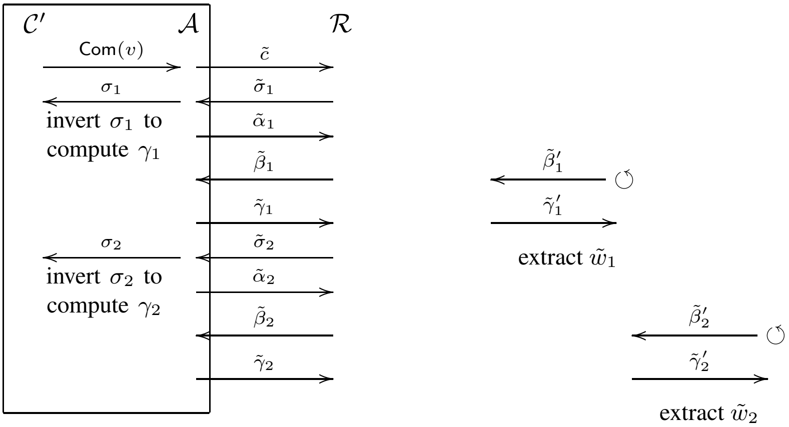


Fig. 4. A pictorial representation of $\mathsf{ext\text{-}sta}^{\mathcal{S}'}(\mathsf{Com}(v))$

Lemma 3 (Non-malleability amplification [15]). *Let $\langle \mathcal{C}, \mathcal{R} \rangle$ be a $k(n)$-round natural non-malleable commitment scheme for identities of length $t(n)$ with computational complexity $p(n)$. Then, there exists a $15k(n)$-round natural non-malleable commitment scheme for identities of length $2^{t(n)-1}$ with computational complexity $2^{t(n)}p(n) + k(n)poly(n) + poly(n)$.*

Non-malleability w.r.t. *k*-round protocols. The concept of non-malleability is traditionally only considered in a setting where a man-in-the middle adversary is participating in two (or more) executions of the *same* protocol. We here consider a notion of non-malleability with respect to arbitrary k-round protocols.

Consider a one-many man-in-the-middle adversary A that participates in one left interaction—communicating with a machine B—and in many right interactions—acting as a committer using the commitment scheme $(\mathcal{C}, \mathcal{R})$. As in the standard definition of non-malleability, A can adaptively choose the identities in the right interactions. We denote by $\mathsf{mim}^{B,A}(y, z)$ the random variable consisting of the view of $A(z)$ in a man-in-the-middle execution when communicating with $B(y)$ on the left and honest receivers on the right, combined with the values $A(z)$ commits to on the right. Intuitively, we say that $(\mathcal{C}, \mathcal{R})$ is one-many non-malleable w.r.t B if $\mathsf{mim}^{B,A}(y_1, z)$ and $\mathsf{mim}^{B,A}(y_2, z)$ are indistinguishable, whenever interactions with $B(y_1)$ and $B(y_2)$ cannot be distinguished. More formally, let $\mathsf{view}_A[\langle B(y), A(z)\rangle]$ denote the view of $A(z)$ in an interaction with $B(y)$.

Definition 2. *Let $(\mathcal{C}, \mathcal{R})$ be a commitment scheme, and B an interactive Turing machine. We say the commitment scheme $(\mathcal{C}, \mathcal{R})$ is* one-many non-malleable w.r.t. *B, if for every probabilistic polynomial-time man-in-the-middle adversary A, and every two sequences $\{y_n^1\}_{n\in N}$ and $\{y_n^2\}_{n\in N}$, such that*

$$\left\{\mathsf{view}_A[\langle B(y_n^1), A(z)\rangle]\right\}_{n\in N, z\in\{0,1\}^*} \approx \left\{\mathsf{view}_A[\langle B(y_n^2), A(z)\rangle]\right\}_{n\in N, z\in\{0,1\}^*}$$

it holds that:

$$\left\{\mathsf{mim}^{B,A}(y_n^1, z)\right\}_{n\in N, z\in\{0,1\}^*} \approx \left\{\mathsf{mim}^{B,A}(y_n^2, z)\right\}_{n\in N, z\in\{0,1\}^*}$$

We say that $(\mathcal{C}, \mathcal{R})$ is one-many non-malleable w.r.t k-round protocols if $(\mathcal{C}, \mathcal{R})$ is one-many non-malleable w.r.t any machine B that interacts with the man-in-the-middle adversary in k rounds.

Modifying our construction. We describe a variant of our construction in Section 3 that is one-many non-malleable w.r.t $(2c-1)$-round protocols for any constant $c > 1$. In addition, the protocol now handles identities of length $c \log\log\log n + O(1)$, although the increase is not necessary for non-malleability amplification. Specifically, we follow the multiple slot approach in [22] to boost the number of slots from 2 to $2c$. On input a tag $\mathsf{id} \in 0, 1, \ldots, d^c - 1$, let $(\mathsf{id}_1, \ldots, \mathsf{id}_c)$ denote the base d representation of id. For $j = 1, 2, \ldots, c$, we will pick a challenge of length ℓ_{id_j} for the $2j-1$'th slot, and a challenge of length $\ell_{d-1-\mathsf{id}_j}$ for the $2j$'th slot.

The analysis. First, we need to verify that the modified construction remains one-many non-malleable (w.r.t. itself). Indeed, the proof of Lemma 2 and the analysis in Section A extend in a straight-forward manner to $c > 1$, except in the proof of simulation-soundness, where it is slightly more involved. We will consider two broad classes of scheduling strategies:

- For all $j = 1, 2, \ldots, c$: γ_{2j-1} is contained in Slot $2j-1$ and γ_{2j} is contained in Slot $2j$.
- There exists some j where one of Slot $2j-1$ or Slot $2j$ contains none of $\gamma_1, \ldots, \gamma_{2c}$.

The previous analysis will still go through, except we now lose a factor $\frac{1}{(dc)^{\Omega(c)}}$ (as opposed to $1/d$ from before) in the probability of inverting the one-way permutation.

Next, we argue that the modified construction is one-many non-malleable w.r.t $(2c-1)$-round protocols. This follows from the fact that we now have $2c$ rewinding slots on the right (c.f. [15]) so that there will always be a slot on the right that does not contain any message from the $(2c-1)$-round protocol executing on the left.

5 Construction from Sub-exponential One-Way Functions

We need to make two modifications to the protocol in Section 3 in order to handle a general one-way function f instead of a one-way permutation π with sub-exponential hardness.

Modifying receiver's challenge. Following [3], we will replace the challenge that the receiver sends at the start of each of the two slots with a 3-round challenge response protocol. This is essentially a cut-and-choose protocol that guarantees that the receiver sends challenges in the range of the one-way function f. Again, we fix some input length ℓ for f corresponding to the desired level of security for the slot.

$\mathcal{R} \to \mathcal{C}$: Pick s_i^b at random from $\{0,1\}^\ell$ and send $y_i^b = f(s_i^b)$ for $b = 0, 1,\ i = 1, 2, \ldots, n$.

$\mathcal{C} \to \mathcal{R}$: Send $\mu = (\mu_1, \ldots, \mu_n)$ at random from $\{0,1\}^n$.

$\mathcal{R} \to \mathcal{C}$: Send $(s_1^{\mu_1}, \ldots, s_n^{\mu_n})$.

$\mathcal{C}$: Verify that for all $i = 1, 2, \ldots, n$: $f(s_i^{\mu_i}) = y_i$.

The sender will then run $\langle \mathcal{P}_{\text{WIPOK}}, \mathcal{V}_{\text{WIPOK}} \rangle$ on the instance $(c, y_1^0, y_1^1, \ldots, y_n^0, y_n^1, \mu)$ w.r.t. the following relation:

$$\Lambda_{\mathsf{Com}} = \{((c, y_1^0, y_1^1, \ldots, y_n^0, y_n^1, \mu), (v, r, i, s)), \mid c = \mathsf{Com}(v; r) \text{ OR } f(s) = y_i^{1-\mu_i}\}$$

The challenge-response protocol has the following properties (cf. [3]):

- With probability $1 - 2^{-n}$ over μ, if the sender accepts at the end of the challenge-response protocol, then there exists a trapdoor witness for the relation Λ_{Com}. Indeed, a trapdoor witness exists unless at most one value in each pair (y_i^0, y_i^1) lies in $f(\{0,1\}^\ell)$, in which case there exists at most one μ for which the sender will not abort.
- It is computationally infeasible for a $2^{O(\ell)}$-time adversary to find a trapdoor witness for the relation Λ_{Com} if f is an exponential one-way function.

Modifying the commitment schemes. We will use Naor's commitment scheme [19] in Com and in $\langle \mathcal{P}_{\text{WIPOK}}, \mathcal{V}_{\text{WIPOK}} \rangle$. Specifically, we will commit v by committing to each bit of v in parallel. We may set the values of $T_0, \ldots, T_d$ as before. The complexity of breaking a T_d-hiding commitment via brute-force is now $\text{poly}(n) \cdot 2^{O((\log T_d)^\kappa))}$ (for some constant $\kappa > 1$ that depends on the seed length of pseudorandom generators from one-way functions in [12]). We can then set $\ell_{d+1} = n^{\Theta(\kappa/\delta)}$ to ensure that $T_{d+1}^{1/2} > \text{poly}(n) \cdot 2^{O((\log T_d)^\kappa))}$.

Acknowledgments. We thank Vinod Vaikuntanathan for an inspiring discussion on non-malleability.

References

[1] Barak, B.: How to go beyond the black-box simulation barrier. In: FOCS, pp. 106–115 (2001)

[2] Barak, B.: Constant-round coin-tossing with a man in the middle or realizing the shared random string model. In: FOCS, pp. 345–355 (2002)

[3] Bellare, M., Jakobsson, M., Yung, M.: Round-optimal zero-knowledge arguments based on any one-way function. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 280–305. Springer, Heidelberg (1997)

[4] Canetti, R., Goldreich, O., Goldwasser, S., Micali, S.: Resettable zero-knowledge. In: STOC, pp. 235–244 (2000)

[5] Di Crescenzo, G., Ishai, Y., Ostrovsky, R.: Non-interactive and non-malleable commitment. In: Proc. 30th STOC, pp. 141–150 (1998)

[6] Dolev, D., Dwork, C., Naor, M.: Nonmalleable cryptography. SIAM J. Comput. 30(2), 391–437 (2000)

[7] Feige, U., Shamir, A.: Zero knowledge proofs of knowledge in two rounds. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 526–544. Springer, Heidelberg (1989)

[8] Goldreich, O., Kahan, A.: How to construct constant-round zero-knowledge proof systems for NP. J. Cryptology 9(3), 167–190 (1996)

[9] Goldreich, O., Krawczyk, H.: On the composition of zero-knowledge proof systems. SIAM J. Comput. 25(1), 169–192 (1996)

[10] Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: STOC, pp. 218–229 (1987)

[11] Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. J. ACM 38(3), 691–729 (1991); Prelim. version in FOCS 1986.

[12] 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)

[13] Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: STOC, pp. 44–61 (1989)

[14] Katz, J., Ostrovsky, R., Smith, A.: Round efficiency of multi-party computation with a dishonest majority. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 578–595. Springer, Heidelberg (2003)

[15] Lin, H., Pass, R.: Non-malleability amplification. In: STOC, pp. 189–198 (2009)

[16] Lin, H., Pass, R., Venkitasubramaniam, M.: Concurrent non-malleable commitments from any one-way function. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 571–588. Springer, Heidelberg (2008)

[17] Lin, H., Pass, R., Venkitasubramaniam, M.: A unified framework for concurrent security: universal composability from stand-alone non-malleability. In: STOC, pp. 179–188 (2009)

[18] Liskov, M., Lysyanskaya, A., Micali, S., Reyzin, L., Smith, A.: Mutually independent commitments. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 385–401. Springer, Heidelberg (2001)

[19] Naor, M.: Bit commitment using pseudorandomness. J. Cryptology 4(4), 151–158 (1991)

[20] Ostrovsky, R., Persiano, G., Visconti, I.: Simulation-based concurrent non-malleable commitments and decommitments. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 91–108. Springer, Heidelberg (2009)

[21] 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)

[22] Pass, R.: Bounded-concurrent secure multi-party computation with a dishonest majority. In: STOC, pp. 232–241 (2004)

[23] Pass, R., Rosen, A.: Bounded-concurrent secure two-party computation in a constant number of rounds. In: FOCS, pp. 404–413 (2003)

[24] Pass, R., Rosen, A.: New and improved constructions of nonmalleable cryptographic protocols. SIAM J. Comput. 38(2), 702–752 (2008); Preliminary version in STOC 2005

[25] Pass, R., Rosen, A.: Concurrent nonmalleable commitments. SIAM J. Comput. 37(6), 1891–1925 (2008); Preliminary version in FOCS 2005

[26] Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: FOCS, pp. 543–553 (1999)

Appendix

A nmCom Is One-Many Non-malleable

Here, we establish a stronger claim, namely that the protocol nmCom is in fact one-many non-malleable for identities of length $\log \log \log n + O(1)$. We do not need this stronger property, although it is of independent interest. To see why the claim holds, suppose there are m right interactions, where the tags are respectively $(\widetilde{\mathsf{id}}_1, \ldots, \widetilde{\mathsf{id}}_m)$ and the committed values are respectively $(\tilde{v}_1, \ldots, \tilde{v}_m)$. We modify Step 2 to extract each $\tilde{v}_i$ on the right where $\widetilde{\mathsf{id}}_i \neq \mathsf{id}$. As before, we will sample one transcript τ, and then attempt to extract witnesses for each of the m right executions. We need an expected $2m|\Gamma|$ rewindings, $2|\Gamma|$ for each of the m right executions. Next, we will need to show that the probability that the extractor outputs `fail` for any of the m right interactions in negligible. If this probability is at least ϵ, then there is some right interaction for which the extractor outputs `fail` for that interaction with probability at least $\frac{\epsilon}{m}$. Simply repeat the analysis for simulation-soundness in Section 3.3 for this execution (and simulate $\mathcal{R}$ for the other $m-1$ interactions internally).

Constructing Verifiable Random Functions with Large Input Spaces

Susan Hohenberger[1,⋆] and Brent Waters[2,⋆⋆]

[1] Johns Hopkins University
susan@cs.jhu.edu
[2] University of Texas at Austin
bwaters@cs.utexas.edu

Abstract. We present a family of verifiable random functions which are provably secure for exponentially-large input spaces under a non-interactive complexity assumption. Prior constructions required either an interactive complexity assumption or one that could tolerate a factor 2^n security loss for n-bit inputs. Our construction is practical and inspired by the pseudorandom functions of Naor and Reingold and the verifiable random functions of Lysyanskaya. Set in a bilinear group, where the Decisional Diffie-Hellman problem is easy to solve, we require the ℓ-Decisional Diffie-Hellman Exponent assumption in the standard model, without a common reference string. Our core idea is to apply a simulation technique where the large space of VRF inputs is collapsed into a small (polynomial-size) input in the view of the reduction algorithm. This view, however, is information-theoretically hidden from the attacker. Since the input space is exponentially large, we can first apply a collision-resistant hash function to handle arbitrarily-large inputs.

1 Introduction

Verifiable Random Functions (VRFs) were proposed by Micali, Rabin, and Vadhan [24]. VRFs behave similar to Pseudo Random Functions (PRFs) [16] in that an (efficient) attacker should not be able to distinguish the value of $f_K(x)$ from a random value even if it is given oracle access to the function $f_K(\cdot)$ at several other points. However, VRFs have the additional property that the party holding the seed will publish a commitment to the function and is able to *non-interactively* convince a verifier that a given evaluation is correct (i.e., matches the public commitment) without sacrificing the pseudorandom property on other inputs. In addition, the proof must be verifiable without the benefit of a common

[⋆] Supported by NSF CNS-0716142, Department of Homeland Security Grant 2006-CS-001-000001-02 (subaward 641) and a Microsoft Research New Faculty Fellowship.

[⋆⋆] Supported by NSF CNS-0716199, CNS-0915361, and CNS-0952692, Air Force Office of Scientific Research (AFO SR) under the MURI award for "Collaborative policies and assured information sharing" (Project PRESIDIO), Department of Homeland Security Grant 2006-CS-001-000001-02 (subaward 641), and the Alfred P. Sloan Foundation.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 656–672, 2010.

reference string (CRS). Finally, the verification should remain secure even if the public commitment were setup in a malicious manner.

The VRF definition of security limits the types of tools we can apply to solving the problem and restricts us from using several "traditional" approaches. For example, at first glance it might seem possible to construct VRFs in a straightforward manner by applying PRFs together with a Non-Interactive Zero Knowledge Proof [4,5] (NIZK) system. A tempting approach is to publish a commitment to a PRF seed and then the seed holder can apply the NIZK machinery to produce non-interactive proofs. A typical proof would at one stage allow a reduction algorithm to simulate proofs (via knowledge of the CRS setup) even when the algorithm has no knowledge of the function's seed. However, since the definition of a Verifiable Random Function disallows the use of a trusted setup, the NIZK paradigm cannot be applied.

Without being able to simulate proofs, any reduction algorithm that proves pseudorandomness faces the following predicament. First, for any x for which it is asked to give out $f_K(x)$ and a proof it must be able to produce the actual (unique) output of $f_K(x)$. Since there is no interaction or trusted setup, the algorithm is not able to "lie" at any stage. Second, the reduction must be able to use an attacker that can distinguish $f_K(x^*)$ from a random value at a certain x^*. In order to make use of this attacker, it follows that the reduction algorithm must *not* know how to evaluate $f_K(\cdot)$ at certain points.

Meeting these two restrictions will require a new approach to constructing pseudorandom functions that moves past traditional constructions. For instance, to prove that the Goldreich, Goldwasser and Micali PRF construction [16] is pseudorandom one must go through several hybrid experiments, where the reduction algorithm will not know how to correctly evaluate the PRF on any input. This approach will not work for proving VRFs, since the reduction algorithm must provide an evaluation and prove (without lying) that it is correct.

Constructing and Proving Security of VRFs. For the reasons above, existing VRF systems employ a different strategy when proving the security of VRFs. Almost all proofs of VRF constructions (that do not rely on interactive assumptions) [24,14,1] use a type of "all but one" technique for proving pseudorandoness. In these proofs a reduction algorithm will first guess the attacker's challenge input as some random string w in $\{0,1\}^n$, where n is the bit length of inputs. Next, it will set up the commitment such that it knows the function at all $2^n - 1$ inputs values $x \neq w$. The algorithm then must "hope" that the challenge input lands on w. For instance, the Micali-Rabin-Vadhan (VUF[1]) reduction [24] publishes a commitment $r \pmod N$ such that it knows $2^n - 1$ roots of r for primes p_x where $x \neq w$ and hopes that the attacker provides it the p_w-th root of r.

The main drawback of this style of proof is that the error and time component of the reduction respectively degrade and blowup by a factor 2^n, which

[1] VUF stands for *verifiable unpredictable function.* It relaxes the pseudorandomness requirement of the VRF, so that an (efficient) attacker should not be able to predict the value of $f_K(x)$ even if it is given oracle access to the function $f_K(\cdot)$ and its proof at several other points.

is exponential in the input length n. The error reduction decreases by a factor of 2^n from the guessing of the challenge input and the time of the reduction requires $2^n - 1$ steps to "plant" knowledge of $f_K(x)$ for all $x \neq w$. For this reason these VRF systems when applied to large input sizes need to rely on strong assumptions that can absorb the loss of security. Furthermore, in the ℓ-type assumptions used in bilinear map constructions of Dodis-Yampolskiy [14], the number of terms (i.e., ℓ) associated with the assumption increases exponentially in n. [2] In general, we would like to prove security for large input spaces based on a "smaller" and more standard complexity assumption, which contains at most a polynomial number of terms.

Indeed, in their recent paper, Abdalla, Catalano and Fiore [1] stated that an open problem was to construct a VRF "supporting exponentially large (in the security parameter) identity spaces and provably secure under non interactive assumptions".

Our Approach. In this work, we aim to realize VRFs with large input sizes without applying complexity leveraging or interactive assumptions. Our main technique is that we apply a reduction technique where the input space of size 2^n is compressed *in the reduction algorithm's view* to a much smaller space. We can parameterize this compression such that the reduction algorithm knows the PRF value for all but a set S of size $\approx 1/q(\lambda)$ of the input, where $q(\lambda)$ is the (polynomial) number of queries made by an attacker and λ is a security parameter. We then "hope" that the challenge input lands in S and can finish the simulation without aborting a non-negligible fraction of the time.

Our construction makes use of bilinear groups. It has a similar structure to the PRF of Naor-Reingold [27] and the VRF of Lysyanskaya [23]. The setup algorithm will choose a group $\mathbb{G}$ of prime order p along with random group elements $g, h, U_0 = g^{u_0}, \ldots, U_n = g^{u_n}$ for random $u_0, \ldots, u_n \in \mathbb{Z}_p$. The evaluation of the VRF on input $x = x_1 \ldots x_n$ is

$$e(g^{u_0 \prod_{i=1}^{n} u_i^{x_i}}, h).$$

Proofs of the VRF are given using a step ladder approach in a manner similar to that appearing in other works [23,1].

We prove the security of our scheme under the ℓ-Decisional Diffie-Hellman Exponent assumption [7] for $\ell = O(q(\lambda) \cdot n)$. This assumption gives the reduction algorithm g^{a^i} for $i = 1$ to 2ℓ except for a "hole" at $i = \ell$. In our reduction, we associate each U_i value with a value $g^{a^{y_j}}$ for some y_j. (The terms are further randomized so as to information-theoretically hide y_j from the outside. We ignore the randomization terms for this discussion.) For any input x, the reduction can evaluate the function and give a proof if $y_0 + \sum_{i=1}^{n} y_i^{x_i} \neq \ell$. For all other inputs $x \in S$ such that $y_0 + \sum_{i=1}^{n} y_i^{x_i} = \ell$, the reduction algorithm can successfully use an answer to defeat the DDHE assumption.

[2] We note that the second construction of Abdalla-Catalano-Fiore [1] has a polynomial number of assumption terms, but exponential degradation in the input size.

To achieve a polynomial (in n) reduction we must find a way to put a proper fraction of the inputs in S and to make the distribution of inputs in S close to random across the coins of the reduction. For this final goal, we parameterize and analyze our scheme in a manner similar to the Waters' [29] Identity-Based Encryption system. In this system, Waters showed how to partition a fraction of $\approx 1/q(\lambda)$ of the inputs into what he called a challenge set S. We will apply a similar partitioning approach, except we must adapt it to the multiplicative structure of our VRF.

We finally note that once we achieve a VRF for large enough input size n, we can apply one of two techniques to get a VRF for the input domain of $\{0,1\}^*$. First, we could simply let the setup algorithm choose a collision resistant hash function $H : \{0,1\}^* \rightarrow \{0,1\}^n$. The VRF would first hash the input down to n bits and then apply the core VRF. It is fairly straightforward to show that an attack would imply either finding a collision or attacking the core VRF. Another technique is to apply the tree-based extension given by Micali, Rabin, and Vadhan (MRV) [24] which allows extension to unbounded size inputs. This tree-based technique works if there are no collisions discovered in the core VRF applied at each node (i.e., no two nodes have the same label). In order for this to occur, the core input size must be large, which requires complexity leveraging in the MRV RSA construction, but does not when using our techniques.

1.1 Related Work

The concept of *pseudorandom functions* was proposed by Goldreich, Goldwasser and Micali [16]. They provided a definition and gave a generic method of constructing them from any one-way permutation. An efficient PRF based on the Decisional Diffie-Hellman assumption was proposed by Naor and Reingold [27].

Micali, Rabin and Vadhan [24] proposed the extension to verifiable random functions. They gave an RSA-type construction and proved security under what they called the RSA $s(k)$-Hardness Assumption. Roughly, for input length $a(k)$, the security of the VRF was $s'(k) = s(k)^{1/3}/(\text{poly}(k)\cdot 2^{a(k)})$. They then provided a tree-based method for extending the input size to $\{0,1\}^*$. Their construction elegantly showed how to first give a Verifiable Unpredictable Function (VUF) and then apply the Goldreich-Levin [17] hard core bit technique to get a VRF.

Lysyanskaya [23] provided the first VRF scheme from bilinear maps, which was also constructed as a transformation from a VUF. Our VRF construction follows a similar structure and is inspired by that of Lysyanskaya, although we will give a direct VRF construction without first providing a VUF. Dodis [13] extended the work of Lysyanskaya and showed how to give efficient constructions of a VRF directly (i.e., without going through any generic transformations). His VRF was also *distributed* in the sense that a collection of servers can hold shares of the seed and a certain threshold of these servers must cooperate to compute $f_K(x)$ or distinguish its outputs from random. Unfortunately, both of these works rely on interactive complexity assumptions (for large input spaces.)

Dodis and Yampolskiy [14] gave a very efficient VRF under a non-interactive assumption by applying the deterministic version of Boneh-Boyen [6] signatures.

In a bilinear group $\mathbb{G}$ of prime order p, its seed is a single element of $\mathbb{Z}_p$ and its proof is a single element of $\mathbb{G}$. Its main drawback is that its security only holds for small input spaces. For n-bit inputs, the scheme's security relies on the ($\ell = 2^n$)-Decisional Diffie-Hellman Inversion assumption with a 2^n factor blowup in the time component.

Recently, Abdalla, Catalano and Fiore [1] gave two VRF constructions and showed some connections to Identity-Based Encryption [28,8]. In particular, they showed that any IBE scheme with certain properties (e.g., deterministic key generation) implies VRFs, although some of these properties only appear in random oracle constructions of IBE systems.

Chase and Lysyanskaya [12] introduced a concept that they called a *simultable* VRF. Simultable VRFs allow the use of a common reference string (CRS) in order to simulate a proof of the PRF output. Connections to multi-theorem NIZKs were given. We note that reintroducing a CRS removes some of the fundamental challenges in constructing a VRF that we described above.

Brakerski, Goldwasser, Rothblum and Vaikuntanathan [10] introduced a relaxation of VRFs that they called *weak* VRFs. A weak VRF is similar to a VRF except it only needs to be secure if the attacker is allowed to see queries at inputs chosen *randomly.* While this does not meet the full goals of VRFs, the authors showed that weak VRFs imply NIZKs and provided constructions of weak VRFs from simple assumptions.

Applications of VRFs. VRFs have a variety of interesting applications, partially because they allow a short commitment to an exponential number of pseudorandom bits. Abdalla et al. [1] provide a nice summary of applications where VRFs are used as a building block, including resettable zero-knowledge proofs [25], micropayment schemes [26], updatable zero-knowledge databases [22] and verifiable transaction escrow schemes [21], to name a few. It also appears likely to us that suitable VRFs could be a useful alternative in several applications which, as part of the system, output the value of the PRF together with a proof (interactive or non-interactive) that the evaluation was correct and has some additional properties. Examples of this include compact e-cash [11], keyword search [15], set intersection protocols [18], and adaptive oblivious transfer protocols [20].

2 Definition

Definition 1 (Verifiable Random Function). *Let* $F : \{0,1\}^{\mathsf{seed}(\lambda)} \times \{0,1\}^{\mathsf{in}(\lambda)} \to \{0,1\}^{\mathsf{out}(\lambda)}$, *where* $\mathsf{seed}, \mathsf{in}, \mathsf{out}$ *are all polynomials in the security parameter* 1^λ, *be an efficient function. We say that* F *is a* verifiable random function *if there exist algorithms* $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ *such that*

- $\mathsf{Setup}(1^\lambda)$ *outputs a pair of keys* (pk, sk)*;*
- $\mathsf{Prove}_{sk}(x)$ *outputs a pair* $(F_{sk}(x), \pi_{sk}(x))$*, where* $F_{sk}(x)$ *is the function value and* $\pi_{sk}(x)$ *is the proof of correctness; and*
- $\mathsf{Verify}_{pk}(x, y, \pi)$ *verifies that* $y = F_{sk}(x)$ *using the proof* π*.*

Formally, we require the following properties:

1. **Provability:** *For all* $(pk, sk) \in \mathsf{Setup}(1^\lambda)$ *and inputs* $x \in \{0,1\}^{\mathsf{in}(\lambda)}$, *if* $(y, \pi) = \mathsf{Prove}_{sk}(x)$, *then* $\mathsf{Verify}_{pk}(x, y, \pi) = 1$.

2. **Uniqueness:** *For all* $(pk, sk) \in \mathsf{Setup}(1^\lambda)$ *and inputs* $x \in \{0,1\}^{\mathsf{in}(\lambda)}$, *there does not exist a tuple* (y_1, y_2, π_1, π_2) *such that:* *(1)* $y_1 \neq y_2$, *(2)* $\mathsf{Verify}_{pk}(x, y_1, \pi_1) = 1$, *and (3)* $\mathsf{Verify}_{pk}(x, y_2, \pi_2) = 1$.

3. **Pseudorandomness:** *For all p.p.t. distinguishers* $D = (D_1, D_2)$, *there exists a negligible function* μ *such that:*

$$\Pr[(pk, sk) \leftarrow \mathsf{Setup}(1^\lambda);\ (x, s) \leftarrow D_1^{\mathsf{Prove}(\cdot)}(1^\lambda, pk);\ y_0 = F_{sk}(x);$$
$$y_1 \leftarrow \{0,1\}^{\mathsf{out}(\lambda)};\ b \leftarrow \{0,1\};\ b' \leftarrow D_2^{\mathsf{Prove}(\cdot)}(y_b, s)\ :$$
$$b = b' \wedge x \notin S] \leq \frac{1}{2} + \mu(\lambda),$$

where S *is the set of all inputs that* D *queries to its oracle* Prove.

3 Algebraic Settings

We describe a scheme set in bilinear groups of prime order.

Bilinear Groups. Let $\mathbb{G}$ and $\mathbb{G}_T$ be algebraic groups. A *bilinear map* is an efficient mapping $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ which is both: (*bilinear*) for all $g \in \mathbb{G}$ and $a, b \leftarrow \mathbb{Z}$, $e(g^a, g^b) = e(g, g)^{ab}$; and (*non-degenerate*) if g generates $\mathbb{G}$, then $e(g, g) \neq 1$.

3.1 Assumption

We will consider the following previously used assumption.

Assumption 1 (ℓ-Decisional Diffie-Hellman Exponent [7,9]). *Let* $\mathbb{G}, \mathbb{G}_T$ *be groups of prime order* $p \in \Theta(2^\lambda)$. *For all p.p.t. adversaries* $\mathcal{A}$, *there exists a negligible function* μ *such that*

$$\Pr[g, h \leftarrow \mathbb{G};\ a \leftarrow \mathbb{Z}_p;\ y_0 = e(g, h)^{a^\ell};\ y_1 \leftarrow \mathbb{G}_T;\ b \leftarrow \{0,1\};$$
$$b' \leftarrow \mathcal{A}(g, h, g^a, \ldots, g^{a^{\ell-1}}, g^{a^{\ell+1}}, \ldots, g^{a^{2\ell}}, y_b)\ :\ b = b'] \leq \frac{1}{2} + \mu(\lambda).$$

4 VRF Construction from the DDHE Assumption

Setup(1^λ). We describe a system for inputs of length n, a polynomial in 1^λ.[3] The setup algorithm first chooses a bilinear group $\mathbb{G}$ of prime order p. It selects random generators $g, h \in \mathbb{G}$. It next selects random values $u_0, u_1, \ldots, u_n \in \mathbb{Z}_p$ and sets $U_0 = g^{u_0}, U_1 = g^{u_1}, \ldots, U_n = g^{u_n}$. It then sets the keys as:

$$pk = (\mathbb{G}, p, g, h, U_0, \ldots, U_n), \quad sk = (\mathbb{G}, p, g, h, u_0, \ldots, u_n).$$

[3] Due to the fact that n can be polynomial in the security parameter, we can accept inputs of arbitrary length by first applying a collision resistant hash function.

Evaluate(sk, x). For $x \in \{0,1\}^n$, the function F_{sk} evaluates $x = x_1 x_2 \ldots x_n$ as:

$$F_{sk}(x) = e(g^{u_0 \prod_{i=1}^{n} u_i^{x_i}}, h)$$

Prove(sk, x). This algorithm outputs $F_{sk}(x)$ together with a proof π comprised as follows. For $i = 1$ to n, compute $\pi_i = g^{\prod_{j=1}^{i} u_j^{x_j}}$. Next, compute $\pi_0 = g^{u_0 \prod_{j=1}^{n} u_j^{x_j}}$. Output the proof

$$\pi = (\pi_0, \pi_1, \ldots, \pi_n).$$

We observe that this formulation of π is redundant. It is not necessary to include π_i when $x_i = 0$, since in this case, we have $\pi_i = \pi_{i-1}$ (for $i > 1$) and $\pi_1 = g$ (for $i = 1$).

Verify(pk, x, y, π). The first step is to verify that all parts of the input are prop erly encoded group elements; in particular, that the proof $\pi = (\pi_0, \ldots, \pi_n)$ contains legal encodings of elements in $\mathbb{G}$. Next, the proof is verified in a step-by-step manner by checking that

$$e(\pi_1, g) = \begin{cases} e(g, g) & \text{if } x_1 = 0; \\ e(U_1, g) & \text{otherwise.} \end{cases}$$

and then for $i = 2$ to n, it holds that

$$e(\pi_i, g) = \begin{cases} e(\pi_{i-1}, g) & \text{if } x_i = 0; \\ e(\pi_{i-1}, U_i) & \text{otherwise.} \end{cases}$$

and finally that

$$e(\pi_0, g) = e(\pi_n, U_0) \text{ and } e(\pi_0, h) = y.$$

Output 1 if and only if all checks verify.

Efficiency Discussion. The output of the PRF $F_{sk}(\cdot)$ is one element in $\mathbb{G}_T$. As noted above, our representation of π is redundant and can be simplified. For an n-bit input x, the proof π requires at most $\mathbf{ones}(x) + 1 \le n + 1$ elements in $\mathbb{G}$, where $\mathbf{ones}(\cdot)$ counts the number of bits set to 1 in the input. (Individual) verification of the VRF output requires $\mathbf{ones}(x) + 3 \le n + 3$ pairings, if $e(g, g)$ is provided in the public key.

For applications where several VRF outputs need to be verified at the same time, one can apply standard batching techniques [2] to perform N verifications for n-bit inputs at a cost of $O(n)$ total pairing operations. The batch verification algorithm takes as input N tuples of the form $(x_i, y_i = f_{sk}(x_i), \pi_i)$ and outputs 1 if and only if all individual proofs verify, with an error rate of 2^{-k} for security parameter k.

The batching algorithm would first verify the respective group memberships of all y_i and all values in $\pi_i = (\pi_{i,0}, \ldots, \pi_{i,n})$. It then chooses random $r_0, \ldots, r_N \in \{0,1\}^k$ and verifies that:

$$e(\prod_{i=1}^{N} \pi_{i,1}^{r_i}, g) = e(g^{\sum_{i=1}^{N}(1-x_i) r_i} \cdot U_1^{\sum_{i=1}^{N} x_i r_i}, g)$$

and then for $t = 2$ to n, it holds that

$$e(\prod_{i=1}^{N} \pi_{i,t}^{r_i}, g) = e(\prod_{i=1}^{N} \pi_{i,t-1}^{(1-x_i)r_i}, g) \cdot e(\prod_{i=1}^{N} \pi_{i,t-1}^{x_i r_i}, U_t)$$

To see the above, recall that $e(1, g) = 1$. Finally, we check that

$$e(\prod_{i=1}^{N} \pi_{i,0}^{r_i}, g \cdot h) = e(\prod_{i=1}^{N} \pi_{i,n}^{r_i}, U_0) \cdot \prod_{i=1}^{N} y_i^{r_i}.$$

Output 1 if and only if all checks verify.

5 Proof of DDHE VRF

Theorem 2. *The VRF construction in Section 4 is secure with respect to Definition 1 under the ℓ-DDHE assumption.*

Proof. The *provability* property is verifiable in a straightforward manner from the construction. The *uniqueness* property also follows easily from the group structure; that is, for any input, there is only one group element in $\mathbb{G}$ that is the valid output and moreover, that it is not possible (even for an unbounded adversary) to devise a valid proof for another element.

Showing *pseudorandomness* will require more work. To show pseudorandomness, we will employ a proof technique from the Waters IBE system [29] that allows us to partition the inputs into two sets: those the simulator can properly answer and those we hope the adversary chooses as a challenge. The main difficulty in adapting this technique is that Waters was able to manipulate the randomness in the IBE keys during simulation, whereas we are now dealing with a deterministic function evaluation. Nevertheless, by strengthening the complexity assumption and making subtle changes throughout the proof, we are able to complete the argument.

Suppose there is a p.p.t. distinguisher D which makes Q Prove queries in the pseudorandomness game and succeeds with probability $\frac{1}{2} + \epsilon$. Then we show how to use D to create an adversary $\mathcal{B}$ which breaks the ℓ-DDHE assumption with probability $\frac{1}{2} + \frac{3\epsilon}{64Q(n+1)}$, where $\ell = 4Q(n + 1)$ and n is the bit length of the VRF input.

On input $(\mathbb{G}, p, g, h, g^a, \ldots, g^{a^{\ell-1}}, g^{a^{\ell+1}}, \ldots, g^{a^{2\ell}}, Y)$, our ℓ-DDHE solver $\mathcal{B}$ proceeds as:

Setup. The simulator first sets an integer $m = 4Q$ and chooses an integer, k, uniformly at random between 0 and n. Recall that Q is the number of queries made by the distinguisher and n is the bit length of the VRF input. It then chooses random integers $r_1, \ldots, r_n, r'$ between 0 and $m - 1$. Additionally, the simulator chooses random values $s_1, \ldots, s_n, s' \in \mathbb{Z}_p$. These values are all kept internal to the simulator. Intuitively, the r values will be used to embed the challenge, while the s values will be used as blinding factors to present the proper distribution to the distinguisher.

For $x \in \{0,1\}^n$, let $X \subseteq \{1, \dots, n\}$ be the set of the all i for which $x_i = 1$. To ease our analysis, we define the functions:

$$C(x) = m(1+k) + r' + \sum_{i \in X} r_i \ , \ \hat{C}(x,i) = \sum_{j=1}^{i} x_j r_j$$

$$J(x) = s' \prod_{i \in X} s_i \ , \ \hat{J}(x,i) = \prod_{j=1}^{i} s_j^{x_j}$$

For inputs $x \in \{0,1\}^n$, we define the binary function

$$K(x) = \begin{cases} 0 & \text{if } r' + \sum_{j=1}^{n} x_j r_j \equiv 0 \mod m; \\ 1 & \text{otherwise.} \end{cases}$$

The simulator sets $U_0 = (g^{a^{m(1+k)+r'}})^{s'}$ and $U_i = (g^{a^{r_i}})^{s_i}$ for $i = 1$ to n. It outputs the public key as $(\mathbb{G}, p, g, h, U_0, \dots, U_n)$, where implicitly the secret key contains the values $u_0 = a^{m(1+k)+r'} s'$ and $\{u_i = a^{r_i} s_i\}_{i \in [1,n]}$.

Prove. The distinguisher, D, will ask for VRF evaluations and proofs. On query input x, the simulator first checks if $C(x) = \ell$ and aborts if this is true. Otherwise, it outputs the value

$$F(x) = e((g^{a^{C(x)}})^{J(x)}, h).$$

It also computes $\pi_0 = (g^{a^{C(x)}})^{J(x)}$ and $\pi_i = (g^{a^{\hat{C}(x,i)}})^{\hat{J}(x,i)}$ for $i = 1$ to n, and then outputs the proof $\pi = (\pi_0, \pi_1, \dots, \pi_n)$.

Given the above settings, it is easy to verify that for any value of $x \in \{0,1\}^n$:

1. The maximum value of $C(x)$ is $m(1+n) + (1+n)(m-1) < 2m(1+n) = 2\ell$.
2. For any $i \in [1,n]$, the maximum value of $\hat{C}(x,i)$ is $(m-1)n < m(n+1) = \ell$.

Thus, if $C(x) \neq \ell$, then the simulator can always correctly answer all parts of the query.

Response. Eventually D will provide a challenge input x^*. If $C(x^*) = \ell$, $\mathcal{B}$ will return the value Y. When D responds with a guess b', $\mathcal{B}$ will also output b' as its ℓ-DDHE guess. If $C(x^*) \neq \ell$, $\mathcal{B}$ outputs a random bit as its ℓ-DDHE guess.

This ends our description of ℓ-DDHE adversary $\mathcal{B}$.

A Series of Games Analysis. We now argue that any successful adversary D against our scheme will have success in the game presented by $\mathcal{B}$. To do this, we first define a sequence of games, where the first game models the real security game and the final game is exactly the view of the adversary when interacting with $\mathcal{B}$. We then show via a series of claims that if D is successful against Game j, then it will also be successful against Game $j+1$.

Game 1: This game is defined to be the same as the VRF security game in Definition 1.

Game 2: The same as Game 1, with the exception that we keep a record of each query made by D, which we'll denote as $\overrightarrow{x} = (x^{(1)}, \ldots, x^{(Q)}, x^*)$, where x^* is the challenge input. At the end of the game, we set $m = 4Q$ and choose random integers $\overrightarrow{r} = (r_1, \ldots, r_n, r')$ between 0 and $m - 1$ and a random integer k between 0 and n. We define the regular abort indicator function:

$$\tau(\overrightarrow{x}, \overrightarrow{r}, k) = \begin{cases} 1 & \text{if } r' + \sum_{j=1}^{n} x_j^* r_j \neq m(n-k) \ \bigvee_{i=1}^{Q} \ K(x^{(i)}) = 0; \\ 0 & \text{otherwise.} \end{cases}$$

This function $\tau(\overrightarrow{x}, \overrightarrow{r}, k)$ evaluates to 0 if the queries $\overrightarrow{x}$ will not cause a regular abort for the given choice of simulation values $\overrightarrow{r}, k$. Consider the probability over all simulation values for the given set of queries $\overrightarrow{x}$ as $\zeta(\overrightarrow{x}) = \Pr_{\overrightarrow{r},k}[\tau(\overrightarrow{x}, \overrightarrow{r}, k) = 0]$.

As in [29], the simulator estimates $\zeta(\overrightarrow{x})$ as ζ' by evaluating $\tau(\overrightarrow{x}, \overrightarrow{r}, k)$ with fresh random $\overrightarrow{r}, k$ values a total of $O(\epsilon^{-2} \ln(\epsilon^{-1}) \zeta_{\mathsf{min}}^{-1} \ln(\zeta_{\mathsf{min}}^{-1}))$ times. This does not require running the distinguisher again.

D's success in the game is then determined as follows:

1. *Regular Abort.* If $\tau(\overrightarrow{x}, \overrightarrow{r}, k) = 1$, then flip a coin $b \in \{0, 1\}$ and say that D wins if $b = 0$ and loses otherwise.
2. *Balancing (Artificial) Abort.*[4] Let $\zeta_{\mathsf{min}} = \frac{1}{8Q(n+1)}$ as derived from Claim 5. If $\zeta' \geq \zeta_{\mathsf{min}}$, $\mathcal{B}$ will abort with probability $\frac{\zeta' - \zeta_{\mathsf{min}}}{\zeta'}$ (not abort with probability $\frac{\zeta_{\mathsf{min}}}{\zeta'}$). If it aborts, flip a coin $b \in \{0, 1\}$ and say that D wins if $b = 0$ and loses otherwise.
3. Otherwise, D wins if it correctly guessed b' as in the real security game.

Game 3: The same as Game 2, with the exception that $\mathcal{B}$ tests if any abort conditions are satisfied, with each new query, and if so, follows the abort procedure immediately (i.e., flips a coin $b \in \{0, 1\}$ and says that D wins if $b = 0$.)

Game 3 is exactly the view of D when interacting with $\mathcal{B}$. We will shortly prove that if D succeeds in Game 1 with probability $\frac{1}{2} + \epsilon$, then it succeeds in Game 3 with probability $\geq \frac{1}{2} + \frac{3\epsilon}{64Q(n+1)}$.

Establishing Three Claims about the Probability of Aborting. Before doing so, we establish one claim which was used above and two claims which will be needed shortly. Our first claim helps us establish a minimum probability that a given set of queries do not cause a *regular* abort. We use this minimum during our balancing abort in Game 2, to "even out" the probability of an abort over all

[4] In Waters [29], this is called the artificial abort. Recently, Bellare and Ristenpart provided an analysis of the Waters' IBE without the artificial abort [3]. We could use their techniques here for an alternative, tighter analysis, but we would need to expand the input size of our ℓ-DDHE assumption by a factor of $1/\epsilon$, where the distinguisher's advantage is $1/2 + \epsilon$.

possible queries. In the next two claims, we employ Chernoff Bounds to establish upper and lower bounds for *any* abort (regular or balancing) for any set of queries. The latter two claims will be used in the analysis of D's probability of success in Game 2.

Claim. Let $\zeta_{\mathsf{min}} = \frac{1}{8Q(n+1)}$. For any query vector $\overrightarrow{x}$, $\zeta(\overrightarrow{x}) \geq \zeta_{\mathsf{min}}$.

Proof of Claim 5 is similar to a related argument in [29] and appears in Appendix A.

Claim. For any set of queries $\overrightarrow{x}$, the probability that there is an abort (i.e., regular or balancing) is $\geq 1 - \zeta_{\mathsf{min}} - \frac{3}{8}\zeta_{\mathsf{min}}\epsilon$.

Proof. Let $\zeta_x = \zeta(\overrightarrow{x})$, as defined in Section 5, be the probability that a set of queries $\overrightarrow{x}$ do not cause a regular abort. In Game 2, $T = O(\epsilon^{-2}\ln(\epsilon^{-1})\zeta_{\mathsf{min}}^{-1}\ln(\zeta_{\mathsf{min}}^{-1}))$ samples are taken to approximate this value as ζ'_x. By Chernoff Bounds, we have that for all $\overrightarrow{x}$,

$$\Pr[T\zeta'_x < T\zeta_x(1-\frac{\epsilon}{8})] < e^{-[128\epsilon^{-2}\ln((\epsilon/8)^{-1})\zeta_{\mathsf{min}}^{-1}\ln(\zeta_{\mathsf{min}}^{-1})(\zeta_{\mathsf{min}})(\epsilon/8)^2/2]},$$

which reduces to

$$\Pr[\zeta'_x < \zeta_x(1-\frac{\epsilon}{8})] < \zeta_{\mathsf{min}}\frac{\epsilon}{8}.$$

Recall that for a measured ζ'_x an artificial abort will not happen with probability $\zeta_{\mathsf{min}}/\zeta'_x$. The probability of aborting is

$$\begin{aligned}\Pr[\mathsf{abort}] = 1 - \Pr[\overline{\mathsf{abort}}] &= 1 - \Pr[\overline{\mathsf{RA}}]\Pr[\overline{\mathsf{AA}}] = 1 - \zeta_x \Pr[\overline{\mathsf{AA}}] \\ &\geq 1 - \zeta_x(\zeta_{\mathsf{min}}\frac{\epsilon}{8} + \frac{\zeta_{\mathsf{min}}}{\zeta_x(1-\epsilon/8)}) \\ &\geq 1 - (\zeta_{\mathsf{min}}\frac{\epsilon}{8} + \frac{\zeta_{\mathsf{min}}}{1-\epsilon/8}) \\ &\geq 1 - (\frac{\zeta_{\mathsf{min}}\epsilon}{8} + \zeta_{\mathsf{min}}(1+\frac{2\epsilon}{8})) \\ &\geq 1 - \zeta_{\mathsf{min}} - \zeta_{\mathsf{min}}\frac{3\epsilon}{8}\end{aligned}$$

Claim. For any set of queries $\overrightarrow{x}$, the probability that there is no abort (i.e., regular or balancing) is $\geq \zeta_{\mathsf{min}} - \frac{1}{4}\zeta_{\mathsf{min}}\epsilon$.

Proof. Let $\zeta_x = \zeta(\overrightarrow{x})$, as defined in Section 5, be the probability that a set of queries $\overrightarrow{x}$ do not cause a regular abort. In Game 2, $T = O(\epsilon^{-2}\ln(\epsilon^{-1})\zeta_{\mathsf{min}}^{-1}\ln(\zeta_{\mathsf{min}}^{-1}))$ samples are taken to approximate this value as ζ'_x. By Chernoff Bounds, we have that for all $\overrightarrow{x}$,

$$\Pr[T\zeta'_x > T\zeta_x(1+\frac{\epsilon}{8})] < e^{-[256\epsilon^{-2}\ln((\epsilon/8)^{-1})\zeta_{\mathsf{min}}^{-1}\ln(\zeta_{\mathsf{min}}^{-1})](\zeta_{\mathsf{min}})(\epsilon/8)^2/4]},$$

which reduces to

$$\Pr[\zeta'_x > \zeta_x(1+\frac{\epsilon}{8})] < \zeta_{\mathsf{min}}\frac{\epsilon}{8}.$$

The probability of not aborting is equal to the probability of not regular aborting times the probability of not artificial aborting. Recall that for a measured ζ'_x an artificial abort (AA) will not happen with probability $\zeta_{\mathsf{min}}/\zeta'_x$. Therefore, for any x, the $\Pr[\overline{\mathsf{AA}}] \geq (1 - \frac{\zeta_{\mathsf{min}}\epsilon}{8})\frac{\zeta_{\mathsf{min}}}{\zeta_x(1+\epsilon/8)}$. It follows that

$$\Pr[\overline{\mathsf{abort}}] \geq \zeta_x(1 - \frac{\zeta_{\mathsf{min}}\epsilon}{8})\frac{\zeta_{\mathsf{min}}}{\zeta_x(1+\epsilon/8)} \geq \zeta_{\mathsf{min}}(1 - \frac{\epsilon}{8})^2 \geq \zeta_{\mathsf{min}}(1 - \frac{1}{4}\epsilon).$$

Analyzing D's Probability of Success in the Games. Define D's probability of success in Game x as $\mathbf{Adv}_D[\text{Game } x]$. We reason about the probability of D's success in the series of games as follows.

Lemma 1. *If* $\mathbf{Adv}_D[\text{Game 1}] = \frac{1}{2} + \epsilon$, *then* $\mathbf{Adv}_D[\text{Game 2}] \geq \frac{1}{2} + \frac{3\cdot\epsilon}{64Q(n+1)}$.

Proof. We begin by observing that $\mathbf{Adv}_D[\text{Game 2}]$ is

$$\begin{aligned}
&= \mathbf{Adv}_D[\text{Game 2}|\mathsf{abort}] \cdot \Pr[\mathsf{abort}] + \mathbf{Adv}_D[\text{Game 2}|\overline{\mathsf{abort}}] \cdot \Pr[\overline{\mathsf{abort}}] && (1)\\
&= \frac{1}{2}\Pr[\mathsf{abort}] + \mathbf{Adv}_D[\text{Game 2}|\overline{\mathsf{abort}}] \cdot \Pr[\overline{\mathsf{abort}}] && (2)\\
&= \frac{1}{2}\Pr[\mathsf{abort}] + \Pr[b = b'|\overline{\mathsf{abort}}] \cdot \Pr[\overline{\mathsf{abort}}] && (3)\\
&= \frac{1}{2}\Pr[\mathsf{abort}] + \Pr[b = b'] \cdot \Pr[\overline{\mathsf{abort}}|b = b'] && (4)\\
&= \frac{1}{2}\Pr[\mathsf{abort}] + (\frac{1}{2} + \epsilon) \cdot \Pr[\overline{\mathsf{abort}}|b = b'] && (5)\\
&\geq \frac{1}{2}(1 - \zeta_{\mathsf{min}} - s_1) + (\frac{1}{2} + \epsilon)(\zeta_{\mathsf{min}} - s_2) && (6)\\
&\geq \frac{1}{2} + \epsilon \cdot \zeta_{\mathsf{min}} - (s_1 + s_2) && (7)\\
&= \frac{1}{2} + \frac{3 \cdot \epsilon \cdot \zeta_{\mathsf{min}}}{8} && (8)\\
&= \frac{1}{2} + \frac{3 \cdot \epsilon}{64Q(n+1)} && (9)
\end{aligned}$$

Equation 2 follows from the fact that, in the case of abort, D's success is determined by a coin flip. It would be very convenient if we could claim that $\mathbf{Adv}_D[\text{Game 2} \mid \overline{\mathsf{abort}}] = \mathbf{Adv}_D[\text{Game 1}]$, but unfortunately, this is false. The event that D wins Game 2 and the event of an abort are not independent; however, we have inserted the balancing abort condition in the attempt to lessen the dependence between these events. Equation 3 simply states that, when there is no abort, D wins if and only if it guesses correctly. Equation 4 follows from Bayes' Theorem. In Equation 5, we observe that $\Pr[b = b']$ is exactly D's success in Game 1.

Now, the purpose of our balancing abort is to even the probability of aborting, for all queries of D, to be roughly ζ_{min}. This will also get rid of the conditional dependence on $b = b'$. There will be a small error, which must be taken into account. Suppose that $\Pr[\mathsf{abort}] \geq 1 - \zeta_{\mathsf{min}} - s_1$ and $\Pr[\overline{\mathsf{abort}}] \geq \zeta_{\mathsf{min}} - s_2$, which must hold for some error values s_1, s_2, then we derive Equation 6. Algebraic manipulation and recalling that $\epsilon \leq \frac{1}{2}$, brings us to Equation 7.

We set $\zeta_{\mathsf{min}} = \frac{1}{8Q(n+1)}$ from Claim 5. We know, for all queries, that $\Pr[\mathsf{abort}] \geq 1 - \zeta_{\mathsf{min}} - s_1$ where $s_1 = \frac{3}{8}\zeta_{\mathsf{min}}\epsilon$ from Claim 5 and that $\Pr[\overline{\mathsf{abort}}] \geq \zeta_{\mathsf{min}} - s_2$ where $s_2 = \frac{1}{4}\zeta_{\mathsf{min}}\epsilon$ from Claim 5. Plugging these values into Equations 7 and 8 establishes the lemma.

Lemma 2. $\mathbf{Adv}_D[\text{Game 3}] = \mathbf{Adv}_D[\text{Game 2}]$.

Proof. We make the explicit observation that these games are equivalent by observing that their only difference is the time at which the regular aborts occur. The artificial abort stage is identical. All public parameters, evaluations and proofs have the same distribution up to the point of a possible abortion. In Game 2, the simulator receives all the queries $\overrightarrow{x}$, then checks if $\tau(\overrightarrow{x}, \overrightarrow{r}, k) = 1$ and aborts, taking a random guess, if so. In Game 3, the simulator checks with each new query x if $K(x) = 0$, which implies that the ending τ evaluation will be 1, and aborts, taking a random guess, if so. Therefore, the output distributions will be the same.

Tightness of the Reduction. Using the (asymptotically) tighter analysis techniques of Hofheinz and Kiltz [19], the $1/n$ factor loss in our reduction, that occurs due to the Balancing Abort in Game 2, could be reduced to $1/\sqrt{n}$. Since the $1/Q$ factor loss is the dominating term in our concrete analysis, this improved analysis may provide only modest gains in practice.

6 Conclusion and Open Directions

Verifiable random functions are an interesting and useful cryptographic primitive, but to date, all known constructions for exponentially-large message spaces required interactive complexity assumptions or their concrete security degraded by an exponential factor. In this work, we presented an efficient construction which can handle arbitrarily-large inputs (by first applying a collision-resistant hash function) based on the ℓ-Decisional Diffie-Hellman Exponent assumption. Our security proof used techniques similar to the Waters IBE [29], where we partitioned the input space into those for which we can provide a proof and those which we cannot. We then showed that with non-negligible probability, the adversary will only query us on inputs for which we can provide proofs, except for the challenge query, for which the proof is unknown. The main technical difference when applying Waters' proof techniques is we must move from an additive to a multiplicative structure, work without randomness in the output to manipulate during the reduction, and operate under a different complexity assumption. Fortunately, we were still able to properly simulate access to an exponentially-large input space using a complexity assumption with only a polynomial-size input.

We believe this work is an important step towards better understanding how to construct verifiable random functions. It leaves open many interesting questions.

First, it would be interesting to improve on the efficiency of our construction, especially by realizing a seed, proof size and verification time that are sublinear in the bit-length of the input. Second, one would like to know if it is possible to realize a VRF under a complexity assumption with a fixed input size, such as Decisional Diffie-Hellman. If this is not possible, perhaps one can show that a q-based assumption (with at least a polynomial number of terms) is inherently necessary. Finally, it would be interesting to see if this new construction allows for any additional applications of VRFs or if it can be used to reduce the overall complexity assumptions required by any constructions using VRFs.

Acknowledgments

We thank Qiong Huang for helpful comments.

References

1. 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)
2. 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)
3. 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)
4. Blum, M., Feldman, P., Micali, S.: Non-interactive zero-knowledge and its applications (extended abstract). In: STOC, pp. 103–112 (1988)
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.: Short signatures without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 382–400. Springer, Heidelberg (2004)
7. 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)
8. Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
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. Brakerski, Z., Goldwasser, S., Rothblum, G.N., Vaikuntanathan, V.: Weak verifiable random functions. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 558–576. Springer, Heidelberg (2009)
11. Camenisch, J., Hohenberger, S., Lysyanskaya, A.: Compact E-Cash. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 302–321. Springer, Heidelberg (2005)
12. Chase, M., Lysyanskaya, A.: Simulatable VRFs with applications to multi-theorem NIZK. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 303–322. Springer, Heidelberg (2007)

13. Dodis, Y.: Efficient construction of (distributed) verifiable random functions. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 1–17. Springer, Heidelberg (2003)
14. Dodis, Y., Yampolskiy, A.: A Verifiable Random Function with Short Proofs an Keys. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 416–431. Springer, Heidelberg (2005)
15. 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)
16. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. Journal of the ACM 33(4), 792–807 (1986)
17. Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: STOC 1989, pp. 25–32 (1989)
18. 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)
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. 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)
21. Jarecki, S., Shmatikov, V.: Handcuffing big brother: an abuse-resilient transaction escrow scheme. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 590–608. Springer, Heidelberg (2004)
22. Liskov, M.: Updatable zero-knowledge databases. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 174–198. Springer, Heidelberg (2005)
23. Lysyanskaya, A.: Unique signatures and verifiable random functions from the DH-DDH separation. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 597–612. Springer, Heidelberg (2002)
24. Micali, S., Rabin, M.O., Vadhan, S.P.: Verifiable random functions. In: Symposium on Foundations of Computer Science (FOCS), pp. 120–130. IEEE Computer Society, Los Alamitos (1999)
25. Micali, S., Reyzin, L.: Soundness in the public-key model. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 542–565. Springer, Heidelberg (2001)
26. Micali, S., Rivest, R.L.: Micropayments revisited. In: Preneel, B. (ed.) CT-RSA 2002. LNCS, vol. 2271, pp. 149–163. Springer, Heidelberg (2002)
27. Naor, M., Reingold, O.: Number-theoretic constructions of efficient pseudo-random functions. Journal of the ACM 51(2), 231–262 (2004)
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 (1984)
29. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 320–329. Springer, Heidelberg (2005)

Appendix

A Proof of Claim 5

Proof. In other words, the probability of the simulation *not* triggering a general abort is at least ζ_{min}. This analysis follows that of [29], which we reproduce here for completeness. Without loss of generality, we can assume the adversary always makes the maximum number of queries Q (since the probability of not aborting increases with fewer queries). Fix an arbitrary $\overrightarrow{x} = (x^{(1)}, \ldots, x^{(Q)}, x^*) \in \{0,1\}^n$. Then, with the probability over the choice of $\overrightarrow{r}, k$, we have that $\Pr[\overline{\mathsf{abort}} \text{ on } \overrightarrow{x}]$ is

$$= \Pr[\bigwedge_{i=1}^{Q} K(x^{(i)}) = 1 \ \wedge \ r' + \sum_{j=1}^{n} x_j^* r_j = m(n-k)] \tag{10}$$

$$= (1 - \Pr[\bigvee_{i=1}^{Q} K(x^{(i)}) = 0]) \Pr[r' + \sum_{j=1}^{n} x_j^* r_j = m(n-k) | \bigwedge_{i=1}^{Q} K(x^{(i)}) = 1] \tag{11}$$

$$\geq (1 - \sum_{i=1}^{Q} \Pr[K(x^{(i)}) = 0]) \Pr[r' + \sum_{j=1}^{n} x_j^* r_j = m(n-k) | \bigwedge_{i=1}^{Q} K(x^{(i)}) = 1] \tag{12}$$

$$= (1 - \frac{Q}{m}) \cdot \Pr[r' + \sum_{j=1}^{n} x_j^* r_j = m(n-k) \mid \bigwedge_{i=1}^{Q} K(x^{(i)}) = 1] \tag{13}$$

$$= \frac{1}{n+1} \cdot (1 - \frac{Q}{m}) \cdot \Pr[K(x^*) = 0 \mid \bigwedge_{i=1}^{Q} K(x^{(i)}) = 1] \tag{14}$$

$$= \frac{1}{n+1} \cdot (1 - \frac{Q}{m}) \cdot \frac{\Pr[K(x^*) = 0] \cdot \Pr[\bigwedge_{i=1}^{Q} K(x^{(i)}) = 1] \mid K(x^*) = 0]}{\Pr[\bigwedge_{i=1}^{Q} K(x^{(i)}) = 1]]} \tag{15}$$

$$\geq \frac{1}{(n+1)m} \cdot (1 - \frac{Q}{m}) \cdot \Pr[\bigwedge_{i=1}^{Q} K(x^{(i)}) = 1] \mid K(x^*) = 0] \tag{16}$$

$$= \frac{1}{(n+1)m} \cdot (1 - \frac{Q}{m}) \cdot (1 - \Pr[\bigvee_{i=1}^{Q} K(x^{(i)}) = 0] \mid K(x^*) = 0]) \tag{17}$$

$$\geq \frac{1}{(n+1)m} \cdot (1 - \frac{Q}{m}) \cdot (1 - \sum_{i=1}^{Q} \Pr[K(x^{(i)}) = 0] \mid K(x^*) = 0]) \tag{18}$$

$$= \frac{1}{(n+1)m} \cdot (1 - \frac{Q}{m})^2 \tag{19}$$

$$\geq \frac{1}{(n+1)m} \cdot (1 - \frac{2Q}{m}) \tag{20}$$

$$= \frac{1}{8Q(n+1)} \tag{21}$$

Equations 13 and 16 derive from $\Pr[K(x) = 0] = \frac{1}{m}$ for any query x. Equation 14 gets a factor of $\frac{1}{n+1}$ from the simulator taking a guess of k. Equation 15 follows from Bayes' Theorem. Equation 19 follows from the pairwise independence of the probabilities that $K(x) = 0, K(x') = 0$ for any pair of queries $x \neq x'$, since they will differ in at least one random r_j value. Equation 21 follows from our setting of $m = 4Q$.

Adaptive Trapdoor Functions and Chosen-Ciphertext Security

Eike Kiltz[1], Payman Mohassel[2], and Adam O'Neill[3]

[1] CWI, Amsterdam, Netherlands
kiltz@cwi.nl
[2] University of Calgary, AB, Canada
pmohasse@cpsc.ucalgary.ca
[3] Georgia Institute of Technology, Atlanta, GA, USA
amoneill@cc.gatech.edu

Abstract. We introduce the notion of *adaptive* trapdoor functions (ATDFs); roughly, ATDFs remain one-way even when the adversary is given access to an inversion oracle. Our main application is the black-box construction of chosen-ciphertext secure public-key encryption (CCA-secure PKE). Namely, we give a black-box construction of CCA-Secure PKE from ATDFs, as well as a construction of ATDFs from correlation-secure TDFs introduced by Rosen and Segev (TCC '09). Moreover, by an extension of a recent result of Vahlis (TCC '10), we show that ATDFs are strictly *weaker* than the latter (in a black-box sense). Thus, adaptivity appears to be the weakest condition on a TDF currently known to yield the first implication.

We also give a black-box construction of CCA-secure PKE from a natural extension of ATDFs we call *tag-based* ATDFs that, when applied to our constructions of the latter from either correlation-secure TDFs, or lossy TDFs introduced by Peikert and Waters (STOC '08), yield precisely the CCA-secure PKE schemes in these works. This helps to unify and clarify their schemes. Finally, we show how to realize tag-based ATDFs from an assumption on RSA inversion not known to yield correlation-secure TDFs.

1 Introduction

Historically, the notion of one-way trapdoor functions (OW-TDFs) has played a central role in the study of cryptographic protocols, in particular for semantically-secure public-key encryption (PKE); see e.g. [23,40,4]. However, it is well-known that semantic security alone is not sufficient in many applications; rather, encryption must be secure against *active* adversaries, say, who can inject packets into the network and observe decryptions or actions taken based on them. As a result, resistance to so-called *chosen-ciphertext attacks* (CCA) [36] has become the "gold standard" for security of PKE.

But, whereas there is a simple, black-box construction of semantically secure PKE from OW-TDFs [22], the same is not true of CCA-secure PKE. Instead, early constructions were based on generic non-interactive zero-knowledge

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 673–692, 2010.

proofs [32]. This calls into question the applicability of the TDF concept in the design of CCA-secure PKE. Indeed, the most successful approach for designing practical CCA-secure PKE schemes so far has been based on specific number theoretic assumptions (e.g., [18,24]) and algebraic primitives such as hash proof systems [17] or algebraic set systems [16], which bypass TDFs. However, Peikert and Waters [35], and subsequently Rosen and Segev [37], recently introduced novel strengthenings to the notion of OW-TDFs and showed that these *do* imply simple, black-box constructions of CCA-secure PKE.

Still, we find an underlying "theory" of such stronger TDFs and their relation to CCA-secure PKE lacking. To this end we put forth a notion of *adaptive* trapdoor functions and study its relations to CCA-secure PKE. Surprisingly, we find that adaptivity, a seemingly fundamental notion in the context of chosen-ciphertext security, serves to weaken the assumptions on a TDF needed to imply black-box CCA-secure PKE, as well as to unify and clarify the schemes of [35,37]. Moreover, it leads to new ones, realized from assumptions not known to imply the notions of [35,37].

1.1 Our Contributions

Adaptive trapdoor functions. The central notion we introduce are adaptive trapdoor functions (ATDFs). Loosely speaking, ATDFs remain one-way even when the adversary is given access to an inversion oracle, which it may query on points other than its challenge. We also introduce a natural extension we call tag-based adaptive trapdoor functions (TB-ATDFs), which in addition to the normal input also take a tag. For TB-ATDFs, the adversary may query its oracle on any point, but on a tag other than the challenge one. These notions are quite simple and intuitive but to the best of our knowledge have not appeared before. (There have, however, been similar notions that we discuss later.)

CCA-Secure PKE from ATDFs. As our first result, we give black-box constructions of CCA-secure PKE from both ATDFs and TB-ATDFs. While constructing CCA-secure PKE from TB-ATDFs is straightforward, constructing the former from ATDFs turns out to be more subtle. We apply the classical construction of one-bit PKE using the hardcore bit of the ATDF [8], but it is important here that the ciphertext not contain the message xor'ed with the latter; rather the message is encrypted *as* the hardcore bit itself. By a recent result of Myers and Shelat [30], this construction implies a black-box many-bit scheme as well. On the other hand, hybrid encryption permits a much more efficient direct construction of such a scheme in the case that the ATDF is a permutation or has linearly many simultaneous hardcore bits.

Construction of ATDFs. In the random oracle model [6], the notions of ATDF and TDF are equivalent.[1] To construct ATDFs in the standard model,

[1] For example, a TDF defined as $f(x) := (g(x), H(x))$ is adaptive one-way if TDF g is one-way and H is modeled as a random oracle.

we examine the relation of ATDFs and TB-ATDFs to the recently-introduced notions of correlated-product TDFs (CP-TDFs) [37] and lossy TDFs (LTDFs) [35]. Intuitively, CP-TDFs remain one-way even if the adversary sees many independent instances of the TDF evaluated on the same input, and LTDFs are TDFs whose description is indistinguishable from that of a function that loses information about its input (i.e., has a bounded range). Inspired by the constructions of CCA-secure PKE in [35,37] (which are based on earlier work by Dolev et al. [19]), we show simple, black-box constructions of both ATDFs and TB-ATDFs from CP-TDFs. Since as shown in [37], LTDFs imply the latter,[2] this also gives us ATDFs and TB-ATDFs from LTDFs. However, we show that ATDFs and TB-ATDFs allow much more efficient direct constructions using an all-but-one TDF (ABO-TDF) [35] as well.

Notably, when we apply our general construction of CCA-secure PKE to our constructions of TB-ATDFs from CP-TDFs and lossy+ABO-TDFs, what we obtain are precisely CCA-secure PKE schemes of [37] and [35], respectively. This means that these works were implicitly constructing TB-ATDFs, and that the latter "abstracts out" a particular aspect of their constructions not formalized before. This unifies and clarifies their schemes from a conceptual standpoint and also leads to optimized constructions.

A BLACK-BOX SEPARATION. Very recently, Vahlis [39] showed that there is no black-box construction of CP-TDFs from OW-TDFs. We observe here that his result extends to rule out a black-box construction of the former from ATDFs as well, by using the same "breaking" oracle. (This does not immediately rule out a black-box construction of CP-TDFs from TB-ATDFs, but we also rule this out by giving a construction of TB-TDFs from exponentially-hard ATDFs; the latter is separated from CP-TDFs by our extension of Vahlis's result as well.) Combined with the above-mentioned constructions, this means that, surprisingly, ATDFs and TB-ATDFs are *strictly weaker* than CP-TDFs and LTDFs. The relations between the different primitives are pictured in Figure 1. The figure also contains some related existing notions discussed below.

TB-ATDF FROM II-RSA. Finally, we show that TB-ATDFs are realizable from specific assumptions not known to imply CP-TDFs. Namely, we consider the "instance-independent" RSA assumption (II-RSA) introduced (in a more general form) by Paillier and Villar [33]. Roughly, our assumption says that solving an RSA challenge $y = x^e \bmod N$ remains hard even when the adversary is given access to an inversion oracle that on input (y', e') returns $y'^{1/e'} \bmod N$, where $e \neq e'$ are primes. We show that II-RSA gives rise to a TB-ATDF. This also leads to a very efficient CCA-secure RSA-based PKE scheme in the standard model (though based on an interactive, non-standard assumption).

[2] The original construction of [37] assumes "sufficient" lossiness; this result was recently refined by Mol and Yilek [29], who showed that losing a non-negligible fraction of a single bit suffices.

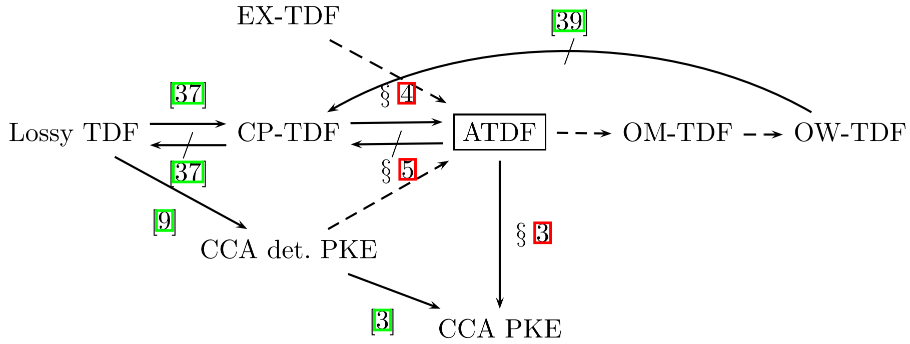


Fig. 1. Relations between the various security notions on trapdoor functions and CCA-secure PKE, centered around our new notion of adaptive trapdoor functions (ATDF). → is an implication while $\not\rightarrow$ is a black-box separation. Dashed lines indicate trivial implications mentioned in the introduction. The considered security notions for TDFs are: extractable TDF (EX-TDF), lossy TDF, correlated-product TDF (CP-TDF), one-more TDF (OM-TDF), one-way TDF (OW-TDF), and CCA-secure deterministic PKE.

1.2 Related Work

Related notions. Pandey et al. [34] introduced a notion they called "adaptive one-way functions," although their notion would be more accurately referred to as adaptive *tag-based* one-way functions. Besides the obvious difference of not having a trapdoor (the inversion oracle in their security experiment is unbounded), their notion differs from ours in that *it does not have a public key.* This is crucial for the applications of [34] to non-malleable commitment but also makes it much harder to construct. Indeed, they are not known to be realizable based on any standard assumptions.

Bellare et al. [5] made an earlier "adaptive assumption" on RSA, namely the One-More RSA assumption. A straightforward formalization of this security property to "one-more TDFs" (OM-TDFs)[3] yields a weaker primitive than ATDFs. In particular, it seems difficult based on the state-of-the-art to give a black-box construction of CCA-secure PKE (or ATDFs) from OM-TDFs. In [12], Canetti and Dakdouk define the notion of extractable trapdoor functions (EX-TDFs), which essentially says that no efficient adversary can compute $f(x)$ without "knowing" x (similar to the notion of plaintext-awareness for PKE [7]). This notion implies ATDFs but unfortunately no instantiation of EX-TDFs based on standard assumption is known (the authors only provide constructions of extractable one-way functions, without a trapdoor).

In another line of work with very different motivation, Bellare et al. [2] introduced a strengthening to OW-TDF they called "deterministic encryption", which includes a CCA-secure variant. CCA-secure deterministic encryption (secure

[3] Informally, a TDF is *one-more secure* if no efficient adversary can invert the TDF on $m+1$ challenges (obtained by querying a challenge oracle, for uniformly chosen preimages) given access to an inversion oracle that was queried up to m times.

for encrypting a single message) can be viewed as a strengthening of ATDFs that additionally hides all partial information and allows for high-entropy input. CCA-secure deterministic encryption was constructed from CPA-secure PKE (satisfying a minor technical condition) in the random oracle model in [2] and in the standard model from LTDFs in [9]. We note that [3] gave a direct construction of CCA-secure PKE from CCA-secure deterministic encryption.

In the randomized encryption context, we mention the related notion of *tag-based* encryption [27,1,26]. Indeed, TB-ATDF can be viewed an analogue of selective-tag weakly CCA-secure PKE [26] in the TDF context. We also point out that the related notion of "one-way CCA" for encryption has surfaced before; see, e.g., [33]. (We stress that the difference is not just conceptual, as this notion is for *randomized* encryption.)

Work on black-box constructions. The importance of giving black-box constructions in cryptography is well-understood. A complementary line of work, starting with the seminal paper of Impagliazzo and Rudich [25], seeks to understand the limitations of such constructions. In the context of PKE, Choi et al. [14] recently showed a black-box construction of a *non-malleable* (i.e. NM-CPA) PKE scheme from any semantically-secure (i.e. IND-CPA) one, whereas [21] showed that there is no such construction of CCA-secure PKE whose decryption algorithm does not call the encryption algorithm of the starting scheme. In fact, CCA-secure PKE seems to be the remaining fundamental cryptographic task for which we know a non-black-box construction (from "enhanced" OW-TDPs) but not a corresponding black-box one. We hope that our work brings us closer to this goal.

1.3 Open Problems

Our works raises a number of interesting open problems. It may be interesting to consider other natural security notions for TDFs (e.g., non-malleability or q-bounded adaptivity [15]) and study their instantiation from standard assumptions, their implications for PKE, as well as their relation to existing notions from Figure 1. Furthermore, some of the relations in Figure 1, in particular between TDFs and ATDFs, are open.

Lossy TDFs are only known to be instantiable from *decisional* assumptions (such as DDH and QR), whereas we show that ATDFs are also instantiable from a computational assumption (though a non-standard and interactive one, namely II-RSA). An interesting open question is whether it is possible to instantiate ATDFs from more standard computational assumptions (such as RSA or CDH). One could also try to define a different security notion for TDFs, weaker than adaptivity, that admits instantiations from standard computational assumptions but still suffices for black-box CCA-secure PKE.

Finally, we are optimistic that ATDFs may be useful in the general context of black-box constructions of cryptograhpic primitives secure against adaptive attacks.

2 Preliminaries

Notation. If x and y are (binary) strings, then $x \| y$ denotes an encoding from which they are uniquely recoverable. If $k \in \mathbb{N}$ then 1^k denotes the string of k ones. If S is a set then $s \stackrel{\$}{\leftarrow} S$ denotes the operation of picking an element s of S uniformly at random. We write $\mathsf{A}(x, y, \ldots)$ to indicate that A is an algorithm (i.e., a Turing Machine) with inputs $x, y, \ldots$ and by $z \stackrel{\$}{\leftarrow} \mathsf{A}(x, y, \ldots)$ we denote the operation of running A with inputs $(x, y, \ldots)$ and letting z be the output. We write $\mathsf{A}^{\mathcal{O}_1, \mathcal{O}_2, \ldots}(x, y, \ldots)$ to indicate that A is an algorithm with inputs $x, y, \ldots$ and access to oracles $\mathcal{O}_1, \mathcal{O}_2, \ldots$. With PT we denote polynomial time and with PPT we denote probabilistic polynomial time.

CCA-secure PKE. A *public key encryption* scheme $\mathsf{PKE} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathrm{MsgSp} = \mathrm{MsgSp}(k)$ consists of three PT algorithms, of which the first two, Kg and Enc, are probabilistic and the last one, Dec, is deterministic. Public/secret keys for security parameter $k \in \mathbb{N}$ are generated using $(pk, sk) \stackrel{\$}{\leftarrow} \mathsf{Kg}(1^k)$. Given such a key pair, a message $m \in \mathrm{MsgSp}$ is encrypted via $c \stackrel{\$}{\leftarrow} \mathsf{Enc}(pk, m)$; a ciphertext is decrypted by $m \leftarrow \mathsf{Dec}(sk, c)$. For correctness, we require that for all $k \in \mathbb{N}$, all messages $m \in \mathrm{MsgSp}$, it must hold that $\Pr[\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m] = 1$, where the probability is taken over the above randomized algorithms and $(pk, sk) \stackrel{\$}{\leftarrow} \mathsf{Kg}(1^k)$.

Let A be an adversary against PKE and define its IND-CCA-*advantage* as

$$\mathbf{Adv}^{\mathrm{cca}}_{\mathsf{PKE},\mathsf{A}}(k) \;=\; 2 \cdot \Pr\left[b = b' : \begin{array}{c} (pk, sk) \stackrel{\$}{\leftarrow} \mathsf{Kg}(1^k) \\ (m_0, m_1, \mathrm{st}) \stackrel{\$}{\leftarrow} \mathsf{A}^{\mathcal{O}(sk,\cdot)}(pk) \\ b \stackrel{\$}{\leftarrow} \{0,1\} \,;\; c^* \stackrel{\$}{\leftarrow} \mathsf{Enc}(pk, m_b) \\ b' \stackrel{\$}{\leftarrow} \mathsf{A}^{\mathcal{O}(sk,\cdot)}(c^*, \mathrm{st}) \end{array} \right] - 1,$$

where $\mathcal{O}(sk, c) = \mathsf{Dec}(sk, c)$, and in the second phase ("guess phase") A is not allowed to query $\mathcal{O}(sk, \cdot)$ for the challenge ciphertext c^*. We also require that m_0 and m_1 are of the same length. (Here st is some arbitrary state information.) We say that PKE is IND-CCA-secure if $\mathbf{Adv}^{\mathrm{cca}}_{\mathsf{PKE},\mathsf{A}}(\cdot)$ is negligible for all such PPT adversaries A.

3 Adaptive TDFs and CCA-Secure PKE Schemes

In this section, we introduce our notion of adaptive trapdoor functions (ATDFs) and an extension we call tag-based adaptive trapdoor functions (TB-ATDFs). We then show black-box constructions of CCA-secure PKE from these notions.

3.1 Adaptive Trapdoor Functions

Trapdoor Functions. Recall that a *trapdoor function* (TDF) is a triple of algorithms, where Tdg is probabilistic and on input 1^k generates an evaluation/trapdoor key-pair $(ek, td) \stackrel{\$}{\leftarrow} \mathsf{Tdg}(1^k)$, $\mathsf{F}(ek, \cdot)$ implements a function $f_{ek}(\cdot)$

over $\{0,1\}^k$ and $\mathsf{F}^{-1}(td,\cdot)$ implements its inverse $f_{ek}^{-1}(\cdot)$. Here we require TDFs to be injective. (Following [4], however, one can extend our results to poly-to-one TDFs as well.) Note that the above definition is purely functional and does not impose any security requirement.

One-wayness. First we recall the standard notion of one-wayness for trapdoor functions. Let A be an inverter and define its *OW-advantage* against TDF as

$$\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{TDF},\mathsf{A}}(k) \;=\; \Pr\left[x = x' : \begin{array}{l}(ek,td) \xleftarrow{\$} \mathsf{Tdg}(1^k)\,;\; x \xleftarrow{\$} \{0,1\}^k \\ y \leftarrow \mathsf{F}(ek,x)\,;\; x' \xleftarrow{\$} \mathsf{A}(ek,y)\end{array}\right].$$

Trapdoor function TDF is *one-way* if $\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{TDF},\mathsf{A}}(\cdot)$ is negligible for every PPT inverter A.

Adaptive one-wayness. Intuitively, adaptivity means that one-wayness holds even when the adversary may query an inverse oracle on points other than its challenge. Let $\mathsf{TDF} = (\mathsf{Tdg},\mathsf{F},\mathsf{F}^{-1})$ be a trapdoor function. Let A be an inverter and define its *AOW-advantage* against TDF as

$$\mathbf{Adv}^{\mathrm{aow}}_{\mathsf{TDF},\mathsf{A}}(k) \;=\; \Pr\left[x = x' : \begin{array}{l}(ek,td) \xleftarrow{\$} \mathsf{Tdg}(1^k)\,;\; x \xleftarrow{\$} \{0,1\}^k \\ y \leftarrow \mathsf{F}(ek,x)\,;\; x' \xleftarrow{\$} \mathsf{A}^{\mathsf{F}^{-1}(td,\cdot)}(ek,y)\end{array}\right],$$

where we demand that A does not query y to its oracle. Note that the behavior of oracle when queried on a y' outside the range of $\mathsf{F}(td,\cdot)$ is undefined; it returns whatever $\mathsf{F}^{-1}(td,y')$ does in this case (typically $\perp$). We say that TDF is *adaptive one-way* (or simply adaptive) if $\mathbf{Adv}^{\mathrm{atdf}}_{\mathsf{TDF},\mathsf{A}}(\cdot)$ is negligible for every such PPT inverter A.

Tag-based adaptive one-wayness. A *tag-based* TDF is a triple of algorithms $\mathsf{TDF}_{tag} = (\mathsf{Tdg}_{tag},\mathsf{F}_{tag},\mathsf{F}^{-1}_{tag})$ with associated *tag-space* $\mathit{TagSp}(k)$, where Tdg_{tag} is probabilistic and on input 1^k generates an evaluation/trapdoor key-pair $(ek,td) \xleftarrow{\$} \mathsf{Tdg}_{tag}(1^k)$. Furthermore, for every $t \in \mathit{TagSp}(k)$, $\mathsf{F}_{tag}(ek,t,\cdot)$ implements a function $f_{ek,t}(\cdot)$ over $\{0,1\}^k$ and $\mathsf{F}^{-1}_{tag}(td,t,\cdot)$ implements its inverse $f^{-1}_{ek,t}(\cdot)$. Let $\mathsf{A} = (\mathsf{A}_1,\mathsf{A}_2)$ be an inverter and define its *TB-AOW-advantage* against TDF_{tag} as

$$\mathbf{Adv}^{\text{tb-aow}}_{\mathsf{TDF}_{tag},\mathsf{A}}(k) \;=\; \Pr\left[x = x' : \begin{array}{l}t \xleftarrow{\$} \mathsf{A}_1(1^k)\,;\; (ek,td) \xleftarrow{\$} \mathsf{Tdg}_{tag}(1^k) \\ y \leftarrow \mathsf{F}_{tag}(ek,t,x)\,;\; x' \xleftarrow{\$} \mathsf{A}_2^{\mathsf{F}^{-1}_{tag}(td,\cdot,\cdot)}(ek,t,y)\end{array}\right],$$

where we demand that A_2 does not make a query of the form $\mathsf{F}^{-1}_{tag}(td,t,\cdot)$ to its oracle. We say that TDF_{tag} is *tag-based adaptive one-way* if $\mathbf{Adv}^{\text{tb-atdf}}_{\mathsf{TDF}_{tag},\mathsf{A}}(\cdot)$ is negligible for every such PPT inverter A.

In the above experiment the "challenge tag" t is independent of ek and hence it may also be called selective-tag security (similar to selective-ID security for IBE schemes). Stronger variants of this security notion can be obtained by allowing the adversary choose the challenge-tag t adaptively.

We note that typically one requires the size of the tag-space to be super-polynomial. In fact, TB-ATDFs with polynomial-size tag-space can be constructed from any OW-TDF, but are not sufficient for our applications.

Relations between ATDFs and TB-ATDFs. Note that tag-based TDFs can be viewed as a specific type of TDF in which the first part of the input is output in the clear. Using this observation, it is not difficult to show that ATDFs and TB-ATDFs are equivalent under *exponential hardness*, meaning that if we start with an exponentially-hard version of one primitive it implies an exponentially-hard version of the other; see the full version for details. It is an open question whether ATDFs and TB-ATDFs are equivalent in general.

3.2 CCA-Secure PKE from ATDFs

Construction from ATDFs. We show how to construct a one-bit CCA-secure PKE scheme from an ATDF. By a recent result of Myers and Shelat [30], this implies a black-box construction of a many-bit scheme as well.

Let $\mathsf{TDF} = (\mathsf{Tdg}, \mathsf{F}, \mathsf{F}^{-1})$ be a TDF and $\mathsf{hc}(\cdot)$ be a hardcore bit, for example the Goldreich-Levin bit [22]. We construct PKE scheme $\mathsf{PKE}[\mathsf{TDF}] = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ with message-space $\{0,1\}$ as follows:

- **Key Generation**: On input 1^k, run $(ek, td) \overset{\$}{\leftarrow} \mathsf{Tdg}$ and return (ek, td).
- **Encryption**: On input ek, m, where $m \in \{0,1\}$ do:

```
For $i = 1$ up to $k$:
    $x \overset{\$}{\leftarrow} \{0,1\}^k$ ; $h \leftarrow \mathsf{hc}(x)$ ; If $h = m$ then return $\mathsf{F}(ek, x) \| 0$
Return $m \| 1$.
```

- **Decryption**: On inputs td and $c = c_1 \| \mathsf{flag}$, if $\mathsf{flag} = 1$ then return c_1, else return $\mathsf{hc}(\mathsf{F}^{-1}(td, c_1))$.

It is clear that the above construction satisfies *correctness*. (Note that if the encryption algorithm happens to output the message in the clear it is still correctly decrypted, so this is a security, not a functionality, concern.) We now turn to security.

Theorem 1. *If* TDF *is adaptive one-way, then the* $\mathsf{PKE}[\mathsf{TDF}]$ *defined above is* $\mathsf{IND\text{-}CCA}$*-secure.*

The proof reduces $\mathsf{IND\text{-}CCA}$ security of the scheme to security of a hardcore bit by turning an adversary against the former into a distinguisher for the hardcore bit that is given k independent samples, and then applying a hybrid argument. We note that as a consequence, security of the scheme is only loosely related to security of the underlying hardcore bit (losing a factor $1/k$).

Proof (of Theorem 1). Given an adversary A against the PKE scheme, we transform its IND-CCA experiment via a sequence of games:

- **Game G_1**: The IND-CCA experiment.
- **Game G_2:** Instead of computing the hardcore bits using $\mathsf{hc}(\cdot)$, the encryption algorithm encrypts the challenge message by picking a uniformly random bit on each iteration of the for-loop. That is, the second line in the for-loop is replaced with "$h \xleftarrow{\$} \{0,1\}$."
- **Game G_3:** If the for-loop in the encryption algorithm completes its execution (without satisfying the $h = m$ condition), instead of returning the challenge message in the clear, it simply returns $\perp$ to the adversary. That is, the last line in the encryption algorithm is replaced with "Return $\perp$."

For $i \in \{1,2,3\}$, let $\Pr[\mathsf{A}^{G_i} \Rightarrow b]$ denote the probability that A outputs the challenge bit b when executed in Game G_i (taken over the coins of the game and of A).

We first claim that if there is an inverter A against TDF such that $\Pr[\mathsf{A}^{G_1} \Rightarrow b] - \Pr[\mathsf{A}^{G_2} \Rightarrow b]$ is non-negligible, then so is $\mathbf{Adv}^{\mathrm{aow}}_{\mathsf{TDF},\mathsf{A}}$. To show this, it suffices by a standard hybrid argument and security of $\mathsf{hc}(\cdot)$ to give a k-sample distinguisher D against $\mathsf{hc}(\cdot)$ whose advantage is non-negligible in this case. That is, D is given an input $ek, (y_1, h_1), \ldots, (y_k, h_k)$ where $y_i = \mathsf{F}(ek, x_i)$ and either $h_i = \mathsf{hc}(x_i)$ or is a uniformly random bit for all $1 \leq i \leq k$; D also has oracle access to $\mathsf{F}^{-1}(td, \cdot)$, which it may query on any y such that $y \neq y_i$ for all $1 \leq i \leq k$. Define D on inputs $ek, (y_1, h_1), \ldots, (y_k, h_k)$ as follows:

- Run A on input ek. When A makes a decryption query $c = c_1 \| \mathsf{flag}$, if $\mathsf{flag} = 1$ then respond with c_1 and otherwise $\mathsf{hc}(\mathsf{F}^{-1}(c_1))$. Let (m_1, m_2, st) be the output of A.
- Choose $b \xleftarrow{\$} \{0,1\}$ and find the least i^* such that $h_{i^*} = m_b$. If no such i^* exists, then set $c^* \leftarrow m_b \| 1$. Otherwise, set $c^* \leftarrow y_{i^*} \| 0$.
- Run A on inputs (c^*, st). When A makes a decryption query $c = c_1 \| \mathsf{flag}$, if $\mathsf{flag} = 1$ respond with c_1 and otherwise $\mathsf{hc}(\mathsf{F}^{-1}(c_1))$. Let b' be the output of A. Return 1 if $b = b'$ and 0 otherwise.

To show that D has the claimed property, note that the only way it can fail to give a perfect simulation of either Game G_1 or G_2 is if A makes a query of the form $y_i \| 0$ for some $1 \leq i \leq k$. It suffices to bound the probability of this when A is executed as in Game G_1, as follows. In the case $c^* = y_{i^*} \| 0$, A does not query $y_{i^*} \| 0$ by definition, and the probability it queries $y_i \| 0$ for any $i \neq i^*$ is at most $(k-1)q/2^{k-1}$ (where q an upper-bound on its number of decryption queries), where we use the fact that conditioning on $\mathsf{hc}(x_i) = 1 - m_b$ reduces the min-entropy of each x_i by at most 1 bit and $\mathsf{F}(ek, \cdot)$ is an injection. A similar analysis pertains to the case $c^* = m_b \| 0$. Overall, the probability is at most $kq/2^{k-1}$.

We next claim that $\Pr[\mathsf{A}^{G_2} \Rightarrow 1] - \Pr[\mathsf{A}^{G_3} \Rightarrow 1] \leq 2^{-k}$. This follows by using the fact that in Game G_2 the hardcore bits used to encrypt the challenge message have been replaced with uniformly random ones.

Finally, observe that $\Pr[\mathsf{A}^{G_3} \Rightarrow b] = 1/2$, since in this game A gets no information about b. Combining the above gives the theorem.

Construction from TB-ATDF. Our construction of CCA-secure PKE from a TB-ATDF is much simpler. It additionally makes use of a strongly one-time unforgeable signature scheme (see e.g. [37] for the definition). For simplicity, we give the construction below for the case of 1-bit messages. It is easy to extend it to a many-bit scheme, essentially by concatenating many applications of the TB-ATDF under independent inputs but the same tag.

Let $\mathsf{TDF}_{tag} = (\mathsf{Tdg}_{tag}, \mathsf{F}_{tag}, \mathsf{F}_{tag}^{-1})$ be a tag-based TDF and let $\mathsf{hc}(\cdot)$ be a hardcore bit. Let $\mathsf{OTS} = (\mathsf{K}, \mathsf{S}, \mathsf{V})$ be a signature scheme whose verification keys are contained in the tag-space of TDF_{tag}. We construct PKE scheme $\mathsf{PKE}[\mathsf{TDF}_{tag}, \mathsf{OTS}] = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ with message-space $\{0,1\}$ as follows:

- **Key Generation:** On input 1^k, run $(ek, td) \xleftarrow{\$} \mathsf{Tdg}_{tag}$ and return (ek, td).
- **Encryption:** On input ek, m where $m \in \{0,1\}$, run $(sk, vk) \xleftarrow{\$} \mathsf{K}(1^k)$ and choose $x \xleftarrow{\$} \{0,1\}^k$. Set $y_1 \leftarrow \mathsf{F}_{tag}(ek, vk, x)$ and $y_2 \leftarrow \mathsf{hc}(x) \oplus m$; also, set $\sigma \leftarrow \mathsf{S}(sk, y_1 \| y_2)$. Return $y_1 \| y_2 \| vk \| \sigma$.
- **Decryption:** On inputs td and $y = y_1 \| y_2 \| vk \| \sigma$, if $\mathsf{V}(vk, \sigma) = 1$ then set $x \leftarrow \mathsf{TDF}_{tag}(td, vk, y_1)$ and return $\mathsf{hc}(x) \oplus y_2$; , otherwise return $\perp$.

We have the following theorem.

Theorem 2. *If* TDF_{tag} *is adaptive one-way and* OTS *is one-time strongly unforgeable, then then* $\mathsf{PKE}[\mathsf{TDF}_{tag}, \mathsf{OTS}]$ *is* IND-CCA*-secure.*

The proof is straightforward and hence omitted.

Optimizations. Our construction of CCA-secure PKE from ATDFs can be simplified and made much more efficient if the given ATDF is a permutation or has linearly many simultaneous hardcore bits. Namely, in this case one can use the ATDF as a key-encapsulation mechanism (KEM) for an IND-CCA-secure symmetric encryption scheme.

Additionally, for some *specific* hardcore bits one may be able to sample uniformly from the set $\{x \in \{0,1\}^k \mid \mathsf{hc}(x) = b\}$ more efficiently than by repeated sampling of the uniform distribution on $\{0,1\}^k$. (Indeed, this is the case for the universally-hardcore Goldreich-Levin bit [22].) This translates to a corresponding efficiency improvement in the scheme.

Our construction of CCA-secure PKE from TB-ATDFs can also be made much more efficient if the given TB-ATDF is a permutation (for every tag) or has linearly many simultaneous hardcore bits. The idea is to first construct a selective-tag weakly CCA secure tag-PKE scheme in the sense of [26] by using the TB-ATDF as a KEM for a one-time CPA-secure symmetric encryption scheme. Then, as shown in [26], we can apply the transform of Boneh et al. [10] to obtain a CCA-secure PKE scheme, which uses only symmetric-key primitives.

4 Constructing ATDFs from Stronger TDFs

Inspired by the constructions of CCA-secure PKE in [35,37], we show that both ATDFs and TB-ATDFs can be constructed in a simple black-box manner from correlated-product TDFs [37]. As shown in [37], lossy TDFs (LTDFs) [35] imply CP-TDFs, thus by our result above they imply ATDFs and TB-ATDFs too.

However, we are able to give much more efficient direct construction in combination with an all-but-one TDF (ABO-TDF) as defined by [35].

4.1 Constructions from Correlated-Product TDFs

One-wayness under correlated-product. We first recall the notion of one-wayness under correlated product [37]. Let $\mathsf{TDF} = (\mathsf{Tdg}, \mathsf{F}, \mathsf{F}^{-1})$ be a trapdoor function, and let $\mathcal{C}_t$ be such that $\mathcal{C}_t(1^k)$ is distributed over $\{0,1\}^{tk}$ for a polynomial $t = t(k)$. Let A be an inverter and define its $\mathcal{C}_t$-*CP-advantage* against TDF as

$$\mathbf{Adv}^{\mathrm{cpow}}_{\mathsf{TDF},\mathsf{A}}(k) \;=\; \Pr\left[\begin{array}{rl} & (ek_i, td_i) \xleftarrow{\$} \mathsf{Tdg}(1^k), 1 \le i \le t\,; \\ (x_1, \ldots, x_t) & (x_1, \ldots, x_t) \xleftarrow{\$} \mathcal{C}_t(1^k)\,; \\ = (x'_1, \ldots, x'_t) \;:& y \leftarrow (f_{ek_1}(x_1), \ldots, f_{ek_t}(x_t))\,; \\ & (x'_1, \ldots, x'_t) \xleftarrow{\$} \mathsf{A}(ek_1, \ldots, ek_t, y) \end{array}\right].$$

We say that TDF *one-way under* $\mathcal{C}_t$*-correlated-product* if $\mathbf{Adv}^{\mathrm{cpow}}_{\mathsf{TDF},\mathsf{A}}(\cdot)$ is negligible for any PPT inverter A.

The cannonical $\mathcal{C}_t$ considered by [37] is such that $x_1 = x_2 = \ldots = x_t$, where x_1 is random. We call TDFs secure in this sense *one-way under* t*-correlated-product* (t-CP-TDF).

Construction of ATDFs. Let $\mathsf{TDF}_1 = (\mathsf{Tdg}_1, \mathsf{F}_1, \mathsf{F}_1^{-1})$ be a TDF, where we assume wlog (by suitable padding) that $\mathsf{TDF}_1(ek, \cdot)$ has a fixed output length $n = n(k)$. We construct $\mathsf{TDF} = (\mathsf{Tdg}, \mathsf{F}, \mathsf{F}^{-1})$ as follows:

- **Key Generation:** On input 1^k, let $(ek_0, td_0) \xleftarrow{\$} \mathsf{Tdg}(1^k)$ and for all $b \in \{0,1\}$ and $1 \le i \le n$ set $(ek_i^b, td_i^b) \xleftarrow{\$} \mathsf{Tdg}_1(1^k)$. Let $ek \leftarrow (ek_0, (ek_1^0, ek_1^1), \ldots, (ek_n^0, ek_n^1))$ and $td \leftarrow (td_0, (td_1^0, td_1^1), \ldots, (td_n^0, td_n^1))$. Return (ek, td).
- **Evaluation:** On inputs ek, x, return $\mathsf{F}_1(ek_0, x) \| \mathsf{F}_1(ek_1^{b_1}, x) \| \ldots \| \mathsf{F}_1(ek_n^{b_n}, x)$, where b_i denotes the ith bit of $\mathsf{F}(ek_0, x)$ for $1 \le i \le n$.
- **Inversion:** On inputs td and $y = y_0 \| y_1 \| \ldots \| y_n$, let $x \leftarrow \mathsf{F}_1^{-1}(td_0, y_0)$. Return x if $x = \mathsf{F}^{-1}(td_i^{b_i}, y_i) = \mathsf{F}_1^{-1}(td_0, y_0)$ for $0 \le i \le n$, where b_i denotes the ith bit of y_0, otherwise return $\perp$.

We have the following theorem.

Theorem 3. *If* TDF_1 *is a* $(n+1)$*-CP-TDF then* TDF *is an ATDF.*

Proof. Given an adversary A against TDF, we describe below an adversary B against TDF_1 such that $\mathbf{Adv}^{\mathrm{(n+1)\text{-}cpow}}_{\mathsf{B},\mathsf{TDF}_1} = \mathbf{Adv}^{\mathrm{aow}}_{\mathsf{A},\mathsf{TDF}}$.

On inputs $ek_1, \ldots, ek_{n+1}, y$ where $y = (\mathsf{F}_1(ek_1, x_1), \ldots, \mathsf{F}_1(ek_{n+1}, x_{n+1}))$, B sets $ek_0 \leftarrow ek_1$ and $ek_i^{b_i} \leftarrow ek_{i+1}$ for all $1 \le i \le n$, where b_i denotes the ith bit of $\mathsf{F}_1(ek_1, x_1)$. It then chooses $(ek_i^{1-b_i}, td_i^{1-b_i}) \xleftarrow{\$} \mathsf{Tdg}_1(1^k)$ for all $1 \le i \le n$. It runs A on inputs ek, y for ek defined as in the key generation algorithm of TDF. When A makes an inversion query $y' = y'_0 \| y'_1 \| \ldots \| y'_n$, B chooses an index i such that $b'_i \ne b_i$, where b'_i denotes the ith bit of y'_0. (As we argue below, such i must exist.) It sets $x' \leftarrow \mathsf{F}^{-1}(td_i^{b'_i}, y'_i)$. If $\mathsf{F}(ek, x') = y'$ then it returns x' to A and otherwise returns $\perp$. Finally, when A halts B returns its output.

It is clear that B satisfies the desired property. To finish the proof it remains to argue that index i used in answering A's inversion queries always exists. But this follows directly from injectivity of $\mathsf{F}(ek_0, \cdot)$ and the fact that A is not allowed to make an inversion query equal to its challenge.

Remarks. We note that it is possible to make the scheme more efficient by additionally using a universal one-way family (aka. TCR) of hash functions [31]. Then, the "selector" bits $b_1, \ldots, b_n$ in the construction are replaced with the bits of the hash of $\mathsf{F}(ek_0, x)$. We also note that following [37] it is possible to give a construction based on a CP-TDF allowing a slightly weaker correlation among the inputs.

Construction of TB-ATDFs. The above construction of ATDFs can easily be modified to give a construction of TB-ATDFs as well. The difference is that in the "selector" bits $b_1, \ldots, b_n$ are replaced with the bits $t_1, \ldots, t_n$ of the tag t. Notably, when we apply our construction of CCA-secure PKE from TB-ATDFs given in Section 3) to the resulting TB-ATDF, we obtain precisely the CCA-secure PKE scheme of [37].

4.2 Constructions from Lossy and All-But-One TDFs

We first recall the notion of lossy TDFs and their generalization called all-but-one TDFs from [35].

Lossy TDFs. A (k, ℓ)-*LTDF* is a quadruple $\mathsf{LTDF} = (\mathsf{LTdg}, \mathsf{LTdg}', \mathsf{LF}, \mathsf{LF}^{-1})$ of algorithms, where the triple $(\mathsf{LTdg}, \mathsf{LF}, \mathsf{LF}^{-1})$ is a TDF on $\{0,1\}^k$. We require that (1) the function $\mathsf{LTDF}(ek', \cdot)$ has a range of size at most 2^ℓ (where $\ell = \ell(k)$) for every ek', and (2) the keys ek, ek' are computationally indistinguishable, over the choice of $(ek, td) \overset{\$}{\leftarrow} \mathsf{LTdg}(1^k)$ and $ek' \overset{\$}{\leftarrow} \mathsf{LTdg}'(1^k)$.

All-but-one TDFs. An (k, ℓ)-*ABO-TDF* with branch-space $\{0,1\}^{n=n(k)}$ is a triple $\mathsf{ABO} = (\mathsf{ABO\text{-}Tdg}, \mathsf{ABO\text{-}F}, \mathsf{ABO\text{-}F}^{-1})$ of algorithms, where for every $r \neq r' \in \{0,1\}^n$, the triple $(\mathsf{ABO\text{-}Tdg}(1^k, r), \mathsf{ABO\text{-}F}(r', \cdot, \cdot), \mathsf{ABO\text{-}F}^{-1}(r', \cdot, \cdot))$ is a TDF on $\{0,1\}^k$ (the "lossy branch" r is passed as an input to $\mathsf{ABO\text{-}Tdg}$). We further require (1) for every $r \in \{0,1\}^n$ and ek', the function $\mathsf{ABO\text{-}F}(r, ek', \cdot)$ has range-size at most 2^ℓ (where $\ell = \ell(k)$), and (3) for every $r \neq r' \in \{0,1\}^n$, the keys ek_1, ek_2 are computationally indistinguishable, over $(ek_1, td_1) \overset{\$}{\leftarrow} \mathsf{ABO\text{-}Tdg}(1^k, r)$ and $(ek_2, td_2) \overset{\$}{\leftarrow} \mathsf{ABO\text{-}Tdg}(1^k, r')$.

Construction of ATDFs. Our construction simplifies the construction of CCA-secure deterministic encryption given in [9]. Let $\mathsf{LTDF} = (\mathsf{LTdg}, \mathsf{LTdg}', \mathsf{LF}, \mathsf{LF}^{-1})$ be a (k, ℓ_1)-LTDF and let $\mathsf{ABO} = (\mathsf{ABO\text{-}Tdg}, \mathsf{ABO\text{-}F}, \mathsf{ABO\text{-}F}^{-1})$ be a (k, ℓ_2)-ABO-TDF; for simplicity we assume its branch-space is $\{0,1\}^n$ for $n = n(k)$. Let $T : R \to (\{0,1\}^n \setminus \{0^n\})$ be a hash function, where R denotes the range of $\mathsf{LF}(ek, \cdot)$. We construct $\mathsf{TDF}[\mathsf{LTDF}, \mathsf{ABO}, T]$ as follows.

- **Key Generation:** On input 1^k, run $(ek_{\mathrm{ltf}}, td_{\mathrm{ltf}}) \xleftarrow{\$} \mathsf{LTdg}(1^k)$ and $(ek_{\mathrm{abo}}, td_{\mathrm{abo}}) \xleftarrow{\$} \mathcal{F}_{\mathrm{abo}}(1^k, 0^n)$. Return $((ek_{\mathrm{ltf}}, ek_{\mathrm{abo}}), (td_{\mathrm{ltf}}, td_{\mathrm{abo}}))$.
- **Evaluation:** On inputs $(ek_{\mathrm{ltf}}, ek_{\mathrm{abo}})$ and $x \in \{0,1\}^k$, set $y_1 \leftarrow \mathsf{LF}(ek_{\mathrm{ltf}}, x)$ and $y_2 \leftarrow \mathsf{ABO\text{-}F}(T(y_1), ek_{\mathrm{abo}}, x)$. Return $y_1 \| y_2$.
- **Inversion:** On inputs $(td_{\mathrm{ltf}}, td_{\mathrm{abo}})$ and $y = y_1 \| y_2$, set $x \leftarrow \mathsf{LF}^{-1}(td, y_1)$. If $y_2 = \mathsf{ABO\text{-}F}(T(y_1), ek_{\mathrm{abo}}, x)$ then return x, otherwise return $\perp$.

We have the following theorem.

Theorem 4. *If $\ell_1 + \ell_2 = k - \omega(\log k)$ and T is TCR, then $\mathsf{TDF}[\mathsf{LTDF}, \mathsf{ABO}, T]$ defined above is an ATDF.*

The proof follows [35] and is given in the full version.

Construction of TB-ATDFs. Similarly to our construction of ATDF from CP-TDF, the above construction can be adapted to construct a tag-based ATDF instead. The difference is that in the evaluation algorithm, instead of evaluating the all-but-one TDF at branch $T(y_1)$, it is evaluated at branch t, where the latter is the input tag. As before, when we apply our general construction of CCA-secure PKE from TB-ADTFs given in Section 3 to the resulting TB-ATDF, we obtain precisely the CCA-secure PKE scheme of [35].

5 On the Complexity of Adaptive TDFs

In this section, we show that there is no black-box construction of CP-TDFs from either ATDFs or TB-ATDFs; combined with the results of Section 4, this shows that the latter are (surprisingly) strictly *weaker* primitives (in a black-box sense). We then show that TB-ATDFs can be realized based on an assumption on RSA inversion not known to imply CP-TDFs.

5.1 A Black-Box Separation

Very recently, Vahlis [39] showed that there is no black-box construction of CP-TDFs from one-way TDFs. We observe here that his proof extends to rule out a black-box construction of CP-TDFs from either ATDFs or TB-ATDFs.

Theorem 5. *There is no black-box construction of CP-TDFs from ATDFs or TB-ATDFs.*

The theorem actually follows by extending Vahlis's proof to rule out a black-box construction of CP-TDFs from *exponentially-hard* ATDFs. Since as discussed in Section 3, TB-ATDFs are implied by exponentially-hard TDFs, this rules out a black-box construction of CP-TDFs from TB-ATDFs as well.

Since Vahlis's proof is rather technical we avoid explaining its details here. Instead, we describe the high-level ideas and point out a minor change needed to give our claimed result.

Similar to most black-box separation results, in order to show that there is no black-box construction of primitive P_1 from primitive P_2, the proof starts by defining an *ideal oracle* O (the ideal version of P_2), and a *break oracle* B. One then shows that (i) there exist an adversary A that breaks any construction of P_1, with the help of a small number of queries to B and (ii) P_2 can be securely realized using the ideal oracle O, even when the adversary is given access to B.

Oracle O. The ideal oracle O is essentially an ideal trapdoor permutation (as described in several previous works [20]). Roughly speaking, O is defined as a triple of functions (g, e, d) sampled uniformly at random from the set of all functions with the following property: g maps trapdoors to public keys; $e(pub, \cdot)$ is an independent permutation for every public key pub, and $d(pri, \cdot)$ inverts $e(pub, \cdot)$ if pri is the trapdoor corresponding to pub. One may assume that trapdoors, public keys, and inputs are all of the same length, i.e. equal to the security parameter. Also note that there is no need to explicitly define d as it is determined given the definitions of g and e.

It is easy to see that oracle O is an ATDF; in fact, it is an *exponentially-hard* ATDF. However, as pointed out in [39], O is also correlation secure as the permutations for every public key is chosen independently and uniformly at random.

oracle B. Oracle B is specially designed to break the security of a CP-TDF. It takes as input a triple of circuits (G^O, E^O, D^O) which are candidates for a correlation secure TDF, two public keys PUB_1, PUB_2 and the values $E(PUB_1, x)$ and $E(PUB_2, x)$. The naive solution would be to let oracle B return x. However, this would make oracle B too powerful and would allow an adversary to break the security of any ideal TDF by letting the two public keys be $pub_1 = pub_2$. This problem is solved by requiring that the public keys of O encoded in PUB_1 are *distinct* from those encoded in PUB_2. An additional problem is caused by the fact that the adversary can make queries that contain invalid public keys, while detecting invalid keys by oracle B can render it too powerful. This issue is resolved by requiring the adversary to provide a partial oracle $O' = (g', e', d')$ that is defined on a small part of the domain of (g, e, d) such that relative to O', PUB_1 and PUB_2 are valid public keys.

We refer the reader to [39] for a more formal description of oracles O and B. The following claims (proven in [39]), complete the argument.

Claim 1. ([39]) There exist an adversary that only makes polynomially many queries (in the security parameter) to oracles O and B, and breaks the security of any CP-TDF function with non-negligible probability.

Claim 2. Let $\mathsf{TDF} = (G^O, E^O, D^O)$ be the trapdoor function that simply forwards its inputs to $O = (g, e, d)$. For any adversary A that makes polynomially many queries to oracles B and O, A's advantage in breaking TDF in the ATDF game is negligible.

In [39], Claim 2 is proven for the case when A is playing the OW-TDF game. However, the proof easily extends to the case of adaptive TDFs. Particularly, the bulk of the proof consists of describing a simulator S that simulates the answers for queries made to oracle B. For consistency purposes, S keeps a list O^* of all the query/answers made to the challenge function $e(pub^*, \cdot)$ where pub^* is the challenge public key. In case of ATDFs, S needs to do the same for any query $e^{-1}(pub^*, \cdot)$ made to the inversion oracle. The rest of the proof stays the same.

Note that, in the above discussion, we did not restrict the running time of the adversaries. Instead, we only required that the number of queries they make to the oracles is small. It is however easy to bring things to the world of polynomial-time adversaries by giving everyone access to a PSPACE oracle (see [25]).

5.2 Tag-Based ATDF from an Assumption on RSA Inversion

To further demonstrate the usefulness of our new notions, we show that TB-ATDFs are realizable from an assumption on RSA inversion not known to imply a CP-TDF.

Instance-independent RSA [33,13]. The instance-independent RSA assumption (II-RSA) speaks to the difficulty of solving the RSA problem — that is, computing e-th roots modulo $N = pq$ — even if given access to an oracle that computes e'-th roots modulo N for $e' \neq e$. Of course, some additional restriction on the exponents is necessary for this assumption to hold; in what follows we require that e, e' are primes. To define the assumption formally, let the tuple of algorithms $(\mathsf{RSA_g}, \mathsf{RSA}, \mathsf{RSA}^{-1})$ be defined in the natural way with the exception that the exponent e is no longer generated by the key generation step. That is, on input 1^k, algorithm $\mathsf{RSA_g}$ generates (ek, td) where $ek = N = pq$, and $td = (p, q)$ for two uniformly chosen $k/2$-bit primes p, q. Moreover, we require p, q to be *safe* primes, meaning $(p-1)/2, (q-1)/2$ are also prime. On inputs $e \in \mathbb{Z}^*_{(p-1)(q-1)}$, $x \in \mathbb{Z}^*_N$ and N, algorithm RSA returns $c = x^e \mod N$. On inputs $(p, q), e, y$, algorithm RSA^{-1} computes $d \leftarrow e^{-1} \mod (p-1)(q-1)$ and returns $y^d \mod N$. Let $n = n(k)$ be an integer. For an inverter $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ define its *II-RSA advantage for n* as

$$\mathbf{Adv}^{\text{II-RSA}}_{\mathsf{A},n}(k) \;=\; \Pr\left[x = x' : \begin{array}{c} e \xleftarrow{\$} \mathcal{P}_n \; ; \; (ek, td) \xleftarrow{\$} \mathsf{RSA_g}(1^k) \\ x \xleftarrow{\$} \mathbb{Z}_N \; ; \; y \leftarrow \mathsf{RSA}(ek, e, x) \\ x' \xleftarrow{\$} \mathsf{A}_2^{\mathsf{RSA}^{-1}(td,\cdot,\cdot)}(ek, e, y) \end{array} \right]$$

where here and in what follows $\mathcal{P}_n$ denotes the set of all n-bit primes and we require that A_2 only makes queries of the form $\mathsf{RSA}^{-1}(td, e', y')$ for primes $e' \neq e$. We say that the $\mathsf{II\text{-}RSA}$ *holds for n* if $\mathbf{Adv}^{\text{II-RSA}}_{\mathsf{A},n}(\cdot)$ is negligible for every such PPT inverter A.

Discussion. $\mathsf{II\text{-}RSA}$ was first conjectured (in a more general form) by Paillier and Villar [33], whose work was concerned with showing that several RSA-based schemes *cannot* be proven secure in the standard model. More recently, Chevallier-Mames and Joye [13] observed that $\mathsf{II\text{-}RSA}$ can be used to *construct* CCA-secure encryption as well. We note that [33] actually considered the assumption parameterized by a *fixed* "challenge" e (e.g., $e = 3$). We follow the formulation [13] and choose e at random from the set of all primes of a given length.

PRIME SEQUENCE GENERATOR. Our construction uses the "prime sequence generator" of [11], which for any $n \in \mathbb{N}$ with $k \geq (n+1)/2$ probabilistically constructs an efficiently computable, (with overwhelming probability) injective map $\mathsf{phash}_n \colon \{0,1\}^k \to \mathcal{P}_n$. First, one chooses a random $2(n+1)^2$-wise-independent function $Q \colon \{0,1\}^k \times \{1,\ldots,2(n+1)^2\} \to \{0,1\}^n$ using the standard polynomial evaluation construct over $\mathbb{F}_{2^{\kappa+1}}$. Then for $t \in \{0,1\}^k$, we define $\mathsf{phash}_n(t)$ to be the first prime in the sequence $Q(t,1),\ldots,Q(t,2(n+1)^2)$.

TAG-BASED ATDF FROM II-RSA. Let phash_n be as defined above for $k \geq (n+1)/2$. We construct a tag-based ATDF $\mathsf{TDF}_{tag}[\mathsf{phash}_n] = (\mathsf{Tdg}_{tag}, \mathsf{F}_{tag}, \mathsf{F}_{tag}^{-1})$ with tag-space $\{0,1\}^k$ as follows:

- **Key Generation:** On input 1^k, return $(ek, td) \xleftarrow{\$} \mathsf{RSA_g}(1^k)$.
- **Evaluation:** On inputs x, $ek = N$, and tag $t \in \{0,1\}^k$, return $\mathsf{RSA}(ek, \mathsf{phash}_n(t), x)$.
- **Inversion:** On inputs y, $td = (p,q)$ and tag $t \in \{0,1\}^k$, return $\mathsf{RSA}^{-1}(td, \mathsf{phash}_n(t), y)$.

We have the following theorem.

Theorem 6. *Let* phash_n *be as above. If the* $\mathsf{II\text{-}RSA}$ *assumption holds for* n *then* $\mathsf{TDF}_{tag}[\mathsf{phash}_n]$ *defined above is a TB-ATDF (in fact, it is a TB-ATDP).*

We stress that the use of the "prime sequence generator" in the construction does not introduce any unproven assumption.

Proof. (Sketch.) Given an adversary A against $\mathsf{TDF}_{tag}[\mathsf{phash}_n]$, we consider two games, which we call G_1 and G_2. Game G_1 is just the TB-ATDF experiment with A against $\mathsf{TDF}_{tag}[\mathsf{phash}_n]$. For Game G_2, we modify the inversion oracle to return $\perp$ whenever A makes an inversion query on a tag t' such that $\mathsf{phash}_n(t') = \mathsf{phash}_n(t)$, where t is the challenge tag. For $i \in \{1,2\}$, let $\Pr\left[\, \mathsf{A}^{G_i} \Rightarrow x \,\right]$ denote the probability that A returns the challenge input x when executed in G_i.

First, we claim that $\Pr\left[\, \mathsf{A}^{G_1} \Rightarrow x \,\right] - \Pr\left[\, \mathsf{A}^{G_2} \Rightarrow x \,\right] \leq 2^{-\Omega(k)}$. This follows from the analysis of the prime sequence generator in [11], who show that with probability at least $1 - 2^{-\Omega(n)}$ over the choice of Q in its construction, the set $\{\mathsf{phash}_n(t) \,:\, t \in \{0,1\}^k\}$ contains 2^k *random* and *distinct* n-bit primes.

Next, we claim that we can construct an adversary B_2 against $\mathsf{II\text{-}RSA}$ such that $\mathbf{Adv}^{\text{II-RSA}}_{\mathsf{B}_2} = \Pr\left[\, \mathsf{A}^{G_2} \Rightarrow x \,\right]$, which completes the proof. Note that an adversary against $\mathsf{II\text{-}RSA}$ receives the challenge exponent e "from the outside," so we need a way of "programming" the prime sequence generator at a given point. For this we can use the ideas of [28], who show that for any $t^* \in \{0,1\}^n$ and random $e^* \in \mathcal{P}_n$, it is possible to construct the polynomial $Q = Q_{t^*,e^*}$ for the prime sequence generator in such a way that $\mathsf{phash}_n(t^*) = e^*$ and that for every t_0^*, t_1^*, the distribution of these Q's are $2^{-\Omega(n)}$-close.

AN EFFICIENT CCA-SECURE RSA-BASED PKE SCHEME. The above construction of TB-ATDP leads to a very efficient CCA-secure RSA-based PKE scheme in the standard model. Namely, we apply the "optimized" construction of CCA-secure PKE from TB-ATDF given in Section 3.

Recall that this construction proceeds in two steps. First, we construct a selective-tag weakly CCA-secure PKE scheme in the sense of [26] by using the TB-ATDF as a key-encapsulation mechanism for a one-time IND-CPA secure symmetric encryptions scheme. We note that to extract enough hardcore bits from only one application of RSA, we can combine II-RSA with the "small-solutions" RSA problem of [38]. Furthermore, by strengthening II-RSA to allow e, e' to be composites such that $\gcd(e, e') = 1$ and quantifying over *all* e in the assumption, we can "heuristically" use a cryptographic hash function with 512-bit output in place of the prime sequence generator for 80-bit security.

The construction then applies the MAC-based BCHK-transform [10] to obtain a fully CCA-secure PKE scheme. The resulting scheme has ciphertexts containing only one group element and, assuming the strengthening to II-RSA discussed above, its encryption time is dominated by one 512-bit exponentiation. In terms of applicability, however, it is unclear if such a standard-model PKE scheme secure based on an interactive assumption about RSA (such as II-RSA) is preferable to a random-oracle scheme based on its one-wayness (such as RSA-OAEP [7]).

Acknowledgements

We thank Mihir Bellare for helpful discussions that improved the presentation of our results. We also thank Chris Peikert, Ramarathnam Venkatesan, and the anonymous reviewers of Eurocrypt 2010 for helpful comments. Eike Kiltz was supported by the research program Sentinels. Sentinels is being financed by Technology Foundation STW, the Netherlands Organization for Scientific Research (NWO), and the Dutch Ministry of Economic Affairs. Adam O'Neill was supported in part by Alexandra Boldyreva's NSF CAREER award 0545659 and NSF Cyber Trust award 0831184 and thanks her for her support.

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. 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)
3. 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)
4. Bellare, M., Halevi, S., Sahai, A., Vadhan, S.P.: Many-to-one trapdoor functions and their relation to public-key cryptosystems. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 283–298. Springer, Heidelberg (1998)
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. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Ashby, V. (ed.) ACM CCS 1993: 1st Conference on Computer and Communications Security, November 1993, pp. 62–73. ACM Press, New York (1993)
7. Bellare, M., Rogaway, P.: Optimal asymmetric encryption. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 92–111. Springer, Heidelberg (1995)
8. Blum, M., Micali, S.: How to generate cryptographically strong sequences of pseudo-random bits. SIAM J. Comput. 13(4), 850–864 (1984)
9. 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)
10. Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM Journal on Computing 36(5), 915–942 (2006)
11. Cachin, C., Micali, S., Stadler, M.: Computationally private information retrieval with polylogarithmic communication. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 402–414. Springer, Heidelberg (1999)
12. Canetti, R., Dakdouk, R.R.: Towards a theory of extractable functions. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 595–613. Springer, Heidelberg (2009)
13. 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)
14. Choi, S.G., Dachman-Soled, D., Malkin, T., Wee, H.: Simple, black-box constructions of adaptively secure protocols. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 387–402. Springer, Heidelberg (2009)
15. 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)
16. 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)
17. 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)
18. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing 33(1), 167–226 (2003)
19. Dolev, D., Dwork, C., Naor, M.: Nonmalleable cryptography. SIAM Journal on Computing 30(2), 391–437 (2000)
20. Gennaro, R., Trevisan, L.: Lower bounds on the efficiency of generic cryptographic constructions. In: 41st Annual Symposium on Foundations of Computer Science, November 2000, pp. 305–313. IEEE Computer Society Press, Los Alamitos (2000)
21. 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)
22. Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: 21st Annual ACM Symposium on Theory of Computing, May 1989, pp. 25–32. ACM Press, New York (1989)

23. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)
24. 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)
25. Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: 21st Annual ACM Symposium on Theory of Computing, May 1989, pp. 44–61. ACM Press, New York (1989)
26. 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)
27. 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)
28. Micali, S., Rabin, M.O., Vadhan, S.P.: Verifiable random functions. In: 40th Annual Symposium on Foundations of Computer Science, October 1999, pp. 120–130. IEEE Computer Society Press, Los Alamitos (1999)
29. Mol, P., Yilek, S.: Chosen-ciphertext security from slightly lossy trapdoor functions. In: PKC (2010)
30. Myers, S., Shelat, A.: Bit encryption is complete. In: 50th Annual Symposium on Foundations of Computer Science. IEEE Computer Society Press, Los Alamitos (2009)
31. Naor, M., Yung, M.: Universal one-way hash functions and their cryptographic applications. In: 21st Annual ACM Symposium on Theory of Computing, May 1989, pp. 33–43. ACM Press, New York (1989)
32. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: 22nd Annual ACM Symposium on Theory of Computing, May 1990. ACM Press, New York (1990)
33. Paillier, P., Villar, J.L.: Trading one-wayness against chosen-ciphertext security in factoring-based encryption. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 252–266. Springer, Heidelberg (2006)
34. 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)
35. 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, May 2008, pp. 187–196. ACM Press, New York (2008)
36. 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)
37. 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)
38. Steinfeld, R., Pieprzyk, J., Wang, H.: On the provable security of an efficient RSA-based pseudorandom generator. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 194–209. Springer, Heidelberg (2006)

39. 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)
40. Yao, A.C.: Theory and applications of trapdoor functions. In: 23rd Annual Symposium on Foundations of Computer Science, November 1982, pp. 80–91. IEEE Computer Society Press, Los Alamitos (1982)

Author Index