Jerzy Kaczorowski
Josef Pieprzyk
Jacek Pomykała (Eds.)

LNCS 10737

Number-Theoretic Methods in Cryptology

First International Conference, NuTMiC 2017
Warsaw, Poland, September 11–13, 2017
Revised Selected Papers

Springer

Lecture Notes in Computer Science 10737

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

More information about this series at http://www.springer.com/series/7410

Jerzy Kaczorowski · Josef Pieprzyk
Jacek Pomykała (Eds.)

Number-Theoretic Methods in Cryptology

First International Conference, NuTMiC 2017
Warsaw, Poland, September 11–13, 2017
Revised Selected Papers

Springer

Editors

Jerzy Kaczorowski
Adam Mickiewicz University
Poznań
Poland

Jacek Pomykała
University of Warsaw
Warsaw
Poland

Josef Pieprzyk
Queensland University of Technology
Brisbane, QLD
Australia

and

Institute of Computer Science
Polish Academy of Sciences
Warsaw
Poland

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-76619-5 ISBN 978-3-319-76620-1 (eBook)
https://doi.org/10.1007/978-3-319-76620-1

Library of Congress Control Number: 2018934356

LNCS Sublibrary: SL4 – Security and Cryptology

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by the registered company Springer International Publishing AG
part of Springer Nature
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The First Number-Theoretic Methods in Cryptology (NuTMiC) Conference was held at the University of Warsaw, Poland, during September 11–13, 2017. The aim of the conference is to cross-pollinate number theory and cryptology. Besides the well-established connections between the two domains such as primality testing, factorization, elliptic curves, lattices (to mention a few), the conference endeavors to forge new ones that would encompass number theory structures and algorithms that have never been used in cryptology before. It is expected that these new connections will lead to novel, more efficient and secure cryptographic systems and protocols (such as one-way functions, pseudorandom number generators, encryption algorithms, digital signatures, etc.). The conference topics include lattice-based cryptography, elliptic curves and bilinear-based cryptography, L-functions with applications to cryptology, large sieve methods in cryptography and exponential sums over finite fields and randomness extractors.

We received 32 submissions. The review process was conducted in two phases. In the first, the papers were lightly reviewed with emphasis on helpful comments and feedback. There were 21 papers that were chosen for conference presentation. The final papers were collected after the conference for these proceedings. The papers and were subject to a rigorous review. The proceedings include 15 peer-reviewed papers and three invited talks.

We would like to thank the Program Committee members and the external reviewers for their time and effort. We also thank the local organizers who made the conference a success. In particular, Marek Janiszewski, Aleksandra Dolot, Daniel Waszkiewicz, and Marcin Tunia took care of the conference website, helped us with EasyChair, and manned the conference desk. Bartosz Źrałek helped us with e-mail communication and financial overview. We would like to express our appreciation to Springer for their support and help in the production of the conference proceedings. We thank the EasyChair team for letting us use the server.

Last but not least, we highly appreciate the support the conference received from the Faculty of Mathematics, Informatics, and Mechanics of the University of Warsaw (MIMUW) and Warsaw Center of Mathematics and Computer Science (WCMCS). In particular, the Dean of MIMUW, Professor Paweł Strzelecki, welcomed the participants and hosted the conference in his department facilities. Professor Krzysztof Barański, Director of the Institute of Mathematics, and Professor Anna Zdunik, Chair WCMCS MIMUW, supported the conference financially. We gladly acknowledge the continuous assistance of the university administration units: financial, international collaboration, and audiovisual/technical services.

December 2017

Jerzy Kaczorowski
Josef Pieprzyk
Jacek Pomykała

NuTMiC 2017

The First Conference on Number-Theoretic Methods in Cryptology
Warsaw University, Warsaw, Poland
September 11–13, 2017

In Co-operation with IACR

General Co-chairs

Jacek Pomykała	University of Warsaw, Poland
Piotr Sapiecha	Warsaw University of Technology, Poland

Organizing Committee

Chris Charnes	IAP(T) TU Darmstadt, Germany
Aleksandra Dolot	Warsaw University of Technology, Poland
Robert Dryło	Warsaw School of Economics, Poland
Konrad Durnoga	University of Warsaw, Poland
Marek Janiszewski	Warsaw University of Technology, Poland
Mariusz Skałba	University of Warsaw, Poland
Krzysztof Szczypiorski	Warsaw University of Technology, Poland
Janusz Szmidt	Military Communication Institute, Poland
Marcin Tunia	Warsaw University of Technology, Poland
Daniel Waszkiewicz	Warsaw University of Technology, Poland
Konrad Wrona	NATO Communications and Information Agency, The Netherlands
Bartosz Żrałek	University of Warsaw, Poland

Program Co-chairs

Jerzy Kaczorowski	Adam Mickiewicz University and Institute of Mathematics, Polish Academy of Sciences, Poland
Josef Pieprzyk	Queensland University of Technology, Australia and Institute of Computer Science, Polish Academy of Sciences, Poland
Jacek Pomykała	University of Warsaw, Poland

Program Committee

Tomasz Adamski	Warsaw University of Technology, Poland
Andrzej Białynicki-Birula	University of Warsaw, Poland
Xavier Boyen	Queensland University of Technology, Australia
Chris Charnes	IAP(T) TU Darmstadt, Germany
Henri Cohen	Université de Bordeaux, France
Nicolas Courtois	University College London, UK
Andrzej Dąbrowski	University of Szczecin, Poland
Gerhard Frey	University of Duisburg-Essen, Germany
Jerzy Gawinecki	Military University of Technology, Warsaw, Poland
Katalin Gyarmati	Eötvös Loránd University, Hungary
Harald Helfgott	Georg-August-Universität Göttingen, Germany and École Normale Supérieure, Paris, France
Jerzy Jaworski	Adam Mickiewicz University, Poland
Zbigniew Jelonek	Institute of Mathematics, Polish Academy of Sciences, Warsaw, Poland
Przemysław Koprowski	University of Silesia, Poland
Mieczysław Kula	University of Silesia, Poland
Zbigniew Kotulski	Warsaw University of Technology, Poland
Bogdan Ksieżopolski	Maria Curie-Skłodowska University, Poland
Alessandro Languasco	Università di Padova, Italy
Tomasz Łuczak	Adam Mickiewicz University, Poland
Giuseppe Molteni	Università di Milano, Italy
Andrew Odlyzko	University of Minnesota, USA
Andrzej Paszkiewicz	Warsaw University of Technology, Poland
Rene Peralta	Computer Security Division, NIST, USA
Alberto Perelli	Università di Genova, Italy
Jerzy Pejaś	West Pomeranian University of Technology, Poland
Olivier Ramarè	Aix Marseille Universitè, France
András Sárközy	Eötvös Loránd University, Hungary
Andrzej Schinzel	Institute of Mathematics, Polish Academy of Sciences, Poland
Jennifer Seberry	University of Wollongong, Australia
Igor Shparlinski	University of New South Wales, Australia
Mariusz Skałba	University of Warsaw, Poland

Marian Srebrny	Institute of Computer Science, Polish Academy of Sciences, Poland
Janusz Stokłosa	Poznań University of Technology, Poland
Janusz Szmidt	Military Communication Institute, Zegrze, Poland
Huaong Wang	Nanyang Technological University, Singapore

Steering Committee

Xavier Boyen	Queensland University of Technology, Australia
Nicolas Courtois	University College London, UK
Chris Charnes	IAP(T) TU Darmstadt, Germany
Gerhard Frey	University of Duisburg-Essen, Germany
Jerzy Kaczorowski	Adam Mickiewicz University and Institute of Mathematics, Polish Academy of Sciences, Poland
Rene Peralta	Computer Security Division, NIST, USA
Josef Pieprzyk	Queensland University of Technology, Australia and Institute of Computer Science, Polish Academy of Sciences, Poland
Jacek Pomykała	University of Warsaw, Poland
Igor Shparlinski	University of New South Wales, Australia
Huaong Wang	Nanyang Technological University, Singapore

Additional Reviewers

Konrad Durnoga
Robert Dryło
Maciej Grześkowiak
Renata Kawa
Piotr Sapiecha
Daniel Waszkiewicz
Bartosz Źrałek

Abstracts of Invited Talks

Arithmetic Geometry: Deep Theory, Efficient Algorithms and Surprising Applications

Gerhard Frey

University of Duisburg-Essen

One of the most astonishing success stories in recent mathematics is arithmetic geometry, which unifies methods from classical number theory with algebraic geometry ("schemes"). In particular, an the extremely important role is played by the Galois groups of base schemes like rings of integers of number fields or rings of holomorphic functions of curves over finite fields. These groups are the algebraic analogues of topological fundamental groups, and their representations induced by the action on divisor class groups of varieties over these domains yielded spectacular results like Serre's Conjecture for two-dimensional representations of the Galois group of $\mathbb{Q}$, which implies for example the modularity of elliptic curves over $\mathbb{Q}$ and so Fermat's Last Theorem (and much more).

At the same time the algorithmic aspect of arithmetical objects like lattices and ideal class groups of global fields became more and more important and accessible, stimulated by and stimulating the advances in theory. An outstanding result is the theorem of F. Heß and C.Diem yielding that the addition in divisor class groups of curves of genus g over finite fields $\mathbb{F}_q$ is (probabilistically) of polynomial complexity in g (fixed) and $\log(q)$ (g fixed). So one could hope to use such groups for public key cryptography, e.g. for key exchange, as established by Diffie-Hellman for the multiplicative group of finite fields.

The obtained insights play not only a constructive role but also a destructive role for the security of such systems. Algorithms for fast scalar multiplication and point counting (e.g. the algorithm of Schoof-Atkin-Elkies) make it possible to find divisor class groups in cryptographically relevant ranges but, at the same time, yield algorithms for the computation of discrete logarithms that are in many cases "too fast" for security. The good news is that there is a narrow but not empty range of candidates usable for public key cryptography and secure against all known attacks based on conventional computer algorithms: carefully chosen curves of genus 1 (elliptic curves) and hyperelliptic curves of genus ≤ 3 over prime fields.

In the lecture we gave an overview on the methods and results for the rather satisfying situation of elliptic and hyperelliptic cryptography–as long as we restrict the algorithms to classical bit-operations. But the possibility of the existence of quantum computers in a not too far future forces to look for alternatives.

Therefore we formulated a rather abstract setting for Diffie-Hellman key exchange schemes using (closely related) categories for the exchange partners, for which push-outs exist and are computable. The DL-systems with cyclic groups are the easiest realizations (and by Shor's algorithm cracked in polynomial time), the next level are G-sets (G a semi group) with a commutativity condition. If G is abelian (e.g. equal to $\mathbb{N}$)

then an algorithm of Kuperberg for the hidden shift problem with subexponential complexity can be applied, for general groups no such algorithm is known (but the commutation condition is difficult to realize).

Using fundamental results of M. Deuring about isogenies of elliptic curves we described the system of Couveignes-Stolbunov for key exchange using the isogeny graph of ordinary elliptic curves with endomorphism ring O, which is a G-set with $G = \mathrm{Pic}(O)$ and so only of subexponential security under quantum computing, and the system of De Feo using supersingular elliptic curves (and nicely fitting into our categorical frame) for which no non-exponential quantum computer attack is known till now.

A Babystep-Giantstep Method for Faster Deterministic Integer Factorization

Markus Hittmeir

University of Salzburg, Hellbrunnerstraße 34, 5020 Salzburg
markus.hittmeir@sbg.ac.at

We consider the problem of computing the prime factorization of integers. In practice, a large variety of probabilistic and heuristic methods is used for this task. However, none of these algorithms is efficient and the problem itself is assumed to be computationally hard. The difficulty of factoring large numbers is fundamental for the security of several cryptographical systems, one of which is the public-key scheme RSA.

A more theoretical aspect of integer factorization concerns deterministic algorithms and the rigorous analysis of their runtime complexities. In the years from 1974 to 1977, Pollard and Strassen developed such a method and proved that it runs in time $\widetilde{O}(N^{1/4})$. Since the seventies, the logarithmic factors in the bound have been refined and other deterministic algorithms running in $\widetilde{O}(N^{1/3})$ have been found. However, the bound $\widetilde{O}(N^{1/4})$ has been state of the art for the last forty years.

In this paper, we obtain an improvement by a superpolynomial factor. The runtime complexity of our algorithm is of the form

$$\widetilde{O}\left(N^{1/4}\exp(-C\log N/\log\log N)\right).$$

To describe our approach, we consider the case $N = pq$, where p and q are unknown prime factors and $p < q$. We will employ a refined babystep-giantstep method to solve the discrete logarithm problem $a^X \equiv a^{N+1} \bmod N$ for a certain $a \in \mathbb{Z}$ coprime to N. The purpose of this procedure is to determine $S := p+q$. Knowing S allows us to factor N immediately.

Let $\Delta \leq N^{1/2}$ be a parameter. The scheme for our main algorithm is as follows:

1. Use the Pollard-Strassen approach to search for p in the interval $[1,\Delta]$. If p is found, stop. If p is not found, go to Step 2.
2. Use $\Delta < p < N^{1/2}$ to find $S := p+q$, the sum of the prime factors of N.
3. Knowing N and S, compute p and q.

To speed up the application of the babystep-giantstep method in Step 2 and to optimize the value for Δ, we will consider so called modular hyperbolas $\mathcal{H}_{N,m}$. They are defined as the sets of solutions (x,y) to the congruence equation $N \equiv xy \bmod m$. Clearly, the corresponding set $\mathcal{L}_{N,m}$ consisting of the elements $x+y \bmod m$ for $(x,y) \in \mathcal{H}_{N,m}$ contains the residue of S modulo m. If r is prime, than the cardinality of $\mathcal{L}_{N,r}$ is about half of the possible residue classes modulo r. Considering all primes up to a suitable bound B, we deduce significant information about S. For example, let $N = 3823 \cdot 2069$ and $m = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11$. Then $\mathcal{L}_{N,m}$ contains only 40 elements. As a result, the residue of S modulo m is restricted to $40/2310 = 1.7\%$ of all residue classes modulo m. The information obtained by this idea yields the main contribution to our improvement.

A Crossbred Algorithm for Solving Boolean Polynomial Systems

Antoine Joux[1] and Vanessa Vitse[2]

[1] Chaire de Cryptologie de la Fondation de l'UPMC, Sorbonne Universités,
UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, Paris, France
antoine.joux@m4x.org

[2] Institut Fourier, Université Grenoble-Alpes, Grenoble, France
vanessa.vitse@univ-grenoble-alpes.fr

Abstract. We consider the problem of solving multivariate systems of Boolean polynomial equations: starting from a system of m polynomials of degree at most d in n variables, we want to find its solutions over $\mathbb{F}_2$. Except for $d = 1$, the problem is known to be NP-hard, and its hardness has been used to create public cryptosystems; this motivates the search for faster algorithms to solve this problem. After reviewing the state of the art, we describe a new algorithm and show that it outperforms previously known methods in a wide range of relevant parameters. In particular, the first named author has been able to solve all the Fukuoka Type I MQ challenges, culminating with the resolution of a system of 148 quadratic equations in 74 variables in less than a day (and with a lot of luck).

Contents

Invited Talk

Elliptic Curves in Cryptography

Public-Key Cryptography

Lattices in Cryptography

Number Theory

Pseudorandomness

Algebraic Structures and Analysis

Invited Talk

A Crossbred Algorithm for Solving Boolean Polynomial Systems

Antoine Joux[1](✉) and Vanessa Vitse[2]

[1] Chaire de Cryptologie de la Fondation de l'UPMC, Sorbonne Universités,
UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, Paris, France
antoine.joux@m4x.org

[2] Institut Fourier, Université Grenoble-Alpes, Grenoble, France
vanessa.vitse@univ-grenoble-alpes.fr

Abstract. We consider the problem of solving multivariate systems of Boolean polynomial equations: starting from a system of m polynomials of degree at most d in n variables, we want to find its solutions over $\mathbb{F}_2$. Except for $d = 1$, the problem is known to be NP-hard, and its hardness has been used to create public cryptosystems; this motivates the search for faster algorithms to solve this problem. After reviewing the state of the art, we describe a new algorithm and show that it outperforms previously known methods in a wide range of relevant parameters. In particular, the first named author has been able to solve all the Fukuoka Type I MQ challenges, culminating with the resolution of a system of 148 quadratic equations in 74 variables in less than a day (and with a lot of luck).

Keywords: Multivariate polynomial systems · Gröbner basis · XL
Multivariate cryptography · Algebraic cryptanalysis

1 Introduction

The resolution of systems of polynomial equations is a fundamental mathematical tool with numerous applications. It is well known that solving systems of multivariate equations is NP-hard in general, but it does not preclude from seeking the most efficient algorithms; besides, systems coming from applications are often easier to solve than predicted by the worst-case complexity. In this paper, we mostly focus on random instances which is presumably the hardest case.

Actually, there is a subtlety in the signification of "solving". Usually, it means finding all solutions of a given system, i.e. all tuples $(x_1, \ldots, x_n) \in K^n$ satisfying

$$\begin{cases} f_1(x_1, \ldots, x_n) = 0 \\ \qquad\vdots \\ f_m(x_1, \ldots, x_n) = 0 \end{cases}$$

J. Kaczorowski et al. (Eds.): NuTMiC 2017, LNCS 10737, pp. 3–21, 2018.
https://doi.org/10.1007/978-3-319-76620-1_1

where $f_1, \ldots, f_m$ are elements of $K[X_1, \ldots, X_n]$. This is mostly fine if the system has a finite number of solutions, or more precisely is zero-dimensional. *Mostly* because this approach ignores the solutions that may exist in a field extension or at infinity, and also because the solution set may be too large to be practically listed. In this latter case, or if the solution set has positive dimension, the alternative is to find a practical description of the corresponding algebraic variety, and Gröbner bases usually fill that role. Note that, in many applications, including cryptographic ones, it can be sufficient to find a single solution of a system. We also consider this weaker form of solving.

In this article, we focus on systems of quadratic (i.e. total degree 2) equations, as it is the simplest case beyond the polynomially-solvable linear case. The method we propose can also be applied to systems with a higher degree, but of course the complexity quickly grows with the degree. Note that there exists a general method to transform a system of arbitrary high degree equations into an equivalent quadratic system. This is done by introducing new variables to encode high degree monomials and new equations relating them. Due to the large number of new variables, combining this approach with the resolution of a quadratic system is usually very unefficient.

More importantly, our work focuses on the Boolean case, i.e. we are looking for solutions in $\mathbb{F}_2^n$ of systems of quadratic polynomials with coefficients in the field with two elements $\mathbb{F}_2$. This is relevant for applications in computer science, in coding theory and cryptography (see for instance [7,16,20]); furthermore, any polynomial system defined over a binary field $\mathbb{F}_{2^d}$ can be translated using Weil descent as a system over $\mathbb{F}_2$. The Boolean case has two important implications:

- Since we are looking for solutions defined over $\mathbb{F}_2$ and not an extension, we can add the field equations $x_i^2 + x_i = 0$ to the system. Equivalently, we can work in the Boolean polynomial ring $B[X_1, \ldots, X_n] = \mathbb{F}_2[X_1, \ldots, X_n]/(X_1^2 + X_1, \ldots, X_n^2 + X_n)$, where the equations become simpler since no variable may occur with (individual) degree equal to 2 or more.
- In small finite fields, exhaustive search becomes a viable option. This is obviously true for $\mathbb{F}_2$, but also in a lesser extent for other small finite fields such as $\mathbb{F}_3$ or $\mathbb{F}_5$. Our new algorithm, as most current algorithms for solving Boolean systems, partly relies on exhaustive search.

Despite this, Boolean quadratic systems still capture the NP-hardness of polynomial solving. In fact, because the 3-SAT problem can be reduced to the resolution of such systems [12], the existence of an algorithm with worst-case subexponential complexity would refute the Exponential Time Hypothesis [15], a conjecture in complexity theory, generalizing P $\neq$ NP and widely believed to be true.

For the analysis of our algorithm, we will consider systems of random equations, i.e. where the monomial coefficients are chosen independently and uniformly in $\{0, 1\}$. The behaviour of such systems differs according to the relative values of m and n [13]. If $m < n$ (there are more unknowns than equations), the system is underdetermined and admits on average $O(2^{n-m})$ solutions. If $m = n$, the system is determined, and has at least one solution with a probability converging to $1-1/e$ has n grows to infinity. If $m > n$ (there are more equations than unknowns) the system is overdetermined and has no solution with overwhelming probability.

But in practical applications such as cryptography, the polynomial systems, even when overdetermined, always have at least one solution. For this reason, we also consider random consistent systems, i.e. chosen uniformly from the set of systems of m quadratic Boolean polynomials in n variables with at least one solution in $\mathbb{F}_2^n$. Then when m is larger than n, this forced solution is unique with overwhelming probability.

2 State of the Art

2.1 Under- and Overdetermined Systems

Extremely overdetermined ($m > n(n+1)/2$) or underdetermined ($n > m(m+1)$) random Boolean quadratic systems can be solved in polynomial time. The first case simply requires Gaussian elimination on the equations and can be seen as a particular instance of the general approach presented in Sect. 2.4. The second case was solved by Kipnis et al. in [16]. At PKC 2012, Thomae and Wolf [21] have generalized the algorithm of Kipnis-Patarin-Goubin to other underdetermined systems, and their complexity interpolates between polynomial for $n > m(m+1)$ and exponential for n close to m.

Beyond these two extremes, the $m = n$ case is essentially the hardest. For $m > n$, the additional information given by the extra equations can simplify the problem. And when $n > m$, it is always possible to specialize $n - m$ variables (i.e. set them to arbitrary values) and get back to the case of as many equations as unknowns – at least, if we only seek a single solution which it usually the case for such underdetermined systems.

2.2 Exhaustive Search

Obviously, since there are 2^n possible values to instantiate n variables in $\mathbb{F}_2$, it is possible to evaluate the m polynomials for all values in order to find all solutions. At first glance, this costs $m \cdot 2^n$ evaluations of a degree d polynomial. However, this first estimation is too pessimistic. Optimizing the 2^n evaluations is quite subtle, but Bouillaguet et al. proposed in [3] a faster method that relies on the remark that if we know the evaluation of a polynomial at one point and only change the value of one variable, the evaluation at the new point can be computed faster. Their idea is based on the use of partial derivatives of the polynomials. Combined with the use of Gray codes and other techniques, it allows to find all solutions of a system of m Boolean quadratic equations in n variables in $O(\ln(n)2^n)$ elementary operations. Remarkably, this complexity does not depend of m; but obviously if only one solution is needed the search will finish faster for smaller m since there are more solutions then. This fast exhaustive search algorithm is implemented in the `libFES` library (http://www.lifl.fr/~bouillag/fes/) and holds several resolution records.

2.3 A Provable Method Faster than Exhaustive Search

Recently, Lokshtanov et al. [18] proposed a probabilistic method that outperforms exhaustive search asymptotically. Their idea stems from the following observation: $(x_1, \ldots, x_n) \in \mathbb{F}_2^n$ is a solution of the polynomial system generated by $f_1, \ldots, f_m$ if and only if $y = (x_{k+1}, \ldots, x_n)$ is a solution of the equation

$$\prod_{a \in \mathbb{F}_2^k} (1 - \prod_{i=1}^{m} (1 - f_i(a, y))) = 0.$$

Instead of working with this unwieldly polynomial, they consider its probabilistic counterpart

$$R(y) = \sum_{a \in \mathbb{F}_2^k} t_a \prod_{i=1}^{l} (1 - \sum_{j=1}^{m} s_{aij} f_j(a, y))$$

where s_{aij} and t_a are chosen independently and uniformly in $\mathbb{F}_2$ and $l \leq m$ is a parameter. If y is the last part of a solution, then $R(y)$ is uniformly distributed in $\mathbb{F}_2$, but otherwise $R(y) = 0$ with a probability greater than $(1 - 2^{-l})^{2^k}$. By performing several complete evaluations of R on all its 2^{n-k} input values of y, for varying coefficients s_{aij}, t_a, it is possible to recover with high probability the last part of all the solutions of the system. Overall, the complexity is in $\tilde{O}(2^{0.8765n})$, faster than the brute force approach.

As far as we know, this method as not been implemented and it seems unlikely that it outperforms exhaustive search in the range of systems which can be solved with current computer. However, it is remarkable that it asymptotically beats brute force without relying on any heuristic hypothesis concerning the given system. An unfortunate consequence is that the method cannot take advantage of a large value of m compared to n, since it would necessarily requires some hypothesis of “independence” between the equations. Indeed, if we don’t care about independence, it is easy to add extra equations by taking linear combinations of the initial ones. As a final remark, one should note that the algorithm of Lokshtanov et al. makes the assumption that the number of solutions of the system is smaller than $2^{0.8765n}$, since otherwise, it would not be possible to list all of them in the indicated complexity.

2.4 Algebraic Methods

Algebraic methods consider systems of polynomial equations by looking at the ideals they generate and try to solve them by finding a good representation of the corresponding ideal. More precisely, let $\mathcal{F} = \{f_1, \ldots, f_m\}$ be a family of elements in a multivariate polynomial ring $K[X_1, \ldots, X_n]$ and form the ideal $I = \langle f_1, \ldots, f_m \rangle$ generated by the family $\mathcal{F}$. By definition, I is the following set of polynomials:

$$I = \left\{ \sum_{i=1}^{m} p_i f_i \mid (p_1, \ldots, p_m) \in K[X_1, \ldots, X_n]^m \right\}.$$

Thus, for any element f of the ideal I, there exist polynomials $p_1, \ldots, p_m$ such that $f = \sum_{i=1}^{m} p_i f_i$; in other words, there exists an integer $D = \max\{\deg p_i : 1 \leq i \leq m\}$ such that f belongs to the vector space

$$V_{\mathcal{F},D} = Span_K \left\{ uf_i | \, i \in [1;m]; u \text{ a monomial with} \deg u \leq D - \deg f_i \right\}.$$

Macaulay matrices. T he above observation implies that relevant information on the ideal I can be obtained by studying these vector spaces and motivates the following definition.

Definition 1. *For any integer k, let T_k be the set of monomials of $K[X_1, \ldots, X_n]$ of degree smaller than or equal to k. The degree D Macaulay matrix of $\mathcal{F}$, denoted by $Mac_D(\mathcal{F})$, is the matrix with coefficients in K whose columns are indexed by T_D, whose lines are indexed by the set $\left\{(u, f_i) \left| i \in [1;m]; u \in T_{D-\deg(f_i)} \right.\right\}$, and whose coefficients are those of the products uf_i in the basis T_D.*

Macaulay matrices can be thought as multivariate analogs of the classical Sylvester matrix. Lazard first showed in [17] that they can be used to compute Gröbner bases: for any monomial order $\succ$, there exists a degree D such that if the columns of $\mathrm{Mac}_D(\mathcal{F})$ are sorted according to $\succ$, the rows of its reduced echelon form contains the coefficients of a Gröbner basis of I. This idea of expressing many multiples of a family of polynomials in matrix form and reducing the resulting matrices is at the heart of most current algorithms for computing Gröbner bases, such as F4, F5, XL and their many variants [6,9,10].

When K is equal to $\mathbb{F}_2$, we usually want to add the field equations $X_i^2 = X_i$ for all $i \in [1;m]$. As stated before, it is more efficient to work directly in the quotient algebra $B[X_1, \ldots, X_n] = \mathbb{F}_2[X_1, \ldots, X_n]/(X_1^2 + X_1, \ldots, X_n^2 + X_n)$ (B stands for Boolean). The definition can be adapted by requiring that every monomial (either in T_k or in the products uf_i) has degree strictly smaller than 2 in each variable. Of course, we can proceed in a similar way when working over $\mathbb{F}_q$ with q small.

In many situations, the system $f_1 = \cdots = f_m = 0$ is overdetermined and so has none or very few solutions. This implies that the ideal $I = \langle f_1, \ldots, f_m \rangle$ will contain 1 (if there is no solution) or linear polynomials, from which it is easy to deduce the solutions. Again, such low degree equations can be obtained by reducing the Macaulay matrix $\mathrm{Mac}_D(\mathcal{F})$, with its columns sorted by total degree, for some degree D. The smallest such integer D is called the *degree of regularity* of the system and denoted by D_{reg}. (Note that this only one out of many other definitions of D_{reg}.)

With this approach, solving an overdetermined system of Boolean quadratic polynomials amounts to computing the row echelon form of a large matrix, for a total cost in

$$\tilde{O}\left(\binom{n}{D_{reg}}^{\omega}\right),$$

where ω is the exponent of matrix multiplication (smallest known value is $\omega = 2.373$; in practice $\omega = 2.807$ with Strassen algorithm). But this Macaulay matrix is extremely sparse: by design, it has at most $1 + n(n+1)/2$ non zero coefficients per row, which is negligible compared to its number of columns when n goes to infinity (as soon as $D > 2$, of course). This suggests that instead of Gaussian elimination, sparse linear algebra techniques such as block Lanczös algorithm [19] or block Wiedemann algorithm [22] could be used. Indeed, it is possible to probabilistically test the consistency of a Boolean quadratic system in $\tilde{O}\left(\binom{n}{D_{reg}}^2\right)$ and to find a (small number of) solution(s) if any exists. It remains an open problem to find all solutions with the same complexity, when there are many.

However, determining the degree of regularity is not straightforward, although a practical option is to reduce several Macaulay matrices in increasing degrees until enough linear polynomials have been found. Asymptotic estimates exist for an important class of systems, called "semi-regular"; heuristic arguments and experimental evidence suggest that random systems fall in this class with overwhelming probability. In this case, Bardet et al. showed in [2] that if $m \sim \alpha n$ ($\alpha \geq 1$ fixed), as n goes to infinity, then $D_{reg} \sim M(\alpha)\, n$ where $M(\alpha)$ is an explicit decreasing function of α. In particular for $\alpha = 1$, with $\omega = 2$ this yields an asymptotic complexity of $\tilde{O}(2^{0.8728n})$, faster than exhaustive search and even than Lokshtanov et al. But this complexity is conditional to the semi-regularity of the system: it is conjectured to hold with probability converging to 1 as n grows, but exceptional systems may be harder to solve with this technique. By contrast, the complexity of the methods of Sects. 2.2 and 2.3 does not rely on any assumption.

In practice, computing the row echelon form of the D_{reg} Macaulay matrix of $f_1, \ldots, f_m$ is too costly to be efficient. In particular, for the Boolean case, it has been estimated (see [3]) that these methods would not outperform exhaustive search for any value of n smaller than 200. Nevertheless, algebraic algorithms have proven themselves to be very efficient on specific systems with extra algebraic properties which imply a low degree of regularity. A striking example is given by systems arising in the Hidden Field Equations cryptosystem [20]. In this case, a consequence of the presence of a hidden backdoor is a degree of regularity smaller than expected [8,14], leading to devastating attacks [11].

The FXL and BooleanSolve hybrid Algorithms. Lazard's resolution method [17] was rediscovered as the XL algorithm fifteen years later by Courtois et al. in [6]. This last paper also introduced a variant called FXL, which combines exhaustive search with linear algebra on a Macaulay matrix and improves on the above algebraic technique. It takes as input the family $\mathcal{F} = \{f_1, \ldots, f_m\} \subset B[X_1, \ldots, X_n]$ of Boolean polynomials, a parameter $k \leq n$ and proceeds as follows:

1. For each $a = (a_{k+1}, \ldots, a_n) \in \mathbb{F}_2^{n-k}$, compute the specialized polynomials $f_{1,a}, \ldots, f_{m,a}$ where $f_{i,a} = f_i(X_1, \ldots, X_k, a_{k+1}, \ldots, a_n) \in B[X_1, \ldots, X_k]$.
2. Using the Macaulay matrix of $f_{1,a}, \ldots, f_{m,a}$ in degree D_{reg}, check if the specialized system $f_{1,a} = \cdots = f_{m,a} = 0$ admits a solution. If no,

continue with the next value of $a \in \mathbb{F}_2^{n-k}$; otherwise, find the solution $(x_1, \ldots, x_k, a_{k+1}, \ldots, a_n)$ using e.g. exhaustive search on $x_1, \ldots, x_k$.

A first complexity analysis of FXL was given in [23,24]. Specializing the equations allows to dramatically reduce the size of the Macaulay matrices, not only because the number of variables diminishes, but also because the degree of regularity decreases as the ratio between the number of equations and the number of variables goes up. Of course, it induces a factor 2^{n-k} in the complexity, corresponding to the number of times the second step has to be executed.

In this second step, for most values of a the specialized system will have no solution, meaning that 1 is in the ideal. As discussed above, it is possible to take advantage of the sparsity of the Macaulay matrix in testing this property; this was done in [4]. Indeed, the full row echelon form of the matrix is not needed; one just has to test whether a constant polynomial can be found in the Macaulay matrix, using for instance a probabilistic method based on block Lanczös algorithm.

Bardet et al. give a thorough analysis of this hybrid approach (renamed BooleanSolve algorithm) in [1]. Under the assumption that the specialized systems still behave like random ones—more precisely, that they remain semi-regular ("strong semi-regularity")—it is possible to derive a complexity estimate. In the case $m \sim \alpha n$ ($\alpha \geq 1$ fixed and n going to infinity), for $\alpha < 1.82$ the asymptotically best choice is $k \approx 0.55\,\alpha n$, for a complexity in $\tilde{O}(2^{(1-0.208\alpha)n})$. In particular, for $\alpha = 1$ this yields a (conditional) complexity of $\tilde{O}(2^{0.792n})$. For $\alpha \geq 1.82$ the asymptotically best choice is $k = n$, i.e. no variables are specialized: the system is too overdetermined for the algorithm, and it does not improve on the standard reduction of the full Macaulay matrix.

3 Our Crossbred Algorithm

3.1 General Principle

In the FXL/BooleanSolve algorithm, the most costly step is the linear algebra of the Macaulay matrix, which is performed 2^{n-k} times. In order to avoid this problem, we propose a new method that performs the specialization step on $n-k$ variables *after* working with the Macaulay matrix.

Basic idea. A first idea is to construct a degree D Macaulay matrix, sort its columns in lexicographical order, then compute the last rows of its row echelon form. This allows to generate degree D equations in which k variables have been eliminated, and this resulting system can then be solved using exhaustive search on $n - k$ variables. As a toy example, we can consider the following system:

$$\begin{cases} X_1X_3 + X_2X_4 + X_1 + X_3 + X_4 = 0 \\ X_2X_3 + X_1X_4 + X_3X_4 + X_1 + X_2 + X_4 = 0 \\ X_2X_4 + X_3X_4 + X_1 + X_3 + 1 = 0 \\ X_1X_2 + X_1X_3 + X_2X_3 + X_3 + X_4 + 1 = 0 \\ X_1X_2 + X_2X_3 + X_1X_4 + X_3 = 0 \\ X_1X_3 + X_1X_4 + X_3X_4 + X_1 + X_2 + X_3 + X_4 = 0 \end{cases}$$

The corresponding degree 2 Macaulay matrix, in lex order, is

$$\begin{array}{ccccccccccc} X_1X_2 & X_1X_3 & X_1X_4 & X_1 & X_2X_3 & X_2X_4 & X_2 & X_3X_4 & X_3 & X_4 & 1 \\ \hline 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{array}$$

and its reduced row echelon form is

$$\begin{array}{ccccccccccc} X_1X_2 & X_1X_3 & X_1X_4 & X_1 & X_2X_3 & X_2X_4 & X_2 & X_3X_4 & X_3 & X_4 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{array}$$

We obtain two equations not involving X_1, namely $X_2X_3 + X_3X_4 + X_3 + X_4 + 1 = 0$ and $X_2X_4 + X_2 = 0$, which can be solved for instance with exhaustive search; the solutions thus found must then be checked for compatibility with the remaining equations in X_1.

An obvious drawback of this method is that in order to eliminate a significant number of variables, the degree D should be taken large enough, and reducing large Macaulay matrices is quickly prohibitive.

A more refined variant. An important remark is that it is not necessary to completely eliminate k variables. We now illustrate this with the same example. First, we sort the columns, this time according to the graded reverse lexicographic order (grevlex), and obtain the following row echelon form:

$$\begin{array}{ccccccccccc} X_1X_2 & X_1X_3 & X_2X_3 & X_1X_4 & X_2X_4 & X_3X_4 & X_1 & X_2 & X_3 & X_4 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \end{array}$$

The last three equations have degree 1 in X_1, X_2, X_3:

$$\begin{cases} (X_4+1)X_1 + X_2 + X_3 + 1 = 0 \\ (X_4+1)X_2 = 0 \\ X_1 + X_2 + (X_4+1)X_3 + 1 = 0 \end{cases}$$

Consequently, for any assignation of the last variable, we obtain a system that can be easily solved using linear algebra. Reducing the same Macaulay matrix, we have thus "eliminated" three variables from the exhaustive search procedure. This is somewhat reminiscent of Kipnis-Goubin-Patarin algorithm [16] for solving extremely underdetermined quadratic systems, whose principle is also to generate enough equations of the form

$$P_1(X_{k+1},\ldots,X_n)X_1 + \cdots + P_k(X_{k+1},\ldots,X_n)X_k + Q(X_{k+1},\ldots,X_n) = 0,$$

yielding a linear system once the variables $X_{k+1},\ldots,X_n$ are specialized.

3.2 Description of the Algorithm

Our algorithm implements this idea in a scalable way. It depends on three parameters D, d and k, with $D \geq 2, 1 \leq d < D$ and $1 \leq k \leq n$. To simplify the description, for any polynomial $p \in B[X_1,\ldots,X_n]$, we let $\deg_k p$ stand for the total degree in $X_1,\ldots,X_k$.

When $d = 1$, it works as proposed above: from the degree D Macaulay matrix (sorted by decreasing value of $\deg_k$), we generate new equations that are linear in $X_1,\ldots,X_k$, i.e. we eliminate all monomials of degree larger than 1 in these variables. This can be achieved by computing elements in the left kernel of the truncated matrix, from which the monomials containing at most one of the variables $X_1,\ldots,X_k$ have been removed. The choice of D is a critical parameter, it must be large enough for reduced equations to exist and as small as possible if we want the dimension of the Macaulay matrix to remain manageable. Note that, since the new equations become linear after performing the evaluation of variables X_{k+1} to X_n, it is sufficient to have a little more than k equations of this form.

To extend this to larger values of d, we want to construct new equations of degree at most d in the first variables $X_1,\ldots,X_k$. For large systems, this allows to select smaller values of D and to work with smaller Macaulay matrices. However, the number of equations that we need in this context to solve the system after specialization through linear algebra becomes larger. Of course, when d is equal to or larger than the degree of the initial equations, these initials equations can be included in the pool of equations that we are keeping for specialization.

The main difficulty of this method is to analyze the optimal choices of parameters D, d and k for given values of the number of variables n, the number of equations m and the degree of these equations (2 if we restrict ourselves to quadratic systems).

We give below a pseudo-code description of the algorithm. The algorithm considers the two following submatrices of the full degree D Macaulay matrix:

- $\mathrm{Mac}^{(k)}_{D,d}(\mathcal{F})$ is the submatrix of $\mathrm{Mac}_D(\mathcal{F})$ whose rows correspond to products uf_i with $\deg_k u \geq d-1$
- $M^{(k)}_{D,d}(\mathcal{F})$ is the submatrix of $\mathrm{Mac}^{(k)}_{D,d}(\mathcal{F})$ whose columns correspond to monomials $\textsc{m}$ with $\deg_k m > d$.

Basically, the algorithm works as follows:

1. Search elements $v_1, \ldots, v_r$ in the kernel of $M^{(k)}_{D,d}(\mathcal{F})$.
2. Compute the polynomials p_i corresponding to $v_i.\mathrm{Mac}^{(k)}_{D,d}(\mathcal{F})$; they have total degree at most D, and at most d in $X_1, \ldots, X_k$.
3. For all $a = (a_{k+1}, \ldots, a_n) \in \mathbb{F}_2^{n-k}$:
   (a) Create the degree d Macaulay matrix $\mathrm{Mac}_d(\mathcal{F}^*)$ corresponding to the polynomials in $\mathcal{F}$ (partially) evaluated at a
   (b) Evaluate the polynomials p_i at a and append them to $\mathrm{Mac}_d(\mathcal{F}^*)$
   (c) Check if the resulting system (of degree d) if solvable in $X_1, \ldots, X_k$.

As a further refinement, it is possible to add an outer layer of hybridation. Indeed, we can start by iterating through the possible values of the h last variables $X_{n-h+1}, \ldots, X_n$, and apply the above algorithm 2^h times to the specialized systems of m quadratic equations in $n-h$ variables. The main interest of this outer hybridation is to allow an easy parallelization between 2^h computers and sometimes to offer a slightly better choice of parameters (see Sect. 3.3). Nevertheless, in some sense, it goes against the philosophy of the algorithm and we do not expect this parameter to be asymptotically useful.

Note that the idea of reducing a part of a large Macaulay matrix before specializing variables was already suggested by Courtois in [5], but in a rather different form. However, his algorithm seems to be unefficient according to the analysis given in [23].

3.3 Valid Parameters and Asymptotic Analysis

The parameters D, d and k (and h when outer hybridation is used) control the course of the algorithm, but finding optimal (or even functional) values is far from obvious. As a first remark, since we want to find new relations of degree at most d in the first k variables, cancellations of the highest degree parts in $X_1, \ldots, X_k$ must occur. Thus under a strong semi-regularity assumption, we

Algorithm 1. The crossbred algorithm

```
procedure System Resolution($\mathcal{F} = (f_1, \ldots, f_m)$)
                ▷ System of $m$ equations in $n$ variables. Parameters $D, d$ and $k$.
    Construct $\mathrm{Mac}_{D,d}^{(k)}(\mathcal{F})$ and $M_{D,d}^{(k)}(\mathcal{F})$
    Find $r$ linearly independent elements $(v_1, \ldots, v_r)$ in the (left) kernel of $M_{D,d}^{(k)}(\mathcal{F})$.
                ▷ Using (sparse) linear algebra.
    For all $i \in [1; r]$ compute the polynomial $p_i$ corresponding to $v_i.\mathrm{Mac}_{D,d}^{(k)}(\mathcal{F})$.
                ▷ Polynomials of total degree at most $D$ and degree at most $d$ in $(X_1, \ldots, X_k)$.
    Perform fast evaluation on $(f_1, \ldots, f_m, p_1, \ldots, p_r), n, k$ with
        Callback procedure
                ▷ Get $(f_1^*, ..., f_m^*, p_1^*, ..., p_r^*)$ evaluated at each $(x_{k+1}, ..., x_n) \in \{0,1\}^{n-k}$
            Construct the Macaulay matrix $\mathrm{Mac}_d(\mathcal{F}^*)$ of degree $d$ from $(f_1^*, \ldots, f_m^*)$
            Append $(p_1^*, \ldots, p_r^*)$ to $\mathrm{Mac}_d(\mathcal{F}^*)$
            Use (dense) linear algebra to test the consistency of resulting system,
                ▷ As in XL every monomial is viewed as an independent variable.
            if System is consistent then
                Extract values of $(X_1, \ldots, X_k)$ and test the candidate solution.
                Print any valid solution.
            end if
        end callback
end procedure
```

Algorithm 2. Fast Evaluation of a polynomial (over $\mathbb{F}_2$)

```
procedure Fast Evaluation($(P_1, \ldots, P_R), \ell, k$, Callback action)
                ▷ Polynomials of degree $D$ in $\ell$ variables.
    if $\ell = k$ then
        Perform Callback action on $(P_1, \ldots, P_R)$ and $(x_{k+1}, \ldots, x_n)$
    else
        Write each $P_i$ as $P_i^{(0)} + X_\ell \cdot P_i^{(1)}$
        Let $x_\ell \leftarrow 0$
        Fast evaluate on $(P_1^{(0)}, \ldots, P_R^{(0)}), \ell - 1, k$ and Callback action.
        Let $x_\ell \leftarrow 1$
        Fast evaluate on $(P_1^{(0)} + P_1^{(1)}, \ldots, P_R^{(0)} + P_R^{(1)}), \ell - 1, k$ and Callback action.
    end if
end procedure
```

obtain that the parameter D must be greater than or equal to the degree of regularity of a semi-regular system of m equations in k variables.

In addition to that, we need (under a regularity assumption) to compute the number of equations that can be obtained for the final linearized system and check that it is at least[1] equal to the number of monomials in the first k variables of degree at most d. We now explain how this is done in the case where $d = 1$ and $D = 3, 4$ that covers all of the reported experiments and seems to be the only viable choice for any feasible computation.

[1] Having a bit more equations is even better, since this leads to a smaller number of consistent systems of evaluation that lead to a finally incorrect solution.

With $d = 1$, the matrix $\mathrm{Mac}_d(\mathcal{F}^*)$ is empty and the linear algebra is simply performed on the evaluated linear polynomials $(p_1^*, \dots, p_r^*)$ in k variables. Thus it suffices to check that $r \geq k + 1$. As a consequence, we need enough linearly independent elements in the kernel of $M_{D,1}^{(k)}(\mathcal{F})$ which are not in the kernel of $\mathrm{Mac}_{D,1}^{(k)}(\mathcal{F}) = \mathrm{Mac}_D(\mathcal{F})$ (otherwise we get the trivial equation $0 = 0$). A lower bound on that number is simply given by the rank $R_{D,1}$ of $\mathrm{Mac}_D(\mathcal{F})$ minus the number of columns of $M_{D,1}^{(k)}(\mathcal{F})$.

The number $N_{D,d}^{(k)}$ of columns of $M_{D,d}^{(k)}(\mathcal{F})$ corresponds to the number of monomials labeling its columns and is given by the formula:

$$N_{D,d}^{(k)} = \sum_{d_k=d+1}^{D} \sum_{d'=0}^{D-d_k} \binom{k}{d_k}\binom{n-k}{d'}.$$

The number of independent rows of $\mathrm{Mac}_{D,1}^{(k)}(\mathcal{F}) = \mathrm{Mac}_D(\mathcal{F})$ is simple to evaluate when $D = 3$. In that case and in our range of values of (m, n), under the regularity assumption the matrix has full rank. Since every polynomial in $\mathcal{F}$ is multiplied by the monomials $1, x_1, x_2, \dots x_n$, there are $R_{3,1} = (n + 1) \cdot m$ rows. For $D = 4$, it is slightly more complicated, because we need to account for the trivial relations of the form $f_i f_j + f_j f_i = 0$ and $(f_i + 1) f_i = 0$. As a consequence, the Macaulay matrix $\mathrm{Mac}_D(\mathcal{F})$ has $R_{4,1} = (1 + n + n\,(n - 1)/2) \cdot m - m\,(m + 1)/2$ independent rows. Table 1 illustrates this on a few parameters extracted from Sect. 4.

Table 1. Examples of parameters' computation

n_0	m	h	$n = n_0 - h$	k	D	$N_{D,1}^{(k)}$	$R_{D,1}$	Exp. num of polys
35	35	0	35	9	3	1056	1260	204
35	70	0	35	14	3	2366	2520	154
41	41	0	41	11	4	31075	34481	3406
41	82	0	41	15	3	3290	3444	154
74	148	12	62	23	4	277288	278166	878

For $d > 1$, and under semi-regularity hypotheses, the number $R_{D,d}(\mathcal{F})$ of new independent polynomials coming from the reduction of the matrix $\mathrm{Mac}_{D,d}^{(k)}(\mathcal{F})$ can also be expressed using binomial coefficients, or more concisely as the coefficient in $X^D Y^d$ of the bivariate generating series

$$S_{D,d}(X, Y) = \frac{(1 + X)^{n-k}}{(1 - X)(1 - Y)} \left(\frac{(1 + XY)^k}{(1 + X^2 Y^2)^m} - \frac{(1 + X)^k}{(1 + X^2)^m} \right).$$

This gives a way to test the admissibility of the parameters: k, D and d are admissible if the coefficient of $X^D Y^d$ of the series

$$S_{D,d}(X, Y) - \frac{(1 + Y)^k}{(1 - X)(1 - Y)(1 + Y^2)^m}$$

is non-negative; this series has been used to find optimal parameters for the crossbred algorithm complexity given in Figs. 2, 3 and 4.

This expression can also be used for asymptotic analysis. When n grows to infinity, and $m = \lfloor \alpha n \rfloor, D = \lfloor \varDelta n \rfloor, d = \lfloor \delta n \rfloor, k = \lfloor \kappa n \rfloor$ with fixed $\alpha \leq 1, 0 < \delta \leq \varDelta < 1, 0 < \kappa < 1$, the asymptotic behavior of the coefficient of $X^{\lfloor \varDelta n \rfloor} Y^{\lfloor \delta n \rfloor}$ can be obtained using adaptations of the saddle-point method. This gives an asymptotic range of admissible parameters, among which it remains to optimize the overall complexity. However, we have found that for these asymptotically large values, the optimal choice is $\varDelta = \delta$, and the crossbred algorithm degenerates to FXL/BooleanSolve; thus our algorithm does not improve the asymptotic complexity of solving multivariate quadratic Boolean systems[2].

Despite this, for tractable numbers of equations and variables our experiments demonstrated that the crossbred algorithm is much more efficient than any other known methods, see next section for detailed reports. Our estimations indicate that for moderately overdetermined systems, it is only for very large numbers of equations and variables (several hundreds, see Figs. 2, 3 and 4) that the optimal choice of parameters makes our algorithm equivalent to FXL/BooleanSolve; the resolution of such systems is obviously completely out of reach with current computers.

4 Experiments and Timings

4.1 Implementation Specifics

In our implementation, the sparse linear algebra in the first phase of the crossbred algorithm is performed using the block Lanczös algorithm of Montgomery [19]. The multiplication of matrices are applied in parallel on 128 vectors, taking advantage of the 128-bit integers available in modern CPUs.

Similarly, the fast polynomial evaluation and the resolution of the linear systems in the second phase work in parallel using 128-bit operations. In addition, since in practice we have $d = 1$, the linear systems in phase 2 are small enough to be solved using a quadratic number of 128-bit word operations.

4.2 Fukuoka Type I MQ Challenge

In order to experimentally test this new algorithm, we decided to tackle the Fukuoka MQ Challenges [25]. These challenges, available on the website https://www.mqchallenge.org/, were issued in 2015 with the explicit goal to help assess the hardness of solving systems of quadratic equations. The Type I challenges consist of $2n$ Boolean quadratic equations in n variables, and the designers have ensured that every system has some forced solution. At the time we started, the record on $n = 66$ was held by Chou, Niederhagen and Yang, using a fast Gray code enumeration technique on an array of FPGA. It took a little less than 8 days

[2] Because of this negative result coupled with page limitations, we chose not to include the derivation of the above series and its coefficient asymptotics in this article.

using 128 Spartan 6 FPGAs. It is interesting to note that this allows for a much faster resolution than on CPU based computation. According to our estimations, using libFES the same computation would have taken about 61 000 cpu · days using Intel Core i7 processors at 2.8 GHz. Note that as mentioned before, the BooleanSolve algorithm on such systems performs no exhaustive search and boils down to working with the full Macaulay matrix, for an asymptotic complexity in $\tilde{O}(2^{0.585n})$.

We found the solution of all the remaining challenges $n = 67 \ldots 74$ by running our code on a heterogenous network of computers at the LIP6 laboratory. Due to this heterogeneity, the timings are not very precise and the running of identical jobs greatly varied depending on the individual machine it runned on. The processors types in the cluster ranged from Opteron 2384 at 2.8 GHz to Xeon 2690 at 2.6 GHz (the latter being about four times faster). Timings[3] are given in Table 2.

Table 2. Fukuoka challenge

Number of vars ($m = 2n$)	External hybridation h	Parameters $(D, n-h-k)$	Max CPU (estimate)	Real CPU (rounded)
67	9	$(4, 36)$	6 200 h	3 100 h
68	9	$(4, 37)$	11 200 h	4 200 h
69	9	$(4, 38)$	15 400 h	15 400 h
70	13	$(4, 34)$	33 000 h	16 400 h
71	13	$(4, 35)$	60 000 h	13 200 h
72	13	$(4, 36)$	110 000 h	71 800 h
73	13	$(4, 37)$	190 000 h	14 300 h
74	12	$(4, 39)$	360 000 h	8 100 h

4.3 Crossover Point Compared to Fast Enumeration when $m = n$

In addition to the above records, which take advantage of having twice as many equations as variables, it is interesting to compare the relative performances of our algorithm and fast enumeration when $m = n$. We ran experiments on Intel Core i7 laptop at 2.8 GHz for values of n ranging from 35 to 46. For the fast enumeration, we used the state of the art library libFES. The results are summarized in Table 3 and represented in Fig. 1. It makes clear that the cross-over point is at $n = 37$ when $m = n$. The table also contains timings of our code when $m = 2n$, in order to illustrate the gain in terms of running time when extra equations are available. Since our code is much less optimized than libFES, the cross-over point value might be slightly pessimistic.

[3] As remarked in the abstract, the last two entries in this table correspond to extremely lucky running times. The desired solution just happened to be found by the first series of parallel jobs.

Table 3. Comparison with libFES (Timings for full enumeration of search space)

Number of vars	libFES	Our code $(m = n)$	Parameters $(D, n - k)$	Our code $(m = 2n)$	Parameters $(D, n - k)$
35	2.3 s	3.8 s	$(3, 26)$	0.6 s	$(3, 21)$
36	4.9 s	7.2 s	$(3, 27)$	0.9 s	$(3, 22)$
37	9.6 s	9.5 s	$(3, 27)$	1.5 s	$(3, 23)$
38	20.1 s	16.5 s	$(3, 28)$	2.5 s	$(3, 24)$
39	40.2 s	33 s	$(3, 29)$	2.7 s	$(3, 24)$
40	84 s	65 s	$(3, 30)$	4.8 s	$(3, 25)$
41	162 s	131 s	$(4, 30)$	9 s	$(3, 26)$
42	317 s	242 s	$(4, 31)$	18 s	$(3, 27)$
43	642 s	437 s	$(4, 32)$	36 s	$(3, 28)$
44	1380 s	850 s	$(4, 33)$	71 s	$(3, 29)$
45	2483 s	989 s	$(4, 33)$	146 s	$(3, 30)$
46	5059 s	1905 s	$(4, 34)$	151 s	$(3, 30)$

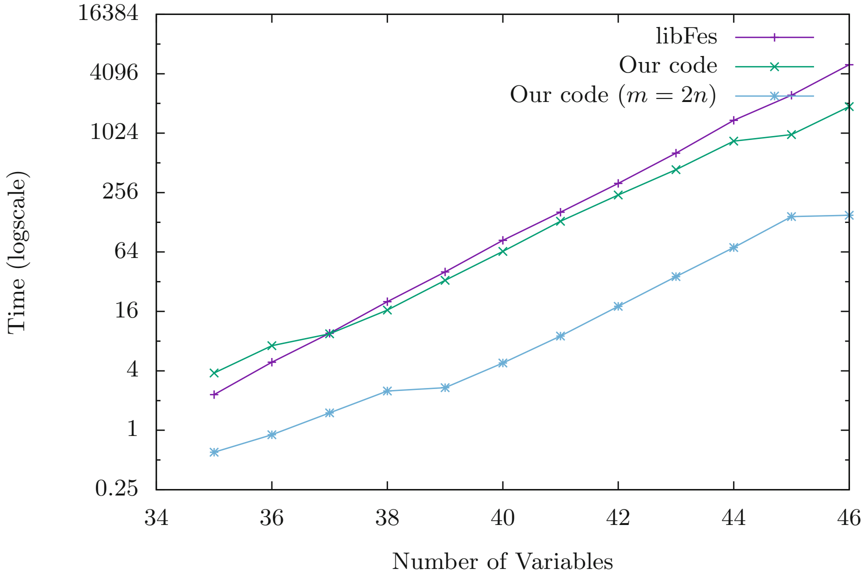


Fig. 1. Comparison with libFES

4.4 Complexity Estimates for Cryptographic Sizes

To illustrate the efficiency of the crossbred algorithm in the cryptographic range and help crypto-designers to assess its impact on their parameter choices, we give in Figs. 2, 3 and 4 the comparisons between the complexities of Macaulay

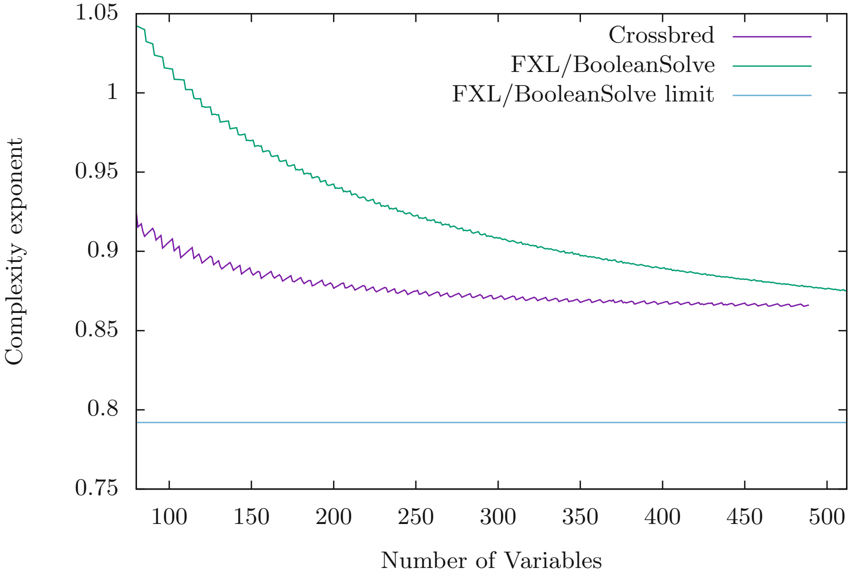


Fig. 2. Comparison between the FXL/BooleanSolve and crossbred algorithm complexities when $\alpha = 1$.

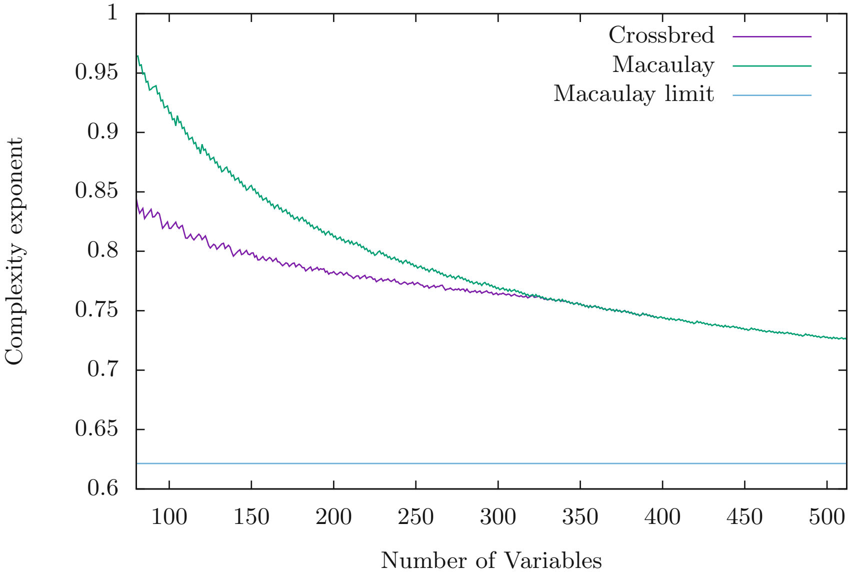


Fig. 3. Comparison between the Macaulay and crossbred algorithm complexities when $\alpha = 1.82$

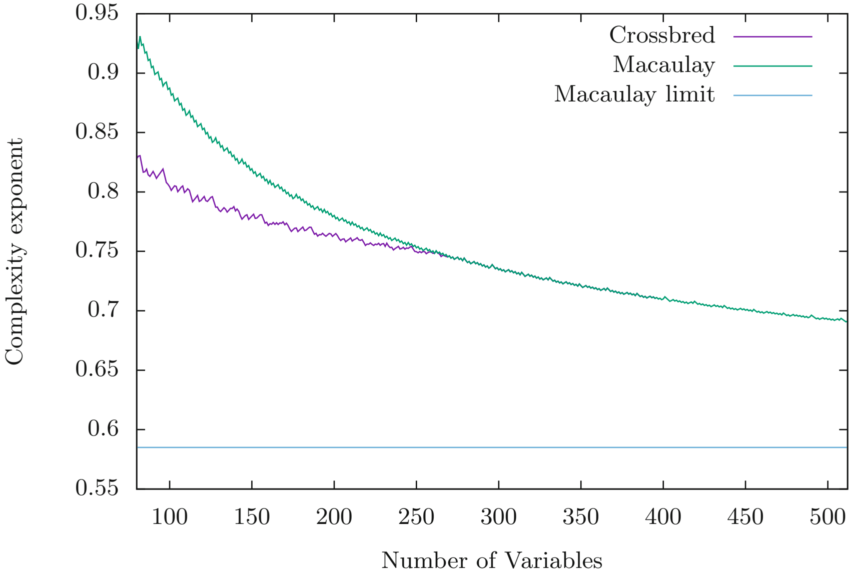


Fig. 4. Comparison between the Macaulay and crossbred algorithm complexities when $\alpha = 2$

(or FXL/BooleanSolve when $1 \leq \alpha < 1.82$) and our crossbred algorithms, focused on three specific values of α. The graphs also show the corresponding asymptotic exponent coming from the analysis of [1], which is only reached for extremely large values of n.

5 Conclusion

In this article, we have presented a new "crossbred" algorithm for solving systems of Boolean equations, using both exhaustive search and the ideal-based approach of Lazard. The main idea of the new algorithm is to reduce a partial Macaulay matrix before a specialization of part of the variables.

We have demonstrated that our mixed approach decisively beats the fast enumeration technique of [3] for large real-world (over)determined systems of Boolean quadratic polynomials. In particular, we have been able to solve all the Fukuoka Type I MQ Challenges [25] up to the last system of 148 quadratic equations in 74 variables, whereas the previous record using fast enumeration consisted in the resolution of a system of 132 equations in 66 variables. Note that, for such parameters ($m = 2n$) the hybrid BooleanSolve algorithm of [1] is optimal with an empty hybridation and thus becomes equivalent to the classical Lazard method [17]. In fact, the asymptotic analysis of these algorithms are quite technical, especially for the crossbred method. Moreover, the asymptotic regime is only reached for values of n which are far beyond any accessible or cryptographically interesting sizes. Despite its practical performance, in that asymptotic context the crossbred algorithm does not seem to offer an improved complexity.

In particular, as mentioned in [1,3], pre-existing algebraic methods are not expected to beat brute force for $n = m$ and n lower than 200. Yet, we have demonstrated that in practice the crossover point between exhaustive search and our method is $n = 37$.

We have only implemented and tested the case of quadratic systems over $\mathbb{F}_2$. However, the same principle applies to higher degree and other (small) finite fields of coefficients.

References

1. Bardet, M., Faugère, J.-C., Salvy, B., Spaenlehauer, P.-J.: On the complexity of solving quadratic Boolean systems. J. Complex. **29**(1), 53–75 (2013)
2. Bardet, M., Faugère, J.-C., Salvy, B., Yang, B.-Y.: Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems. Presented at MEGA 2005, Eighth International Symposium on Effective Methods in Algebraic Geometry (2005)
3. Bouillaguet, C., Chen, H.-C., Cheng, C.-M., Chou, T., Niederhagen, R., Shamir, A., Yang, B.-Y.: Fast exhaustive search for polynomial systems in $\mathbb{F}_2$. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 203–218. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15031-9_14
4. Cheng, C.-M., Chou, T., Niederhagen, R., Yang, B.-Y.: Solving quadratic equations with XL on parallel architectures. In: Prouff, E., Schaumont, P. (eds.) CHES 2012. LNCS, vol. 7428, pp. 356–373. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33027-8_21
5. Courtois, N.T.: Algebraic attacks over $GF(2^k)$, application to HFE challenge 2 and Sflash-v2. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 201–217. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24632-9_15
6. Courtois, N., Klimov, A., Patarin, J., Shamir, A.: Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 392–407. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_27
7. Cox, D.A., Little, J., O'Shea, D.: Using Algebraic Geometry, 2nd edn. Springer, New York (2005). https://doi.org/10.1007/b138611
8. Dubois, V., Gama, N.: The degree of regularity of HFE systems. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 557–576. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17373-8_32
9. Faugère, J.-C.: A new efficient algorithm for computing Gröbner bases (F4). J. Pure Appl. Algebra **139**(1–3), 61–88 (1999)
10. Faugère, J.-C.: A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In: Proceedings of ISSAC 2002, pp. 75–83. ACM, New York (2002)
11. Faugère, J.-C., Joux, A.: Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 44–60. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_3

12. Fraenkel, A.S., Yesha, Y.: Complexity of problems in games, graphs and algebraic equations. Discret. Appl. Math. **1**, 15–30 (1979)
13. Fusco, G., Bach, E.: Phase transition of multivariate polynomial systems. Math. Struct. Comput. Sci. **19**(1), 9–23 (2009)
14. Granboulan, L., Joux, A., Stern, J.: Inverting HFE is quasipolynomial. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 345–356. Springer, Heidelberg (2006). https://doi.org/10.1007/11818175_20
15. Impagliazzo, R., Paturi, R.: On the complexity of k-SAT. J. Comput. Syst. Sci. **62**(2), 367–375 (2001)
16. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced Oil and Vinegar signature schemes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 206–222. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_15
17. Lazard, D.: Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations. In: van Hulzen, J.A. (ed.) EUROCAL 1983. LNCS, vol. 162, pp. 146–156. Springer, Heidelberg (1983). https://doi.org/10.1007/3-540-12868-9_99
18. Lokshtanov, D., Paturi, R., Tamaki, S., Williams, R., Yu, H.: Beating brute force for systems of polynomial equations over finite fields. In: 27th ACM-SIAM Symposium on Discrete Algorithms (SODA 2017) (to appear)
19. Montgomery, P.L.: A block Lanczos algorithm for finding dependencies over GF(2). In: Guillou, L.C., Quisquater, J.-J. (eds.) EUROCRYPT 1995. LNCS, vol. 921, pp. 106–120. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-49264-X_9
20. Patarin, J.: Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms. In: Maurer, U. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 33–48. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68339-9_4
21. Thomae, E., Wolf, C.: Solving underdetermined systems of multivariate quadratic equations revisited. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 156–171. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30057-8_10
22. Thomé, E.: Subquadratic computation of vector generating polynomials and improvement of the block Wiedemann algorithm. J. Symb. Comput. **33**(5), 757–775 (2002)
23. Yang, B.-Y., Chen, J.-M.: All in the XL family: theory and practice. In: Park, C., Chee, S. (eds.) ICISC 2004. LNCS, vol. 3506, pp. 67–86. Springer, Heidelberg (2005). https://doi.org/10.1007/11496618_7
24. Yang, B.-Y., Chen, J.-M., Courtois, N.T.: On asymptotic security estimates in XL and Gröbner bases-related algebraic cryptanalysis. In: Lopez, J., Qing, S., Okamoto, E. (eds.) ICICS 2004. LNCS, vol. 3269, pp. 401–413. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30191-2_31
25. Yasuda, T., Dahan, X., Huang, Y.-J., Takagi, T., Sakurai, K.: MQ challenge: hardness evaluation of solving multivariate quadratic problems. In: NIST Workshop on Cybersecurity in a Post-Quantum World (2015). http://eprint.iacr.org/2015/275

Elliptic Curves in Cryptography

Generation and Implementation of Cryptographically Strong Elliptic Curves

Przemysław Dąbrowski, Rafał Gliwa, Janusz Szmidt(✉), and Robert Wicik

Wojskowy Instytut Łączności, Warszawska 22A, 05-150 Zegrze Południowe, Poland
{p.dabrowski,r.gliwa,j.szmidt,r.wicik}@wil.waw.pl

Abstract. Elliptic curves over finite fields are an essential part of public key cryptography. The security of cryptosystems with elliptic curves is based on the computational intractability of the Elliptic Curve Discrete Logarithm Problem (ECDLP). The paper presents requirements which cryptographically secure elliptic curves have to satisfy, together with their justification and some relevant examples of elliptic curves. We implemented modular arithmetic in a finite field, the operations on an elliptic curve and the basic cryptographic protocols.

Keywords: Elliptic curve cryptography · Modular arithmetic
Digital signature ECDSA · Diffie-Hellman key agreement

1 Introduction

Elliptic curves (EC) are widely used in public key cryptography systems, e.g. for key agreement, encryption, digital signatures and pseudo-random generators. As mathematical objects they may form elliptic curve groups with a finite number of elements (points), involving arithmetic operations on those elements. Therefore elliptic curves may be defined among others over real numbers, over the binary field or over the prime field $\mathbb{F}_p$, where p is a prime number. Due to some possible attacks it is safer to use the latter curves and that is why we focus on them. Moreover, prime field curves are very fast on processors, because CPUs usually have an advanced integer multiplier circuit built-in. But the greatest advantage of Elliptic Curve Cryptography (ECC) is the same level of security as in RSA cryptosystems, provided by significantly shorter key sizes. For example, to ensure 128-bit security strength (compared to symmetric cryptography), a 3072-bit RSA public key is necessary, and in the case of ECC only a 256-bit key, see Table 1. So the time spent on performing cryptographic operation decreases.

The U.S. National Institute of Standards and Technology (NIST) has included ECC in the set of recommended algorithms, extending it by Elliptic Curve Diffie-Hellman Algorithm (ECDH) for key exchange and Elliptic Curve Digital Signature Algorithm (ECDSA) for digital signature. The U.S. National Security Agency (NSA) allows their use to protect information classified up to secret. The ECDSA and ECDH are also recommended by NATO as interoperability standards [7]. However, in national systems, specific elliptic curves

J. Kaczorowski et al. (Eds.): NuTMiC 2017, LNCS 10737, pp. 25–36, 2018.
https://doi.org/10.1007/978-3-319-76620-1_2

along with independently designed implementation of elliptic curve arithmetic are required.

This paper presents five required criteria for an elliptic curve in order to be considered cryptographically secure. We put together the security conditions of the Brainpool Standard [3] and twist security requirements [1,2]. We give some examples of curves over finite prime fields $\mathbb{F}_p$ that satisfy these criteria.

As concerns arithmetic operations we present results on implementation of modular arithmetic in the finite prime field $\mathbb{F}_p$ and operations in the finite group $E(\mathbb{F}_p)$ of points on an elliptic curve. Basic arithmetic operations in the finite field are addition and multiplication modulo p. The most time-consuming operation is exponentiation. Effective implementation of modular arithmetic requires special algorithms for modular reduction, because many processors do not have fast division. We implement and compare three algorithms of multiplication with modular reduction: classical, Barrett and Montgomery methods. Then we use modular arithmetic to implement operations on elliptic curves.

Point addition is the basic operation on an elliptic curve. Points can be represented in affine or projective coordinates. Addition of points in affine coordinates needs exponentiation. We can avoid this by performing addition using projective coordinates. ECDSA and ECDH use point multiplication implemented as repeated additions (and doublings). We will compare effectiveness of point multiplication for two types of point representation and three methods of modular multiplication.

Table 1. Number of bits of cryptographic keys

Symmetric algorithms	RSA	Elliptic curves
80	1 024	160
128	3 072	256
256	15 360	512

2 Group of Points on an Elliptic Curve

For basic facts about finite fields and elliptic curves see e.g., [5,6]. Let $p > 3$ be a prime number and let $\mathbb{F}_p$ denote the finite field of p elements

$$\mathbb{F}_p = \{0, 1, \ldots, p-1\},$$

with addition and multiplication modulo p. An elliptic curve over the field $\mathbb{F}_p$ is the set of solutions of an equation

$$E : y^2 = x^3 + Ax + B \bmod p \tag{1}$$

together with "a point at infinity" O, where the coefficients $A, B \in \mathbb{F}_p$ satisfy

$$\varDelta = 4A^3 + 27B^2 \neq 0 \bmod p. \tag{2}$$

The Eq. (1) is called the Short Weierstrass Form. This set forms an abelian group with neutral element O and the addition law, called point addition, given for example in [6]. Points on the curve are given in terms of their x and y coordinates (x, y). The group $E(\mathbb{F}_p)$ of points of the elliptic curve (1) defined over the finite field $\mathbb{F}_p$ has order $\#E(\mathbb{F}_p)$, which satisfies the Hasse inequality

$$p + 1 - 2\sqrt{p} < \#E(\mathbb{F}_p) < p + 1 + 2\sqrt{p}.$$

The exact value of $\#E(\mathbb{F}_p)$ can be calculated using the SEA–algorithm, whose optimized implementation is available in Magma [10]. A special kind of point addition is adding a point to itself, which is called point doubling. Point multiplication is an operation of taking a point on the curve and multiplying it by a natural number. In practice it means a sequence of addition and doubling operations.

Elliptic curve cryptosystems draw their strength from intractability of the Elliptic Curve Discrete Logarithm Problem (ECDLP), defined as follows: given an elliptic curve E defined over $\mathbb{F}_p$ and two points $P, Q \in E(\mathbb{F}_p)$, find an integer d such that $Q = dP$. There is no efficient algorithm to solve ECDLP (for sufficiently large prime p) and it is more difficult than the general DLP for the same length. In Elliptic Curve Cryptography (ECC) for a given elliptic curve the point Q is the public key, and the number d is the private key.

3 Generation of Elliptic Curves

We consider elliptic curves over finite fields $\mathbb{F}_p$, where p is a prime number of suitable size in bits. There exist a lot of standard elliptic curves, given by many official bodies, for use in cryptographic applications [3,7,11,14]. The definition of the curve includes a set of parameters, called domain parameters, which must be shared by two parties to use the ECC algorithm. These are (p, A, B, P_0, n, h), where:

- p is a prime number, which defines the field $\mathbb{F}_p$,
- A and B are integers, coefficients of the short Weierstrass Eq. (1) and satisfying (2),
- $P_0 \in E(\mathbb{F}_p)$ is the base point, called also a generator of a cyclic group,
- n is the order of the cyclic group generated by P_0, that is, the smallest natural number n such that $nP_0 = O$,
- $h = \#E(\mathbb{F}_p)/n$ is called a cofactor; h must be small ($h \leqslant 4$), preferably $h = 1$.

Our purpose is to find customized curves of different sizes, which are cryptographically secure. We consider the bit lengths 160, 256, 384 and 512 of the basic primes p, which were generated according to the algorithm given in [3] from seeds taken from a random number generator. We invoke a search procedure, which selects pseudo-random coefficients A and B. If the coefficients satisfy (2), then the security requirements are checked for the resulting curve. These requirements ensure resistance against known attacks on ECDLP. The elliptic curves are accepted and the search process stops when criteria 1 to 4 below are fulfilled. Then the length of the largest prime factor of the order of the twisted elliptic curve is computed. The following set of security criteria is checked:

1. The group order $\#E(\mathbb{F}_p)$ of the elliptic curve is a prime number ($h = 1$) in order to prevent a small-subgroup attack [3,6] and Pohlig-Hellman attack [12]. Every non-identity point on the curve is a generator of the group of all points of the curve. The curves with prime group order have no points of order 2 ($P + P = O$), and therefore no points with y-coordinate $y = 0$.
2. The order $n = \#E(\mathbb{F}_p)$ is less than the prime number p ($n < p$) [3]. This requirement is necessary to avoid overruns in implementation, since in some cases, even the bit-length of n can exceed the bit-length of p. Elliptic curves with $n = p$ are called trace one curves (or anomalous curves). Satoh and Araki [13] proposed an efficient solution to the ECDLP on trace one curves, so one must exclude such cases.
3. Immunity to attacks using the Weil-pairing or Tate-pairing. These attacks allow the embedding of the group $E(\mathbb{F}_p)$ into the group of units of an extension field $\mathrm{GF}(p^l)$ of degree l of the field $\mathbb{F}_p$. For such cases subexponential attacks on DLP exist. Let $l = \min(t\colon n \mid p^t - 1)$, i.e. l is the order of p modulo n. By Fermat's little theorem, l divides $n - 1$. One can compute the exact value of l by factoring $(n - 1)$. The requirement to avoid the above attack is that $(n - 1)/l \leqslant 100$ [3], which means that l is close to the maximal possible value. Therefore, verification of this condition for a k-bit curve requires the factorization of a number of length up to k bits.
4. The class number of the field $K = \sqrt{-d}$ is greater than 10 000 000 [3]. d is defined as $d = (4p - u^2)/v^2$, where $u = n - p - 1$, and $v = \max\{a\colon a^2 \mid 4p - u^2\}$, so d is the square-free part of $4p - u^2$. This condition protects against attacks exploiting a small value of the class number. According to Gerhard Frey [4] this criterion is intended to protect against possible liftings to curves with CM (Complex Multiplication) over number fields, and in the background there are duality theorems and relations to Brauer groups which could be a danger for the discrete logarithm. The papers of Huang and Raskind, e.g. [8], could go into this direction. Hence one should avoid elliptic curves whose ring of endomorphisms has a class group of small order, and since in most cases this is very mild and easily satisfied condition it may not hurt to have it. Further constraints may be deducted from the paper [9].
5. The twist criterion. By the Hasse-Weil theorem the elliptic curve E has $p+1+t$ points defined over $\mathbb{F}_p$, where $|t| < 2\sqrt{p}$. The twisted curve E^{tw} then has $p + 1 - t$ points defined over $\mathbb{F}_p$. This implies that the order of the group of the twisted elliptic curve is given by the formula

   $$\#E^{tw}(\mathbb{F}_p) = 2p + 2 - \#E(\mathbb{F}_p).$$

   Practically, the largest prime factor of $\#E^{tw}(\mathbb{F}_p)$ has to be longer than 100 bits. Active attacks on elliptic curves which do not satisfy the twist criterion are described in [1,2].

We provide below domain parameters for sample 160, 256, 384 and 512-bit curves. We give detailed results regarding the twist criterion, including the order of the twisted curve $\#E^{tw}(\mathbb{F}_p)$, its prime factors (factorization) and the length of the largest one of them (len). As can be seen, not all of them satisfy the twist criterion. Tables 6, 7, 8 and 9 present curves which satisfy the first four security criteria.

4 Arithmetic Implementation of Group Operations on Elliptic Curves

Execution of cryptographic algorithms and protocols defined using elliptic curves, like ECDSA or ECDH, requires implementation of operations on elliptic curves in the group of points on the curve. For curves defined over finite prime fields first we should implement modular arithmetic, then group operations, among which point addition is fundamental. In this section we describe software implementation dedicated for 32-bit processors.

4.1 Modular Arithmetic

Suppose the curve (1) is defined over the finite field $\mathbb{F}_p$ with prime characteristic $p > 3$. The prime number and the integers of the finite field can be represented in computer memory by sequences of bits or words. So, positive integers $0 \leqslant x < p$ and the prime number p can be written in binary notation:

$$x = (b_{i-1}, b_{i-2}, \ldots, b_1, b_0)_2$$

or

$$x = (w_{j-1}, w_{j-2}, \ldots, w_1, w_0)_w,$$

where $b_0, b_1, \ldots, b_{i-1}$ represent bits from $\{0, 1\}$ and $w_0, w_1, \ldots, w_{j-1}$ words from the set $\{0, 1, \ldots, w-1\}$.

Aiming at effective implementation, the radix (base) w should be chosen close to the word size of the processor, for example $w = 2^{32}$. Then a 512-bit integer is represented by 16 32-bit computer words. Intermediate results, before reduction modulo p, may be larger – the sum of two 512-bit integers has 17 32-bit words, and their product has 32 32-bit words. Therefore in our modular arithmetic dedicated for elliptic curves defined for max 512-bit p integers we use 35 32-bit words. We have implemented basic arithmetic operations in the finite field:

- addition (mod p),
- multiplication (mod p).

Moreover we have implemented additional arithmetic operations:

- subtraction (mod p),
- exponentiation (mod p),

- inverse (mod p),
- division,
- square root,

and other types of non-arithmetic operations:

- bitwise logic operations: OR, XOR, AND, NOT,
- shifts and rotations,
- comparisons,
- assign and copy values,
- pseudo random generator,
- primality test.

Effective implementation of modular arithmetic requires special algorithms for multiplication with modular reduction, because many processors do not have fast division. We have implemented three algorithms for modular multiplication:

- classical [17],
- proposed by Montgomery [16],
- with reduction proposed by Barrett [15].

The classical algorithm calculates the remainder of division by any number p. The Montgomery and Barrett algorithms calculate reduction modulo p without performing classical division. Montgomery modular multiplication and multiplication with Barrett reduction require precomputations.

Before performing calculations using the Montgomery algorithm, we have to compute

$$\gamma = -p^{-1} \bmod w, \tag{3}$$

where w is a base of integer representation which is relatively prime to p.

Before performing calculations using the Barrett algorithm, we have to compute

$$\mu = \frac{2^{2i}}{p}, \tag{4}$$

where i is the number of bits in the binary representation of p.

A basic arithmetic operation used in public key cryptography algorithms and protocols like RSA, DSA and DH is modular exponentiation. We have checked the execution time of this operation with the three above listed algorithms of modular multiplication for 1024 and 2048-bit integers: the base, the exponent and the modulus. The average execution times of modular exponentiation implemented in C, performed on a personal computer with a 3.6 GHz processor are presented in Table 2.

We have also tested the execution times of modular exponentiation implemented in C and applied in embedded Linux, performed on a development board with a 400 MHz ARM processor. The average results are presented in Table 3.

We can see that exponentiation with the Montgomery method is the fastest. These results cover exponentiation with precomputations (3) and (4).

Table 2. Execution times of modular exponentiation on PC

No. of bits of integers	The method of modular multiplication		
	classical	Barrett	Montgomery
1024	38 ms	30 ms	13 ms
2048	288 ms	225 ms	100 ms

Table 3. Execution times of modular exponentiation on ARM

No. of bits of integers	The method of modular multiplication		
	classical	Barrett	Montgomery
1024	250 ms	166 ms	103 ms
2048	1 762 ms	1 243 ms	787 ms

4.2 Arithmetic for Elliptic Curves over a Finite Prime Field

When using elliptic curve cryptography we perform operations on points (x, y) on the curve (1). The basic operation is point addition. The affine coordinates $R = (x_2, y_2)$ of the sum of $P = (x_0, y_0)$ and $Q = (x_1, y_1)$ are calculated modulo p using the formulas

$$x_2 = \lambda^2 - x_0 - x_1, \qquad y_2 = \lambda(x_0 - x_2) - y_0,$$

where

$$\lambda = \frac{y_0 - y_1}{x_0 - x_1} \qquad \text{for } (x_0, y_0) \neq (x_1, y_1), \tag{5}$$

$$\lambda = \frac{3x_0^2 + A}{2y_0} \qquad \text{for } (x_0, y_0) = (x_1, y_1). \tag{6}$$

Addition of a point to itself is called point doubling. Note that the sum of (x_0, y_0) and $(x_0, -y_0)$ is the neutral element in the group (the point at infinity O).

Calculating λ when adding different (5) points and when point doubling (6) needs division in the finite prime field, which can be calculated via inversion, $v^{-1} = v^{p-2} \bmod p$. Unfortunately, this requires exponentiation in the prime field.

The ECDSA and ECDH algorithms use as a basic operation multiplication of a point on the curve E by a number d, where $1 < d < p$. The multiplication algorithm requires k point doublings and on average $k/2$ point additions, where k is the number of bits in d. Additionally, ECDSA and ECDH require procedures for point generation and verification to check that the equation of the curve is satisfiesd.

Thus, our elliptic curve arithmetic consists of the following operations on the group of points:

- generation,
- addition,

- doubling,
- multiplication,
- verification.

These operations have been implemented in affine and projective coordinates. We can easily convert affine coordinates (x, y) of a point on $E : y^2 = x^3+Ax+B$ into projective coordinates (x, y, z) on an equivalent curve $E' : zy^2 = x^3+Axz^2+Bz^3$ and invert this by taking $z = 1$. Formulas for point addition and doubling in projective coordinates do not use divisions (5) and (6), computed via time-consuming exponentiation. Inverting calculations from projective coordinates to affine coordinates require divisions x/z and y/z in the finite prime field.

We present execution times of point multiplication for 256, 384 and 512-bit integers: EC parameters A and B, coordinates of points x, y, z and the modulus p. We use affine and projective coordinates and three multiplication algorithms with reduction. We apply Montgomery modular multiplication in affine coordinates which needs fast modular exponentiation. Projective coordinates need just simple modular operations: multiplication, squaring, adding and subtraction to carry out point addition and doubling, so we apply multiplication and other operations with classical and Barrett modular reductions. Average execution times of point multiplication implemented in C, performed on a personal computer with a 3.6 GHz processor, are presented in Table 4.

We can see that point multiplication is the fastest in projective coordinates and with Barrett reduction. These results covers calculations with precomputations (3) and (4).

We have also tested the execution times of point multiplication implemented in C and applied in embedded Linux, performed on a development board with a 400 MHz ARM processor. The average results are presented in Table 5.

Table 4. Execution times of point multiplication on PC

No. of bits of integers	The method of point representation and the method of modular multiplication		
	Affine Montgomery	Projective classical	Projective Barrett
256	84 ms	8 ms	7 ms
384	378 ms	27 ms	24 ms
512	1 270 ms	64 ms	55 ms

Point multiplication is the basic operation performed during key pair generation and then during signing and verification signature using ECDSA. Key generation and signing require one point multiplication and signature verification, that is, two point multiplications. So, the time of key generation and signing for a 384-bit p is 24 ms on PC with a 3.6 GHz processor, and about 557 ms on a platform with a 400 MHz ARM processor with our fastest implementation of EC arithmetic. Signature verification takes 48 ms and 1.1 s respectively.

The main operation performed during key agreement according to the ECDH protocol is point multiplication as well. Each party calculates one point multiplication for their own key pair generation and then one point multiplication for each key agreement. So each operation takes, depending on the platform, respectively 24 ms and 557 ms.

Execution times presented in Tables 2, 3, 4 and 5 can be reduced if special parameters of the elliptic curve are used. We will continue our work to optimize the code of our modular and EC arithmetic. In the next stage we will implement modular arithmetic and arithmetic on elliptic curves in FPGA hardware, which is required for the higher levels of security.

Table 5. Execution times of point multiplication on ARM

No. of bits of integers	The method of point representation and the method of modular multiplication		
	Affine Montgomery	Projective classical	Projective Barrett
256	2 798 ms	242 ms	181 ms
384	12 881 ms	718 ms	557 ms
512	41 647 ms	1 623 ms	1 261 ms

5 Summary

Elliptic Curve Cryptography is the most efficient way to achieve secure key establishment and digital signature. The basis for implementation of any ECC cryptosystem is a secure elliptic curve, defined by its coefficients, the order and the base point. Although there are many recommended domain parameter sets, it is especially important to be able to generate and to implement unique, customized elliptic curves, satisfying required conditions. This will allow the development and efficient implementation of national systems in the field of public key cryptography. In this paper we have identified five main criteria that must be met by curves to be considered cryptographically secure. According to these criteria we showed that finding secure elliptic curves with lengths from 160 to 512 bits is feasible. Adding the twist criterion limits the number of secure elliptic curves. After choosing the parameter set domain, we implemented the finite prime field arithmetic and elliptic curve arithmetic.

Implementation of arithmetic in the group of points on an elliptic curve requires fast modular operations, suitable methods of modular reduction and suitable representation of point coordinates. We implemented modular arithmetic for 32-bit processors, including three methods of modular multiplication: classical, Montgomery and with Barrett reduction, and two types of coordinate representation: affine and projective. Having all these possibilities available, we compared execution times of the basic modular operation, the integer exponentiation, and the basic elliptic curve operation – point multiplication, in various configurations. In the case of modular exponentiation, the best result was achieved by Montgomery multiplication. In the case of point multiplication the best result was achieved by projective coordinates and Barrett reduction.

In the next step we plan to implement modular arithmetic and elliptic curve arithmetic in field programmable gate arrays (FPGA). Arithmetic in groups of points on elliptic curves is intended to be used for implementation of ECC algorithms and protocols for digital signature (ECDSA) and key agreement (ECDH), which will replace previously used ones.

Appendix: The Examples of Elliptic Curves

Table 6. Domain parameters for a 160-bit elliptic curve

p	`0xE75F077B3804BAB2C122344DFD04FCE951DA7027`
A	`0xE06C22F8F36E2468E2B5F27CCBD57D9DC6B23400`
B	`0x68F4C31B7CE82460D372864AB2C8C1CCE5F29283`
P_0	$x(P_0) =$ `0x37A8D5420536D5F3071C706D66A5CE4C07C700D9`
	$y(P_0) =$ `0x6BEF365071D253DEA39FC3088E3C0CCC6FF47F09`
n	`0xE75F077B3804BAB2C121924451CFCFFBABE5FBB3`
h	1
$\#E^{tw}(\mathbb{F}_p)$	`0xE75F077B3804BAB2C122D657A83A29D6F7CEE49D`
prime factors	`[<3,1>, <90281,1>, <426611,1>, <11431915082479720108730664390719752669, 1>]`
len	124

Table 7. Domain parameters for a 256-bit elliptic curve

p	`0xA4701F69D1D96BCEE3719029B6C8F3F1C0318B00FBC76A4FBAE54A2D84BA90C3`
A	`0x1C417D163830A291B2F769BE7737E29112C4D400ECC3A22726E589289084DC67`
B	`0x9D3B1FE4E68A23711EC1D7D92251D14C0CE040CB21EF11DA66012DDD79402E72`
P_0	$x(P_0) =$ `0x7EE932CFAA5B1EFE0297815BF0036DFFB4C9B70708B344481504C36D4C24BEB9`
	$y(P_0) =$ `0x5195AC4DA0186C7B3FBDF20AF09F64276EE25C689ACDF8174E2D4BD8BFC50D25`
n	`0xA4701F69D1D96BCEE3719029B6C8F3F0485F90CF0AC53F6344F9D95622C730AD`
h	1
$\#E^{tw}(\mathbb{F}_p)$	`0xA4701F69D1D96BCEE3719029B6C8F3F338038532ECC9953C30D0BB04E6ADF0DB`
prime factors	`[<137,1>, <77127527346223,1>, <7039001868391305024827233109584083029781968094947827782607661, 1>]`
len	203

Table 8. Domain parameters for a 384-bit elliptic curve

p	0x950FD23F7FCDB5D647C6087B67A238B8C94A33898021E71451B5F922A277D40F89C561387B978CC057749BE485C3621F
A	0x2704ED36195B700E6DA4A3B98DEF52342094C6AA34A71A36F64D0F3E2A38432D1E7C85004583CC3246254258B392508D
B	0x7C1CA2774E5FABC0EB668323DC507E2BF0FD936BFBAFAEABED0E1F5740D196276C5A9EE60D40957F67E6333320359295
P_0	$x(P_0) =$ 0x1C56954F12FC79768A87CAC920323115E50B1DA42542A380E1265779A32A2D23F9BEE6FCA61BCB057AFB26ECA927E51C $y(P_0) =$ 0x8A28CF52E6B00BF935667D90092EA01504133AA556C23C9462AF727DF244464D6B575F1B61C3FBA27E242ABBCE28121B
n	0x950FD23F7FCDB5D647C6087B67A238B8C94A33898021E713C6FEECAB3B86DB8D79D79B916E3E2F199A1C8098D2C8035D
h	1
$\#E^{tw}(\mathbb{F}_p)$	0x950FD23F7FCDB5D647C6087B67A238B8C94A33898021E714DC6D059A0968CC9199B326DF88F0EA6714CCB73038BEC0E3
prime factors	[<5,1>, <4588542219799413391900517767130438062170403601256391422007358414020609723036030758374529156592774903913259915106759, 1>]
len	381

Table 9. Domain parameters for a 512-bit elliptic curve

p	0xE121B140806D878B50656F5A5AEF0FBE3A912FD8526A10EB6177EC6C4FF5808C2F6812C529097FCA07F5F7D57B1F1E7FEB41CA7FEDF8C647CD5FD40DB53EC107
A	0x8A3DFCDF063ABB966D72DB6C328346B937D6BE075049D765474730D8A13415D550ABB77C00343AD0C0B03D784F4F4EC5158BC43DC5C2AC33C6620031510FD69
B	0x8B3423FE32644E29860D667CDFA9CB62376F32A9404D470EAF9BE87E35144F120B5E9238402DA3105D09B096C52081ECBAD2D687EC3D42C07274812E10414827
P_0	$x(P_0) =$ 0xAE3D9F24D26B6B5E944EA30DA95AF78FC922E00BA6052B0FD8C1605B540A33C7BFA7EC6106A13A52661ABE77E4C6C02154AD7FC97FD68E352E67A38DAF1DFE9E $y(P_0) =$ 0xBF95072569330E0088BC5C82DEE33543AED1EB3090618B15873A1617F4F4D0FD9D22394AD380EA96AE478FAEC4D9A8693B60EBEA58983676277AC5ECF1C6B085
n	0xE121B140806D878B50656F5A5AEF0FBE3A912FD8526A10EB6177EC6C4FF5808A7A65BC602CD79AA4EF4B9B598746662B67160EC7634F74B183CB8409B8F2856F
h	1
$\#E^{tw}(\mathbb{F}_p)$	0xE121B140806D878B50656F5A5AEF0FBE3A912FD8526A10EB6177EC6C4FF5808DE46A692A253B64EF20A054516EF7D6D46F6D863878A217DE16F42411B18AFCA1
prime factors	[<17,1>, <263, 1>, <73777333199494915133765584973, 1>,<35745935778927025970192786500550827320017892278246393359096806176990291228433366122561111388192728299916229683860346449603, 1>]
len	404

References

1. Bernstein, D.J., Lange, T.: SafeCurves: choosing safe curves for elliptic-curve cryptography. http://safecurves.cr.yp.to
2. Bernstein, D.J., Chou, T., Chuengsatiansup, Ch., Huelsing, A., Lange, T., Niederhagen, R., Van Vredendaal, Ch.: How to manipulate curve standards: a white paper for the black hat. In: Cryptology ePrint Archive, 2014/571 (2014). www.iacr.org
3. ECC Brainpool: ECC Brainpool Standard Curves and Curve generation (2005). www.ecc-brainpool.org/download/Domain-parameters.pdf
4. Frey, G.: Private Communication (2017)
5. Gawinecki, J., Szmidt, J.: Zastosowanie ciał skończonych i krzywych eliptycznych w kryptografii (Applications of finite fields and elliptic curves in cryptography). Wojskowa Akademia Techniczna, Warszawa (1999)
6. Hankerson, D., Menezes, A., Vanstone, S.: Guide to Elliptic Curve Cryptography. Springer, New York (2004). https://doi.org/10.1007/b97644. ISBN 0-387-95273-X
7. INFOSEC Technical and Implementation Directive on Cryptographic Security and Cryptographic Mechanisms, AC/322-D/0047-REV2, 11 March 2009
8. Huang, M.-D., Raskind, W.: Signature calculus and discrete logarithm problems. In: Hess, F., Pauli, S., Pohst, M. (eds.) ANTS 2006. LNCS, vol. 4076, pp. 558–572. Springer, Heidelberg (2006). https://doi.org/10.1007/11792086_39
9. Jao, D., Miller, S.D., Venkatesen, R.: Ramanujan graphs and the random reducibility of discrete log on isogenous elliptic curves (2004). www.iacr.org
10. Magma Computational Algebra System. www.magma.math.usyd.edu.au
11. NIST: Recommended Elliptic Curves for Federal Government Use (1999)
12. Pohlig, S., Hellman, M.: An improved algorithm for computing logarithms over GF(p) and its cryptographic significance. IEEE Trans. Inf. Theory **24**, 106–110 (1978)
13. Satoh, T., Araki, K.: Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves. Comm. Math. Univ. Sancti Pauli **47**, 81–92 (1998)
14. SEC2: Recommended Elliptic Curve Domain Parameters. Certicom Research, 27 January (2010). Version 2.0
15. Barrett, P.: Implementing the rivest Shamir and Adleman public key encryption algorithm on a standard digital signal processor. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 311–323. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-47721-7_24
16. Montgomery, P.: Modular multiplication without trial division. Math. Comput. **44**, 519–521 (1985)
17. Menezes, A., van Oorschot, P., Vanstone, S.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1996)

On the Possibility of Transformation of Multidimensional ECDLP into 1-Dimensional ECDLP

Michał Wroński(✉) and Tomasz Kijko(✉)

Faculty of Cybernetics, Institute of Mathematics and Cryptology,
Military University of Technology in Warsaw, Kaliskiego 2, 00-908 Warsaw, Poland
{michal.wronski,tomasz.kijko}@wat.edu.pl

Abstract. In this article the attack on elliptic curve discrete logarithm problem (ECDLP) with partial information is considered. If unknown bits of discrete logarithm are continuous then 1-dimensional algorithms for ECDLP may be used. One of these algorithms is improved Gaudry-Schost using equivalence classes which requires $O(1.47\sqrt{n})$ operations. It will be showed that if unknown bits are not continuous and are given in $c > 1$ partitions and also two most significant bits are known, transformation of this partitions into one partition to use 1-dimensional algorithm without increasing size of the problem is impossible. It is also showed that in some situations it is better to "forget" some of known bits to transform the problem to 1-dimensional ECDLP.

Keywords: Elliptic curve · ECDLP · Partial information
Multidimensional Gaudry-Schost algorithm

1 Introduction

Many cryptographic algorithms base on elliptic curve discrete logarithm problem (ECDLP). If point (generator) P on elliptic curve is given and it is required to compute $Q = [K]P$, then computations are very fast. If Q and P are known but K is unknown, then it is computationally hard to find K. If some information about K is given (values of some bits of K are known), then it is easier to find all bits of K. It should be noted that electronic devices are not resistant for side channel attacks. Using attacks like power analysis [4] or acoustic cryptanalysis [2], attacker is able to find some information about K (in perfect situation it is possible to find out all bits of K).

More realistic situation is that some bits will be known with almost 100% probability and the others with smaller probability. Such cases were considered for example in work [5]. In this article will be assumed that bits are known with 100% probability or are unknown (there is probability 50% that given bit is equal to 0 or 1).

J. Kaczorowski et al. (Eds.): NuTMiC 2017, LNCS 10737, pp. 37–47, 2018.
https://doi.org/10.1007/978-3-319-76620-1_3

2 Attack with Partial Information

Below are described different cases of attacks with partial information.

2.1 Only Least Significant Bits are Unknown

Let's consider the situation that some the least significant continuous bits are known:

$$K = \underbrace{known\ bits}_{m}\,\underbrace{unknown\ bits}_{l}.$$

It is the simplest situation. All unknown bits of K are continuous and given in set $\{U, U+1, \ldots, U+2^l-1\}$, where

$$U = \underbrace{known\ bits}_{m}\,\underbrace{00\ldots00}_{l}.$$

In this case for example Pollard's lambda [6] or 1-dimensional Gaudry-Schost algorithm [1] may be applied.

2.2 Only Some Continuous Bits are Unknown

This situation may be illustrated as below:

$$K = \underbrace{known\ bits}_{m_2}\,\underbrace{unknown\ bits}_{l}\,\underbrace{known\ bits}_{m_1}.$$

This problem may be solved as previous. Let's see that it is possible to find new generator $P' = [2^{m_1}]P$ and then search for solution in interval of size 2^l. Let

$$U = \underbrace{known\ bits}_{m_2}\,\underbrace{00\ldots00}_{l}\,\underbrace{known\ bits}_{m_1}.$$

Then $Q = [U]P + [K']P'$, where $K' \in \{0, \ldots, 2^l-1\}$. There are required some transformations to get 1-dimensional ECDLP.

If $P' = [2^{m_1}]P$, then $P = [2^{-m_1}]P'$. Because it is assumed that $Ord(P) = p$ is prime, then the element 2^{m_1} is invertible modulo p and $Q = [U \cdot 2^{-m_1}]P' + [K']P'$. In such case it is possible to search for solution of K' in set

$$\{U \cdot 2^{-m_1}, U \cdot 2^{-m_1} + 1, \ldots, U \cdot 2^{-m_1} + 2^l - 1\}.$$

If K' is found then it is easy to find $K = U + K' \cdot 2^{m_1}$.

2.3 Only Some Continuous Bits in the Middle are Known. Most and Least Significant Bits are Unknown

In [3] is described method of searching for solution if the most and the least significant bits of K are unknown. Let's consider the situation where:

$$K = \underbrace{unknown\ bits}_{l_2}\ \underbrace{known\ bits}_{m_1}\ \underbrace{unknown\ bits}_{l_1}.$$

Then K may be presented as:

$$K = d_2 2^{m_1+l_1} + K_1 2^{l_1} + d_1,$$

where $d_1 \in \{0, \ldots, 2^{l_1} - 1\}$, $d_2 \in \{0, \ldots, 2^{l_2} - 1\}$ and

$$K_1 = \underbrace{0 \ldots 0}_{l_2}\ \underbrace{known\ bits}_{m_1}\ \underbrace{0 \ldots 0}_{l_1}.$$

Numbers d_1 and d_2 are unknown. Let's assume that number R from set $\{1, \ldots, p-1\}$ for which $R \cdot 2^{m_1+l_1} = fp + s$, where $|s| < \frac{p}{2}$ is given. Let's see that:

$$\begin{aligned} RK &= Rd_2 2^{m_1+l_1} + RK_1 2^{l_1} + Rd_1 = (fp+s)d_2 + RK_1 2^{l_1} + Rd_1 \\ &= d_2 fp + sd_2 + RK_1 2^{l_1} + Rd_1 = d_2 fp + RK_1 2^{l_1} + d', \end{aligned}$$

where $d' = sd_2 + Rd_1$.
Now it is easy to see that:

$$[RK]P = [R]([K]P) = [R]\overline{P} = [d_2 fp + RK_1 2^{l_1} + d']P = [RK_1 2^{l_1} + d']P,$$

where $\overline{P} = [K]P$.

Because K_1 is known, then:

$$\begin{aligned} P' &= [R]\overline{P} - [RK_1 2^{l_1}]P = [R](\overline{P} - [K_1 2^{l_1}] = [d']P, \\ [R]Q &= [RK]P = [RK_1 2^{l_1} + d']P \end{aligned}$$

and

$$Q' = [R]Q - [RK_1 2^{l_1}]P = [R](Q - [K_1 2^{l_1}]P) = [d']P.$$

Finally it is required only to find the value d'.

If s is positive, then d' must be in the set

$$\{0, \ldots, R \cdot (2^{l_1} - 1) + s \cdot (\frac{p}{2^{m_1+l_1}} - 1)\}.$$

In this case it is possible to use 1-dimensional Pollard's lambda or Gaudry-Schost algorithm.

If s is negative, then d' must be in the set

$$\{s \cdot (\frac{p}{2^{m_1+l_1}} - 1), \ldots, R \cdot (2^{l_1} - 1)\}$$

and the same methods may be used. The size of both intervals is the same and is $R2^{l_1} + |s|\frac{p}{2^{m_1+l_1}}$ at most. To minimize the complexity of computations for intervals, it is required to find such R that the value $R2^{l_1} + |s|\frac{p}{2^{m_1+l_1}}$ is as small as possible.

It should be also noted that there must exist $s \equiv R2^{m_1+l_1}(mod\ p)$. It is not always possible to find such R that the interval in which searching is made would have length $\frac{p}{2^{m_1}} \approx 2^{n-m_1}$. The best case is when p is Mersenne prime number $p = 2^n - 1$, then for $R = 2^{n-m_1-l_1}$ and for $s = 1$ interval has bitlength $l_1 + l_2 = n - m_1$. There are methods of choosing R to minimize the length of interval for other primes, but such interval will be always bigger than 2^{n-m_1} and will be about size $\frac{\sqrt{2p}}{2^{\frac{m_1}{4}}}$. These methods are described with details in [3].

2.4 There are Many Unknown Bits Given in Many Disjoint Intervals

Now let's consider the situation where:

$$K = \underbrace{known\ bits}_{m_{c+1}}\ \underbrace{unknown\ bits}_{l_c}\ \underbrace{\ldots\ldots}_{\ldots}\ \underbrace{known\ bits}_{m_3}\ \underbrace{unknown\ bits}_{l_2}$$
$$\underbrace{known\ bits}_{m_2}\ \underbrace{unknown\ bits}_{l_1}\ \underbrace{known\ bits}_{m_1}.$$

It will be also assumed that at least 2 most significant bits of K are known $(m_{c+1} \geq 2)$.

Let's assume that sum of unknown bits is equal to $k = \sum_{j=1}^{c} l_j$, and c is number of disjoint intervals in which bits of K are unknown.

Let's suppose that P is point on elliptic curve (generator) and $Ord(P) = p$. There is also given point Q for which $Q = [K]P$, where the value K is sought. We are searching for K. Now let's assume that:

$$U = \underbrace{known\ bits}_{m_{c+1}}\ \underbrace{0\ldots0}_{l_c}\ \underbrace{\ldots\ldots}_{\ldots}\ \underbrace{known\ bits}_{m_3}\ \underbrace{0\ldots0}_{l_2}\ \underbrace{known\ bits}_{m_2}\ \underbrace{0\ldots0}_{l_1}\ \underbrace{known\ bits}_{m_1}.$$

Then

$$Q = [K]P = [U]P + [\sum_{i=1}^{c} d_i 2^{a_i}]P,$$

where $d_i 2^{a_i}$ is generator of interval i. So $a_i = \sum_{j=1}^{i-1} l_j + \sum_{j=1}^{i} m_j$ and $d_i \in \{0, \ldots, 2^{l_i} - 1\}$ for $i \in \{1, \ldots, c\}$, because i-th interval has bitlength of l_i bits. Then:

$$Q' = Q - [U]P = [K]P - [U]P = [K - U]P = [\sum_{i=1}^{c} d_i 2^{a_i}]P = [K']P',$$

where $K' \in \{0, \ldots, 2^k - 1\}$. In such case the inequality $\sum_{i=1}^{c} d_i 2^{a_i} < \frac{p}{2}$ also holds (that is why it is assumed that at least two most significant bits of K are unknown).

Theorem 1. *The transformation of the problem described above into 1-dimensional ECDLP of size $2^{l_1+l_2+\ldots+l_c} = 2^k$ is impossible.*

Proof (Theorem 1)

Let's suppose that transformation described in Sect. 2.4 to get one interval of length 2^k is possible. To get 1-dimensional ECDLP the result must be generated by one generator $P' = [t]P, t \in F_p^*$. The interval may begin from value v. Then should hold $[\sum_{i=1}^{c} d_i 2^{a_i}]P = [v]P' + [s]P' = [v+s]P'$, where for every possible values of d_i the value of s would be in interval $s \in \{0, \ldots, 2^k - 1\}$ (because it is required to search for solution in one interval).

But $P' = [t]P$ and then:

$$[\sum_{i=1}^{c} d_i 2^{a_i}]P = [K']P' = [v]P' + [s]P' = [(v+s)t]P.$$

Now it is easy to see that:

$$\sum_{i=1}^{c} d_i 2^{a_i} = (v+s)t$$

and because $t \neq 0$ then also

$$\sum_{i=1}^{c} d_i 2^{a_i} t^{-1} = v + s.$$

If $d_0 = d_1 = \ldots = d_c = 0$ there must exist $s_0 \in \{0, \ldots, 2^k - 1\}$ for which $(v + s_0)t = 0$. Because $t \neq 0$, then $v + s_0 = 0$, so $v = -s_0$. But if $s_0 \in \{0, \ldots, 2^k - 1\}$ then $v \in \{-2^k + 1, \ldots, 0\}$.

To finish the Proof of Theorem 1 it is required to prove the Lemma 1.

Lemma 1. *The only possible values of v are 0 or $-2^k + 1$.*

Proof (Lemma 1)

If $\sum_{i=1}^{c} d_i 2^{a_i} t^{-1} = v + s$, then

$$\sum_{i=1}^{c} (-d_i) 2^{a_i} t^{-1} = -v - s.$$

If $-2^k + 1 < v < 0$, then there exists some $s_f \in \{0, \ldots, 2^k - 1\}$ satisfying:

$$v + 2^k - 1 \geq v + s_f > 0$$

and

$$0 > -v - s_f > v.$$

So both values $[\sum_{i=1}^{c} d_i 2^{a_i} t^{-1}]P$ and $[\sum_{i=1}^{c} (-d_i) 2^{a_i} t^{-1}]P$ must be some multiplicities of P' from set $\{v, \ldots, v + 2^k - 1\}$.

Then

$$v + s_f = \sum_{i=1}^{c} d_i 2^{a_i} t^{-1}$$

is in set $\{v, \ldots, v + 2^k - 1\}$ and also

$$-v - s_f = \sum_{i=1}^{c} (-d_i) 2^{a_i} t^{-1}$$

is in the same set.

Let's see that because the inequality $0 \leq \sum_{i=1}^{c} d_i 2^{a_i} < \frac{p}{2}$ must hold then $\frac{p}{2} < \sum_{i=1}^{c} (-d_i) 2^{a_i} \leq p$ (operations modulo prime p are made).

So if value $(v+s_f)t = \sum_{i=1}^{c} d_i 2^{a_i}$ is in the set $\{0, \ldots, \frac{p}{2}\}$ then $0 > -v-s_f > v$ and

$$(-v - s_f)t = \sum_{i=1}^{c} (-d_i) 2^{a_i}$$

should be in the same set $\{0, \ldots, \frac{p}{2}\}$ what is impossible because $(-v - s_f)t$ is in set $\{p - \frac{p}{2}, \ldots, p-1, 0\}$.

Finally, $v = -2^k + 1$ or $v = 0$. □

Fact 1. *If* $d_0 = d_1 = \ldots = d_c = 0$ *then*

$$\sum_{i=1}^{c} d_i 2^{a_i} = \sum_{i=1}^{c} (-d_i) 2^{a_i} = 0.$$

Fact 2. *The case when* $v = -2^k + 1$ *may be transformed into the case* $v = 0$*: it is sufficient to operate not on* P' *but on* $-P'$*.*

Fact 3. *If it is assumed that* $v = 0$ *then* $K' \in \{0, \ldots, 2^k - 1\}$*.*

Using results from Lemma 1 it is possible to finish Proof of Theorem 1.

Proof (Theorem 1)

Let's see that for every $d_i \in \{0, \ldots, 2^{l_i} - 1\}$ must hold $(\sum_{i=1}^{c} d_i g_i) \leq 2^k - 1$, where $g_i = 2^{a_i} t^{-1}$. If $g_i \leq 2^k - 1$, then $d_i g_i \leq 2^k - 1$. That is because (of course when d_i may be equal to 2, so when $l_i > 1$):

$$2g_i = g_i + g_i \leq 2^k - 1 + 2^k - 1 = 2(2^k - 1) < \frac{p}{2} + \frac{p}{2} = p,$$

so $2g_i < p$ and therefore $2g_i < \frac{p}{2}$ and because these operations are performed in integer ring, not modulo, then $2g_i < 2^k - 1$.

Repeating this step it is easy to observe that of course for every $d_i \in \{0, \ldots, 2^{l_i} - 1\}$ holds $d_i g_i \leq 2^k - 1$ so also $(2^{l_i} - 1) g_i \leq 2^k - 1$ (in integer ring, not modulo p) holds. Because intervals are disjoint, then number $2^{l_i} g_i$ cannot be in set $\{0, \ldots, 2^k - 1\}$. So the following inequality must hold:

$$(2^{l_i} - 1) g_i \leq 2^k - 1 < 2^{l_i} g_i.$$

By analogy:
$\sum_{i=1}^{c} d_i g_i \leq 2^k - 1$ for every $d_i \in \{0, \ldots, 2^{l_i} - 1\}$.

The question is how to find elements $d_i \in \{0, \ldots, 2^{l_i} - 1\}$, $i = \overline{1, c}$, for which $\sum_{i=1}^{c} d_i g_i = 1$.

For some $i \in \{1, \ldots, c\}$ must hold $g_i = 1$, where $d_i = 1$ and for $j \neq i$ $d_j = 0$. Otherwise, for some $\sum_{i=1}^{c} d_i g_i$, where $d_i \in \{0, \ldots, 2^{l_i} - 1\}$, the inequality $\sum_{i=1}^{c} d_i g_i > 2^k - 1$ would hold (in modulo, not integer ring) which cannot occur.

There must also hold $(2^{l_i} - 1)g_i \leq 2^k - 1 < 2^{l_i} g_i$. Because operations are made in integer ring (not modulo p):

$$\frac{2^k - 1}{2^{l_i}} < g_i \leq \frac{2^k - 1}{2^{l_i} - 1},$$

then

$$g_i = 1$$

if and only if

$$\frac{2^k - 1}{2^{l_i} - 1} = 1.$$

So finally $k = l_i$.

It means that to make these transformations there cannot be $c > 1$ disjoint intervals and in consequence it is impossible to transform

$$[\sum_{i=1}^{c} d_i 2^{a_i}]P = [v]P' + [s]P'$$

into 1-dimensional ECDLP. □

3 Results

It was showed that if many disjoint intervals of unknown bits are given and two most significant bits of K are known, then it is impossible to transform the problem from Sect. 2.4 into 1-dimensional ECDLP without increasing size of the problem. It is very important from practical point of view, because multidimensional ECDLP has longer expected time of searching for solution.

The table below shows the expected time of solving ECDLP for different dimensions. The expected time [7] is given by formula $T_{1,n} = O(1.47\sqrt{n})$ in average case for 1-dimensional problem (so $A_1 = 1.47$) and $T_{c,n} = O(A_c\sqrt{n})$ in average case for c dimensional problem, where $c > 1$ and $A_c = (\frac{2}{\sqrt{3}})^c\sqrt{\pi}$ (Table 1).

Fraction $\frac{A_c}{A_1}$ shows how much time is needed in average case to solve problem with the same number of unknown bits in c-dimensional ECDLP comparing to 1-dimensional ECDLP (Table 2).

Table 1. Values of coefficients A_c for different number of dimensions c

c	1	2	3	4	5	6	7	8	9	10
A_c	1.47	2.36	2.73	3.15	3.64	4.20	4.85	5.60	6.47	7.47

Table 2. Comparison of average times required for solving two dimensional ECDLP ($A_{2,k}$) with 1-dimensional ($A_{1,k+m_1}$) for k unknown bits and different m_1. In 1-dimensional ECDLP m_1 bits are "forgotten"

l_1	1	2	3	4	5	6	7	8	9	10
$\frac{A_{2,k}}{A_{1,k+m_1}}$	1.14	0.80	0.57	0.40	0.28	0.20	0.14	0.10	0.07	0.05

3.1 Practical Implementation

In order to practical comparison of 1-dimensional Gaudry-Schost with two-dimensional Gaudry-Schost algorithm authors considered following ECDLP problem: compute discrete logarithm of a point Q in cyclic group generated by the point P on elliptic curve $E/\mathbb{F}_p$, where

1. Elliptic curve $E/\mathbb{F}_p$ is the NIST $P-192$ curve, the point P is the generator of cyclic subgroup of order n.
2. The number of unknown bits of K (where $Q=[K]P$) is k.

The number K has a form

$$K=\underbrace{known\ bits}_{m_2}\underbrace{unknown\ bits}_{l_2}\underbrace{known\ bits}_{m_1}\underbrace{unknown\ bits}_{l_1},$$

where $l_1+l_2=k$. For 1-dimensional Gaudry-Schost algorithm known bits to get one joint interval are "forgotten". Then the number U has a form:

$$K=\underbrace{known\ bits}_{m_2}\underbrace{unknown\ bits}_{l_2+m_1+l_1}.$$

It means that $k'=l_1+m_1+l_2$ bits of the number U must be computed.

For experimental comparison authors considered cases presented in Table 3 (for 2-dimensional ECDLP).

Table 3. Experimental parameters

l_1	l_2	m_1	$k=l_1+l_2$
33	33	1	66
38	28	1	66
43	23	1	66

For 1-dimensional ECDLP k' was equal to 67 (since $m_1 = 1$). In practical implementation authors compared average times of computing ECDLP with:

- 1-dimensional improved Gaudry-Schost algorithm on equivalence classes (parallel version),
- two-dimensional Gaudry-Schost algorithm (parallel version).

For all cases described in Table 3 authors generated randomly 64 discrete logarithms to compute 64 points $Q = [K]P$, where $K = a_2 \cdot 2^{m_1+l_1} + a_1$, $a_2 \in \{0, \ldots, 2^{m_2-1}\}$ and $a_1 \in \{0, \ldots, 2^{m_1-1}\}$. Computations were repeated 7 times (which generates 1344 results for each algorithm). There were obtained average times t_1, t_2 and standard deviations s_1, s_2:

1. for 1-dimensional Gaudry-Schost algorithm:

$$t_1 = 119.24s, \quad s_1 = 55.71s,$$

2. for two-dimensional Gaudry-Schost algorithm:

$$t_2 = 130.76s, \quad s_2 = 58.13s.$$

It is worth to point that experimental results differ from theoretical. Expected value of $A_{2,k}/A_{1,k+1} = 1.14$ is bigger than $t_2/t_1 = 1.10$. The difference between these results may be caused by the value of m_1. When $m_1 = 1$ it is possible that during pseudorandom walks both "tame" and "wild kangaroos" can "jump" as far that they reenter the searching area. This situation is more probable for $m_1 = 1$ than for bigger values of m_1. Average time for multidimensional Gaudry-Schost algorithm does not include the influence of distance between both intervals of unknown bits.

Let's describe two starting areas: **T** for "tame" and **W** for "wild kangaroos". Both situations, for $m_1 = 1$ and $m_1 = 2$, are presented on Figs. 1 and 2.

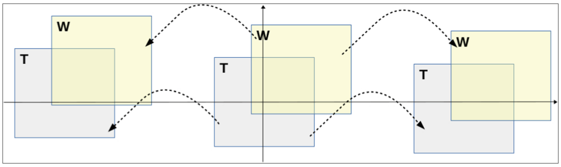


Fig. 1. Possible reentering the searching area for $m_1 = 1$

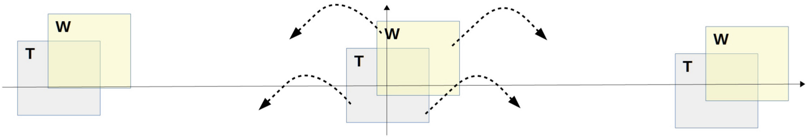


Fig. 2. Possible reentering the searching area for $m_1 = 2$

4 Transformation of c-Dimensional ECDLP into $(c-1)$-Dimensional ECDLP for $c \geq 3$

One can ask if it is computationally efficient to transform c-dimensional ECDLP into $(c-1)$-dimensional ECDLP for $c \geq 3$ by "forgetting" known bits. Let's take a look at analytic comparison of average time required to solve c-dimensional ECDLP. If the number of unknown bits is equal to k, then $N = 2^k$ and thus $T_{c,2^k} = O((\frac{2}{\sqrt{3}})^c\sqrt{\pi 2^k})$. Let's now assume that there is k unknown bits given in c disjoint intervals and some value of known bits $m_2, m_3, \ldots, m_c$ is equal to one, so we have similar situation as in Sect. 2.4.

The natural idea would be trying to "forget" this one known bit to achieve $(c-1)$-dimensional ECDLP of size 2^{k+1}. But let's see that:

$$\left(\frac{2}{\sqrt{3}}\right)^c \sqrt{\pi 2^k} < \left(\frac{2}{\sqrt{3}}\right)^{c-1} \sqrt{\pi 2^{k+1}},$$

because $\frac{2}{\sqrt{3}} < \sqrt{2}$.

Now it is easy to see that if $c \geq 3$ then any trying of reduction of c-dimensional ECDLP into ECDLP of smaller dimension by forgetting some of known bits is not computationally efficient. It is worth to note that transformation of 2-dimensional ECDLP into 1-dimensional ECDLP is computationally efficient because 1-dimensional Gaudry-Schost is more interesting in many practical applications and thus this algorithm is more elaborate and asymptotic formula for average time of solving 1-dimensional ECDLP using 1-dimensional Gaudry-Schost algorithm much differs from asymptotic formula for multidimensional Gaudry-Schost algorithm.

Also reduction of 3-dimensional ECDLP into 1-dimensional ECDLP is computationally inefficient even if both values m_1 and m_2 are equal to one:

$$K = \underbrace{known\ bits}_{m_2}\ \underbrace{unknown\ bits}_{l_2}\ \underbrace{known\ bits}_{m_2}\ \underbrace{unknown\ bits}_{l_2}\ \underbrace{known\ bits}_{m_1}\ \underbrace{unknown\ bits}_{l_1},$$

because such inequality holds:

$$\left(\frac{2}{\sqrt{3}}\right)^3 \sqrt{\pi 2^k} < 1.47\sqrt{\pi 2^{k+2}}$$

and it is easy to see that in this case transformation of 3-dimensional ECDLP into 1-dimensional ECDLP is not the best idea.

5 Conclusion

It was showed that if c disjoint intervals $(c > 1)$ of unknown bits are given and two most significant bits of K are known, then it is impossible to transform c-dimensional ECDLP into 1-dimensional ECDLP of the same size. It is very important from practical point of view, because multidimensional ECDLP has longer expected time of searching for solution. Experimental results show that for two disjoint intervals of unknown bits it is computationally efficient to join these intervals by "forgetting" known bit between them (situation when $m_1 = 1$). For $m_1 > 1$ using multidimensional Gaudry-Schost algorithm should be faster than 1-dimensional version with "forgotten" known bits between intervals.

References

1. Gaudry, P., Schost, É.: A low-memory parallel version of Matsuo, Chao, and Tsujii's algorithm. In: Buell, D. (ed.) ANTS 2004. LNCS, vol. 3076, pp. 208–222. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24847-7_15
2. Genkin, D., Shamir, A., Tromer, E.: RSA key extraction via low-bandwidth acoustic cryptanalysis. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 444–461. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44371-2_25
3. Gopalakrishnan, K., Thériault, N., Yao, C.Z.: Solving discrete logarithms from partial knowledge of the key. In: Srinathan, K., Rangan, C.P., Yung, M. (eds.) INDOCRYPT 2007. LNCS, vol. 4859, pp. 224–237. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-77026-8_17
4. Goubin, L.: A refined power-analysis attack on elliptic curve cryptosystems. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 199–211. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36288-6_15
5. Lange, T., van Vredendaal, C., Wakker, M.: Kangaroos in side-channel attacks. In: Joye, M., Moradi, A. (eds.) CARDIS 2014. LNCS, vol. 8968, pp. 104–121. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16763-3_7
6. Pollard, J.M.: Kangaroos, monopoly and discrete logarithms. J. Cryptol. **13**, 437–447 (2000)
7. Ruprai, R.S.: Improvements to the Gaudry-Schost algorithm for multidimensional discrete logarithm problems and applications. Royal Holloway University of London (2010)

Explicit Bound for the Prime Ideal Theorem in Residue Classes

Maciej Grześkowiak(✉)

Faculty of Mathematics and Computer Science, Adam Mickiewicz University,
Umultowska 87, 61-614 Poznań, Poland
maciejg@amu.edu.pl

Abstract. We give explicit numerical estimates for the generalized Chebyshev functions. Explicit results of this kind are useful for estimating the computational complexity of algorithms which generate special primes. Such primes are needed to construct an elliptic curve over a prime field using the complex multiplication method.

1 Introduction

Let K denote any fixed totally imaginary field of discriminant $\Delta = \Delta(K)$ and degree $[K : \mathbb{Q}] = 2r_2$, where $2r_2$ is the number of complex-conjugate fields of K. Denote by $\mathfrak{f}$ a given non-zero integral ideal of the ring of algebraic integers $\mathcal{O}_K$ and by $H \pmod{\mathfrak{f}}$ any ideal class mod $\mathfrak{f}$ in the "narrow" sense. Let $h^*_{\mathfrak{f}}(K)$ be the number of elements of H. Let $\chi(H)$ be a character of the abelian group of ideal classes $H \pmod{\mathfrak{f}}$, and let $\chi(\mathfrak{a})$ be the usual extension of $\chi(H)$. Let $s = \sigma + it$. The Hecke–Landau zeta-functions associated to χ, are defined by

$$\zeta(s, \chi) = \sum_{\mathfrak{a} \in \mathcal{O}_K} \frac{\chi(\mathfrak{a})}{(N\mathfrak{a})^s}, \qquad \sigma > 1, \tag{1}$$

where $\mathfrak{a}$ runs through integral ideals and $N\mathfrak{a}$ is the norm of $\mathfrak{a}$. Throughout, χ_0 denotes the principal character modulo $\mathfrak{f}$. Let

$$E_0 = E_0(\chi) = \begin{cases} 1 \text{ for } \chi = \chi_0 \\ 0 \text{ for } \chi \neq \chi_0 \end{cases}$$

If χ is a primitive character, then $\zeta(s, \chi)$ satisfies the functional equation

$$\Phi(s, \chi) = W(\chi)\Phi(1 - s, \overline{\chi}), \qquad |W(\chi)| = 1,$$

where

$$\Phi(s, \chi) = A(\mathfrak{f})^s \Gamma(s)^{r_2} \zeta(s, \chi)$$

The author was partially supported by the grant no. 2013/11/B/ST1/02799 from the National Science Centre.

J. Kaczorowski et al. (Eds.): NuTMiC 2017, LNCS 10737, pp. 48–68, 2018.
https://doi.org/10.1007/978-3-319-76620-1_4

and

$$A(\mathfrak{f}) = (2\pi)^{-r_2}\sqrt{|\Delta|N\mathfrak{f}}. \tag{2}$$

Let $\Lambda(\mathfrak{a})$ be the generalized Mangoldt function. Fix $X \mod \mathfrak{f} \in H$. We define,

$$\Psi(x, X) = \sum_{\substack{x \leq N\mathfrak{a} \leq 2x \\ \mathfrak{a} \in X}} \Lambda(\mathfrak{a}) = \sum_{\substack{x \leq N\mathfrak{p}^m \leq 2x \\ \mathfrak{p}^m \in X}} \log N\mathfrak{p},$$

where $\mathfrak{p}$ runs through prime ideals of $\mathcal{O}_K$. The aim of this paper is to prove the following theorem.

Theorem 1. *Let K, Δ, $\mathfrak{f}$, $\zeta(s,\chi)$ denote respectively any algebraic number field of degree $[K:\mathbb{Q}] = 2r_2$, the discriminant of K, any integral ideal in K and any Hecke-Landau zeta function with a character χ modulo $\mathfrak{f}$. Fix $0 < \varepsilon < 1$. If $|\Delta| \geq 9$ and there is no zero in the region*

$$\sigma \geq 1 - 0.0795\left(\log|\Delta| + 0.7761\log\left((|t|+1)^{2r_2}(N\mathfrak{f})^{1-E_0}\right)\right)^{-1}, \tag{3}$$

then

$$\Psi(x, X) \geq \frac{x(1-\varepsilon)}{h^*_{\mathfrak{f}}(K)},$$

for

$$\log x \geq \left(23.148\sqrt{r_2}\left(1 + \left(2\log\left(\frac{c_1\sqrt{r_2}}{0.117\varepsilon}\right)\right)^{\frac{1}{2}} + \frac{2}{3}\log\left(\frac{c_1\sqrt{r_2}}{0.117\varepsilon}\right)\right)\right)^2,$$

where

$$c_1 = \left(40506.547|\Delta|^{\frac{1.933}{r_2}} + 15061.779|\Delta|^{\frac{1.289}{r_2}}(N\mathfrak{f})^{\frac{1}{r_2}}h^*_{\mathfrak{f}}(K)\right)r_2^2\log(|\Delta|N\mathfrak{f}).$$

Remark 1. For real $\chi \pmod{\mathfrak{f}}$ the function $\zeta(s,\chi)$ may have a real, simple zero in (3). However, we can check numerically whether a Hecke-Landau ζ function has a simple real zero in (3) using scripts for computing zeros of zeta functions associated to characters of finite order [13].

Explicit results of this kind are useful for estimating the computational complexity of algorithms which generates special primes. Such primes can be used in computational number theory and cryptography. In order to analyse the running time of these algorithms one need an explicit bound for the number of desired primes from the interval $[x, 2x]$, $x \geq x_0$, where x_0 is computed explicitly. We give an example of such an algorithm. For this reason we recall the following definition [6].

Definition 1. *Let p, q be a pair of primes and $\Delta < 0$. The primes p, q are defined to be CM-primes with respect to Δ if there exist integers f and t such that*

$$|t| \leq 2\sqrt{p}, \quad q|p+1-t, \quad 4p - t^2 = \Delta f^2. \tag{4}$$

If CM-primes p and q with respect to Δ and integers f, t are given, then an ordinary elliptic curve E over $\mathbb{F}_p$ of cardinality $p+1-t$ can be constructed using complex multiplication method [1,3]. Let $E(\mathbb{F}_p)$ be the group of points on E over $\mathbb{F}_p$, and let $|E(\mathbb{F}_p)|$ be the order of $E(\mathbb{F}_p)$. The group $E(\mathbb{F}_p)$ can be used to implement public key cryptographic systems, based on intractability of the discrete logarithm problem (DLP). To make the DLP in $E(\mathbb{F}_p)$ intractable, it is essential to generate a large prime p, and a curve E defined over $\mathbb{F}_p$, such that $|E(\mathbb{F}_p)|$ has a large prime factor q. In [6] an algorithmic method for constructing a pair (E, p) such that $|E(\mathbb{F}_p)|$ has a large prime factor q is given. Fix K an imaginary quadratic number field, and positive integers m, n, $(n, m) = 1$. Then the algorithm generates $\alpha \in \mathcal{O}_K$ such that $q = N_{K/\mathbb{Q}}(\alpha) \equiv m \pmod{n}$ is a prime, and $x \leq q \leq 2x$ for sufficiently large $x \geq x_0$. Given α, q a prime p, $x < p < x^{\frac{5}{2-5\varepsilon}}$, is constructed, where $0 < \varepsilon < \frac{2}{5}$. For more algorithms of this kind we refer the reader to [7,8].

Let $x \in \mathbb{R}$, and let $W(x)$ be the Lambert W function such that $W(x)e^{W(x)} = x$. If $-e^{-1} \leq x \leq 0$, then there are two possible real values of $W(x)$. We denote the branch satisfying $-1 < W(x)$ by $W_0(x)$ and the branch satisfying $W(x) \leq -1$ by $W_{-1}(x)$. Fix $X \pmod{\mathfrak{f}} \in H$. We define

$$\psi(x, X) := \sum_{\substack{N\mathfrak{p}^m < x \\ \mathfrak{p}^m \in X}} \log N\mathfrak{p},$$

where $\mathfrak{p}$ runs through prime ideals of $\mathcal{O}_K$. Theorem 1 follows from Theorem 2.

Theorem 2. *Let K, Δ, $\mathfrak{f}$, $\zeta(s, \chi)$ denote respectively any algebraic number field of degree $[K : \mathbb{Q}] = 2r_2$, the discriminant of K, any integral ideal in K and any Hecke-Landau zeta function with a character χ modulo $\mathfrak{f}$. Let $A_0 = 0.7761$. If $|\Delta| \geq 9$ and there is no zero in the region*

$$\sigma \geq 1 - 0.0795 \left(\log|\Delta| + A_0 \log\left((|t| + 1)^{2r_2} (N\mathfrak{f})^{1-E_0(\chi)} \right) \right)^{-1},$$

then

$$\psi(x, X) \geq \frac{x}{h_{\mathfrak{f}}^*(K)} - \frac{c_2 x}{h_{\mathfrak{f}}^*(K)} (\log x)^{\frac{1}{2}} e^{-0.0432 r_2^{-1/2} \sqrt{\log x}},$$

and

$$\psi(x, X) \leq \frac{x}{h_{\mathfrak{f}}^*(K)} + \frac{c_3 x}{h_{\mathfrak{f}}^*(K)} (\log x)^{\frac{1}{2}} e^{-0.0459 r_2^{-1/2} \sqrt{\log x}}$$

for $x \geq \exp\left(116 r_2 \log\left(2|\Delta|^{\frac{1}{A_0 r_2}} (N\mathfrak{f})^{\frac{1}{r_2}}\right)\right)$, *where*

$$c_2 = \left(10756.967|\Delta|^{\frac{3}{2A_0 r_2}} + 3999.824|\Delta|^{\frac{1}{A_0 r_2}} (N\mathfrak{f})^{\frac{1}{r_2}} h_{\mathfrak{f}}^*(K)\right) r_2^2 \log(|\Delta| N\mathfrak{f}),$$

$$c_3 = \left(18164.326|\Delta|^{\frac{3}{2A_0 r_2}} + 6754.144|\Delta|^{\frac{1}{A_0 r_2}} (N\mathfrak{f})^{\frac{1}{r_2}} h_{\mathfrak{f}}^*(K)\right) r_2^2 \log(|\Delta| N\mathfrak{f}).$$

Proof. See Sect. 2.

We are now in a position to prove Theorem 1.

Proof. By Theorem 2 we have

$$\psi(2x, X) - \psi(x, X) \geq \frac{x}{h_{\mathfrak{f}}^*(K)} - \frac{c_1 x}{h_{\mathfrak{f}}^*(K)} (\log x)^{\frac{1}{2}} e^{-0.0432 r_2^{-1/2} \sqrt{\log x}},$$

where

$$c_1 = \left(2c_2 \left(1 + \frac{\log 2}{\log x}\right)^{\frac{1}{2}} + c_3\right) \leq 2.077 c_2 + c_3$$

for $x \geq \exp\left(116 r_2 \log\left(2|\Delta|^{\frac{1}{A_0 r_2}} (N\mathfrak{f})^{\frac{1}{r_2}}\right)\right)$. Fix $0 < \varepsilon < 1$. If

$$c_1 (\log x)^{\frac{1}{2}} e^{-0.0432 r_2^{-1/2} \sqrt{\log x}} \leq \varepsilon,$$

then

$$0.0432 r_2^{-\frac{1}{2}} (\log x)^{\frac{1}{2}} \geq -W_{-1}\left(\frac{-0.0432\varepsilon}{c_1 \sqrt{r_2}}\right).$$

By [2, Theorem 1]

$$\log x \geq \left(23.148\sqrt{r_2}\left(1 + \left(2\log\left(\frac{c_1\sqrt{r_2}}{0.117\varepsilon}\right)\right)^{\frac{1}{2}} + \frac{2}{3}\log\left(\frac{c_1\sqrt{r_2}}{0.117\varepsilon}\right)\right)\right)^2.$$

This finishes the proof.

2 The Proof of Theorem 2

The proof of Theorem 2 rests on the following lemmas and theorems.

Theorem 3. *Let* K, $\mathfrak{f}$, $\zeta(s,\chi)$ *denote respectively any algebraic number field of degree* $n \geq 2$, *any integral ideal in* K *and any Hecke-Landau zeta function with a character* χ *modulo* $\mathfrak{f}$. *Let futher*

$$L(t) = \log|\Delta| + A_0 \log\left((|t|+1)^n (N\mathfrak{f})^{1-E_0}\right) \geq 2.097. \tag{5}$$

Then in the case of the complex χ *in the region*

$$\sigma \geq 1 - \frac{A_1}{L(t)} \geq 1 - 0.037911 = 0.962089 = A_2 \tag{6}$$

there is no zero of $\zeta(s,\chi)$, *where* $A_0 = 0.7761$, $A_1 = 0.0795$. *For real* χ (mod $\mathfrak{f}$) *the function* $\zeta(s,\chi)$ *may have a real, simple zero in (6).*

Proof. See [5, Theorem 2].

Lemma 1. *Let $s = \sigma + it$, $0 < \eta \leq \frac{1}{4}$, $A_3 = 75.472$, $A_4 = 0.010$ and $|\Delta| \geq 9$. Assume that there is no exceptional zero in the region (6). Then in the strip $1 - \frac{A_1}{6L(t)} \leq \sigma \leq 3$ we have*

$$\left|\frac{\zeta'}{\zeta}(s, \chi_0) + \frac{1}{s-1}\right| \leq \phi_0(t, r_2, \eta, \Delta, \mathfrak{f}),$$

where

$$\begin{aligned}\phi_0(t, r_2, \eta, \Delta, \mathfrak{f}) = 32 \log &\left(L(t)(|t|+4)(|t|+2)^{r_2(1+\eta)}\left(1 + A_3 L(t)\right)^{2r_2}\right) \\ &+ 32 \log\left(A_3(|\Delta| N\mathfrak{f})^{\frac{1+\eta}{2}} \zeta(1+\eta)^{2r_2}\right) + 8 A_3 r_2 L(t) + \frac{A_4 r_2}{L(t)},\end{aligned} \tag{7}$$

and

$$\left|\frac{\zeta'}{\zeta}(s, \chi)\right| \leq \phi(t, r_2, \eta, \Delta, \mathfrak{f}),$$

where

$$\begin{aligned}\phi(t, r_2, \eta, \Delta, \mathfrak{f}) = 32 \log &\left(\left(1 + A_3 L(t)\right)^{2r_2} (|t|+4)^{r_2(1+2\eta)}\right) \\ &+ 32 \log\left(1.4(1+\varepsilon_\chi) A(\mathfrak{f})^{1+2\eta} \zeta(1+\eta)^{2r_2}\right) + 4 A_3 r_2 L(t) + \frac{A_4 r_2}{L(t)}\end{aligned} \tag{8}$$

for any character $\chi \neq \chi_0$ modulo $\mathfrak{f}$, where $\varepsilon_\chi = 0$ or 1 to accordingly whether χ is primitive or not.

Proof. See Sect. 3.

Lemma 2. *Let ϕ_0, ϕ be functions defined in Lemma 1. Let $T \geq 1$, $w \geq 1$, $|\Delta| \geq 9$, $c_4 = \frac{1}{\sqrt{2wr_2}}$ and*

$$c_0 = c_0(\Delta, \mathfrak{f}, r_2, E_0) = |\Delta|^{-\frac{1}{2A_0 r_2}} (N\mathfrak{f})^{-\frac{1-E_0}{2r_2}}. \tag{9}$$

If

$$T + 1 = c_0 \exp\left(c_4 \sqrt{\log x}\right), \tag{10}$$

then

$$\begin{aligned}\phi(T, r_2, \eta, \Delta, \mathfrak{f}) &\leq 287.790 r_2^{\frac{3}{2}} \log(|\Delta| N\mathfrak{f})(\log x)^{\frac{1}{2}} \\ \phi_0(T, r_2, \eta, \Delta, \mathfrak{f}) &\leq 479.346 r_2^{\frac{3}{2}} \log(|\Delta| N\mathfrak{f})(\log x)^{\frac{1}{2}}\end{aligned}$$

for $x \geq \exp\left((c_4^{-1} \log(2c_0^{-1}))^2\right)$.

Proof. By (5), (10) we obtain

$$L(T) = \frac{A_0\sqrt{2r_2}}{\sqrt{w}}(\log x)^{\frac{1}{2}}, \quad x \geq \exp\left((c_4^{-1}\log(2c_0^{-1}))^2\right). \tag{11}$$

Since $T+1 \geq 2$, $\log x \geq 2r_2 w\left(\log(2c_0^{-1})\right)^2 \geq 2(\log(2\cdot 9^{\frac{1}{2A_0}}))^2 \geq 8.892$, and hence $x > e^{8.892}$. Let $\eta = \frac{1}{4}$. We have $\zeta\left(\frac{5}{4}\right) \leq 4.596$,

$$\begin{aligned}32\log(1+A_3L(T))^{2r_2} &\leq 32r_2\log\log x + 64r_2\log(A_0\sqrt{2r_2}(A_3+\frac{1}{2.097})) \\ &\leq 190.837r_2^{\frac{3}{2}}\log\log x,\end{aligned}$$

$$32r_2(1+2\eta)\log(T+4) \leq 32r_2(1+2\eta)\left(\frac{1}{\sqrt{2wr_2}}(\log x)^{\frac{1}{2}} + \log\frac{5}{2}\right) \leq 48.691r_2^{\frac{1}{2}}(\log x)^{\frac{1}{2}},$$

$$4A_3r_2L(T) \leq 4\sqrt{2}A_0A_3r_2^{\frac{3}{2}}(\log x)^{\frac{1}{2}} \leq 331.344r_2^{\frac{3}{2}}(\log x)^{\frac{1}{2}},$$

$$\begin{aligned}32\log\left(1.4(1+\varepsilon_\chi)A(\mathfrak{f})^{1+2\eta}\zeta(1+\eta)^{2r_2}\right) + \frac{A_4r_2}{L(T)} &\leq 32\log\left(2.8\zeta(1+\eta)^{2r_2}\right) \\ + 16(1+2\eta)\log(|\Delta|N\mathfrak{f}) + \frac{A_4r_2}{0.429} &\leq 83.417r_2\log(|\Delta|N\mathfrak{f}).\end{aligned}$$

By the above and (8) we obtain

$$\phi(T, r_2, \eta, \Delta, \mathfrak{f}) \leq 287.790r_2^{\frac{3}{2}}\log(|\Delta|N\mathfrak{f})(\log x)^{\frac{1}{2}},$$

Similarly,

$$32\log L(T) \leq 16\log\log x + 32\log(\sqrt{2r_2}A_0) \leq 24.686r_2^{\frac{1}{2}}\log\log x,$$

$$32\log(T+4)^{r_2(1+\eta)+1} \leq 81.151r_2(\log x)^{\frac{1}{2}},$$

and

$$32\log\left(A_3(|\Delta|N\mathfrak{f})^{\frac{1+\eta}{2}}\zeta(1+\eta)^{2r_2}\right) + \frac{A_4r_2}{L(t)} \leq 127.393r_2\log(|\Delta|N\mathfrak{f}).$$

By the above and (7) we obtain

$$\phi_0(T, r_2, \eta, \Delta, \mathfrak{f}) \leq 479.346r_2^{\frac{3}{2}}\log(|\Delta|N\mathfrak{f})(\log x)^{\frac{1}{2}}.$$

This finishes the proof.

Lemma 3. *Let $T \geq 1$, $w \geq 1$, $|\Delta| \geq 9$, and let $k \geq 1$. Let c_0, c_4, T be defined as in Lemma 2. If*

$$T + 1 = c_0\exp\left(\sqrt{\frac{\log x}{2wr_2}}\right), \tag{12}$$

then

$$\frac{1}{T^k} \leq 2^k c_0^{-k} e^{-kc_4\sqrt{\log x}} \qquad \textit{for} \qquad \log x \geq (c_4^{-1}\log(2c_0^{-1}))^2, \tag{13}$$

$$\log(e(T+k)) \leq c_4\sqrt{\log x} + \log\left(e\left(\frac{k+1}{2}\right)\right). \tag{14}$$

Proof. By (12) we have

$$\frac{1}{T^k} = \exp(-k\log(c_0e^{c_4\sqrt{\log x}}(1-(c_0e^{c_4\log x})^{-1}))) \leq \exp(-k\log(\frac{1}{2}c_0e^{c_4\sqrt{\log x}})),$$

for $\log x \geq (c_4^{-1}\log(2c_0^{-1}))^2$. The proof of (14) is left to the reader. This finishes the proof.

Lemma 4. *For $T \geq 1$ we have*

$$\int_T^\infty t^{-2}dt \leq T^{-1}, \qquad \int_T^\infty t^{-2}\log(t+4)dt \leq T^{-1}\log(e(T+4))$$

Proof. The proof is left to the reader.

Lemma 5. *Let $L(t)$ be the function which occur in (5). For $T \geq 1$ we have*

$$\int_T^\infty t^{-2}L(t)dt \leq c_5T^{-1}\log(e(T+4)),$$

where $c_5 = 1.09r_2\log\left(|\Delta|(N\mathfrak{f})^{A_0(1-E_0)}\right)$.

Proof. We have

$$\int_T^\infty t^{-2}L(t)dt \leq 2r_2A_0\int_T^\infty t^{-2}\log(t+4)dt + \log\left(|\Delta|(N\mathfrak{f})^{A_0(1-E_0)}\right)\int_T^\infty t^{-2}dt.$$

The Lemma 5 follows from Lemma 4. This finishes the proof.

Lemma 6. *Let $L(t)$ be the function which occur in (5). For $T \geq 1$ we have*

$$\int_T^\infty t^{-2}\log(1+A_3L(t))^{2r_2}dt \leq c_6T^{-1}\log(e(T+4)),$$

where $c_6 = 11.605r_2^2\log\left(|\Delta|(N\mathfrak{f})^{A_0(1-E_0)}\right)$, and A_3 is the constant appearing in Lemma 1.

Proof. By (5) we have

$$\int_T^\infty t^{-2}\log\left(1+A_3L(t)\right)^{2r_2}dt \le 2r_2c_7\int_T^\infty t^{-2}dt+2r_2\int_T^\infty t^{-2}L(t)dt,$$

where $c_7=\log\left(A_3\left(1+\frac{1}{2.097A_3}\right)\right)$. The Lemma 6 follows from Lemmas 4 and 5. This finishes the proof.

Lemma 7. *Let ϕ_0 be the function which occur in (7). For $T\ge 1$ we have*

$$\int_T^\infty \phi_0(t,r_2,\eta,\varDelta,\mathfrak{f})t^{-2}dt\le c_8T^{-1}\log(e(T+4)),$$

where $c_8=1138.428r_2^2\log(|\varDelta|(N\mathfrak{f})^{\frac{5}{8}})$.

Proof. By (5) and (7) with $\eta=\frac{1}{4}$ we have

$$\begin{aligned}\int_T^\infty \phi_0(t,r_2,\eta,\varDelta,\mathfrak{f})t^{-2}dt &\le (40r_2+32)\int_T^\infty t^{-2}\log(t+4)dt\\ &+\left(32\log(A_3(|\varDelta|N\mathfrak{f})^{\frac{5}{8}})+64r_2\log\zeta\left(\frac{5}{4}\right)+\frac{A_4r_2}{2.097}\right)\int_T^\infty t^{-2}dt\\ &+(32+8A_3r_2)\int_T^\infty t^{-2}L(t)dt+32\int_T^\infty t^{-2}\log\left(1+A_3L(t)\right)^{2r_2}dt.\end{aligned}$$

The Lemma 7 follows from Lemmas 4, 5 and 6. This finishes the proof.

Lemma 8. *Let ϕ be the function which occur in (8). For $T\ge 1$ we have*

$$\int_T^\infty \phi(t,r_2,\eta,\varDelta,\mathfrak{f})t^{-2}dt\le c_9T^{-1}\log(e(T+4)),$$

where $c_9=835.777r_2^2\log\left(|\varDelta|N\mathfrak{f}\right)$.

Proof. By (5) and (8) with $\eta=\frac{1}{4}$ we have

$$\begin{aligned}\int_T^\infty \phi(t,r_2,\eta,\varDelta,\mathfrak{f})t^{-2}dt &\le \left(32\log\left(2.8A(\mathfrak{f})^{\frac{3}{2}}\zeta\left(\frac{5}{4}\right)^{2r_2}\right)+\frac{A_4r_2}{2.097}\right)\int_T^\infty t^{-2}dt\\ &+32\int_T^\infty t^{-2}\log\left(1+A_3L(t)\right)^{2r_2}dt+48r_2\int_T^\infty t^{-2}\log\left(t+4\right)dt\\ &+4A_3r_2\int_T^\infty L(t)t^{-2}dt.\end{aligned}$$

The Lemma 8 follows from Lemmas 4, 5 and 6. This finishes the proof.

We are now in a position to prove Theorem 2.

Proof. Fix $T \geq 1$, and let $c = 1 + \frac{1}{\log x}$. Fix $X \pmod{\mathfrak{f}}$. We define

$$\psi_1(x, X) := \int_0^x \psi(t, X)dt, \tag{15}$$

and

$$\gamma(n) = \sum_{\substack{N\mathfrak{p}^m = n \\ \mathfrak{p}^m \in X}} \log N\mathfrak{p}.$$

Hence,

$$\psi(x, X) = \sum_{n \leq x} \gamma(n).$$

By partial summation we obtain

$$\sum_{n \leq x} (x - n)\gamma(n) = \int_0^x \psi(t, X)dt.$$

Now, we write

$$f(s, \chi) = \frac{x^{s-1}}{s(s+1)} \left[-\frac{\zeta'}{\zeta}(s, \chi) \right].$$

By Theorem B [10, see p. 31] and the orthogonality properties of $\chi \pmod{\mathfrak{f}}$ we deduce the formula

$$\sum_{n \leq x} (x - n)\gamma(n) = \frac{x^2}{2\pi i h_{\mathfrak{f}}^*(K)} \sum_{\chi} \overline{\chi}(X) \int\limits_{c-i\infty}^{c+i\infty} f(s, \chi)ds, \tag{16}$$

where $c > 1$. Let A_1 be the constat appearing in (6), and let $B = \frac{A_1}{6} = 0.01325$. We define the contour $\mathcal{C}$ consisting of the following parts:

$$\begin{aligned} \mathcal{C}_1 &: s = c + it, \text{where} \quad -T \leq t \leq T, \\ \mathcal{C}_2 &: s = \sigma + iT, \text{where} \ \ 1 - \frac{B}{L(T)} \leq \sigma \leq c, \\ \mathcal{C}_3 &: s = 1 - \frac{B}{L(T)} + it, \text{where} \quad -T \leq t \leq T. \end{aligned} \tag{17}$$

and of $\mathcal{C}_2'$ situated symmetrically to $\mathcal{C}_2$. If $\chi = \chi_0$, them $\frac{\zeta'}{\zeta}(s, \chi)$ has a first order pole of residue -1 at $s = 1$. From the Cauchy formula we get

$$\frac{1}{2\pi i} \int\limits_{\mathcal{C}_1} f(s, \chi)ds = \frac{\delta(\chi)}{2} - \frac{1}{2\pi i} \int\limits_{\mathcal{C}_2 + \mathcal{C}_3 + \mathcal{C}_2'} f(s, \chi)ds, \tag{18}$$

where

$$\delta(\chi) = \begin{cases} 1 \text{ if } \chi = \chi_0, \\ 0 \text{ if } \chi \neq \chi_0. \end{cases}$$

From (15), (16) and (18) we obtain

$$\left| \psi_1(x, X) - \frac{x^2}{2h^*_{\mathfrak{f}}(K)} \right| \le \frac{x^2(I_1 + I_2 + I_3)}{h^*_{\mathfrak{f}}(K)} + \frac{x^2(J_1 + J_2 + J_3)}{h^*_{\mathfrak{f}}(K)}, \tag{19}$$

where

$$I_1 + I_2 + I_3 = \left| \frac{1}{2\pi i} \int\limits_{c-i\infty}^{c-iT} f(s, \chi_0) \right| + \left| \frac{1}{2\pi i} \int\limits_{\mathcal{C}_2 + \mathcal{C}_3 + \mathcal{C}'_2} f(s, \chi_0) ds \right| + $$
$$+ \left| \frac{1}{2\pi i} \int\limits_{c+iT}^{c+i\infty} f(s, \chi_0) ds \right|,$$

$$J_1 + J_2 + J_3 = \left| \sum_{\chi \neq \chi_0} \overline{\chi}(X) \frac{1}{2\pi i} \int\limits_{c-i\infty}^{c-iT} f(s, \chi) ds \right| +$$
$$+ \left| \sum_{\chi \neq \chi_0} \overline{\chi}(X) \frac{1}{2\pi i} \int\limits_{\mathcal{C}_2 + \mathcal{C}_3 + \mathcal{C}'_2} f(s, \chi) ds \right| + \left| \sum_{\chi \neq \chi_0} \overline{\chi}(X) \frac{1}{2\pi i} \int\limits_{c+iT}^{c+i\infty} f(s, \chi) ds \right|.$$

We define

$$h_0(s, \chi_0) = \left[-\frac{\zeta'}{\zeta}(s, \chi_0) - \frac{1}{s-1} \right] \frac{x^{s-1}}{s(s+1)}, \quad h_1(s) = \frac{x^{s-1}}{s(s+1)(s-1)}.$$

Then

$$f(s, \chi_0) = h_0(s, \chi_0) + h_1(s). \tag{20}$$

We estimate the above integrals. Let $T \ge 1$, $x \ge e^{8.892}$, $1 < c = 1 + \frac{1}{\log x} \le 1.12$. We need to consider the following cases:

1. Bound over $\mathcal{C}_2$ and $\mathcal{C}'_2$, case $\chi = \chi_0$. In this case $c_0 \le 9^{-\frac{1}{2A_0}} \le \frac{1}{4}$. From Lemmas 1, 2 and 3 we obtain

$$\left| \frac{1}{2\pi i} \int\limits_{\mathcal{C}_2} f(\sigma + iT, \chi_0) d\sigma \right| \le \frac{e}{2\pi T^2 \log x} \phi_0(T, r_2, \eta, \Delta, \mathfrak{f}) + \frac{e}{2\pi T^3 \log x}$$
$$\le c_0^{-3} r_2^{\frac{3}{2}} \log(|\Delta| N\mathfrak{f}) (\log x)^{-\frac{1}{2}} \left(\frac{2ec_0 479.346}{\pi} + \frac{4e}{\pi (\log x)^{\frac{1}{2}}} \right) e^{-2c_4 \sqrt{\log x}}$$
$$\le c_{10} (\log x)^{-\frac{1}{2}} e^{-2c_4 \sqrt{\log x}},$$

where $c_{10} = 208.540|\Delta|^{\frac{3}{2A_0 r_2}} r_2^{\frac{3}{2}} \log(|\Delta| N\mathfrak{f})$. The same bound holds with $\int_{\mathcal{C}_2'}$ in place of $\int_{\mathcal{C}_2}$.

2. Bound over $\mathcal{C}_3$, case $\chi = \chi_0$. Lemmas 1, 2 and 3 shows that

$$\begin{aligned}\left|\frac{1}{2\pi i}\int_{\mathcal{C}_3} h_0\left(1-\frac{B}{L(T)}+it,\chi_0\right)dt\right| &\leq \frac{1}{\pi}x^{-\frac{B}{L(T)}}\phi_0(T,r_2,\eta,\Delta,\mathfrak{f})\\ &\cdot\int_0^T \frac{dt}{\left(1-\frac{B}{L(T)}\right)^2+t^2} \leq \frac{1}{\pi}2.01e^{-c_{18}\sqrt{\log x}}\phi_0(T,r_2,\eta,\Delta,\mathfrak{f})\\ &\leq 306.687 r_2^{\frac{3}{2}}\log(|\Delta|N\mathfrak{f})(\log x)^{\frac{1}{2}}e^{-c_{18}\sqrt{\log x}},\end{aligned}$$

where $c_{18} = \frac{B\sqrt{w}}{A_0\sqrt{2r_2}}$. Indeed, $1-\frac{B}{L(T)} \geq 1-\frac{0.0133}{2.097} \geq 0.993$, and

$$\begin{aligned}\int_0^T \frac{dt}{\left(1-\frac{B}{L(T)}\right)^2+t^2} &= \int_0^1 \frac{dt}{\left(1-\frac{B}{L(T)}\right)^2+t^2}+\int_1^T \frac{dt}{\left(1-\frac{B}{L(T)}\right)^2+t^2}\\ &\leq \int_0^1 \frac{dt}{(0.993)^2}+\int_1^T\frac{dt}{t^2} \leq \frac{1}{(0.993)^2}+1 \leq 2.01.\end{aligned}$$

Moreover,

$$\begin{aligned}&\left|\frac{1}{2\pi i}\int_{\mathcal{C}_3} h_1\left(1-\frac{B}{L(t)}+it\right)ds\right|\\ &\quad\leq \frac{1}{\pi}x^{-\frac{B}{L(T)}}\int_0^T \frac{dt}{\left|1-\frac{B}{L(T)}+it\right|\left|2-\frac{B}{L(T)}+it\right|\left|-\frac{B}{L(T)}+it\right|} \leq \frac{c_{11}}{\pi}e^{-c_{18}\sqrt{\log x}},\end{aligned}$$

where $c_{11} = \frac{1}{0.993|0.993-1|(0.993+1)}+1 \leq 73.185$. By the above and (20),

$$\left|\frac{1}{2\pi i}\int_{\mathcal{C}_3} f\left(1-\frac{B}{L(T)}+it,\chi_0\right)dt\right| \leq c_{12}(\log x)^{\frac{1}{2}}e^{-c_{18}\sqrt{\log x}},$$

where $c_{12} = 267.495 r_2^{\frac{3}{2}}\log(|\Delta|N\mathfrak{f})$. Hence,

$$\begin{aligned}I_2 &\leq |\Delta|^{\frac{3}{2A_0 r_2}} r_2^{\frac{3}{2}}\log(|\Delta|N\mathfrak{f})(\log x)^{\frac{1}{2}}e^{-c_{18}\sqrt{\log x}}\\ &\cdot\left(267.495+\frac{2\cdot 360.992}{\log x}e^{-(2c_4-c_{18})\sqrt{\log x}}\right) \leq c_{13}(\log x)^{\frac{1}{2}}e^{-c_{18}\sqrt{\log x}},\end{aligned} \tag{21}$$

if $w < \frac{2A_0}{B} = 117.148$, where $c_{13} = 348.69|\Delta|^{\frac{3}{2A_0 r_2}} r_2^{\frac{3}{2}}\log(|\Delta|N\mathfrak{f})$.

3. Bound over $\mathcal{C}_2$ and $\mathcal{C}_2'$, case $\chi \neq \chi_0$. From Lemmas 1, 2 and 3 we obtain

$$\left| \frac{1}{2\pi i} \int\limits_{\mathcal{C}_2} f(\sigma + iT, \chi) ds \right| \leq c_{14} (\log x)^{-\frac{1}{2}} e^{-2c_4 \sqrt{\log x}},$$

where $c_{14} = 498.025 |\Delta|^{\frac{1}{A_0 r_2}} (N\mathfrak{f})^{\frac{1}{r_2}} r_2^{\frac{3}{2}} \log(|\Delta| N\mathfrak{f})$. The same bound holds with $\int_{\mathcal{C}_2'}$ in place of $\int_{\mathcal{C}_2}$.
4. Bound over $\mathcal{C}_3$, case $\chi \neq \chi_0$. Lemmas 1, 2 and 3 shows that

$$\begin{aligned} \left| \frac{1}{2\pi i} \int\limits_{\mathcal{C}_3} f\left(1 - \frac{B}{L(T)} + it, \chi\right) ds \right| &\leq \frac{1}{\pi} x^{-\frac{B}{L(T)}} \phi(T, r_2, \eta, \Delta, \mathfrak{f}) \int\limits_0^T \frac{dt}{\left(1 - \frac{B}{L(T)}\right)^2 + t^2} \\ &\leq \frac{1}{\pi} 2.01 e^{-c_{18}\sqrt{\log x}} \phi(T, r_2, \eta, \Delta, \mathfrak{f}) \leq c_{15} (\log x)^{\frac{1}{2}} e^{-c_{18}\sqrt{\log x}}, \end{aligned}$$

where $c_{15} = 184.129 r_2^{\frac{3}{2}} \log(|\Delta| N\mathfrak{f})$. Hence, by the above

$$\begin{aligned} J_2 &\leq \sum_{\chi \neq \chi_0} \overline{\chi}(X) \left(2c_{14} (\log x)^{-\frac{1}{2}} e^{-2c_4\sqrt{\log x}} + c_{15} (\log x)^{\frac{1}{2}} e^{-c_{18}\sqrt{\log x}} \right) \\ &\leq c_{16} (\log x)^{\frac{1}{2}} e^{-c_{18}\sqrt{\log x}}, \end{aligned} \tag{22}$$

if $w < \frac{2A_0}{B} = 117.148$, where $c_{16} = 296.146 h_{\mathfrak{f}}^*(K) |\Delta|^{\frac{1}{A_0 r_2}} (N\mathfrak{f})^{\frac{1}{r_2}} r_2^{\frac{3}{2}} \log(|\Delta| N\mathfrak{f})$.
5. Bound for $\int_{c+iT}^{c+i\infty}$, case $\chi = \chi_0$. By (20) and Lemmas 1, 3, 7 we obtain

$$\begin{aligned} \left| \frac{1}{2\pi i} \int\limits_{c+iT}^{c+i\infty} f(s, \chi_0) ds \right| &\leq \left| \frac{1}{2\pi i} \int\limits_{c+iT}^{c+i\infty} h_0(s, \chi_0) ds \right| + \left| \frac{1}{2\pi i} \int\limits_{c+iT}^{c+i\infty} h_1(s) ds \right| \\ &\leq \frac{e}{2\pi} \int\limits_T^\infty \phi_0(t, r_2, \eta, \Delta, \mathfrak{f}) t^{-2} dt + \frac{e}{2\pi} \int\limits_T^\infty t^{-3} dt \leq \frac{e}{2\pi} c_8 \frac{\log(e(T+4))}{T} \\ &+ \frac{e}{4\pi T^2} \leq \frac{e c_8}{\pi c_0^2} (\log x)^{\frac{1}{2}} \left(\frac{c_0}{\sqrt{2w}} + \frac{1.917 c_0}{(\log x)^{\frac{1}{2}}} + \frac{c_0}{2c_8 (\log x)^{\frac{1}{2}}} \right) e^{-c_4\sqrt{\log x}} \\ &\leq c_{17} (\log x)^{\frac{1}{2}} e^{-c_4\sqrt{\log x}} \end{aligned}$$

for $\log x \geq (c_4^{-1} \log(2c_0^{-1}))^2 \geq 8.892$, with $(\log x)^{\frac{1}{2}} \geq 2.98$, $c_0 \leq 1$, $w = 58$. where $c_{17} = 724.845 |\Delta|^{\frac{1}{A_0 r_2}} r_2^2 \log(|\Delta| (N\mathfrak{f})^{\frac{5}{8}})$. The same bound holds with $\int_{c-i\infty}^{c-iT}$ in place of $\int_{c+iT}^{c+i\infty}$. Hence,

$$I_1 + I_3 \leq 2c_{17} (\log x)^{\frac{1}{2}} e^{-c_4\sqrt{\log x}}. \tag{23}$$

6. Bound for $\int_{c+iT}^{c+i\infty}$, case $\chi \neq \chi_0$. Lemmas 1, 3 and 8 shows that

$$\left|\frac{1}{2\pi i}\int\limits_{c+iT}^{c+i\infty} f(s,\chi)ds\right| \leq \frac{e}{2\pi}\int\limits_{T}^{\infty}\phi(t,r_2,\eta,\varDelta,\mathfrak{f})t^{-2}dt \leq \frac{e}{2\pi}c_9\frac{\log(e(T+4))}{T}$$
$$\leq \frac{ec_9}{\pi c_0}(\log x)^{\frac{1}{2}}\left(\frac{1}{\sqrt{2w}}+\frac{1.917}{(\log x)^{\frac{1}{2}}}\right)e^{-c_4\sqrt{\log x}} \leq c_{19}(\log x)^{\frac{1}{2}}e^{-c_4\sqrt{\log x}}.$$

where $c_{19} = 532.042|\varDelta|^{\frac{1}{2A_0r_2}}(N\mathfrak{f})^{\frac{1}{2r_2}}r_2^2\log(|\varDelta|N\mathfrak{f})$, and $w = 58$. The same bound holds with $\int_{c-i\infty}^{c-iT}$ in place of $\int_{c+iT}^{c+i\infty}$. Hence,

$$J_1 + J_3 \leq 2c_{19}h_{\mathfrak{f}}^*(K)(\log x)^{\frac{1}{2}}e^{-c_4\sqrt{\log x}}. \tag{24}$$

By (21), (23) we have

$$I_1 + I_2 + I_3 \leq c_{20}(\log x)^{\frac{1}{2}}e^{-c_{18}\sqrt{\log x}}, \tag{25}$$

where $c_{20} = 3585.536|\varDelta|^{\frac{3}{2A_0r_2}}r_2^2\log(|\varDelta|N\mathfrak{f})$, for $1 \leq w < \frac{A_0}{B} = 58.57$. From (22), (24) we obtain

$$J_1 + J_2 + J_3 \leq c_{21}(\log x)^{\frac{1}{2}}e^{-c_{18}\sqrt{\log x}}, \tag{26}$$

where $c_{21} = 1333.230h_{\mathfrak{f}}^*(K)|\varDelta|^{\frac{1}{A_0r_2}}(N\mathfrak{f})^{\frac{1}{r_2}}r_2^2\log(|\varDelta|N\mathfrak{f})$ for $1 \leq w < \frac{A_0}{B} = 58.57$. Now, by (19), (25), (26) we obtain

$$\left|\psi_1(x,X) - \frac{x^2}{2h_{\mathfrak{f}}^*(K)}\right| \leq \frac{x^2}{h_{\mathfrak{f}}^*(K)}c_{22}(\log x)^{\frac{1}{2}}e^{-c_{18}\sqrt{\log x}}$$

where $c_{22} = c_{20} + c_{21}$. Now, let $x > 2$, and h be a function of x satisfying $0 < h < \frac{1}{2}x$. Let $W(x) = c_{22}(\log x)^{\frac{1}{2}}e^{-c_{18}\sqrt{\log x}}$. Since $\psi(t,X)$ is an increasing function

$$\psi(x,X) \geq \frac{1}{h}\int_{x-h}^{x}\psi(t,X)dt = \frac{\psi_1(x,X)-\psi_1(x-h,X)}{h}$$
$$\geq \frac{x}{h_{\mathfrak{f}}^*(K)} - \frac{x^2}{hh_{\mathfrak{f}}^*(K)}W(x) - \frac{h}{2h_{\mathfrak{f}}^*(K)} - \frac{x^2+h^2}{hh_{\mathfrak{f}}^*(K)}W(x-h).$$

Taking $h = xe^{-\frac{1}{2}c_{18}\sqrt{\log x}}$ and $x > \exp(\frac{2\log 2}{c_{18}})^2$, we get

$$\begin{aligned}\psi(x,X) \geq{}& \frac{x}{h_{\mathfrak{f}}^*(K)} - \frac{x}{h_{\mathfrak{f}}^*(K)}c_{22}(\log x)^{\frac{1}{2}}e^{-\frac{1}{2}c_{18}\sqrt{\log x}} - \frac{1}{2h_{\mathfrak{f}}^*(K)}xe^{-\frac{1}{2}c_{18}\sqrt{\log x}}\\ &- \frac{x}{h_{\mathfrak{f}}^*(K)}c_{22}(\log x)^{\frac{1}{2}}e^{-c_{18}(c_{23}-0.5)\sqrt{\log x}} - \frac{x}{h_{\mathfrak{f}}^*(K)}c_{22}(\log x)^{\frac{1}{2}}e^{-c_{18}(c_{23}+0.5)\sqrt{\log x}}\\ \geq{}& \frac{x}{h_{\mathfrak{f}}^*(K)} - \frac{x}{h_{\mathfrak{f}}^*(K)}c_{22}(\log x)^{\frac{1}{2}}e^{-0.47c_{18}\sqrt{\log x}}(3+c_{24})\\ \geq{}& \frac{x}{h_{\mathfrak{f}}^*(K)} - \frac{c_2x}{h_{\mathfrak{f}}^*(K)}(\log x)^{\frac{1}{2}}e^{-0.47c_{18}\sqrt{\log x}},\end{aligned} \tag{27}$$

where $c_{23} = (1 - \frac{\log 2}{\log x})^{\frac{1}{2}}$, $0.97 \le c_{23} \le 0.98$, $c_{24} = \frac{1}{2c_{22}}(\log x)^{-\frac{1}{2}} \le 0.0001$, and $c_2 = c_{22}(3 + c_{24})$. On the other hand,

$$\begin{aligned}\psi(x,X) &\le \frac{1}{h}\int_x^{x+h} \psi(t,X)dt = \frac{\psi_1(x+h,X) - \psi_1(x,X)}{h}\\ &\le \frac{x}{h_{\mathfrak{f}}^*(K)} + \frac{h}{2h_{\mathfrak{f}}^*(K)} + \frac{(x+h)^2}{hh_{\mathfrak{f}}^*(K)}W(x+h) + \frac{x^2}{hh_{\mathfrak{f}}^*(K)}W(x)\\ &\le \frac{x}{h_{\mathfrak{f}}^*(K)} + \frac{x}{h_{\mathfrak{f}}^*(K)}c_{22}(\log x)^{\frac{1}{2}}e^{-\frac{1}{2}c_{18}\sqrt{\log x}}(c_{25} + 5c_1)\\ &\le \frac{x}{h_{\mathfrak{f}}^*(K)} + \frac{c_3 x}{h_{\mathfrak{f}}^*(K)}(\log x)^{\frac{1}{2}}e^{-\frac{1}{2}c_{18}\sqrt{\log x}}\end{aligned}$$

where $c_{25} = \frac{1}{2c_{22}c_{26}}(\log x)^{-\frac{1}{2}} \le 0.001$, $c_{26} = \left(1 + \frac{\log\frac{3}{2}}{\log x}\right)^{\frac{1}{2}} \le 1.013$, $c_3 = c_{22}(c_{25} + 5c_{26})$. Putting $c_{18} = \frac{B\sqrt{58}}{A_0\sqrt{2r_2}} = 0.0919\sqrt{r_2}$ we obtain the result. This finishes the proof.

3 Proof of Lemma 1

The proof of Lemma 1 rests on the following lemmas. We first recall the well-known theorem of Phragmen-Lindelöf.

Lemma 9. *Let $f(s)$ be a regular function and $|f(s)| \le c_1 \exp(c_2|t|)$ in the region $\sigma_1 \le \sigma \le \sigma_2$, $-\infty < t < \infty$. Suppose further $|f(s)| \le M$ on the lines $\sigma = \sigma_1$ and $\sigma = \sigma_2$ of the complex plane. Then $|f(s)| \le M$ in the region $\sigma_1 \le \sigma \le \sigma_2$, $-\infty < t < \infty$.*

Proof. See [4, Lemma, p. 61]

Lemma 10. *Let $[K : \mathbb{Q}] = 2r_2$ and $0 < \eta \le \frac{1}{4}$. In the region $-\eta \le \sigma \le 3$ we have the estimate*

$$|\zeta(\sigma + it, \chi)| \le 1.4^{r_2}(1 + \varepsilon_\chi)A(\mathfrak{f})^{1+2\eta}\zeta(1+\eta)^{2r_2}(|t| + 4)^{r_2(1+2\eta)}$$

for any character $\chi \ne \chi_0$ modulo $\mathfrak{f}$, where $\varepsilon_\chi = 0$ or 1 to accordingly whether χ is primitive or not.

Proof. Consider

$$g(s,\chi) = \frac{\zeta(s,\chi)}{\zeta(1-s,\overline{\chi})},$$

where χ is a primitive character $\mod \mathfrak{f}$ and $\zeta(s,\chi)$ is defined in (1). From the functional equation for $\zeta(s,\chi)$ it follows that

$$g(s,\chi) = W(\chi)A(\mathfrak{f})^{1-2s}\left(\frac{\Gamma(1-s)}{\Gamma(s)}\right)^{r_2}, \tag{28}$$

where $A(\mathfrak{f})$ is defined in (2). We estimate $g(s,\chi)$ on the line $s=-\eta+it$, $0\le\eta\le\frac{1}{4}$ using the following inequality (see [4], p. 58)

$$\left|\frac{\Gamma(1-s)}{\Gamma(s)}\right|\le 1.4\max(1,|s|^{1+2\eta}). \tag{29}$$

From (28) and (29) we obtain

$$|g(-\eta+it,\chi)|\le 1.4^{r_2}\,A(\mathfrak{f})^{1+2\eta}(\max(1,|-\eta+it|^{(1+2\eta)}))^{r_2} \tag{30}$$

for $-\infty<t<\infty$. Write

$$G(s,\chi)=\frac{\zeta(s,\chi)}{(s+1)^{r_2(1+2\eta)}}. \tag{31}$$

From (30) we have

$$\begin{aligned}|G(-\eta+it,\chi)|&\le 1.4^{r_2}A(\mathfrak{f})^{1+2\eta}|\zeta(1+\eta-it,\overline{\chi})|\\&\le 1.4^{r_2}A(\mathfrak{f})^{1+2\eta}\zeta(1+\eta)^{2r_2}\end{aligned} \tag{32}$$

for $\chi\ne\chi_0$. If χ is not a primitive character, then there is an ideal $\mathfrak{f}_0$ which divides $\mathfrak{f}$, and there is a primitive character $\lambda\pmod{\mathfrak{f}_0}$ such that

$$\zeta(s,\chi)=\zeta(s,\lambda)\prod_{\mathfrak{p}|\mathfrak{f},\mathfrak{p}\nmid\mathfrak{f}_0}\left(1-\frac{\lambda(\mathfrak{p})}{(N\mathfrak{p})^s}\right).$$

Write $\mathfrak{f}=\mathfrak{f}_0\mathfrak{f}_1$. From [4, see p. 60] we get

$$\left|\prod_{\mathfrak{p}|\mathfrak{f},\mathfrak{p}\nmid\mathfrak{f}_0}\left(1-\frac{\lambda(\mathfrak{p})}{(N\mathfrak{p})^{\eta-it}}\right)\right|\le 2(N\mathfrak{f}_1)^{\frac{1}{2}+\eta}.$$

Hence,

$$\begin{aligned}|G(-\eta+it,\chi)|&\le 1.4^{r_2}(1+\varepsilon_\chi)A(\mathfrak{f})^{1+2\eta}|\zeta(1+\eta-it,\overline{\chi})|\\&\le 1.4^{r_2}(1+\varepsilon_\chi)\,A(\mathfrak{f})^{1+2\eta}\zeta(1+\eta)^{2r_2}\end{aligned} \tag{33}$$

for any character $\chi\ne\chi_0$ modulo $\mathfrak{f}$, where $\varepsilon_\chi=0$ or 1 to accordingly whether χ is primitive or not. On the other hand,

$$|G(3+it,\chi)|\le\frac{|\zeta(3+it,\overline{\chi})|}{(4+it)^{r_2(1+2\eta)}}\le\frac{1}{4^{r_2}}\zeta(3)^{2r_2}. \tag{34}$$

Using the estimate

$$|\zeta(s,\chi)|\le A_1e^{A_2|t|},$$

which is valid in the strip $-\eta\le\sigma\le 3$, where A_1,A_2 depends on K, χ, and $\mathfrak{f}$, we get

$$|G(s,\chi)|=O(e^{A_3|t|}) \tag{35}$$

for $-\eta \le \sigma \le 3$. From (33)–(35) and Lemma 9 we obtain

$$|G(s,\chi)| \le 1.4^{r_2}(1+\varepsilon_\chi)A(\mathfrak{f})^{1+2\eta}\zeta(1+\eta)^{2r_2} \tag{36}$$

in the strip $-\eta \le \sigma \le 3$. From (36), (31)

$$|\zeta(s,\chi)| \le 1.4^{r_2}(1+\varepsilon_\chi)A(\mathfrak{f})^{1+2\eta}\zeta(1+\eta)^{2r_2}(|t|+4)^{r_2(1+2\eta)}$$

for any character $\chi \ne \chi_0$ modulo $\mathfrak{f}$. This finishes the proof.

We denote by $\zeta_K(s)$ the Dedekind zeta-function.

Lemma 11. *For $\sigma > 1$ we have*

$$\frac{1}{|\zeta(\sigma+it,\chi)|} \le \zeta_K(\sigma).$$

Proof. See [9, Lemma 2.4].

Lemma 12. *Let $[K:\mathbb{Q}] = 2r_2$ and $0 < \eta \le \frac{1}{4}$. In the region $-\eta \le \sigma \le 1+\eta$, $-\infty < t < \infty$ we have estimate*

$$|(s-1)\zeta(s,\chi_0)| \le (3+|t|)(1+|t|)^{r_2(1+\eta-\sigma)}(|\Delta|N\mathfrak{f})^{\frac{1+\eta-\sigma}{2}}\zeta_K(1+\eta).$$

Proof. See [4, Eq. (5.4), p. 61].

Lemma 13. *Let $f(s)$ be a function regular in the disk $|s-s_0| \le r$ and satisfying the inequality*

$$\left|\frac{f(s)}{f(s_0)}\right| \le M.$$

If $f(s) \ne 0$ in the region $|s-s_0| \le \frac{r}{2}$, $\Re(s-s_0) > 0$, then

$$\Re\frac{f'}{f}(s_0) \ge -\frac{4}{r}\log M.$$

Proof. See [12, Satz 4.5, p. 384].

Lemma 14. *Let $f(s)$ be a function regular in the disk $|s-s_0| \le R$ and satisfying the conditions*

$$\Re f(s) \le M \qquad \text{for} \qquad |s-s_0| = R$$

Then

$$|f^{(k)}(s)| \le 2k!(M - \Re f(s_0))\frac{R}{(R-r)^{k+1}}, \qquad k \ge 1.$$

in the circle $|s-s_0| \le r < R$.

Proof. See [12, Satz 4.2, p. 383].

We are in a position to prove Lemma 1.

Proof. Let $B = \frac{A_1}{6} = 0.01325$, where A_1 is the constant appearing in (6). Let $s_0 = \sigma_0 + it_0$, $t_0 \geq 0$,

$$\sigma_0 = 1 + \frac{B}{L(t_0)}. \tag{37}$$

where $L(t_0)$ is defined in (5). We define the function

$$H(s,\chi) = \log \frac{g(s,\chi)}{g(s_0,\chi)}, \quad g(s,\chi) = h(s,\chi)\prod_{\rho}(s-\rho)^{-1},$$

where $h(s,\chi) = \zeta(s,\chi)$ if $\chi \neq \chi_0$ and $h(s,\chi_0) = (s-1)\zeta(s,\chi_0)$, where ρ are zeros of the function $h(s,\chi)$ in the disk $|s-s_0| \leq \frac{1}{2}$. Firstly, we estimate $\left|\frac{g(s,\chi)}{g(s_0,\chi)}\right|$. Lemmas 10, 11 and 12 shows that in the disk $|s-s_0| \leq 1$

$$\left|\frac{\zeta(\sigma+it,\chi)}{\zeta(s_0,\chi)}\right| \leq 1.4^{r_2}(1+\varepsilon_\chi)A(\mathfrak{f})^{1+2\eta}\zeta(1+\eta)^{2r_2}\zeta_K(\sigma_0)(t_0+4)^{r_2(1+2\eta)}, \tag{38}$$

for any character $\chi \neq \chi_0$ modulo $\mathfrak{f}$, where $\varepsilon_\chi = 0$ or 1 to accordingly whether χ is primitive or not, and

$$\left|\frac{\zeta(\sigma+it,\chi_0)(s-1)}{\zeta(s_0,\chi_0)(s_0-1)}\right| \tag{39}$$
$$\leq \frac{L(t_0)}{B}(4+|t_0|)(2+|t_0|)^{r_2(1+\eta)}(|\Delta|N\mathfrak{f})^{\frac{1+\eta}{2}}\zeta_K(1+\eta)\zeta_K(\sigma_0).$$

On the circle $|s-s_0| = 1$, $|s_0-\rho| \leq \frac{1}{2}$ and $|s-\rho| \geq \frac{1}{2}$. From (38), (39) and the maximum modulus principle we obtain

$$\left|\frac{\zeta(s,\chi)\prod_\rho(s_0-\rho)}{\zeta(s_0,\chi)\prod_\rho(s-\rho)}\right| \tag{40}$$
$$\leq 1.4(1+\varepsilon_\chi)A(\mathfrak{f})^{1+2\eta}\zeta(1+\eta)^{2r_2}\zeta_K(\sigma_0)(t_0+4)^{r_2(1+2\eta)},$$

and

$$\left|\frac{(s-1)\zeta(s,\chi_0)\prod_\rho(s_0-\rho)}{(s_0-1)\zeta(s_0,\chi_0)\prod_\rho(s-\rho)}\right| \leq \left|\frac{(s-1)\zeta(s,\chi_0)\prod_\rho(s_0-\rho)}{(\sigma_0-1)\zeta(s_0,\chi_0)\prod_\rho(s-\rho)}\right| \tag{41}$$
$$\leq \frac{L(t_0)}{B}(4+|t_0|)(2+|t_0|)^{r_2(1+\eta)}(|\Delta|N\mathfrak{f})^{\frac{1+\eta}{2}}\zeta_K(1+\eta)\zeta_K(\sigma_0)$$

in the disk $|s-s_0| \leq 1$. Secondly, we apply Lemma 14 to the function $H(s,\chi)$ with $k=1$, $R=\frac{1}{2}$ and $r=\frac{1}{4}$. The function $H(s,\chi)$ is regular in the disk $|s-s_0| \leq \frac{1}{2}$, so by (40), (41) we obtain

$$\Re H(s,\chi) = \log\left|\frac{g(s,\chi)}{g(s_0,\chi)}\right|$$
$$\leq \begin{cases} \log\left(1.4(1+\varepsilon_\chi)A(\mathfrak{f})^{1+2\eta}\zeta(1+\eta)^{2r_2}\zeta_K(\sigma_0)(t_0+4)^{r_2(1+2\eta)}\right), & \text{if } \chi \neq \chi_0 \\ \log\left(\frac{L(t_0)}{B}(4+|t_0|)(2+|t_0|)^{r_2(1+\eta)}(|\Delta|N\mathfrak{f})^{\frac{1+\eta}{2}}\zeta_K(1+\eta)\zeta_K(\sigma_0)\right), & \text{if } \chi = \chi_0. \end{cases}$$

in the disk $|s-s_0| \leq \frac{1}{2}$. Therefore, in the disk $|s-s_0| \leq \frac{1}{4}$ we have

$$\left|\frac{\zeta'}{\zeta}(s,\chi) - \sum_\rho \frac{1}{s-\rho}\right| \\ \leq 16\log\left(1.4(1+\varepsilon_\chi)A(\mathfrak{f})^{1+2\eta}\zeta(1+\eta)^{2r_2}\zeta_K(\sigma_0)(t_0+4)^{r_2(1+2\eta)}\right), \tag{42}$$

and

$$\left|\frac{\zeta'}{\zeta}(s,\chi_0) + \frac{1}{s-1} - \sum_\rho \frac{1}{s-\rho}\right| \\ \leq 16\log\left(\frac{L(t_0)}{B}(4+|t_0|)(2+|t_0|)^{r_2(1+\eta)}(|\Delta|N\mathfrak{f})^{\frac{1+\eta}{2}}\zeta_K(1+\eta)\zeta_K(\sigma_0)\right). \tag{43}$$

Finally, we estimate $|\sum_\rho \frac{1}{s_0-\rho}|$ and $|\sum_\rho \frac{1}{s-\rho}|$. In [11] Israilov shows that, if $1 < \sigma \leq 2$ then

$$-\frac{\zeta'}{\zeta}(\sigma) < \frac{1}{\sigma-1} - \gamma + C_1(\sigma-1),$$

where $C_1 = 0.1875463$. Hence, (37) shows

$$\left|\frac{\zeta'}{\zeta}(s_0,\chi)\right| \leq 2r_2\frac{L(t_0)}{B} + 2r_2C_1\frac{B}{L(t_0)}, \tag{44}$$

and

$$\left|\frac{\zeta'}{\zeta}(s_0,\chi_0) + \frac{1}{s_0-1}\right| \leq 4r_2\frac{L(t_0)}{B} + 2r_2C_1\frac{B}{L(t_0)}. \tag{45}$$

By (42), (44), we obtain

$$\left|\sum_\rho \frac{1}{s_0-\rho}\right| \leq 16\log\left(1.4(1+\varepsilon_\chi)A(\mathfrak{f})^{1+2\eta}\zeta(1+\eta)^{2r_2}\zeta_K(\sigma_0)(t_0+4)^{r_2(1+2\eta)})\right) \\ + 2r_2\frac{L(t_0)}{B} + 2r_2C_1\frac{B}{L(t_0)}, \tag{46}$$

and (43), (45)

$$\left|\sum_\rho \frac{1}{s_0-\rho}\right| \leq 16\log\left(\frac{L(t_0)}{B}(4+|t_0|)(2+|t_0|)^{r_2(1+\eta)}(|\Delta|N\mathfrak{f})^{\frac{1+\eta}{2}}\zeta_K(1+\eta)\zeta_K(\sigma_0)\right) \\ + 4r_2\frac{L(t_0)}{B} + 2r_2C_1\frac{B}{L(t_0)} \tag{47}$$

in the circle $|s - s_0| \leq \frac{1}{4}$. Now, we define

$$r_1 = \frac{2B}{L(t_0)} < \frac{1}{4}.$$

By Theorem 3, the function $\zeta(s,\chi) \neq 0$ in the region $|s - s_0| \leq r$, $\Re(s - s_0) > -2r_1$. Hence

$$|s_0 - \rho| \geq 2r_1, \quad |s - \rho| \geq \frac{1}{2}|s_0 - \rho|, \quad \Re(s_0 - \rho) \geq 2r_1$$

for all zeros ρ in the disk $|s - s_0| \leq \frac{1}{4}$, and for s in the disk $|s - s_0| \leq r_1$. For $|s - s_0| \leq r_1$ we obtain

$$\begin{aligned}\left|\sum_\rho \frac{1}{s-\rho} - \sum_\rho \frac{1}{s_0-\rho}\right| &\leq \sum_\rho \frac{|s-s_0|}{|s-\rho||s_0-\rho|} \leq \sum_\rho \frac{r_1}{\frac{1}{2}|s_0-\rho|^2} \\ &\leq \sum_\rho \frac{\Re(s_0-\rho)}{|s_0-\rho|^2} \leq \sum_\rho \Re\frac{1}{s_0-\rho} \leq \left|\sum_\rho \frac{1}{s_0-\rho}\right|.\end{aligned}$$

Thus,

$$\left|\sum_\rho \frac{1}{s-\rho}\right| \leq 2\left|\sum_\rho \frac{1}{s_0-\rho}\right|. \tag{48}$$

From (42), (46) and (48) we have

$$\begin{aligned}\left|\frac{\zeta'}{\zeta}(s,\chi)\right| &\leq 4r_2\frac{L(t_0)}{B} + 4r_2C_1\frac{B}{L(t_0)} \\ &\quad + 32\log\left(1.4(1+\varepsilon_\chi)A(\mathfrak{f})^{1+2\eta}\zeta(1+\eta)^{2r_2}\zeta_K(\sigma_0)(t_0+4)^{r_2(1+2\eta)}\right),\end{aligned}$$

and by (42), (47) and (48)

$$\begin{aligned}\left|\frac{\zeta'}{\zeta}(s,\chi) + \frac{1}{s-1}\right| &\leq 8r_2\frac{L(t_0)}{B} + 4r_2C_1\frac{B}{L(t_0)} \\ &\quad + 32\log\left(\frac{L(t_0)}{B}(4+|t_0|)(2+|t_0|)^{r_2(1+\eta)}(|\Delta|N\mathfrak{f})^{\frac{1+\eta}{2}}\zeta_K(1+\eta)\zeta_K(\sigma_0)\right)\end{aligned}$$

in the disk $|s - s_0| \leq r_1$, and consequently the above estimates hold in the strip

$$1 - \frac{B}{L(t)} = 1 - \frac{A_1}{6L(t)} < \sigma < 1 + \frac{3B}{L(t)} = 1 + \frac{A_1}{2L(t)}.$$

Now suppose that $s_0 = \sigma_0 + it_0$, $t_0 \geq 0$, where

$$1 + \frac{A_1}{2L(t_0)} \leq \sigma_0 \leq 3.$$

Lemma 13 and (38), (39) show that

$$-\frac{\zeta'}{\zeta}(\sigma_0,\chi) \le 4\log\Big(1.4(1+\varepsilon_\chi)A(\mathfrak{f})^{1+2\eta}\zeta(1+\eta)^{2r_2}\zeta_K(\sigma_0)(t_0+4)^{r_2(1+2\eta)}\Big)$$

for any character $\chi \neq \chi_0$ modulo $\mathfrak{f}$, where $\varepsilon_\chi = 0$ or 1 to accordingly whether χ is primitive or not, and

$$-\frac{\zeta'}{\zeta}(\sigma_0,\chi_0) \le \frac{1}{\sigma_0-1} + 4\log\left(\frac{L(t_0)}{B}(4+|t_0|)(2+|t_0|)^{r_2(1+\eta)}(|\Delta|N\mathfrak{f})^{\frac{1+\eta}{2}}\zeta_K(1+\eta)\zeta_K(\sigma_0)\right).$$

Therefore,

$$\left|\frac{\zeta'}{\zeta}(\sigma_0,\chi)\right| \le 4\log\Big(1.4(1+\varepsilon_\chi)A(\mathfrak{f})^{1+2\eta}\zeta(1+\eta)^{2r_2}\zeta_K(\sigma_0)(t_0+4)^{r_2(1+2\eta)}\Big)$$

for any character $\chi \neq \chi_0$ modulo $\mathfrak{f}$, where $\varepsilon_\chi = 0$ or 1 to accordingly whether χ is primitive or not, and

$$\left|\frac{\zeta'}{\zeta}(\sigma_0,\chi_0)\right| \le \frac{2L(t_0)}{A_1} + 4\log\left(\frac{L(t_0)}{B}(4+|t_0|)(2+|t_0|)^{r_2(1+\eta)}(|\Delta|N\mathfrak{f})^{\frac{1+\eta}{2}}\zeta_K(1+\eta)\zeta_K(\sigma_0)\right).$$

Write $f(s,\chi) = \zeta(s,\chi)$, $\chi \neq \chi_0$ and $f(s,\chi_0) = \zeta(s,\chi_0)(s-1)$. By the above we obtain

$$\left|\frac{f'}{f}(s_0,\chi)\right| = \left|\frac{\zeta'}{\zeta}(s_0,\chi)\right| \le \left|\frac{\zeta'}{\zeta}(\sigma_0,\chi)\right| \le 4\log\Big(1.4(1+\varepsilon_\chi)A(\mathfrak{f})^{1+2\eta}\zeta(1+\eta)^{2r_2}\zeta_K(\sigma_0)(t_0+4)^{r_2(1+2\eta)}\Big)$$

for any character $\chi \neq \chi_0$ modulo $\mathfrak{f}$, where $\varepsilon_\chi = 0$ or 1 to accordingly whether χ is primitive or not, and

$$\left|\frac{f'}{f}(s_0,\chi_0)\right| = \left|\frac{\zeta'}{\zeta}(s_0,\chi_0) + \frac{1}{s_0-1}\right| \le \left|\frac{\zeta'}{\zeta}(\sigma_0,\chi)\right| + \frac{1}{\sigma_0-1} \le \frac{4L(t_0)}{A_1} + 4\log\left(\frac{L(t_0)}{B}(4+|t_0|)(2+|t_0|)^{r_2(1+\eta)}(|\Delta|N\mathfrak{f})^{\frac{1+\eta}{2}}\zeta_K(1+\eta)\zeta_K(\sigma_0)\right).$$

The proof is completed by applying

$$\zeta_K(\sigma_0) \le \zeta(\sigma_0)^{2r_2} \le \left(1+\frac{6L(t_0)}{A_1}\right)^{2r_2}.$$

References

1. Atkin, A., Morain, F.: Elliptic curves and primality proving. Technical report Project ICSLA RR-1256, INRIA (1990)
2. Chatzigeorgiou, I.: Bounds on the Lambert function and their application to the outage analysis of user cooperation. IEEE Commun. Lett. **17**(8), 1505–1508 (2013)
3. Dupont, R., Enge, A., Morain, F.: Building curves with arbitrary small MOV degree over finite prime fields. J. Cryptol. **18**(2), 79–89 (2005)
4. Fryska, T.: An estimate of the order of the Hecke-Landau $\zeta(s, \chi)$-functions. Funct. Approx. Comment. Math. **16**, 55–62 (1988)
5. Fryska, T.: Some effective estimates for the roots of the Dirichlet L-series, II. Funct. Approx. Comment. Math. **16**, 21–36 (1988)
6. Grześkowiak, M.: An algorithmic construction of finite elliptic curves of order divisible by a large prime. Fund. Inform. **136**(4), 331–343 (2015)
7. Grześkowiak, M.: Algorithms for relatively cyclotomic primes. Fund. Inform. **125**(2), 161–181 (2013)
8. Grześkowiak, M.: Algorithms for pairing-friendly primes. In: Cao, Z., Zhang, F. (eds.) Pairing 2013. LNCS, vol. 8365, pp. 215–228. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-04873-4_13
9. Grześkowiak, M.: Explicit zero counting formula for Hecke-Landau zeta-functions. Bull. Aust. Math. Soc. **95**(3), 400–411 (2017)
10. Ingham, A.E.: The Distribution of Prime Numbers. Cambridge University Press, Cambridge (1932)
11. Israilov, M.: The Laurent expansion of the Riemann Zeta function. Trudy Mat. Inst. Steklov. **158**, 98–104 (1981)
12. Prachar, K.: Primzahlverteilung, Berlin (1957)
13. Radziejewski, M.: On the distribution of algebraic numbers with prescribed factorization properties. Acta Arith. **116**(2), 153–171 (2005)

Public-Key Cryptography

Short Solutions to Nonlinear Systems of Equations

Alan Szepieniec(✉) and Bart Preneel

imec-COSIC, KU Leuven, Leuven, Belgium
{alan.szepieniec,bart.preneel}@esat.kuleuven.be

Abstract. This paper presents a new hard problem for use in cryptography, called Short Solutions to Nonlinear Equations (SSNE). This problem generalizes the Multivariate Quadratic (MQ) problem by requiring the solution be short; as well as the Short Integer Solutions (SIS) problem by requiring the underlying system of equations be nonlinear. The joint requirement causes common solving strategies such as lattice reduction or Gröbner basis algorithms to fail, and as a result SSNE admits shorter representations of equally hard problems. We show that SSNE can be used as the basis for a provably secure hash function. Despite failing to find public key cryptosystems relying on SSNE, we remain hopeful about that possibility.

Keywords: Signature scheme · Hard problem · Post-quantum · MQ
SIS · SSNE · Hash function

1 Introduction

The widely deployed RSA and elliptic curve cryptosystems rely on the hardness of the integer factorization and discrete logarithm problems respectively, which are in fact easy to solve on quantum computers by means of Shor's algorithm [30]. These encryption and signature schemes will therefore become insecure once large enough quantum computers are built; and as a result we need to design, develop and deploy cryptography capable of resisting attacks by quantum computers, despite running on classical computers.

A number of hard problems have been proposed to replace integer factorization and discrete logarithms for precisely this purpose of offering *post-quantum* security. For instance, the problem of finding short vectors in high-dimensional lattices relates to normed linear algebra problems such as SIS [1] and LWE [28], which in turn generate many types of public key cryptosystems. Finding satisfying solutions to systems of multivariate quadratic (MQ) systems of equations seems to be hard even if the quadratic map embeds a secret trapdoor allowing only the secret-key holder to generate signatures [14]. Evaluating isogenies between elliptic curves is easy, but finding the isogeny from input and output images is hard; this enables a rather direct adaptation of the Diffie-Hellman key agreement protocol [20]. Even traditionally symmetric problems such as

J. Kaczorowski et al. (Eds.): NuTMiC 2017, LNCS 10737, pp. 71–90, 2018.
https://doi.org/10.1007/978-3-319-76620-1_5

hash function inversion have been used to generate stateless digital signature schemes [5]. However, in nearly all post-quantum cryptosystems to date, either the public key or else the ciphertext or signature is huge—measurable in tens of kilobytes if not megabytes[1]. In the interest of easing the transition away from the quantum-insecure but very low-bandwidth ECDSA, designing a post-quantum signature scheme with short signatures or ciphertexts *and* short public keys is a major open problem.

In this paper, we propose a new cryptographic problem called Short Solutions to Nonlinear Equations (SSNE) and argue that it is likely hard, even for quantum computers. Informally, our new hard problem asks to find a *short* solution to a system of *non-linear* multivariate polynomial equations, and thus generalizes both the Short Integer Solution (SIS) problem where the system is linear, and the Multivariate Quadratic (MQ) problem where the solution need not be short. Adopting both requirements renders standard attack strategies inapplicable or wildly inefficient.

Nevertheless, we show in Sect. 4 that it is possible to attack SSNE with limited success, in a way that improves over brute force search. We take this attack and its limitations into account and delineate a niche of parameter space in which brute force is the most efficient attack strategy. As a result, SSNE offers a denser encoding of computational hardness than either SIS or MQ, and if it is possible to design public key cryptosystems that rely on this hard problem, it holds promise of generating a smaller public keys, ciphertexts and signatures than their MQ and SIS counterparts without incurring a security cost.

While designing a public key cryptosystem on top of SSNE remains an open problem, designing a hash function whose security relies on SSNE does not, as this problem is solved in Sect. 5. This result does not merely serve to demonstrate design of cryptographic primitives in lieu of the comparably more difficult end-goal of designing public key functionalities; it has standalone value as well. From the point of view of provable security, very few hash functions come with a security proof showing that finding a solution implies solving a hard problem that is defined independently of the hash function itself. Therefore these not-provably-secure hash functions offer less assurance of security than provably secure hash functions whose underlying hard problems are studied independently. Moreover, it is prudent to diversify the hard problems upon which cryptographic primitives rely, in order to isolate the effects of cryptanalytic breakthroughs.

2 Preliminaries

Notation. We denote by $\mathbb{F}_q$ the finite field of q elements. The integer range $\{a, a+1, \ldots, b-1, b\}$ is denoted by $[a : b]$. Vectors are denoted in boldface, *e.g.*, $\mathbf{x}$ and matrices by capital letters, *e.g.*, A, with indexation starting at zero.

[1] The curious exception to this rule is the supersingular isogeny Diffie-Hellman key agreement scheme, but even so it does not seem possible to use this construction for small signature schemes.

The slice of A consisting of rows i—j and columns k—l is denoted by $A_{[i:j,k:l]}$, and we drop the, $k : l$ when slicing from a vector instead of a matrix.

Lattices. A *lattice* of dimension n and embedding degree m is a discrete n-dimensional subspace of $\mathbb{R}^m$; without loss of generality, we consider subspaces of $\mathbb{Z}^m$. Any such lattice $\mathcal{L}$ can be described as the set of integer combinations of a set of vectors $\mathbf{b_0}, \ldots, \mathbf{b_{n-1}} \in \mathbb{Z}^m$, which is called a *basis* for the lattice and is not unique for a given lattice. A lattice $\mathcal{L}$ is *q-ary* whenever membership of a point $\mathbf{p} \in \mathbb{Z}^m$ is decided by $\mathbf{p} \bmod q$, *i.e.*, with each component reduced modulo q.

The LLL algorithm [23] takes a matrix of integers $A \in \mathbb{Z}^{h \times w}$ whose rows span a lattice, and outputs another matrix $B \in \mathbb{Z}^{h \times w}$ whose rows span the same lattice but are much shorter in length. Without loss of generality we assume the LLL algorithm also outputs a unitary matrix U such that $UA = B$. The shortest basis vector produced by LLL when applied to a lattice spanned by h vectors of w elements, is bounded in length by

$$\|\mathbf{b_0}\|_2 \leq \left(\frac{4}{4\delta - 1}\right)^{(w-1)/4} \mathsf{det}(\mathcal{L})^{1/w}, \tag{1}$$

where $\frac{1}{4} < \delta \leq 1$ is the LLL parameter and where the *determinant of the lattice* is given by $\mathsf{det}(\mathcal{L}) = \mathsf{det}(AA^\mathsf{T})^{1/2} = \mathsf{det}(BB^\mathsf{T})^{1/2}$ if A and B have linearly independent rows.

In the case of q-ary matrices, a basis matrix can be obtained by adjoining the original basis matrix with qI. LLL will return a $(w + h) \times w$ matrix whose first w rows consist of all zeros. The determinant of q-ary lattices of this dimension is q^{w-h} with high probability [26], which means that the length of the shortest nonzero vector in the output of LLL is bounded by

$$\|\mathbf{b_0}\|_2 \leq \left(\frac{4}{4\delta - 1}\right)^{(w-1)/4} q^{(w-h)/w}. \tag{2}$$

The ith *successive minimum* $\lambda_i(\mathcal{L})$ of a lattice $\mathcal{L}$ is the smallest $\rho \in \mathbb{R}$ such that the hypersphere with radius ρ centered at the origin contains at least i independent lattice points. According to the m-dimensional ball argument of Micciancio and Regev [26], the first successive minimum of a random q-ary lattice of dimension h and embedding dimension w can be approximated by

$$\lambda_0(\mathcal{L}) \approx \sqrt{\frac{w}{2\pi e}}\, q^{(w-h)/w}. \tag{3}$$

3 Short Solutions to Nonlinear Equations

Our hard problem generalizes the Multivariate Quadratic (MQ) problem as well as the Short Integer Solution (SIS) problem. After presenting the definitions we consider some straightforward attacks. In the next section we consider a more sophisticated one.

MQ Problem. Given a quadratic map $\mathcal{P} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$ consisting of m polynomials in n variables of degree at most 2, find a vector $\mathbf{x} \in \mathbb{F}_q^n$ such that $\mathcal{P}(\mathbf{x}) = \mathbf{0}$.

The MQ problem is **NP**-hard in general as well as empirically hard on average whenever $m \approx n$. The best known attack is the hybrid attack [6], which consists of guessing some variables so as to overdetermine the system of equations and then solving it using a Gröbner basis type solver such as F_4 [16] or XL [13]. The reduced cost of solving the overdetermined system compensates for the increased cost of retrying a new guess whenever it leads to no solutions. The complexity of the optimal-trade-off hybrid attack approaches $2^{C_q n}$ as $n \gg q \rightarrow \infty$ with $C_q = \omega(1.38 - 0.44\,\omega \log_2 q)$ and where $\omega \geq 2$ is the exponent of matrix multiplication complexity [7]. However, when $q \gg n$, the cost of even one random guess beyond the number of variable-fixes that makes the system a determined one, dominates the attack complexity. In this case the complexity of a purely algebraic attack can be estimated using the *degree of regularity* D_{reg} of the system. For semi-regular quadratic systems [3,4] (which we assume random quadratic systems are), the degree of regularity is equal to the degree of the first term with a non-positive coefficient of the power series expansion of

$$\mathrm{HS}(z) = \frac{(1-z^2)^m}{(1-z)^n}. \tag{4}$$

At this point, the Gröbner basis computation using F_4 or XL boils down to performing sparse linear algebra in the Macaulay matrix whose polynomials have degree D_{reg}. The complexity of this task is $O\left(\binom{n+D_{\mathrm{reg}}+1}{D_{\mathrm{reg}}}^2\right)$ in terms of the number of finite field operations, which in turn are polynomial in $\log q$. In summary, the complexity of solving the MQ problem is *exponential* in $n \approx m$, but barely affected by q.

SIS Problem. Given a matrix $A \in \mathbb{F}_q^{n \times m}$ with $m > n$, find a nonzero vector $\mathbf{x} \in \mathbb{Z}^m \backslash \{\mathbf{0}\}$ such that $A\mathbf{x} = \mathbf{0} \bmod q$ and $\|\mathbf{x}\|_2 \leq \beta$.

While not **NP**-hard, SIS does offer an average-case to worst-case reduction: solving random SIS instances is at least as hard as solving the lattice-based Shortest Independent Vectors Problem (SIVP) up to an approximation factor of $\tilde{O}(\beta\sqrt{n})$ in the worst case [25]. The most performant attack on SIS is indeed running a lattice-reduction algorithm such as BKZ 2.0 [8] to find short vectors in the associated lattice which is spanned by the kernel vectors of A. The complexity of this task is captured by the *root Hermite factor* $\delta > 1$, which approaches 1 for more infeasible computations. For a given δ the optimal number of columns of A to take into account (*i.e.*, by setting the coefficients of $\mathbf{x}$ associated to the other columns to zero) is given by $m = \sqrt{n \log_2 q / \log_2 \delta}$. At this point the average length of the lattice points found is $2^{2\sqrt{n \log_2 q \log_2 \delta}}$ and cryptographic security requires β to be smaller than this number. Albrecht *et al.* estimate the complexity of obtaining lattice points of this quality as $0.009/\log_2^2 \delta + 4.1$ in terms of the base-2 logarithm of the number of time steps [2]. The key takeaway is that the complexity of SIS grows *exponentially* in m and n, but *polynomially* in q and β.

SSNE Problem (Short Solutions to Nonlinear Equations)**.** Given a map $\mathcal{P} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$ consisting of m polynomials in n variables over a prime field $\mathbb{F}_q$ and with $\mathsf{deg}(\mathcal{P}) \geq 2$, find a vector $\mathbf{x} \in \mathbb{Z}^n$ such that $\mathcal{P}(\mathbf{x}) = 0 \bmod q$ and $\|\mathbf{x}\|_2 \leq \beta$.

It is clear that the attack strategies that work for MQ and SIS do not apply out of the box to the SSNE problem. The random guess of the hybrid attack on MQ might fix the first few variables to small values, but offers no guarantee that an algebraic solution to the other variables is small. Alternatively, one can drop the random guess and compute a Gröbner basis for the under-determined system. Even if the resulting Gröbner basis consists of a reasonable number of polynomials of reasonable degrees, obtaining a short vector in the variety associated with the Gröbner basis seems like a hard problem in and of itself. Alternatively, one can linearize the system by introducing a new variable for every quadratic term and treat the resulting matrix of coefficients as the matrix of a SIS instance. However, in this case it is unclear how to find the correct length bound β as it now applies to a vector of quadratic monomials. Nevertheless, we now show under which conditions or adaptations an algebraic attack and attack based on lattice reduction are possible.

3.1 Algebraic Attack

The constraint $\|\mathbf{x}\|_2 \leq \beta$ can be formulated algebraically. Assume $\beta < q/2$, and let $b = \lfloor \beta \rfloor$. Then any solution $\mathbf{x}$ to the SSNE problem must consist of coefficients in $[-b : b]$. For any such coefficient x_i, the polynomial $\prod_{j=-b}^{b}(x_i - j)$ must evaluate to zero. Therefore, by appending these polynomials to $\mathcal{P}$, one obtains a less under-determined system and perhaps even a determined one. If that is the case, XL and F$_4$ will find a short solution; however, the Gröbner basis computation must reach degree $2b$ for the added polynomials to make a difference, and for sufficiently large β even this task is infeasible. It is possible to generalize this strategy so as to require that the sums-of-squares of all subsets of the coefficients of $\mathbf{x}$ are smaller than β. This method cannot work when $\beta > q$, but can be effective when β is small—say, a handful of bits.

Alternatively, it is possible to run down the unsigned bit expansion of every component of $\mathbf{x}$ and introduce a new variable $x_{i,j}$ for each bit and one for each component's sign s_i. This transformation adds n equations of the form $x_i = s_i \sum_{j=0}^{\lceil \mathsf{log}_2 q \rceil} 2^j x_{i,j}$, $n\lceil \mathsf{log}_2 q \rceil$ equations of the form $x_{i,j}(1 - x_{i,j}) = 0$, and n equations of the form $(s_i - 1)(s_i + 1) = 0$. The advantage of having access to this bit expansion is that the constraint $\|x\|_2 \leq \beta$ can now be expressed as $\lceil \mathsf{log}_2 q \rceil$ equations modulo q, *even when* $\beta > q$.

In both cases, the system of equations becomes infeasibly large whenever β grows, which is exactly the intended effect from a design perspective. Phrased in terms of the security parameter κ, we have

Design Principle 1: *β must be large:* $\mathsf{log}_2 \beta > \kappa$.

Note that β cannot be larger than $\sqrt{n}(q-1)/2$ because in that case *any* solution vector $\mathbf{x}$ satisfies the shortness criterion, which can therefore be forgotten at no cost in favor of a very fast algebraic solution. In fact, we want a random solution to the system of equations to satisfy $\|\mathbf{x}\|_2 \leq \beta$ with at most a negligible probability. Design principle 2 requires this probability to be at most $2^{-\kappa}$, where κ is the targeted security level.

Design Principle 2: *β must not be too large: $n\mathsf{log}_2 q \geq \kappa + n\mathsf{log}_2\beta$.*

3.2 Lattice Attack

In the relatively small dimensions considered for SSNE, basic lattice reduction algorithms such as LLL [23] manage to find the shortest vector in polynomial time with all but absolute certainty. Moreover, the nonlinear system $\mathcal{P}(\mathbf{x}) = \mathbf{0}$ can always[2] be represented as a linear system $P\bar{\mathbf{x}} = \mathbf{0}$, where P is the Macaulay matrix of $\mathcal{P}$ and $\bar{\mathbf{x}}$ is the vector of all monomials in $\mathbf{x}$ that appear in $\mathcal{P}$. If the solution $\mathbf{x}$ to $\mathcal{P}(\mathbf{x}) = \mathbf{0}$ is short enough, then its expansion into $\bar{\mathbf{x}}$ will also be a solution to $P\bar{\mathbf{x}} = \mathbf{0}$—and might be found quickly by lattice-reducing any basis for the kernel of P and weighting the columns as necessary.

In fact, the vector $\bar{\mathbf{x}}$ associated with a solution $\mathbf{x}$ to $\mathcal{P}(\mathbf{x}) = \mathbf{0}$ will *always* lie in the kernel of P, although not every kernel vector corresponds to a solution. Since $\bar{\mathbf{x}}$ is necessarily in the lattice spanned by the kernel vectors of P, the only way to hide it from lattice-reduction is to make it long—as long as random lattice vectors taken modulo q. The rationale behind the next design principle is to require that some of the quadratic monomials $\bar{\mathbf{x}}$ are of the order of magnitude of q (possibly after modular reduction).

Design Principle 3: $\mathbf{x}$ *must not be too small: $\mathsf{log}_2\|\mathbf{x}\|_2^2 \geq \mathsf{log}_2 q$.*

A straightforward attack strategy to cope with this design principle is to focus only on those columns of P that correspond to the monomials of degree 1 in $\bar{\mathbf{x}}$. Lattice reduction will then find short kernel vectors for this reduced matrix $\tilde{P}$. The attack runs through linear combinations of these small reduced kernel vectors until it finds a small linear combination $\mathbf{c}$ such that $\mathcal{P}(\mathbf{c}) = \mathbf{0}$. A rigorous argument counts the number of points in this lattice that have the correct length and then computes the proportion of them that solve $\mathcal{P}(\mathbf{x}) = \mathbf{0}$, and infers from this a success probability and hence a running time for the attack. A far simpler but heuristic argument pretends that the nonlinear monomials of $\bar{\mathbf{x}}$ multiply with their matching columns from P and thus generate a *uniformly random* offset vector $\mathbf{p}$. The attacker succeeds only when $\mathbf{p} + \tilde{P}\mathbf{x} = \mathbf{0}$, which can be engineered to occur with at most a negligible probability.

Design Principle 4: *The output space must be large enough: $m\mathsf{log}_2 q \geq \kappa$.*

Lattice-reduction has been used in the past to find small solutions to univariate and multivariate polynomial equations, for instance in the context of factoring RSA moduli $n = pq$ where some of the bits of p or q are known. These applications of LLL were first discovered by Coppersmith [9,10], and were then

[2] This assumes that $\mathcal{P}$ has no constant terms, but the same arguments apply with minor modifications even if it does.

expanded on by Howgrave-Graham [19], Jutla [21], Coron [11,12], and most recently by Ritzenhofen [29]. The common strategy behind all these attacks is to generate clever algebraic combinations of the polynomials but which must be linearly independent. LLL is run either on the resulting system's Macaulay matrix or on its kernel matrix to find either polynomial factors with small coefficients or else short roots. However, this family of methods is only effective when the targeted solution is short enough. In particular, if $X_i \in \mathbb{Z}$ is a bound on x_i, *i.e.*, $|x_i| \leq X_i$, then success is only guaranteed whenever for every term $t \in \mathbb{F}_q[\mathbf{x}]$ of every polynomial of $\mathcal{P}$ (interpreted as $t \in \mathbb{Z}[\mathbf{x}]$)

$$|t(X_1, \ldots, X_n)| < q. \tag{5}$$

This success criterion is inconsistent with design principle 3.

3.3 Additional Considerations

Note that the shortness constraint $\|\mathbf{x}\|_2 \leq \beta$ does not have to apply to all variables. Even requiring only $\sqrt{\sum_{i \in S} x_i^2} \leq \beta$ where the sum is taken only over a non-empty subset S of the variables suffices to capture the hardness of the problem. More generally, the problem can be defined with respect to any weight matrix $W \in \mathbb{Z}^{n \times n}$, namely by requiring that $\mathbf{x}^\mathsf{T} W \mathbf{x} \leq \beta^2$. Diagonalization of W leads to a partitioning of the variables into one set which should be short and one set whose length does not matter. Nevertheless, one should be careful to ensure that the number of short variables must be larger than the dimension of the variety. Otherwise the shortness constraint is no constraint at all because it is possible to guess the short variables and subsequently solve for the remaining variables using a Gröbner basis algorithm.

Design Principle 5. *There should be more small variables than the dimension of the variety:* $\mathsf{rank}(W + W^\mathsf{T}) > \mathsf{dim}\, V(\mathcal{P}) = n - m$.

Remark. The concise way to capture "the number of variables that must be small after optimal basis change" is indeed $\mathsf{rank}(W + W^\mathsf{T})$. To see this, observe that $\mathbf{x}^\mathsf{T} W \mathbf{x}$ is a quadratic form and therefore equal to $\mathbf{x}^\mathsf{T}(W + A)\mathbf{x}$ for any skew-symmetric matrix A (*i.e.*, square matrix such that $A^\mathsf{T} = -A$). Up to additions of skew-symmetric matrices and up to constant factors we have $W \equiv W + W^\mathsf{T}$. This latter form is preferred for diagonalization, which finds an invertible basis change S such that makes $S^\mathsf{T}(W + W^\mathsf{T})S$ diagonal. The zeros on this diagonal indicate the variables whose size is unconstrained. Moreover, the rank of $W + W^\mathsf{T}$ cannot change under left or right multiplication by invertible matrices such as S^T or S.

3.4 Estimating Hardness

The main selling point of the SSNE problem is that neither the algebraic solvers nor lattice-reduction algorithms seem to apply, and as a result of this immunity it admits a much conciser encapsulation of cryptographic hardness. In MQ problems, the hardness derives from the large number of variables and equations n and m, and is largely independent of the field size q. In SIS problems, the hardness derives mostly from the large lattice dimension n, although the field size q and length constraint β are not entirely independent. Since both Gröbner basis and lattice-reduction algorithms do not apply, the hardness of SSNE problems must be much more sensitive to the size of the search space than their MQ and SIS counterparts. In particular, this sensitivity allows designers to achieve the same best attack complexity while shrinking m and n in exchange for a larger q—a trade-off that makes perfect sense because in all cases the representation of a single problem instance is *linear* in $\mathsf{log}_2\, q$ and *polynomial* in m and n.

All five design principles, including design principle 6 which will be derived in Sect. 4, have a limited range of applicability. No known algorithm solves SSNE problems for which all six criteria are met, faster than the following brute force search does. In the most optimistic scenario, no such algorithm exists. We invite the academic community to find attacks on SSNE that outperform this brute force search. In Sect. 5 we propose a hash function whose security relies on the assumption that either such an algorithm does not exist or that if it does, it does not beat brute force by any significant margin.

A brute force strategy must only search across $\mathbb{F}_q^{n-m}$. Each guess of the first $n - m$ variables is followed by an algebraic solution to the remaining system of m equations in m variables. If m is not too large then the task of finding this solution algebraically is rather fast, and the complexity of this joint task is dominated by $O(q^{n-m})$. In quantum complexity, Grover's algorithm [18] offers the usual square root speed-up of $O(q^{(n-m)/2})$.

4 An Algebraic-Lattice Hybrid Attack

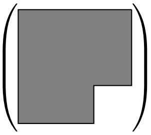
In this section we describe an attack that applies when $m(m+1)/2 \leq n$ and manages to produce somewhat short solutions. In a nutshell, the attack treats the polynomial system as a UOV$^-$ public key. A UOV reconciliation attack recovers the secret decomposition and at this point the attacker samples vinegar and oil variables such that the resulting "signature" is small. We consider the various steps separately. This section uses the terms "signature" and "solution" interchangeably because in the context of attacks on UOV they are identical.

4.1 UOV

Unbalanced Oil and Vinegar [22] is an MQ signature scheme with parameters $n = o + v$, $v \approx 2o$ and $m = o$. The public key is a homogeneous quadratic map $\mathcal{P} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$. The secret key is a decomposition of this public map into $\mathcal{F} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$ and $S \in \mathsf{GL}_n(\mathbb{F}_q)$ such that $\mathcal{P} = \mathcal{F} \circ S$. While S is a randomly chosen invertible matrix, $\mathcal{F}$ must have a special structure. All m components $f_i(\mathbf{x})$ partition the variables into two sets: vinegar variables $x_0, \ldots, x_{v-1}$, which are quadratically mixed with all other variables; and oil variables $x_v, \ldots, x_{n-1}$.

Visually, the matrix representations of these quadratic forms have an all-zero[3] $o \times o$ block:

$$f_i(\mathbf{x}) = \mathbf{x}^{\mathsf{T}}$$



$$\mathbf{x}. \qquad (6)$$

In order to compute a signature for a document $d \in \{0,1\}^*$, the signer computes its hash $\mathbf{y} = \mathsf{H}(d)$. He then chooses a random assignment to the vinegar variables and substitutes these into the system of equations $\mathcal{P}(\mathbf{x}) = \mathbf{y}$, or more explicitly

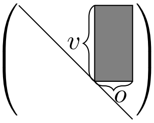
$$\begin{cases} \vdots \\ \sum_{j=0}^{v-1} \sum_{k=0}^{j} f_{j,k}^{(i)} \underline{x_j} \underline{x_k} + \sum_{j=0}^{v-1} \sum_{k=v}^{n-1} f_{j,k}^{(i)} \underline{x_j} x_k = y_i \ , \\ \vdots \end{cases} \qquad (7)$$

where $f_{j,k}^{(i)}$ represents the coefficient of the monomial $x_j x_k$ of the ith component of $\mathcal{F}$. The underlining indicates vinegar variables, which are substituted for their assignments. It is clear from this indication that the system of equations has become linear in the remaining oil variables, and since $m = o$, it has one easily computed solution in the generic case. The signer chooses a different assignment to the vinegar variables until there is one solution. At this point, the signature $\mathbf{s} \in \mathbb{F}_q^n$ is found by computing $\mathbf{s} = S^{-1}\mathbf{x}$. It is verified through evaluation of $\mathcal{P}$, *i.e.*, $\mathcal{P}(\mathbf{s}) \overset{?}{=} \mathsf{H}(d)$.

4.2 Reconciliation Attack

The reconciliation attack [15] is essentially an algebraic key recovery attack: the variables are the coefficients of S^{-1} and the equations are obtained by requiring that all the polynomials be of the same form as Eq. 6. Naïvely, this requires solving a quadratic system of $mo(o+1)$ equations in n^2 variables. However, the attack relies on the observation that there is almost always a viable S'^{-1} compatible with (6) but of the form

$$S'^{-1} =$$



$$. \qquad (8)$$

This observation is justified by the fact that only the coefficients of S^{-1} that are located in the rightmost o columns appear as indeterminates in the coefficients that are equated to zero. Moreover, any linear recombination of these columns also maps the oil-times-oil coefficients to zero and therefore we might as well consider only the representative of this equivalence class (equivalence under

[3] Or since it represents a quadratic form, skew-symmetric instead of all-zero.

linear recombination of the rightmost o columns) whose bottom right $o \times o$ block is the identity matrix.

The use of this observation reduces the number of variables to $v \times o$. Moreover, the key observation behind the reconciliation attack is that the o columns of S'^{-1} can be found iteratively, solving a new quadratic system at each step. Moreover, the authors of this attack argue that the complexity of this strategy is dominated by the first step, which requires solving only m equations in v variables [15].

These optimizations are no issue in our attack on SSNE. The parameters m and n are generally small enough to make naïvely solving a quadratic system of $mo(o+1)/2$ equations in n^2 variables feasible. However, for generic systems, whenever $mo(o+1)/2 > n^2$ there might not exist a $S^{-1} \in \mathsf{GL}_n(\mathbb{F}_q)$ that brings $\mathcal{P}$ into the form of Eq. 6. But choosing o to be different from m might bring a suitable S^{-1} back into existence. This motivates the following definition.

Definition 1 (o-reconcilable). *A system $\mathcal{P}$ of m multivariate quadratic polynomials in n variables over $\mathbb{F}_q$ is o-*reconcilable *iff there exists an $S \in \mathsf{GL}_n(\mathbb{F}_q)$ such that $\mathcal{P} \circ S$ partitions the n variables into $v = n - o$ vinegar variables and o oil variables distinguished by $\mathcal{P} \circ S$ being linear in the oil variables.*

Remark. Clearly, constant and linear terms are linear in all variables under any change of basis. Reconcilability considers only the quadratic part of the polynomials and without loss of generality we may restrict attention to their homogeneous quadratic part.

Theorem 1 (m-reconcilability of UOV). *Let $\mathcal{P} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$ be the public key of a UOV cryptosystem. Then $\mathcal{P}$ is m-reconcilable.*

Proof. Trivial: follows from construction of $\mathcal{P} = \mathcal{F} \circ S$. $\mathcal{F}$ induces the required partition into oil and vinegar variables. □

Theorem 2 ($\lfloor n/2 \rfloor$-reconcilability when $m = 1$). *Assume q is odd. Let $\mathcal{P} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q$ be a single quadratic polynomial. Then $\mathcal{P}$ is $\lfloor n/2 \rfloor$-reconcilable.*

Proof. Let $Q_p \in \mathbb{F}_q^{n \times n}$ be a symmetric matrix representation of $\mathcal{P}(\mathbf{x})$ via $\mathcal{P}(\mathbf{x}) = \mathbf{x}^\mathsf{T} Q_p \mathbf{x}$. Then Q_p is diagonalizable, *i.e.*, there exists an invertible matrix $A \in \mathbb{F}_q^{n \times n}$ such that $A^\mathsf{T} Q_p A$ is nonzero only on the diagonal.

All non-zero elements on the diagonal must be one except for the last which might be the smallest quadratic non-residue in $\mathbb{F}_q$. Now choose a random symmetric matrix $Q_f \in \mathbb{F}_q^{n \times n}$ such that the lower right $\lfloor n/2 \rfloor \times \lfloor n/2 \rfloor$ block consists of all zeros and such that $\mathsf{rank}(Q_f) = \mathsf{rank}(Q_p)$. It is also diagonalizable: there is an invertible matrix $B \in \mathbb{F}_q^{n \times n}$ such that $B^\mathsf{T} Q_f B$ is a diagonal matrix consisting of all ones except for the last element which might be the smallest quadratic non-residue. If $B^\mathsf{T} Q_f B = A^\mathsf{T} Q_p A$ we are done because $\mathcal{F} = \mathcal{P} \circ B^{-1} \circ A$ induces the required partition. If $B^\mathsf{T} Q_f B \neq A^\mathsf{T} Q_p A$ they must differ in the last diagonal element. So then multiply any one nonzero row of Q_f by any quadratic residue and obtain another diagonalization. Now $B^\mathsf{T} Q_f B = A^\mathsf{T} Q_p A$ must hold and we are done. □

Theorem 3. *In the generic case, a system of m quadratic polynomials in n variables over $\mathbb{F}_q$ is o-reconcilable when $m(o+1)/2 \leq n$.*

Proof. The number of coefficients of S^{-1} that are involved in the $mo(o+1)/2$ equations that set the oil-times-oil coefficients to zero is no, corresponding the rightmost $n \times o$ block of S^{-1}. The other elements of S^{-1} do not affect these coefficients. This leads to a system of $mo(o+1)/2$ quadratic equations in no variables which generically has solutions when $mo(o+1)/2 \leq no$, or equivalently when $m(o+1)/2 \leq n$. □

4.3 Generating Small Solutions

After obtaining an o-reconciliation $(\mathcal{F}, S)$, the task is to obtain a solution $\mathbf{x}$ such that $\mathcal{F}(\mathbf{x}) = \mathbf{0}$ and such that $S^{-1}\mathbf{x}$ is small. The partitioning of $\mathbf{x}$ into the vinegar variables $x_0, \ldots, x_{v-1}$ and the oil variables $x_v, \ldots, x_{n-1}$ separates the shortness objective into two parts. On the one hand, the *vinegar contribution*

$$\left(S^{-1}\right)_{[:,0:(v-1)]} \mathbf{x}_{[0:(v-1)]} \tag{9}$$

must be small; on the other hand, the *oil contribution*

$$\left(S^{-1}\right)_{[:,v:(n-1)]} \mathbf{x}_{[v:(n-1)]} \tag{10}$$

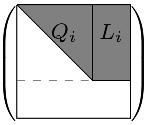
must be small as well. The reason for this separation is not just that the vinegar variables and oil variables are determined in separate steps; in fact, determining vinegar variables that lead to a small vinegar contribution is easy. The form of Eq. 8 guarantees that small vinegar variables will map onto a small vinegar contribution. Therefore, the only requirement for selecting vinegar variables is that they be small enough, say roughly $q^{1/2}$. By contrast, the process of finding suitable oil variables is far more involved.

A quadratic map where $o > m$ can be thought of as a UOV$^-$ map, *i.e.*, a UOV map with $o-m$ dropped components. This gives the signer, or an attacker who possesses the reconciliation, $o-m$ degrees of freedom for selecting the oil variables. Coupled with the freedom afforded by the choice of vinegar variables, the signer or attacker can generate a vector $\mathbf{x}$ such that $S^{-1}\mathbf{x}$ is short.

The task is thus to find an assignment to the oil variables such that (a) $\mathcal{F}(\mathbf{x}) = \mathbf{0}$ is satisfied; and (b) $\left(S^{-1}\right)_{[:,v:(n-1)]} \mathbf{x}_{v:(n-1)}$ is small as well. Constraint (a) is satisfiable whenever $m \leq o$ (in the generic case, *i.e.*, assuming certain square matrices over $\mathbb{F}_q$ are invertible). Constraint (b) requires $o > m$ and the resulting vector can be made shorter with growing $o - m$.

The matrix representation of a quadratic form is equivalent under addition of skew-symmetric matrices, which in particular means that it is always possible to choose an upper-triangular representation even of UOV maps such as Eq. 6. The ith equation of $\mathcal{F}(\mathbf{x}) = \mathbf{0}$ can therefore be described as

$$f_i(\mathbf{x}) = \mathbf{x}^\mathsf{T}$$



$$\mathbf{x} + \boldsymbol{\ell}^{(i)\mathsf{T}}\mathbf{x} + c_i = 0 \qquad (11)$$

$$\left(\mathbf{x}^\mathsf{T}_{[0:(v-1)]} L_i + \boldsymbol{\ell}^{(i)\mathsf{T}}_{[v:(n-1)]}\right) \mathbf{x}_{[v:(n-1)]} = -\mathbf{x}^\mathsf{T}_{[0:(v-1)]} Q_i \mathbf{x}_{[0:(v-1)]} - \boldsymbol{\ell}^{(i)\mathsf{T}}_{[0:(v-1)]} \mathbf{x}_{[0:(v-1)]} - c_i . \qquad (12)$$

All m equations can jointly be described as $A\mathbf{x}_{[v:(n-1)]} = \mathbf{b}$ for some matrix $A \in \mathbb{F}_q^{m \times o}$ and vector $\mathbf{b} \in \mathbb{F}_q^m$, because the vinegar variables $\mathbf{x}_{[0:(v-1)]}$ assume constant values. Let $\mathbf{x}^{(p)}$ be any particular solution to this linear system, and let $\mathbf{x}_0^{(k)}, \ldots, \mathbf{x}_{o-m-1}^{(k)}$ be a basis for the right kernel of A. Any weighted combination of the kernel vectors plus the particular solution, is still a solution to the linear system:

$$\forall (w_0, \ldots, w_{o-m-1}) \in \mathbb{F}_q^{o-m} \, . \, A \left(\mathbf{x}^{(p)} + \sum_{i=0}^{o-m-1} w_i \mathbf{x}_i^{(k)} \right) = \mathbf{b} . \qquad (13)$$

This means we have $o-m$ degrees of freedom with which to satisfy constraint (b).

In fact, we can use LLL for this purpose in a manner similar to the clever selection of the vinegar variables. The only difference is that the weight associated with the vector $\mathbf{x}^{(p)}$ must remain 1 because otherwise constraint (a) is not satisfied. This leads to the following application of the embedding method.

Identify $\mathbf{x}^{(p)}$ and all $\mathbf{x}_i^{(k)}$ by their image after multiplication by $\left(S^{-1}\right)_{[:,v:(n-1)]}$, thus obtaining $\mathbf{z}^{(p)} = \left(S^{-1}\right)_{[:,v:(n-1)]} \mathbf{x}^{(p)}$ and $\mathbf{z}_i^{(k)} = \left(S^{-1}\right)_{[:,v:(n-1)]} \mathbf{x}_i^{(k)}$. Then append q^2 to $\mathbf{z}^{(p)}$ and 0 to all $\mathbf{z}_i^{(k)}$, and stack all these vectors in column form over a diagonal of q's to obtain the matrix C:

$$C = \left(\begin{array}{ccc|c} — & \mathbf{z}^{(p)\mathsf{T}} & — & q^2 \\ — & \mathbf{z}_0^{(k)\mathsf{T}} & — & 0 \\ & \vdots & & \vdots \\ — & \mathbf{z}_{o-m-1}^{(k)\mathsf{T}} & — & 0 \\ \hline q & & & \\ & \ddots & & \\ & & q & \end{array} \right) . \qquad (14)$$

Run LLL on this matrix to obtain a reduced basis matrix $B \in \mathbb{Z}^{(o-m+1+n)\times(n+1)}$ of which the first n rows are zero, and a unimodular matrix U satisfying $B = UC$. The appended q^2 element guarantees that the row associated with the particular solution will never be added to another row because that would increase the size of the basis vectors. As a result, there will be one row in the matrix B that ends in q^2. Moreover, this row will be short because it was reduced by all other rows. We now proceed to derive an upper bound for the size of this vector considering only the first n elements, *i.e.*, without the q^2. Unfortunately, the best upper bound we can prove rigorously is $\lceil \frac{q}{2} \rceil \sqrt{n}$, but we can rely on the following heuristic argument for a meaningful result.

Let s be the index of this targeted row. Without row s and omitting the last column, the nonzero rows of B form an LLL-reduced basis for a q-ary lattice of dimension $o-m$ and embedding dimension n. We approximate the sizes of these vectors using $\lambda_i(\mathcal{L}) \approx \lambda_0(\mathcal{L})$. Coupled with the m-dimensional ball argument of Micciancio and Regev for estimating the first successive minimum [26], this gives

$$\|\mathbf{b}_\ell\|_2 \lesssim 2^{(o-m)/2} \sqrt{\frac{n}{2\pi e}} q^{(n-o+m)/n}. \tag{15}$$

Moreover, row s (considered without the q^2) cannot be much larger than this quantity because it is LLL-reduced with respect to vectors of this size. So $\|\mathbf{b}_s\|_2 \approx \|\mathbf{b}_\ell\|_2$. Our experiments show that this heuristic bound is followed quite closely in practice for small m, n and large q.

The solution $\mathbf{s} = S^{-1}\mathbf{x}$ consists of two parts: the vinegar contribution and the oil contribution. Therefore, we can bound the size of the whole thing.

$$\|\mathbf{s}\|_2 \leq \|S^{-1}_{[:,0:(v-1)]}\mathbf{x}_{[0:(v-1)]}\|_2 + \|S^{-1}_{[:,v:(n-1)]}\mathbf{x}_{[v:(n-1)]}\|_2 \tag{16}$$

$$\lesssim \sqrt{n-o} \cdot q^{1/2} + 2^{(o-m)/2} \sqrt{\frac{n}{2\pi e}} q^{(n-o+m)/n}. \tag{17}$$

Or if we treat n, m, o, v as small constants,

$$\|\mathbf{s}\|_2 \in O\left(q^{(n-o+m)/n}\right). \tag{18}$$

4.4 Summary

Figure 1 shows pseudocode for the algebraic-lattice hybrid attack algorithm.

Line 1 attempts to launch a UOV reconciliation attack, but the algorithm fails when this attack is unsuccessful. In fact, the criterion for success is precisely whether the map $\mathcal{P}$ is o-reconcilable. Generically, this criterion is only satisfied for $m(o+1)/2 \leq n$, as per Theorem 3, although it is certainly possible to construct maps that are o-reconcilable for $m(o+1)/2 > n$—indeed, standard UOV public keys match this ungeneric description. A prudent strategy for maps whose structure is unknown is to try step 1 for several values of o and to pick the decomposition of $\mathcal{P}$ where o is largest. However, in this case the length of the returned solution is not fixed beforehand but depends on the largest o for which step 1 succeeds.

With this algebraic-lattice hybrid attack in mind, we formulate the last design principle for SSNE instances. The rationale is that the targeted solution should be significantly smaller (*i.e.*, κ bits, spread over n variables) than what the algebraic-lattice hybrid attack can produce.

Design Principle 6: *Let o be the largest integer such that the system is o-reconcilable. If $o > m$ then guarantee that*

$$\frac{\kappa}{n} + \mathsf{log}_2\beta \leq \frac{n-o+m}{n+1} \mathsf{log}_2 q. \tag{19}$$

```
algorithm ALHA
input: $\mathcal{P} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$ — a quadratic map
     : $o \in \mathbb{Z}$ — number of oil variables
output: $\mathbf{s} \in \mathbb{F}_q^n$ such that $\mathcal{P}(\mathbf{s}) = \mathbf{0}$
        and such that $\|\mathbf{s}\|_2 \in O(q^{o/n} + q^{(n-o+m)/(n+1)})$
    ▷ find decomposition $\mathcal{P} = \mathcal{F} \circ S$ where $\mathcal{F}$ is quadratic but linear in
    $x_{n-o}, \ldots, x_{n-1}$, and where $S \in \mathsf{GL}_n(\mathbb{F}_q)$
    try: $\mathcal{F}, S \leftarrow$ UOV Reconciliation Attack$(\mathcal{P}, o)$

    ▷ get vinegar variables $x_0, \ldots, x_{n-o-1}$
    $\mathbf{x}_{[0:n-o-1]} \xleftarrow{\$} [-\lfloor q^{1/2} \rfloor : \lfloor q^{1/2} \rfloor]^{n-o}$

    ▷ get oil variables $x_{n-o}, \ldots, x_{n-1}$
    Find $A \in \mathbb{F}_q^{m \times o}$ and $\mathbf{b} \in \mathbb{F}_q^m$ such that $A\mathbf{x}_{[(n-o):(n-1)]} = \mathbf{b} \Leftrightarrow \mathcal{F}(\mathbf{x}) = \mathbf{0}$
    Find particular solution $\mathbf{x}^{(p)}$ to $A\mathbf{x}_{[(n-o):(n-1)]} = \mathbf{b}$
    Find kernel vectors $\mathbf{x}_0^{(k)}, \ldots, \mathbf{x}_{o-m-1}^{(k)}$ of $A$
    $\mathbf{z}^{(p)} \leftarrow \left(S^{-1}\right)_{[:,(n-o):(n-1)]} \mathbf{x}^{(p)}$
    for $i \in [0 : (o-m-1)]$ do:
        $\mathbf{z}_i^{(k)} \leftarrow \left(S^{-1}\right)_{[:,(n-o):(n-1)]} \mathbf{x}_i^{(k)}$
    end
    Compile matrix $C$ from $\mathbf{z}^{(p)}$ and $\mathbf{z}_i^{(k)}$          ▷ according to Eqn. 14
    $U, B \leftarrow \mathsf{LLL}(C)$
    Find $s$ such that $B_{[s,:]}$ ends in $q^2$
    $\mathbf{x}_{[(n-o):(n-1)]} \leftarrow \mathbf{x}^{(p)} + \sum_{i=0}^{o-m-1} U_{[s,1+i]} \mathbf{x}_i^{(k)}$

    ▷ join vinegar and oil variables, and find inverse under $S$
    $\mathbf{s} \leftarrow S^{-1}\mathbf{x}$
    return $\mathbf{s}$
```

Fig. 1. Algebraic-lattice hybrid attack.

4.5 Discussion

Equation 15 is an upper bound whereas we actually need a lower bound in order to delineate a portion of the parameter space where the attack does not apply. In practice, the short solutions found by the algebraic lattice hybrid attack are indeed shorter than the heuristic upper bound of Eq. 17. Nevertheless, the solutions found by the attack have length very close to this bound, to the point where it is a suitable estimate. Figure 2 plots in full blue the minimum length of solutions found by the algebraic lattice hybrid attack across one hundred trials for various modulus sizes. This graph follows the dashed green line, which represents the estimate or heuristic upper bound of Eq. 17, quite closely. Both are far apart from the recommendation of design principle 6, which is drawn in full red. This graph represents many random SSNE instances with $m = 2$ and $n = 9$. The same behavior was observed across a wide range of parameter choices.

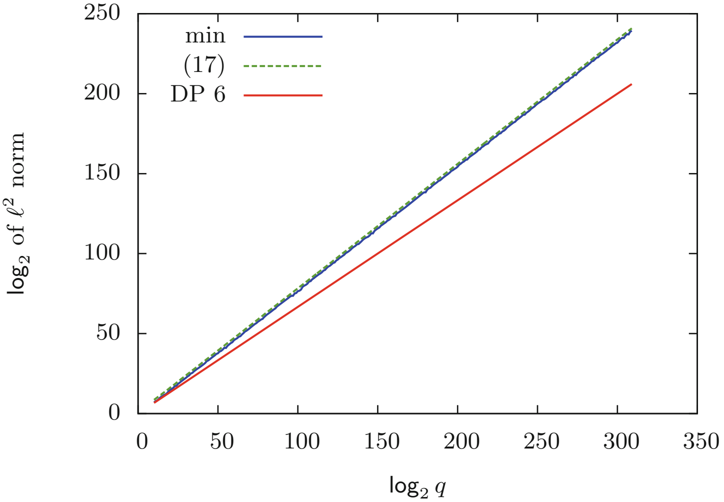


Fig. 2. Comparison of prediced length against experimental length of solutions obtained by the algebraic-lattice hybrid attack.

It is worth stressing that the algebraic-lattice hybrid attack applies only when $o > m$. When $o = m$ it does not produce solutions that are shorter than random vectors in $\mathbb{F}_q^n$, and when $o < m$ there is no guarantee it will find even one solution. Obviously, instead of requiring β to be significantly smaller than the expected length of this attack's solutions, the designer might also choose n and m so as to render the algebraic-lattice hybrid attack inapplicable.

5 Hash Function

At this time we do not know how to use SSNE to generate short-message public key functionalities. The next best option is to generate a hash function, which is what this section is about.

The resulting design does not merely exemplify using the SSNE problem constructively; it has concrete advantages over other hash functions as well. For instance, not only is the SSNE hash function provable secure (in contrast to all widely deployed hash functions), but it also relies on a *different* hard problem, which is likely to be unaffected by potential future breakthroughs in cryptanalysis of other hard problems. Also, our hash function has essentially optimal output length in terms of security: for κ bits of security against collision finders the output is 2κ bits long. This stands in contrast to many other provably secure hash functions which either have larger outputs or else require purpose-defeating post-processing functions to shrink them.

Additionally, because the hash function is built on top of SSNE instances, it requires comparably few finite field multiplications to compute. This property of having low multiplication complexity is interesting from the point of view of multiparty computation, zero-knowledge proofs, and fully homomorphic encryption, where multiplication operations are typically so expensive as to compel minimization at all costs. However, this argument ignores the cost of the bit shuffling, which are nonlinear operations over the finite field.

We note that it is possible to generate digital signature schemes from just hash functions [5,17], although the size and generation time of the signatures scales poorly. Nevertheless, anyone wanting to implement this signature scheme's key generation or signature generation procedures in a distributed manner—for instance, in order to require majority participation—must develop applied multiparty computation protocols and must consequently look to minimize multiplication complexity. Therefore, the SSNE hash function might be a good candidate for instantiating hash-based digital signature schemes with if they must enable distributed key and signature generation.

5.1 Description

We use the Merkle-Damgård construction, which requires dividing the data stream into a sequence of size b blocks. At every iteration, one data block is consumed and it is compressed with the state in order to produce a new state. The hash value is the output of the compression function after the last block has been consumed. The concept is described visually in Fig. 3.

Before applying the sequence of compression functions, the data stream $x \in \{0,1\}^*$ must first be expanded into a multiple of b bits. Let $\ell = |x|$ be the number of bits before padding, and let $\llcorner \ell \lrcorner$ be its expansion and $|\ell|$ the number of bits in this expansion. The expansion function is given by

$$\mathsf{expand} : \{0,1\}^\ell \rightarrow \{0,1\}^{\lceil(\ell+|\ell|)/b\rceil b} = x \mapsto x \| 0^{-\ell \,\mathsf{mod}\, b} \| 0^{-|\ell| \,\mathsf{mod}\, b} \| \llcorner \ell \lrcorner . \tag{20}$$

Let q be the largest prime smaller than $2^{2\kappa}$, where κ is the targeted security level. For the purpose of defining this hash function, the elements of $\mathbb{F}_q$ are $\{0, \ldots, q-1\}$. The compression function itself decomposes into $f = \mathcal{P} \circ r$. The purpose of $r : \{0,1\}^b \times \mathbb{F}_q \rightarrow \mathbb{F}_q^2$ is to permute the bits and output two integers inside $[0 : \lceil q^{3/4} \rceil - 1]$, which are then interpreted as small elements of $\mathbb{F}_q$. In particular, on input $(s, e) \in \{0,1\}^b \times \mathbb{F}_q$, this reshuffling function takes the most

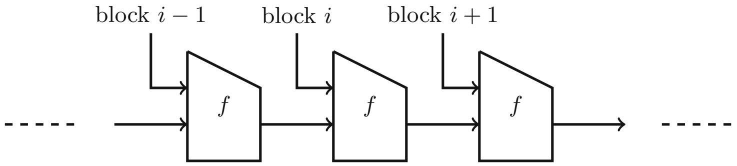


Fig. 3. Merkle-Damgård construction for hash functions.

```
algorithm Hash
input: $x \in \{0,1\}^\ell$ — bitstring of any length
output: $h \in \{0,1\}^{2\kappa}$ — hash value

$h \leftarrow \lfloor (\pi^{-1} - \frac{1}{4}) 2^{2\kappa+2} \rfloor$
$x' \leftarrow \mathsf{expand}(x)$
for $i \in [0 : |x'|/b]$ do:
    $e_1, e_2 \leftarrow r(x'_{[ib:(ib+b-1)]}, h)$
    $h \leftarrow \mathcal{P}(e_1, e_2)$
end
return $\llcorner h \lrcorner$
```

Fig. 4. Hash function relying on SSNE.

significant $\frac{1}{4}\lceil \log_2 q \rceil$ bits of e, appends them to s, and reinterprets this bitstring as an integer. Formally, r maps

$$r : \left(s_{b-1} \| \cdots \| s_0, \sum_{i=0}^{\lceil \log_2 q \rceil - 1} 2^i e_i \right) \mapsto \left(\left(\sum_{i=0}^{b-1} 2^i s_i \right) + \left(\sum_{i=b}^{\lceil \frac{3}{4} \log_2 q \rceil - 1} 2^i e_{i+b/2} \right), \sum_{i=0}^{\lceil \frac{3}{4} \log_2 q \rceil - 1} 2^i e_i \right) . \tag{21}$$

In particular, this implies that $b = \frac{1}{2}\lceil \log_2 q \rceil$.

The map $\mathcal{P} : \mathbb{F}_q^2 \rightarrow \mathbb{F}_q$ is a single homogeneous cubic polynomial in two variables. There are $\binom{5}{2} = 10$ coefficients which are assigned indices lexicographically from 0 to 9. Then the ith coefficient has a bit expansion equal to the first 2κ bits in the expansion of π^{i+1}, without the leading 1.

The description of the hash function is complete except for one remaining item. The initial state element, *i.e.*, the field element that is fed into the very first compression function must still be specified. For this value we choose the first 2κ bits of π^{-1}, again without the leading 1. The formal description of the algorithm is given in Fig. 4.

5.2 Security

The key property a hash function should possess is collision-resistance, which informally states that it should be difficult to find two different inputs $x, y \in \{0,1\}$ such that $\mathsf{Hash}(x) = \mathsf{Hash}(y)$. Collision-resistance implies weaker properties such as second preimage resistance and first preimage resistance (also known as one-wayness). Therefore, it suffices to show that collisions are hard to find. We demonstrate this fact by showing that any pair of colliding values implies a collision for $\mathcal{P}$, which should be difficult to find because that task requires solving a hard SSNE instance.

First, consider that expand is injective. To see this, assume there are two different strings x and y that have the same output under expand. Then $|x| \neq |y|$ because otherwise the appended tail is the same and then the difference must be present in their images under expand as well. However, the last b bits of the images under expand uniquely determine the length of the original strings and this quantity must be the same, which contradicts $|x| \neq |y|$. This argument assumes the length of the inputs is less than $2^b = 2^\kappa$, which is reasonable from a practical point of view. Since expand is injective, it cannot be the source of a collision.

Next, the permutation of bits r is a bijection. It cannot be the source of a collision either.

Therefore, the only source of collisions contained in the description of the hash function is $\mathcal{P}$. Finding a collision means finding a pair of vectors $\mathbf{a}, \mathbf{b} \in \mathbb{F}_q^2$ whose elements have at most $\frac{6}{4}\kappa$ bits, such that $\mathcal{P}(\mathbf{a}) = \mathcal{P}(\mathbf{b})$. One can re-write this equation in terms of the difference $\mathbf{d}$ from the mean $\mathbf{c} = (\mathbf{a} + \mathbf{b})/2$. The equation then becomes

$$\mathcal{P}(\mathbf{c} + \mathbf{d}) - \mathcal{P}(\mathbf{c} - \mathbf{d}) = \mathbf{0}. \tag{22}$$

This expression is useful because its degree in $\mathbf{c}$ is one less, *i.e.*, 2 instead of 3. Therefore, by choosing a random value for $\mathbf{d}$ the attacker finds $\mathbf{c}$ by solving a *quadratic*, instead of *cubic*, SSNE instance. (In fact, this argument was precisely the motivation for a degree-3 polynomial map $\mathcal{P}$ to begin with; to kill an attack strategy that involves only finding short solutions to *linear* equations.) The parameters of the hash function were chosen to ensure that the SSNE instance of Eq. 22 (with randomly chosen $\mathbf{d}$) satisfies all design principles.

6 Conclusion

This paper presents a new hard problem called SSNE, which is the logical merger of the SIS and MQ problems. However, in contrast to both the SIS and MQ problems, the hardness of an SSNE instance grows linearly with the size of the modulus q. This linear scaling stands in stark contrast to the quadratic and cubic scaling of the SIS and MQ problems; and therefore, if it is possible to generate post-quantum public key cryptosystems from SSNE as it is from SIS and MQ, then these cryptosystems are very likely to require dramatically less bandwidth for having smaller public keys, ciphertexts, or signatures.

Indeed, the goal of the research that lead to the writing of this paper was to generate *public key* cryptosystems with exactly those properties. Needless to say, we have failed in that endeavor. Some of the design principles came about as a result of a process of design and attack. At least from a superficial point of view, this failure suggests that the design principles are incompatible with strategies for generating public key cryptosystems. Nevertheless, we remain hopeful about the possibility of finding strategies that are compatible with the design principles and leave their discovery as an open problem.

Acknowledgments. The authors would like to thank Fré Vercauteren and Wouter Castryck for useful discussions and references, as well as the anonymous reviewers for helpful comments. Alan Szepieniec is being supported by a Ph.D. grant from the Institute for the Promotion of Innovation through Science and Technology in Flanders

(IWT-Vlaanderen). This work was supported in part by the Research Council KU Leuven: C16/15/058. In addition, this work was supported by the European Commission through the Horizon 2020 research and innovation programme under grant agreement No. H2020-ICT-2014-644371 WITDOM and H2020-ICT-2014-645622 PQCRYPTO.

References

1. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: Miller [27] , pp. 99–108. http://doi.acm.org/10.1145/237814.237838
2. Albrecht, M.R., Cid, C., Faugère, J., Fitzpatrick, R., Perret, L.: On the complexity of the BKW algorithm on LWE. Des. Codes Crypt. **74**(2), 325–354 (2015). https://doi.org/10.1007/s10623-013-9864-x
3. Bardet, M.: Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie. Ph.D. thesis, Pierre and Marie Curie University, Paris, France (2004). https://tel.archives-ouvertes.fr/tel-00449609
4. Bardet, M., Faugere, J.C., Salvy, B.: On the complexity of gröbner basis computation of semi-regular overdetermined algebraic equations. In: Proceedings of the International Conference on Polynomial System Solving, pp. 71–74 (2004)
5. Bernstein, D.J., Hopwood, D., Hülsing, A., Lange, T., Niederhagen, R., Papachristodoulou, L., Schneider, M., Schwabe, P., Wilcox-O'Hearn, Z.: SPHINCS: practical stateless hash-based signatures. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 368–397. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-5_15
6. Bettale, L., Faugère, J., Perret, L.: Hybrid approach for solving multivariate systems over finite fields. J. Math. Crypt. **3**(3), 177–197 (2009). https://doi.org/10.1515/JMC.2009.009
7. Bettale, L., Faugère, J., Perret, L.: Solving polynomial systems over finite fields: improved analysis of the hybrid approach. In: van der Hoeven, J., van Hoeij, M. (eds.) International Symposium on Symbolic and Algebraic Computation, ISSAC 2012, Grenoble, France, 22–25 July 2012, pp. 67–74. ACM (2012). http://doi.acm.org/10.1145/2442829.2442843
8. Chen, Y., Nguyen, P.Q.: BKZ 2.0: better lattice security estimates. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 1–20. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_1
9. Coppersmith, D.: Finding a small root of a bivariate integer equation; factoring with high bits known. In: Maurer [24], pp. 178–189. https://doi.org/10.1007/3-540-68339-9_16
10. Coppersmith, D.: Finding a small root of a univariate modular equation. In: Maurer [25], pp. 155–165. https://doi.org/10.1007/3-540-68339-9_14
11. Coron, J.-S.: Finding small roots of bivariate integer polynomial equations revisited. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 492–505. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_29
12. Coron, J.-S.: Finding small roots of bivariate integer polynomial equations: a direct approach. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 379–394. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74143-5_21
13. Courtois, N., Klimov, A., Patarin, J., Shamir, A.: Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 392–407. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_27

14. Ding, J., Yang, B.Y.: Multivariate public key cryptography. In: Bernstein, D.J., Buchmann, J., Dahmen, E. (eds.) Post-Quantum Cryptography, pp. 193–241. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-88702-7_6
15. Ding, J., Yang, B.-Y., Chen, C.-H.O., Chen, M.-S., Cheng, C.-M.: New differential-algebraic attacks and reparametrization of rainbow. In: Bellovin, S.M., Gennaro, R., Keromytis, A., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 242–257. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68914-0_15
16. Faugere, J.C.: A new efficient algorithm for computing Gröbner bases (F4). J. Pure Appl. Algebra **139**(1), 61–88 (1999)
17. Goldreich, O.: The Foundations of Cryptography: Basic Applications, vol. 2. Cambridge University Press, Cambridge (2004)
18. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Miller [28], pp. 212–219. http://doi.acm.org/10.1145/237814.237866
19. Howgrave-Graham, N.: Finding small roots of univariate modular equations revisited. In: Darnell, M. (ed.) Cryptography and Coding 1997. LNCS, vol. 1355, pp. 131–142. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0024458
20. Jao, D., De Feo, L.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: Yang, B.-Y. (ed.) PQCrypto 2011. LNCS, vol. 7071, pp. 19–34. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25405-5_2
21. Jutla, C.S.: On finding small solutions of modular multivariate polynomial equations. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 158–170. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0054124
22. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 206–222. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_15
23. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann. **261**(4), 515–534 (1982). https://doi.org/10.1007/BF01457454
24. Maurer, U. (ed.): EUROCRYPT 1996. LNCS, vol. 1070. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68339-9
25. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on Gaussian measures. SIAM J. Comput. **37**(1), 267–302 (2007). https://doi.org/10.1137/S0097539705447360
26. Micciancio, D., Regev, O.: Lattice-based cryptography. In: Bernstein, D.J., Buchmann, J., Dahmen, E. (eds.) Post-Quantum Cryptography, pp. 147–191. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-88702-7_5
27. Miller, G.L. (ed.): Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, 22–24 May 1996. ACM (1996)
28. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, 22–24 May 2005, pp. 84–93. ACM (2005). http://doi.acm.org/10.1145/1060590.1060603
29. Ritzenhofen, M.: On efficiently calculating small solutions of systems of polynomial equations: lattice-based methods and applications to cryptography. Ph.D. thesis, Ruhr University Bochum (2010). http://www-brs.ub.ruhr-uni-bochum.de/netahtml/HSS/Diss/RitzenhofenMaike/
30. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: 35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20–22 November 1994, pp. 124–134. IEEE Computer Society (1994). http://dx.doi.org/10.1109/SFCS.1994.365700

A Novel RSA-Like Cryptosystem Based on a Generalization of the Rédei Rational Functions

Nadir Murru(✉) and Francesco M. Saettone

Department of Mathematics "G. Peano", University of Turin,
Via Carlo Alberto 10, 10122 Torino, Italy
nadir.murru@unito.it, francesco.saettone@edu.unito.it

Abstract. In this paper we present a novel RSA-like cryptosystem. Specifically, we define a novel product that arises from a cubic field connected to the cubic Pell equation. We discuss some interesting properties and remarks about this product that can also be evaluated through a generalization of the Rédei rational functions. We then exploit these results to construct a novel RSA-like scheme that is more secure than RSA in broadcast applications. Moreover, our scheme is robust against the Wiener attack and against other kind of attacks that exploit the knowledge of a linear relation occurring between two plaintexts.

Keywords: Cubic Pell equation · Public cryptography
Rédei function · RSA

1 Introduction

RSA cryptosystem is one of the most famous public key scheme and is based on the existence of an one-way trapdoor function, which is easy to compute and difficult to invert without knowing some information. However, some attacks are possible when, e.g., the private key is small [23] or the public key is small [5]. Further attacks have been reviewed in [11] exploiting possible extra information (such as the knowledge of linear relations occurring between two plaintexts). Moreover, RSA leaks some vulnerabilities in broadcast applications [9]. Hence, during the years, RSA-like schemes (see, e.g., [2,6,13,15,17]) have been proposed in order to overcome some of the previous vulnerabilities.

In this paper, we present a novel RSA-like scheme that is more secure than RSA in some of the previous situations, like broadcast scenarios or considering the Wiener attack and others. Our scheme is based on a particular group equipped with a non-standard product that we have found working on a cubic field related to the cubic Pell equation (which is a generalization of the Pell equation, one of the most famous equations in number theory). This group appears to have many interesting properties and connections that should be further investigated. In fact, we would like to point out that in this work we give a first idea

J. Kaczorowski et al. (Eds.): NuTMiC 2017, LNCS 10737, pp. 91–103, 2018.
https://doi.org/10.1007/978-3-319-76620-1_6

about the potentiality of this group in cryptographic applications, with the aim of providing an original point of view for exploiting number theory in cryptography and opening new studies. Certainly, our scheme should be more investigated under several perspectives, such as its efficiency. However, it appears very promising due to the definition itself and the many properties and connections to different topics.

The paper is structured as follows. In Sect. 2, we introduce a group with a non-standard product starting from a cubic field. Section 3 is devoted to the presentation of our cryptosystem and its discussion. Moreover, we see that powers with respect to our product can be evaluated by means of a generalization of the Rédei rational functions (Rédei rational functions are classical and very interesting functions in number theory). In Sect. 4 we present the conclusion.

2 A Product Related to the Cubic Pell Equation

The Pell equation $x^2 - dy^2 = 1$, for d positive integer non-square and x, y unknowns, is one of the most famous Diophantine equations. Its generalization to the cubic case is given by the following equation:

$$x^3 + ry^3 + r^2z^3 - 3rxyz = 1 \tag{1}$$

where r is a given non-cubic integer and x, y, z unknown numbers whose values we are seeking over the integers. This equation is considered the more natural generalization of the Pell equation, since it arises considering the unitary elements of a cubic field as well as the Pell equation can be introduced considering unitary elements of a quadratic field. Specifically, let $(\mathbb{F}, +, \cdot)$ be a field and $t^3 - r$ an irreducible polynomial in $\mathbb{F}[t]$. Let us consider the quotient field $\mathbb{A} = \mathbb{F}[t]/(t^3 - r) = \{x + yt + zt^2 : x, y, z \in \mathbb{F}\}$. The quotient field $\mathbb{A}$ naturally induces a product between triples of elements of $\mathbb{F}$ as follows:

$$(x_1, y_1, z_1) \bullet (x_2, y_2, z_2) := (x_1x_2 + (y_2z_1 + y_1z_2)r, x_2y_1 + x_1y_2 + rz_1z_2, y_1y_2 + x_2z_1 + x_1z_2)$$

for $(x_1, y_1, z_1), (x_2, y_2, z_2) \in \mathbb{F}^3$ and the norm of an element is given by

$$N(x, y, z) := x^3 + ry^3 + r^2z^3 - 3rxyz,$$

see, e.g., [1], p. 98. Considering the unitary elements we get the cubic Pell curve

$$\mathcal{C} = \{(x, y, z) \in \mathbb{F}^3 : x^3 + ry^3 + r^2z^3 - 3rxyz = 1\}.$$

In [4], Christofferson widely studied the more general equation

$$x^3 + rb^2y^3 + r^2bz^3 - 3rbxyz = c,$$

whose the cubic Pell equation is a particular case for $b = c = 1$ and r not a cube, providing also a complete bibliography up to 1956. It is worth noting that it is still lacking an algorithm for generating the solutions of such an equation (for any value of r) similar to that for the quadratic Pell case (see, e.g., [1]).

Proposition 1. *$(\mathcal{C}, \bullet)$ is a commutative group with identity $(1, 0, 0)$ and the inverse of an element (x, y, z) is*

$$(\bar{x}, \bar{y}, \bar{z}) := (-x + ryz, rz^2 - xy, y^2 - xz).$$

Proof. The proof is straightforward and is left to the reader.

Remark 1. In $\mathbb{F}^3$ an element (x, y, z) is invertible with respect to $\bullet$ if and only if $N(x, y, z) \neq 0$ and its inverse is

$$\left(\frac{\bar{x}}{N(x, y, z)}, \frac{\bar{y}}{N(x, y, z)}, \frac{\bar{z}}{N(x, y, z)} \right).$$

Remark 2. When $\mathbb{F} = \mathbb{R}$, the cubic Pell curve $\mathcal{C}$ contains the solutions of the cubic Pell equation.

Remark 3. The Pell equation can be introduced considering the unitary elements of $\mathbb{R}[t]/(t^2 - d)$, d positive integer non-square, where the product between elements is

$$(x_1, y_1)(x_2, y_2) = (x_1 x_2 + d y_1 y_2, x_1 y_2 + y_1 x_2).$$

Starting from $\mathbb{A}$, we can introduce a new group with a non-standard product having interesting properties that can be also exploited for creating a novel RSA-like cryptosystem. Let us consider the quotient group $B := \mathbb{A}^*/\mathbb{F}^*$. An element in B is the equivalence class of elements in $\mathbb{A}^*$, i.e., $[m + nt + pt^2] \in B$ is the equivalence class of $m + nt + pt^2 \in \mathbb{A}^*$ defined by

$$[m + nt + pt^2] := \{\lambda m + \lambda nt + \lambda pt^2 : \lambda \in \mathbb{F}^*\}.$$

We can now rewrite the elements of B. Given $m + nt + pt^2 \in \mathbb{A}^*$, if $m \neq 0$ and $n = p = 0$, then

$$[m + nt + pt^2] = [m] = [1_{\mathbb{F}^*}].$$

If $n \neq 0$ and $p = 0$, then

$$[m + nt + pt^2] = [m + nt] = [m + t].$$

Finally, if $p \neq 0$, then

$$[m + nt + pt^2] = [m + nt + t^2].$$

Thus, the group B is

$$B = \{[m + nt + t^2] : m, n \in \mathbb{F}\} \cup \{[m + t] : m \in \mathbb{F}\} \cup \{[1_{\mathbb{F}^*}]\}.$$

Now, we can write the elements of B with a new notation. Fixed an element $\alpha \notin \mathbb{F}$, the elements of B can be written as couples of the kind (m, n), with $m, n \in \mathbb{F}$, or (m, α), with $m \in \mathbb{F}$, or (α, α). Hence the group B is

$$B = (\mathbb{F} \times \mathbb{F}) \cup (\mathbb{F} \times \{\alpha\}) \cup (\{\alpha\} \times \{\alpha\}).$$

With this new notation and remembering that $\mathbb{A} = \mathbb{F}[x]/(t^3 - r)$, we can obtain a commutative product $\odot$ in B, where (α, α) is the identity, having the following rules:

- $(m, \alpha) \odot (p, \alpha) = (mp, m + p)$
- $(m, n) \odot (p, \alpha) = \begin{cases} \left(\dfrac{mp + r}{n + p}, \dfrac{m + np}{n + p}\right), & \text{if} \quad n + p \neq 0 \\ \left(\dfrac{mp + r}{m - n^2}, \alpha\right), & \text{if} \quad n = -p, m - n^2 \neq 0 \\ (\alpha, \alpha), & \text{otherwise} \end{cases}$
- $(m, n) \odot (p, q) = \begin{cases} \left(\dfrac{mp + (n + q)r}{m + p + nq}, \dfrac{np + mq + r}{m + p + nq}\right), & \text{if } m + p + nq \neq 0 \\ \left(\dfrac{mp + (n + q)r}{np + mq + r}, \alpha\right), & \text{if } m + p + nq = 0, np + mq + r \neq 0 \\ (\alpha, \alpha), & \text{otherwise} \end{cases}$

As a consequence, the following proposition holds.

Proposition 2. *$(B, \odot)$ is a commutative group with identity (α, α). The inverse of an element (m, n), with $m - n^2 \neq 0$, is $\left(\frac{nr - m^2}{m - n^2}, \frac{r - mn}{m - n^2}\right)$. The inverse of an element (m^2, m) is $(-m, \alpha)$. Viceversa, the inverse of an element (m, α) is $(-m^2, m)$.*

Remark 4. When $\mathbb{F} = \mathbb{R}$, the element α can be viewed as ∞ and the points in B of the kind (m, ∞), (∞, ∞) as points at infinity.

Furthermore, if we consider $\mathbb{F} = \mathbb{Z}_p$ where p is prime, then we have a field, so $B = \mathbb{A}^*/\mathbb{F}^* = \mathbb{Z}_p^*[t]/\mathbb{Z}_p^*$ is a field too. It is easy to notice that the point $0 = [0 : 0 : 0] \notin B$ and we can consider the equivalence relation $\sim$ induced by the action of $\mathbb{Z}_p^*$ on the set $\mathbb{Z}_p^*[t]$ such that $b_1 \sim b_2 \iff \exists \lambda \in \mathbb{Z}_p^* : b_1 = \lambda b_2$ and now it is clear that B is a projective space.

Remark 5. If $\mathbb{F}$ is not a finite field, let us denote B as B_0, $B_1 = B_0^*/\mathbb{F}^*$, $B_n = B_{n-1}^*/\mathbb{F}^*$ and so $\forall n$, then we have $B_{n+1} \subset B_n$ and so we have a directed system, in fact $\forall n\ B_n \subset B_0$; moreover let us consider the family of maps $\{\phi_{n,n+1}\}_n$ with $\phi_{n,n+1} : B_{n+1} \hookrightarrow B_n$, where $\phi_{n,n} = id_{B_n}$, such that $\phi_{n,n+1} \circ \phi_{n+1,m} = \phi_{n,m}$ and $\phi_{n,m} : B_m \hookrightarrow B_n$. At this point it is clear that $(\{B_n\}, \phi_{n,n+1})$ is a projective system, hence we naturally consider the inverse limit $\varprojlim B_i$, that is equipped with a family of projection maps $\{p_n\}_n$ such that the inverse limit has the following universal property, showed by the commutative diagram

$$\begin{tikzcd}
 & \mathbb{A}^* \arrow[d, "\exists!\phi"] \arrow[ddl, bend right, "\pi_{n+1}"'] \arrow[dd, bend right=60, "\pi_n"'] \arrow[ddr, bend left, "\pi_{n-1}"] & \\
 & \varprojlim B_i \arrow[dl, "p_{n+1}"'] \arrow[d, "p_n"] \arrow[dr, "p_{n-1}"] & \\
B_{n+1} \arrow[r, "\phi_{n,n+1}"'] & B_n \arrow[r, "\phi_{n-1,n}"'] & B_{n-1}
\end{tikzcd}$$

with $\pi_n \circ p_n^{-1} = id_{B_n}$

Remark 6. We consider $\mathbb{F}$ as a topological field, so that $\mathcal{C}$ has the topology induced as a subset of $\mathbb{F}^3$. The cubic Pell curve

$$\mathcal{C} = \{(x, y, z) \in \mathbb{F}^3 : N(x, y, z) := x^3 + ry^3 + r^2z^3 - 3rxyz = 1\},$$

endowed with the non standard product we have previously defined, can be studied as a topological group. Indeed the group operation

$$\mathcal{C} \times \mathcal{C} \longrightarrow \mathcal{C}, ((x_1, y_1, z_1), (x_2, y_2, z_2)) \longmapsto (x_1x_2, y_1y_2, z_1z_2)$$

is a continuous mapping and the inversion map $\mathcal{C} \longrightarrow \mathcal{C}, (x, y, z) \longmapsto (\bar{x}, \bar{y}, \bar{z})$ is likewise continuous, according to the fact that $N(x, y, z) = 1$. If $\mathbb{F} = \mathbb{R}$, then we can consider $\mathcal{C}$ equipped with the Euclidean topology, otherwise if $\mathbb{F} = \mathbb{Z}_p$, then the discrete topology is the most natural topology we can put on it, but maybe it is not the only one interesting, even if the only one that is T_0.

3 A Public-Key Cryptosystem

3.1 The Scheme

When $\mathbb{F} = \mathbb{Z}_p$ (and fixing $\alpha = \infty$), the situation is interesting for cryptographic applications. Indeed, in this case we have $\mathbb{A} = GF(p^3)$, i.e., $\mathbb{A}$ is the Galois field of order p^3. Thus, by construction, B is a cyclic group of order $\dfrac{p^3 - 1}{p - 1} = p^2 + p + 1$, with respect to a well-defined product, and an analogous of the little Fermat's theorem holds:

$$(m, n)^{\odot p^2+p+1} \equiv (\infty, \infty) \pmod{p}, \tag{2}$$

where the power is evaluated by using the product $\odot$, for any $m \in \mathbb{Z}_p$ and $n \in \mathbb{Z}_p \cup \{\infty\}$.

Remark 7. It follows from (2) that

$$(m,n)^{\odot(p^2+p+1)(q^2+q+1)} \equiv (\infty,\infty) \pmod N,$$

where $N = pq$, for p and q prime numbers. This does not mean that, when B is constructed over $\mathbb{Z}_N$, B is a group. In this case we only have an analogous of the Euler's theorem. In other words when we construct B over $\mathbb{Z}_p$ (p prime) our product $\odot$ works like the standard product in $\mathbb{Z}_p$. Moreover, when we consider B over $\mathbb{Z}_N$, our product $\odot$ works like the standard product in $\mathbb{Z}_N$.

As a consequence we can construct a public-key cryptosystem similar to the RSA scheme, but using our product $\odot$.

The following steps describe the keys generation:

- choose two prime numbers p, q
- compute $N = pq$
- choose an integer e such that $(e, (p^2+p+1)(q^2+q+1)) = 1$
- choose a non-cube integer r in $\mathbb{Z}_p$, $\mathbb{Z}_q$ and $\mathbb{Z}_N$
- compute d such that $ed \equiv 1 \pmod{(p^2+p+1)(q^2+q+1)}$.

The public encryption key is (N, e, r) and the secret decryption key is (p, q, d). Given a pair of messages m_1 and m_2 in $\mathbb{Z}_N$, they can be encrypted by

$$(c_1, c_2) \equiv (m_1, m_2)^{\odot e} \pmod N.$$

The receiver can decrypt the messages evaluating

$$(c_1, c_2)^{\odot d} \pmod N.$$

3.2 Some Remarks

In the following, we discuss some peculiarities of our cryptosystem.

First, our scheme is more secure than RSA in broadcast scenarios, i.e., when the plaintext is encrypted for different receivers using the same public exponent and it is possible to recover the plaintext message by solving a set of congruences of polynomials [9]. However, this attack can not be applied when the trapdoor function is not a simple monomial power as in RSA [12]. Thus, this kind of attacks fails in our scheme.

Another classical attack against the RSA scheme is the Wiener attack [23]. Said e and d the public and private exponents, respectively, in the RSA scheme the following relation holds

$$ed - k\varphi(N) = 1$$

for a certain integer k, where φ is the Euler totient function and $N = pq$ (for p and q prime numbers) is the modulo with respect to messages are encrypted and decrypted. For large values of N the following bounds hold:

$$N - 3\sqrt{N} < \varphi(N) < N \tag{3}$$

The Wiener attack exploits properties of continued fractions. Indeed, thanks to the previous inequalities, we have

$$|\frac{k}{d} - \frac{e}{N}| < \frac{1}{2d^2},$$

i.e., by Legendre theorem, d is the denominator of a convergent of the continued fraction expansion of $\frac{e}{N}$ and consequently the private exponent d can be recovered. In our case, the role of $\varphi(N)$ is substituted by $(p^2+p+1)(q^2+q+1)$. This leads to a less efficient evaluation of the decryption exponent, however in this situation inequalities similar to (3) can not be found, making the Wiener attack not usable against our scheme. Moreover, for the same reason, further attacks exploiting continued fractions, reviewed in [7], fail in our case.

Remark 8. The private exponent d can be effectively recovered by using the Wiener attack if it is less than $N^{1/4}$, where N is the RSA-modulo. A typical size of the RSA-modulo is 1024–bit. Thus, in this case, it is required that the size of d must be at least 256 bits long in order to avoid the Wiener attack, but this is unfortunate for low-power devices [3]. Using the proposed scheme, the dimension of the private exponent could be less than 256 bits without being affected by the Wiener attack.

Finally, our scheme appears to be robust against another class of attack presented in [20] (see also [11], Sect. 3.1, for a review of the attack). We recall this attack here for the reader. It is supposed that it is known a linear relation between two plaintexts M_1 and M_2:

$$M_2 = M_1 + \mathit{\Delta}$$

where $\mathit{\Delta}$ is known and $C_1 \equiv M_1^e \pmod{N}$, $C_2 \equiv M_2^e \pmod{N}$. In this case, the attack can retrieve the plaintext messages evaluating the greatest common divisor of the polynomials

$$x^e - C_1 \pmod{N}, \quad (x + \mathit{\Delta})^e - C_2 \pmod{N}.$$

In our case, the situation is more complicated, since the exponentiation yields rational functions and not polynomials. Moreover, in our case, we deal with bivariate polynomials.

3.3 Evaluation of the Powers with Respect to $\odot$ by Means of Generalized Rédei Functions

The Rédei rational functions were introduced by Rédei in [21] from the development of $(z + \sqrt{d})^n$, where z is an integer and d a non-square positive integer. We can define the Rédei polynomials $N_n(d, z)$ and $D_n(d, z)$ as follows:

$$(z + \sqrt{d})^n = N_n(d, z) + D_n(d, z)\sqrt{d}, \quad \forall n \geq 0.$$

The Rédei polynomials have the following closed form:

$$N_n(d,z)=\sum_{k=0}^{[n/2]}\binom{n}{2k}d^k z^{n-2k},\quad D_n(d,z)=\sum_{k=0}^{[n/2]}\binom{n}{2k+1}d^k z^{n-2k-1}.$$

The Rédei rational functions are defined by

$$Q_n(d,z)=\frac{N_n(d,z)}{D_n(d,z)},\quad \forall n\geq 1$$

and can be also evaluated by means of powers of matrices. Indeed, we have

$$\begin{pmatrix} z & d \\ 1 & z \end{pmatrix}^n=\begin{pmatrix} N_n & dD_n \\ D_n & N_n \end{pmatrix},$$

see [8].

They are classical and interesting functions in number theory since, for instance, they provide approximations of square roots, are permutations in finite fields and Rédei polynomials belong to the class of the Dickson polynomials [14]. Moreover, they have been applied in several contexts, like the creation of a cryptographic system based on the Dickson scheme [18] and the generation of pseudorandom sequences [22].

Here, we see that the powers of elements in B can be evaluated by means of a certain generalization to the cubic case of the Rédei functions.

Starting from the development of $(z_1+z_2\sqrt[3]{r}+\sqrt[3]{r^2})^n$, with $z_1,z_2,r\in\mathbb{F}$ and r non-cube, we can introduce three sequences of polynomials $A_n(r,z_1,z_2)$, $B_n(r,z_1,z_2)$, $C_n(r,z_1,z_2)$ that generalize the Rédei polynomials. We define

$$(z_1+z_2\sqrt[3]{r}+\sqrt[3]{r^2})^n=A_n(r,z_1,z_2)+B_n(r,z_1,z_2)\sqrt[3]{r}+C_n(r,z_1,z_2)\sqrt[3]{r^2},\quad \forall n\geq 0.$$

Hence, the rational functions $\dfrac{A_n}{C_n}$ and $\dfrac{B_n}{C_n}$, for $n\geq 1$ can be considered a generalization to the cubic case of the Rédei rational functions.

Remark 9. Let us observe that for introducing the generalized Rédei functions, it is not necessary to work in a field. Indeed, the previous definition works even in the case that z_1,z_2,r belongs to a commutative ring with identity. Indeed, the original Rédei polynomials were introduced in $\mathbb{Z}$. We have chosen to define the generalized Rédei polynomials in the field $\mathbb{F}$ only for being consistent with the notation used for introducing B as a group and not introducing new notation.

In the following proposition, we see that also the generalized Rédei polynomials can be evaluated by means of a matricial approach.

Proposition 3. *Let $A_n(r,z_1,z_2), B_n(r,z_1,z_2), C_n(r,z_1,z_2)$ be the generalized Rédei polynomials, then*

$$\begin{pmatrix} z_1 & r & rz_2 \\ z_2 & z_1 & r \\ 1 & z_2 & z_1 \end{pmatrix}^n=\begin{pmatrix} A_n & rC_n & rB_n \\ B_n & A_n & rC_n \\ C_n & B_n & A_n \end{pmatrix},\quad \forall n\geq 0$$

Proof. In the following, for the seek of simplicity we omit the dependence on r, z_1, z_2. We prove the thesis by induction on n.
Basis: for $n = 0$ we have $A_0 = 1, B_0 = 0, C_0 = 0$ and $(z_1 + z_2\sqrt[3]{r} + \sqrt[3]{r^2})^0 = 1$, i.e.,

$$\begin{pmatrix} z_1 & r & rz_2 \\ z_2 & z_1 & r \\ 1 & z_2 & z_1 \end{pmatrix}^0 = \begin{pmatrix} A_0 & 0 & 0 \\ 0 & A_0 & 0 \\ 0 & 0 & A_0 \end{pmatrix}.$$

Similarly, it is straightforward to check the cases $n = 1, 2$.
Inductive step: we assume the statement holds for some natural number $n - 1$ and we prove that holds for n too. We have

$$\begin{pmatrix} z_1 & r & rz_2 \\ z_2 & z_1 & r \\ 1 & z_2 & z_1 \end{pmatrix}^n = \begin{pmatrix} z_1 & r & rz_2 \\ z_2 & z_1 & r \\ 1 & z_2 & z_1 \end{pmatrix}^{n-1} \begin{pmatrix} z_1 & r & rz_2 \\ z_2 & z_1 & r \\ 1 & z_2 & z_1 \end{pmatrix}$$

$$= \begin{pmatrix} A_{n-1} & rC_{n-1} & rB_{n-1} \\ B_{n-1} & A_{n-1} & rC_{n-1} \\ C_{n-1} & B_{n-1} & A_{n-1} \end{pmatrix} \begin{pmatrix} z_1 & r & rz_2 \\ z_2 & z_1 & r \\ 1 & z_2 & z_1 \end{pmatrix}.$$

Thus, we have to show that

$$\begin{cases} A_n = z_1 A_{n-1} + rz_2 C_{n-1} + rB_{n-1} \\ B_n = z_1 B_{n-1} + z_2 A_{n-1} + rC_{n-1} \\ C_n = z_1 C_{n-1} + z_2 B_{n-1} + A_{n-1} \end{cases}.$$

By definition of generalized Rédei polynomials, we have

$$(z_1 + z_2\sqrt[3]{r} + \sqrt[3]{r^2})^n = A_n + B_n\sqrt[3]{r} + C_n\sqrt[3]{r^2}.$$

On the other hand

$$(z_1 + z_2\sqrt[3]{r} + \sqrt[3]{r^2})^n = (z_1 + z_2\sqrt[3]{r} + \sqrt[3]{r^2})^{n-1}(z_1 + z_2\sqrt[3]{r} + \sqrt[3]{r^2})$$
$$= (A_{n-1} + B_{n-1}\sqrt[3]{r} + C_{n-1}\sqrt[3]{r^2})(z_1 + z_2\sqrt[3]{r} + \sqrt[3]{r^2})$$

from which, expanding the last product, the thesis easily follows.

In the next proposition, we see that these functions can be used in order to evaluate powers of elements (z_1, z_2) in B.

Proposition 4. *Given* $(z_1, z_2) \in B$ *and let* $A_n(r, z_1, z_2), B_n(r, z_1, z_2), C_n(r, z_1, z_2)$ *be the generalized Rédei polynomials, we have*

$$(z_1, z_2)^{\odot n} = \begin{cases} \left(\dfrac{A_n}{C_n}, \dfrac{B_n}{C_n}\right), & \text{if} \quad C_n \neq 0 \\ \left(\dfrac{A_n}{B_n}, \alpha\right), & \text{if} \quad B_n \neq 0,\ C_n = 0 \\ (\alpha, \alpha), & \text{if} \quad B_n = C_n = 0 \end{cases},$$

for $n \geq 1$.

Proof. By the previous proposition, we have

$$\begin{pmatrix} A_n & rC_n & rB_n \\ B_n & A_n & rC_n \\ C_n & B_n & A_n \end{pmatrix} \begin{pmatrix} A_m & rC_m & rB_m \\ B_m & A_m & rC_m \\ C_m & B_m & A_m \end{pmatrix} = \begin{pmatrix} A_{m+n} & rC_{m+n} & rB_{m+n} \\ B_{m+n} & A_{m+n} & rC_{m+n} \\ C_{m+n} & B_{m+n} & A_{m+n} \end{pmatrix},$$

from which we get

$$\begin{cases} A_{m+n} = A_mA_n + rB_mC_n + rB_nC_m \\ B_{m+n} = A_mB_n + A_nB_m + rC_mC_n \\ C_{m+n} = A_mB_n + B_mB_n + A_nC_m \end{cases}.$$

Thus, if $C_m, C_n \neq 0$ and $C_{m+n} = A_mB_n + B_mB_n + A_nC_m \neq 0$, i.e., $\frac{A_n}{C_n} + \frac{A_m}{C_m} + \frac{B_mB_n}{C_nC_m} \neq 0$ (that is the condition $m+p+nq \neq 0$ for the product $(m,n) \odot (p,q)$), we have

$$\begin{cases} \dfrac{A_{m+n}}{C_{m+n}} = \dfrac{\dfrac{A_n}{C_n}\dfrac{A_m}{C_m} + r\dfrac{B_m}{C_m} + r\dfrac{B_n}{C_n}}{\dfrac{A_m}{C_m} + \dfrac{B_n}{C_n}\dfrac{B_m}{C_m} + \dfrac{A_n}{C_n}} \\ \dfrac{B_{m+n}}{C_{m+n}} = \dfrac{\dfrac{B_n}{C_n}\dfrac{A_m}{C_m} + \dfrac{B_m}{C_m}\dfrac{A_n}{C_n} + r}{\dfrac{A_m}{C_m} + \dfrac{B_n}{C_n}\dfrac{B_m}{C_m} + \dfrac{A_n}{C_n}} \end{cases}$$

and this is equivalent to say that

$$\left(\frac{A_{m+n}}{C_{m+n}}, \frac{B_{m+n}}{C_{m+n}}\right) = \left(\frac{A_n}{C_n}, \frac{B_n}{C_n}\right) \odot \left(\frac{A_m}{C_m}, \frac{B_m}{C_m}\right).$$

In the case that $B_{m+n} \neq 0$ $C_{m+n} = A_mB_n + B_mB_n + A_nC_m = 0$, i.e., $\frac{A_n}{C_n} + \frac{A_m}{C_m} + \frac{B_mB_n}{C_nC_m} = 0$ (that is the condition $m+p+nq = 0$ for the product $(m,n) \odot (p,q)$), then we have

$$\left(\frac{A_{m+n}}{B_{m+n}}, \alpha\right) = \left(\frac{A_m}{C_m}, \frac{B_m}{C_m}\right) \odot \left(\frac{A_n}{C_n}, \frac{B_n}{C_n}\right).$$

Now, considering that $\left(\dfrac{A_1}{C_1}, \dfrac{B_1}{C_1}\right) = (z_1, z_2)$, the thesis follows.

When we consider elements of the kind (z, α) in B, the previous generalized Rédei functions can not be applied for evaluating the powers. However, in the following proposition, we see how these powers can be evaluated in a similar way.

Proposition 5. *Given* $(z_1, \alpha) \in B$ *and let* $\bar{A}_n(r, z_1)$, $\bar{B}_n(r, z_1)$, $\bar{C}_n(r, z_1)$ *be polynomials defined by*

$$(z_1 + \sqrt[3]{r})^n = \bar{A}_n(r, z_1) + \bar{A}_n(r, z_1)\sqrt[3]{r} + \bar{A}_n(r, z_1)\sqrt[3]{r^2}, \quad \forall n \geq 1.$$

We have that

1. $$\begin{pmatrix} z_1 & 0 & r \\ 1 & z_1 & 0 \\ 0 & 1 & z_1 \end{pmatrix}^n = \begin{pmatrix} \bar{A}_n & \bar{C}_n & r\bar{B}_n \\ \bar{B}_n & \bar{A}_n & \bar{C}_n \\ \bar{C}_n & \bar{B}_n & \bar{A}_n \end{pmatrix}, \quad \forall n \geq 0$$

2. $$(z_1, \alpha)^{\odot n} = \begin{cases} \left(\dfrac{\bar{A}_n}{\bar{C}_n}, \dfrac{\bar{B}_n}{\bar{C}_n} \right), & \textit{if} \quad \bar{C}_n \neq 0 \\ \left(\dfrac{\bar{A}_n}{\bar{B}_n}, \alpha \right), & \textit{if} \quad \bar{B}_n \neq 0,\ \bar{C}_n = 0 \\ (\alpha, \alpha), & \textit{if} \quad \bar{B}_n = \bar{C}_n = 0 \end{cases}$$

Proof. The proofs are similar to proofs of Propositions 3 and 4 and are left to the reader.

Remark 10. As we have already pointed out, the generalized Rédei functions can be used for evaluating powers in B even in the case that we are working in a ring and not in the field $\mathbb{F}$. Let us note that in this case B is not a group but the product is well-defined and the powers can be evaluated by Propositions 4 and 5. In this case conditions "$\neq 0$" means "is invertible".

4 Conclusion

In this paper, we have proposed a novel RSA-like scheme that is more secure than RSA in broadcast applications and is not affected by the Wiener attack. Moreover, it appears more robust than RSA with respect to other attacks that exploit the knowledge of a linear relation occurring between two plaintexts. This scheme has been developed by using a new group equipped with a non-standard product whose powers can be evaluated by means of some generalized Rédei functions. This group and its product have shown many interesting properties and relations highlighting that they are worth investigating due to their perspectives. Certainly, in this work we have only given an idea of their use in cryptographic applications, but the present scheme should be further discussed and improved. In the following, we advise some further studies:

- In [16], the author exhibits an algorithm of complexity $O(log_2(n))$ with respect to addition, subtraction and multiplication to evaluate Rédei rational functions over a ring. It will be interesting to study a similar algorithm in order to obtain an efficient method for evaluating the generalized Rédei functions introduced in this paper, so that the encryption cost of our algorithm is equal to the encryption cost of the RSA scheme or less considering that in our scheme we encrypt two messages at once.
- We conjecture that $(B, \odot)$ and $(\mathcal{C}, \bullet)$ are isomorphic. Proving this fact and finding the isomorphism lead to important consequences. First, the isomorphism could be exploited in order to improve our scheme following the ideas of RSA-like schemes based on isomorphism between two groups (see, e.g.,

[12,19]). Moreover, in this way a method for generating the solutions of the cubic Pell equation could be found (note that such a method is still missing [1]). As a special case, we will also state that the number of solutions of the cubic Pell equation in $\mathbb{Z}_p$ is p^2+p+1 (as numerical simulations appear to confirm). One could try to show that $B \simeq \mathcal{C}$ using the Short Five Lemma [10]: if in the following diagram we have two exact sequences, that is $\ker g = \mathrm{Im} f$ and $\ker k = \mathrm{Im} h$, whew both k and g are surjections and both h and f are injections, under the hypothesis that two of the down arrows are isomorphism, then the last down arrow is an isomorphism too.

$$\begin{tikzcd}
0 \arrow[r] & B \arrow[r, "f"] \arrow[d, "\simeq"] & \mathbb{Z}_p[t]/(\beta(t)) \arrow[r, "g"] \arrow[d, "id"] & \mathbb{A} \arrow[r] \arrow[d, "id"] & 0 \\
0 \arrow[r] & \mathcal{C} \arrow[r, "h"'] & \mathbb{Z}_p[t]/(\beta(t)) \arrow[r, "k"'] & \mathbb{A} \arrow[r] & 0
\end{tikzcd}$$

So our goal is to find an appropriate $(\beta(t))$ and the maps previously introduced, with particular attention to the degree of the polynomial $(\beta(t))$. For now, we were only able to find the following morphism

$$\epsilon : \begin{cases} B \to \mathcal{C} \\ (m,n) \mapsto \left(\dfrac{m^3+6mnr+n^3r+r^2}{m^3+rn^3+r^2-3rmn}, \dfrac{3(m^2n+mr+n^2r)}{m^3+rn^3+r^2-3rmn}, \dfrac{3(m^2+mn^2+nr)}{m^3+rn^3+r^2-3rmn} \right) \\ (m,\alpha) \mapsto \left(1, \dfrac{3m^2}{m^3+r}, \dfrac{3m}{m^3+r}\right) \\ (\alpha,\alpha) \mapsto (1,0,0) \end{cases}$$

Moreover, let us recall that $\mathbb{Z}_p$ has non-cubic residues only when $p \equiv 1 \pmod 3$, and consequently 3 divides p^2+p+1. Thus, when we consider $\mathbb{F} = \mathbb{Z}_p$, we are able to construct the group B only for the prime numbers p such that p^2+p+1 is divisible by 3. Then we have observed that we have $|Im\epsilon| = \frac{|B|}{3}$.

- The scheme should be studied from a computational point of view, in order to give more precise and effective results about its efficiency and security. In this paper, we have only investigated some improvements regarding the security from a theoretical point of view.

References

1. Barbeau, E.J.: Pell's Equation. Springer, New York (2003). https://doi.org/10.1007/b97610
2. Bellini, E., Murru, N.: An efficient and secure RSA-like cryptosystem exploiting Rédei rational functions over conics. Finite Fields Appl. **39**, 179–194 (2016)
3. Boneh, D.: Twenty years of attacks on the RSA cryptosystem. Notices Amer. Math. Soc. **46**, 203–213 (1999)

4. Christofferson, S.: Über eine Klasse von kubischen diophantischen Gleichungen mit drei Unbekannten. Arkiv för Matematik **3**(4), 355–364 (1957)
5. Coppersmith, D.: Small solutions to polynomial equations, and low exponent RSA vulnerabilities. J. Cryptol. **10**(4), 233–260 (1997)
6. Demytko, N.: A new elliptic curve based analogue of RSA. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 40–49. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48285-7_4
7. Dujella, A.: Continued fractions and RSA with small secret exponent. Tatra Mt. Math. Publ. **29**, 101–112 (2004)
8. von zur Gathen, J.: Tests for permutation polynomials. SIAM J. Comput. **20**, 591–602 (1991)
9. Hastad, J.: N using RSA with low exponent in a public key network. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 403–408. Springer, Heidelberg (1986). https://doi.org/10.1007/3-540-39799-X_29
10. Jacobson, N.: Basic Algebra II. W. H. Freeman and Company, San Francisco (1989)
11. Joye, M., Quisquater, J.-J.: Protocol failures for RSA-like functions using Lucas sequences and elliptic curves. In: Lomas, M. (ed.) Security Protocols 1996. LNCS, vol. 1189, pp. 93–100. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-62494-5_8
12. Koyama, K.: Fast RSA-type schemes based on singular cubic curves $y^2 + axy \equiv x^3 \pmod{n}$. In: Guillou, L.C., Quisquater, J.-J. (eds.) EUROCRYPT 1995. LNCS, vol. 921, pp. 329–340. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-49264-X_27
13. Koyama, K., Maurer, U.M., Okamoto, T., Vanstone, S.A.: New public-key schemes based on elliptic curves over the ring $\mathbb{Z}_n$. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 252–266. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_20
14. Lidl, R., Mullen, G.L., Turnwald, G.: Dickson polynomials. Pitman monographs surveys in pure applied mathematics, vol. 65. Longman, Harlow (1993)
15. Loxtou, J.H., Khoo, D.S.P., Bird, G.J., Seberry, J.: A cubic RSA code equivalent to factorization. J. Cryptol. **5**(2), 139–150 (1992)
16. More, W.: Fast evaluation on Rédei functions. Appl. Algebra Eng. Commun. Comput. **6**(3), 171–173 (1995)
17. Naccache, D., Stern, J.: A new public-key cryptosystem. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 27–36. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-69053-0_3
18. Nobauer, R.: Cryptanalysis of the Rédei scheme. Contrib. Gen. Algebra **3**, 255–264 (1984)
19. Padhye, S.: A public key cryptosystem based on Pell equation. IACR Cryptol. ePrint Arch. 191 (2006)
20. Patarin, J.: Some serious protocol failures for RSA with exponent e of less than 32 bits. CIRM Luminy, France, 25–29 September 1995
21. Rédei, L.: Uber eindeuting umkehrbare polynome in endlichen korpen. Acta Sci. Math. (Szeged) **11**, 85–92 (1946)
22. Topuzoglu, A., Winterhof, A.: Topics in geometry, coding theory and cryptography. Algebra Appl. **6**, 135–166 (2006)
23. Wiener, M.J.: Cryptanalysis of short RSA secret exponents. IEEE Trans. Inf. Theory **36**, 553–558 (1990)

Commutativity, Associativity, and Public Key Cryptography

Jacques Patarin[1] and Valérie Nachef[2(✉)]

[1] Laboratoire de Mathématiques de Versailles, UVSQ, CNRS,
Université de Paris-Saclay, 78035 Versailles, France
jpatarin@club-internet.fr
[2] Department of Mathematics, University of Cergy-Pontoise,
UMR CNRS 8088, 95000 Cergy-Pontoise, France
valerie.nachef@u-cergy.fr

Abstract. In this paper, we will study some possible generalizations of the famous Diffie-Hellman algorithm. As we will see, at the end, most of these generalizations will not be secure or will be equivalent to some classical schemes. However, these results are not always obvious and moreover our analysis will present some interesting connections between the concepts of commutativity, associativity, and public key cryptography.

Keywords: Diffie-Hellman algorithms · Chebyshev polynomials
New public key algorithms

1 Introduction

Classical Diffie-Hellman Key-Exchange Algorithm. The Diffie-Hellman algorithm [5] was the first published key exchange algorithm (1976). In fact, it is rather a two-party key establishment protocol, which also has "ephemeral public key" features. The new functionalities it offers has created a whole new area of science and engineering: public-key cryptography. Since 1976, many more algorithms have been found, and some of them can be seen as generalizations of the original Diffie-Hellman algorithm, for example when the computations are done in an elliptic curve instead of (mod p), where p is a prime number. In this paper, we will study some other possible generalizations and the link between this problem and commutativity or associativity in some mathematical structures (with one way properties).

Let us first recall what was the original Diffie-Hellman algorithm. Let p be a prime number and g be an element of $\mathbb{Z}/p\mathbb{Z}$ such that $x \mapsto g^x \pmod{p}$ is (as far as we know) a one way function. Typically p has more than 1024 bits and g can be a generator of $\mathbb{Z}/p\mathbb{Z}$. Let Alice and Bob (as in the original paper of Diffie and Hellman) be the two persons who want to communicate. Alice randomly chooses a secret value a between 1 and $p-1$, and she sends the value $A = g^a \pmod{p}$ to Bob. Similarly, Bob randomly chooses a secret value b between 1 and $p-1$ and

J. Kaczorowski et al. (Eds.): NuTMiC 2017, LNCS 10737, pp. 104–117, 2018.
https://doi.org/10.1007/978-3-319-76620-1_7

sends $B = g^b \pmod{p}$ to Alice. Then Alice and Bob are both able to compute a common key $K = g^{a.b} \pmod{p}$ (Alice by computing $K = B^a \pmod{p}$, and Bob by computing $K = A^b \pmod{p}$). However, if an adversary, Charlie is a passive observer of the messages exchanged on the line, he will obtain A and B, but, if $x \mapsto g^x \pmod{p}$ is one way, he will not obtain a and b, and if the so called "Diffie-Hellman" problem is difficult, he will not be able to compute K.

Remark 1. If Charlie is also able to send messages, it is well known that this simple algorithm can be attacked by a man in the middle attack. So the messages MUST be authenticated somewhat, for example in usual HTTPS web, the problem is solved. However, this is not the aim of this paper.

DH in a More General Frame. We can state this problem in a more general frame as proposed by Couveignes [4]:

If A is a (semi-)group and G is a set, then a (left) group action φ of A on G is a function:

$$\varphi\colon A \times G \to X\colon (a, g) \mapsto \varphi(a, g)$$

that satisfies the following two axioms (where we denote $\varphi(a, g)$ as $a \cdot g$):

- Identity $e \cdot g = g$ for all g in G. (Here, e denotes the identity element of A if A is a group).
- Compatibility $(ab) \cdot g = a \cdot (b \cdot g)$ for all $a, b \in A$ and all $g \in G$.

We say that A acts transitively on G. We now suppose that A is abelian. We require that the action $(a, g) \to a \cdot g$ is easy to compute but that given g and h in G, it is difficult to compute a such that $a \cdot g = h$. Now we can state DH in this more general frame. Let $g \in G$. Alice choose randomly a secret value $a \in A$ and send the value $a \cdot g$ to Bob. Similarly, Bob choose a secret value $b \in A$ and send $b \cdot g$ to Alice. Then Alice and Bob will share the common value $ab \cdot g$.

In this paper, we look for specific constructions that allow to use the algorithm, and we will assume that Charlie remains a passive attacker and does not create/modify/suppress any messages.

We propose a first type of construction with $A = \mathbb{N}$. For this it is enough to have a set G with a associative composition $*$. The structure of $\mathbb{N}$-set is given by $g^n := n - 1$-fold composition of g with itself. It is obvious that $\left(g^a\right)^b = g^{a \cdot b} = \left(g^b\right)^a$ and so we can use it for key exchange if, and only if, $g \mapsto g^n$ is one way to make the system secure.

In this paper we shall construct compositions $*$ on (affine) curves of genus 0 over finite fields. To find them we first go to such curves over $\mathbb{R}$ and use addition formulas for trigonometric functions to define compositions over $\mathbb{R}$. The next step then is to describe these compositions given by transcendental functions algebraically over the finite fields. Since associativity is inherited, we can use them to define a $\mathbb{N}$-set.

Remark 2

- We recall that the algebraic addition law on elliptic curves E (i.e. curves of genus 1) over any field is modeled after the addition theorems of elliptic functions, e.g. the Weierstrass $\wp$-function and $\wp'$-function.
- The $\wp$-function alone yields a partial $\mathbb{N}$-structure on $\mathbb{F}_q$ by the well-known formulas for the $X-$coordinate of the point $n \cdot (x, y)$ for $(x, y) \in E(\mathbb{F}_q)$.
- A generalization for hyperelliptic curves of genus ≥ 2 is, at least in principle, given by Theta-functions. For $g = 2$ this becomes very efficient [7].

Another possible constructions is to choose A as a family of functions with the composition law and that are pairwise commuting, defined on a space X. The action is then defined by $(f, x) \in A \times X \mapsto f(x)$. Here, A, is not necessary a group, but we can observe that commutativity is needed to be able to have DH in this context. In general, it is quite easy to design a very general commutative internal law on the elements (for example if $a \leq b$ we define $a * b$ as a fixed random element $\varphi(a, b)$ and if $b < a$, we define $a * b$ as $\varphi(b, a)$), but we want here associativity, not commutativity. On the opposite, for functions $f(x) = x^a$ and $g(x) = x^b$, we want $f \circ g = g \circ f$, i.e. commutativity. Here the composition of functions $\circ$ is always associative, but we want commutativity.

	Commutativity	Associativity
On the elements	Easy	What we want
On the functions	What we want	Easy

Quantum Computing on These Structures. We know that the quantum Shor's algorithm for factoring number or computing discrete logarithm $\pmod{p}$ in a finite field is polynomial. In the more general frame of groups operating on sets, when the group is abelian, one can only expect subexponential security [6]. Thus in our constructions, one cannot expect to obtain exponential security against quantum computing. This justifies the Feo [9] system using isogenies of supersingular elliptic curves.

Organisation of the Paper. In part I, we will concentrate on the first type of constructions, i.e. on the "associativity" property. In part II, we will concentrate on the second type on constructions, i.e. on the "commutativity" property, to generalize the fact that $(g^a) \circ (g^b) = (g^b) \circ (g^a)$, on the mathematical structure $(G, \circ)$.

Part I: Associative Properties on the Elements

In this part, we focus on the first type of construction and we present two examples. We work on affine curves of genus 0. Thus will end up with algebraic linear group of dimension 0. Indeed, we can get only tori or additive groups. This implies that we come to discrete logarithm in the multiplicative group of finite fields, as our examples will show.

2 Associativity with $a\sqrt{1+b^2}+b\sqrt{1+a^2}$

To generalize the Diffie-Hellman algorithm by working in a structure $(G,*)$ different from $(\mathbb{Z}/p\mathbb{Z},\times)$, we want:

- $*$ to be associative
- $x \mapsto g^x$ to be one way (from the best known algorithms, the existence of proven one way functions is an open problem since it would imply $P \neq NP$).

Moreover, we would like G to be as small as possible, but with a security greater than 2^{80}. Therefore, elements of G would have typically between 80 bits (or 160 bits if from a collision $g^x = g^y A$, we can find z such that $g^z = A$) and 2048 bits for example, since the computation of $a * b$ is expected to be fast. This is what we have on elliptic curves, but is it possible to suggest new solutions? Ideally, it would be great to generate a "random associative" structure on elements of size, say, about 200 bits for example. It is very easy to generate "random commutative" structures on elements of such size. Let for example a and b be two elements of 256 bits. If $a \leq b$, we can choose $a*b$ to be anything (for example $a * b = AES - CBC_k(a\|b)$ where k is a random value of 128 bits to be used as the AES key) and if $b < a$ then to define $b * a$ as $a * b$. However here we want to design a "random associative" structure on elements of about 200 bits and not a "random commutative" structure, and this is much more difficult! In fact, for associativity structure of this size, we do not know how to get them if we do not create a specific mathematical structure that gives the associativity. But then, there is a risk that such a structure could be used to attack the scheme. In this section, we will study an example of associativity created in this way. More precisely, we will study the operation $a * b = a.\sqrt{1+b^2} + b.\sqrt{1+a^2}$ on a set G where $., +$ and $\sqrt{}$ can be defined (we will see examples below). Let us first see why $*$ is associative on various G.

2.1 Associativity in $(\mathbb{R},*)$

Definition 1. $\forall a,b \in \mathbb{R},\ a * b = a.\sqrt{1+b^2} + b.\sqrt{1+a^2}$.

We will see that $(\mathbb{R},*)$ is a group. In fact the only difficult part in the proof is to prove the associativity of $*$. We will see 3 different proofs of this fact, since all of these proofs are interesting.

Associativity of *: Proof n°1. A nice way to prove the associative property is to notice that sinh function is a bijection from $\mathbb{R}$ to $\mathbb{R}$ that satisfies: $\forall a \in \mathbb{R}$, $\forall b \in \mathbb{R}$, $\sinh(a+b) = \sinh(a) * \sinh(b)$ (since $\sinh(a+b) = \sinh a \cosh b + \sinh b \cosh a$). This shows that sinh is an isomorphism from $(\mathbb{R},+)$ to $(\mathbb{R},*)$ and therefore $*$ is associative and $(\mathbb{R},*)$ is a group.

Associativity of *: Proof n°2

Theorem 1

$$\forall a \in \mathbb{R},\, \forall b \in \mathbb{R},\; \left(a\sqrt{b^2+1}+b\sqrt{a^2+1}\right)^2+1=\left(ab+\sqrt{a^2+1}\sqrt{b^2+1}\right)^2$$

Proof. It is obvious by developing the two expressions.

Theorem 2

$$\forall a, b, c \in \mathbb{R},\; (a*b)*c = a*(b*c)$$

Proof Let $\alpha = a\sqrt{b^2+1}+b\sqrt{a^2+1}$. Then $A=(a*b)*c=\alpha*c=\alpha\sqrt{c^2+1}+c\sqrt{\alpha^2+1}$. Now from Theorem 1, $\sqrt{\alpha^2+1}=ab+\sqrt{a^2+1}\sqrt{b^2+1}$ (this is true even when $a<0$ or $b<0$). Therefore $(a{*}b){*}c=(a\sqrt{b^2+1}{+}b\sqrt{a^2+1})\sqrt{c^2+1}+abc+c\sqrt{a^2+1}\sqrt{b^2+1}$. Similarly, let $\beta=b\sqrt{c^2+1}+c\sqrt{b^2+1}$. Then $B=a*(b*c)=a*\beta=a\sqrt{\beta^2+1}+\beta\sqrt{a^2+1}$. Then from Theorem 1, $\sqrt{\beta^2+1}=bc+\sqrt{b^2+1}\sqrt{c^2+1}$. Therefore $B=a*(b*c)=abc+a\sqrt{b^2+1}\sqrt{c^2+1}+(b\sqrt{c^2+1}+c\sqrt{b^2+1})\sqrt{a^2+1}$. Thus we obtain $A=B$.

Associativity of *: Proof n°3. Here, we will define a law on $\mathbb{R}^2$, called "Domino Law" and represented by $\boxminus$.

Definition 2. *Let $(a,\alpha)\in\mathbb{R}^2$ and $(b,\beta)\in\mathbb{R}^2$. Then the $\boxminus$ law is defined by*

$$(a,\alpha)\boxminus(b,\beta)=(a\beta+b\alpha,ab+\alpha\beta)$$

We can notice that $\boxminus$ is very similar to the multiplication in $\mathbb{C}$, except that we have $ab+\alpha\beta$ instead of $ab-\alpha\beta$. Here $a\beta+b\alpha$ is the analog of the imaginary part and $ab+\alpha\beta$ is the analog of the real part.

Proposition 1. *The $\boxminus$ law is associative:*

$$\forall(a,\alpha),(b,\beta),(c,\gamma),\quad (a,\alpha)\boxminus[(b,\beta)\boxminus(c,\gamma)]=[(a,\alpha)\boxminus(b,\beta)]\boxminus(c,\gamma)$$

Proof. It is easy to see that

$$\begin{aligned}(a,\alpha)\boxminus[(b,\beta)\boxminus(c,\gamma)]&=[(a,\alpha)\boxminus(b,\beta)]\boxminus(c,\gamma)\\&=(abc+a\beta\gamma+b\alpha\gamma+c\alpha\gamma,ab\gamma+ac\beta+\alpha bc+\alpha\beta\gamma)\end{aligned}$$

Corollary 1. *The $*$ law is associative.*

Proof. First, using Theorem 1, we notice that $(a,\sqrt{1+a^2})\boxminus(b,\sqrt{1+b^2})=(a*b,\sqrt{1+(a*b)^2})$. Therefore, the associativity of $\boxminus$ implies the associativity of $*$, since $*$ is the restriction of $\boxminus$ on the curve $b^2=a^2+1$.

2.2 Application to Finite Fields: A New Group $(P,*)$ for Cryptography

Let K be a finite field. Let $P(K)=\{x\in K,\exists\alpha\in K,1+x^2=\alpha^2\}$. When $a\in P$, let $\sqrt{a^2+1}$ denote any value α such that $\alpha^2=a^2+1$ (we will choose later if $\sqrt{a^2+1} = \alpha$ or $\sqrt{a^2+1} = -\alpha$). At this stage, we will only need that $\sqrt{a^2+1}$ denotes always the same value, α, or $-\alpha$ when a is fixed. When there is no ambiguity, $P(K)$ will be simply denoted by P.

Theorem 3

$$\forall a \in P,\, \forall b \in P,\, (a\sqrt{b^2+1} + b\sqrt{a^2+1})^2 + 1 = (ab + \sqrt{a^2+1}\sqrt{b^2+1})^2$$

Proof. As with Theorem 1, the proof is obvious: we just have to develop the two expressions.

Definition 3. *When $a \in P$ and $b \in P$, we will denote by $a * b = a\sqrt{b^2+1} + b\sqrt{a^2+1}$.*

Remark 3. For $\sqrt{a^2+1}$ we have two possibilities in K, α and $-\alpha$, and for $\sqrt{b^2+1}$, we also have two possibilities, β and $-\beta$. Therefore, for $a * b$, we have so far 4 possibilities. So far we just assume that one of these possibilities is choosen, and later (at the end of this Sect. 2.2) we will see how to choose one of these 4 possibilities in order to have a group $(P, *)$. Moreover we will always choose $\sqrt{1} = 1$.

Theorem 4. *$*$ is associative on P.*

Proof. This comes directly from Theorem 3 with the same proof as proof n°2 on $(\mathbb{R}, *)$.

Therefore, we can design a variant of the Diffie-Hellman scheme on $(P, *)$. To be more precise, we will now explain how to compute $\sqrt{1+a^2}$ explicitly.

Theorem 5. *We have the following properties:*
$\forall a \in P,\, a * 0 = 0 * a = a$ $\quad$ $\forall a, b \in P,\, (-a) * (-b) = -(a * b)$
$\forall a \in P,\, a * (-a) = (-a) * a = 0$ $\quad$ $\forall a, b \in P,\, (-a) * b = -(a * (-b))$

Proof. This comes immediately from $\sqrt{1} = 1$ and from the fact that $\sqrt{a^2+1}$ will always be the same value in all the expressions used for $*$.

Theorem 6. $\forall a, b \in P,\ a * b \in P$.

Proof. From Theorem 3, $1 + (a * b)^2$ is a square.

Theorem 7

$$[\forall a, b \in P,\, \sqrt{(ab + \sqrt{a^2+1}\sqrt{b^2+1})^2} = ab + \sqrt{a^2+1}\sqrt{b^2+1}]$$
$$\Longrightarrow \forall a, b, c \in P,\ a * (b * c) = (a * b) * c$$

Proof. Let $A = (a * b) * c$ and $B = a * (b * c)$. Let $\alpha = a\sqrt{b^2+1} + b\sqrt{a^2+1}$. Let $\beta = b\sqrt{c^2+1} + c\sqrt{b^2+1}$. From Theorem 7 we have $\sqrt{\alpha^2+1} = \pm ab + \sqrt{a^2+1}\sqrt{b^2+1}$ and similarly $\sqrt{\beta^2+1} = \pm bc + \sqrt{b^2+1}\sqrt{c^2+1}$. Therefore $A = (a\sqrt{b^2+1} + b\sqrt{a^2+1})\sqrt{c^2+1} \pm c(ab + \sqrt{a^2+1}\sqrt{b^2+1})$ and $B = (b\sqrt{c^2+1} + c\sqrt{b^2+1})\sqrt{a^2+1} \pm a(bc + \sqrt{b^2+1}\sqrt{c^2+1})$. We see that if here we will have two "+", then $A = B$, i.e. a sufficient condition to have $A = B$ is to have $\forall a, b \in P,\, \sqrt{(ab + \sqrt{a^2+1}\sqrt{b^2+1})^2} = ab + \sqrt{a^2+1}\sqrt{b^2+1}$.

We will denote by $\sharp$ this condition

$$\forall a, b \in P,\ \sqrt{(ab+\sqrt{a^2+1}\sqrt{b^2+1})^2} = ab+\sqrt{a^2+1}\sqrt{b^2+1}\ (\sharp)$$

From Theorem 3, $\sharp$ also means:

$$\forall a, b \in P,\ \sqrt{1+(a*b)^2} = ab+\sqrt{1+a^2}\sqrt{1+b^2} \quad (\sharp\sharp)$$

From ($\sharp\sharp$) and $a*b = a\sqrt{1+b^2}+b\sqrt{1+a^2}$, we see that from $(a, \sqrt{1+a^2})$, $(b, \sqrt{1+b^2})$, we can compute $\left(a*b, \sqrt{1+(a*b)^2}\right)$ with 4 multiplications and 2 additions in K. With $a = b$ in ($\sharp$), we obtain:

$$\forall a \in P,\ \sqrt{(2a^2+1)^2} = 2a^2+1\ (\natural)$$

2.3 A Toy Example for $(P, *)$

Here we have $K = \mathbb{Z}/19\mathbb{Z}$ with $p = 19$ ($p \equiv 3 \pmod 4$ as wanted). The set of all the squares of K is $C = \{0, 1, 4, 5, 6, 7, 9, 11, 16, 17\}$.

$\forall a \in K, a^2+1$ is a square $\Leftrightarrow a^2 \in \{0, 4, 5, 6, 16\} \Leftrightarrow a \in P$ with $P = \{0, 2, 4, 5, 9, 10, 14, 15, 17\}$. We denote by P this set. Therefore in P we have 9 values (i.e. $\frac{p-1}{2}$ values). For example, let assume that we want to compute $5*9$. We have: $5*9 = 5\sqrt{82}+9\sqrt{26} = 5\sqrt{6}+9\sqrt{7}$. Now $\sqrt{6}$ can be 5 or 14, and $\sqrt{7}$ can be 8 or 11, so for $5*9$ we have 4 possibilities here. In order to see what the exact values are for $\sqrt{6}$ and $\sqrt{7}$, we use the formula: $\forall a \in P,\ \sqrt{(2a^2+1)^2} = 2a^2+1$ ($\natural$). To compute $\sqrt{6}$, we first solve the equation $(2a^2+1)^2 = 6$. This gives $2a^2+1 = 5$ or 14, thus $2a^2 = 4$ or 13. Since $2^{-1} = 10 \pmod{19}$), we obtain $a^2 = 40$ or 130, i.e. $a^2 = 2$ or 16. This gives $a = 4$ or 15. Now, ($\natural$) with $a = 4$ (or 15) gives: $\sqrt{6} = 14$.

Similarly, to compute $\sqrt{7}$ we first solve the equation $(2a^2+1)^2 = 7$. This gives $2a^2+1 = 11$ or 8. Thus we have $2a^2 = 10$ or 17 and $a^2 = 5$ or 13. Thus $a = 9$ or 10. Now ($\natural$) with $a = 9$ (or 10) gives: $\sqrt{7} = 11$. Finally $5*9 = 5\sqrt{6}+9\sqrt{7} = 17$. All the values $a*b$ with $a, b \in P$ can be computed in the same way. We obtain like this the table below of the group $(P, *) = P(\mathbb{Z}/19\mathbb{Z})$.

2.4 A More General Context

Definition and Properties. The Domino Law can be defined also on $P \times P$. It is still associative (the proof is similar to the one given for $\mathbb{R}^2$) (Table 1).

Proposition 2. *Let $(a, b) \in P \times P$, then $(a, b) \boxminus (a, b) = (2ab, a^2+b^2)$. If $(a, b)^2_{\boxminus} = (A, B)$, then $A + B = (a+b)^2$.*

More generally, $\forall k \in \mathbb{N}$, if *$(a, b)^k_{\boxminus} = (A, B)$ then $A + B = (a+b)^k$.*

Proof. For $k = 2$, the computation is straightforwards. Then, the proof is done by induction.

Corollary 2. *Proposition 2 shows that computing logarithms in $(P \times P, \boxminus)$ is equivalent to computing logarithms in $(K, .)$*

Proof. The proof is obvious.

Table 1. $P(\mathbb{Z}/19\mathbb{Z})$

$*$	0	2	4	5	9	10	14	15	17
0	0	2	4	5	9	10	14	15	17
2	2	17	5	10	14	4	15	9	0
4	4	5	9	14	2	15	17	0	10
5	5	10	14	15	17	9	0	2	4
9	9	14	2	17	5	0	10	4	15
10	10	4	15	9	0	14	2	17	5
14	14	15	17	0	10	2	4	5	9
15	15	9	0	2	4	17	5	10	14
17	17	0	10	4	15	5	9	14	2

Application to $a * b = a\sqrt{1+b^2} + b\sqrt{1+a^2}$

Proposition 3. *We have:* $(a, \sqrt{1+a^2}) \boxminus (b, \sqrt{1+b^2}) = \left(a * b, \sqrt{1 + (a*b)^2}\right)$.

Hence $\forall k,\ (a, \sqrt{1+a^2})^k_{\boxminus} = (a^k_*, \sqrt{1+(a^k_*)^2})$

Corollary 3. *This proposition shows that computing logarithms in* $(P, *)$ *is equivalent to computing logarithms in* $(K, .)$*.*

Proof. We want to compute k such that $a^k_* = \alpha$ (a and α are known). We first choose β such that $\beta^2 = \alpha^2 + 1$. Then, we want to find (a, b) satisfying $b^2 = a^2 + 1$ such that $(a, b)^k_{\boxminus} = (A, B)$. Since $\alpha + \beta = (a + b)^k$, this equation gives k by using the discrete log.

Therefore the cryptographic scheme based on $(P, *)$ is essentially similar to the classical cryptographic scheme based on discrete logarithms on $(K, .)$.

3 Associativity Based on the Hyperbolic Tangent

3.1 The General Case

In this section, we will use the tanh function to obtain associativity. This function is a bijection from $\mathbb{R}$ to $]-1, 1[$ and we have the formula

$$\tanh(a + b) = \frac{\tanh a + \tanh b}{1 + \tanh a \tanh b}$$

Thus if we define on $]-1, 1[$ the following law: $a * b = (a + b)(1 + ab)^{-1}$ we obtain a group since tanh is an isomorphism from $(\mathbb{R}, +)$ to $(]-1, 1[, *)$. Similarly, we will work on finite fields. Let K be a finite field. We suppose that in K, -1 is not a square. When we can perform the computation (i.e. when $ab \neq -1$), we define:

$$a * b = (a + b)(1 + ab)^{-1}$$

We have the following properties:

Proposition 4. *1.* $\forall a \in K,\ a * 0 = a$.
2. $\forall a \in K \setminus \{-1\}$, $a * 1 = 1$ *and* $\forall a \in K \setminus \{1\}$, $a * (-a) = 0$.
3. $\forall a, b,\ ab \neq -1,\ (-a) * (-b) = -(a * b)$.
4. $\forall a, b, c$, $(a * b) * c = a * (b * c)$ *when the computation is possible, i.e.* $*$ *is associative.*

Proof. Properties 1, 2 and 3 are straightforward. We will prove that $*$ is associative.

$$(a * b) * c = [(a + b)(1 + ab)^{-1} + c][1 + (a + b)(1 + ab)^{-1}c]^{-1}$$

We multiply by $(1 + ab)(1 + ab)^{-1}$. This gives:

$$(a * b) * c = [((a + b)(1 + ab)^{-1} + c)(1 + ab)][(1 + (a + b)(1 + ab)^{-1}c)(1 + ab)]^{-1}$$

$$(a * b) * c = [a + b + c + abc][(1 + ab + bc + ac]^{-1}$$

Similarly

$$a * (b * c) = [a + (b + c)(1 + bc)^{-1}][1 + a(b + c)(1 + bc)^{-1}]^{-1}$$

Here we multiply by $(1 + bc)(1 + bc)^{-1}$ and we obtain

$$a * (b * c) = [a + b + c + abc][(1 + ab + bc + ac]^{-1}$$

Remark 4. There is an analog with the addition law of the speed in simple relativity: $\frac{v_1+v_2}{1+\frac{v_1 v_2}{c}}$. From this, it is also possible to justify associativity from physical considerations.

3.2 A Toy Example

In Table 2, we give the example of the construction of a group denoted $(Q(K), *)$ when $K = \mathbb{Z}/19\mathbb{Z}$ and $*$ is the law based on the tanh function. Here -1 is not a square since $19 \equiv 3 \pmod 4$. We already know that 1 and 18 are not elements of $Q(K)$. When we do the computations, we obtain that for $Q(K) = \{0, 2, 3, 4, 7, 12, 15, 16, 17\}$. We also have that $Q(K) = \langle 3 \rangle$.

3.3 Computing Log with $*$ (Analog of tanh)

We will now study the power for $*$ of an element of K. We will use the following notation: $a_*^k = \underbrace{a * a * \ldots * a}_{k \text{ times}}$.

Proposition 5. *Suppose that we can perform the computations (i.e. we never obtain the value* -1 *during the computations).* $\forall a \in K, \forall k$, $a_*^k = s_k t_k^{-1}$ *with* $s_k = (1 + a)^k - (1 - a)^k$ *and* $t_k = (1 + a)^k + (1 - a)^k$*. Then* $s_k + t_k = 2(1 + a)^k$.

Table 2. $(Q(\mathbb{Z}/19\mathbb{Z}), *)$

$*$	0	2	3	4	7	12	15	16	17
0	0	2	3	4	7	12	15	16	17
2	2	16	17	7	12	15	3	4	0
3	3	17	12	2	16	4	7	0	15
4	4	7	2	15	3	17	0	12	16
7	7	12	16	3	17	0	2	15	4
12	12	15	4	17	0	2	16	3	7
15	15	3	7	0	2	16	4	17	12
16	16	4	0	12	15	3	17	7	2
17	17	0	15	16	4	7	12	2	3

Proof. We have $a_*^1 = a * 0$. Then $a_*^2 = a * a = 2a(1+a^2)^{-1}$. Since $s_2 = 2a$ and $t_2 = 2(1+a^2)$, we have $a_*^2 = s_1 t_2^{-1}$. Suppose that $a_*^{k-1} = s_{k-1}t_{k-1}^{-1}$. Then $a_*^k = a * a_*^{k-1} = (a + s_{k-1}s_{k-1}^{-1})(1 + as_{k-1}s_{k-1})^{-1}$. We multiply this expression by $t_{k-1}t_{k-1}^{-1}$. We obtain that $a_*^k = s_k s_k^{-1}$ with $s_k = at_{k-1} + s_{k-1}$ and $t_k = t_{k-1} + as_{k-1}$. Thus we can write:

$$\begin{bmatrix} s_k \\ t_k \end{bmatrix} = \begin{bmatrix} 1 & a \\ a & 1 \end{bmatrix} \begin{bmatrix} s_{k-1} \\ t_{k-1} \end{bmatrix}$$

This gives:

$$\begin{bmatrix} s_k \\ t_k \end{bmatrix} = A^{k-1} \begin{bmatrix} s_1 \\ t_1 \end{bmatrix}$$

with $A = \begin{bmatrix} 1 & a \\ a & 1 \end{bmatrix}$. By diagonalizing the matrix A, we obtain that:

$$a_*^k = s_k t_k^{-1} \text{ with } s_k = (1+a)^k - (1-a)^k \text{ and } t_k = (1+a)^k + (1-a)^k$$

Then we get $u_k + v_k = (1+a)^k$. This can also be proved by induction.

Corollary 4. *If a_*^k exists, then $(-a)_*^k = -a_*^k$.*

Corollary 5. *Let $a \in K$.*

1. *If there exists $k(a) \in \mathbb{N}^*$ such that $\forall k < k(a)$, $s_k \neq 0, t_k \neq 0$ and $s_{k(a)} = 0, t_{k(a)} \neq 0$, then $(\langle a \rangle, *)$ is a group.*
2. *If there exists $k'(a) \in \mathbb{N}^*$ such that $\forall k < k'(a)$, $s_k \neq 0, t_k \neq 0$ and $t_{k'(a)} = 0$, then a does not generate a group.*

We recall the results obtained in Proposition 5: $\forall a \in K$, $\forall k$, $a_*^k = s_k t_k^{-1}$ with $s_k = (1+a)^k - (1-a)^k$ and $t_k = (1+a)^k + (1-a)^k$. Then $s_k + t_k = 2(1+a)^k$. Let $\alpha = a_*^k$. It is possible to compute k from α and a like this:

$$\alpha = a_*^k = s_k t_k^{-1} = \frac{(1+a)^k - (1-a)^k}{(1+a)^k + (1-a)^k}$$

$$\alpha = \frac{1 - \left(\frac{1-a}{1+a}\right)^k}{1 + \left(\frac{1-a}{1+a}\right)^k}$$

Then we can find $\left(\frac{1-a}{1+a}\right)^k$ and finally we obtain k by using the discrete log. This shows that computing logarithms for the $*$ law is essentially the same as for the classical case. Therefore the cryptographic scheme based on this law $*$ (analog to tanh) is again essentially similar to the classical cryptographic scheme based on the discrete logarithm.

4 Widen the Range

As pointed out by Jérôme Plût to us, it seems that there is a little hope to find "magic algebraic curves" that are more efficient than elliptic curves. In particular, our curve $b^2 = a^2 + 1$ had little chance to be useful due to general results on the classification of algebraic groups. For any abelian algebraic group, there exist unique decompositions:

- $0 \to G^0 \to G \to \pi_0(G) \to 0$ where G^0 is connexe and $\pi(G)$ is étale.
- $0 \to L \to G^0 \to A \to 0$ where A is an abelian variety and L is a linearizable group.
- $0 \to U \to L \to T \to 0$ where T is a torus, and U is unipotent.

The first and the third decompositions are rather simple. The second one is more complicated and can be found in [1].

Therefore the only possibility to get more efficient systems is to use curves of genus larger than 1 and varieties related to their Jacobians, which are accessible to effective computation. But because of security reasons it is very doubtful that one can use curves of genus larger than 3 (key word: index-calculus). As said already in Remark 2, Theta functions lead to the very efficient Kummer surfaces for $g = 2$.

Part II: Commutative Properties on the Functions

5 Chebyshev Polynomials

To generalize the Diffie-Hellman Algorithm by using $(f \circ g)(a) = (g \circ f)(a)$, we want:

- f and g to be one way
- f and g to be easy to compute
- $f \circ g = g \circ f$, i.e. commutativity.

The value a is typically between 80 and 2048 bits (as in Sect. 2). Ironically, here (unlike in Part I) associativity is very easy, since $\circ$ is always associative, but we want commutativity on f and g, and this is not easy to obtain. In part I, we had a law $*$ on elements of G with about 160 bits, but here, we work with functions f and g on G and we have more functions from G to G than elements of G. Moreover $a_*^i = \underbrace{a * a * \ldots * a}_{i \text{ times}}$ can be computed in $O(\ln i)$ with square and multiply, while $f^i(a) = f[f \ldots f(a))]$ would generally require $O(i)$ computations of f. An interesting idea is to use the Chebyshev polynomials (cf. [2,8,10–12,15] for example). In [14], the structure of Chebyshev polynomials on $\mathbb{Z}/p\mathbb{Z}$ is also studied. However, as mentioned in some of these papers, and as we will see below, public key schemes based on Chebyshev polynomials have often exactly the same security than public key schemes based on monomials. We will present here only a few properties.

Some Properties of Chebyshev Polynomials on $\mathbb{R}$

The Chebyshev polynomials T_n can be defined as the polynomials such that:

$$\cos nx = T_n(\cos x) \tag{1}$$

Since $\cos a + \cos b = 2\cos(\frac{a+b}{2})\cos(\frac{a-b}{2})$, we have: $\cos(n+1)x + \cos(n-1)x = 2\cos x \cos nx$, and therefore we have:

$$T_{n+1}(X) = 2XT_n(X) - T_{n-1}(X). \tag{2}$$

For example, the first polynomials are: $T_0 = 1$, $T_1 = X$, $T_2 = 2X^2 - 1$, $T_3 = 4X^3 - 3X$, $T_4 = 8X^4 - 8X^2 + 1$. From 1, we can see that the Chebyshev polynomials commute: $(T_n(T_m(X)) = T_m(T_n(X))$ since $\cos(nm)x = \cos(mn)x$. Therefore, we can design analog of the Diffie-Hellman or RSA schemes by using Chebyshev polynomials instead of the monomial transformation $X \mapsto X^a$. Moreover, from 2, we can write:

$$\begin{bmatrix} T_n(X) \\ T_{n+1}(X) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 2X \end{bmatrix} \begin{bmatrix} T_{n-1}(X) \\ T_n(X) \end{bmatrix}$$

and this gives

$$\begin{bmatrix} T_n(X) \\ T_{n+1}(X) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 2X \end{bmatrix}^n \begin{bmatrix} 1 \\ X \end{bmatrix} \tag{3}$$

Now from 3 we can obtain:

$$T_n(X) = U \circ X^n \circ U^{-1} \tag{4}$$

with $U(X) = \frac{X + \frac{1}{X}}{2}$ if $X \neq 0$ and $U^{-1}(X) = X + \sqrt{X^2 - 1}$ if $X > 1$. Therefore, if $|X| \geq 1$, $T_n(X) = \frac{1}{2}\left(\left(X - \sqrt{X^2-1}\right)^n + \left(X + \sqrt{X^2-1}\right)^n\right)$. Property (4) is very nice since it shows that we can compute $T_n(X)$ about as fast as a X^n (and we use an analog of the square and multiply algorithm), so we can compute

$T_n(X)$ efficiently even when n has a few hundred or thousands of bits. However, property 4 also shows that $T_n(X)$ and X^n are essentially the same operation since U and U^{-1} can be considered as public.

Properties of Chebyshev Polynomials on Other Spaces.
For cryptographic use, it has been suggested to use Chebyshev polynomials on various spaces. In fact, it could be assumed that the analysis of Chebyshev polynomials properties for cryptography would depend on the type of space where the computations are done (finite fields with characteristic equal or not equal to 2, computations modulo n with n prime or not prime, etc.). However, most of the time, the above properties on real numbers suggest that public key cryptography based on Chebyshev polynomials is essentially the same as (classical) public key cryptography based on X^n (see [8,10–12,14,15] for details).

Remark 5. After our presentation at the NuTMiC conference (Warsaw 2017), Gérard Maze pointed out to us that in his PhD Thesis (Chap. 6) [13], he had also studied how to use Chebyshev polynomials for public key cryptography. His conclusions were similar to ours, i.e. when the Chebyshev polynomials are properly used, the resulting schemes are essentially the same as schemes based on discrete log.

6 Commutativity with Other Polynomials

We first give the definition of a commutative family of polynomials.

Definition 4. *Let (Q_n) be a family of polynomials. We say that we have a family of polynomials that commute if $\forall n$, $\forall m$, $Q_n \circ Q_m = Q_m \circ Q_n$.*

If we look for infinite family of polynomials satisfying commutativity, the Block and Thielman theorem [3] shows that we do not have many solutions. More precisely:

Theorem 8 *(Bloch and Thielman 1951)***.** *Let (Q_n) be a polynomial of degree n. If $(Q_n)_{n\geq 1}$ is a family of polynomials that commute, then there exists a polynomial of degree 1, U, such that, either for all n, $Q_n = U \circ X^n \circ U^{-1}$ or for all n, $Q_n = U \circ T_n \circ U^{-1}$, where T_n is the Chebyshev polynomial of degree n.*

For cryptographic use, we may look for "sufficiently large" families of polynomials that commute (instead of "infinite families") but it seems difficult to find new large families. Some suggestions are given in [13], but more possibilities should exist and could be the subject of further work.

7 Conclusion

In this paper, we investigated several methods to construct algebraic generalizations of the Diffie-Hellman key exchange algorithm. However, after our analysis, it appears that the proposed schemes are essentially equivalent to the classical ones. Nevertheless, the study showed that there are interesting connections between associativity, commutativity and the construction of such algorithms. We also explained that there is little hope to find “magic algebraic curves” more efficient than elliptic curves and we suggested to study “large” but not infinite families of polynomials that commute for further analysis.

Acknowledgment. The authors want to thank Jérôme Plût, Gerhard Frey and Gérard Maze for very useful comments and particularly Gerhard Frey for his help to improve the presentation of this paper. We have met Gerhard Frey and Gérard Maze at NuTMiC 2017 in Poland.

References

1. Barsotti, I.: Un Teorema di structura per le varietà di gruppali. Rend. Acc. Naz. Lincei **18**, 43–50 (1955)
2. Bergamo, P., D’Arco, P., de Santis, A., Kocarev, L.: Security of Public Key Cryptosystems based on Chebyshev Polynomials. arXiv:cs/0411030v1, 1 February 2008
3. Block, H.D., Thielman, H.P.: Commutative Polynomials. Quart. J. Math. Oxford Ser. **2**(2), 241–243 (1951)
4. Couveignes, J.M.: Hard Homogeneous Spaces. Cryptology ePrint archive: 2006/291: Listing for 2006
5. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Trans. Inf. Theor. **22**(6), 644–654 (1976)
6. Frey17: Deep Theory, efficient algorithms and surprising applications. In: NuTMiC (2017)
7. Gaudry, P., Lubicz, D.: The arithmetic of characteristic 2 Kummer surfaces and of elliptic Kummer lines. Finite Fields Appl. **15**(2), 246–260 (2009)
8. Hunziker, M., Machiavelo, A., Parl, J.: Chebyshev polynomials over finite fields and reversibility of σ-automata on square grids. Theor. Comput. Sci. **320**(2–3), 465–483 (2004)
9. Jao, D., De Feo, L.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: Yang, B.-Y. (ed.) PQCrypto 2011. LNCS, vol. 7071, pp. 19–34. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25405-5_2
10. Kocarev, L., Makraduli, J., Amato, P.: Public key encryption based on Chebyshev polynomials. Circ. Syst. Sign. Process. **24**(5), 497–517 (2005)
11. Li, Z., Cui, Y., Jin, Y., Xu, H.: Parameter selection in public key cryptosystem based on Chebyshev polynomials over finite field. J. Commun. **6**(5), 400–408 (2011)
12. Lima, J.B., Panario, D., Campello de Sousa, R.M.: Public-key cryptography based on Chebyshev polynomials over $G(q)$. Inf. Process. Lett. **111**, 51–56 (2010)
13. Maze, G.: Algebraic Methods for Constructing One-Way Trapdoor Functions. Ph.D. thesis - University oof Notre Dame (2003). http://user.math.uzh.ch/maze/
14. Rosen, J., Scherr, Z., Weiss, B., Zieve, M.: Chebyshev mappings over finite fields. Amer. Math. Monthly **119**, 151–155 (2012)
15. Sun, J., Zhao, G., Li, X.: An improved public key encryption algorithm based on Chebyshev polynomials. TELKOMNIKA **11**(2), 864–870 (2013)

Lattices in Cryptography

Computational Differential Privacy from Lattice-Based Cryptography

Filipp Valovich(✉) and Francesco Aldà(✉)

Faculty of Mathematics, Horst Görtz Institute for IT Security,
Ruhr-Universität Bochum, Universitätsstrasse 150, 44801 Bochum, Germany
{filipp.valovich,francesco.alda}@rub.de

Abstract. In this work we investigate the problem of private statistical analysis of time-series data in the distributed and semi-honest setting. In particular, we study some properties of Private Stream Aggregation (PSA), first introduced by Shi et al. 2011. This is a computationally secure protocol for the collection and aggregation of data in a distributed network and has a very small communication cost. In the non-adaptive query model, a secure PSA scheme can be built upon any key-homomorphic *weak* pseudo-random function as shown by Valovich 2017, yielding security guarantees in the *standard model* which is in contrast to Shi et al. We show that every mechanism which preserves (ϵ, δ)-differential privacy in effect preserves *computational* (ϵ, δ)-differential privacy when it is executed through a secure PSA scheme. Furthermore, we introduce a novel perturbation mechanism based on the *symmetric Skellam distribution* that is suited for preserving differential privacy in the distributed setting, and find that its performances in terms of privacy and accuracy are comparable to those of previous solutions. On the other hand, we leverage its specific properties to construct a computationally efficient prospective post-quantum protocol for differentially private time-series data analysis in the distributed model. The security of this protocol is based on the hardness of a new variant of the Decisional Learning with Errors (DLWE) problem. In this variant the errors are taken from the symmetric Skellam distribution. We show that this new variant is hard based on the hardness of the standard Learning with Errors (LWE) problem where the errors are taken from the discrete Gaussian distribution. Thus, we provide a variant of the LWE problem that is hard based on conjecturally hard lattice problems and uses a discrete error distribution that is similar to the continuous Gaussian distribution in that it is closed under convolution. A consequent feature of the constructed prospective post-quantum protocol is the use of the same noise for security and for differential privacy.

1 Introduction

The framework of statistical disclosure control aims at providing strong privacy guarantees for the records stored in a database while enabling accurate

The research was supported by the DFG Research Training Group GRK 1817/1.

J. Kaczorowski et al. (Eds.): NuTMiC 2017, LNCS 10737, pp. 121–141, 2018.
https://doi.org/10.1007/978-3-319-76620-1_8

statistical analyses to be performed. In recent years, *differential privacy* has become one of the most important paradigms for privacy-preserving statistical analyses. According to Nissim, a pioneer in this area of research, "there is a great promise for the marriage of Big Data and Differential Privacy".[1] It combines mathematically rigorous privacy guarantees with highly accurate analyses over larger data sets. Generally, the notion of differential privacy is considered in the centralised setting where we assume the existence of a *trusted curator* (see Blum et al. [6], Dwork [11], Dwork et al. [13], McSherry and Talwar [19]) who collects data in the clear, aggregates and perturbs it properly (e.g. by adding Laplace noise) and publishes it. In this way, the output statistics are not significantly influenced by the presence (resp. absence) of a particular record in the database.

In this work we study how to preserve differential privacy when we cannot rely on a trusted curator. In this so-called *distributed setting*, the users have to send their own data to an untrusted aggregator. Preserving differential privacy and achieving high accuracy in the distributed setting is of course harder than in the centralised setting, since the users have to execute a perturbation mechanism on their own. In order to achieve the same accuracy as provided by well-known techniques in the centralised setting, Shi et al. [26] introduce the *Private Stream Aggregation* (PSA) scheme, a cryptographic protocol enabling each user to securely send encrypted time-series data to an aggregator. The aggregator is then able to decrypt the aggregate of all data in each time-step, but cannot retrieve any further information about the individual data. Using such a protocol, the task of perturbation can be split among the users, such that *computational* differential privacy, a notion first introduced by Mironov et al. [22], is preserved *and* high accuracy is guaranteed. For a survey of applications, we refer to [26].

Related Work. In [26], a PSA scheme for sum queries was provided that satisfies strong security guarantees under the Decisional Diffie-Hellman (DDH) assumption. However, this instantiation has some limitations. First, the security only holds in the random oracle model; second, its decryption algorithm requires the solution of the discrete logarithm in a given range, which can be very time-consuming if the number of users and the plaintext space are large. Third, a connection between the security of a PSA scheme and computational differential privacy is not explicitly shown. In a subsequent work by Chan et al. [8], this connection is still not completely established.

By lowering the requirements of Aggregator Obliviousness introduced in [26] by abrogating the attacker's possibility to *adaptively compromise* users during the execution of a PSA scheme with time-series data, Valovich [28] shows that a PSA scheme achieving this lower security level can be built upon any *key-homomorphic weak pseudo-random function*. Since weak pseudo-randomness can be achieved in the standard model, this condition also enables secure schemes in the standard model. Furthermore, an instantiation of this result based on the DDH assumption was given in [28], where decryption is always efficient.

[1] http://bigdata.csail.mit.edu/Big_Data_Privacy.

Joye and Libert [16] provide a protocol with the same security guarantees in the random oracle model as in [26]. The security of their scheme relies on the Decisional Composite Residuosity assumption (rather than DDH as in [26]) and as a result, in the security reduction they can remove a factor which is cubic in the number of users. However, their scheme involves a semi-trusted party for setting some public parameters. In this work, we provide an instantiation of the *generic* PSA construction from [28] which relies on the Decisional Learning with Errors assumption. While in this generic security reduction a *linear* factor in the number of users cannot be avoided, our construction *does not* involve any trusted party and has security guarantees in the standard model. In a subsequent work [5], a generalisation of the scheme from [16] is obtained based on smooth projective hash functions (see [9]). This generalisation allows the construction of secure protocols based on various hardness assumptions. However, the dependencies on a semi-trusted party (for most of the instantiations) and on a random oracle remain.

Contributions. In this regard, our results are as follows. First, reduction-based security proofs for cryptographic schemes usually require an attacker in the corresponding security game to send two different plaintexts (or plaintext collections) to a challenger. The adversary receives then back a ciphertext which is the encryption of one of these collections and has to guess which one it is. In any security definition for a PSA scheme, these collections must satisfy a particular requirement, i.e. they must lead to the same aggregate, since the attacker has the capability to decrypt the aggregate (different aggregates would make the adversary's task trivial). In general, however, this requirement cannot be satisfied in the context of differential privacy. Introducing a novel kind of security reduction which deploys a *biased coin* flip, we show that, whenever a randomised perturbation procedure is involved in a PSA scheme, the requirement of having collections with equal aggregate can be abolished. Using this property, we are able to show that if a mechanism preserves differential privacy, then it preserves computational differential privacy when it is composed with a secure PSA scheme. This provides the missing step in the analysis from [26].

Second, we introduce the *Skellam mechanism* that uses the symmetric Skellam distribution and compare it with the geometric mechanism by Ghosh et al. [14] and the binomial mechanism by Dwork et al. [12]. All three mechanisms preserve differential privacy and make use of discrete probability distributions. Therefore, they are well-suited for an execution through a PSA scheme. For generating the right amount of noise among all users, these mechanisms apply two different approaches. While in the geometric mechanism, with high probability, only one user generates the noise necessary for differential privacy, the binomial and Skellam mechanisms allow all users to generate noise of small variance, that sums up to the required value for privacy by the reproducibility property of the binomial and the Skellam distributions. We show that the theoretical error bound of the Skellam mechanism is comparable to the other two. At the same time, we provide experimental results showing that the geometric and Skellam mechanisms have a comparable accuracy in practice, while beating the one of the binomial mechanism. The advantage of the Skellam mechanism is that, based on the previously mentioned results, it can be used it to construct a secure, prospective post-quantum PSA scheme for sum queries that automatically preserves computational differential privacy. The corresponding weak pseudo-random function for this protocol is constructed from the *Learning with Errors* (LWE) problem. Regev [25] provided a worst-case search-to-decision reduction and Micciancio and Mol [21] provided a sample preserving search-to-decision reduction for certain cases in the average case. Moreover, in [25] the average-case-hardness of the search-version of the LWE problem was established by the construction of an efficient quantum algorithm for worst-case lattice problems using an efficient solver of the LWE problem if the error distribution χ is a *discrete Gaussian* distribution. Accordingly, most cryptographic applications of the LWE problem used a discrete Gaussian error distribution for their constructions. We will take advantage of the reproducibility of the Skellam distribution for our DLWE-based PSA scheme by using errors following the symmetric Skellam distribution rather than the discrete Gaussian distribution, which is not reproducible. The result is that the sum of the errors generated by every user to secure their data is also a Skellam variable and therefore sufficient for preserving differential privacy. Hence, we show the average-case-hardness of the LWE problem with errors drawn from the Skellam distribution. Our proof is inspired by techniques used in [10], where a variant of the LWE problem with uniform errors on a small support is shown to be hard.[2] Consequently, we obtain a lattice-based secure PSA scheme for analysing sum queries under differential privacy where the noise is used both for security and for preserving differential privacy at once.

Other Related Work. A series of works deals with a distributed generation of noise for preserving differential privacy. Dwork et al. [12] consider the Gaussian distribution for splitting the task of noise generation among all users. In [2], the generation of Laplace noise is performed in a distributed manner by generating the difference of two Gamma distributed random variables as a share of a Laplace distributed random variable. In [24], each user generates a share of Laplace noise by generating a vector of four Gaussian random variables. However, the aforementioned mechanisms generate noise drawn according to continuous distributions, but for the use in a PSA scheme discrete noise is required. Therefore, we consider proper discrete distributions.

2 Preliminaries

Notation 1. *Let $q > 2$ be a prime. We handle elements from $\mathbb{Z}_q$ as their central residue-class representation. This means that $x' \in \mathbb{Z}_q$ is identified with $x \equiv x' \bmod q$ for $x \in \{-(q-1)/2, \ldots, (q-1)/2\}$ thereby lifting x' from $\mathbb{Z}_q$ to $\mathbb{Z}$.*

[2] Although the uniform distribution is reproducible as well, the result from [10] does not provide a proper error distribution for our DLWE-based PSA scheme, since a differentially private mechanism with uniform noise provides no accuracy to statistical data analyses.

2.1 Problem Statement

In this work, we consider a distributed and semi-honest setting where n users are asked to participate in some statistical analyses but do not trust the data analyst/aggregator, who is honest but curious. Therefore, the users cannot provide their own data in the clear. Moreover, they communicate solely and independently with the untrusted aggregator, who wants to analyse the users data by means of time-series queries and aims at obtaining answers as accurate as possible. For a sequence of time-steps $t \in T$, where T is a discrete time period, the analyst sends queries which are answered by the users in a distributed manner.

We also assume that some users may act in order to compromise the privacy of the other participants. More precisely, we assume the existence of a publicly known constant $\gamma \in (0, 1]$ which is the a priori estimate of the lower bound on the fraction of uncompromised users who honestly follow the protocol and want to release useful information about their data (with respect to a particular query f), while preserving (ϵ, δ)-differential privacy. The remaining $(1-\gamma)$-fraction of users is compromised and aims at violating the privacy of uncompromised users, but honestly follows the protocol. For that purpose, these users form a coalition with the analyst and send her auxiliary information, e.g. their own secrets.

For computing the answers to the aggregator's queries, a special cryptographic protocol, called Private Stream Aggregation (PSA) scheme, is used by *all* users. In contrast to common secure multi-party techniques (see [15,18]), this protocol requires each user to send only one message per query to the analyst. In connection with a differentially private mechanism, a PSA scheme assures that the analyst is only able to learn a noisy aggregate of users' data (as close as possible to the real answer) and nothing else. Specifically, for preserving (ϵ, δ)-differential privacy, it would be sufficient to add a single copy of (properly distributed) noise Y to the aggregated statistics. Since we cannot add such noise once the aggregate has been computed, the users have to generate and add noise to their original data in such a way that the sum of the errors has the same distribution as Y. For this purpose, we see two different approaches. Firstly, with small probability a user adds noise sufficient to preserve the privacy of the entire statistics. This probability is calibrated in such a way only one of the n users is actually expected to add noise at all. Shi et al. [26] investigate this method using the geometric mechanism from [14]. Secondly, each user generates noise of small variance, such that the sum of all noisy terms suffices to preserve differential privacy of the aggregate. The binomial mechanism from [12] and the Skellam mechanism from this work serve these purposes.[3] Since the protocol used for the data transmission is *computationally* secure, the entire mechanism preserves a computational version of differential privacy as described in Sect. 4.

[3] Due to the use of a cryptographic protocol, the plaintexts have to be discrete. This is the reason why we use discrete distributions for generating noise.

2.2 Definitions

Differential Privacy. We will always assume that a differentially private mechanism is applied in the distributed setting. We recall that a randomised mechanism preserves differential privacy if its application on two adjacent databases (databases differing in one entry only) leads to close distributions of the outputs.

Definition 1 (Differential Privacy [13]). *Let $\mathcal{R}$ be a (possibly infinite) set and let $n \in \mathbb{N}$. A randomised mechanism $\mathcal{A} : \mathcal{D}^n \to \mathcal{R}$ preserves (ϵ, δ)-differential privacy (short:* DP*), if for all adjacent databases $D_0, D_1 \in \mathcal{D}^n$ and all measurable $R \subseteq \mathcal{R}$: $\Pr[\mathcal{A}(D_0) \in R] \leq e^{\epsilon} \cdot \Pr[\mathcal{A}(D_1) \in R] + \delta$.*

The probability space is defined over the randomness of $\mathcal{A}$.

Thus, the presence or absence of a single user does not affect the probability of any outcome by too much. The aim of the analyst is to obtain information from the database. Therefore it processes queries to the database which are answered while preserving DP. In the literature, there are well-established mechanisms for preserving DP (see [13,19]).[4] In order to privately evaluate a query, these mechanisms draw error terms according to some distribution depending on the query's global sensitivity. For any $D \in \mathcal{D}^n$, the global sensitivity $S(f)$ of a query $f : \mathcal{D}^n \to \mathbb{R}$ is defined as the maximum change (in terms of the L_1-norm) of $f(D)$, which can be produced by a change of one entry (i.e. the absence of one user) in D. In particular, we will consider sum-queries $f_{\mathcal{D}} : \mathcal{D}^n \to \mathbb{Z}$ or $f_{\mathcal{D}} : \mathcal{D}^n \to [-m', m']$ for some integer m' defined as $f_{\mathcal{D}}(D) := \sum_{i=1}^{n} d_i$, for $D = (d_1, \ldots, d_n) \in \mathcal{D}^n$ and $\mathcal{D} \subseteq \mathbb{Z}$. If the entries in D are bounded by m, then $S(f_{\mathcal{D}}) \leq m$. For measuring how well the output of a mechanism $\mathcal{A}$ estimates the real data with respect to a particular query f (mapping into a metric space), we use the notion of (α, β)-accuracy, defined as $\Pr[|\mathcal{A}(D) - f(D)| \leq \alpha] \geq 1 - \beta$.

The use of a cryptographic protocol for transferring data provides a computational security level. If such a protocol is applied to preserve DP, this implies that only a computational level of DP can be provided. The definition of computational differential privacy was first provided in [22] and extended in [8].

Definition 2 (Computational Differential Privacy [8]). *Let κ be a security parameter and $n \in \mathbb{N}$ with $n = \mathsf{poly}(\kappa)$. A randomised mechanism $\mathcal{A} : \mathcal{D}^n \to \mathcal{R}$ preserves computational (ϵ, δ)-differential privacy (short:* CDP*), if for all adjacent databases $D_0, D_1 \in \mathcal{D}^n$ and all probabilistic polynomial-time distinguishers $\mathcal{D}_{\mathsf{CDP}}$: $\Pr[\mathcal{D}_{\mathsf{CDP}}(1^{\kappa}, \mathcal{A}(D_0)) = 1] \leq e^{\epsilon} \cdot \Pr[\mathcal{D}_{\mathsf{CDP}}(1^{\kappa}, \mathcal{A}(D_1)) = 1] + \delta + \mathsf{neg}(\kappa)$, where $\mathsf{neg}(\kappa)$ is a negligible function in κ. The probability space is defined over the randomness of $\mathcal{A}$ and $\mathcal{D}_{\mathsf{CDP}}$.*

Private Stream Aggregation. We define the Private Stream Aggregation scheme and give a security definition for it. Thereby, we mostly follow the concepts introduced in [26], though we deviate in a few points. A PSA scheme is a

[4] These mechanisms work in the centralised setting, where a *trusted curator* sees the full database in the clear and perturbs it properly.

protocol for safe distributed time-series data transfer which enables the receiver (here: the untrusted analyst) to learn nothing else than the sums $\sum_{i=1}^{n} x_{i,j}$ for $j = 1, 2, \ldots$, where $x_{i,j}$ is the value of the ith participant in time-step j and n is the number of participants (or users). Such a scheme needs a key exchange protocol for all n users together with the analyst as a precomputation (e.g. using multi-party techniques), and requires each user to send exactly one message in each time-step $j = 1, 2, \ldots$.

Definition 3 (Private Stream Aggregation [26]**).** *Let κ be a security parameter, $\mathcal{D}$ a set and $n = poly(\kappa)$, $\lambda = poly(\kappa)$. A* Private Stream Aggregation *(PSA) scheme $\Sigma = (\mathsf{Setup}, \mathsf{PSAEnc}, \mathsf{PSADec})$ is defined by three ppt algorithms:*

Setup: $(\mathsf{pp}, T, s_0, s_1, \ldots, s_n) \leftarrow \mathsf{Setup}(1^\kappa)$ *with public parameters* pp*, $T = \{t_1, \ldots, t_\lambda\}$ and secret keys s_i for all $i = 1, \ldots, n$.*
PSAEnc: *For $t_j \in T$ and all $i = 1, \ldots, n$: $c_{i,j} \leftarrow \mathsf{PSAEnc}_{s_i}(t_j, x_{i,j})$ for $x_{i,j} \in \mathcal{D}$.*
PSADec: *Compute $\sum_{i=1}^{n} x'_{i,j} = \mathsf{PSADec}_{s_0}(t_j, c_{1,j}, \ldots, c_{n,j})$ for $t_j \in T$ and ciphers $c_{1,j}, \ldots, c_{n,j}$. For all $t_j \in T$ and $x_{1,j}, \ldots, x_{n,j} \in \mathcal{D}$ the following holds:*

$$\mathsf{PSADec}_{s_0}(t_j, \mathsf{PSAEnc}_{s_1}(t_j, x_{1,j}), \ldots, \mathsf{PSAEnc}_{s_n}(t_j, x_{n,j})) = \sum_{i=1}^{n} x_{i,j}.$$

The Setup-phase has to be carried out just once and for all, and can be performed with a secure multi-party protocol among all users and the analyst. In all other phases, no communication between the users is needed.

The system parameters pp are public and constant for all time-steps with the implicit understanding that they are used in Σ. Every user encrypts her value $x_{i,j}$ with her own secret key s_i and sends the ciphertext to the analyst. If the analyst receives the ciphertexts of *all* users in a time-step t_j, it computes the aggregate with the decryption key s_0. For a particular time-step, let the users' values be of the form $x_{i,j} = d_{i,j} + e_{i,j}$, $i = 1, \ldots, n$, where $d_{i,j} \in \mathcal{D}$ is the original data of the user i and $e_{i,j}$ is her error term. It is reasonable to assume that $e_{i,j} = 0$ for the $(1 - \gamma) \cdot n$ compromised users, since this can only increase their chances to infer some information about the uncompromised users. There is no privacy-breach if only one user adds the entirely needed noise (first approach) or if the uncompromised users generate noise of low variance (second approach), since the single values $x_{i,j}$ are encrypted and the analyst cannot learn anything about them, except for their aggregate.

Security. Since our model allows the analyst to compromise users, the aggregator can obtain auxiliary information about the data of the compromised users or their secret keys. Even then a secure PSA scheme should release no more information than the aggregate of the uncompromised users' data.

We can assume that an adversary knows the secret keys of the entire compromised coalition. If the protocol is secure against such an attacker, then it is also secure against an attacker without the knowledge of every key from the coalition. Thus, in our security definition we consider the most powerful adversary.

Definition 4 (Non-adaptive Aggregator Obliviousness [28]**).** *Let κ be a security parameter. Let $\mathcal{T}$ be a ppt adversary for a PSA scheme $\Sigma = (\mathsf{Setup}, \mathsf{PSAEnc}, \mathsf{PSADec})$ and let $\mathcal{D}$ be a set. We define a security game between a challenger and the adversary $\mathcal{T}$.*

Setup. *The challenger runs the* Setup *algorithm on input security parameter κ and returns public parameters* pp*, public encryption parameters T with $|T| = \lambda = poly(\kappa)$ and secret keys $s_0, s_1, \ldots, s_n$. It sends $\kappa, \mathsf{pp}, T, s_0$ to $\mathcal{T}$. $\mathcal{T}$ chooses $U \subseteq [n]$ and sends it to the challenger which returns $(s_i)_{i \in [n] \setminus U}$.*

Queries. *$\mathcal{T}$ is allowed to query $(i, t_j, x_{i,j})$ with $i \in U, t_j \in T, x_{i,j} \in \mathcal{D}$ and the challenger returns $c_{i,j} \leftarrow \mathsf{PSAEnc}_{s_i}(t_j, x_{i,j})$.*

Challenge. *$\mathcal{T}$ chooses $t_{j^*} \in T$ such that no encryption query with t_{j^*} was made. (If there is no such t_{j^*} then the challenger simply aborts.) $\mathcal{T}$ queries two different tuples $(x^{[0]}_{i,j^*})_{i \in U}, (x^{[1]}_{i,j^*})_{i \in U}$ with $\sum_{i \in U} x^{[0]}_{i,j^*} = \sum_{i \in U} x^{[1]}_{i,j^*}$. The challenger flips a random bit $b \leftarrow_R \{0, 1\}$. For all $i \in U$ the challenger returns $c_{i,j^*} \leftarrow \mathsf{PSAEnc}_{s_i}(t_{j^*}, x^{[b]}_{i,j^*})$.*

Queries. *$\mathcal{T}$ is allowed to make the same type of queries as before restricted to encryption queries with $t_j \neq t_{j^*}$.*

Guess. *$\mathcal{T}$ outputs a guess about b.*

The adversary's probability to win the game (i.e. to guess b correctly) is $1/2 + \nu(\kappa)$. A PSA scheme is non-adaptively aggregator oblivious *or achieves* non-adaptive Aggregator Obliviousness *(*AO1*), if there is no ppt adversary $\mathcal{T}$ with advantage $\nu(\kappa) > \mathsf{neg}(\kappa)$ in winning the game.*

Encryption queries are made only for $i \in U$, since knowing the secret key for all $i \in [n] \setminus U$ the adversary can encrypt a value autonomously. If encryption queries in time-step t^*_j were allowed, then no deterministic scheme would be aggregator oblivious. The adversary $\mathcal{T}$ can determine the original data of all $i \in [n] \setminus U$ for every time-step, since it knows $(s_i)_{i \in [n] \setminus U}$. Then $\mathcal{T}$ can compute the aggregate of the uncompromised users' data.

The security definition indicates that $\mathcal{T}$ cannot distinguish between the encryptions of two different data collections $(x^{[0]}_i)_{i \in U}, (x^{[1]}_i)_{i \in U}$ with the same aggregate at time-step t^*. For proving that a secure PSA scheme in the sense of Definition 4 can be used for computing differentially private statistics with small error, we have to slightly modify the security game such that an adversary may choose adjacent (and non-perturbed) databases, as it is required in the definition of differential privacy. For details, see Sect. 4.2.

Weak PRF. In our security analysis, we make use of the following definition.

Definition 5 (Weak PRF [23]**).** *Let κ be a security parameter. Let A, B, C be sets with sizes parameterised by a complexity parameter κ. A family of functions $\mathcal{F} = \{\mathsf{F}_a \,|\, \mathsf{F}_a : B \to C\}_{a \in A}$ is called a* weak PRF family*, if for all ppt algorithms $\mathcal{D}^{\mathcal{O}(\cdot)}_{PRF}$ with oracle access to $\mathcal{O}(\cdot)$ (where $\mathcal{O}(\cdot) \in \{\mathsf{F}_a(\cdot), \mathsf{rand}(\cdot)\}$) on any polynomial number of given uniformly chosen inputs, we have: $|\Pr[\mathcal{D}^{\mathsf{F}_a(\cdot)}_{PRF}(\kappa) = 1] - \Pr[\mathcal{D}^{\mathsf{rand}(\cdot)}_{PRF}(\kappa) = 1]| \leq \mathsf{neg}(\kappa)$, where $a \leftarrow \mathcal{U}(A)$ and $\mathsf{rand} \in \{f \,|\, f : B \to C\}$ is a random mapping from B to C.*

3 Main Result

In this work we prove the following result by showing the connection between a key-homomorphic weak pseudo-random function and a differentially private mechanism for sum queries.

Theorem 1. *Let $\epsilon > 0$, $w < w' \in \mathbb{Z}$, $m, n \in \mathbb{N}$ with $\max\{|w|, |w'|\} < m$. Let $\mathcal{D} = \{w, \ldots, w'\}$ and $f_{\mathcal{D}}$ be a sum query. If there exist groups $G' \subseteq G$, a key-homomorphic weak pseudo-random function family mapping into G' and an efficiently computable and efficiently invertible homomorphism $\varphi : \{-mn, \ldots, mn\} \to G$ injective over $\{-mn, \ldots, mn\}$, then there exists an efficient mechanism for $f_{\mathcal{D}}$ that preserves (ϵ, δ)-*CDP *for any $0 < \delta < 1$ with an error bound of $O(S(f_{\mathcal{D}})/\epsilon \cdot \log(1/\delta))$ and requires each user to send exactly one message.*

The proof of Theorem 1 is provided in the next two sections. In Sect. 4 we recall from [28] how to construct a general PSA scheme from a key-homomorphic weak PRF. Subsequently, we show that a secure PSA scheme in composition with a DP-mechanism preserves CDP. In Sect. 5, based on the DLWE problem with errors drawn from a Skellam distribution, we provide an instantiation of a key-homomorphic weak PRF. This yields a concrete efficient PSA scheme that automatically embeds a DP-mechanism with accuracy as stated in Theorem 1.

4 From Key-Homomorphic Weak PRF to CDP

We give a condition for the existence of secure PSA schemes and then analyse its connection to CDP.

4.1 From Key-Homomorphic Weak PRF to Secure PSA

Now we state the condition for the existence of secure PSA schemes for sum queries in the sense of Definition 4.

Theorem 2 (Weak PRF gives secure PSA scheme [28]). *Let κ be a security parameter, and $m, n \in \mathbb{N}$ with $\log(m) = \mathsf{poly}(\kappa), n = \mathsf{poly}(\kappa)$. Let $(G, \cdot), (S, *)$ be finite abelian groups and $G' \subseteq G$. For some finite set M, let $\mathcal{F} = \{\mathsf{F}_s \,|\, \mathsf{F}_s : M \to G'\}_{s \in S}$ be a (possibly randomised) weak PRF family and let $\varphi : \{-mn, \ldots, mn\} \to G$ be a mapping. Then the following PSA scheme $\Sigma = (\mathsf{Setup}, \mathsf{PSAEnc}, \mathsf{PSADec})$ achieves* AO1*:*

Setup: $(\mathsf{pp}, T, s_0, s_1, \ldots, s_n) \leftarrow \mathsf{Setup}(1^\kappa)$, *where* pp *are parameters of $G, G', S, M, \mathcal{F}, \varphi$. The keys are $s_i \leftarrow \mathcal{U}(S)$ for all $i \in [n]$ with $s_0 = (*_{i=1}^n s_i)^{-1}$ and $T \subset M$ such that all $t_j \in T$ are chosen uniformly at random from M, $j = 1, \ldots, \lambda = \mathsf{poly}(\kappa)$.*

PSAEnc: *Compute $c_{i,j} = \mathsf{F}_{s_i}(t_j) \cdot \varphi(x_{i,j})$ in G for $x_{i,j} \in \widehat{\mathcal{D}} = \{-m, \ldots, m\}$ and public parameter $t_j \in T$.*

PSADec: *Compute $V_j = \varphi^{-1}(S_j)$ (if possible) with $S_j = \mathsf{F}_{s_0}(t_j) \cdot c_{1,j} \cdot \ldots \cdot c_{n,j}$.*

If $\mathcal{F}$ contains only deterministic functions that are homomorphic over S, if φ is homomorphic and injective over $\{-mn, \ldots, mn\}$ and if the $c_{i,j}$ are encryptions of the $x_{i,j}$, then $V_j = \sum_{i=1}^{n} x_{i,j}$, i.e. then PSADec *correctly decrypts $\sum_{i=1}^{n} x_{i,j}$.*

The reason for not including the correctness property in the main statement is that in Sect. 5 we will provide an example of a secure PSA scheme based on the DLWE problem that does not have a fully correct decryption algorithm, but a noisy one. This noise is used for establishing the security of the protocol and for preserving the differential privacy of the decryption output.

Hence, we need a key-homomorphic weak PRF and a mapping which homomorphically aggregates all users' data. Since every data value is at most m, the scheme correctly retrieves the aggregate, which is at most $m \cdot n$. Importantly, the product of all pseudo-random values $\mathsf{F}_{s_0}(t), \mathsf{F}_{s_1}(t), \ldots, \mathsf{F}_{s_n}(t)$ is the neutral element in the group G for all $t \in T$. Since the values in T are uniformly distributed in M, it is enough to require that $\mathcal{F}$ is a *weak* PRF family. Thus, the statement of Theorem 2 does not require a random oracle.

4.2 From Secure PSA to CDP

In this section, we describe how to preserve CDP using a PSA scheme. Specifically, let $\mathcal{A}$ be a mechanism which, given some event *Good*, evaluates a statistical query over a database $D \in \mathcal{D}^n$ preserving ϵ-DP. Furthermore, let Σ be a secure PSA scheme. We show that $\mathcal{A}$ executed through Σ preserves ϵ-CDP given *Good*. Assume $\Pr[\neg \mathit{Good}] \leq \delta$. Then it is immediate that $\mathcal{A}$ preserves (ϵ, δ)-CDP *unconditionally* if executed through Σ. Due to space limitations, we provide the proof of these results in the full version.

Theorem 3 (**DP and AO1 give CDP**)**.** *Let $\mathcal{A}$ be a randomised mechanism that gets as inpout some database $D = (d_1, \ldots, d_n) \in \mathcal{D}^n$, generates some database $D' = D'(D) = (x_1(d_1), \ldots, x_n(d_n)) = (x_1, \ldots, x_n)$ and outputs $S = \sum_{i=1}^{n} x_i$, such that ϵ-DP for D is preserved. Let Σ be a PSA scheme that gets as input values $x_1, \ldots, x_n$, outputs ciphers $c_1 = c_1(x_1), \ldots, c_n = c_n(x_n)$ and $S = \sum_{i=1}^{n} x_i$ and achieves* AO1*. Then the composition of Σ with $\mathcal{A}$ achieves* AO1 *and preserves ϵ-*CDP*.*

5 A Weak PRF for CDP Based on DLWE

We are ready to show how Theorem 2 contributes to build a prospective post-quantum secure PSA scheme for differentially private data analyses with a relatively high accuracy. Concretely, we can build a secure PSA scheme from the DLWE assumption with errors sampled according to the symmetric Skellam distribution. These errors automatically provide enough noise to preserve DP.

5.1 The Skellam Mechanism for Differential Privacy

In this section we recall the geometric mechanism from [14] and the binomial mechanism from [12] and introduce the Skellam mechanism. Since these mechanisms make use of a discrete probability distribution, they are well-suited for an execution through a secure PSA scheme, thereby preserving CDP as shown in the last section.

Definition 6 (Symmetric Skellam Distribution [27]**).** *Let $\mu > 0$. A discrete random variable X is drawn according to the symmetric Skellam distribution with parameter μ (short: $X \leftarrow \mathrm{Sk}(\mu)$) if its probability distribution function $\psi_\mu \colon \mathbb{Z} \mapsto \mathbb{R}$ is $\psi_\mu(k) = e^{-\mu} I_k(\mu)$, where I_k is the modified Bessel function of the first kind (see pp.* 374–378 *in* [1]*).*

A random variable $X \leftarrow \mathrm{Sk}(\mu)$ can be generated as the difference of two Poisson variables with mean μ, (see [27]) and is therefore efficiently samplable. We use the fact that the sum of independent Skellam random variables is a Skellam random variable.

Lemma 4 (Reproducibility of Sk(μ) [27]**).** *Let $X \leftarrow \mathrm{Sk}(\mu_1)$ and $Y \leftarrow \mathrm{Sk}(\mu_2)$ be i.i.d. Then $Z := X + Y$ is distributed according to $\mathrm{Sk}(\mu_1 + \mu_2)$.*

An induction step shows that the sum of n i.i.d. symmetric Skellam variables with variance μ is a symmetric Skellam variable with variance $n\mu$. The proofs of the following Theorems 5 and 6 are based on standard concentration inequalities and are provided in the full version.

Theorem 5 (Skellam Mechanism). *Let $\epsilon > 0$. For every database $D \in \mathcal{D}^n$ and query f with sensitivity $S(f)$ the randomised mechanism $\mathcal{A}(D) := f(D) + Y$ preserves (ϵ, δ)-DP, if $Y \leftarrow \mathrm{Sk}(\mu)$ with*

$$\mu = \frac{\log(1/\delta) + \epsilon}{1 - \cosh(\epsilon/S(f)) + (\epsilon/S(f)) \cdot \sinh(\epsilon/S(f))}.$$

Remark 1. *The bound on μ from Theorem 5 is smaller than $2 \cdot (S(f)/\epsilon)^2 \cdot (\log(1/\delta) + \epsilon)$, thus for the standard deviation $\sqrt{\mu}$ of $Y \leftarrow \mathrm{Sk}(\mu)$ it holds that $\sqrt{\mu} = O(S(f) \cdot \sqrt{\log(1/\delta)}/\epsilon)$.*

Executing this mechanism through a PSA scheme requires the use of the known constant γ which denotes the a priori estimate of the lower bound on the fraction of uncompromised users. For this case, we provide the accuracy bound for the Skellam mechanism.

Theorem 6 (Accuracy of the Skellam Mechanism). *Let $\epsilon > 0, 0 < \delta < 1$, $S(f) > 0$ and let $0 < \gamma < 1$ be the a priori estimate of the lower bound on the fraction of uncompromised users in the network. By distributing the execution of a perturbation mechanism as described above and using the parameters from Theorem 5, we obtain (α, β)-accuracy with*

$$\alpha = \frac{S(f)}{\epsilon} \cdot \left(\frac{1}{\gamma} \cdot \left(\log\left(\frac{1}{\delta}\right) + \epsilon \right) + \log\left(\frac{2}{\beta}\right) \right).$$

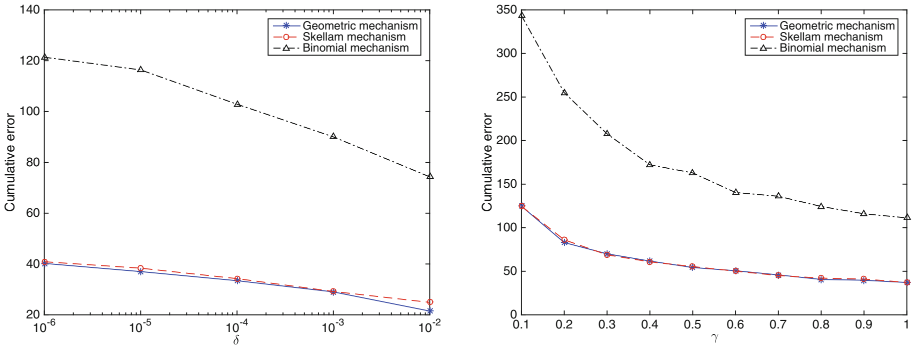


Fig. 1. Empirical error of the geometric, Skellam and binomial mechanisms. The fixed parameters are $\epsilon = 0.1, S(f) = 1, n = 1000$. The plot on the left shows the mean of the error in absolute value for variable δ and $\gamma = 1$ over 1000 repeats, the plot on the right is for variable γ and $\delta = 10^{-5}$.

Theorem 6 shows that for constant δ, β, γ the error of the Skellam mechanism is bounded by $O(S(f)/\epsilon)$. This is the same bound as for the geometric mechanism (see Theorem 3 in [26]) and the binomial mechanism from [12]. Therefore, the Skellam mechanism has the same accuracy as known solutions. In Fig. 1, an empirical comparison between the mechanisms shows that the error of the geometric and the Skellam mechanisms have a very similar behaviour for both variables δ and γ, while the error of the binomial mechanism is roughly three times larger. On the other hand, as pointed out in Sect. 2.1, the execution of the geometric mechanism through a PSA scheme requires each user to generate full noise with a small probability. Complementary, the Skellam mechanism allows all users to simply generate noise of small variance. This fact makes the Skellam mechanism tremendously advantageous over the geometric mechanism, since it permits to construct a PSA scheme based on the DLWE problem, which automatically preserves CDP without any loss in the accuracy compared to state-of-the-art solutions.

5.2 Hardness of the LWE Problem with Errors Following the Symmetric Skellam Distribution

For constructing a secure PSA scheme, we consider the following λ-bounded (Decisional) Learning with Errors problem and prove the subsequent result.

Definition 7 ($\boldsymbol{\lambda}$-bounded LWE). *Let κ be a security parameter, let $\lambda = \lambda(\kappa) = \mathsf{poly}(\kappa)$ and $q = q(\kappa) \geq 2$ be integers and let χ be a distribution on $\mathbb{Z}_q$. Let $\mathbf{x} \leftarrow \mathcal{U}(\mathbb{Z}_q^\kappa)$, let $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{\lambda \times \kappa})$ and let $\mathbf{e} \leftarrow \chi^\lambda$. The goal of the* $\mathrm{LWE}(\kappa, \lambda, q, \chi)$ *problem is, given $(\mathbf{A}, \mathbf{A}\mathbf{x} + \mathbf{e})$, to find $\mathbf{x}$. The goal of the* $\mathrm{DLWE}(\kappa, \lambda, q, \chi)$ *problem is, given $(\mathbf{A}, \mathbf{y})$, to decide whether $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$ or $\mathbf{y} = \mathbf{u}$ with $\mathbf{u} \leftarrow \mathcal{U}(\mathbb{Z}_q^\lambda)$.*

Theorem 7 (LWE with Skellam-distributed errors). *Let κ be a security parameter and let $\lambda = \lambda(\kappa) = \mathsf{poly}(\kappa)$ with $\lambda > 3\kappa$. Let $q = q(\kappa) = \mathsf{poly}(\kappa)$ be a sufficiently large prime modulus and $\rho > 0$ such that $\rho q \geq 4\lambda\sqrt{\kappa}s$. If there exists a ppt algorithm that solves the* $\mathrm{LWE}(\kappa, \lambda, q, \mathrm{Sk}((\rho q)^2/4))$ *problem with more than negligible probability, then there exists an efficient quantum-algorithm that approximates the decisional shortest vector problem (*GapSVP*) and the shortest independent vectors problem (*SIVP*) to within $\tilde{O}(\lambda\kappa/\rho)$ in the worst case.*

Based on the same assumptions, the decisional problem $\mathrm{DLWE}(\kappa, \lambda, q, \mathrm{Sk}((\rho q)^2/4))$ is also hard due to the search-to-decision reduction from [21].

Notions of LWE can be found in the full version. As mentioned in the introduction, our proof of Theorem 7 uses ideas from [10]. Since the Skellam distribution is both reproducible and well-suited for preserving differential privacy (see Theorem 5), the error terms in our DLWE-based PSA scheme are used for two tasks: establishing the cryptographic security of the scheme and the distributed noise generation to preserve differential privacy.

As observed in [10], considering a λ-bounded LWE problem, where the adversary is given $\lambda(\kappa) = \mathsf{poly}(\kappa)$ samples, poses no restrictions to most cryptographic applications of the LWE problem, since they require only an a priori fixed number of samples. In our application to differential privacy, we identify λ with the number of queries in a pre-defined time-series.

Entropy and Lossy Codes. We introduce the conditional min-entropy as starting point for our technical tools. It can be seen as a measure of ambiguity.

Definition 8 (Conditional min-entropy [10]). *Let χ be a probability distribution with finite support $Supp(\chi)$ and let $X, \tilde{X} \leftarrow \chi$. Let f, g be two (possibly randomised) maps on the domain $Supp(\chi)$. The (f, g)-*conditional min-entropy *$H_\infty(X \mid f(X) = g(\tilde{X}))$ of X is defined as*

$$H_\infty(X \mid f(X) = g(\tilde{X})) = -\log_2\left(\max_{\xi \in Supp(\chi)}\{\Pr[X = \xi \mid f(X) = g(\tilde{X})]\}\right).$$

In the remainder of the work we consider $f = f_{\mathbf{A},\mathbf{e}}$ and $g = g_{\mathbf{A},\mathbf{e}}$ as maps to the set of LWE instances, i.e. $f_{\mathbf{A},\mathbf{e}}(\mathbf{y}) = g_{\mathbf{A},\mathbf{e}}(\mathbf{y}) = \mathbf{A}\mathbf{y} + \mathbf{e}$. Now we provide the notion of lossy codes, which is the main technical tool used in the proof of the hardness result.

Definition 9 (Families of Lossy Codes [10]). *Let κ be a security parameter, let $\lambda = \lambda(\kappa) = \mathsf{poly}(\kappa)$ and let $q = q(\kappa) \geq 2$ be a modulus, $\Delta = \Delta(\kappa)$ and let χ be a distribution on $\mathbb{Z}_q$. Let $\{\mathcal{C}_{\kappa,\lambda,q}\}$ be a family of distributions, where $\mathcal{C}_{\kappa,\lambda,q}$ is defined on $\mathbb{Z}_q^{\lambda\times\kappa}$. The distribution family $\{\mathcal{C}_{\kappa,\lambda,q}\}$ is Δ-*lossy *for the error distribution χ, if the following hold:*

1. *$\mathcal{C}_{\kappa,\lambda,q}$ is pseudo-random: It holds that $\mathcal{C}_{\kappa,\lambda,q} \approx_c \mathcal{U}(\mathbb{Z}_q^{\lambda\times\kappa})$.*
2. *$\mathcal{C}_{\kappa,\lambda,q}$ is lossy: Let $f_{\mathbf{B},\mathbf{b}}(\mathbf{y}) = \mathbf{B}\mathbf{y} + \mathbf{b}$. Let $\mathbf{A} \leftarrow \mathcal{C}_{\kappa,\lambda,q}$, $\tilde{\mathbf{x}} \leftarrow \mathcal{U}(\mathbb{Z}_q^\kappa), \tilde{\mathbf{e}} \leftarrow \chi^\lambda$, let $\mathbf{x} \leftarrow \mathcal{U}(\mathbb{Z}_q^\kappa)$ and $\mathbf{e} \leftarrow \chi^\lambda$. Then it holds that*

$$\Pr_{(\mathbf{A},\tilde{\mathbf{x}},\tilde{\mathbf{e}})}[H_\infty(\mathbf{x} \mid f_{\mathbf{A},\mathbf{e}}(\mathbf{x}) = f_{\mathbf{A},\tilde{\mathbf{e}}}(\tilde{\mathbf{x}})) \geq \Delta] \geq 1 - \mathsf{neg}(\kappa).$$

3. $\mathcal{U}(\mathbb{Z}_q^{\lambda\times\kappa})$ *is non-lossy: Let* $f_{\mathbf{B},\mathbf{b}}(\mathbf{y}) = \mathbf{B}\mathbf{y} + \mathbf{b}$*. Let* $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{\lambda\times\kappa})$*,* $\tilde{\mathbf{x}} \leftarrow \mathcal{U}(\mathbb{Z}_q^{\kappa}), \tilde{\mathbf{e}} \leftarrow \chi^{\lambda}$*, let* $\mathbf{x} \leftarrow \mathcal{U}(\mathbb{Z}_q^{\kappa})$ *and* $\mathbf{e} \leftarrow \chi^{\lambda}$*. Then it holds that*

$$\Pr_{(\mathbf{A},\tilde{\mathbf{x}},\tilde{\mathbf{e}})}[H_\infty(\mathbf{x} \mid f_{\mathbf{A},\mathbf{e}}(\mathbf{x}) = f_{\mathbf{A},\tilde{\mathbf{e}}}(\tilde{\mathbf{x}})) = 0] \geq 1 - \mathsf{neg}(\kappa).$$

It is not hard to see that the map-conditional entropy suffices for showing that the existence of a lossy code for the error distribution χ implies the hardness of the LWE problem with error distribution χ.

Theorem 8 (Lossy code gives hard LWE [10]). *Let* κ *be a security parameter, let* $\lambda = \lambda(\kappa) = \mathsf{poly}(\kappa)$ *and let* $q = q(\kappa)$ *be a modulus. Let the distribution* χ *on* $\mathbb{Z}_q$ *be efficiently samplable. Let* $\Delta = \Delta(\kappa) = \omega(\log(\kappa))$*. Then the* LWE$(\kappa, \lambda, q, \chi)$ *problem is hard, given that there exists a family* $\{\mathcal{C}_{\kappa,\lambda,q}\} \subseteq \mathbb{Z}_q^{\lambda\times\kappa}$ *of* Δ*-lossy codes for the error distribution* χ*.*

Thus, for our purposes it suffices to show the existence of a lossy code for the error distribution Sk(μ). First, it is easy to show that $\mathcal{U}(\mathbb{Z}_q^{\lambda\times\kappa})$ is always non-lossy if the corresponding error distribution χ can be bounded, thus the third property of Definition 9 is satisfied.

Lemma 9 (Non-lossiness of $\mathcal{U}(\mathbb{Z}_q^{\lambda\times\kappa})$ [10]). *Let* κ *be a security parameter and* χ *a probability distribution on* $\mathbb{Z}$*. Assume the support of* χ *can be bounded by* $r = r(\kappa) = \mathsf{poly}(\kappa)$*. Moreover, let* $q > (4r + 1)^{1+\tau}$ *for a constant* $\tau > 0$ *and* $\lambda = \lambda(\kappa) > (1 + 2/\tau)\kappa$*. Let* $f_{\mathbf{B},\mathbf{b}}(\mathbf{y}) = \mathbf{B}\mathbf{y} + \mathbf{b}$*. Let* $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{\lambda\times\kappa})$*,* $\tilde{\mathbf{x}} \leftarrow \mathcal{U}(\mathbb{Z}_q^{\kappa}), \tilde{\mathbf{e}} \leftarrow \chi^{\lambda}$*, let* $\mathbf{x} \leftarrow \mathcal{U}(\mathbb{Z}_q^{\kappa})$ *and* $\mathbf{e} \leftarrow \chi^{\lambda}$*. Then*

$$\Pr_{(\mathbf{A},\tilde{\mathbf{x}},\tilde{\mathbf{e}})}[H_{f_{\mathbf{A},\mathbf{e}}, f_{\mathbf{A},\tilde{\mathbf{e}}},\tilde{\mathbf{x}}}(\mathbf{x}) = 0] \geq 1 - \mathsf{neg}(\kappa).$$

For the first and the second properties we construct a lossy code for the Skellam distribution as follows. It is essentially the same construction that was used for the uniform error distribution in [10].

Construction 1 (Lossy code for the symmetric Skellam distribution). *Let* κ *be an even security parameter, let* $\lambda = \lambda(\kappa) = \mathsf{poly}(\kappa)$*,* $\nu > 0$ *and let* $q = q(\kappa)$ *be a prime modulus. The distribution* $\mathcal{C}_{\kappa,\lambda,q,\nu}$ *defined on* $\mathbb{Z}_q^{\lambda\times\kappa}$ *is specified as follows. Choose* $\mathbf{A}' \leftarrow \mathcal{U}(\mathbb{Z}_q^{\lambda\times\kappa/2})$*,* $\mathbf{T} \leftarrow \mathcal{U}(\mathbb{Z}_q^{\kappa/2\times\kappa/2})$ *and* $\mathbf{G} \leftarrow D(\nu)^{\lambda\times\kappa/2}$*. Output* $\mathbf{A} = (\mathbf{A}'||(\mathbf{A}'\mathbf{T} + \mathbf{G}))$*.*

From the matrix version of the LWE problem and the search-to-decision reduction from [21], it is straightforward to see that $\mathcal{C}_{\kappa,\lambda,q,\nu}$ is pseudo-random assuming the hardness of the LWE$(\kappa, \lambda, q, D(\nu))$ problem.

It remains to show that Construction 1 satisfies Property 2 of Definition 9. We first state four supporting claims. The proofs are provided in the full version.

Lemma 10. *Let* κ *be an even integer,* $\mathbf{A} = (\mathbf{A}'||(\mathbf{A}'\mathbf{T} + \mathbf{G}))$ *with* $\mathbf{A}' \in \mathbb{Z}_q^{\lambda\times\kappa/2}$*,* $\mathbf{T} \in \mathbb{Z}_q^{\kappa/2\times\kappa/2}$*,* $\mathbf{G} \in \mathbb{Z}_q^{\lambda\times\kappa/2}$*. For all* $\mathbf{x} \in \mathbb{Z}_q^{\kappa/2}$ *there is a* $\mathbf{x}' \in \mathbb{Z}_q^{\kappa}$ *with* $\mathbf{A}\mathbf{x}' = \mathbf{G}\mathbf{x}$*.*

Lemma 11. $-C + \sqrt{C^2+1} \geq \exp(-C)$ *for all* $C \geq 0$.

Lemma 12. *Let* κ *be a security parameter, let* $s = s(\kappa) = \omega(\log(\kappa))$ *and let* $\nu = \nu(\kappa) = \mathsf{poly}(\kappa)$. *Let* $\lambda = \lambda(\kappa) = \mathsf{poly}(\kappa), 0 < \zeta = \zeta(\kappa) = \mathsf{poly}(\kappa)$ *be integers. Let* $\mathbf{G} \leftarrow D(\nu)^{\lambda \times \zeta}$. *Then for all* $\mathbf{z} \in \{0,1\}^\zeta$ *the following hold:*

1. $\Pr[||\mathbf{Gz}||_\infty > \zeta\sqrt{\nu}] \leq \mathsf{neg}(\kappa)$, *where* $||\cdot||_\infty$ *is the supremum norm.*
2. $\Pr[||\mathbf{Gz}||_2^2 > \lambda\zeta^2\nu] \leq \mathsf{neg}(\kappa)$, *where* $||\cdot||_2$ *is the Euclidean norm.*

Lemma 13. *Let* κ *be an even security parameter and* $\mathbf{A} \in \mathbb{Z}_q^{\lambda \times \kappa}$. *Let* $s = s(\kappa) = \omega(\log(\kappa))$, *let* $\mu = \mu(\kappa)$, *let* $q = \mathsf{poly}(\kappa)$ *be a sufficiently large prime modulus and let* $\lambda = \lambda(\kappa) = \mathsf{poly}(\kappa)$ *be even. Let* $\mathbf{e}, \tilde{\mathbf{e}} \leftarrow \mathrm{Sk}(\mu)^\lambda$ *and let* $\tilde{\boldsymbol{\xi}} = \arg\max_{\boldsymbol{\xi} \in \mathbb{Z}_q^\kappa} \{\Pr_{\mathbf{e}}[\mathbf{e} = \mathbf{A}\boldsymbol{\xi} + \tilde{\mathbf{e}}]\}$. *Let* $\mathbf{u} = \mathbf{A}\tilde{\boldsymbol{\xi}} + \tilde{\mathbf{e}}$. *Then* $||\mathbf{u}||_1 \leq \lambda s\sqrt{\mu}$ *with probability* $1 - \mathsf{neg}(\kappa)$.

We now show the lossiness of Construction 1 for the error distribution $\mathrm{Sk}(\mu)$.

Lemma 14 (Lossiness of Construction 1). *Let* κ *be an even security parameter,* $s = s(\kappa) = \omega(\log(\kappa))$, *let* $\nu = \nu(\kappa)$, *let* $q = \mathsf{poly}(\kappa)$ *be a sufficiently large prime modulus, let* $\lambda = \lambda(\kappa) = \mathsf{poly}(\kappa)$ *and let* $\varDelta = \varDelta(\kappa) = \omega(\log(\kappa))$. *Let* $\mu = \mu(\kappa) \geq 4\lambda^2\nu s^2$. *Let* $f_{\mathbf{B},\mathbf{b}}(\mathbf{y}) = \mathbf{By} + \mathbf{b}$. *Let* $\mathbf{A} \leftarrow \{\mathcal{C}_{\kappa,\lambda,q,\nu}\}$ *for* $\{\mathcal{C}_{\kappa,\lambda,q,\nu}\}$ *as in Construction 1,* $\tilde{\mathbf{x}} \leftarrow \mathcal{U}(\mathbb{Z}_q^\kappa), \tilde{\mathbf{e}} \leftarrow \mathrm{Sk}(\mu)^\lambda$, $\mathbf{x} \leftarrow \mathcal{U}(\mathbb{Z}_q^\kappa)$ *and* $\mathbf{e} \leftarrow \mathrm{Sk}(\mu)^\lambda$. *Then* $\Pr_{(\mathbf{A},\tilde{\mathbf{x}},\tilde{\mathbf{e}})}[H_\infty(\mathbf{x} \mid f_{\mathbf{A},\mathbf{e}}(\mathbf{x}) = f_{\mathbf{A},\tilde{\mathbf{e}}}(\tilde{\mathbf{x}})) \geq \varDelta] \geq 1 - \mathsf{neg}(\kappa)$.

Proof. Let $(\mathbf{Mz})_j$ denote the j^{th} entry of $\mathbf{Mz}$ for a matrix $\mathbf{M}$ and a vector $\mathbf{z}$. Let $\mathbf{A} = (\mathbf{A}'||(\mathbf{A}'\mathbf{T}+\mathbf{G}))$ be distributed according to $\mathcal{C}_{\kappa,\lambda,q,\nu}$ with $\mathbf{A}' \leftarrow \mathcal{U}(\mathbb{Z}_q^{\lambda \times \kappa/2})$, $\mathbf{T} \leftarrow \mathcal{U}(\mathbb{Z}_q^{\kappa/2 \times \kappa/2})$ and $\mathbf{G} \leftarrow D(\nu)^{\lambda \times \kappa/2}$. Let $\tilde{\mathbf{e}} = (\tilde{e}_j)_{j=1,\ldots,\lambda} \leftarrow \mathrm{Sk}(\mu)^\lambda$ and let $\tilde{\boldsymbol{\xi}} = \arg\max_{\boldsymbol{\xi} \in \mathbb{Z}_q^\kappa} \{\Pr_{\mathbf{e}}[\mathbf{e} = \mathbf{A}\boldsymbol{\xi} + \tilde{\mathbf{e}}]\}$. Then we have the following:

$$\begin{aligned}
&\Pr_{(\mathbf{A},\tilde{\mathbf{x}},\tilde{\mathbf{e}})}[H_\infty(\mathbf{x} \mid f_{\mathbf{A},\mathbf{e}}(\mathbf{x}) = f_{\mathbf{A},\tilde{\mathbf{e}}}(\tilde{\mathbf{x}})) \geq \varDelta] \\
&= \Pr_{(\mathbf{A},\tilde{\mathbf{x}},\tilde{\mathbf{e}})}\left[\max_{\boldsymbol{\xi} \in \mathbb{Z}_q^\kappa}\left\{\Pr_{(\mathbf{x},\mathbf{e})}[\mathbf{x} = \boldsymbol{\xi} \mid \mathbf{Ax} + \mathbf{e} = \mathbf{A}\tilde{\mathbf{x}} + \tilde{\mathbf{e}}]\right\} \leq 2^{-\varDelta}\right] \\
&= \Pr_{(\mathbf{A},\tilde{\mathbf{x}},\tilde{\mathbf{e}})}\left[\max_{\boldsymbol{\xi} \in \mathbb{Z}_q^\kappa}\left\{\Pr_{(\mathbf{x},\mathbf{e})}[\mathbf{Ax} + \mathbf{e} = \mathbf{A}\tilde{\mathbf{x}} + \tilde{\mathbf{e}} \mid \mathbf{x} = \boldsymbol{\xi}]\frac{\Pr_{\mathbf{x}}[\mathbf{x} = \boldsymbol{\xi}]}{\Pr_{(\mathbf{x},\mathbf{e})}[\mathbf{Ax} + \mathbf{e} = \mathbf{A}\tilde{\mathbf{x}} + \tilde{\mathbf{e}}]}\right\}\right. \\
&\quad \left.\leq 2^{-\varDelta}\right] \qquad (1) \\
&= \Pr_{(\mathbf{A},\tilde{\mathbf{x}},\tilde{\mathbf{e}})}\left[\max_{\boldsymbol{\xi} \in \mathbb{Z}_q^\kappa}\left\{\Pr_{\mathbf{e}}[\mathbf{e} = \mathbf{A}(\tilde{\mathbf{x}} - \boldsymbol{\xi}) + \tilde{\mathbf{e}}]\right.\right. \\
&\quad \left.\left.\cdot\frac{\Pr_{\mathbf{x}}[\mathbf{x} = \boldsymbol{\xi}]}{\sum_{\mathbf{z} \in \mathbb{Z}_q^\kappa}\Pr_{\mathbf{e}}[\mathbf{e} = \mathbf{A}(\tilde{\mathbf{x}} - \mathbf{z}) + \tilde{\mathbf{e}}] \cdot \Pr_{\mathbf{x}}[\mathbf{x} = \mathbf{z}]}\right\} \leq 2^{-\varDelta}\right] \\
&= \Pr_{(\mathbf{A},\tilde{\mathbf{x}},\tilde{\mathbf{e}})}\left[\max_{\boldsymbol{\xi} \in \mathbb{Z}_q^\kappa}\left\{\Pr_{\mathbf{e}}[\mathbf{e} = \mathbf{A}(\tilde{\mathbf{x}} - \boldsymbol{\xi}) + \tilde{\mathbf{e}}]\right.\right. \\
&\quad \left.\left.\cdot\frac{1}{\sum_{\mathbf{z} \in \mathbb{Z}_q^\kappa}\Pr_{\mathbf{e}}[\mathbf{e} = \mathbf{A}(\tilde{\mathbf{x}} - \mathbf{z}) + \tilde{\mathbf{e}}]}\right\} \leq 2^{-\varDelta}\right] \qquad (2)
\end{aligned}$$

$$= \Pr_{(\mathbf{A},\tilde{\mathbf{x}},\tilde{\mathbf{e}})}\left[\max_{\boldsymbol{\xi}\in\mathbb{Z}_q^\kappa}\left\{\Pr_{\mathbf{e}}[\mathbf{e}=\mathbf{A}\boldsymbol{\xi}+\tilde{\mathbf{e}}]\cdot\frac{1}{\sum_{\mathbf{z}\in\mathbb{Z}_q^\kappa}\Pr_{\mathbf{e}}[\mathbf{e}=\mathbf{A}(\tilde{\mathbf{x}}-\mathbf{z})+\tilde{\mathbf{e}}]}\right\}\le 2^{-\Delta}\right] \tag{3}$$

$$= \Pr_{(\mathbf{A},\tilde{\mathbf{e}})}\left[\max_{\boldsymbol{\xi}\in\mathbb{Z}_q^\kappa}\left\{\frac{\Pr_{\mathbf{e}}[\mathbf{e}=\mathbf{A}\boldsymbol{\xi}+\tilde{\mathbf{e}}]}{\sum_{\mathbf{z}\in\mathbb{Z}_q^\kappa}\Pr_{\mathbf{e}}[\mathbf{e}=\mathbf{A}\mathbf{z}+\tilde{\mathbf{e}}]}\right\}\le 2^{-\Delta}\right] \tag{4}$$

$$= \Pr_{(\mathbf{A},\tilde{\mathbf{e}})}\left[\frac{\sum_{\mathbf{z}\in\mathbb{Z}_q^\kappa}\Pr_{\mathbf{e}}[\mathbf{e}=\mathbf{A}\mathbf{z}+\tilde{\mathbf{e}}]}{\Pr_{\mathbf{e}}[\mathbf{e}=\mathbf{A}\tilde{\boldsymbol{\xi}}+\tilde{\mathbf{e}}]} > 2^{\Delta}\right] \tag{5}$$

$$= \Pr_{(\mathbf{A},\tilde{\mathbf{e}})}\left[\sum_{\mathbf{z}\in\mathbb{Z}_q^\kappa}\frac{\Pr_{\mathbf{e}}[\mathbf{e}=\mathbf{A}(\mathbf{z}+\tilde{\boldsymbol{\xi}})+\tilde{\mathbf{e}}]}{\Pr_{\mathbf{e}}[\mathbf{e}=\mathbf{A}\tilde{\boldsymbol{\xi}}+\tilde{\mathbf{e}}]} > 2^{\Delta}\right] \tag{6}$$

$$= \Pr_{(\mathbf{A},\tilde{\mathbf{e}})}\left[\sum_{\mathbf{z}\in\mathbb{Z}_q^\kappa}\prod_{j=1}^{\lambda}\frac{I_{(\mathbf{A}(\mathbf{z}+\tilde{\boldsymbol{\xi}}))_j+\tilde{e}_j}(\mu)}{I_{(\mathbf{A}\tilde{\boldsymbol{\xi}})_j+\tilde{e}_j}(\mu)} > 2^{\Delta}\right]$$

$$\ge \Pr_{(\mathbf{A},\tilde{\mathbf{e}})}\left[\sum_{\mathbf{z}\in\mathbb{Z}_q^\kappa}\prod_{j=1}^{\lambda}\prod_{k=1+(\mathbf{A}\tilde{\boldsymbol{\xi}})_j+\tilde{e}_j}^{(\mathbf{A}(\mathbf{z}+\tilde{\boldsymbol{\xi}}))_j+\tilde{e}_j}\frac{-k+\sqrt{k^2+\mu^2}}{\mu} > 2^{\Delta}\right] \tag{7}$$

$$\ge \Pr_{(\mathbf{A},\tilde{\mathbf{e}})}\left[\sum_{\mathbf{z}\in\mathbb{Z}_q^\kappa}\prod_{j=1}^{\lambda}\left(\frac{-((\mathbf{A}(\mathbf{z}+\tilde{\boldsymbol{\xi}}))_j+\tilde{e}_j)}{\mu}+\sqrt{\left(\frac{(\mathbf{A}(\mathbf{z}+\tilde{\boldsymbol{\xi}}))_j+\tilde{e}_j}{\mu}\right)^2+1}\right)^{(\mathbf{A}\mathbf{z})_j} > 2^{\Delta}\right] \tag{8}$$

$$\ge \Pr_{(\mathbf{A},\tilde{\mathbf{e}})}\left[\sum_{\mathbf{z}\in\mathbb{Z}_q^\kappa}\prod_{j=1}^{\lambda}\exp\left(-\frac{(\mathbf{A}(\mathbf{z}+\tilde{\boldsymbol{\xi}}))_j+\tilde{e}_j}{\mu}\right)^{(\mathbf{A}\mathbf{z})_j} > 2^{\Delta}\right] \tag{9}$$

$$\ge \Pr_{(\mathbf{A},\tilde{\mathbf{e}})}\left[\sum_{\mathbf{z}\in\mathbb{Z}_q^\kappa}\exp\left(-\frac{||\mathbf{A}\mathbf{z}||_2^2+||\mathbf{A}\mathbf{z}||_\infty\cdot||\mathbf{A}\tilde{\boldsymbol{\xi}}+\tilde{\mathbf{e}}||_1}{\mu}\right) > 2^{\Delta}\right] \tag{10}$$

$$\ge \Pr_{\mathbf{A}}\left[\sum_{\mathbf{z}\in\mathbb{Z}_q^\kappa}\exp\left(-\frac{||\mathbf{A}\mathbf{z}||_2^2+||\mathbf{A}\mathbf{z}||_\infty\cdot\lambda s\sqrt{\mu}}{\mu}\right) > 2^{\Delta}\right] - \mathsf{neg}(\kappa) \tag{11}$$

$$\ge \Pr_{\mathbf{G}}\left[\sum_{\mathbf{z}\in\mathbb{Z}_q^{\kappa/2}}\exp\left(-\frac{||\mathbf{G}\mathbf{z}||_2^2+||\mathbf{G}\mathbf{z}||_\infty\cdot\lambda s\sqrt{\mu}}{\mu}\right) > 2^{\Delta}\right] - \mathsf{neg}(\kappa). \tag{12}$$

Equation (1) is an application of the Bayes rule and Eq. (2) applies, since $\mathbf{x}$ is sampled according to a uniform distribution. Equation (3) applies, since maximising over $\boldsymbol{\xi}$ is the same as maximising over $\tilde{\mathbf{x}}-\boldsymbol{\xi}$. Equation (4) is valid since in the denominator we are summing over all possible $\mathbf{z}\in\mathbb{Z}_q^\kappa$. Equation (5) holds by definition of $\tilde{\boldsymbol{\xi}}$. Equation (6) is an index shift by $\tilde{\boldsymbol{\xi}}$. Inequation (7) follows from essential properties of the modified Bessel functions (iterative application of Theorem 1.1 in [17]). Note that the modified Bessel function of the first kind is symmetric when considered over integer orders. Therefore, from this point of the chain of (in)equations (i.e. from Inequation (7)), we can assume that $\tilde{e}_j \geq 0$. Moreover, we can assume that $(\mathbf{Az})_j \geq 0$, since otherwise $I_{(\mathbf{Az})_j+\tilde{e}_j}(\mu) > I_{-(\mathbf{Az})_j+\tilde{e}_j}(\mu)$. I.e. if $(\mathbf{Az})_j < 0$, then we implicitly change the sign of the j^{th} row in the original matrix $\mathbf{A}$ while considering the particular $\mathbf{z}$. In this way, we are always considering the worst-case scenario for every $\mathbf{z}$. Note that this step does not change the distribution of $\mathbf{A}$, since $\{\mathcal{C}_{\kappa,\lambda,q,\nu}\}$ is symmetric. Inequation (8) holds, since $f_\mu(k) = (-k + \sqrt{k^2 + \mu^2})/\mu$ is a monotonically decreasing function. Inequation (9) follows from Lemma 11 by setting $C = ((\mathbf{Az})_j + \tilde{e}_j)/\mu$. Inequation (10) holds because of the Hölder's inequality. Inequation (11) follows from Lemma 13. Inequation (12) follows from Lemma 10, since $\mathbf{A} = (\mathbf{A}'||(\mathbf{A}'\mathbf{T} + \mathbf{G}))$.

Now consider the set $\mathcal{Z} = \{0,1\}^{\kappa/2}$. Then $|\mathcal{Z}| = 2^{\kappa/2}$. Since $\mu \geq 4\lambda^2\nu s^2$, from Lemma 12, it follows that with probability $1 - \mathsf{neg}(\kappa)$ over $(\mathbf{G}, \tilde{\mathbf{e}})$ we have

$$\sum_{\mathbf{z}\in\mathcal{Z}} \exp\left(-\frac{||\mathbf{Gz}||_2^2 + ||\mathbf{Gz}||_\infty \cdot \lambda s\sqrt{\mu}||_1}{\mu}\right) \geq 2^{\kappa/2} \cdot \exp\left(-\frac{\kappa}{4} - \frac{\kappa}{16s^2}\right),$$

where the norm is computed in the central residue-class representation of the elements in $\mathbb{Z}_q$. Moreover we have $2^{\kappa/2} \cdot \exp\left(-\frac{\kappa}{4} - \frac{\kappa}{16s^2}\right) > C^\kappa$ for some constant $C > 1$. Therefore

$$\Pr_{(\mathbf{A},\tilde{\mathbf{x}},\tilde{\mathbf{e}})}[H_\infty(\mathbf{x} \,|\, f_{\mathbf{A},\mathbf{e}}(\mathbf{x}) = f_{\mathbf{A},\tilde{\mathbf{e}}}(\tilde{\mathbf{x}})) \geq \varDelta] = 1 - \mathsf{neg}(\kappa).$$

□

Putting the previous results together, we finally show the hardness of the LWE problem with errors drawn from the symmetric Skellam distribution.

Proof (Proof of Theorem 7). By a result from [25], the LWE$(\kappa, \lambda, q, D(\nu))$ problem is hard for $\nu = (\alpha q)^2/(2\pi) > 2\kappa/\pi$, if there exists no efficient quantum algorithm approximating the decisional shortest vector problem (GapSVP) and the shortest independent vectors problem (SIVP) to within $\tilde{O}(\kappa/\alpha)$ in the worst case. Let $q = q(\kappa) = \mathsf{poly}(\kappa)$, $s = s(\kappa) = \omega(\log(\kappa))$ and $\lambda > 3\kappa$. Then for $\varDelta = \omega(\log(\kappa))$, Lemma 9, the pseudo-randomness of Construction 1 and Lemma 14 provide that Construction 1 gives us a family of $\varDelta$-lossy codes for the symmetric Skellam distribution with variance $\mu \geq 4\lambda^2\nu s^2$. As observed in Theorem 8, this is sufficient for the hardness of the LWE$(\kappa, \lambda, q, \mathrm{Sk}(\mu))$ problem. Setting $\rho = 2\alpha\lambda s$ yields $(\rho q)^2 > 16\lambda^2\kappa s^2$ and the claim follows. □

By the search-to-decision reduction from [21] we obtain the hardness of the DLWE problem as a corollary.

5.3 A CDP-Preserving PSA Scheme Based on DLWE

Security of the scheme. We can build an instantiation of Theorem 2 (without correct decryption) based on the DLWE$(\kappa, \lambda, q, \chi)$ problem as follows.

Set $S = M = \mathbb{Z}_q^\kappa, G = \mathbb{Z}_q$, choose $\mathbf{s}_i \leftarrow \mathcal{U}(\mathbb{Z}_q^\kappa)$ for all $i = 1, \ldots, n$ and $\mathbf{s}_0 = -\sum_{i=1}^n \mathbf{s}_i$, set $\mathsf{F}_{\mathbf{s}_i}(\mathbf{t}) = \langle \mathbf{t}, \mathbf{s}_i \rangle + e_i$ (which is a so-called *randomised* weak pseudo-random function as described in [3,4]), where $e_i \leftarrow \chi$ (for the uncompromised users) and let φ be the identity function. Therefore

$$\langle \mathbf{t}, \mathbf{s}_i \rangle + e_i + d_i = c_{i,\mathbf{t}} \leftarrow \mathrm{PSAEnc}_{\mathbf{s}_i}(\mathbf{t}, d_i)$$

for data value $d_i \in \mathbb{Z}_q$, $i = 1, \ldots, n$. The decryption function is defined by

$$\sum_{i=1}^n d_i + \sum_{i=1}^n e_i = \langle \mathbf{t}, \mathbf{s}_0 \rangle + \sum_{i=1}^n \mathsf{F}_{\mathbf{s}_i}(\mathbf{t}) + d_i = \langle \mathbf{t}, \mathbf{s}_0 \rangle + \sum_{i=1}^n c_{i,\mathbf{t}} = \mathrm{PSADec}_{\mathbf{s}_0}(\mathbf{t}, c_{1,\mathbf{t}}, \ldots, c_{n,\mathbf{t}}).$$

Thus, the decryption is not perfectly correct anymore, but yields a noisy aggregate. Let $\gamma \in (0, 1]$ be the a priori known fraction of uncompromised users in the network. Then we can construct the following DLWE-based PSA schemes.

Example 1 [28]**.** *Let* $\chi = D(\nu/(\gamma n))$ *with parameter* $\nu/(\gamma n) = 2\kappa/\pi$*, then the* $\mathrm{DLWE}(\kappa, \lambda, q, \chi)$ *problem is hard and the above scheme is secure.*

Example 2. *Let* $\chi = \mathrm{Sk}(\mu/(\gamma n))$ *with variance* $\mu/(\gamma n) = 4\lambda^2\kappa s^2$*, where* $\lambda = \lambda(\kappa) = \mathsf{poly}(\kappa)$ *with* $\lambda > 3\kappa$*. Then the* $\mathrm{DLWE}(\kappa, \lambda, q, \chi)$ *problem is hard and the above scheme is secure.*

Remark 2. *The original result from* [25] *states that the LWE problem is hard in the set* $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ *when the noise is distributed according to the* continuous *Gaussian distribution (with a certain bound on the variance) modulo* 1*. Although the continuous Gaussian distribution is reproducible as well, it does not seem to fit well for a DLWE-based PSA scheme: For data processing reasons the values would have to be discretised. Therefore the resulting noise would follow a distribution which is not reproducible anymore.*[5]

Differential privacy and accuracy. The total noise $\sum_{i=1}^n e_i$ in Example 2 is distributed according to $\mathrm{Sk}(\mu)$ due to Lemma 4. Thus, in contrast to the total noise in Example 1, the total noise in Example 2 preserves the distribution of the single noise and can be used for preserving differential privacy of the correct sum by splitting the task of perturbation among the users.

We identify $|T| = \lambda$, i.e. the number of queries is equal to the number of equations in the instance LWE problem. Due to sequential composition,[6] in order to preserve (ϵ, δ)-DP for all λ queries together, the executed mechanism must preserve $(\epsilon/\lambda, \delta)$-$\mathsf{DP}$ for each query. Therefore we must use $\mathrm{Sk}(\lambda^2\mu)$-noise in each query in order to preserve (ϵ, δ)-DP for all λ queries. By Theorem 6, the error in each query within T is bounded by $O(\lambda S(f) \cdot \log(1/\delta)/\epsilon)$ which is consistent with the effects of sequential composition.

[5] In [7] it was shown that the sum of n discrete Gaussians each with parameter σ^2 is statistically close to a discrete Gaussian with parameter $\nu = n\sigma^2$ if $\sigma > \sqrt{n}\eta_\varepsilon(\Lambda)$ for some smoothing parameter $\eta_\varepsilon(\Lambda)$ of the underlying lattice Λ. However, as pointed out in [28], this approach is less suitable for our purpose if the number of users is large, since the aggregated decryption outcome would have a an error with a variance of order $\nu = \Omega(n^2)$ (in Example 2 the variance is only of order $O(\lambda^2\kappa n)$).

[6] See for instance Theorem 3 in [20].

Combining Security, Privacy and Accuracy. Let $S(f) = \lambda w$ and for each time-step $t_j \in T$, let the data of each user come from $\{-w/2, \ldots, w/2\}$. For $\mu = 2 \cdot (\lambda w/\epsilon)^2 \cdot (\log(1/\delta) + \epsilon)$, it follows from the previous discussion and Remark 1 that if every user adds $\mathrm{Sk}(\mu/(\gamma n))$-noise to her data in every time-step $t_j \in T$, then this suffices to preserve (ϵ, δ)-DP for all λ sum-queries executed during T.

Furthermore, if for a security parameter κ we have that $\mu/(\gamma n) = 4\lambda^2 \kappa s^2$, then we obtain a secure protocol for sum-queries, where the security is based on prospectively hard lattice problems. As we showed in Sect. 4.2, a combination of these two results provides (ϵ, δ)-CDP for all λ sum-queries.

Now assume that for $\mu = 4\gamma n \lambda^2 \kappa s^2$, every uncorrupted user in the network adds $\mathrm{Sk}(\mu/(\gamma n))$-noise to her data for each of the λ queries in order to securely encrypt it using the scheme from Example 2. Then there exist ϵ, δ such that the decryption output preserves (ϵ, δ)-CDP for all λ queries. In order to calculate ϵ and δ, we set $2 \cdot (\lambda w/\epsilon)^2 \cdot (\log(1/\delta) + \epsilon) = 4\gamma n \lambda^2 \kappa s^2$. Hence, for all λ queries, the secure scheme preserves (ϵ, δ)-CDP with $\epsilon = \epsilon(\kappa) \approx \sqrt{\frac{w^2 \cdot \log(1/\delta)}{2\gamma n \kappa s^2}}$, indicating that $\epsilon = \epsilon(\kappa)$ depends on $1/\kappa$. Note that this is consistent with the original definition of CDP from [22]. Thus, in addition to a privacy/accuracy trade-off there is also a security/accuracy trade-off. More specifically, depending on κ and n we obtain an upper bound on the (α, β)-accuracy for every single query executed during T:

$$\alpha = \frac{w}{\epsilon/\lambda} \cdot \left(\frac{1}{\gamma} \cdot \left(\log\left(\frac{1}{\delta}\right) + \epsilon \right) + \log\left(\frac{2}{\beta}\right) \right) = O(\lambda s \sqrt{\kappa \cdot n} + \lambda w).$$

Finally, we are able to prove our main result, Theorem 1, which follows from the preceding analyses.

Proof (of Theorem 1). The claim follows from Theorem 3 together with Theorem 2 (instantiated with the efficient constructions in Example 2) and from Theorem 5 together with Theorem 6. □

References

1. Abramowitz, M., Stegun, I.A.: Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. Dover Publications, New York (1964)
2. Ács, G., Castelluccia, C.: I have a DREAM! (DiffeRentially privatE smArt Metering). In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 118–132. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24178-9_9
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). https://doi.org/10.1007/978-3-642-03356-8_35
4. Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom functions and lattices. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 719–737. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_42

5. Benhamouda, F., Joye, M., Libert, B.: A new framework for privacy-preserving aggregation of time-series data. ACM Trans. Inf. Syst. Secur. **18**(3), 10 (2016)
6. Blum, A., Ligett, K., Roth, A.: A learning theory approach to non-interactive database privacy. In: Proceedings of STOC 2008, pp. 609–618 (2008)
7. Boneh, D., Freeman, D.M.: Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 1–16. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19379-8_1
8. Chan, T.-H.H., Shi, E., Song, D.: Privacy-preserving stream aggregation with fault tolerance. In: Keromytis, A.D. (ed.) FC 2012. LNCS, vol. 7397, pp. 200–214. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32946-3_15
9. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46035-7_4
10. Döttling, N., Müller-Quade, J.: Lossy codes and a new variant of the learning-with-errors problem. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 18–34. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_2
11. Dwork, C.: Differential privacy: a survey of results. In: Agrawal, M., Du, D., Duan, Z., Li, A. (eds.) TAMC 2008. LNCS, vol. 4978, pp. 1–19. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79228-4_1
12. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: privacy via distributed noise generation. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 486–503. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_29
13. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14
14. Ghosh, A., Roughgarden, T., Sundararajan, M.: Universally utility-maximizing privacy mechanisms. In: Proceedings of STOC 2009, pp. 351–360 (2009)
15. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. J. ACM **33**(4), 792–807 (1986)
16. Joye, M., Libert, B.: A scalable scheme for privacy-preserving aggregation of time-series data. In: Sadeghi, A.-R. (ed.) FC 2013. LNCS, vol. 7859, pp. 111–125. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39884-1_10
17. Laforgia, A., Natalini, P.: Some inequalities for modified bessel functions. J. Inequal. Appl. **2010**(1), 253035 (2010)
18. Lindell, Y., Pinkas, B.: Secure multiparty computation for privacy-preserving data mining. J. Priv. Confid. **1**(1), 5 (2009)
19. McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: Proceedings of FOCS 2007, pp. 94–103 (2007)
20. McSherry, F.D.: Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In: Proceedings of SIGMOD ICMD 2009, pp. 19–30 (2009)
21. Micciancio, D., Mol, P.: Pseudorandom Knapsacks and the sample complexity of LWE search-to-decision reductions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 465–484. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_26
22. Mironov, I., Pandey, O., Reingold, O., Vadhan, S.: Computational differential privacy. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 126–142. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03356-8_8

23. Naor, M., Reingold, O.: Synthesizers and their application to the parallel construction of pseudo-random functions. In: Proceedings of FOCS 1995, pp. 170–181 (1995)
24. Rastogi, V., Nath, S.: Differentially private aggregation of distributed time-series with transformation and encryption. In: Proceedings of SIGMOD 2010, pp. 735–746 (2010)
25. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of STOC 2005, pp. 84–93 (2005)
26. Shi, E., Chan, T.H., Rieffel, E.G., Chow, R., Song, D.: Privacy-preserving aggregation of time-series data. In: Proceedings of NDSS 2011 (2011)
27. Skellam, J.G.: The frequency distribution of the difference between two poisson variates belonging to different populations. J. Roy. Stat. Soc. **109**(3), 296 (1946)
28. Valovich, F.: Aggregation of time-series data under differential privacy. In: Publication at LATINCRYPT 2017 (2017)

Explicit Formula for Gram-Schmidt Vectors in LLL with Deep Insertions and Its Applications

Junpei Yamaguchi[1] and Masaya Yasuda[2](✉)

[1] Graduate School of Mathematics, Kyushu University,
744 Motooka Nishi-ku, Fukuoka 819-0395, Japan
ma216010@math.kyushu-u.ac.jp

[2] Institute of Mathematics for Industry, Kyushu University,
744 Motooka Nishi-ku, Fukuoka 819-0395, Japan
yasuda@imi.kyushu-u.ac.jp

Abstract. Lattice basis reduction algorithms have been used as a strong tool for cryptanalysis. The most famous one is LLL, and its typical improvements are BKZ and LLL with deep insertions (DeepLLL). In LLL and DeepLLL, at every time to replace a lattice basis, we need to recompute the Gram-Schmidt orthogonalization (GSO) for the new basis. Compared with LLL, the form of the new GSO vectors is complicated in DeepLLL, and no formula has been known. In this paper, we give an explicit formula for GSO in DeepLLL, and also propose an efficient method to update GSO in DeepLLL. As another work, we embed DeepLLL into BKZ as a subroutine instead of LLL, which we call "DeepBKZ", in order to find a more reduced basis. By using our DeepBKZ with blocksizes up to $\beta = 50$, we have found a number of new solutions for the Darmstadt SVP challenge in dimensions from 102 to 123.

Keywords: Lattice basis reduction · LLL with deep insertions
Shortest Vector Problem (SVP)

1 Introduction

Fix $n > 0$. Given n linearly independent column vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^n$, the set of integral linear combinations of the vectors $\mathbf{b}_i$ is called a (full-rank) *lattice* of dimension n. The $n \times n$ matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ is called a *basis* of the lattice. Given an input basis, *lattice basis reduction* aims to find a new basis with short and nearly orthogonal basis vectors. Lattice basis reduction algorithms have various applications in both computational algebraic number theory and cryptanalysis (see [6,18] for example). The most famous lattice basis reduction is the LLL algorithm, proposed in 1982 by Lenstra, Lenstra and Lovász [17]. It computes a reduced basis with provable output quality in polynomial-time in the dimension of an input basis. A typical improvement of LLL is the block Korkine-Zolotarev (BKZ) reduction algorithm proposed by Schnorr and Euchner

J. Kaczorowski et al. (Eds.): NuTMiC 2017, LNCS 10737, pp. 142–160, 2018.
https://doi.org/10.1007/978-3-319-76620-1_9

[22] in 1994 (the concept was first introduced by Schnorr [20]). It can be regarded as a blockwise generalization of LLL. While BKZ with high blocksizes is much stronger than LLL, it is hard to analyze the complexity of BKZ (see [14] for a useful upper bound of the complexity of BKZ). Another improvement of LLL was suggested also in [22], whose idea is called a *deep insertion*. While only adjacent basis vectors are swapped in LLL, *non-adjacent vectors* can be swapped in DeepLLL. The output quality of DeepLLL is often better than LLL in practice (see [10] for their experimental results).

As is mentioned in textbooks [3, Sect. 5.1] and [6, Sect. 2.6.2], one obstacle of DeepLLL is that it is very difficult to keep track of the GSO vectors after every deep insertion. Let $\mathfrak{S}_n$ denote the group of permutations among n elements. Given an element $\sigma \in \mathfrak{S}_n$ and a basis $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ of a lattice L, let $\sigma(\mathbf{B}) := [\mathbf{b}_{\sigma(1)}, \ldots, \mathbf{b}_{\sigma(n)}]$ denote the reordered basis of L. For $1 \leq i < k \leq n$, we define $\sigma_{i,k} \in \mathfrak{S}_n$ as $\sigma_{i,k}(\ell) = \ell$ for $\ell < i$ or $\ell > k$, $\sigma_{i,k}(i) = k$, and $\sigma_{i,k}(\ell) = \ell - 1$ for $i + 1 \leq \ell \leq k$. Then the reordered basis is given by

$$\sigma_{i,k}(\mathbf{B}) = [\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}, \mathbf{b}_k, \mathbf{b}_i, \ldots, \mathbf{b}_{k-1}, \mathbf{b}_{k+1}, \ldots, \mathbf{b}_n],$$

which is obtained by inserting $\mathbf{b}_k$ between $\mathbf{b}_{i-1}$ and $\mathbf{b}_i$ (i.e., a deep insertion). For $2 \leq \ell \leq n$, let π_ℓ denote the orthogonal projection from $\mathbb{R}^n$ over the orthogonal supplement of the $\mathbb{R}$-vector space $\langle \mathbf{b}_1, \ldots, \mathbf{b}_{\ell-1} \rangle_{\mathbb{R}}$ (we also set $\pi_1 = \mathrm{id}$). In the following, we give an explicit formula for the new GSO vectors of $\sigma_{i,k}(\mathbf{B})$:

Theorem 1. *Let $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ be a basis, and $\mathbf{B}^* = [\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*]$ denote its GSO with coefficients $\mu_{i,j}$ and $B_j = \|\mathbf{b}_j^*\|^2$. For $1 \leq i < k \leq n$, let $\mathbf{C} = \sigma_{i,k}(\mathbf{B})$ and $\mathbf{C}^* = [\mathbf{c}_1^*, \ldots, \mathbf{c}_n^*]$ denote its GSO. Then $\mathbf{c}_j^* = \mathbf{b}_j^*$ for $1 \leq j \leq i-1$ and $k+1 \leq j \leq n$. We also have $\mathbf{c}_i^* = \pi_i(\mathbf{b}_k)$ and*

$$\mathbf{c}_j^* = \frac{D_j^{(k)}}{D_{j-1}^{(k)}} \mathbf{b}_{j-1}^* - \frac{\mu_{k,j-1} B_{j-1}}{D_{j-1}^{(k)}} \sum_{\ell=j}^{k} \mu_{k,\ell} \mathbf{b}_\ell^* \tag{1}$$

for $i + 1 \leq j \leq k$, where set $D_\ell^{(k)} = \|\pi_\ell(\mathbf{b}_k)\|^2 = \sum_{j=\ell}^{k} \mu_{k,j}^2 B_j$ for $1 \leq \ell \leq k$. With respect to the squared lengths $C_j = \|\mathbf{c}_j^*\|^2$, we have $C_i = D_i^{(k)}$ and

$$C_j = \frac{D_j^{(k)}}{D_{j-1}^{(k)}} B_{j-1} \tag{2}$$

for $i + 1 \leq j \leq k$.

As an application of Theorem 1, we propose a method to efficiently update GSO in DeepLLL. Compared to the Gram-Schmidt algorithm [9, Algorithm 23], our GSO update algorithm has much faster performance, and it makes DeepLLL practical as well as LLL. In order to obtain a more reduced basis, we embed DeepLLL into BKZ as a subroutine instead of LLL, which we call *DeepBKZ*. Our experiments show that DeepBKZ can find a more reduced basis than the original BKZ with reasonable running time. In practice, DeepBKZ with blocksize

$\beta = 40$ achieves the Hermite factor 1.0095^n on average for random lattices of dimensions $100 \leq n \leq 115$ in the sense of Goldstein and Mayer [12]. In fact, DeepBKZ found new solutions (i.e., shorter lattice vectors) for the Darmstadt SVP challenge [7] in dimensions $n = 102$–107, 109–113, 115, 117, 119 and 123. For example, in dimension $n = 123$, we used DeepBKZ with full enumeration for blocksizes up to $\beta = 50$; it took about three weeks to find a new solution with our non-optimal implementation over a general-purpose PC.

Notation. The symbols $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$ denote the ring of integers, the field of rational numbers, and the field of real numbers, respectively. In this paper, we represent all vectors in column format. For $\mathbf{a} = (a_1, \ldots, a_n)^t \in \mathbb{R}^n$, let $\|\mathbf{a}\|$ denote its Euclidean norm. For $\mathbf{a} = (a_1, \ldots, a_n)^t$ and $\mathbf{b} = (b_1, \ldots, b_n)^t \in \mathbb{R}^n$, let $\langle \mathbf{a}, \mathbf{b} \rangle$ denote the inner product $\sum_{i=1}^n a_i b_i$.

2 Preliminaries

In this section, we review lattices, GSO, LLL and DeepLLL algorithms.

2.1 Lattices and GSO

For a positive integer n, linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{Z}^n$ define the (full-rank) lattice (here we consider only integral lattices for simplicity)

$$L = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z} \ (1 \leq \forall i \leq n) \right\}$$

of dimension n with basis $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{Z}^{n \times n}$. Every lattice has infinitely many bases; If $\mathbf{B}_1$ and $\mathbf{B}_2$ are two bases, then there exists a unimodular matrix $\mathbf{V} \in \mathrm{GL}_n(\mathbb{Z})$ such that $\mathbf{B}_1 = \mathbf{B}_2 \mathbf{V}$. For a basis $\mathbf{B}$ of L, the volume of L is defined as $\mathrm{vol}(L) = |\det(\mathbf{B})| > 0$, which is independent of the choice of the bases.

The GSO of $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ is the orthogonal family $\mathbf{B}^* = [\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*]$, recursively defined by $\mathbf{b}_1 := \mathbf{b}_1$ and

$$\mathbf{b}_i^* := \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \mathbf{b}_j^*, \ \mu_{i,j} := \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\|\mathbf{b}_j^*\|^2} \text{ for } 1 \leq j < i \leq n. \tag{3}$$

We remark that $\mathbf{B}$ should be regarded as an *ordered* set for its GSO. Let $\mathbf{U} = (\mu_{i,j})$, where we set $\mu_{i,i} = 1$ for all i and $\mu_{i,j} = 0$ for all $j > i$. Then we have $\mathbf{B} = \mathbf{B}^* \mathbf{U}^t$ and $\mathrm{vol}(L) = \prod_{i=1}^n \|\mathbf{b}_i^*\|$. For $1 \leq \ell \leq n$, the orthogonal projection π_ℓ over the orthogonal supplement of the $\mathbb{R}$-vector space $\langle \mathbf{b}_1, \ldots, \mathbf{b}_{\ell-1} \rangle_{\mathbb{R}}$ is computed as

$$\pi_\ell(\mathbf{x}) = \sum_{i=\ell}^{n} \frac{\langle \mathbf{x}, \mathbf{b}_i^* \rangle}{\|\mathbf{b}_i^*\|^2} \mathbf{b}_i^* \text{ for any } \mathbf{x} \in \mathbb{R}^n.$$

Algorithm 1. LLL [17]

```
Input: A basis $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ of a lattice $L$, and a reduction parameter $\frac{1}{4} < \alpha < 1$
Output: An $\alpha$-LLL-reduced basis of $L$
Set $k \leftarrow 2$
while $k \leq n$ do
    Size-reduce $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ /* At each $k$, we recursively change $\mathbf{b}_k \leftarrow \mathbf{b}_k - \lceil \mu_{k,j} \rfloor \mathbf{b}_j$ for $1 \leq j \leq k-1$ (see [9, Algorithm 24] for details) */
    if Lovász condition (4) is not satisfied then
        Swap $\mathbf{b}_k$ with $\mathbf{b}_{k-1}$, and set $k \leftarrow \max(2, k-1)$
    else
        Set $k \leftarrow k+1$
    end if
end while
```

2.2 LLL and DeepLLL

Here we briefly review the LLL and DeepLLL algorithms. Let us recall the notion of LLL-reduction [17] (see also [18, Chap. 2] for details).

Definition 1 (LLL). *Let* $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ *be a basis, and* $\mathbf{B}^* = [\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*]$ *denote its GSO with coefficients* $\mu_{i,j}$. *Given* $\frac{1}{4} < \alpha < 1$, *the basis* $\mathbf{B}$ *is called* α-LLL-reduced *if the following two conditions are satisfied:*

1. *(Size-reduced)* $|\mu_{i,j}| \leq 1/2$ *for any* $1 \leq j < i \leq n$.
2. *(Lovász condition)* $\alpha \|\mathbf{b}_{k-1}^*\|^2 \leq \|\pi_{k-1}(\mathbf{b}_k)\|^2$ *for any* $2 \leq k \leq n$. *Since* $\pi_{k-1}(\mathbf{b}_k) = \mathbf{b}_k^* + \mu_{k,k-1}\mathbf{b}_{k-1}^*$, *this condition can be rewritten as*

$$\|\mathbf{b}_k^*\|^2 \geq (\alpha - \mu_{k,k-1}^2)\|\mathbf{b}_{k-1}^*\|^2. \tag{4}$$

In Algorithm 1, we present the LLL algorithm [17] (see also [18, Algorithm 6 in Chap. 2] or [9, Algorithm 24] for details). In LLL, only adjacent basis vectors $\mathbf{b}_{k-1}$ and $\mathbf{b}_k$ can be swapped. In DeepLLL [22], non-adjacent basis vectors can be changed; Given a reduction parameter $\frac{1}{4} < \alpha < 1$, a basis vector $\mathbf{b}_k$ is inserted between $\mathbf{b}_{i-1}$ and $\mathbf{b}_i$ for $1 \leq i < k \leq n$ if the *deep exchange condition*

$$\|\pi_i(\mathbf{b}_k)\|^2 < \alpha \|\mathbf{b}_i^*\|^2$$

is satisfied. In this case, the new GSO vector at the i-th position is given by $\pi_i(\mathbf{b}_k)$, which is strictly shorter than the old GSO vector $\mathbf{b}_i^*$. In Algorithm 2, we present the DeepLLL algorithm [22] (see also [3, Fig. 5.1] or [6, Algorithm 2.6.4]). The output basis of DeepLLL is α-DeepLLL-reduced, defined below:

Definition 2 (DeepLLL). *Given* $\frac{1}{4} < \alpha < 1$, *a basis* $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ *is called* α-DeepLLL-reduced *if the following two conditions are satisfied:*

1. *The basis* $\mathbf{B}$ *is size-reduced.*
2. *We have* $\|\pi_i(\mathbf{b}_k)\|^2 \geq \alpha \|\mathbf{b}_i^*\|^2$ *for any* $1 \leq i < k \leq n$.

Algorithm 2. DeepLLL [22]

```
Input: A basis $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ of a lattice $L$, and a reduction parameter $\frac{1}{4} < \alpha < 1$
Output: An $\alpha$-DeepLLL-reduced basis of $L$
 1: Set $k \leftarrow 2$
 2: while $k \leq n$ do
 3:   Size-reduce $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ as well as in Algorithm 1
 4:   Set $C \leftarrow \|\mathbf{b}_k\|^2$ and $i \leftarrow 1$
 5:   while $i < k$ do
 6:     if $C \geq \alpha \|\mathbf{b}_i^*\|^2$ then
 7:       Compute $C \leftarrow C - \mu_{k,i}^2 \|\mathbf{b}_i^*\|^2$ and set $i \leftarrow i + 1$ /* $C = \|\pi_i(\mathbf{b}_k)\|^2$ */
 8:     else
 9:       Set $\mathbf{B} \leftarrow \sigma_{i,k}(\mathbf{B})$ and update the GSO of $\mathbf{B}$ /* A deep insertion */
10:       Set $k \leftarrow \max(i, 2)$ and go back to step 3
11:     end if
12:   end while
13:   Set $k \leftarrow k + 1$
14: end while
```

3 Proof of Theorem 1

In this section, we shall prove Theorem 1. Throughout this section, we fix a basis $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$, and its GSO vectors $\mathbf{B}^* = [\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*]$ with coefficients $\mu_{i,j}$ and $B_j = \|\mathbf{b}_j^*\|^2$. We also fix $1 \leq i < k \leq n$, and set $\mathbf{C} = \sigma_{i,k}(\mathbf{B})$. As in Theorem 1, let $\mathbf{C}^* = [\mathbf{c}_1^*, \ldots, \mathbf{c}_n^*]$ denote its GSO with $C_j = \|\mathbf{c}_j^*\|^2$. In the next subsection, we first recall the well-known explicit GSO formula in the LLL case.

3.1 Explicit GSO Formula in the LLL Case

If $i = k - 1$ (i.e., the LLL case), we have the well-known formula for the GSO of $\mathbf{C} = \sigma_{k-1,k}(\mathbf{B})$; By [9, Lemma 17.4.3], we have $\mathbf{c}_j^* = \mathbf{b}_j^*$ for $j \neq k - 1, k$, and for $j = k - 1, k$ we have

$$\begin{cases} \mathbf{c}_{k-1}^* = \pi_{k-1}(\mathbf{b}_k) = \mathbf{b}_k^* + \mu_{k,k-1}\mathbf{b}_{k-1}^*, \\ \mathbf{c}_k^* = \dfrac{B_k}{C_{k-1}}\mathbf{b}_{k-1}^* - \dfrac{\mu_{k,k-1}B_{k-1}}{C_{k-1}}\mathbf{b}_k^*. \end{cases} \tag{5}$$

Thanks to this formula, we can efficiently update GSO after every swap in LLL (see [3, Fig. 4.1] or [6, Algorithm 2.6.3] for a procedure of LLL). On the other hand, since such formula for DeepLLL is not known, the Gram-Schmidt algorithm [9, Algorithm 23] is adopted to recompute GSO after every deep insertion (see [3, Fig. 5.1] or [6, Algorithm 2.6.4] for a procedure of DeepLLL).

3.2 Proof of Formula (1)

Here we give a proof of formula (1) in Theorem 1. Throughout our proof, we simply write D_ℓ for $D_\ell^{(k)}$. It follows from definition (3) that we have $\mathbf{c}_j^* = \mathbf{b}_j^*$ for

$1 \leq j \leq i-1$ and $k+1 \leq j \leq n$, and $\mathbf{c}_i^* = \pi_i(\mathbf{b}_k)$. Fixing k, we shall prove (1) by induction on index i from $i = k-1$ to 1; For $i = k-1$, formula (1) is consistent with GSO formula (5) in the LLL case. Now we assume that formula (1) holds for the case $i+1$. In other words, we assume that the GSO of $\sigma_{i+1,k}(\mathbf{B})$ is given by formula (1). Let $\mathbf{G} = \sigma_{i+1,k}(\mathbf{B}) = [\mathbf{g}_1, \ldots, \mathbf{g}_n]$ denote the reordered basis, that is, we have

$$\begin{cases} \mathbf{g}_{i+1} = \mathbf{b}_k, \ \mathbf{g}_j = \mathbf{b}_{j-1} \ (i+2 \leq j \leq k), \\ \quad \mathbf{g}_j = \mathbf{b}_j \ (1 \leq j \leq i \text{ and } k+1 \leq j \leq n). \end{cases}$$

Let $\mathbf{G}^* = [\mathbf{g}_1^*, \ldots, \mathbf{g}_n^*]$ be its GSO. By the induction assumption, we have

$$\begin{cases} \mathbf{g}_j^* = \mathbf{b}_j^* \ (\ 1 \leq j \leq i \text{ and } k+1 \leq j \leq n), \ \mathbf{g}_{i+1}^* = \pi_{i+1}(\mathbf{b}_k), \\ \mathbf{g}_j^* = \dfrac{D_j}{D_{j-1}} \mathbf{b}_{j-1}^* - \dfrac{\mu_{k,j-1} B_{j-1}}{D_{j-1}} \displaystyle\sum_{\ell=j}^{k} \mu_{k,\ell} \mathbf{b}_\ell^* \ (\ i+2 \leq j \leq k). \end{cases} \tag{6}$$

Now we set $D_i' = \|\pi_i'(\mathbf{g}_{i+1})\|^2$, where let π_i' denote the orthogonal projection over the orthogonal supplement of $\langle \mathbf{g}_1, \ldots, \mathbf{g}_{i-1} \rangle_{\mathbb{R}}$. Note that $\mathbf{C} = \sigma_{i,k}(\mathbf{B}) = \sigma_{i,i+1}(\mathbf{G})$. By applying formula (5) to $\sigma_{i,i+1}(\mathbf{G})$, we have

$$\begin{cases} \mathbf{c}_i^* = \pi_i'(\mathbf{g}_{i+1}), \ \mathbf{c}_{i+1}^* = \dfrac{G_{i+1}}{D_i'} \mathbf{g}_i^* - \dfrac{\eta G_i}{D_i'} \mathbf{g}_{i+1}^* \ \left(\eta = \dfrac{\langle \mathbf{g}_{i+1}, \mathbf{g}_i^* \rangle}{G_i} \right), \\ \mathbf{c}_j^* = \mathbf{g}_j^* \ (1 \leq j \leq i-1 \text{ and } i+2 \leq j \leq n), \end{cases}$$

where let $G_j = \|\mathbf{g}_j^*\|^2$ for $1 \leq j \leq n$. Since $\mathbf{c}_j^* = \mathbf{g}_j^*$ for $i+2 \leq j \leq k$, formula (1) holds for any $i+2 \leq j \leq k$. Therefore it is sufficient to prove (1) only for the case $j = i+1$. Note $G_i = B_i$ and $G_{i+1} = D_{i+1}$ by (6). We have

$$\begin{cases} \eta G_i = \langle \mathbf{g}_{i+1}, \mathbf{g}_i^* \rangle = \langle \mathbf{b}_k, \mathbf{b}_i^* \rangle = \mu_{k,i} B_i, \\ \ D_i' = \|\pi_i'(\mathbf{g}_{i+1})\|^2 = \|\mathbf{g}_{i+1}^*\|^2 + \eta^2 \|\mathbf{g}_i^*\|^2 = D_{i+1} + \mu_{k,i}^2 B_i = D_i. \end{cases}$$

Since $\mathbf{g}_i^* = \mathbf{b}_i^*$ and $\mathbf{g}_{i+1}^* = \pi_{i+1}(\mathbf{b}_k)$ by (6), we clearly have

$$\mathbf{c}_{i+1}^* = \frac{D_{i+1}}{D_i} \mathbf{b}_i^* - \frac{\mu_{k,i} B_i}{D_i} \sum_{\ell=i+1}^{k} \mu_{k,\ell} \mathbf{b}_\ell^*.$$

This completes the proof of formula (1) by induction. □

Remark 1. Given any vector $\mathbf{v}$, an explicit GSO formula for

$$[\mathbf{b}_1, \ldots, \mathbf{b}_{k-1}, \mathbf{v}, \mathbf{b}_k, \ldots, \mathbf{b}_n]$$

is shown in [25, Proposition 4.2]. Compared to the formula, Theorem 1 is specific to the reordered basis $\mathbf{C} = \sigma_{i,k}(\mathbf{B})$. The above proof is quite different from [25] and it is based on easy induction with GSO formula (5) in the LLL case.

3.3 Proof of Formula (2)

For the squared lengths C_j, the case $j = i$ is clear by definition. For $i+1 \leq j \leq k$, from formula (1), we clearly have

$$\begin{aligned} C_j &= \frac{D_j^2}{D_{j-1}^2} B_{j-1} + \frac{\mu_{k,j-1}^2 B_{j-1}^2}{D_{j-1}^2} \sum_{\ell=j}^{k} \mu_{k,\ell}^2 B_\ell \\ &= \frac{B_{j-1}}{D_{j-1}^2} \left(D_j^2 + \mu_{k,j-1}^2 B_{j-1} D_j \right) = \frac{B_{j-1} D_j}{D_{j-1}} \end{aligned}$$

since $D_j + \mu_{k,j-1}^2 B_{j-1} = D_{j-1}$ by definition. This completes the proof. □

Remark 2. The *potential* of a basis $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ is defined as

$$\mathrm{Pot}(\mathbf{B}) := \prod_{i=1}^{n} \mathrm{vol}(L_i)^2 = \prod_{i=1}^{n} \|\mathbf{b}_i^*\|^{2(n-i+1)},$$

where L_i denotes the lattice spanned by $[\mathbf{b}_1, \ldots, \mathbf{b}_i]$ for $1 \leq i \leq n$. It is well known that the potential plays an important role in showing that LLL is a polynomial-time algorithm (see [3, Sect. 4.3] for details). For the reordered basis $\mathbf{C} = \sigma_{i,k}(\mathbf{B})$, Fontein et al. [8, Lemma 1] proved the relation

$$\mathrm{Pot}(\mathbf{C}) = \mathrm{Pot}(\mathbf{B}) \prod_{j=i}^{k-1} \frac{\|\pi_j(\mathbf{b}_k)\|^2}{\|\mathbf{b}_j^*\|^2}. \tag{7}$$

With this relation, Fontein et al. proposed polynomial-time variants of DeepLLL.

Fixing k, Fontein et al. proved Eq. (7) by induction on i from $k-1$ to 1. In contrast, with formula (2) in Theorem 1, we can *directly* obtain Eq. (7); We have

$$\begin{aligned} \frac{\mathrm{Pot}(\mathbf{C})}{\mathrm{Pot}(\mathbf{B})} &= \frac{D_i^{n-i+1} \times \prod_{j=i+1}^{k} \left(\frac{D_j}{D_{j-1}} B_{j-1} \right)^{n-j+1}}{\prod_{j=i}^{k} B_j^{n-j+1}} \\ &= \frac{D_i^{n-i+1} \times \left(\frac{D_{i+1}}{D_i} \right)^{n-i} \left(\frac{D_{i+2}}{D_{i+1}} \right)^{n-i-1} \cdots \left(\frac{D_k}{D_{k-1}} \right)^{n-k+1}}{B_i \cdots B_{k-1} \cdot B_k^{n-k+1}} \\ &= \frac{D_i \cdots D_{k-1} \cdot D_k^{n-k+1}}{B_i \cdots B_{k-1} \cdot B_k^{n-k+1}} = \frac{D_i \cdots D_{k-1}}{B_i \cdots B_{k-1}} \end{aligned}$$

since $D_k = B_k$ by definition. □

4 Efficient GSO Update in DeepLLL

Let $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ be a basis, and $\mathbf{B}^* = [\mathbf{b}_1, \ldots, \mathbf{b}_n^*]$ denote its GSO with coefficients $\mu_{i,j}$ and $B_j = \|\mathbf{b}_j^*\|^2$. In this section, we consider how to efficiently update the GSO of the reordered basis $\mathbf{C} = \sigma_{i,k}(\mathbf{B})$ in step 9 of Algorithm 2 for fixed $1 \leq i < k \leq n$. Let $\mathbf{C}^* = [\mathbf{c}_1^*, \dots, \mathbf{c}_n^*]$ denote the GSO of $\mathbf{C}$ with coefficients

$$\xi_{\ell,j} := \frac{\langle \mathbf{c}_\ell, \mathbf{c}_j^* \rangle}{\|\mathbf{c}_j^*\|^2} \text{ for } 1 \leq j < \ell \leq n.$$

The GSO vectors $\mathbf{c}_j^*$ and their squared lengths $C_j = \|\mathbf{c}_j^*\|^2$ are computable by Theorem 1. We can also compute the GSO coefficients $\xi_{\ell,j}$ directly as follows:

Proposition 1. *The GSO coefficients $\xi_{\ell,j}$ are as follows:*

(A) For $i+1 \leq j \leq k$, we have

$$\xi_{\ell,j} = \begin{cases} \mu_{\ell-1,j-1} - \dfrac{\mu_{k,j-1}}{D_j^{(k)}} \displaystyle\sum_{m=j}^{\ell-1} \mu_{\ell-1,m}\mu_{k,m}B_m & (j+1 \leq \ell \leq k), \\ \mu_{\ell,j-1} - \dfrac{\mu_{k,j-1}}{D_j^{(k)}} \displaystyle\sum_{m=j}^{k} \mu_{\ell,m}\mu_{k,m}B_m & (k+1 \leq \ell \leq n). \end{cases}$$

(B) For $j = i$, we have

$$\xi_{\ell,i} = \begin{cases} \dfrac{1}{D_i^{(k)}} \displaystyle\sum_{m=i}^{\ell-1} \mu_{\ell-1,m}\mu_{k,m}B_m & (i+1 \leq \ell \leq k), \\ \dfrac{1}{D_i^{(k)}} \displaystyle\sum_{m=i}^{k} \mu_{\ell,m}\mu_{k,m}B_m & (k+1 \leq \ell \leq n). \end{cases}$$

(C) For $1 \leq j \leq i-1$, we have $\xi_{\ell,j} = \mu_{\ell-1,j}$ for $i+1 \leq \ell \leq k$ and $\xi_{i,j} = \mu_{k,j}$.
(D) For the other indices $1 \leq j < \ell \leq n$, we have $\xi_{\ell,j} = \mu_{\ell,j}$.

Proof. It easily follows from Theorem 1. □

4.1 Efficient GSO Update Algorithm

As well as in LLL, it is sufficient to update the GSO coefficients $\xi_{\ell,j}$ and the squared lengths C_j in DeepLLL (namely, the update of the GSO vectors $\mathbf{c}_j^*$ is unnecessary). Moreover, the update of the GSO information is dominant (since we do not use naive size reduction [18, Algorithm 3 in Chap. 2] but partial size reduction [9, Algorithm 24] at each iteration as described in step 3 of Algorithm 1). In Algorithm 4 (in Appendix A), we give an algorithm to efficiently update the GSO information, which can be applied as a sub-function in DeepLLL (Algorithm 2).

Here we discuss the complexity of our GSO update algorithm (Algorithm 4) for the update of $(\mu_{\ell,j})$ and B_j of the reordered basis $\mathbf{B} \leftarrow \sigma_{i,k}(\mathbf{B})$; Set $X = \max_{1 \leq j \leq k} \|\mathbf{b}_j\|$ for an input basis $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_n]$. We note that steps 10 and 13 are dominant in Algorithm 4. In accordance with the proof of

[9, Theorem 17.3.4], the computation of $\mu_{\ell,j} P_j$, $\mu_{\ell-1,j} P_j$ and TS_ℓ in steps 10 and 13 requires $O\left(k^2 \log^2 X\right)$ bit operations when we use exact $\mathbb{Q}$ arithmetic. Since there are at most $O\left(k(n-1)\right) = O\left(kn\right)$ operations to perform steps 10 and 13, the complexity of Algorithm 4 is $O\left(k^3 n \log^2 X\right)$. On the other hand, the complexity of the Gram-Schmidt algorithm for updating $\mu_{\ell,j}$ with $1 \leq j < \ell \leq k$ is $O\left(k^4 n \log^2 X\right)$ [9, Theorem 17.3.4] when we use exact $\mathbb{Q}$ arithmetic. It also requires $O\left(k^3 n^2 \log^2 X\right)$ for updating the other $\mu_{\ell,j}$, and hence the complexity of the Gram-Schmidt algorithm in DeepLLL is $O\left(k^3 n^2 \log^2 X\right)$. Therefore Algorithm 4 is asymptotically n times faster than the Gram-Schmidt algorithm for each GSO update of the reordered basis $\sigma_{i,k}(\mathbf{B})$.

4.2 Implementation Results of DeepLLL

We implemented DeepLLL (Algorithm 2) using the NTL library [23] of C++ programs (we used the `g++` compiler with `-O3 -std=c++11` option). Our experiments ran on an Intel Xeon CPU E5-2670@2.60 GHz with 16 GB memory. We used the `long long` data type for lattice basis vectors, and the `long double` for GSO vectors and coefficients in our programs (we did not use exact $\mathbb{Q}$ arithmetic). We generated bases $\mathbf{B}$ of full-rank lattices of dimensions $n = 100, 200, 300$ and 400 with entries less than 20-bit. In Table 1, we summarize our experimental results on the average performance of DeepLLL with the Gram-Schmidt and our GSO update algorithms. For each dimension n, we generated 100 bases. From Table 1, DeepLLL with our algorithm (Algorithm 4) is about 23.4 (resp., 57.0, 103.7 and 145.7) times faster than with the Gram-Schmidt algorithm in total for $n = 100$ (resp., $n = 200$, 300 and 400). This is due to that the cost of GSO updates is more dominant in DeepLLL for higher dimensions. Hence we estimate that DeepLLL with our algorithm is much faster than with the Gram-Schmidt algorithm for higher dimensions.

Table 1. Performance of DeepLLL with the Gram-Schmidt and our GSO update algorithms (Algorithm 4) for n-dimensional bases with entries less than 20-bit

	(i) With GS alg	(ii) With our alg	Ratio (i)/(ii)
$n = 100$	0.351 s	0.015 s	23.4
$n = 200$	8.494 s	0.149 s	57.0
$n = 300$	60.86 s	0.587 s	103.7
$n = 400$	231.1 s	1.585 s	145.7

5 DeepBKZ: Embedding of DeepLLL into BKZ

Let $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ be a basis of a lattice L, and $\mathbf{B}^* = [\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*]$ denote its GSO. For $1 \leq j \leq k \leq n$, we denote by $\mathbf{B}_{[j,k]}$ the local projected block basis

$$[\pi_j(\mathbf{b}_j), \pi_j(\mathbf{b}_{j+1}), \ldots, \pi_j(\mathbf{b}_k)],$$

Algorithm 3. DeepBKZ (cf., BKZ [22])

```
Input: A basis $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ of a lattice $L$, a BKZ blocksize $\beta \in \{2, \ldots, n\}$, and
    a reduction parameter $\frac{1}{4} < \alpha < 1$ of DeepLLL
Output: An $(\alpha, \beta)$-DeepBKZ-reduced basis $\mathbf{B}$ of $L$
$\mathbf{B} \leftarrow$ DeepLLL $(\mathbf{B}, \alpha)$ /* Compute $\mu_{i,j}$ and $\|\mathbf{b}_j^*\|^2$ */
Set $z \leftarrow 0$ and $j \leftarrow 0$
while $z < n - 1$ do
    Set $j \leftarrow (j \bmod (n-1)) + 1$, $k \leftarrow \min(j + \beta - 1, n)$, $h \leftarrow \min(k+1, n)$
    Compute $\mathbf{v} \leftarrow \mathrm{Enum}(\mu_{[j,k]}, \|\mathbf{b}_j^*\|^2, \ldots, \|\mathbf{b}_k^*\|^2)$ /* Find $\mathbf{v} = (v_j, \ldots, v_k) \in$
    $\mathbb{Z}^{k-j+1}$ such that $\|\pi_j(\sum_{i=j}^k v_i \mathbf{b}_i)\| = \lambda_1(L_{[j,k]})$ by enumeration */
    if $\mathbf{v} \neq (1, 0, \ldots, 0)$ then
        Set $z \leftarrow 0$ and call Modified-DeepLLL$(\mathbf{b}_1, \ldots, \mathbf{b}_{j-1}, \mathbf{w}, \mathbf{b}_j, \ldots, \mathbf{b}_h)$ at
        stage $j$ /* Insert $\mathbf{w} = \sum_{i=j}^k v_i \mathbf{b}_i$ and remove the linear dependency */
    else
        Set $z \leftarrow z + 1$ and call DeepLLL $([\mathbf{b}_1, \ldots, \mathbf{b}_h], \gamma)$ at stage $h - 1$
    end if
end while
```

and by $L_{[j,k]}$ the lattice spanned by $\mathbf{B}_{[j,k]}$ of dimension $k - j + 1$. Let $\lambda_1(L)$ denote the first successive minimum of a lattice L. The basis $\mathbf{B}$ is called *BKZ-reduced* [20] with blocksize $\beta \geq 2$ and factor $\frac{1}{4} < \alpha < 1$ if it is α-LLL-reduced and it satisfies $\|\mathbf{b}_j^*\| = \lambda_1(L_{[j,k]})$ for every $1 \leq j \leq n$ with $k = \min(j + \beta - 1, n)$ (we simply call the basis *β-BKZ-reduced* when the LLL-reduction parameter α is unconscious). Given a basis $\mathbf{B}$ of a lattice L, the BKZ algorithm [22, Sect. 6] computes a β-BKZ-reduced basis of L. For higher β, BKZ outputs a more reduced basis than LLL and DeepLLL in practice.

The original BKZ uses LLL as a subroutine to reduce local bases $\mathbf{B}_{[j,k]}$ (cf., BKZ 2.0 [5], an updated version of BKZ, adopts aborted-BKZ with small blocksizes in preprocessing of local blocks $\mathbf{B}_{[j,k]}$ before enumeration for higher blocksizes $\beta \geq 50$). In this section, we embed DeepLLL into BKZ (instead of LLL), and show experimental results on the running time and the output quality. Here let us define a new reduction notion.

Definition 3 (DeepBKZ). *Let $\frac{1}{4} < \alpha < 1$ and $\beta \geq 2$. A basis* $\mathbf{B}$ *is called* (α, β)-DeepBKZ-reduced *if it is both α-DeepLLL-reduced and β-BKZ-reduced.*

A basis $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ is called *HKZ-reduced* [15] if it is size-reduced and it satisfies $\|\mathbf{b}_i^*\| = \lambda_1(\pi_i(L))$ for any $1 \leq i \leq n$. The notion of BKZ-reduction is a local block version of HKZ-reduction. It is clear that any HKZ-reduced basis is also (α, β)-DeepBKZ-reduced for any $\frac{1}{4} < \alpha < 1$ and $\beta \geq 2$. Namely, DeepBKZ-reduction is a middle notion between BKZ and HKZ. Since any lattice L has an HKZ-reduced basis, there always exists a DeepBKZ-reduced basis in L.

5.1 Algorithm and Implementation

Algorithm 3 is our DeepBKZ (we just adopt DeepLLL to reduce local bases $\mathbf{B}_{[j,k]}$ before enumeration). It takes as input a basis $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ of a lattice L, a BKZ blocksize β, and a reduction parameter α of DeepLLL. It outputs an (α, β)-DeepBKZ-reduced basis of L. In step 7 of Algorithm 3, for $h = \min(k+1, n)$ with fixed k, we need to remove the linear dependency of $(h+1)$-vectors

$$[\mathbf{b}_1, \ldots, \mathbf{b}_{j-1}, \mathbf{w}, \mathbf{b}_j, \ldots, \mathbf{b}_h]$$

for a new lattice vector $\mathbf{w} = \sum_{i=j}^{k} v_i \mathbf{b}_i \in L$ (the vector $\mathbf{v} = (v_j, \ldots, v_k) \in \mathbb{Z}^{k-j+1}$ is found by enumeration in step 5 of Algorithm 3). Our algorithm terminates with the same principle as the BKZ algorithm. As in the modified LLL proposed by Pohst [19] (see also [3, Chap. 6] or [6, Sect. 2.6.4]), we can modify DeepLLL (which we call "Modified-DeepLLL") to remove the linear dependency, and we adopt it in step 7 of Algorithm 3.

In our implementation, we adopted Schnorr-Euchner's full enumeration [22] for step 5 of Algorithm 3 (cf., BKZ 2.0 [5] adopts pruned enumeration of [11] with early-abort strategy). Specifically, we implemented the pseudo-code of [11, Algorithm 2 in Appendix B] with full enumeration setting (more specifically, for [11, Algorithm 2 in Appendix B], we set

$$R_1^2 = \cdots = R_n^2 = 0.99 \cdot \|\mathbf{b}_1\|^2$$

as a bounding vector). Our PC environment and implementation information are same as described in Subsect. 4.2. In particular, we used the `long double` data type for GSO information in enumeration (as the progressive BKZ implementation [1], we may use the `double` data type for efficiency).

5.2 Experimental Results

Let L be a lattice of dimension n. The *Hermite factor* of a lattice basis reduction algorithm for a basis of L is defined as

$$\delta := \frac{\|\mathbf{b}_1\|}{\mathrm{vol}(L)^{1/n}}$$

with the output basis $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ (assume that $\mathbf{b}_1$ is the shortest among the basis vectors $\mathbf{b}_j$). This factor is experimentally investigated in [10], and it is shown that the factor gives a good index to measure the practical output quality of lattice basis reduction algorithms. The output quality becomes better as δ is smaller. According to experimental results [10, Fig. 5], the value $\delta^{1/n}$ converges a constant in practical algorithms such as LLL, DeepLLL and BKZ for large n. The limiting constant $\delta^{1/n}$ is called the *Hermite factor constant.*

In Figs. 1, 2, 3 and 4, we show an experimental comparison of the original BKZ (we implemented), BKZ(fplll), and DeepBKZ for bases of [7] of dimensions 100–115 with seed 0 in terms of the Hermite factor constant $\delta^{1/n}$, where

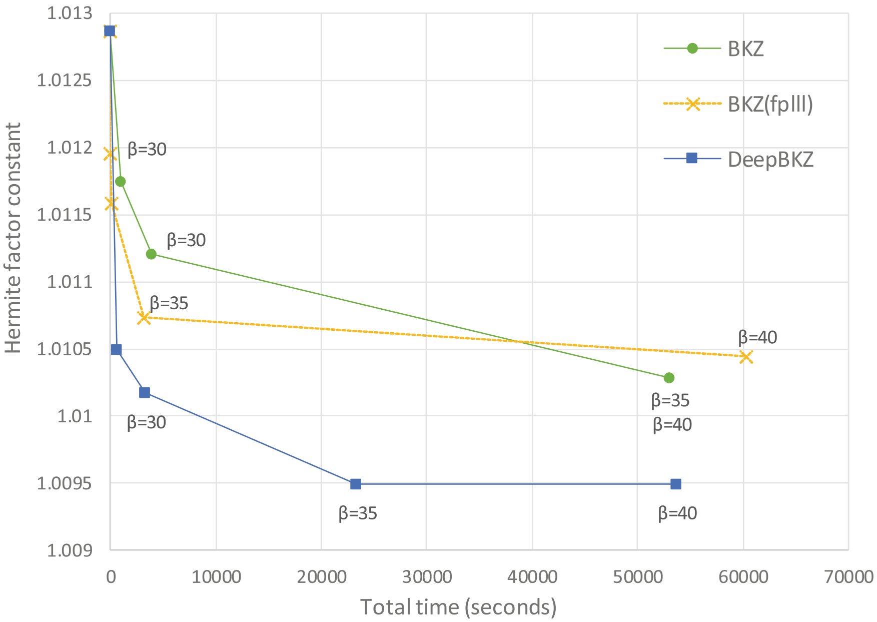


Fig. 1. Transition of the Hermite factor constant $\delta^{1/n}$ of BKZ, BKZ(fplll), DeepBKZ with blocksizes $\beta = 25$–40 for bases of [7] of dimension $n = 100$ with seed 0 (BKZ(fplll) by [24], BKZ and DeepBKZ by our implementation in C++ programs)

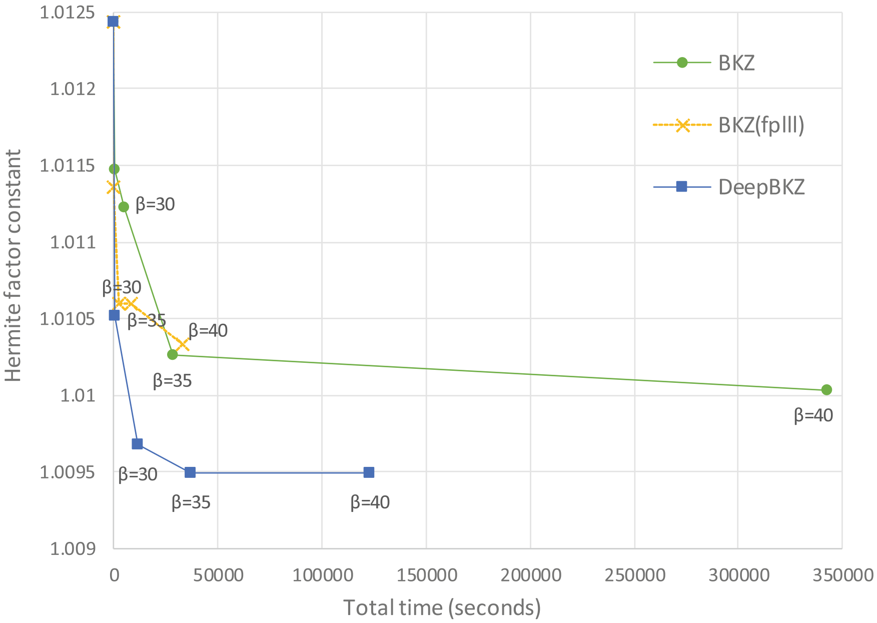


Fig. 2. Same as Fig. 1, but dimension $n = 105$ with seed 0

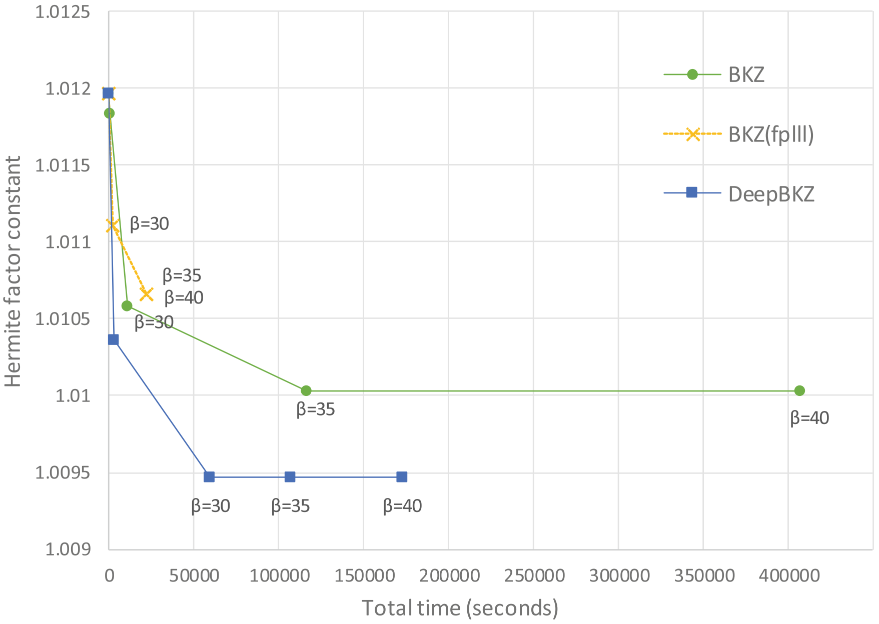


Fig. 3. Same as Fig. 1, but dimension $n = 110$ with seed 0

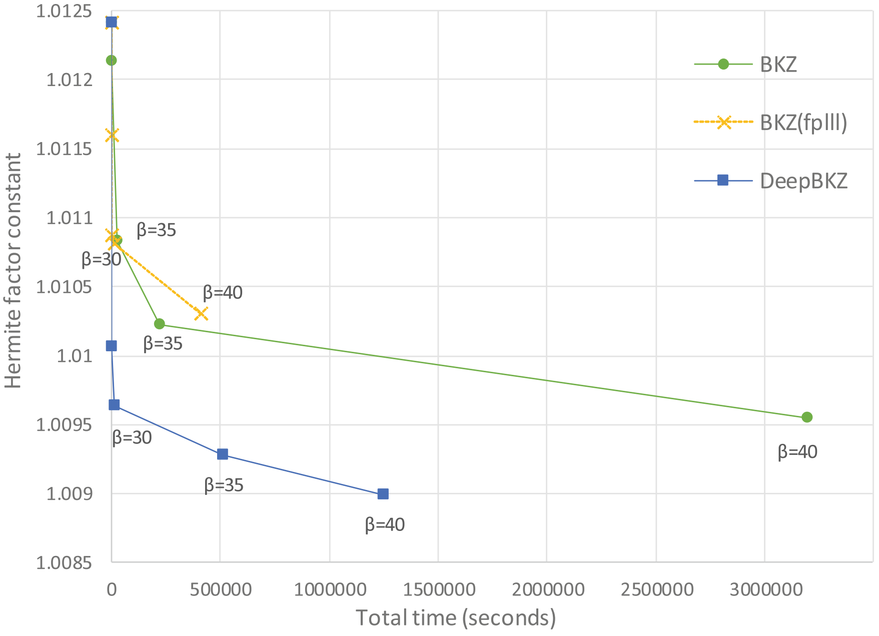


Fig. 4. Same as Fig. 1, but dimension $n = 115$ with seed 0

'BKZ(fplll)' is an implementation of BKZ in the fplll library [24]. In each dimension, we used a BKZ(fplll)-reduced basis with blocksize $\beta = 20$ as an input basis (we used command 'fplll -a bkz -b 20' of [24] without any option). For example, for BKZ, we recursively performed BKZ with blocksize $\beta + 5$ for a β-BKZ-reduced basis with $20 \leq \beta \leq 35$. We took a reduction parameter $\alpha = 0.99$ of LLL and DeepLLL in BKZ and DeepBKZ. In Table 2, we show the average of the Hermite factor constant $\delta^{1/n}$ of BKZ, BKZ(fplll), and DeepBKZ of dimensions $n = 100$–115 with seeds 0–19. In Table 3, we show the average of the total number of tours (here we call one process of steps 5, 6, 7 in Algorithm 3 a *tour*).

Output quality. From Table 2, DeepBKZ outputs a more reduced basis than BKZ and BKZ (fplll) for each blocksize β. According to [10, Table 1], the average of the Hermite factor constant $\delta^{1/n}$ by BKZ with β for random lattices is 1.0128 (resp., 1.0109) for $\beta = 20$ (resp., $\beta = 28$). Furthermore, according to [5, Table 2], it is predicted by simulation (see also (8) below) that $\beta = 85$ (resp., $\beta = 106$) is required to achieve $\delta^{1/n} = 1.01$ (resp., $\delta^{1/n} = 1.009$) in BKZ 2.0 [5], which adopts pruned enumeration and early-abort strategy. In contrast, DeepBKZ with $\beta = 40$ (full enumeration) achieves less than $\delta^{1/n} = 1.0095$ in practice.

Running time. From Figs. 1, 2, 3 and 4, DeepBKZ has reasonable running time compared with BKZ for each β. This is due to that as seen from Table 3, fewer tours are only required in DeepBKZ than BKZ although DeepLLL is still costly in spite of our GSO update algorithm (Algorithm 4). In other words, DeepBKZ can make a basis reduced effectively with fewer insertions of short lattice vectors. Since $[\mathbf{b}_1, \dots, \mathbf{b}_h]$ is α-DeepLLL-reduced before enumeration in DeepBKZ, short lattice vectors $\mathbf{w} = \sum_{i=j}^{k} v_i \mathbf{b}_i \in L$ other than the basis vectors $\mathbf{b}_i$ are inserted in most cases. This is a reason why fewer insertions are required in DeepBKZ. Note that our implementation is non-optimal. In fact, the performance of our C++ program for BKZ is much slower than BKZ(fplll) for most blocksizes β.

Remark 3. In her thesis [4], Chen gives a limiting value of the Hermite factor constant achieved by BKZ-β as a function of β (based on Gaussian Heuristic):

$$\lim_{n \to \infty} \delta^{1/n} \approx \left(\frac{\beta}{2\pi e} (\pi \beta)^{\frac{1}{\beta}} \right)^{\frac{1}{2(\beta-1)}}. \tag{8}$$

On the other hand, Hanrot et al. [14, Theorem 1] showed that the output basis $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_n]$ of their modified version of BKZ, called *terminating BKZ*, satisfies

$$\|\mathbf{b}_1\| \leq 2(\nu_\beta)^{\frac{n-1}{2(\beta-1)} + \frac{3}{2}} \cdot \det(\mathbf{B})^{1/n},$$

where let $\nu_\beta \leq \beta$ denote the maximum of Hermite's constants γ_k for $k \leq \beta$.

5.3 New Solutions by DeepBKZ for the Darmstadt SVP Challenge

With DeepBKZ, we found new solutions (i.e., shorter lattice vectors) for the Darmstadt SVP challenge [7] in dimensions $n = 102$–107, 109–113, 115, 117, 119

Table 2. The average of the Hermite factor constant $\delta^{1/n}$ of BKZ, BKZ(fplll), DeepBKZ for bases of [7] of dimensions $n = 100$–115 with seeds 0–19

Dimension	Algorithm	$\beta = 25$	$\beta = 30$	$\beta = 35$	$\beta = 40$
100	BKZ	1.01170	1.01107	1.01034	1.01011
	BKZ(fplll)	1.01167	1.01110	1.01075	1.01038
	DeepBKZ	1.01043	1.00999	1.00958	1.00949
105	BKZ	1.01170	1.01098	1.01033	1.00996
	BKZ(fplll)	1.01177	1.01099	1.01054	1.01041
	DeepBKZ	1.01041	1.00983	1.00937	1.00924
110	BKZ	1.01166	1.01079	1.01023	1.00996
	BKZ(fplll)	1.01180	1.01105	1.01062	1.01037
	DeepBKZ	1.01010	1.00975	1.00941	1.00916
115	BKZ	1.01181	1.01085	1.01024	1.01002
	BKZ(fplll)	1.01164	1.01089	1.01051	1.01019
	DeepBKZ	1.01021	1.00964	1.00917	1.00899

Table 3. The average of the total number of tours of BKZ and DeepBKZ for bases of [7] of dimensions $n = 100$–115 with seeds 0–19

Dimension	Algorithm	$\beta = 25$	$\beta = 30$	$\beta = 35$	$\beta = 40$
100	BKZ	36632	285249	1812351	2285044
	DeepBKZ	1157	14596	75606	111745
105	BKZ	48759	511875	3005585	7437806
	DeepBKZ	1526	25241	220935	395815
110	BKZ	76977	1147480	7073110	17200141
	DeepBKZ	2113	47295	271418	1045659
115	BKZ	751702	2000791	17494614	24852295
	DeepBKZ	3244	67698	953553	2484898

and 123. For these dimensions, we applied DeepBKZ (Algorithm 3) with full enumeration for blocksizes up to $\beta = 50$ for bases of [7] with seeds 0–10. In fact, $\beta = 40$ or 45 was sufficient to find new solutions in most cases (cf., BKZ 2.0 [5] with blocksize $\beta = 75$, 20%-pruning, found solutions from dimension 90 to 112). In Table 4, we list new short norms among seeds 0–10 in each dimension (in most dimensions from 100 to 120 with other seeds, we found the same short vectors as the previous records). From Table 4, DeepBKZ with $\beta = 50$ achieves less than $\delta^{1/n} = 1.009$ in all dimensions. In dimension $n = 123$ with seed 0, it took about three weeks to find a new solution even with our non-optimal implementation using a PC (our implementation has been improved, and our current program is about 3 times faster than the previous one).

Table 4. New solutions for the Darmstadt SVP challenge [7], found by DeepBKZ with full enumeration for blocksizes up to $\beta = 50$ (in most dimensions, $\beta = 40$ or 45 was sufficient for new solutions)

Dimension	Seed	New norm	Hermite factor constant
123	0	2883	1.00847
119	10	2863	1.00868
117	10	2840	1.00880
115	3	2699	1.00841
113	1	2681	1.00857
112	3	2653	1.00866
111	0	2684	1.00874
110	4	2621	1.00859
109	8	2613	1.00863
107	9	2566	1.00866
106	8	2551	1.00868
105	1	2604	1.00897
104	10	2546	1.00884
103	10	2520	1.00882
102	10	2512	1.00889

6 Conclusion

In this paper, we first gave an explicit formula (Theorem 1) for the GSO in DeepLLL to keep track of the new GSO after every deep insertion. As an application of our GSO formula, we gave an algorithm (Algorithm 4) to efficiently update the GSO information in DeepLLL. Thanks to our GSO update algorithm, DeepLLL can run practically as well as LLL. As another application, we embedded DeepLLL into BKZ in order to obtain a more reduced basis with DeepLLL. Our experiments showed that DeepBKZ (Algorithm 3) can output a more reduced basis than BKZ [22] with reasonable running time in practice. We have found a number of new solutions for the Darmstadt SVP challenge [7] in dimensions from 102 to 123 by DeepBKZ with full enumeration for blocksizes up to $\beta = 50$ (see Table 4).

Acknowledgments. This work was supported by JST CREST Grant Number JPMJCR14D6, Japan. This work was also supported by JSPS KAKENHI Grant Number 16H02830. The authors thank Takuya Hayashi for his useful advices on implementation.

A Efficient GSO Update Algorithm in DeepLLL

In Algorithm 4, we show a detailed algorithm to efficiently update the GSO information in DeepLLL. This algorithm is based on results of Theorem 1 and Proposition 1.

Algorithm 4. Update of the GSO information in DeepLLL

```
Input: GSO information $(\mu_{\ell,j})$ and $B_j = \|\mathbf{b}_j^*\|^2$ of a basis $\mathbf{B}$, and indices $1 \leq i < k \leq n$
Output: Updated GSO information $(\mu_{\ell,j})$ and $B_j$ for the new basis $\mathbf{B} \leftarrow \sigma_{i,k}(\mathbf{B})$
Set $P_k \leftarrow B_k$ and $D_k \leftarrow B_k$
for $j = k - 1$ downto $i$ do
    Compute $P_j \leftarrow \mu_{k,j} B_j$ and $D_j \leftarrow D_{j+1} + \mu_{k,j} P_j$ /* $D_j = \|\pi_j(\mathbf{b}_k)\|^2$ */
end for
Set $S_i = S_{i+1} = \ldots = S_n = 0$
/* By Proposition 1 (A) */
for $j = k$ downto $i + 1$ do
    Compute $T \leftarrow \frac{\mu_{k,j-1}}{D_j}$
    for $\ell = n$ downto $k + 1$ do
        Compute $S_\ell \leftarrow S_\ell + \mu_{\ell,j} P_j$ and $\mu_{\ell,j} \leftarrow \mu_{\ell,j-1} - T S_\ell$
    end for
    for $\ell = k$ downto $j + 2$ do
        Compute $S_\ell \leftarrow S_\ell + \mu_{\ell-1,j} P_j$ and $\mu_{\ell,j} \leftarrow \mu_{\ell-1,j-1} - T S_\ell$
    end for
    if $j \neq k$ then
        Compute $S_{j+1} \leftarrow P_j$ and $\mu_{j+1,j} \leftarrow \mu_{j,j-1} - T S_{j+1}$
    end if
end for
/* By Proposition 1 (B) */
Compute $T \leftarrow \frac{1}{D_i}$
for $\ell = n$ downto $k + 1$ do: Compute $\mu_{\ell,i} \leftarrow T(S_\ell + \mu_{\ell,i} P_i)$;
for $\ell = k$ downto $i + 2$ do: Compute $\mu_{\ell,i} \leftarrow T(S_\ell + \mu_{\ell-1,i} P_i)$;
Compute $\mu_{i+1,i} \leftarrow T P_i$
/* By Proposition 1 (C) */
for $j = 1$ to $i - 1$ do
    Copy $\varepsilon \leftarrow \mu_{k,j}$
    for $\ell = k$ downto $i + 1$ do: Copy $\mu_{\ell,j} \leftarrow \mu_{\ell-1,j}$;
    Copy $\mu_{i,j} \leftarrow \varepsilon$
end for
/* Update of $B_j$ by Theorem 1 */
for $j = k$ downto $i + 1$ do: Compute $B_j \leftarrow \frac{D_j B_{j-1}}{D_{j-1}}$;
Set $B_i \leftarrow D_i$
```

References

1. Aono, Y., Wang, Y., Hayashi, T., Takagi, T.: Improved progressive BKZ algorithms and their precise cost estimation by sharp simulator. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9665, pp. 789–819. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49890-3_30
2. Babai, L.: On Lovász' lattice reduction and the nearest lattice point problem. Combinatorica **6**(1), 1–13 (1986)
3. Bremner, M.R.: Lattice basis reduction: An introduction to the LLL algorithm and its applications. CRC Press, Boca Raton (2011)
4. Chen, Y.: Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe. Ph.D. thesis, Paris 7 (2013)
5. Chen, Y., Nguyen, P.Q.: BKZ 2.0: better lattice security estimates. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 1–20. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_1
6. Cohen, H.: A Course in computational Algebraic Number Theory, Graduate Texts in Mathematics, vol. 138. Springer, Heidelberg (1993). https://doi.org/10.1007/978-3-662-02945-9
7. Darmstadt, T.U.: SVP Challenge. http://www.latticechallenge.org/svp-challenge/
8. Fontein, F., Schneider, M., Wagner, U.: PotLLL: a polynomial time version of LLL with deep insertions. Des. Codes Cryptogr. **73**, 355–368 (2014)
9. Galbraith, S.D.: Mathematics of Public Key Cryptography. Cambridge University Press, Cambridge (2012)
10. Gama, N., Nguyen, P.Q.: Predicting lattice reduction. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 31–51. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_3
11. Gama, N., Nguyen, P.Q., Regev, O.: Lattice enumeration using extreme pruning. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 257–278. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_13
12. Goldstein, D., Mayer, A.: On the equidistribution of Hecke points. Forum Math. **15**, 165–189 (2003)
13. Hanrot, G., Stehlé, D.: Worst-case Hermite-Korkine-Zolotarev reduced lattice bases. RR-6422, INRIA, pp. 1–25 (2008)
14. Hanrot, G., Pujol, X., Stehlé, D.: Analyzing blockwise lattice algorithms using dynamical systems. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 447–464. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_25
15. Korkine, A., Zolotarev, G.: Sur les formes quadratiques. Math. Ann. **6**, 366–389 (1873)
16. Lagarias, J.C., Lenstra, H.W., Schnorr, C.P.: Korkine-Zolotarev bases and successive minima of a lattice and its reciprocal lattice. Combinatorica **10**(4), 333–348 (1990)
17. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann. **261**(4), 515–534 (1982)
18. Nguyen, P.Q., Vallée, B.: The LLL Algorithm, Information Security and Cryptography. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-02295-1
19. Pohst, M.E.: A modification of the LLL reduction algorithm. J. Symb. Comput. **4**, 123–127 (1987)
20. Schnorr, C.P.: A hierarchy of polynomial time lattice basis reduction algorithms. Theoret. Comput. Sci. **53**, 201–224 (1987)

21. Schnorr, C.P.: Lattice reduction by random sampling and birthday methods. In: Alt, H., Habib, M. (eds.) STACS 2003. LNCS, vol. 2607, pp. 145–156. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36494-3_14
22. Schnorr, C.P., Euchner, M.: Lattice basis reduction: improved practical algorithms and solving subset sum problems. Math. Program. **66**, 181–199 (1994)
23. Shoup, V.: NTL: a library for doing number theory. http://www.shoup.net/ntl/
24. The FPLLL development team: FPLLL, a lattice reduction library. https://github.com/fplll/fplll (2016)
25. Yasuda, M., Yokoyama, K., Shimoyama, T., Kogure, J., Koshiba, T.: Analysis of decreasing squared-sum of Gram-Schmidt lengths for short lattice vectors. J. Math. Cryptol. **11**(1), 1–24 (2017)

Number Theory

Factoring n and the Number of Points of Kummer Hypersurfaces mod n

Robert Dryło[1](✉) and Jacek Pomykała[2]

[1] Warsaw School of Economics, al. Niepodległości 162, 02-554 Warszawa, Poland
rdrylo@sgh.waw.pl

[2] Faculty of Mathematics Informatics and Mechanics,
ul. Banacha 2, 02-097 Warsaw, Poland
pomykala@mimuw.edu.pl

Abstract. In this paper we describe the reduction of factorization of a square-free integer n to the problem of determining the number of points in $\mathbb{Z}_n^{d+1}$ on twists of Kummer hypersurfaces $y^k = f(x_1, \ldots, x_d) \bmod n$, where $f(x_1, \ldots, x_d) \in \mathbb{Z}_n[x_1, \ldots, x_d]$ and $k > 1$. This reduction is expected to be polynomial time (in log n) for small k and fixed number of prime divisors of n provided that some necessary for this reduction conditions are satisfied. This extends the known reduction of factorization to determining the number of points on elliptic curves $y^2 = x^3 + ax + b$ over $\mathbb{Z}_n$. In particular our reduction implies that factorization of n can be reduced to determining the number of points on quadrics in $\mathbb{Z}_n^d$, $d > 1$, which extends the known reduction of factorization to determining the order of $\mathbb{Z}_n^*$. We also describe the reduction of factorization to determine the number of points in $\mathbb{P}^2(\mathbb{Z}_n)$ on superelliptic curves $y^k = f(x_1) \bmod n$. To study the complexity of these reductions we introduce some notions and prove useful facts for a more precise analysis. In greater detail we consider the case of the reduction when $n = pq$ is a product of two primes and $k = 2$.

Keywords: Dirichlet characters · Least r-th power nonresidue
Integer factorization · Reductions · Elliptic and hyperelliptic curve
Kummer surface

1 Introduction

Factorization of an integer n can be reduced to the problem of computing the group orders of $G = \mathbb{Z}_n^*$ (see [Bac84]) or $G = E(\mathbb{Z}_n)$ for an elliptic curve E over the ring $\mathbb{Z}_n$ (see e.g. [KuKo98, MMV01, DrPo17]). These reductions are principally based on comparing the orders $P \bmod p$ and $P \bmod q$ for $P \in G$ and distinct primes $p, q \mid n$. If $m = |G|$, while the orders of $P \bmod p$ and $P \bmod q$ are divisible by different maximal powers of 2, then multiplication mP reveals a non-trivial divisor of n, similarly as in Lenstra's elliptic curves factorization method [Len87]. Certainly the probability of choosing a suitable point P depends on the group

J. Kaczorowski et al. (Eds.): NuTMiC 2017, LNCS 10737, pp. 163–177, 2018.
https://doi.org/10.1007/978-3-319-76620-1_10

G structure. The existence of such reduction was used in ([OkUc98]) to prove the tractability of anomalous $E(\mathbb{Z}_n)$- Diffie-Hellman Problem.

The interesting general problem is the estimation of the number of positive integers $n \leq x$ that can be factored non-trivially or completely by the chosen polynomial time algorithm using the given oracle $\mathcal{O}$. The general problem of investigating the positive integers that can be factored non-trivially by the given algorithm $\mathcal{A}$ was already posed in [KnPa76]. In this paper we address to the analogous problem of complete factorization of n provided one can query polynomially many times the given oracle $\mathcal{O}$ answering with the number of points in $\mathbb{Z}_n^{d+1}$ on twists of Kummer hypersurfaces $y^k = f(x_1, x_2, \ldots, x_d) \bmod n$ for small prime values of k . The independent interest is focused on the deterministic polynomial time reduction of factoring n to computing the number of points of the above Kummer hypersurfaces modulo n (see Definition 6 below).

Let $n = p_1 \cdots p_s$ be a squarefree odd integer, $G = E(\mathbb{Z}_n) := E(\mathbb{Z}_{p_1}) \times \ldots \times E(\mathbb{Z}_{p_s})$, $E : y^2 = x^3 + ax + b$, $(k = 2)$. The known polynomial time, asymptotic reductions (i.e. reductions for all squarefree, odd positive integers $n \leq x$ with at most of $o(x)$ exceptions) depend strongly on the fact that such n are D- separable for $D = D(x)$ of polynomial growth (of $\log x$). D-separability means here that n is composed of the prime factors p for which there exists $\alpha \leq D(x)$, coprime to n, that is quadratic nonresidue $\bmod\, p$ and quadratic residue $\bmod\, n/p$, (see [DrPo17]). A possible way of generalization of such results goes towards replacing this requirement by a more general one related to arbitrary $k-$th order nonresidues (see Lemma 5 below for prime value $k = r$ and primes $p \mid n$, $p = 1(\bmod r)$). In such approach we are related to the more general superelliptic curves defined by the equations $y^k = f(x) \bmod n$.

The quantitative results are interesting in the case of deterministic reductions, proving the upper bounds for odd, squarefree, positive integers $n \leq x$ that might not be factored in polynomial time by the given algorithm $\mathcal{A}$. In case when $k \geq 2$ we obviously restrict ourselves to the numbers n having a prime factor $p = 1 \bmod k$. In this note we will show that factorization of a square-free integer n (having a prime divisor $p = 1 \bmod k$) can be reduced to the problem of computing the numbers $|A_\alpha(\mathbb{Z}_n)|$ of points in $\mathbb{Z}_n^{d+1}$ satisfying the equation $A_\alpha : \alpha y^k = f(x)$, where $f \in \mathbb{Z}_n[x_1, \ldots, x_d]$, using a similar way as for elliptic curves. This reduction is based on elementary properties of twists A_α for $\alpha \in \mathbb{Z}_n^*$. We will show that if one knows the numbers of points $|A_{\alpha_1}(\mathbb{Z}_n)|, \ldots, |A_{\alpha_k}(\mathbb{Z}_n)|$ for $\alpha_1, \ldots, \alpha_k \in \mathbb{Z}_n^*$ satisfying a suitable condition (see (4) Sect. 3), and some necessary condition is satisfied, then $\gcd(\sum_{i=1}^k |A_{\alpha_i}(\mathbb{Z}_n)|, n)$ is a non-trivial divisor of n. One should see the analogy between the above approach and factoring idea in the case when $G = \mathbb{Z}_n^*$ (see Lemma 5 below).

Our approach is based on the generalization of idea applied in [DrPo17] for the twisted elliptic curves E_α over $\mathbb{Z}_n$. Namely if $\alpha_1, \ldots, \alpha_h \in \mathbb{F}_q^*$ represent different classes of $\mathbb{F}_q^*/(\mathbb{F}_q^*)^k = \mathbb{F}_q^*/(\mathbb{F}_q^*)^h$, where $h = \gcd(q-1, k)$, then we will prove that $\sum_{i=1}^h |A_{\alpha_i}(\mathbb{F}_q)| = hq^d$ (see Lemma 9 below). Therefore if $\alpha_1, \ldots, \alpha_k \in \mathbb{Z}_n^*$ satisfy the condition (4), Sect. 3, then computing (5), Sect. 3, we detect a nontrivial divisor of n. In Sect. 4 we will estimate the probability of choosing the suitable tuple $(\alpha_1, \dots, \alpha_k) \in (\mathbb{Z}_n^*)^k$. Finally in case when $k = 2$ and n is a product of two primes we prove the unconditional, explicit upper bound for the number of squarefree, odd positive integers that might not be factored in deterministic time $O(\log^9 x)$ with the aid of oracle $OrdE$ (see Sects. 6 and 7).

Let us remark that the specific case when f is quadratic polynomial and $d = 2$, the number of points on the hyperbola $xy = 1(\bmod n)$ is equal to the value of Euler's totient function $\phi(n)$ and the related deterministic reduction is the reduction of factoring n to computing $\phi(n)$, which was investigated in detail in [DuPo17]. In this paper we also show that the above method can be modified to reduce factorization to the problem of computing the number of points in $\mathbb{P}^2(\mathbb{Z}_n)$ on the projective closure of curves $C : y^k = f(x_1)$, where $f \in \mathbb{Z}_n[x_1]$. Curves given by this equation over a field are called superelliptic, and arithmetic in their jacobian was studied in [GPS02]. However this approach will be the subject of a separate work.

2 Basic Notions and the Related Work

Notation

$\mathbb{N}$ – the set of all positive integers
$\mathbb{N}_*$ – the set of all odd, squarefree, positive integers
$\mathbb{N}_s$ – the set of all odd, squarefree positive integers having s prime factors
$\mathbb{N}_s(x, y)$ – the set of $n \in \mathbb{N}_s$, $n \le x$ such that $P_-(n) \ge y$
$\mathbb{N}_s(x)$ – the set of $n \in \mathbb{N}_s$, $n \le x$
Conventionally m, n stand for positive integers, while p, q, r for prime numbers (unless otherwise stated)
ϕ – Euler's totient function
$\omega(n)$ denotes the number of distinct prime divisors of n
$P_-(n)$ stands for the least prime divisor of n
$\pi(t)$ denotes the number of primes $\le t$
$\nu_q(n)$ denotes the highest power of q dividing n
$\#A$ stand for the cardinality of the (finite) set A
For positive arithmetical functions f and g the equality $f = \Theta(g)$ means that $f = O(g)$ and $g = O(f)$
$f \ll g$ means that $f = O(g)$
$\mathrm{ord}_n\, b$ denotes the order of $b \bmod n$, where $\gcd(b, n) = 1$
By $LN(r, n)$ we denote the least positive integer b (coprime to n) that is r-th power nonresidue modulo n, $(r \mid \phi(n))$.
By $LN(\chi)$ we denote the least Dirichlet character χ nonresidue, i.e. the least b such that $\chi(b) \notin \{0, 1\}$
For odd, squarefree number $n = p_1 p_2 \dots, p_s$, $p_1 < p_1 < \dots < p_s$ we denote by $sig(b, n)$ the $s-$ tuple $((b/p_1), (b/p_2), \dots, (b/p_s))$ of Legendre's symbols

Definition 1. *Let $p \mid n$ and r be a prime number dividing $p - 1$. The positive integer b is called $(r, p, n)-$witness if b is r-th power nonresidue $\bmod p$ and r-th power residue modulo q for some prime $q \mid n/p\, (q \ne p)$.*

Definition 2. *By $LS_p(r,n)$ we denote the least $(r,p,n)-$witness. Moreover*

$$LS(r,n) = \max_{p|n,\, r|p-1} LS_p(r,n).$$

The following notion of strong $(r,p,n)-$witness is actually related to the complete factorization of n.

Definition 3. *Let $p \mid n$ and r be a prime number dividing $p-1$. The positive integer b is called strong $(r,p,n)-$witness if b is r-th power nonresidue $\bmod p$ and r-th power residue modulo q for any prime $q \mid n/p\, (q \neq p)$.*

Definition 4. *By $LSS_p(r,n)$ we denote the least strong $(r,p,n)-$witness. Moreover*

$$LSS(r,n) = \max_{p|n,\, r|p-1} LSS_p(r,n).$$

Let us remark that if $n = pq$ is the product of two distinct primes, then the notions $LS_p(2,n)$ and $LSS_p(2,n)$ coincide. In what follows let $D = D(n)$ be a given function of n.

Lemma 5. *Let n and $\phi(n)$ be given, where n is odd, squarefree positive integer. Let $r \mid \phi(n)$ and assume that*

$$LS(r,n) \leq D(n).$$

Then n can be factored nontrivially in $O\big(D(n)\log^2 n\big)$ bit operations. Moreover if

$$LSS(r,n) \leq D(n),$$

then n can be factored completely in $O\big(D(n)\log^3 n\big)$ bit operations.

Proof. The case of nontrivial factorization follows easily form the proof of the complete factorization so we will prove only the second inequality. It follows by induction. We will show how to factor nontrivially any $d \mid n$ such that $r \mid \phi(d)$ and $\omega(d) \geq 2$. Let $d = p_1p_2\ldots p_s$. Since $r \mid \phi(d)$ there exists a prime divisor $p \mid d$ such that $\nu_r(p-1) \geq 1$. Choose $p \mid n$ such that $\nu_r(p-1) \geq \nu_r(q-1)$ for all $q \mid d/p$. We consider two cases.

(1) Assume that $\nu_r(p-1) = \nu_r(q-1)$ for all $q \mid d/p$. Then letting $b = LSS_p(r,d) \leq D$ we conclude that $\nu_r(\operatorname{ord}_p b) = \nu_r(p-1)$, while for any $q \neq p$ we have that $\nu_r(\operatorname{ord}_q b) < \nu_r(q-1)$, hence $\nu_r(\operatorname{ord}_p b) > \nu_r(\operatorname{ord}_q b)$. Without losing the generality we may assume that b is coprime to d/p, since otherwise we could easily detect a nontrivial prime factor of d. Therefore rising b to the powers of type $\phi(n)/r^m$ and computing their residues $\bmod n$, for $m = 1, 2, \ldots$, we obtain that for some $m \geq 1$ it holds

$$q \mid \gcd(b^{\phi(d)/r^m} - 1, d) \mid d/p,$$

giving the required factorization.

(2) Now assume that $\nu_r(p-1) > \nu_r(q-1)$ for some $q \mid d/p$. Then we obtain for $b = LSS_p(r,d) \leq D$ that b is r-th power nonresidue modulo p, where as before we may assume that b is coprime to d/p. Hence $\nu_r(\mathrm{ord}_p\, b) = \nu_r(p-1) > \nu_r(q-1) \geq \nu_r(\mathrm{ord}_q\, b)$. Therefore rising b to the suitable power of type $\phi(n)/r^m$ we obtain similarly as above the nontrivial divisor of d. The number of bit operations is $O(D(n)\omega(n)\log^2 n) = O(D(n)\log^3 n)$. This completes the proof of Lemma 5.

The above lemma can be viewed as a starting point to extend the results proved in [DuPo17, DrPo17] to general oracle $\mathcal{O} = NuSol(g)$, answering with the number of solutions of $g(x_1, ..., x_d, y) = 0 (\bmod n)$, where $g \in \mathbb{Z}_n[x_1, \ldots, x_d, y]$ is of type $g(x_1, \ldots, x_d, y) = y^k - f(x_1, \ldots, x_d)$. In order to formulate the related quantitative results we begin with the following

Definition 6. *By $F_s\left(x, \mathcal{A}, O, t_{\mathcal{A}}, t_O\right) \left(F_*\left(x, \mathcal{A}, O, t_{\mathcal{A}}, t_O\right)\right)$ respectively, we denote the number of $n \in \mathbb{N}_s\, (n \in \mathbb{N}_*), n \leq x$ that can by completely factored by algorithm $\mathcal{A}$ in deterministic time $t_{\mathcal{A}}$ with at most t_O queries the oracle $\mathcal{O}$.*

Let $\mathcal{O} = \varPhi$ be the oracle answering the value of the Euler totient function $\phi(n)$ for positive integer $n \in \mathbb{N}_*$. Approaching the positive answer on the problem posed by Adleman and McCurley (see [AdCu94]), in the paper [DuPo17] it is proved that there exists deterministic algorithm $\mathcal{A}_1$ such that $F_*\left(x, \mathcal{A}_1, \mathcal{O}, t_{\mathcal{A}_1}, t_{\mathcal{O}}\right) = \#\mathbb{N}_*(x)\left(1 + O\left(1/A_1(x)\right)\right)$, where $A_1(x) = (\log x)^{c_1/\log\log\log x}$ for some absolute positive constant c_1, $t_{\mathcal{A}} = O\left((\log x)^5\right)$ and $t_{\mathcal{O}} = O(\log x)$. We recall that the oracle $\varPhi$ is a particular case of the oracle $NuSol(g)$ for $g(x,y) = xy - 1 (\bmod n)$.

Let $\mathcal{O} = OrdE$ where $OrdE$ is the oracle answering with the order of elliptic curve $E(\mathbb{Z}_n)$, over the ring $\mathbb{Z}_n$, $B = B(n)$ and $\mathcal{B}_p(d)$ be the sequence of traces $t_p(E_p(b))$ of elliptic curves given by the Weierstrass equation: $E_p(b) : y^2 = x^3 + x + b (\bmod p)$, $b \leq B$, lying in the arithmetical progression $0 (\bmod d)$, $d \mid p+1$. In [DrPo17] it is proved that if the average value (over $p \leq x$) of $|\#\mathcal{B}_p(d) - \#\mathcal{B}_p/d|$ is bounded by $\log^\delta x$, then there exists deterministic algorithm $\mathcal{A}_2$ such that

$$F_2\left(x, \mathcal{A}_2, \mathcal{O}, t_{\mathcal{A}_2}, t_O\right) = \#\mathbb{N}_2(x)\left(1 + O\left(1/A_2(x)\right)\right),$$

where $A_2(x) = (\log\log x)^c$, $t_{\mathcal{A}_2} = (\log x)^{8+2\delta}$ and $t_{\mathcal{O}} = (\log x)^{4+\delta}$ for arbitrary fixed $c < 1$.

In this paper we will present the theoretical background for the oracle $\mathcal{O} = NuSol(g)$, where $g(x_1, \ldots, x_d, y) = y^k - f(x_1, \ldots, x_d)$. In particular case when $k = 2$ and $f(x_1) = x_1^3 + ax_1 + b$ defines a projective elliptic curve E, we will show (see Sect. 7) the deterministic algorithm $\mathcal{A}_3$ such that for $\mathcal{O} = OrdE$ we have

$$F_2\left(x, \mathcal{A}_3, \mathcal{O}, t_{\mathcal{A}_3}, t_O\right) = \#\mathbb{N}_2(x)\left(1 + O(1/A_3(x))\right), \tag{1}$$

where $A_3(x) = (\log x)^{c/\log\log\log x}$, for some positive constant c. Here $t_{\mathcal{A}_3} = O(\log^9 x)$ and $t_{\mathcal{O}} = O(\log^2 x)$. The investigation of the above counting function is based on the following

Definition 7. *The positive integer $n = pq, p < q$ is called $(y, D)-$ admissible if $P_-(n) \geq y$ and there exist positive integers $b_1, b_2, b_3 \leq D$ such that $sig(b_1, n) = (-1, 1)$, $sig(b_2, n) = (1, -1)$ and $sig(b_3, n) = (-1, -1)$. If n is not $(y, D)-$ admissible it is called $(y, D)-$ exceptional.*

To prove equality (1) we will need the upper bound for $(y, D)-$ exceptional numbers $n = pq$ with $y = \Theta(\log n)$ and $D = \Theta(\log^2 n)$. The proof is principally based on Lemma 13, Lemma 17 and the following

Proposition 8. *Let $n = pq \in \mathbb{N}_2(x, y)$ and assume that for any $m \in \{p, q, pq\}$ we have that $LN(2, m) \leq B$. Then n is $(y, B^2)-$ admissible and $LSS(2, n) \leq B^2$.*

Proof. By the condition $LN(2, n) \leq B$ we can assume without losing the generality that there exists $b_1 \leq B$ is such that $sig(b_1) = (-1, +1)$. Let $b_2 = LN(2, q) \leq B$. Then $b_1 b_2$ has the complementary signature in question that is $\leq B^2$. Therefore for $m = p$ and $m = q$ we have that $LSS_m(2, n) \leq B^2$ and the last conclusion follows.

3 Reduction of Factorization to Computing the Number of Points on $A(\mathbb{Z}_n)$

Let $n = \prod_{i=1}^s p_i$ be a square-free integer, where p_i are odd primes, and let $A(\mathbb{Z}_n) = \{(x, y) \in \mathbb{Z}_n^{d+1} : y^k = f(x)\}$, where $k > 1$ and $f \in \mathbb{Z}_n[x_1, \ldots, x_d]$. Let $A_\alpha : \alpha y^2 = f(x)$ be the twist of A for $\alpha \in \mathbb{Z}_n^*$. We will show that if we know the numbers of points $|A_{\alpha_1}(\mathbb{Z}_n)|, \ldots, A_{\alpha_k}(\mathbb{Z}_n)|$, where $\alpha_1, \ldots, \alpha_k$ satisfy condition (4) below for a prime $q \mid n$ and n/q does not divide $|A(\mathbb{Z}_{n/q})|$ (see Remark 10), then $\gcd(\sum_{i=1}^k |A_{\alpha_i}(\mathbb{Z}_n)|, n)$ is a non-trivial divisor of n. We will need the following elementary facts. For $m \mid n$ let $A(\mathbb{Z}_m)$ be the set of $\mathbb{Z}_m$-points on the reduction $A \bmod m$. By the CRT reductions mod primes $p_i \mid n$ induce the bijection

$$A(\mathbb{Z}_n) \to \prod_{i=1}^s A(\mathbb{F}_{p_i}). \tag{2}$$

Two twists A_α, A_β for $\alpha, \beta \in \mathbb{Z}_n^*$ have the same number of points over $\mathbb{Z}_n$ if $\alpha \equiv \beta \bmod (\mathbb{Z}_n^*)^k$ (kth powers in $\mathbb{Z}_n^*$). It is enough to show this for A defined over a finite field $\mathbb{F}_q$. For $\alpha, \beta \in \mathbb{F}_q^*$ we have the isomorphism $A_\alpha \to A_\beta$, $(x, y) \to (x, (\alpha/\beta)^{1/k} y)$, defined over the smallest extension of $\mathbb{F}_q$ containing a kth root $(\alpha/\beta)^{1/k}$. Thus if $\alpha \equiv \beta \bmod (\mathbb{F}_q^*)^k$, then A_α, A_β are isomorphic over $\mathbb{F}_q$. We will need the following fact, which is well-known for elliptic curves $y^2 = x^3 + ax + b$.

Lemma 9. *Assume that each class of $\mathbb{F}_q^*/(\mathbb{F}_q^*)^k = \mathbb{F}_q^*/(\mathbb{F}_q^*)^h$, where $h = \gcd(k, q-1)$, is represented by exactly k/h elements among $\alpha_1, \ldots, \alpha_k \in \mathbb{F}_q^*$. Then*

$$\sum_{i=1}^k |A_{\alpha_i}(\mathbb{F}_q)| = kq^d. \tag{3}$$

Proof. Assume that $\alpha_1, \ldots, \alpha_h$ represent different classes $U_1, \ldots, U_h$ of $\mathbb{F}_q^*/(\mathbb{F}_q^*)^h$. Let $Z_0 = \{x \in \mathbb{F}_q^d : f(x) = 0\}$ and $Z_i = \{x \in \mathbb{F}_q^d : f(x) \in U_i\}$ for $i = 1, \ldots, h$. Then $\mathbb{F}_q^d = \cup_{i=0}^h Z_i$ is the disjoint union. For each $x \in Z_i$, $i \geq 1$, the equation $\alpha_i y^k = f(x)$ has exactly h solutions $y \in \mathbb{F}_q$, because $f(x)/\alpha_i \in (\mathbb{F}_q^*)^h$, so there is a solution $y \in \mathbb{F}_q$, and remaining solutions over $\mathbb{F}_q$ are of the form $\zeta_h^i y$, $i = 1, \ldots, h$, where an hth primitive root of unity $\zeta_h \in \mathbb{F}_q$, since $h \mid q - 1$. Thus $|A_{\alpha_i}(\mathbb{F}_q)| = |Z_0| + h|Z_i|$, so $\sum_{i=1}^h |A_{\alpha_i}(\mathbb{F}_q)| = h|Z_0| + h\sum_{i=1}^h |Z_i| = hq^d$, which gives (3) since each class of $\mathbb{F}_q^*/(\mathbb{F}_q^*)^h$ is represented by k/h elements α_i.

Given $A/\mathbb{Z}_n$ to reduce factorization of n to computation $|A_\alpha(\mathbb{Z}_n)|$ for $\alpha \in \mathbb{Z}_n^*$ we need to choose

$$\begin{aligned}&\alpha_1, \ldots, \alpha_k \in \mathbb{Z}_n^*, \text{which reduced } \bmod q \text{ satisfy assumptions of}\\ &\text{Lemma 9 for exactly one prime } q \mid n \text{ and for each prime } p \mid n,\\ &p \neq q, \text{ all } \alpha_i \bmod p \text{ are in the same class of } \mathbb{F}_p^*/(\mathbb{F}_p^*)^k.\end{aligned} \tag{4}$$

If (4) is satisfied, then $|A_{\alpha_i}(\mathbb{Z}_{n/q})| = |A_{\alpha_1}(\mathbb{Z}_{n/q})|$ for $i = 1, \ldots, k$. Hence $\sum_{i=1}^k |A_{\alpha_i}(\mathbb{Z}_n)| = |A_{\alpha_1}(\mathbb{Z}_{n/q})|(\sum_{i=1}^k |A_{\alpha_i}(\mathbb{F}_q)|) = kq^d|A_{\alpha_1}(\mathbb{Z}_{n/q})|$. Thus if n/q does not divide $|A_{\alpha_1}(\mathbb{Z}_{n/q})|$, then

$$\gcd(n, \sum_{i=1}^k |A_{\alpha_i}(\mathbb{Z}_n)|) \tag{5}$$

is a nontrivial divisor of n (we assume that k is small and is not divisible by prime factors of n).

Remark 10. Note that the above reduction fails if n divides $|A_\alpha(\mathbb{Z}_n)|$ for each $\alpha \in \mathbb{Z}_n^*$. This happens if $\gcd(q-1, k) = 1$ for each prime $q \mid n$, since then $|A_\alpha(\mathbb{F}_q)| = q^d$, because for each $x \in \mathbb{F}_q^d$ the equation $y^k = f(x)$ has unique solution $y \in \mathbb{F}_q$. Similarly, if we can compute from the equation $y^k = f(x_1, \ldots, x_d)$ one variable as a function of the remaining variables. Clearly it may also happen that $|A_\alpha(\mathbb{Z}_n)| = n^d$ if the above do not hold. For example, if $A : y^2 = x^3 + ax + b$ is an elliptic curve over $\mathbb{Z}_n$ and for each prime $q \mid n$ the reduction $E \bmod q$ is supersingular (i.e., the trace $t_q = 0$ and for the affine part $|A_\alpha(\mathbb{F}_q)| = q$), then $|A(\mathbb{Z}_n)| = n$.

The following corollary shows that the k-tuple satisfying (4) can be efficiently computed provided $LSS_q(k, n)$ is small. Let us also remark that since the notions $LSS(2, n)$ and $LS(2, n)$ are equivalent we obtain in view of Proposition 8 that there are $\#\mathbb{N}_2(x)\big(1 + O((\log x)^{-c_1/\log\log\log x})\big)$ numbers $n \in \mathbb{N}_2(x)$ for which $LSS(2, n)$ is $\leq c_2 \log^2 n$ for some positive constants c_1 and c_2.

Corollary 11. *Let r be a fixed prime dividing $q - 1$, where $q \mid n$. Let $b = LSS_q(r, n)$. Then the r-tuple $(\alpha_1, \alpha_2, \ldots, \alpha_r) = (b, b^2, \ldots, b^r)$ satisfies the requirement (4) (with $k = r$).*

Proof. Let r be a fixed prime and $q = 1(\bmod r)$ be a prime divisor of n. Letting $b = LSS_q(r, n)$ we see that the powers of b represent all distinct cosets of $\mathbb{F}_q^*/(\mathbb{F}_q^*)^r$. On the other hand all such powers are in the same class $1 \in \mathbb{F}_p^*/(\mathbb{F}_p^*)^r$ for all primes $p \mid n/q$. This proves that the requirement (4) is satisfied with $k = r$.

4 Estimate for the Frequency of the Related Tuples $(\alpha_1, \dots, \alpha_k) \in \mathbb{Z}_n^k$

In this section we estimate the probability of choosing k-tuple $(\alpha_1, \dots, \alpha_k) \in \mathbb{Z}_n^k$ satisfying (4). Let $n = \prod_{i=1}^s p_i$, $h_i = \gcd(k, p_i - 1)$, and renumber p_i such that $h_i > 1$ for $i \leq u$ and $h_i = 1$ for $i > u$. Then we have

$$\#\{(\alpha_1, \dots, \alpha_k) \in {\mathbb{Z}_n^*}^k \text{ satisfying (4)}\}\frac{1}{\varphi(n)^k} \geq \frac{uk!}{k^{sk-s+1}}. \tag{6}$$

Note that if $k \mid p_i - 1$ for each $i = 1, \dots, s$, then we have equality. Thus this probability quickly decreases with s and k. To show (6) we will make use of the isomorphism

$$\mathbb{Z}_n^*/(\mathbb{Z}_n^*)^k \cong \prod_{i=1}^{s} \mathbb{Z}_{p_i}/(\mathbb{Z}_{p_i}^*)^{h_i}. \tag{7}$$

Since the classes of $\mathbb{Z}_n^*/(\mathbb{Z}_n^*)^k$ have the same number of elements, the LHS of (6) is equal to the probability of choosing a k-tuple $(U_1, \dots, U_k)$ of classes U_i in $\mathbb{Z}_n^*/(\mathbb{Z}_n^*)^k$ satisfying the following: there exists exactly one $i_0 \leq u$ such that if $U_i \cong U_{i1} \times \cdots \times U_{is}$ by (7), then each class of $\mathbb{F}_{p_{i_0}}^*/(\mathbb{F}_{p_{i_0}}^*)^{h_{i_0}}$ appears in a sequence $U_{1,i_0}, \dots, U_{k,i_0}$ exactly k/h_{i_0} times, and $U_{ji} = U_{1i}$ for each $i \neq i_0$, $j = 1, \dots, k$. The number of such k-tuples $(U_1, \dots, U_k)$ is equal to $\sum_{i_0=1}^{u} \frac{h}{h_{i_0}} \frac{k!}{((k/h_{i_0})!)^{h_{i_0}}}$, where $h = \prod_{i=1}^s h_i$, $\frac{h}{h_{i_0}}$ is the number of choices of U_{ij} on positions $i \neq i_0$, and $\frac{k!}{((k/h_{i_0})!)^{h_{i_0}}}$ is the number of sequences $(U_{i_0 1}, \dots, U_{i_0 k})$ such that each class of $\mathbb{F}_{p_{i_0}}^*/(\mathbb{F}_{p_{i_0}}^*)^{h_{i_0}}$ appears in this sequence exactly k/h_{i_0} times. Thus we have to show

$$\text{LHS of (6)} = \frac{\sum_{i=1}^{u} \frac{h}{h_i} \frac{k!}{((k/h_i)!)^{h_i}}}{h^k} \geq u\frac{k!}{k^{sk-s+1}}.$$

Dividing by $k!$ this follows from

$$\frac{h/h_i}{h^k((k/h_i)!)^{h_i}} \geq \frac{h/h_i}{h^k((k/h_i)^{k/h_i})^{h_i}} = \frac{h/h_i}{h^k(k/h_i)^k}$$

$$= \frac{1}{(h/h_i)^{k-1}k^k} \geq \frac{1}{(k^{s-1})^{k-1}k^k} = \frac{1}{k^{sk-s+1}}.$$

Remark 12. Note that for $k = 2$ factorization of n can be reduced to computation the number of points in $\mathbb{Z}_n^{d+1}$ satisfying the equation $A' : y^2 + f_1(x)y + f_2(x) = 0$, where $f_1, f_2 \in \mathbb{Z}_n[x_1, \dots, x_d]$, because by completing the square

$(y+f_1/2)^2+f_2-f_1^2/4=0$ we have the isomorphism $A' \to A : y^2 = f_1^2/4-f_2(x)$ given by $(x,y) \mapsto (x, y+f_1(x)/2)$. In particular, it is enough to compute the number of points on quadrics $F(x_1,\ldots,x_d)=0$ in $\mathbb{Z}_n^d$, $\deg F=2$, $d \geq 2$. If no squares $x_1^2,\ldots,x_d^2$ do appear in F, that is $F=\sum_{i \neq j} a_{ij}x_ix_j+$ terms of degree <2, then substituting $(x_i+x_j, x_i-x_j) \to (x_i,x_j)$ for some $a_{ij} \neq 0$ we get $x_i^2-x_j^2$.

5 Reduction of Factorization to Computation the Number of Points in $\mathbb{P}^2(\mathbb{Z}_n)$ on Twists of Superelliptic Curves

If we slightly modify the previous method, we will obtain reduction of factorization of a square-free integer n to computation the number $|C(\mathbb{Z}_n)|$ of points in $\mathbb{P}^2(\mathbb{Z}_n)$ on the superelliptic curves $C : z^{\max\{k,\deg f\}}((y/z)^k - f(x/z)) = 0$. This extends the reduction of factorization to the problem of computing the order $|E(\mathbb{Z}_n)|$ of ellitpic curves $E/\mathbb{Z}_n$. We do not consider the reduction to the problem of computing the number of points on the projective closure of the set $y^k = f(x_1,\ldots,x_d)$ for $d>1$, because we need to know the number of points at infinity over a field. Superelliptic curves have one point at infinity over a field if $k \neq \deg f(x)$, and at most k points if $k = \deg f$. We first assume that $k \neq \deg f(x)$, and later extend on the case $k = \deg f(x)$.

Recall that the projective plane $\mathbb{P}^2(\mathbb{Z}_n)$ is defined to be the set of equivalence classes of primitive triples in $\mathbb{Z}_n^3$ (i.e., triples (x_1,x_2,x_3) with $\gcd(x_1,x_2,x_3,n)=1$) with respect to the equivalence $(x_1,x_2,x_3) \sim (y_1,y_2,y_3)$ if $(x_1,x_2,x_3) = u(y_1,y_2,y_3)$ for $u \in \mathbb{Z}_n^*$. By the CRT we have the bijection $C(\mathbb{Z}_n) \to \prod_{i=1}^s C(\mathbb{F}_{p_i})$.

Suppose that we know the numbers $|C_{\alpha_i}(\mathbb{Z}_n)|$ for $\alpha_1,\ldots,\alpha_k$ satisfying (4) for a prime $q \mid n$. If $l = \gcd(|C_{\alpha_1}(\mathbb{F}_q)|,\ldots,|C_{\alpha_k}(\mathbb{F}_q)|)$ is sufficiently small, i.e., $l \leq O(\log^\alpha n)$ for some $\alpha>0$, then varying $m \leq O(\log^\alpha n)$ for $m=l$ we will find the prime

$$q = \frac{m}{kD}(\sum_{i=1}^{k} |C_{\alpha_i}(\mathbb{Z}_n)|) - 1, \tag{8}$$

where $D = \gcd(|C_{\alpha_1}(\mathbb{Z}_n)|,\ldots,|C_{\alpha_k}(\mathbb{Z}_n))|$. This follows from the fact that $\sum_{i=1}^k |C_{\alpha_i}(\mathbb{F}_q)| = k(q+1)$ by Lemma 9 including the point at infinity. Since $|C_{\alpha_i}(\mathbb{F}_p)| = |C_{\alpha_1}(\mathbb{F}_p)|$ for $i=1,\ldots,k$ and primes $p \mid n$, $p \neq q$, we have $D = l|C_{\alpha_1}(\mathbb{Z}_{n/q})|$ and $\sum_{i=1}^k |C_{\alpha_i}(\mathbb{Z}_n)| = |C_{\alpha_1}(\mathbb{Z}_{n/q})|(\sum_{i=1}^k |C_{\alpha_i}(\mathbb{F}_q)|) = k(q+1)$ $|C_{\alpha_1}(\mathbb{Z}_{n/q}|)$. Thus $\sum_{i=1}^k |C_{\alpha_i}(\mathbb{Z}_n)| = k(q+1)D/l$, which gives (8).

Remark. Note that the assumption that l is small is not satisfied in the situation of Remark 10. For example, for an elliptic curve it is not satisfied if $\gcd(q+1-t_q,\ q+1+t_q)$, and so $\gcd(q+1, t_q)$, is large, which is very unlikely, because traces of elliptic curves are almost uniformly distributed in Hasse's interval $[-2\sqrt{q},\ 2\sqrt{q}]$.

Now assume that $k = \deg f(x)$ and the above assumptions on q and $\alpha_1,\ldots,\alpha_k$ hold. The number of points at infinity on C over $\mathbb{F}_q$ is equal to the number of roots $y^k = a_k$, where a_k is the leading coefficient of f. Applying Lemma 9 to the affine part of C and to the equation $y^k = a_k$ we also have $\sum_{i=1}^{k} |C_{\alpha_i}(\mathbb{F}_q)| = k(q+1)$, which gives (8).

6 Investigation of $F_2(x, \mathcal{A}_3, \mathcal{O}, t_{\mathcal{A}_3}, t_{\mathcal{O}})$

In this section we will prove the estimate for $(y, D)-$ exceptional numbers and describe the algorithm $\mathcal{A}_3$ with the oracle $\mathcal{O} = OrdE$. In this connection let us denote by $Adm_2(x, y, D)$ the set of $(y, D)-$ admissible numbers and by $Exc_2(x, y, D)$ the related set of $(y, D)-$ exceptional numbers $n \in \mathbb{N}_2(x, y)$ (see Sect. 2 for definitions).

Let $\mathcal{N}_2(x, B)$ denote the set of positive integers $n \in \mathbb{N}_2(x)$ such that $LN(2, n) \leq B$. We will first prove the asymptotic equality for $\#\mathcal{N}_2(x, B)$ for $B = B(x)$ of magnitude order $\log x$. The main ingredient in the proof is the following

Lemma 13. *(see* [LaWu08]*) There exists a positive absolute constant* c_0 *such that for* $\gamma_0 = \log 2/48$ *we have that* $LN(\chi_d) \leq c_0 \log |d|$, *for all but except* $O(x^{1-\gamma_0/\log\log x})$ *values of* $|d| \leq x$.

Corollary 14. *For* $B = B(x) = c_0 \log x$ *with* c_0 *as above and arbitrary positive constant* $c' < \log 2/48$ *we have*

$$\#\mathcal{N}_2(x, B(x)) = \#\mathbb{N}_2(x)\left(1 + O\left(x^{-c'/\log\log x}\right)\right). \tag{9}$$

Proof. Let n be any odd, positive, squarefree integer. Consider a quadratic Dirichlet character $\chi(\bmod |n'|)$ satisfying the equality (see e.g. [Dav67], Sect. 5)

$$\left(\frac{\alpha}{n}\right) = \left(\frac{n'}{\alpha}\right)_K =: \chi_{n'}(\alpha) \tag{10}$$

where n' is the fundamental discriminant $n' = \prod p'$, with $p' = (-1)^{(p-1)/2}p$ for primes $p \mid n$ and $\left(\frac{n'}{\alpha}\right)_K$ denotes the Kronecker symbol. Obviously if for some $\alpha \leq B(|n'|)$ we have $\chi_{n'}(\alpha) = -1$, then $LN(2, n) \leq B(n)$. To prove (9) it is now sufficient to apply Lemma 13 with $\mid d \mid = n$ yielding

$$0 \leq \#\mathbb{N}_2(x) - \#\mathcal{N}_2(x, B(x)) \ll x^{1-\gamma_0/\log\log x}$$

Therefore in view of the asymptotic equality (11) we conclude that

$$\#\mathcal{N}_2(x, B(x)) = \#\mathbb{N}_2(x) + O(x^{1-\gamma_0/\log\log x}) = \#\mathbb{N}_2(x)\left(1 + O\left(x^{-c'/\log\log x}\right)\right)$$

for any $c' < \gamma_0$ and $x > x_0(c')$, as required.

Lemma 15. *(see [HaWr79], Theorem 437) We have the following asymptotic equality*

$$\#\mathbb{N}_2(x) = \frac{x}{\log x}(\log\log x)\big(1+o(1)\big), \tag{11}$$

as x tends to infinity.

Corollary 16. *There exist positive absolute constants c_1, c_1' and c_2 such that we have for $1 < y < \sqrt{x}/2$ the inequalities*

$$\frac{c_1 x}{\log x}\left(\log\frac{\log x}{\log y} + O\left(\frac{1}{\log y} + \frac{\log y}{\log x}\right)\right) \le \#\mathbb{N}_2(x, y) \le \frac{c_2 x}{\log x}(\log\log x), \tag{12}$$

In particular if $y = \log x$ and x is sufficiently large, then we have

$$\frac{c_1' x}{\log x}(\log\log x) \le \#\mathbb{N}_2(x,\, y) \le \frac{c_2 x}{\log x}(\log\log x), \tag{13}$$

Proof. The proof of an upper bound follows immediately from Lemma 15. To prove the lower bound we derive

$$\#\mathbb{N}_2(x,\, y) \ge \sum_{y \le p_1 < x/(2y)} \; \sum_{y \le p_2 < x/p_1} 1 \ge \sum_{y \le p_1 < x/(2y)} \frac{x/(2p_1)}{\log(x/(2p_1))} \gg \frac{x}{\log x} \sum_{y \le p_1 < x/(2y)} p_1^{-1} \tag{14}$$

Applying the familiar estimate for the prime reciprocal series the above expression is

$$\log\left(\frac{\log(x/(2y))}{\log y}\right) + O\left(\frac{1}{\log y}\right) = \log\left(\frac{\log x}{\log y}\left(1 - \frac{\log(2y)}{\log x}\right)\right) + O\left(\frac{1}{\log y}\right)$$

$$= \log\left(\frac{\log x}{\log y}\right) + O\left(\frac{1}{\log y} + \frac{\log y}{\log x}\right)$$

which proves the first estimate. The second follows letting $y = \log x$ and x be sufficiently large.

Lemma 17. *There exist some positive, absolute constants c_1 and c, so that for $y = \log x$ and $D(x) = (c_1 \log x)^2$ we have*

$$\#Adm_2(x, y, D(x)) = \pi(x)\left(1 + O(1/A_3(x))\right),$$

where $A_3(x) = (\log x)^{c/\log\log\log x}$.

Proof. We will prove that for the choice of y and D as above we obtain that $\#Exc_2(x, y, D)$ is $\ll \pi(x)/A_3(x)$. Assume that $n \in Exc_2(x, y, D)$, where $n = pq$, $p, q \ge y$. If $LN(2, m) \le \sqrt{D}$ for any $m \in \{p, q, pq\}$ then by By Proposition 8 n would be (y, D)-admissible. Therefore $LN(2, m) > \sqrt{D}$ for some $m \in \{p, q, pq\}$. By Lemma 13 we have that the number of $m \in \mathbb{N}_*(t)$ such that $LN(2, m) \ge c_1 \log t$ is $\ll t^{1-c/\log\log t}$, giving the required bound for thel numbers $m = pq$.

Therefore it remains to estimate the number of $n \in \mathbb{N}_2(x)$ having a prime divisor $p \geq y$ such that $LN(2,p) \geq c_1 \log p$.

In this connection let e be a base of natural logarithm and $B = B(t) = c_1 \log t$. Let $P_B(t)$ be the counting function of primes $p \leq t$ such the $LN(2,p) \geq B(t)$. By Lemma 13 we obtain $P_B(t) \ll t^{1-c/\log\log t}$. Hence writing $n = pq$ and applying the bound for the counting function of primes p such that $LN(2,p) \geq B(p)$ we obtain by the partial summation

$$\#Exc_2(x,y,D) \ll x^{1-c/\log\log x} + \sum_{y \leq q \leq x,\ LN(2,q) \geq B} \ \sum_{p \leq x/q} 1$$

$$x^{1-c/\log\log x} + \sum_{y \leq q \leq x/e,\ LN(2,q) \geq B} \frac{x/q}{\log(x/q)}$$

$$\ll x^{1-c/\log\log x} + P_B(x) + x \int_y^{x/e} \frac{P_B(t)}{t^2\left(\log(x/t)\right)} dt$$

$$\ll x^{1-c/\log\log x} + x \int_y^{x/e} \frac{1}{t^{1+c/\log\log t}(\log(x/t))} dt$$

Since $t^{c/\log\log t} > \log t$ we obtain that

$$\#Exc_2(x,y,D) \ll x^{1-c/\log\log x} + \frac{x}{y^{c'/\log\log y}} \int_y^{x/e} \frac{1}{t^{1+c'/\log\log t}(\log t)\log(x/t)} dt$$

for some $c' < c$. Hence changing the variables the last expression is

$$\ll x^{1-c/\log\log x} + \frac{x}{y^{c'/\log\log y}(\log x)} \int_{\log y/\log x}^{1-1/\log x} \frac{1}{u(1-u)} du$$

$$\ll x^{1-c/\log\log x} + \frac{x}{y^{c'/\log\log y}(\log x)} \left(\left| \log\left(\frac{\log y}{\log x}\right) \right| + \left| \log\left(\frac{1}{\log x}\right) \right| \right)$$

$$\ll \frac{x \log\log x}{(\log x)(y^{c'/\log\log y})} \ll \pi(x)/A_3(x)$$

where $y = \log x$. This gives the required estimate for the number of exceptions and therefore completes the proof of Lemma 17.

Now we estimate the running time of the above algorithm. In view of [LePo05] the first step takes $O\left(y \log^{6+\varepsilon} n\right)$ bit operations. In Step 2 we query $O(D(n))$ times the oracle $\mathcal{O}$. Finally Steps 3–4 run in $O\big(D(n)^3 \log^3 n\big)$ bit operations. Hence specifying the parameters $y = \log x$, $D = D(x) = \Theta(\log^2 x)$ the total running time of Algorithm $\mathcal{A}_3$ is $O(y \log^{6+\varepsilon} x + D(x)^3 \log^3 x) = O(\log^9 x)$. Moreover we have the following

Algorithm 1. ($\mathcal{A}_3$) Complete factoring with $\mathcal{O} = OrdE$ oracle

Input: squarefree positive integer $n \in \mathbb{N}_2(x)$ and parameters $y, D = D(n)$
Output: complete factorization $n = pq$ or the answer that n is (y, D)-exceptional.

1. Check whether $p \mid n$ for all $p \leq y$ using the deterministic primality testing
2. For each $1 \leq \alpha \leq D(n)$ use the oracle $\mathcal{O} = OrdE$ answering the values of $|E_\alpha(\mathbb{Z}_n)|$, where $1 \leq \alpha \leq D(n)$.
3. For arbitrary tuples (n_2, n_3, n_4) with coordinates belonging to the set $|E_\alpha(Z_n)|$ for $1 < \alpha \leq D(n)$, check if there exists n_1, such that $n_1 n_4 = n_2 n_3$. If no, continue Step 3 with another choice of (n_2, n_3, n_4). Else solve the following quadratic equation
$$Ax^2 + Bx + C = 0, \tag{15}$$
where $A = 2\left(n_4 n_2^{-1} + 1\right)$, $B = 4(n-1) - \left(n_4 n_2^{-1} + 1\right)(n_1 + n_2)$, $C = -2(n_1 + n_2)$.
4. If x is a positive integer then compute
$$p_1 = \frac{n_1 + n_2}{2x} - 1. \tag{16}$$

If p_1 is a nontrivial divisor of n the algorithm returns the required complete factorization $n = p_1 \frac{n}{p_1}$ and halt, otherwise return to Step 3. If no prime divisor of n is detected for all $\alpha \leq D(n)$ we declare n as $(y, D(n))-$ exceptional and halt.

Remark 18. Since in Step 1 we discover the prime factor of n that is $\geq y = \log x$, one may assume without losing the generality that $n \in \mathbb{N}_2(x, y)$. By Lemma 17 all but except $O(\pi(x)/A_3(x))$ positive integers $n \in \mathbb{N}_2(x, y)$ are $(y, D)-$admissible, where $D = B^2 = \Theta(\log^2 x)$ and any such number has the witness $b \leq D(x)$ having the arbitrary nontrivial (i.e. $\neq (1, 1)$) signature $sig(b, n)$. Therefore such n is completely factored in $O\big(\log^9 n\big)$ deterministic time with at most $O\big(\log^2 n\big)$ queries the oracle $\mathcal{O} = OrdE$.

7 Completion of the Proof of Estimate (1)

In this section we will prove the correctness of algorithm $\mathcal{A}_3$. We start from the general approach. Let $n = \prod_{i=1}^{s} p_i$ be a product of different odd primes p_i where s is arbitrary, fixed positive integer ≥ 2.

The general idea deals with the polynomial time reduction of factorization of n to the problem of computing the number of points $|C_\alpha(\mathbb{Z}_n)|$ in $\mathbb{P}^2(\mathbb{Z}_n)$ on twists of the curve $y^2 = f(x)$. Note that over a field $\mathbb{F}_q$ we can write the number of points as $|C(\mathbb{F}_q)| = q + 1 - t_q$ for an integer t_q (which is the trace if C is an elliptic curve). Then from Lemma 9 we have $|C_\alpha(\mathbb{F}_q)| = q + 1 - (\frac{\alpha}{q}) t_q$ for $\alpha \in \mathbb{F}_q^*$, where $(\frac{\alpha}{q})$ is the Legendre symbol. Thus we have $|C_\alpha(\mathbb{Z}_n)| = \prod_{i=1}^{s}(p_i + 1 - (\frac{\alpha}{p_i}) t_i)$ for $\alpha \in \mathbb{Z}_n^*$. Note that there is a one-to-one correspondence between classes of $\mathbb{Z}_n^*/(\mathbb{Z}_n^*)^2$ and 2^s different sequences of Legendre symbols $((\frac{\alpha}{p_1}), \ldots, (\frac{\alpha}{p_s}))$.

For a generic curve we expect to obtain 2^s different numbers $|C_\alpha(\mathbb{Z}_n)|$ for $\alpha \leq D(n)$ and $\alpha \in \mathbb{Z}_n^*$. If this is the case let n_j be a permutation of them enumerated by $j = (j_1, \dots, j_s) \in \{\pm 1\}^s$. Then we have to solve the following system of equations (including the permutations of n_j).

$$\begin{cases} p_1 \cdots p_s = n \\ \big(p_1 + 1 - j_1 t_1\big) \cdots \big(p_s + 1 - j_s t_s\big) = n_j \text{ for } j = (j_1, \dots, j_s) \in \{\pm 1\}^s. \end{cases}$$

For $s > 2$ the algebraic set given by this system has positive dimension, and we have no efficient method to find integral points in this set. However for $s = 2$ the integral solutions can be determined efficiently since the system has finitely many solutions. Namely let us substitute in the system $x_i = p_i + 1 - t_i$ and $\bar{x_i} = p_i + 1 + t_i$. We have the following system of equations

$$\begin{cases} x_1 x_2 = n_1 \\ \bar{x_1} x_2 = n_2 \\ x_1 \bar{x_2} = n_3 \\ \bar{x_1}\bar{x_2} = n_4, \end{cases}$$

where $\{n_2, n_3, n_4\}$ vary over all permutations of $|C_\alpha(\mathbb{Z}_n)|$ for $\alpha > 1$. The number of all choices of triples (n_2, n_3, n_4) is therefore $\leq D^2(n)^3 = O(\log^6 n)$, but can be obviously reduced if we detect the distinct numbers among $|C_\alpha(\mathbb{Z}_n)|$.
Dividing Eqs. (4) by (2) and (3) by (1) we have $\bar{x_2}/x_2 = n_4/n_2 = n_3/n_1$. Thus the system has solutions iff $n_4/n_2 = n_3/n_1$. Writing $\bar{x_2} = \frac{n_4}{n_2} x_2$ we eliminate one equation yielding the system

$$\begin{cases} x_1 = n_1 x_2^{-1} \\ \bar{x_1} = n_2 x_2^{-1} \\ \bar{x_2} = \frac{n_4}{n_2} x_2. \end{cases}$$

Since $x_i + \bar{x_i} = 2p_i + 2$, we obtain that

$$\begin{cases} p_1 = \frac{1}{2}(n_1 + n_2) x_2^{-1} - 1 \\ p_2 = \frac{1}{2}(\frac{n_4}{n_2} + 1) x_2 - 1. \end{cases}$$

Substituting to the equation $p_1 p_2 = n$ and multiplying by x_2 we conclude that x_2 is a solution of the following quadratic equation

$$\left(\frac{1}{2}\left(\frac{n_4}{n_2} + 1\right) x_2 - 1\right)\left(\frac{1}{2}(n_1 + n_2) - x_2\right) = n x_2.$$

with the solution satisfying the Eq. (15). Having x_2 we can compute x_1 and $\bar{x_1}$ and therefore from the above system of two equations we see that

$$2x(p_1 + 1) = n_1 + n_2$$

hence $p_1 = \frac{n_1 + n_2}{2x} - 1$, as required in (16). The total running time is therefore $O(\log^6 n \log^3 n) = O(\log^9 n)$ bit operations. This completes the proof of estimate (1) of Sect. 2.

References

[AdCu94] Adleman, L.M., McCurley, K.S.: Open problems in number theoretic complexity, II. In: Adleman, L.M., Huang, M.-D. (eds.) ANTS 1994. LNCS, vol. 877, pp. 291–322. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-58691-1_70

[Bac84] Bach, E.: Discrete logarithms and factoring. Computer Science Division, University of California, Berkeley (1984)

[Dav67] Davenport, H.: Multiplicative Number Theory. Markham Publishing Company, Chicago (1967)

[DrPo17] Dryło, R.E., Pomykała, J.: Integer factoring problem and elliptic curves over the ring $\mathbb{Z}_n$ (submitted)

[DuPo17] Durnoga, K., Pomykała, J.: Large sieve, Miller-Rabin compositness witnesses and integer factoring problem. Fundam. Inf. **156**(2), 179–185 (2017)

[GPS02] Galbraith, S., Paulus, S., Smart, N.: Arithmetic on superelliptic curves. Math. Comput. **71**(237), 393–405 (2002)

[HaWr79] Hardy, G.H., Wright, E.M.: An Introduction to the Theory of Numbers, 5th edn. OxfordScience Publications/Clarendon Press, Oxford (1979)

[KuKo98] Kunihiro, N., Koyama, K.: Equivalence of counting the number of points on elliptic curve over the ring Zn and factoring n. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 47–58. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0054116

[KnPa76] Knuth, D.E., Trabb, L.: Analysis of a simple factorization algorithm. Theoret. Comput. Sci. **3**, 321–348 (1976)

[LaWu08] Lau, Y.K., Wu, J.: On the least quadratic non-residue. Int. J. Number Theory **04**, 423 (2008)

[Len87] Lenstra Jr., H.W.: Factoring integers with elliptic curves. Ann. Math. **126**, 649–673 (1987)

[LePo05] Lenstra Jr., H.W.C.: Pomerance, primality testing with Gaussian periods. http://www.ams.org/journals/mcom/2015-84-291/S0025-5718-2014-02840-8

[MMV01] Martin, S., Morillo, P., Villar, J.L.: Computing the order of points on an elliptic curve modulo N is as difficult as factoring N. Appl. Math. Lett. **14**(3), 341–346 (2001)

[OkUc98] Okamoto, T., Uchiyama, S.: Security of an identity-based cryptosystem and the related reductions. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 546–560. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0054153

Detection of Primes in the Set of Residues of Divisors of a Given Number

Rafał Bystrzycki(✉)

Department of Discrete Mathematics, Adam Mickiewicz University in Poznań,
Poznań, Poland
rafbys@amu.edu.pl

Abstract. We consider the following problem: given the set of residues modulo p of all divisors of some squarefree number n, can we find efficiently a small set of residues containing all residues coming from prime factors? We present an algorithm which solves this problem for p and n satisfying some natural conditions and show that there are plenty of such numbers. One interesting feature of the proof is that it relies on additive combinatorics. We also give some application of this result to algorithmic number theory.

1 Introduction

The main part of this paper deals with the following problem: Suppose that for some natural number n and some prime number p we are given the set of residues mod p of all its divisors and we would like to know which of those residues correspond to prime factors of n. For convenience we introduce the following notation:

Notation 1. *A would stand for the set of all divisors of n. A_p would stand for the set of residues mod p of elements of A. Similarly, Γ would stand for the set of prime factors of n and Γ_p would stand for the set of residues mod p of elements of Γ. Also, $\mathbb{Z}_p$ stands for $\mathbb{Z}/p\mathbb{Z}$.*

Ideally, we would like to find Γ_p, but we were unable to achieve that goal. Moreover, it seems to be impossible to get in general with an algorithm using only the information on residues mod p (see Sect. 7). Therefore, we focus on a simpler but still useful task of finding B, a small subset of A_p containing Γ_p. For our application (see Sect. 6) it turns out to be good enough. We firmly believe that more applications of this approach should be found in the future. In the sequel, we are going to provide two algorithms ($B_{rand}(A_p)$ and $B_{struct}(A_p)$) to find such a set B. For brevity, we will denote resulting sets obtained from A with those algorithms by B_{rand} and B_{struct} respectively.

Before we formulate our main theorem, let us provide some definitions which are essential to fully explain its meaning and the idea behind its proof. First, let us recall some basic number-theoretic functions. We will need them to express properties of numbers which make our argument to work.

J. Kaczorowski et al. (Eds.): NuTMiC 2017, LNCS 10737, pp. 178–194, 2018.
https://doi.org/10.1007/978-3-319-76620-1_11

Definition 1. *$\omega(n)$ denotes the number of prime divisors of a number n.*

Definition 2. *$P(m)$ denotes the greatest prime divisor of an integer m.*

Another important number-theoretic functions are $\sigma_k(n)$

Definition 3

$$\sigma_k(n) = \sum_{d|n} d^k.$$

The problem we look at arise naturally when studying the deterministic reduction of factorization to computing the values of $\sigma_k(n)$. We detail this application in Sect. 6.

We are going to present an algorithm which is deterministic, but works only for some inputs. We next show that for a randomly chosen input the algorithm is almost certain to work properly. To formalize this statement we will need the notion of natural density.

Definition 4. *Natural density of a set X of integers is the following limit (if it exists)*

$$\lim_{n\to\infty} \frac{\#\{m \in \mathbb{N} : m < n, m \in X\}}{n} \tag{1}$$

It turns out that the right way of looking at the problem we consider is actually by looking at numbers as elements of a cyclic group $\mathbb{Z}_p^*$. It leads us to consider the following object appearing in additive number theory.

Definition 5. *Let C be a subset of abelian group. Then $\mathcal{P}(C)$ denotes the set of all subset sums of C, namely*

$$\mathcal{P}(C) := \{\sum_{a\in T} a : T \subset C\}.$$

After taking logarithms of elements of the set of all divisors of a given number we get the structure defined above with C being the set of prime factors.

Now we are ready to state our result. The main theorem of this paper is

Theorem 1. *For a given x let $p = \log x^{3+o(1)}$ be a prime such that $p^{0.5-\epsilon} < P(p-1) < p^{0.5+\epsilon}$ and $P(p-1)^2 \nmid (p-1)$ and let $n \le x$ be a squarefree integer such that $\omega(n) \le 2\log\log n$, n has at most $\log\log x^{1+o(1)}$ divisors less than p, no pair of distinct divisors of n is congruent modulo p and the number of its divisors $d > p$ for which $d^{\frac{p-1}{P(p-1)}}$ is congruent to $q^{\frac{p-1}{P(p-1)}}$ or $-q^{\frac{p-1}{P(p-1)}}$ for some prime divisor q is less then $\frac{1}{2}\epsilon 2^{\omega(n)}$. Let A (and A_p) denote the set of divisors of n (and their residues modulo p) and let Γ (and Γ_p) denote the set of prime divisors of n (and their residues modulo p). Then for any $\epsilon > 0$ there exists a deterministic algorithm with running time $O_\epsilon(p^{0.5+o(1)}) = O((\log x)^{1.5+o(1)})$ which finds a set B such that $\Gamma_p \subset B \subset A_p$ and $|B| < \epsilon|A_p|$.*

Although the statement is a bit technical, we are going to show that all conditions appearing in the assumptions are very weak and in fact occur for almost every squarefree number n and for enough primes p in order to be practical. The most interesting novelty in the proof is the heavy use of additive combinatorics in a problem arising from multiplicative number theory. We also give an application of this result to the algorithm which finds factorization of a given number using an oracle for values of functions $\sigma_k(n)$. In fact, the search for deterministic reductions of factorization to some other number-theoretic problems was our original motivation to study this problem.

2 Preliminaries

Let us briefly recall some results from computational number theory, group theory and Fourier analysis. Reader may as well skip this part if he's familiar with those concepts. Concepts from additive combinatorics and analytic number theory are introduced in Sects. 3 and 4 respectively, where they are used.

Lemma 1. *Addition (or subtraction) of two numbers on at most n bits can be performed with $O(n)$ bit operations.*

Theorem 2 (Schönhage - Strassen [16]). *Multiplication of two numbers on at most k bits can be performed with $O(k \log k \log\log k)$ bit operations. In particular it is $O(k^{1+o(1)})$.*

Corollary 1. *Division (with the remainder) of the number N on at most k bits by the number D on at most k bits can be performed with $O(k(\log k)^2 \log\log k)$ bit operations (in particular it is $O(k^{1+o(1)})$).*

Lemma 2. *Values of a polynomial of degree k at a given point can be found with k multiplications and k additions using Horner scheme.*

Lemma 3. *Greatest common divisor of polynomials $f, g \in \mathbb{F}_p[X]$ can be found with Euclid algorithm with $O(deg(f)deg(g))$ operations in $\mathbb{F}_p$.*

Lemma 4. *Exponentiation modulo p to the exponent k can be performed with $O(\log k)$ operations in $\mathbb{F}_p$.*

We recall some basic facts about the structure of $\mathbb{Z}_p^*$. The previous lemma implies that the homomorphism mentioned below can be computed efficiently.

Lemma 5. *If $p-1 = q_1^{e_1} \cdots q_k^{e_k}$ is a prime powers factorization, then*

$$\mathbb{Z}_p^* \simeq \mathbb{Z}_{p-1} \simeq \mathbb{Z}_{q_1^{e_1}} \times \cdots \times \mathbb{Z}_{q_k^{e_k}}.$$

For every $q|(p-1)$

$$a \mapsto a^{\frac{p-1}{q}}$$

is a group homomorphism $\mathbb{Z}_{p-1} \to \mathbb{Z}_q$.

In order to work with the additive notation we will need to take discrete logarithms.

Definition 6. *Let $b, g \in \mathbb{F}_p$. Discrete logarithm of g to the base b is the residue class mod $ord(b)$ of the smallest positive integer k such that $b^k = g$. We denote it with $\log_b(g)$.*

Theorem 3 (Pollard [13]). *Discrete logarithm modulo p can be found with $O(\sqrt{p})$ operations in $\mathbb{F}_p$.*

Definition 7. *Discrete Fourier transform (with size p) of a function $f : \mathbb{F}_p \to \mathbb{C}$ is a function*

$$\hat{f}(\gamma) = \sum_{x \in \mathbb{F}_p} f(x) e^{\frac{2\pi i}{p} x\gamma}.$$

Theorem 4 (Bluestein [4]). *Discrete Fourier transform with size N can be computed with $O(N \log N)$ arithmetical operations.*

Discrete Fourier transform enjoys the following nice property.

Lemma 6 (Parseval identity)

$$\sum_{x \in \mathbb{F}_p} |f(x)|^2 = \frac{1}{p} \sum_{x \in \mathbb{F}_p} |\hat{f}(x)|^2.$$

This fact is particularly useful when applied to the characteristic function of the set $A \subset \mathbb{F}_p$.

Corollary 2

$$\sum_{x \in \mathbb{F}_p} |\hat{A}(x)|^2 = p|A|.$$

3 Algorithms

In this section we present an algorithm which solves the problem stated in the introduction, therefore proving Theorem 1. The algorithm consists of two algorithms, which performed one after another lead to the solution. They are based on two simple observations. We include them as the next two lemmas.

The idea behind the first one is to look for properties of prime numbers which distinguish them from the composite ones. To be more specific, we are interested in properties which are preserved after taking residues mod p. One such property is the large number of multiples in the set of divisors. Algorithm 1 is based on this lemma.

Lemma 7. *If $a \in \Gamma$, then there exist at least $2^{\omega(n)-1}$ elements $b \in A$ such that $a(mod p)b(mod p) \in A_p$.*

Proof. For every $b \in A$ which is not a multiple of a (for $a \in \Gamma$ there are $2^{\omega(n)-1}$ such b's) $ab \in A$ holds, hence also $ab (\mathrm{mod}\, p) \in A_p$.

When we apply Algorithm 1 to $A_p \subset \mathbb{Z}_p$ all elements of Γ_p are included in B_{rand}. But B_{rand} may be too big.

Algorithm 1. $B_{rand}(A_p)$

```
For every $a \in A_p$:
1. set $D_a = 0$
2. For every $b \in A_p$:
   (a) check whether $ab \in A_p$,
   (b) if it's true set $D_a = D_a + 1$.
3. if $D_a \geq \frac{1}{2}|A_p|$, add $a$ to the set $B_{rand}$.
```

The idea behind the second lemma is to realize that the problem is really about $\mathcal{P}(C)$ of some set C in the cyclic group $\mathbb{Z}_{p-1}$ and look for other settings where the corresponding problem is easy to solve. It turns out that one such setting is the semigroup of natural numbers under addition.

Lemma 8. *Let $C \subset \mathbb{N}$. Then there exists a deterministic algorithm which given $\mathcal{P}(C)$ ($|\mathcal{P}(C)| = N$) finds C with running time $O(N \log N)$. Moreover, C can be a multiset and it does not change the conclusion.*

Proof. See Algorithm 2.

Algorithm 2. *C(S)*

```
1. Sort the elements of $S$ in nondecreasing order.
2. Set $D := \emptyset$ and $C := \emptyset$.
3. Move $0$ from $S$ to $D$.
4. Until $|C| = \frac{\log(|S|)}{\log 2}$:
   (a) Set $x$ - the smallest element still in $S$.
   (b) For all elements $d$ in $D$ move $x + d$ from $S$ to $D$.
   (c) Add $x$ to $C$.
```

This algorithm can be easily adapted to handle also sets containing negative integers. It is going to be important that we can easily generalize this problem (and its solution) to multisets (no changes in the algorithm needed).

Algorithm 3. *F(S)*

```
1. Find $\min(S)$ and set $T = \{s - \min(S) : s \in S\}$.
2. Apply Algorithm 2 with $T$ as input to find $\bar{C} = C(T)$.
3. For every $c \in \bar{C}$:
   (a) if $c \in S$ - add $c$ to $F$.
   (b) if $-c \in S$ - add $-c$ to $F$.
```

Corollary 3. *Let $C \subset \mathbb{Z}$. Then there exists a deterministic algorithm which given $S = \mathcal{P}(C)$ ($|\mathcal{P}(C)| = N$) finds a set F such that $C \subset F$ and $|F| < 2|C|$ with running time $O(N \log N)$. Moreover, C can be a multiset and it does not change the conclusion (elements of multisets are counted with multiplicity).*

Proof. Algorithm 3 does the job, since the addition of the constant (which is an element of the input set S) only changes the signs of some elements $g \in C$. Absolute values of elements of C are found in step 2.

In order to adapt this algorithm to the setting of cyclic group it is desirable to contain the set in some short interval. To perform this task it is convenient to work with a group of prime order. Therefore, we would like to have at least a large subgroup of prime order. To find the sought-after interval efficiently, we need to use Fourier transform. In order to optimize its computational complexity we would not like this prime to be too large. This are the reasons for our assumptions on $P(p-1)$.

Algorithm 4. $B_{struct}(A_p)$

```
1. Set $q = P(p-1)$.
2. For every $a \in A_p$ compute $\bar{a} := a^{\frac{p-1}{q}}$.
3. For every $a \in A_p$ compute discrete logarithm $\tilde{a} := \frac{q}{p-1}\log_g(\bar{a})$ (for some generator $g$ of the group $\mathbb{Z}_p^*$). Set $L_q = \{\frac{q}{p-1}\log_g(\bar{a}) : a \in A\} \subset \mathbb{Z}_q$.
4. Find using Fourier transform $d \in \{1, \ldots, q-1\}$ such that for all $\tilde{a} \in L_q$ elements $d \cdot \tilde{a}$ are contained in the interval $[-\frac{q\log(2)}{\log(|A_p|)}, \frac{q\log(2)}{\log(|A_p|)}]$.
5. Find the set $F$ using Algorithm 3 for $\mathbb{Z}$ with $d \cdot L_q$ (with elements treated as integers) as an input.
6. For every $c \in F$ put all corresponding $a \in A_p$ into the set $B_{struct}$ (if $a|n$ as integers include $a$ only if it's prime).
```

Observe that if $d \in \mathbb{Z}_q$ is such that $dA \subset [-\frac{q\log(2)}{\log(|A_p|)}, \frac{q\log(2)}{\log(|A_p|)}]$, then it corresponds to a large Fourier coefficient, namely $\hat{A}(d)$ is greater than $\frac{|A|}{2}$ (say) if x is large enough. Hence in step 5 of Algorithm 4 we first find all Fourier coefficients larger than $\frac{|A|}{2}$. There are at most $\frac{p}{|A|}$ of them because of Parseval identity. Then we can check for all of them whether they satisfy the condition.

Now the analysis of computational complexity of those algorithms is straightforward. First algorithm needs only $O(|A|^2)$ operations in $\mathbb{F}_p$. The most costly step of the second algorithm is step 4, which takes $O(p^{\frac{1}{2}+o(1)})$ operations in $\mathbb{F}_p$. Step 3 takes $O(p^{\frac{1}{4}+o(1)}|A|)$ operations.

To find all divisors which can possibly be prime we need to perform those two algorithms. At least one of them should give us desired set. Justification of this statement finishes the proof of Theorem 1 and it is our main objective in the next section.

4 If B_{rand} Fails, then B_{struct} Works

In this section we present the heart of our proof. This is the part where additive combinatorics come into play. For theoretical consideration it is simpler to look at the set of discrete logarithms of elements of the set A_p. We will denote this set by L.

Notation 2. *Let* $L := \{\log_g(a) : a \in A_p\}$.

Note that to optimize computational complexity of Algorithm 4, we perform exponentiation first and then take discrete logarithms. Exposition becomes clearer with those operations in reversed order, since then we can phrase structural properties of A_p in additive language. Later we work with corresponding subset of integers under addition what makes additive notation more natural here.

Note that definitions and theorems in this section use L to denote general subset of an abelian group. One can think of L defined above as an illustrative example (we will only be concerned with L and related sets: its subsets and homomorphic images).

First, let us define some notion encoding additive structure of a set.

Definition 8. *Let G be an abelian group and $L \subset G$ a finite subset. The energy of L is defined by*

$$E(L) = \frac{1}{|L|^3}|\{(a_1, a_2, a_3, a_4) \in L^4 : a_1 - a_2 = a_3 - a_4\}|.$$

We can think of a set with large additive energy as structural.

The next lemma shows that Algorithm 1 can only fail for A_p, such that L, the set of discrete logarithms of its elements, is additively structured.

Lemma 9 (Katz-Koester [8]). *Let $0 < \rho < 1$ and suppose L_1 and L_2 are two subsets of G, and suppose*

$$L_1 \subset \{z \in G : |(z + L_2) \cap L_2| \geq \rho |L_2|\}.$$

Then

$$E(L_1)E(L_2) \geq \frac{(\log(2))^2}{16} \frac{\rho^4}{(\log(\frac{4}{\rho^2}))^2} \frac{|L_1|}{|L_2|}.$$

Applying this lemma with $L_1 = \{\log_g(b) : b \in B_{rand}\}$ (recall that B_{rand} is the output of Algorithm 1), $L_2 = L$ and $\rho = \frac{1}{2}$ we obtain the bound for the additive energy of L or its large subset L_1. Namely, at least one of those sets satisfies

$$E(L_i) \geq \kappa\sqrt{\epsilon}$$

for some explicit constant κ. In each case there is at least some large subset $L' \subset L$ (namely $|L'| > c(\epsilon)|L|$) with $E(L) \geq \frac{1}{K(\epsilon)}$.

It is more convenient to use some more restrictive notion of additive structure and work with sets satisfying the condition $|L + L| \leq K|L|$ or $|L - L| \leq K|L|$ (look at the definition below) for some constant K (the so called sets with small doubling).

Definition 9. *The sumset of a set L is the set*

$$L + L := \{a + b : a, b \in L\}$$

The difference set of a set L is the set

$$L - L := \{a - b : a, b \in L\}$$

Before we move on, let us formulate some classical results from additive combinatorics which are very useful in the study of sets with small doubling.

Lemma 10 (Ruzsa covering lemma [14]). *For any non-empty sets L, M in abelian group G one can cover M by $\frac{|L+M|}{|L|}$ translates of $L - L$.*

Lemma 11 (Plünnecke inequality [12]). *If $|L + L| \leq K|L|$ or $|L - L| \leq K|L|$, then $|iL - jL| \leq K^{i+j}|L|$ for all non-negative integers i, j.*

The two notions of additive structure are not exactly equivalent, but some sort of equivalence between them is given by the following theorem. The first version of this theorem (with exponential dependence on K) was given by Balog and Szemerédi. First version with polynomial dependence on K was provided by Gowers. We quote the version with currently the best known dependence on K.

Theorem 5 (Balog-Szemerédi-Gowers [15]). *Let be a subset of an abelian group such that $E(L) = \frac{1}{K}$. Then there exists $L' \subset L$ such that $|L'| = \Omega(\frac{1}{K}|L|)$ and*

$$|L' - L'| = O(K^4|L'|).$$

Using this theorem we can find some large more structural subset in our original set, namely the set $L'' \subset L'$ such that $|L''| > c(\epsilon)|L'|$ and $|L'' - L''| < K(\epsilon)|L''|$.

We need even more restrictive notion of additive structure. Its main advantage for us is that it is well-behaved under homomorphisms.

Definition 10. *Let $K \geq 1$. A subset H of an abelian group G is said to be a K-approximate group if it is symmetric ($H = -H$), contains neutral element, and $H + H$ can be covered by at most K traslates of H.*

Lemma 12. *Let L be a subset of an abelian group G containing neutral element. If $|L + L| \leq K|L|$, then there exists a K^3-approximate group H such that $L \subset H$ and $|L| \geq K^{-2}|H|$.*

Proof. Take $H = L - L$ and apply Ruzsa covering lemma (with $L = L$ and $M = H$). The result follows, since $|L| < |L - L| < K^2|L|$ and $|L - L + L| < K^3|L|$ by Plünnecke inequality.

Applying this lemma to the set L'' we obtain a set H, such that $|H| < C(\epsilon)|L''|$, $L'' \subset H$ and H is $K(\epsilon)$-approximate group.

The following lemma appears as an exercise in [18].

Lemma 13. *Let G, G' be abelian groups, $H \subset G$ a K-approximate group and $\phi : G \to G'$ - a homomorphism. Then $\phi(H)$ is a K-approximate group.*

Proof. Let $x_1, \dots, x_K \in G$ be such that $H + H$ is covered by $x_1 + H, \dots, x_k + H$. Then $\phi(x_1) + \phi(H), \dots, \phi(x_K) + \phi(H)$ covers $\phi(H) + \phi(H)$. Clearly, $\phi(e_G) = \phi(e_{G'})$ and $\phi(-a) = -\phi(a)$.

Applying the last lemma to the set H and a homomorphism $\phi : \mathbb{Z}_p^* \to \mathbb{Z}_q$ defined by $a \mapsto \log_g(a^{\frac{p-1}{q}})$, we see that $\phi(H)$ is $K(\epsilon)$-approximate group.

Now, we have got an additively structured set in a large group of prime order. In such a setting we can observe that this set can be compressed to a short interval.

Definition 11 *The diameter diam$L$ of a set L (in $\mathbb{Z}$ or $\mathbb{Z}_m$) is defined as the smallest integer l for which there exist some a, d such that $L \subset a, a+d, \dots, a+ld$.*

Theorem 6 (Green-Ruzsa [7]). *Let q be a prime and let $H \subset \mathbb{Z}_q$ be a set with $|H| = \alpha q$ and $|2H| = K|H|$. Suppose that $\alpha \leq (16K)^{-12K^2}$. Then the diameter of H is at most*

$$12\alpha^{\frac{1}{4K^2}}\sqrt{\log{(\frac{1}{\alpha})}q}.$$

We emphasize the fact that small doubling is really needed here (large additive energy is not enough). Obviously, K-approximate group satisfies $|H + H| \leq K|H|$. Using this theorem, we can therefore find an arithmetic progression P such that $|P| \leq p^{1-\delta(\epsilon)}$ for some $\delta(\epsilon) > 0$ and H (and hence also L'') is contained in P.

Next lemma will bring us back to the set L (or more precisely $\phi(L)$, which is equal to the set L_q in step 3 of Algorithm 4). Roughly speaking, it shows that a structure of $\mathcal{P}(C)$ enables us to control the whole set, when only some part is controlled. The fact that $L = \mathcal{P}(C)$ is crucial here and it is the only part of the proof where we use it.

Lemma 14. *Let $L = \mathcal{P}(C)$ be a subset (L is possibly a multiset) of $\mathbb{Z}_q$ and let $L' \subset L$ be such that $|L'| \geq \epsilon|L|$ (elements counted with multiplicity) and diam$L' \leq q^{1-\delta}$. Then there exists a constant $K(\epsilon) > 0$ such that L is contained in $K(\epsilon)$ translates of a set D with diam$D \leq 2q^{1-\delta}$.*

Proof. Let P be a symmetric arithmetic progression such that some translate x of P contains L' (without loss of generality we can assume that P has the common difference 1, otherwise we can multiply every element by d^{-1}). We are going to construct $m = \lceil \frac{2}{\epsilon} \rceil$ translates $x_i + 2P$ such that $C \subset X + 2P$ for $X = \{x_1, \dots, x_m\}$. For each $g_j \in C$ either g_j belongs to some $x_i + P$ (and then $g_j + L' \subset x_i + x + 2P$ and $L' - g_j \subset x_i + x + 2P$) for some x_i already put in X or there are $|L'| = \epsilon|L|$ elements of L which are of the form $g_j + a'$ or $a' - g_j$ and are not captured by any translate yet. Then we add g_j and $-g_j$ to the set X. We need to add new translates at most $\lceil \frac{1}{\epsilon} \rceil$ times, because it increases by $\epsilon|A|$ the number of elements of A covered. If X is a set of translates covering all $ginC$, then $\mathcal{P}(X)$ are translates covering $\mathcal{P}(C)$ (and there are $2^{|X|}$ of them).

Lemma 15. *Let $L \subset \mathbb{Z}_q$ be a set with diam$A = q^{1-\delta}$. Then there exist $d \in \mathbb{Z}_q^*$ such that $dL \subset [-2q^{1-\frac{\delta}{2}}, 2q^{1-\frac{\delta}{2}}]$. Generally, if L is contained in K translates of a set D with diam$D = q^{1-\delta}$, then there exists $d \in \mathbb{Z}_q^*$ such that $dL \subset [-2q^{1-\frac{\delta}{2K}}, 2q^{1-\frac{\delta}{2K}}]$.*

Proof. Let $a \in L$ be any element. By Pigeonhole Principle, there exist $d < q^{\frac{\delta}{2}}$ such that $da \in [q^{1-\frac{\delta}{2}}, q^{1-\frac{\delta}{2}}]$ (there exist two elements $d_1 a, d_2 a$ in one interval of length $q^{1-\frac{\delta}{2}}$, their difference satisfies the condition). For such d the conclusion holds. To prove the second statement, use multidimensional Pigeonhole Principle to find $d < q^{\frac{\delta}{2}}$ such that $da_i \in [q^{1-\frac{\delta}{2K}}, q^{1-\frac{\delta}{2K}}]$ for $i = 0, \ldots, K-1$, where $a_i + D$ are given translates.

Using the last two lemmas we see that we can find d such that $dL \subset [-2q^{1-\frac{\delta(\epsilon)}{2K(\epsilon)}}, 2q^{1-\frac{\delta(\epsilon)}{2K(\epsilon)}}]$ what proves that a suitable d in step 4 of Algorithm 4 can be found, since $p^{\frac{\delta(\epsilon)}{2K(\epsilon)}} > \frac{\log(|A_p|)}{\log(2)}$ if n is large enough. It finishes the proof of Theorem 1, since the number of elements $a \in A_p$ corresponding to the same $c \in F$ is small by our assumptions (specifically the last, more technical one).

5 There are Plenty of Numbers Satisfying the Conditions

First of all, observe that the fact that for all but $o(x)$ numbers $n \leq x$ the number of prime divisors is right follows from the classical result quoted below.

Theorem 7 (Erdős-Kac [6]). *Denote by $N(x; a, b)$ the number of integers m belonging to the interval $[3, x]$ for which the following inequality holds:*

$$a \leq \frac{\Omega(m) - \log\log m}{\sqrt{\log\log m}} \leq b, \tag{2}$$

where $a < b$ are real numbers with additional possibilities $a = -\infty$ and $b = \infty$. Then, with x tending to infinity, we have

$$\lim_{x\to\infty} \frac{N(x; a, b)}{x} = \frac{1}{\sqrt{2\pi}} \int_a^b \exp\left(-\frac{t^2}{2}\right) dt. \tag{3}$$

It is easy to observe that a typical number cannot have to many small divisors. We will need this fact later.

Lemma 16. *There are $o(x)$ numbers $n \leq x$ such that the number of divisors of n smaller than $(\log x)^{3+o(1)}$ is greater than $(\log\log x)^{1+o(1)}$.*

Proof. It follows from the fact that

$$\sum_{n<(\log x)^3} \frac{x}{n} = O(x \log x).$$

It is possible to find for x large enough the prime with the desired properties of $p-1$. To prove that we need two classical results from analytic number theory.

Lemma 17 (Mertens [9]). *We have*

$$|\sum_{p\leq n}\frac{\log p}{p}-\log n|\leq 2.$$

Theorem 8 (Bombieri - Vinogradov [19]). *Let x and Q be any two positive real numbers with $x^{1/2}\log^{-A}x\leq Q\leq x^{1/2}$. Then*

$$\sum_{q\leq Q}\max_{y<x}\max_{\substack{1\leq a\leq q\\(a,q)=1}}\left|\psi(y;q,a)-\frac{y}{\varphi(q)}\right|=O\left(x^{1/2}Q(\log x)^5\right).$$

This leads to the following statement.

Corollary 4. *Let $\epsilon>0$. Then there exist efficiently computable constants $X_1(\epsilon)$, $\delta(\epsilon)>0$, such that, if $x>X_1$, we have*

$$\sum_{p\leq x, x^{\frac{1}{2}-\epsilon}<P(p-1)<x^{\frac{1}{2}+\epsilon}}1>\delta(\theta)\frac{x}{\log x}.$$

Proof. It suffices to lowerbound the sum $\sum_{x^{\frac{1}{2}-\epsilon}<p<x^{\frac{1}{2}}(\log x)^{-B}}\pi(x,q)$. By Bombieri-Vingradov theorem

$$\sum_{x^{\frac{1}{2}-\epsilon}<p<x^{\frac{1}{2}}(\log x)^{-B}}\pi(x,q)\log q=\frac{x}{\log x}\sum_{x^{\frac{1}{2}-\epsilon}<p<x^{\frac{1}{2}}(\log x)^{-B}}\frac{\log p}{p-1}+O(\frac{x}{\log x}).$$

The last sum is equal $\epsilon\log x+O(1)$ by Mertens' theorem.

To ensure that $P(p-1)^2\nmid(p-1)$ we need the following lemma.

Lemma 18. *There are $O(\frac{x^{\frac{1}{2}+2\epsilon}}{\log x})$ numbers $n\leq x$ such that $q^2|(n-1)$ for some prime number $q>x^{\frac{1}{2}-\epsilon}$. In particular, for $\epsilon<\frac{1}{4}$ there are $o(\frac{x}{\log x})$ such prime numbers.*

Proof. We simply count

$$\sum_{x^{\frac{1}{2}-\epsilon}<q<x^{\frac{1}{2}}}\frac{x}{q}=O(\frac{x^{\frac{1}{2}}}{\log x}x^{2\epsilon}),\tag{4}$$

since there are $O(\frac{x^{\frac{1}{2}}}{\log x})$ primes in this range and at most $\frac{x}{x^{2(\frac{1}{2}-\epsilon)}}=x^{2\epsilon}$ numbers divisible by any of them.

Now, we will prove that given such a prime p we can expect different divisors of n to give different residues. In the proof we are going to use the following lemma which is a discrete analogue of integration by parts (Lemma 2.5.1 in [2]).

Lemma 19. *Let $(a_n)_{n\in\mathbb{N}}$ be the sequence of complex numbers, $A(t) := \sum_{n\leq t} a_n$ and let $f : [1, x] \to \mathbb{C}$ be a C^1-class function. Then:*

$$\sum_{n\leq x} a_n f(n) = A(x)f(n) - \int_1^x A(t)f'(t)dt. \tag{5}$$

Lemma 20. *Let $\epsilon > 0$. For a given prime number such that $p > (\log x)^{2+\epsilon}$ the set of numbers $n \leq x$ such that there exists a pair of distinct divisors of n congruent modulo p respectively has size $o(x)$.*

Proof. Clearly, there are $o(x)$ number $n < x$ divisible by p. We need to bound the size of the set of numbers n such that there exist a pair d_1, d_2 such that $d_1|n$, $d_2|n$ and $d_1 - d_2$ is divisible by p. For n not divisible by p at least one such pair d_1, d_2 (if it exists) must consist of relatively prime numbers. Therefore, the size of the set can be crudely bounded by the following expression

$$\sum_{r<\frac{x}{p}} \sum_{d<\frac{x}{rp}} \frac{x}{d(d+rp)} \tag{6}$$

To bound those sums we can use the following bound for the series $\sum_{n\geq 1} \frac{1}{n(n+r)}$ with parameter r.

$$\sum_{n\geq 1} \frac{1}{n(n+r)} = \sum_{n\geq 1} \frac{1}{r}(\frac{1}{n} - \frac{1}{n+r}) = \frac{1}{r}\sum_{n=1}^{r} \frac{1}{n} = O(\frac{\log r}{r}) \tag{7}$$

Using (7) with parameter rp we can bound (6) by

$$\sum_{r<\frac{x}{p}} \frac{x(\log rp)}{rp} = O(\frac{x(\log x)^2}{p}),$$

using Lemma 19 to get the last inequality.

What has left to show is that a condition set on $d^{\frac{p-1}{P(p-1)}}$'s is satisfied by typical n. First we deal with possible obstruction caused by a divisor which satisfies $d^{\frac{p-1}{P(p-1)}} \equiv \pm 1$.

Lemma 21. *Let p be a prime number and let $I \subset \mathbb{Z}_p^*$ be such that $|I| \leq p^\delta$. If $\log x = o(p^{1-\delta})$, then the set of numbers $n < x$ such that there exists a number $d > p$ which satisfies $d \equiv a$ for some $a \in I$ and $d|n$ has size $o(x)$.*

Proof. It follows from

$$\sum_{1\leq r\frac{x}{p}} \frac{x}{a+pr} = O(\frac{x}{p}\log(\frac{x}{p})).$$

Using this fact we can prove what we need.

Lemma 22. *Let $\epsilon > 0$. Let p be a prime number with $p \geq (\log x)^3$ such that $P(p-1) > (\log x)^{2-\log 2+\epsilon}$. For all but $o(x)$ numbers $n \leq x$ the set of divisors d of n such that $d^{\frac{p-1}{P(p-1)}} \equiv q^{\frac{p-1}{P(p-1)}}$ (mod p) for some $q > p$ prime divisor of n has size $o((\log x)^{\log 2})$.*

Proof. We can estimate the number of triples consisting of a number $n \leq x$ and a pair (d_1, d_2) of relatively prime divisors of n such that $d_1 > p, d_2 > p$ which satisfies $d_1^{\frac{p-1}{P(p-1)}} \equiv d_2^{\frac{p-1}{P(p-1)}}$. Let $I \subset \mathbb{Z}_p^*$ be a subgroup of $P(p-1)$-th powers and let $I_d \subset \mathbb{Z}_p^*$ be a coset of this subgroup containing d. We know that $|I| \leq \frac{p-1}{P(p-1)}$.

$$\sum_{d \leq x} \frac{1}{d} \sum_{a \in I_d} \sum_{1 \leq r \leq \frac{x}{p}} \frac{x}{a+rp} = O(\frac{x\frac{p}{P(p-1)}(\log x)^2}{p}) = O(\frac{x(\log x)^2}{P(p-1)}).$$

All divisors d for which $d^{\frac{p-1}{P(p-1)}} \equiv \pm q^{\frac{p-1}{P(p-1)}} \pmod{p}$ holds for some $q > p$ which is a prime divisor of n are either relatively prime to q (first kind) or they are of the form ds, where $s < p$ and d is either q or a divisor of the first kind (then we call them divisors of the second kind). The number of the divisors of the first type can be bounded by $O((\log x)^{\log 2-\epsilon})$ for all but at most $o(x)$ numbers $n \leq x$ using (5). Taking into account the divisors of the second kind raises this number only $(\log\log x)^{1+o(1)}$ times for all but $o(x)$ numbers $n \leq x$ by Lemma 16.

6 Application

Here we present an application of our result to deterministic polynomial-time reduction of factorization to computing $\sigma_1(n), \ldots, \sigma_M(n)$. This reduction is only proved to work for numbers forming a dense set (not necessarily for all numbers). The reduction is already polynomial-time in its simplest form. If a sufficiently efficient polynomial factoring algorithm is used (namely Shoup's Algorithm for polynomial with linear factors) it can be made to run in time $O((\tau(n))^2 \log n \log\log n \log\log\log n)$. Then our main result only reduces implied constant in $O()$ notation.

It is worth noting here that probabilistic polynomial-time reductions to computing $\sigma_k(n)$ (for a single k) are known [3]. Much more is known about the similar problem concerning Euler totient function $\phi(n)$. There exists a probabilistic polynomial-time reduction which can be easily derandomized under Extended Riemann Hypotheses [11]. Moreover, it can be shown unconditionally to work in deterministic polynomial time for the dense set of integers [5]. There is also unconditional subexponential-time reduction proved to work for any integer [20]. Paper [1] provides extensive survey of problems studied and results obtained in this area.

Algorithm 5. *$N(n, P_1, P_2, \ldots, P_M)$*

```
1. For every $k = 1, \ldots, M$ compute $S_k = \frac{(-1)^{k+1}}{k}(P_k + \sum_{i=1}^{k-1}(-1)^i P_{k-i} S_i)$.
2. Set as $m$ the greatest $k$ such that $S_k \neq 0$.
3. Set as $W \in \mathbb{Z}[X]$ the polynomial $W(X) = X^m + \sum_{i=1}^{m}(-1)^i S_i X^{m-i}$.
4. Factor the polynomial $W(X)$ in $\mathbb{Z}[X]$.
5. If the result consists of linear terms $(X - d_i)$ (for $i = 1, \ldots, m$), sort $d_i$ in
   nonincreasing order.
6. For each $i$ check whether $d_j | d_i$ for some $j < i$; if not, check with what multi-
   plicity $d_i$ divides $n$ and write out $d_i$ with that multiplicity.
```

Theorem 7 implies that in Algorithm 5 parameter $M = \lfloor (\log n)^{\log 2 + o(1)} \rfloor$ can be used and the algorithm would still work for the numbers from the set of natural density equal 1.

We prove

Theorem 9. *There exists a deterministic algorithm which using an oracle for monic polynomial W with all divisors of a given number m as roots computes the factorization of n for numbers n belonging to the set of natural density 1 (it uses the oracle at most twice) with running time $O((\tau(n))^2 \log n \log\log n \log\log\log n)$. In particular, for n belonging to this set this time is $(\log n)^{1+2\log 2+o(1)}$.*

We can assume that n is squarefree because of the following observation.

Lemma 23. *The set of natural numbers $n \leq x$ divisible by a square of an integer larger than $\log\log x$ is of cardinality $o(x)$.*

Proof. The cardinality of the considered set can be upperbounded by

$$x \sum_{\log\log x \leq d < \sqrt{x}} \frac{1}{d^2} + O(\sqrt{x}) \tag{8}$$

(as $\lfloor \frac{x}{d^2} \rfloor = \frac{x}{d^2} + O(1)$) which is $o(x)$ because of the convergence of the series $\sum \frac{1}{d^2}$.

Divisibility by squares of the numbers smaller than $\log\log n$ can be checked by trial division with $(\log n)^{1+o(1)}$ bit operations. If $p^\alpha || n$ the values of functions $\sigma_k(\frac{n}{p^\alpha})$ can be determined using formula $\sigma_k(\frac{n}{p^\alpha}) = \frac{\sigma_k(n)}{\sigma_k(p^\alpha)}$ at the cost of $O(k \log n)$ bit operations.

All divisors which can possibly be prime numbers can be found with Algorithm 6. To find the factorization of n perform the last step of Algorithm 5 on elements of B.

Algorithm 6. *$S(W)$*

```
1. Find a prime number $p$ of the order $(\log n)^{3+o(1)}$ with $P(p-1) = p^{0.5+o(1)}$.
2. Factor $W_p$ with Shoup algorithm and find set of residues $A$.
3. Find the set $B$ with Algorithm 1.
```

4. If $|B| > \epsilon|A|$, find the set B with Algorithm 4.
5. For every element in B perform Hensel lift to the residue modulo p^e (with $e = \lceil \frac{\log n}{\log p} \rceil$).

We need to define some special types of symmetric polynomials.

Definition 12. *Elementary k-th symmetric polynomial of variables $x_1, \ldots, x_m$ is given by*

$$s_k(x_1, \ldots, x_m) = \sum_{1 \le i_1 < \ldots < i_k \le m} x_{i_1} \cdots x_{i_k}. \tag{9}$$

k-th Newton function of variables $x_1, \ldots, x_m$ is given by

$$p_k(x_1, \ldots, x_m) = \sum_{i=1}^{m} x_i^k. \tag{10}$$

Function $\sigma_k(n)$ is equal to $p_k(d_1, \ldots, d_{\tau(n)})$, where $d_1, \ldots, d_{\tau(n)}$ are all divisors of n.

The correctness of the algorithm follows from the two sets of identities given below.

Lemma 24 (Newton identities). *For $1 \le k \le m$ the following identity holds:*

$$p_k + \sum_{i=1}^{k-1} (-1)^i p_{k-i} s_i + (-1)^k k s_k = 0, \tag{11}$$

and for $m < k$:

$$p_k + \sum_{i=1}^{m} (-1)^i p_{k-i} s_i = 0. \tag{12}$$

For a nice proof see [10].

Lemma 25 (Vieta's formulas). *Let R be an unique factorization domain and let $a_m x^m + \ldots + a_0 \in R[X]$ be a polynomial with m roots $x_1, \ldots, x_m$ (in the field of fractions of R). Then the following holds*

$$s_k(x_1, \ldots, x_m) = (-1)^k \frac{a_{m-k}}{a_m}. \tag{13}$$

To bound its running time we need the following two results from algorithmic number theory.

Theorem 10 (Shoup [17]). *Let f be a polynomial over $\mathbb{Z}_p$ of degree m which is a product of m distinct monic polynomials of degree 1. Then f can be factored deterministically with $O(p^{\frac{1}{2}}(\log p)^2 m^{1+o(1)})$ operations in $\mathbb{Z}_p$.*

Lemma 26 [2]. *Hensel lift of a root of polynomial f modulo p to a root modulo p^k can be found with $O(deg(f)(k \log p)^{1+o(1)})$ operations.*

Factorization can be found with Algorithm 6. Computing the coefficients of the polynomial modulo p can be performed in time $O((\tau(n))^2 \log n \log\log n \log\log\log n)$. Factorization of a polynomial with distinct roots over $\mathbb{F}_p$ can be done with Shoup algorithm in time $(\log n)^{\frac{1}{2}3+\log 2+o(1)}$. Algorithms 1 and 4 work in time $(\log n)^{2\log 2+o(1)}$ and $O((\log n)^{1.08+\log 2+o(1)})$ respectively. Hensel lift can be performed in time $o((\tau(n))^2 \log n \log\log n \log\log\log n)$. In this last bound we used our main result to reduce the number of Hensel lifts needed so that their cost does not dominate computational complexity of the algorithm.

From this result we can deduce the following.

Corollary 5. *There exists a deterministic algorithm which for almost every n if the values of functions $\sigma_1(n), \dots, \sigma_{\lfloor (\log n)^{\log 2+o(1)} \rfloor}(n)$ are given, computes the full factorization of n in time $O((\log n)^{1+2\log 2+o(1)})$.*

Proof. Values of $\sigma_k(\frac{n}{p^\alpha})$ can be computed effectively. After computing the residues of $\sigma_k(n)$ modulo $p^{\lceil \frac{\log n}{\log p} \rceil}$ coefficients of the polynomial can be found in time $O((\log n)^{1+2\log 2+o(1)})$. The rest proceeds exactly as in the previous proof.

The approach presented here does not seem to extend to the cases of a single $\sigma_k(n)$ or $\phi(n)$ mentioned in the beginning of this section, neither is it possible to work for any integer as it critically relies on n having the right number of prime factors. On the other hand, it does appear to be possible to significantly reduce the amount of information used by algorithm. It is not needed to know residues of all divisors, knowing a large fraction of them should suffice.

7 Open Problems

The problem considered here leads to the following questions: For a dissociated set C (a dissociated set is a set with all subset sums distinct) in an abelian group G, is C determined uniquely by $S = \mathcal{P}(C)$? Can we find it efficiently?

In general, already the answer to the first question is negative, as the examples below shows.

$$\mathcal{P}(\{2,5\}) = \mathcal{P}(\{5,7\}) \qquad \text{in } \mathbb{Z}_{10}$$

$$\mathcal{P}(\{3,5,6,7\}) = \mathcal{P}(\{1,9,13,15\}) = \mathbb{Z}_{17} \setminus \{2\} \qquad \text{in } \mathbb{Z}_{17}$$

The first example illustrates the obstruction caused by even order of the group and in the second one the set $\mathcal{P}(C)$ almost covers the whole group.

So, probably the right question to ask would be rather: Under what conditions is C determined uniquely by $S = \mathcal{P}(C)$? (Under what conditions can we find it efficiently?)

Acknowledgement. I would like to thank Doctor Bartosz Źrałek for introducing me to the topic of deterministic reductions of integer factorization to computing values of number-theoretic functions and encouraging me to work further on the problem considered here. I would like to thank Professor Tomasz Schoen for fruitful discussions on additive combinatorics and its connection to the problem. I would also like to thank anonymous referee for suggesting a simplification of the proof of Lemma 20 which led to a better estimate and the other anonymous referee for suggestions concerning the notation, which increased the readability of the paper.

References

1. Adleman, L.M., McCurley, K.S.: Open problems in number theoretic complexity, II. In: Adleman, L.M., Huang, M.-D. (eds.) ANTS 1994. LNCS, vol. 877, pp. 291–322. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-58691-1_70
2. Bach, E., Shallit, J.: Algorithmic Number Theory. MIT Press, Cambridge (1996)
3. Bach, E., Miller, G., Shallit, J.: Sums of divisors, perfect numbers and factoring. SIAM J. Comput. **15**(4), 1143–1154 (1986)
4. Bluestein, L.I.: A linear filtering approach to the computation of the discrete Fourier transform. Northeast Electron. Res. Eng. Meet. Rec. **10**, 218–219 (1968)
5. Burthe Jr., R.J.: The average least witness is 2. Acta Arithmetica **80**, 327–341 (1997)
6. Erdős, P., Kac, M.: The Gaussian law of errors in the theory of additive number theoretic functions. Am. J. Math. **62**, 738–742 (1940)
7. Green, B.J., Ruzsa, I.Z.: Sets with small sumsets and rectification. Bull. London Math. Soc. **38**(1), 43–52 (2006)
8. Katz, N.H., Koester, P.: On additive doubling and energy. SIAM J. Discrete Math. **24**(4), 1684–1693 (2010)
9. Mertens, F.: Ein Beitrag zur analytischen Zahlentheorie. J. Reine Angew. Math. **78**, 46–62 (1874)
10. Mead, D.G.: Newton's identities. Am. Math. Mon. **99**(8), 749–751 (1992)
11. Miller, G.: Riemann's hypothesis and tests for primality. J. Comput. Syst. Sci. **13**, 300–317 (1976)
12. Plünnecke, H.: Eine zahlentheorische anwendung der graphtheorie. J. Reine Angew. Math. **243**, 171–183 (1970)
13. Pollard, J.M.: Monte Carlo methods for index computation (mod p). Math. Comput. **32**(143), 918–924 (1978)
14. Ruzsa, I.: Sumsets and structure. In: Combinatorial Number Theory and Additive Group Theory, pp. 87–210 (2009)
15. Schoen, T.: New bounds in Balog-Szemer'edi-Gowers theorem. Combinatorica **34**(5), 1–7 (2014)
16. Schönhage, A., Strassen, V.: Schnelle Multiplikation großer Zahlen. Computing **7**, 281–292 (1971)
17. Shoup, V.: On the deterministic complexity of factoring polynomials over finite fields. Inf. Process. Lett. **33**, 261–267 (1990)
18. Tao, T.C., Vu, H.V.: Additive Combinatorics. Cambridge Studies in Advanced Mathematics, vol. 105. Cambridge University Press, Cambridge (2006)
19. Vinogradov, A.I.: The density hypothesis for Dirichlet L-series. Izv. Akad. Nauk SSSR Ser. Mat. **29**(4), 903–934 (1965). (in Russian)
20. Źrałek, B.: A deterministic version of Pollard's p-1 algorithm. Math. Comput. **79**, 513–533 (2010)

Pseudorandomness

The Measures of Pseudorandomness and the NIST Tests

László Mérai[1], Joël Rivat[2], and András Sárközy[3](✉)

[1] Johann Radon Institute for Computational and Applied Mathematics,
Austrian Academy of Sciences, Altenbergerstr. 69, 4040 Linz, Austria
merai@cs.elte.hu

[2] Institut de Mathématiques de Marseille UMR 7373, Université d'Aix-Marseille,
163 avenue de Luminy, 13288 Marseille Cedex 9, France
joel.rivat@univ-amu.fr

[3] Department of Algebra and Number Theory, Eötvös Loránd University,
Pázmány Péter sétány 1/c, Budapest 1117, Hungary
sarkozy@cs.elte.hu

Abstract. A few years ago new quantitative measures of pseudorandomness of binary sequences have been introduced. Since that these measures have been studied in many papers and many constructions have been given along these lines. In this paper the connection between the new measures and the NIST tests is analyzed. It is shown that finite binary sequences possessing strong pseudorandom properties in terms of these new measures usually also pass or nearly pass most of the NIST tests.

Keywords: Pseudorandom sequences · Binary sequence · NIST tests

2010 Mathematics Subject Classification: 11K45

1 Introduction

The National Institute of Standards and Technology (=NIST) of the US issued the document [26] which we refer to as "NIST tests". We quote the introduction of this documentum: "The need for random and pseudorandom numbers arises in many cryptographic applications. For example, common cryptosystem..." [e.g., the Vernam cipher] "...employs keys that must be generated in a random fashion... This document discusses the randomness testing of random numbers and pseudorandom number generators that may be used for many purposes including cryptographic, modeling and simulation applications. The focus of this document is on those applications where randomness is required for cryptographic

Research partially supported by the Hungarian National Foundation for Scientific Research, grants K119528 and NK104183, by the Austrian Science Fund FWF Project F5511-N26 which is part of the Special Research Program "Quasi-Monte Carlo Methods: Theory and Applications" and by the ANR Project MuDeRa.

J. Kaczorowski et al. (Eds.): NuTMiC 2017, LNCS 10737, pp. 197–216, 2018.
https://doi.org/10.1007/978-3-319-76620-1_12

purposes. A set of statistical tests for randomness is described in this document." The NIST tests is a package consisting of 15 tests, and in each of these 15 cases one has to compute the value of a certain statistics composed from the elements of the given sequence. Then we have to check whether this value is close enough to the expected value of this statistics for a random binary sequence. If, say, we want to check the quality of a PRBG (=pseudorandom bit generator; an algorithm generating a long bit sequence from a short random one called "seed"), then this can be done by testing several bit sequences generated from random seeds by the PRBG; if these sequences pass the NIST tests then the PRBG is suitable for further consideration. As the NIST tests writes: "These tests may be useful as a first step in determining whether or not a generator is suitable for a particular cryptographic application. However, no set of statistical tests can absolutely certify a generator as appropriate for usage in a particular application, i.e., statistical testing cannot serve as a substitute for cryptanalysis." The weak point of this "first step" by using the NIST tests is that they are of a posteriori type, i.e., we do not have any a priori control of the pseudorandom quality of the output sequences of the PRBG so that we do not know anything about the output sequences not tested by the NIST tests.

Thus one might like to replace this a posteriori type testing based on the NIST tests with a method for a priori testing (called "theoretical testing" by Knuth) of all the output sequences of the PRBG (which seems to be a too optimistic goal) or at least to combine and complete the NIST tests by a method of this type (this is a more realistic goal). In 1997 Mauduit and the third author [18] made a significant step in this direction: they introduced certain measures of pseudorandomness, and they presented an example for binary sequence which possess strong pseudorandom properties in terms of these measures. Since that more than 150 papers have been written in which these measures are studied, further measures are introduced, or further "good" constructions are presented; an excellent survey of these papers is given by Gyarmati [12]. It is a natural question to ask: how is this direction related to the NIST tests? Can one, indeed, complete the a posteriori testing by using these new results? A partial answer was given by the second and third author in [25]: they studied the connection of 3 NIST tests and a further often used test with the measures of pseudorandomness introduced in [18], and they showed that the values of the statistics to be computed in each of these tests can be estimated well by using the measures of pseudorandomness mentioned above, moreover, they also presented numerical calculations to show that a "random" sequence selected from a family of binary sequences constructed by using the Legendre symbol [10,18] passes all the NIST tests (of 2005). In this paper our goal is to continue that work in the following direction: we will study the connection between 3 further NIST tests and our measures of pseudorandomness. (The 8 remaining NIST tests are too complicated to study their theoretical connection with the measures of pseudorandomness.) Moreover, we will present further numerical calculations to show that a "random" sequence selected from two other families constructed by using other principles also passes or "almost passes" all the NIST tests.

2 The Measures of Pseudorandomness

First we recall a few definitions and facts from [18] and other related papers that we will need in this paper.

Consider a finite binary sequence

$$E_N = (e_1, \dots, e_N) \in \{-1, +1\}^N. \tag{1}$$

(Note that in the analysis in some of the NIST tests the bit sequences are also transformed into sequences consisting of -1 and $+1$.) Then the *well-distribution measure* of E_N is defined as

$$W(E_N) = \max_{a,b,t} \left| \sum_{j=0}^{t-1} e_{a+jb} \right|, \tag{2}$$

where the maximum is taken over all $a, b, t \in \mathbb{N}$ such that $1 \le a \le a+(t-1)b \le N$, while the *correlation measure of order* k of E_N is defined as

$$C_k(E_N) = \max_{M,D} \left| \sum_{n=1}^{M} e_{n+d_1} e_{n+d_2} \dots e_{n+d_k} \right|, \tag{3}$$

where the maximum is taken over all $D = (d_1 \dots, d_k)$ and M such that $0 \le d_1 < \dots < d_k \le N - M$. Then the sequence is considered as a "good" pseudorandom sequence if both these measures $W(E_N)$ and $C_k(E_N)$ (at least for "small" k) are "small" in terms of N (in particular, both are $o(N)$ as $N \to \infty$). Indeed, it is shown in [4] that for a "truly random" $E_N \in \{-1, +1\}^N$ both $W(E_N)$ and, for fixed k, $C_k(E_N)$ are of order of magnitude $N^{1/2}$ with probability "near 1" (see also [2,14]). Thus for "really good" pseudorandom sequences we expect the measures (2) and (3) to be not much greater than $N^{1/2}$. In [18] a combination of the well-distribution and correlation measures was also introduced: the *combined pseudorandom measure of order* k of the sequence E_N in (1) is defined as

$$Q_k(E_N) = \max_{a,b,t,D} \left| \sum_{j=0}^{t} e_{a+jb+d_1} e_{a+jb+d_2} \dots e_{a+jb+d_k} \right|$$

where the maximum is taken over all $a, b, t \in \mathbb{N}$ and k-tuples $D = (d_1, d_2, \dots, d_k)$ of non-negative integers $d_1 < d_2 < \dots < d_k$ such that all the subscripts $a+jb+d_l$ belong to $\{1, 2, \dots, N\}$. (Clearly, we have $W(E_N) = Q_1(E_N)$ and $C_k(E_N) \le Q_k(E_N)$ for $k \ge 2$.) We will also need the definition of normality measures also introduced in [18]. The *normality measure of order* k of the sequence E_N of form (1) is defined as

$$N_k(E_N) = \max_{X \in \{-1,+1\}^k} \max_{0 < M \le N+1-k} \left| |\{n : 0 \le n \le M,\ (e_{n+1}, \dots, e_{n+k}) = X\}| - \frac{M}{2^k} \right|. \tag{4}$$

It was also shown in [18] (see Proposition 1 and its proof there) that for all N, E_N and $k < N$ we have

$$N_k(E_N) \le \frac{1}{2^k} \sum_{t=1}^{k} \binom{k}{t} C_t(E_N) \le \max_{1 \le t \le k} C_t(E_N). \tag{5}$$

Thus if $C_t(E_N)$ is small for all $t \le k$, then $N_k(E_N)$ is also small.

3 Three Principles for Constructing Large Families of Binary Sequences with Strong Pseudorandom Properties

It is well known that the Legendre polynomial has many pseudorandom properties [6,7]. It was shown in [18] that the Legendre symbol also possesses strong pseudorandom properties in terms of the pseudorandom measures described in Sect. 2: if p is an odd prime, we write $N = p - 1$ and

$$E_N = (e_1, \ldots, e_N) \quad \text{with } e_n = \left(\frac{n}{p}\right) \text{ for } n = 1, \ldots, N,$$

then we have

$$W(E_N) \ll N^{1/2} \log N \qquad \text{and} \qquad C_k(E_N) \ll kN^{1/2} \log N$$

for all $k < N$ (where $\ll$ is Vinogradov's notation: $f(x) \ll g(x)$ means that $f(x) = O(g(x))$; in both cases the implicit constants can be computed explicitly (and are relatively small constants).

Goubin et al. [10] studied the generalization of this construction with $f(n)$ in place of n (where $f(x) \in \mathbb{F}_p[x]$). Their results can be combined in the following way:

Theorem A. *If p is a prime number, $f(x) \in \mathbb{F}_p[x]$ ($\mathbb{F}_p$ being the field of the modulo p residue classes) has degree $k (> 0)$, $f(x)$ has no multiple zero in $\overline{\mathbb{F}_p}$ (= the algebraic closure of $\mathbb{F}_p$), and the binary sequence $E_p = (e_1, \ldots, e_p)$ is defined by*

$$e_n = \begin{cases} \left(\frac{f(n)}{p}\right) & \text{for } (f(n), p) = 1 \\ +1 & \text{for } p \mid f(n), \end{cases} \tag{6}$$

then we have

$$W(E_p) < 10kp^{1/2} \log p.$$

Moreover, assume that also $\ell \in \mathbb{N}$, and one of the following assumptions holds:

(i) $\ell = 2$;
(ii) $\ell < p$, *and 2 is a primitive root modulo* p;
(iii) $(4k)^\ell < p$.

Then we also have

$$C_\ell(E_p) < 10k\ell p^{1/2}\log p.$$

The second principle is to utilize the fact that the multiplicative inverse modulo p is distributed in a random way in $(0,p)$. Denote the least non-negative residue of n modulo p by $r_p(n)$, and if the prime p is fixed, then denote the multiplicative inverse of a modulo p by a^{-1} (so that $a \cdot a^{-1} \equiv 1 \mod p$). The following theorem (here we present the result in a slightly simplified form) was proved in [20].

Theorem B. *Assume that* p *is an odd prime number,* $f(x) \in \mathbb{F}_p[x]$ *has degree* k *with* $0 < k < p$ *and no multiple zero in* $\overline{\mathbb{F}_p}$. *Define the binary sequence* $E_p = (e_1,\dots,e_p)$ *by*

$$e_n = \begin{cases} +1 \text{ if } (f(n),p)=1,\ r_p(f(n)^{-1}) < \frac{p}{2} \\ -1 \text{ if either } (f(n),p)=1,\ r_p(f(n)^{-1}) > \frac{p}{2} \text{ or } p \mid f(n). \end{cases} \tag{7}$$

Then we have

$$W(E_p) \ll kp^{1/2}(\log p)^2.$$

Moreover, if $\ell \in \mathbb{N}$, $2 \le \ell \le \frac{p}{2^k}$ *and* $f(x) \in \mathbb{F}_p[x]$ *is of the form* $f(x) = (x+a_1)(x+a_2)\cdots(x+a_k)$ *with* $a_1,\dots,a_k \in \mathbb{F}_p$ *(*$a_i \ne a_j$ *for* $i \ne j$*) then we also have*

$$C_\ell(E_p) \ll k\ell p^{1/2}(\log p)^{\ell+1}. \tag{8}$$

For further related results see also [5,15,16]. For example Liu [15] gave another (and simpler) condition to control the correlation measure.

Theorem C. *Assume that* p *is an odd prime number,* $f(x) \in \mathbb{F}_p[x]$ *is a polynomial of degree* $(0<)k(<p)$ *such that 0 is its unique zero in* $\mathbb{F}_p$. *If the sequence* E_N *is defined as in Theorem B and* $\ell < p$, *then* (8) *also holds.*

The third construction is based on elliptic curves. Let $p > 3$ be a prime number and let $\mathbf{E}$ be an elliptic curve over $\mathbb{F}_p$ defined by the Weierstrass equation

$$y^2 = x^3 + Ax + B$$

with coefficients $A, B \in \mathbb{F}_p$ and non-zero discriminant (see [30]). The $\mathbb{F}_p$-rational points $\mathbf{E}(\mathbb{F}_p)$ of $\mathbf{E}$ form an Abelian group with the point in infinity $\mathcal{O}$ as the neutral element, where the group operation is denoted by $\oplus$. For a rational point $R \in \mathbf{E}(\mathbb{F}_p)$, a multiple of R is defined by $nR = \bigoplus_{i=1}^{n} R$. Let $\mathbb{F}_p(\mathbf{E})$ be the function field of $\mathbf{E}$ over $\mathbb{F}_p$ and as usual for $f \in \mathbb{F}_p(\mathbf{E})$ let $\deg f$ denote of the degree of f in $\mathbb{F}_p(\mathbf{E})$, see [30]. For example, for the coordinate functions we have $\deg x = 2$ and $\deg y = 3$.

Let $G \in \mathbf{E}(\mathbb{F}_p)$ be of order T and $f \in \mathbb{F}_p(\mathbf{E})$. Define the binary sequence $E_T = (e_1,\dots,e_T)$ by

$$e_n = \begin{cases} \left(\frac{f(nG)}{p}\right) & \text{if } (f(nG),p)=1, \\ +1 & \text{otherwise.} \end{cases} \tag{9}$$

The first author studied the pseudorandomness of this sequence [21]. His results can be combined in the following way:

Theorem D. *Let $G \in \mathbf{E}(\mathbb{F}_p)$ be a generator of $\mathbf{E}(\mathbb{F}_p)$ of prime order T. Let $f \in \mathbb{F}_p(\mathbf{E})$ which is not a perfect square in $\overline{\mathbb{F}_p}(\mathbf{E})$ with degree $k = \deg f$. Then*

$$W(E_T) \leq 6kp^{1/2} \log T.$$

Moreover, assume that also $\ell \in \mathbb{N}$, and one of the conditions (i), (ii), (iii) of Theorem A holds with p replaced by T. Then

$$C_\ell(E_T) < 2\ell kp^{1/2} \log T.$$

4 The "Frequency Test Within a Block"

First we will study the connection of this test (which appears as Sect. 2.2 in [26]) with the measures of pseudorandomness described in Sect. 2. We quote [26]: "The focus of this test is to determine whether the frequency of ones in an M-bit block is approximately $M/2$, as would be expected under an assumption of randomness. For block size $M = 1$, this test degenerates to test 1, the Frequency (Monobit) test" (which was analyzed in [25]).

Let $E_N = (e_1, \dots, e_N) \in \{-1, +1\}^N$ be the sequence to be tested, M the length of each block, and, as [26] writes,

> "Partition the input sequence into $t = [\frac{N}{M}]$ non-overlapping blocks."

(Here and later we adjust the notations of [26] to our notation.) The quotation continues:

> "Discard any unused bits. Determine the proportion π_i of ones in each block of length M for $1 \leq i \leq t$"

Now "Compute the χ^2 statistic

$$X_1 = 4M \sum_{i=1}^{t} \left(\pi_i - \frac{1}{2} \right)^2 ." \tag{10}$$

Then the sequence E_N passes this test if the value of this statistic is small enough in the sense described in [26]; we skip the technical details.

In 2.2.7 [26] writes: "The block size M should be selected such that

$$M \geq 20,\ M > N/100 \text{ and } t < 100." \tag{11}$$

Theorem 1. *Using the notation above and assuming (11), for every $E_N \subset \{-1, +1\}^N$ we have*

$$X_1 \leq 2 \cdot 10^4 \frac{W(E_N)^2}{N}. \tag{12}$$

Proof. Clearly we have

$$\begin{aligned}\pi_i &= \frac{|\{e_j : \ (i-1)M < j \leq iM, e_j = +1\}|}{M} \\ &= \frac{1}{M} \sum_{j=(i-1)M+1}^{iM} \frac{1}{2}(e_j + 1) = \frac{1}{2M} \sum_{j=(i-1)M+1}^{iM} e_j + \frac{1}{2}\end{aligned}$$

whence, using the notation of Sect. 2,

$$\left| \pi_i - \frac{1}{2} \right| = \frac{1}{2M} \left| \sum_{j=(i-1)M+1}^{iM} e_j \right| \leq \frac{1}{2M} W(E_N)$$

for every $1 \leq i \leq t$. Thus it follows from (10) that

$$X_1 \leq 4M \cdot t \left(\frac{1}{2M} W(E_N) \right)^2 = \frac{2t}{M} W(E_N)^2.$$

By using (11), (12) follows from this.

In each of the constructions described in Sect. 3 the upper bound in inequality (12) is less than a constant multiple of a fixed power of $\log N$, so that this upper bound falls just a little short of the desired $<c$ (with a small positive constant c). In many applications this can be interpreted as a strong tendency towards pseudorandomness which is sufficient for our purposes, while if we have to stick to the threshold bound belonging to the test, then this good upper bound points to the direction that choosing successive random sequences from our family studied we have a good chance to find soon a sequence which also satisfies the stronger inequality prescribed in the test.

5 The "Test for the Longest Run of Ones in a Block"

This test appears in Sect. 2.4 of [26]. We quote [26]: "The focus of the test is the longest run of ones within M bit blocks. The purpose of this test is to determine whether the length of the longest run of ones within the tested sequence is consistent with the length of the largest run of ones that would be expected in a random sequence.". The test to answer this question is carried out in [26] in the following way:

Assume the N, M, t are positive integers with

$$N = Mt, \tag{13}$$

N is the length of the sequence $E_N = (e_1, \dots, e_N) \in \{-1, +1\}^N$ to be tested (again we switch from bit sequences to ± 1 sequences), M is taken from a certain special sequence $8, 128, 10^4, \dots$ (see [26]), E_N is split in t blocks of length M, t and thus also N is large enough in terms of M (in particular, for

$M = 8, 128, 10^4$ the number N must be at least $128, 272, 750000$, respectively) and K (=3, 5, 6, ...) is certain positive integer assigned to the given M value. The set $\{0, 1, \dots, M\}$ is split in $K + 1$ disjoint parts $\mathcal{P}_0, \mathcal{P}_1, \dots, \mathcal{P}_K$ so that

$$\{0, 1, \dots, M\} = \mathcal{P}_0 \cup \mathcal{P}_1 \cup \dots \cup \mathcal{P}_K, \quad \mathcal{P}_i \cap \mathcal{P}_j = \emptyset \quad \text{for } 0 \le i < j \le K, \tag{14}$$

e.g., for $M = 10^4$, $K = 6$ in [26] we have

$$\{0, 1, \dots, 10^4\} = \{0, 1, \dots, 10\} \cup \{11\} \cup \{12\} \cup \{13\} \cup \{14\} \cup \{15\} \cup \{16, 17, \dots, 10^4\}.$$

Then for $i = 0, 1, \dots, K$ we count how many of the t blocks is such that the length of the longest run of +1's in it belongs to the part $\mathcal{P}_i$ of $\{0, 1, \dots, M\}$; let ν_i denote the number of blocks with this property. Let π_i be the probability of the event that the length of the longest run of +1's in a random sequence of +1 and −1 with length M is i. The test statistic to be computed is a weighted square mean of the deviations of the ν_i's from their expected values $t\pi_i$:

$$X_2 = \sum_{i=0}^{K} \frac{(\nu_i - t\pi_i)^2}{t\pi_i}, \tag{15}$$

"which, under the randomness hypothesis, has an approximate χ^2-*distribution with* K *degrees of freedom*". Here the theoretical values π_i can be replaced by approximating numerical values which for certain pairs M, K are provided in Sect. 3.4 of [26]. (We remark that for fixed M, the choice of K and the computation of the values approximating π_i is based on the analysis of distribution of the longest run in random walks; see, e.g. [24, Chap. 7].)

We will show that the statistic X_2 in (15) can be estimated in the following way:

Theorem 2. *We have*

$$X_2 \le \frac{M}{N} \left(\sum_{r=1}^{M} \binom{M}{r} Q_r(E_N) \right)^2 .$$

Note that this estimate gives a good bound for X_2 only if N is large in terms of M; the first table in [26], pp. 2–8 seems to indicate that this can be assumed.

Proof. We will use the following notations:

For $Z \in \mathbb{N}$, let Φ_Z be the set of the binary sequences

$$F_Z = (f_1, f_2, \dots, f_Z) \in \{-1, +1\}^Z,$$

for such a sequence F_Z let $\psi(\mathbb{F}_Z)$ denote the length of the longest run of +1's in F_Z, and for $j \in \mathbb{N}$, $jM \le Z$, write $F_Z^{(j,M)} = (f_{(j-1)M+1}, f_{(j-1)M+2}, \dots, f_{jM})$.

Then for $i = 0, 1, \dots, K$, by the definition of ν_i and (13) we have

$$\nu_i = \sum_{\substack{1 \le j \le t \\ \psi(E_N^{(j,M)}) \in \mathcal{P}_i}} 1, \tag{16}$$

and the expectation of ν_i choosing any $F_N \in \Phi_N$ with equal probability $1/2^N$ is

$$\mathbb{E}(\nu_i) = \mathbb{E}\left(\sum_{\substack{1\le j\le t\\ \psi(E_N^{(j,M)})\in\mathcal{P}_i}} 1\right) = t\cdot\mathbb{E}\left(\sum_{\substack{G\in\Phi_M\\ \psi(G)\in\mathcal{P}_i}} 1\right) = t\pi_i. \tag{17}$$

Let $G^{(1)}, G^{(2)}, \dots, G^{(\gamma_i)}$ be the sets G counted in the last sum, and write $\mathcal{G}_i = \{G^{(1)}, G^{(2)}, \dots, G^{(\gamma_i)}\}$ and $G^{(j)} = (g_1^{(j)}, g_2^{(j)}, \dots, g_M^{(j)})$. Each of these sets $G^{(j)}$ contributes by 1 to this sum, and they are to be selected with probability $1/2^M$ uniformly. Thus it follows from (17) that

$$\mathbb{E}(\nu_i) = t\pi_i = t\frac{|\mathcal{G}_i|}{2^M} = \frac{t}{2^M}\gamma_i. \tag{18}$$

Moreover, it follows from (14) that each of the 2^M sets $G \in \Phi_M$ is counted in exactly one $\mathcal{G}_i$ with weight 1, thus we have

$$\sum_{i=0}^{K} |\mathcal{G}_i| = \sum_{i=0}^{K} \gamma_i = 2^M. \tag{19}$$

Now we will estimate ν_i for $0 \le i \le K$. Consider a subset $G^{(\ell)} = (g_1^{(\ell)}, g_2^{(\ell)}, \dots, g_M^{(\ell)}) \in \mathcal{G}_i$. Then for $j = 1, 2, \dots, t$ clearly we have

$$\prod_{x=1}^{M} \frac{1 + e_{(j-1)M+x}g_x^{(\ell)}}{2} = \begin{cases} 1 \text{ if } E_N^{(j,M)} = G^{(\ell)}, \\ 0 \text{ if } E_N^{(j,M)} \ne G^{(\ell)}, \end{cases}$$

whence

$$\sum_{\ell=1}^{\gamma_i}\prod_{x=1}^{M} \frac{1 + e_{(j-1)M+x}g_x^{(\ell)}}{2} = \begin{cases} 1 \text{ if } E_N^{(j,M)} \in \{G^{(1)}, G^{(2)}, \dots, G^{(\gamma_i)}\} = \mathcal{G}_i, \\ 0 \text{ if } E_N^{(j,M)} \notin \mathcal{G}_i, \end{cases}$$

so that by (16) we have

$$\begin{aligned} \nu_i = &\sum_{\substack{1\le j\le t\\ \psi(F_N^{(j,M)})\in\mathcal{P}_i}} 1 = \sum_{1\le j\le t}\sum_{E_N^{(j,M)}\in\mathcal{G}_i} 1 = \sum_{j=1}^{t}\sum_{\ell=1}^{\gamma_i}\prod_{x=1}^{M}\frac{1 + e_{(j-1)M+x}g_x^{(\ell)}}{2} \\ &= \sum_{\ell=1}^{\gamma_i}\sum_{j=1}^{t}\left(\frac{1}{2^M} + \frac{1}{2^M}\sum_{r=1}^{M}\sum_{1\le x_1<\dots<x_r\le M} g_{x_1}^{(\ell)}\dots g_{x_r}^{(\ell)} e_{(j-1)M+x_1}\dots e_{(j-1)M+x_r}\right) \\ &= \frac{t}{2^M}\gamma_i + \frac{1}{2^M}\sum_{\ell=1}^{\gamma_i}\left(\sum_{r=1}^{M}\sum_{1\le x_1<\dots<x_r\le M} g_{x_1}^{(\ell)}\dots g_{x_r}^{(\ell)}\sum_{j=1}^{t} e_{(j-1)M+x_1}\dots e_{(j-1)M+x_r}\right). \end{aligned} \tag{20}$$

It follows from (18) and (20) that

$$\begin{aligned}
|\nu_i - t\pi_i| &= \frac{1}{2^M}\left|\sum_{\ell=1}^{\gamma_i}\left(\sum_{r=1}^{M}\sum_{1\le x_1<\cdots<x_r\le M} g_{x_1}^{(\ell)}\ldots g_{x_r}^{(\ell)}\sum_{j=1}^{t} e_{(j-1)M+x_1}\ldots e_{(j-1)M+x_r}\right)\right|\\
&\le \frac{1}{2^M}\sum_{\ell=1}^{\gamma_i}\sum_{r=1}^{M}\sum_{1\le x_1<\cdots<x_r\le M}\left|g_{x_1}^{(\ell)}\ldots g_{x_r}^{(\ell)}\right|\left|\sum_{j=1}^{t} e_{(j-1)M+x_1}\ldots e_{(j-1)M+x_r}\right|\\
&= \frac{\gamma_i}{2^M}\sum_{r=1}^{M}\binom{M}{r}Q_r(E_N) = \pi_i\sum_{r=1}^{M}\binom{M}{r}Q_r(E_N). \qquad (21)
\end{aligned}$$

By (13), (18), (19) and (21) we have

$$\begin{aligned}
X_2 = \sum_{i=0}^{K}\frac{(\nu_i - t\pi_i)^2}{t\pi_i} &\le \sum_{i=0}^{K}\frac{\pi_i}{t}\left(\sum_{r=1}^{M}\binom{M}{r}Q_r(E_N)\right)^2\\
&= \frac{1}{t}\left(\sum_{r=1}^{M}\binom{M}{r}Q_r(E_N)\right)^2\sum_{i=0}^{K}\pi_i = \frac{M}{N}\left(\sum_{r=1}^{M}\binom{M}{r}Q_r(E_N)\right)^2
\end{aligned}$$

which completes the proof of the theorem.

We remark that in Theorem 2 the statistic X_2 is estimated in terms of the combined pseudorandom measure Q_k, while in the most important constructions studied in Theorems A, B, C and D only the measures W and C_k were estimated, and no estimates are known for the measures Q_k (and the situation is similar in most of the other constructions). However, this gap can be bridged easily, since in most cases the estimate of Q_k can be reduced easily to the estimate of C_k. For example, in case of the Legendre symbol construction (6) studied in Theorem A, we can show that if the sequence E_p is defined by (6) in Theorem A and we assume that all the assumptions in the theorem hold, then we have

$$Q_k(E_p) \le C_k(E_p) + 2k,$$

and in case of the two other constructions similar results could be proved.

6 The "Linear Complexity Test"

The *linear complexity* $L(\tilde{E}_N)$ of a *bit* sequence $\tilde{E}_N = (\tilde{e}_1,\ldots,\tilde{e}_N) \in \{0,1\}^N$ is defined as the length L of a shortest linear recurrence relation (linear feedback shift register – LFSR)

$$\tilde{e}_{n+L} \equiv c_{L-1}\tilde{e}_{n+L-1} + \cdots + c_1\tilde{e}_{n+1} + c_0\tilde{e}_n \pmod 2, \quad 1 \le n \le N-L$$

where $c_0,\ldots,c_{L-1} \in \{0,1\}$, that $\tilde{E}_N$ satisfies, with the convention that $L(\tilde{E}_N) = 0$ if $\tilde{E}_N = (0,\ldots,0)$, and $L(\tilde{E}_N) = N$ if $\tilde{E}_N = (0,\ldots,0,1)$. For binary sequence E_N of form (1) we also define the linear complexity as $L(E_N) = L(\tilde{E}_N)$ with $\tilde{e}_n = (1+e_n)/2$.

The linear complexity is a measure for the unpredictability of a sequence. A large linear complexity is necessary (but not sufficient) for cryptographic applications. The *linear complexity test* appears as Sect. 2.10 in [26]. We quote: "The purpose of this test is to determine whether or not the sequence is complex enough to be considered random. Random sequences are characterized by longer LFSRs. An LFSR that is too short implies non-randomness."

Brandstätter and Winterhof [3] showed that a small correlation measure implies large linear complexity:

$$L(E_N) \geq N - \max_{1 \leq k \leq L(E_N)+1} C_k(E_N). \tag{22}$$

This result provides a lower bound for the linear complexity of sequences generated by using the Legendre symbol (6) and elliptic curves (9). Namely, if E_p is a sequence generated by (6) using a squarefree polynomial $f(x) \in \mathbb{F}_p[x]$ of degree k and 2 is a primitive root modulo p, then Theorem A and (22) imply that

$$p \leq L(E_p) + \max_{1 \leq k \leq L(E_p)+1} C_k(E_p) \ll kLp^{1/2} \log p,$$

so that

$$L(E_p) \gg \frac{p^{1/2}}{k \log p}. \tag{23}$$

Similarly, if the sequence E_T is generated by (9) using a squarefree function $f(x, y) \in \mathbb{F}_p[\mathbf{E}]$ with degree k and 2 is a primitive root modulo T, then Theorem D and (22) imply that

$$L(E_T) \gg \frac{p^{1/2}}{k \log T}.$$

By the celebrated Hasse-Weil Theorem (see e.g. [30], Theorem 4.2) we have $\big|p + 1 - |\mathbf{E}(\mathbb{F}_p)|\big| \leq 2p^{1/2}$, thus

$$L(E_T) \gg \frac{T^{1/2}}{k \log T}. \tag{24}$$

In practice, the bounds (23) and (24) are sometimes sufficient. However, the linear complexity of a truly random binary sequence of length N is around $N/2$, thus in more demanding applications one may have to show that the linear complexity of the given sequence is near $N/2$ (or at least it is $\gg N$); the linear complexity of the sequence can be determined by using the well-known Berlekamp-Massey algorithm [17].

In even more demanding cases one may need an even more precise study of the complexity properties of the sequence. In [26] this is done by the "linear complexity test" described in [26, pp. 2–24]. This test requires a more controlled distribution of the linear complexity of the sequences. Namely, it compares the linear complexity within blocks of length M to the expected value of the linear complexity

$$\mu_M = \frac{M}{2} + \frac{4 + r_2(M)}{18}$$

(here again $r_2(M)$ is the non-negative remainder of M modulo 2).

We quote: "Partition the N-bit sequence into t independent blocks of M bits, where $N = t \cdot M$." Then "determine the linear complexity L_i of each of the t blocks ($i = 1, \ldots, t$)". "For each substring, calculate a value of T_i where

$$T_i = (-1)^M \cdot (L_i - \mu_M) + \frac{2}{9}."$$

Define the intervals:

$$\begin{aligned} &I_0 = (-\infty, -2, 5], \\ &I_j = (-2.5 + j - 1, -2.5 + j], j = 1, \ldots, 5, \\ &I_6 = (2.5, \infty), \end{aligned}$$

and put $v_j = |\{i : T_i \in I_j,\ i = 1, \ldots, t\}|$.

Finally, we define the statistic

$$X_3 = \sum_{j=0}^{6} \frac{(v_j - t \cdot \pi_j)^2}{t \cdot \pi_j},$$

where π_j $(i = 0, \ldots, 6)$ are the probabilities for the classes I_j:

$$\pi_j = P\left((-1)^M \cdot (L(E_M) - \mu_M) + \frac{2}{9} \in I_j\right),$$

where E_M is chosen uniformly from $\{-1, +1\}^M$. The acceptance of the sequence depends on the value of the statistic X_3: one has to compute the "P-value" defined in [26], pp. 2–25, (7) and if the "P-value" is ≥ 0.01, then the sequence passes the test.

If the sequence E_N to be tested possesses strong pseudorandom properties in terms of the measures described in Sect. 2, and the length M of the blocks is much smaller than the length N of the sequence (say, we have $M = o(\log N)$), then one could give a reasonable upper bound for the statistic X_3 by the method used in Sect. 5 (although here even more work and computation would be needed). However, according to "input size recommendation" in [26, Sect. 2.10.7], M must be very large ($500 \leq M \leq 5000$) so that to have $M = o(\log N)$, N must be huge (say, $N > 10^{10000}$), thus we will not present the details here. This, of course, does not mean that shorter sequences with good pseudorandom properties fail this test and, indeed, the numerical examples in Sect. 8 will show that sequences of this type tend to pass this test, but we cannot show that this is necessarily so.

Remark 6.1. As mentioned previously, small linear complexity implies non-randomness. However, recent results show that there are many sequences whose linear complexity is very near to its expected value but which also have some cryptographic weakness: Winterhof and the first author provided a large class of highly predictable sequences whose linear complexity is close to its mean [23]. A simple way to eliminate such sequences is to consider also the *expansion complexity* of the sequences defined in [8,22].

7 Discrete Fourier Transform (Spectral) Test

The “NIST tests” writes: “The purpose of this test is to detect periodic features (i.e., repetitive patterns that are near each other) in the tested sequence that would indicate a deviation from the assumption of randomness”.

This is one of the tests which are too complicated to estimate the statistic to be studied by using our measures of pseudorandomness. Instead, we will do the following: we show that the goal of the test described above can be also achieved by using our measures of pseudorandomness and, indeed, the combined pseudorandom measure of order k described above is especially suitable for this. Take the following example:

Example 7.1. Consider the 4-tuple $+1, -1, -1, +1$, and repeat it $M = 500000$ times. Then letting $N = 4M = 2000000$, we get a binary sequence $E_N = (e_1, e_2, \dots, e_N)$ with $e_{4k-3} = +1$, $e_{4k-2} = -1$, $e_{4k-1} = -1$, $e_{4k} = +1$ for $k = 1, 2, \dots, M$. This sequence is periodic with period 4. Its combined pseudorandom measure of order 4 can be estimated in the following way:

$$Q_4(E_N) \geq \left| \sum_{j=0}^{M-1} e_{4j+1} e_{4j+2} e_{4j+3} e_{4j+4} \right| = \left| \sum_{j=0}^{M-1} 1 \right| = M = \frac{N}{4},$$

so that this measure is big, is as large as $\frac{1}{4}$ times the length of the sequence, much larger than the optimal $\asymp N^{1/2}$. This fact is reflected in the periodicity, thus the sequence is far from being of pseudorandom nature.

Of course, if a sequence is not completely periodic but is only almost periodic with period k, than its Q_k measure is still large.

Applying the Discrete Fourier Transform Test for testing the sequence E_N defined above with 20 samples of length 100000, we find that it fails this test (strongly).

So that both approaches point out the periodic nature of E_N, thus *it fails both tests.*

Now let us study a more complicated example.

Example 7.2. Consider two especially important special sequences: the Rudin-Shapiro sequence (defined by $(-1)^{\sum_i \varepsilon_i(n)\varepsilon_{i+1}(n)}$ where ε_i denotes the i-th binary digit of n) and the Thue-Morse sequence (defined by $(-1)^{\sum_i \varepsilon_i(n)}$). It is known [19] that in both cases if we take a sequence of length N then its correlation measure of order 2 is very large, it is $\gg N$ which is again much larger than the optimal $\asymp N^{1/2}$ so that in terms of the measures of pseudo randomness described above they are both far from being of pseudorandom nature.

But what about the Discrete Fourier Transform Test, do these sequences also fail this test? First consider the Rudin-Shapiro sequence (Fig. 1):

```
------------------------------------------------------------------------------
RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES
------------------------------------------------------------------------------
 C1  C2  C3  C4  C5  C6  C7  C8  C9 C10  P-VALUE  PROPORTION  STATISTICAL TEST
------------------------------------------------------------------------------
  3   3   0   2   3   3   1   2   3   0  0.637119     20/20      Frequency
  4   0   0   0   3   2   0   0   0  11  0.000000 *   16/20   *  BlockFrequency
  2   3   5   2   1   2   1   1   1   2  0.637119     20/20      CumulativeSums
  2   1   6   2   0   3   2   1   1   2  0.213309     20/20      CumulativeSums
  0   0   0   0   0   0   0   0   0  20  0.000000 *   20/20      Runs
 20   0   0   0   0   0   0   0   0   0  0.000000 *    0/20   *  LongestRun
 20   0   0   0   0   0   0   0   0   0  0.000000 *    0/20   *  Rank
 20   0   0   0   0   0   0   0   0   0  0.000000 *    0/20   *  FFT
 20   0   0   0   0   0   0   0   0   0  0.000000 *    0/20   *  Universal
 20   0   0   0   0   0   0   0   0   0  0.000000 *    0/20   *  ApproximateEntropy
 20   0   0   0   0   0   0   0   0   0  0.000000 *    0/20   *  Serial
 20   0   0   0   0   0   0   0   0   0  0.000000 *    0/20   *  Serial
 20   0   0   0   0   0   0   0   0   0  0.000000 *    0/20   *  LinearComplexity
```

Fig. 1. Results of 13 NIST tests for the Rudin-Shapiro sequence

```
------------------------------------------------------------------------------
RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES
------------------------------------------------------------------------------
 C1  C2  C3  C4  C5  C6  C7  C8  C9 C10  P-VALUE  PROPORTION  STATISTICAL TEST
------------------------------------------------------------------------------
  0   0   0   0   0   0   0   0   0  20  0.000000 *   20/20      Frequency
  0   0   0   0   0   0   0   0   0  20  0.000000 *   20/20      BlockFrequency
  0   0   0   0   0   0   0   0   0  20  0.000000 *   20/20      CumulativeSums
  0   0   0   0   0   0   0   0   0  20  0.000000 *   20/20      CumulativeSums
 20   0   0   0   0   0   0   0   0   0  0.000000 *    0/20   *  Runs
 20   0   0   0   0   0   0   0   0   0  0.000000 *    0/20   *  LongestRun
 20   0   0   0   0   0   0   0   0   0  0.000000 *    0/20   *  Rank
 19   0   0   0   0   0   0   0   1   0  0.000000 *    2/20   *  FFT
 20   0   0   0   0   0   0   0   0   0  0.000000 *    0/20   *  Universal
 20   0   0   0   0   0   0   0   0   0  0.000000 *    0/20   *  ApproximateEntropy
 20   0   0   0   0   0   0   0   0   0  0.000000 *    0/20   *  Serial
 20   0   0   0   0   0   0   0   0   0  0.000000 *    0/20   *  Serial
 20   0   0   0   0   0   0   0   0   0  0.000000 *    0/20   *  LinearComplexity
```

Fig. 2. Results of 13 NIST tests for the Thue-Morse sequence

(Here and in some further tables an asterisk indicates after the column "P-value" that the P-values belonging to the sequence studied and the test named in the last column are non-uniform, while after column "proportion" it denotes that many or all of these sequences fail the test in question.) Now consider the Thue-Morse sequence (Fig. 2):

So that both sequences fail this test. The Fourier transform of the Rudin-Shapiro polynomial, whose maximum modulus is very close to its L^2 norm, and this is a very unusual property (see [28]). Similarly, the Fourier transform of the Thue-Morse sequence has a very small L^1 norm (see [9]). This explains why they fail the Discrete Fourier Transform test which detects that they are far from the DFT of a random sequence.

Our examples show that both approaches can be used effectively for detecting some sort of periodicity.

8 Numerical Calculation

In the previous sections we showed that those binary sequences $E_N \in \{-1,+1\}^N$ whose pseudorandom measures $W(E_N)$ and $C_k(E_N)$ are small, also have strong pseudorandom properties in terms of the NIST tests *a priori*, i.e. they provably pass or "almost pass" most of the NIST tests. In this section we test sequences constructed by principles described in Sect. 3 *a posteriori*. The examples show that these sequences typically pass the NIST tests, even if we can only prove a slightly weaker pseudorandomness.

We use "Statistical Test Suite for random and pseudorandom number generators for cryptographic application" (`sts-1.4`) from the National Institute of Standards and Technology (NIST). We chose the following parameters for the test suite (Fig. 3).

`BlockFrequency`	$M = 128$
`OverlappingTemplate`	$m = 9$
`ApproximateEntropy`	$m = 10$
`LinearComplexity`	$M = 500$

Fig. 3. Parameter choices for NIST test suite

In order to save space we omit the results of the non-overlapping template matching (`NonOverlappingTemplate`), the random excursions (`Random Excursions`) and the random excursions variant tests (`RandomExcursions Variant`).

The results of the tests are given in Figs. 4, 5 and 6. Columns `C1` up to `C10` correspond to the frequency specific to the test. Then the `P-VALUE` is the result of the application of a χ^2-test, and `PROPORTION` is the proportion of sequences that pass the test.

8.1 Sequences Generated Using the Legendre Symbol

We constructed 20 sequences with length $p = 10^5 + 3$ by (6) with the first 20 squarefree polynomial of degree 31 with respect to the lexicographic ordering: $f_i(x) = x^{31} + i$ $(i = 1, \ldots, 20)$. Since 2 is a primitive root modulo $p = 10^5 + 3$, Theorem A implies that all the sequences $E_p(i)$ generated with $f_i(x)$ $(i = 1, \ldots, 20)$ have small well-distribution and correlation measures.

```
------------------------------------------------------------------------------
RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES
------------------------------------------------------------------------------
 C1  C2  C3  C4  C5  C6  C7  C8  C9 C10  P-VALUE  PROPORTION  STATISTICAL TEST
------------------------------------------------------------------------------
  0   0   0   0   0   0   0   0   0  20  0.000000 *   20/20      Frequency
  1   4   2   2   2   2   3   3   1   0  0.739918     20/20      BlockFrequency
  0   1   0   0   0   1   1   4   7   6  0.000199     20/20      CumulativeSums
  0   1   0   0   0   1   1   4   7   6  0.000199     20/20      CumulativeSums
  4   4   1   1   3   1   1   3   0   2  0.437274     20/20      Runs
  1   2   2   1   5   1   2   3   1   2  0.637119     20/20      LongestRun
  2   1   0   3   1   0   4   2   5   2  0.213309     20/20      Rank
  0   0   0   0   0  10   0  10   0   0  0.000000 *   20/20      FFT
  2   6   0   0   1   6   2   1   2   0  0.006196     19/20      OverlappingTemplate
  0  20   0   0   0   0   0   0   0   0  0.000000 *   20/20      Universal
  1   1   6   1   1   2   2   1   4   1  0.162606     20/20      ApproximateEntropy
  1   2   1   1   3   2   2   5   1   2  0.637119     20/20      Serial
  0   4   1   3   1   5   0   2   0   4  0.066882     20/20      Serial
  1   4   1   2   2   3   0   1   3   3  0.637119     20/20      LinearComplexity
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Fig. 4. Results of 14 NIST tests for sequences generated by using the Legendre symbol

8.2 Sequences Generated Using the Multiplicative Inverse

We took $p = 2 \cdot 10^5 + 3$ and considered the polynomials

$$f_i(x) = x \cdot \prod_{j=15(i-1)+1}^{15i} (x^2 + j^2), \quad i = 1, \dots, 20.$$

Since $2 \cdot 10^5 + 3 \equiv 3 \pmod 4$, -1 is quadratic non-residue, and the least non-negative remainders of $-j^2$ $(j = 1, \dots, 300)$ modulo p, $r_p(-j^2)$, are also quadratic non-residues. Then these polynomials satisfy the conditions of Theorem C, thus they have small well-distribution and correlation measures. However the sequences generated by (7) with the polynomials $f_i(x)$ have a non-trivial symmetry. Namely, $f_i(-x) = -f_i(x)$, so $e_n = -e_{p-n}$ for all $1 \leq n < p$ if the sequence $E_p = (e_0, \dots, e_{p-1}) \in \{-1, +1\}^p$ is generated such a way. (For tools to detect such symmetries see [11]). To avoid this phenomenon we just considered the first half of the sequences: $E(i)_{(p+1)/2} = \{e_0(i), \dots, e_{(p-1)/2}(i)\}$, where $e_n(i)$ $(0 \leq n < p/2)$ is defined by the rule (7). In this way we obtained 20 sequences of length 100002.

```
------------------------------------------------------------------------------
RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES
------------------------------------------------------------------------------
 C1  C2  C3  C4  C5  C6  C7  C8  C9 C10  P-VALUE  PROPORTION  STATISTICAL TEST
------------------------------------------------------------------------------
  3   2   1   3   0   2   3   4   1   1  0.637119     20/20      Frequency
  5   2   0   1   3   4   2   1   1   1  0.275709     19/20      BlockFrequency
  3   1   0   2   2   1   3   0   3   5  0.275709     20/20      CumulativeSums
  3   2   2   1   4   0   3   1   4   0  0.350485     20/20      CumulativeSums
  2   3   2   5   3   1   1   2   0   1  0.437274     20/20      Runs
  2   1   1   2   5   2   3   0   2   2  0.534146     20/20      LongestRun
  2   3   3   3   0   0   2   1   4   2  0.534146     20/20      Rank
  2   2   1   5   3   3   1   0   2   1  0.437274     20/20      FFT
  4   1   1   1   3   0   2   4   1   3  0.437274     20/20      OverlappingTemplate
  0   0   0   0   0   0  20   0   0   0  0.000000 *   20/20      Universal
  5   4   1   2   2   2   1   1   2   0  0.350485     20/20      ApproximateEntropy
  1   2   1   2   1   0   2   7   4   0  0.017912     20/20      Serial
  2   1   2   0   3   3   3   2   0   4  0.534146     19/20      Serial
  1   3   3   0   3   2   1   4   1   2  0.637119     20/20      LinearComplexity
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Fig. 5. Results of 14 NIST tests for sequences generated by using the multiplicative inverse

8.3 Sequences Generated Using Elliptic Curves

In order to generate sequences with elliptic curves we chose pseudorandom curves and points following the NIST recommendation (FIPS 186-3). We took the prime $p = 10^5 + 3$ and a pseudorandom elliptic curve of the form

$$y^2 = x^3 - 3x + b$$

over $\mathbb{F}_p$ with the additional restriction, that the number T of the $\mathbb{F}_p$-rational points is prime and 2 is a primitive root modulo T. Then we selected a pseudorandom point P on the curve.

Our parameters were the following:

$$\mathbf{E} : y^2 = x^3 - 3x + 74439 \quad \text{over } \mathbb{F}_{10^5+3}.$$

Its cardinality T is 100523. The point was $P = (85611, 76395)$. We took the functions $f_i(x, y) = x^{31} + x + y + i$ $(i = 0, \dots, 19)$. Since 2 is a primitive root modulo T, and the functions $f_i(x, y)$ $(i = 0, \dots, 19)$ are not perfect squares, Theorem D implies, that the well-distribution and correlation measures of sequences generated by the polynomials $f_i(x, y)$ $(i = 0, \dots, 19)$ are small.

Summarizing: we have considered altogether 60 binary sequences which have been proved to possess good pseudorandom properties in terms of the pseudorandom measures described in Sect. 2, and we tested them by 14 NIST tests. 834 times out of 840 the sequences passed the test so that the NIST tests confirmed the good pseudorandom quality of the sequence.

```
------------------------------------------------------------------------------
RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES
------------------------------------------------------------------------------
 C1  C2  C3  C4  C5  C6  C7  C8  C9 C10  P-VALUE  PROPORTION  STATISTICAL TEST
------------------------------------------------------------------------------
  4   1   3   0   1   2   2   2   2   3  0.739918     20/20      Frequency
  0   3   4   0   3   2   2   1   2   3  0.534146     20/20      BlockFrequency
  1   5   2   1   3   1   0   2   3   2  0.437274     20/20      CumulativeSums
  4   0   2   4   2   1   3   1   2   1  0.534146     20/20      CumulativeSums
  4   2   4   3   2   0   2   1   1   1  0.534146     20/20      Runs
  5   3   1   3   4   0   0   3   1   0  0.090936     20/20      LongestRun
  3   4   1   0   2   1   1   1   4   3  0.437274     19/20      Rank
  3   3   2   2   4   1   1   1   1   2  0.834308     20/20      FFT
  4   1   1   2   1   1   6   2   0   2  0.122325     20/20      OverlappingTemplate
  0   0   0   0   0   0   0   0   0  20  0.000000 *   20/20      Universal
  5   1   2   1   0   5   3   1   1   1  0.122325     20/20      ApproximateEntropy
  1   2   2   3   4   0   1   3   2   2  0.739918     19/20      Serial
  1   1   4   2   1   2   4   1   2   2  0.739918     19/20      Serial
  1   3   3   6   0   2   2   1   1   1  0.162606     20/20      LinearComplexity
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Fig. 6. Results of 14 NIST tests for sequences generated by using the elliptic curves

9 Conclusion Remarks

The tables in Sect. 8 show that the sequences "good" in terms of the measures defined in Sect. 2 are usually also "good" in terms of the NIST tests. Does this mean that we may eliminate the NIST tests, replace them by estimating the pseudorandom measures described above? Certainly not: both methods have advantages and disadvantages. The greatest advantage of using the measures of pseudorandomness described above is that at least for certain special sequences they enable us to provide "a priori", "theoretical" testing without any further computations. An other advantage of this method is that in many constructions (like the ones described in Sect. 3) we can give a good upper bound (simultaneously by just a single computation) for the correlation measure of order k of a sequence of length N for every k with $k < N^c$ (say, with $c = 1/4$), and since it is known [4] that the correlation measures of order k, resp. ℓ are independent if $k \nmid \ell$ and $\ell \nmid k$, thus by estimating the correlation measures whose order is less than N^c, we test $N^{c'}$ (with $c' < c$) independent pseudorandom properties of the sequence, while in the NIST tests only 15 properties are tested. On the other hand, the disadvantage of this approach is that in most cases it is very difficult to estimate these measures, e.g. there are no algorithms for estimating correlation measure of high order. On the other hand, NIST provides good and fast algorithms for performing these tests, while its disadvantage is that it can not be used for "a priori", "theoretical" testing.

In Sect. 2 we described only the most important measures of pseudorandomness and in Sect. 3 we presented only three constructions for sequences having good pseudorandom properties in terms of these measures. There are also other measures of pseudorandomness and many further constructions; a survey of these measures and constructions; a survey of these measures and constructions is presented in [12]. In this paper we have been focusing on studying pseudorandom properties of *single binary sequences*. However, as we mentioned in Sect. 1, if our goal is to test the quality of a pseudorandom generator, then it is not enough to restrict ourselves to testing single sequences; one also has to continue the work by using the tools of cryptanalysis for testing the *family of the sequences* generated by the given algorithm. Tools for helping this work also have been introduced (in the spirit of the measures described in Sect. 2): family complexity [1], cross-correlation measure [13], distance minimum and avalanche effect [29], etc., and in each of these cases constructions have been presented for families possessing good pseudorandom properties in terms of these measures. A survey of this type of papers is presented in [27].

References

1. Ahlswede, R., Khachatrian, L.H., Mauduit, C., Sárközy, A.: A complexity measure for families of binary sequences. Period. Math. Hung. **46**, 107–118 (2003)
2. Alon, N., Kohayakawa, Y., Mauduit, C., Moreira, C.G., Rödl, V.: Measures of pseudorandomness for finite sequences: typical values. Proc. Lond. Math. Soc. **95**, 778–812 (2007)
3. Brandstätter, N., Winterhof, A.: Linear complexity profile of binary sequences with small correlation measure. Period. Math. Hung. **52**(2), 1–8 (2006)
4. Cassaigne, J., Mauduit, C., Sárközy, A.: On finite pseudorandom binary sequences VII: the measures of pseudorandomness. Acta Arith. **103**(2), 97–118 (2002)
5. Chen, Z., Lin, Z.: Modified constructions of binary sequences using multiplicative inverse. Appl. Math. J. Chin. Univ. Ser. B **23**(4), 490–500 (2008)
6. Cusick, T.W., et al.: Stream Ciphers and Number Theory. North-Holland Mathematical Library, vol. 55. North-Holland Publishing Co., Amsterdam (1998)
7. Damgård, I.B.: On the randomness of Legendre and Jacobi sequences. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 163–172. Springer, New York (1990). https://doi.org/10.1007/0-387-34799-2_13
8. Diem, C.: On the use of expansion series for stream ciphers. LMS J. Comput. Math. **15**, 326–340 (2012)
9. Fouvry, E., Mauduit, C.: Sommes des chiffres et nombres presques premiers. Math. Ann. **305**, 571–599 (1996)
10. Goubin, L., Mauduit, C., Sárközy, A.: Construction of large families of pseudorandom binary sequences. J. Number Theory **106**(1), 56–69 (2004)
11. Gyarmati, K.: On a pseudorandom property of binary sequences. Ramanujan J. **8**, 289–302 (2004)
12. Gyarmati, K.: Measures of pseudorandomness. In: Finite Fields and Their Applications. Radon Series on Computational and Applied Mathematics, vol. 11, pp. 43–64. De Gruyter, Berlin (2013)
13. Gyarmati, K., Mauduit, C., Sárközy, A.: The cross-correlation measure for families of binary sequences. In: Larcher, G., et al. (eds.) Applied Algebra and Number Theory, pp. 126–143. Cambridge University Press, Cambridge (2014)

14. Kohayakawa, Y., Mauduit, C., Moreira, C.G., Rödl, V.: Measures of pseudorandomness for finite sequences: minimum and typical values. In: Proceedings of WORDS 2003, TUCS General Publications, vol. 27, pp. 159–169. Turku Centre for Computer Science, Turku (2003)
15. Liu, H.: Large families of pseudorandom binary sequences and lattices by using the multiplicative inverse. Acta Arith. **159**(2), 123–131 (2013)
16. Liu, H., Gao, J.: A note on large families of pseudorandom binary sequences and lattices. JISE J. Inf. Sci. Eng. **30**(5), 1635–1654 (2014)
17. Massey, J.L.: Shift-register synthesis and BCH decoding. IEEE Trans. Inf. Theory **IT-15**, 122–127 (1969)
18. Mauduit, C., Sárközy, A.: On finite pseudorandom binary sequences I: measures of pseudorandomness, the Legendre symbol. Acta Arith. **82**, 365–377 (1997)
19. Mauduit, C., Sárközy, A.: On finite pseudorandom binary sequences II. (The Champernowne, Rudin-Shapiro and Thue-Morse sequences. A further construction.). J. Number Theory **73**, 256–276 (1998)
20. Mauduit, C., Sárközy, A.: Construction of pseudorandom binary sequences by using the multiplicative inverse. Acta Math. Hung. **108**(3), 239–252 (2005)
21. Mérai, L.: Construction of pseudorandom binary sequences over elliptic curves using multiplicative characters. Publ. Math. Debr. **80**(1–2), 199–213 (2012)
22. Mérai, L., Niederreiter, H., Winterhof, A.: Expansion complexity and linear complexity of sequences over finite fields. Cryptogr. Commun. **9**(4), 501–509 (2017)
23. Mérai, L., Winterhof, A.: On the Nth linear complexity of automatic sequences. J. Number Theory (2018). https://doi.org/10.1016/j.jnt.2017.11.008
24. Révész, P.: Random Walk in Random and Non-random Environments. World Scientific, Singapore (1990)
25. Rivat, J., Sárközy, A.: On pseudorandom sequences and their application. In: Ahlswede, R., Bäumer, L., Cai, N., Aydinian, H., Blinovsky, V., Deppe, C., Mashurian, H. (eds.) General Theory of Information Transfer and Combinatorics. LNCS, vol. 4123, pp. 343–361. Springer, Heidelberg (2006). https://doi.org/10.1007/11889342_19
26. Rukhin, A., Soto, J., Nechvata, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dra, J., Vo, S.: A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. NIST Special Publication 800-22, Revision 1.a (2001). http://csrc.nist.gov/groups/ST/toolkit/rng/documentation_software.html
27. Sárközy, A.: On pseudorandomness of families of binary sequences. J. Discret. Appl. Math. **216**(3), 670–676 (2017)
28. Shapiro, H.S.: Extremal problems for polynomials and power series. M.S. thesis, M.I.T. (1951)
29. Tóth, V.: Collision and avalanche effect in families of pseudorandom binary sequences. Period. Math. Hungar. **59**, 1–8 (2009)
30. Washington, L.C.: Elliptic Curves: Number Theory and Cryptography, 2nd edn. Chapman & Hall/CRC Press, Boca Raton (2008)

On the Cross-Combined Measure of Families of Binary Lattices and Sequences

Katalin Gyarmati(✉)

MTA-ELTE Geometric and Algebraic Combinatorics Research Group,
Department of Algebra and Number Theory, Institute of Mathematics,
ELTE Eötvös Loránd University, Pázmány Péter Sétány 1/C,
Budapest 1117, Hungary
gykati@cs.elte.hu

Abstract. The cross-combined measure (which is a natural extension of the cross-correlation measure) is introduced and important constructions of large families of binary lattices with optimal or nearly optimal cross-combined measures are presented. These results are also strongly related to the one-dimensional case: An easy method is shown obtaining strong constructions of families of binary sequences with nearly optimal cross-correlation measures based on the previous constructions of families of lattices. The important feature of this result is that so far there exists only one type of construction of *very large* families of binary sequences with small cross-correlation measure, and this only type of construction was based on one-variable irreducible polynomials. However there are relatively fast algorithms to construct one-variable irreducible polynomials, still in certain applications these algorithms are too complicated or are not fast enough, thus it became necessary to show other types of constructions where the generation of sequences is much faster. Using binary lattices based on two-variable irreducible polynomials this problem can be avoided. (Since, contrary to one-variable polynomials, using the Schöneman-Eisenstein criteria it is possible to generate two-variable irreducible polynomials over $\mathbb{F}_p$ easily and very fast.)

Keywords: Pseudorandom · Cross-combined · Cross-correlation
Binary lattices · Binary sequences

2010 Mathematics Subject Classification: Primary: 11K45

1 Introduction

Pseudorandom binary sequences and lattices have many applications in cryptography. One of the main applications is the famous Vernam-cipher encrypting algorithm, where pseudorandom binary sequences are used as key-streams. If in

Research partially supported by Hungarian National Research Development and Innovation Funds NK 104183 and K 119528.

J. Kaczorowski et al. (Eds.): NuTMiC 2017, LNCS 10737, pp. 217–238, 2018.
https://doi.org/10.1007/978-3-319-76620-1_13

place of a text we would like to encrypt an image by Vernam cipher, then the keystream should be a pseudorandom binary lattice in place of a binary sequence. In the present paper I will study large families of binary sequences and lattices and I will extend an important family measure, the cross-correlation measure from families of binary sequences to family of binary lattices.

1.1 Large Families of Pseudorandom Binary Sequences

The constructive and quantitative study of pseudorandomness started by the work of Mauduit and Sárközy [30]. They introduced the following pseudorandom measures in order to study the pseudorandom properties of *finite* binary sequences:

Definition 1. *For a binary sequence* $E_N = (e_1, \dots, e_N) \in \{-1, +1\}^N$ *of length* N*, write* $U(E_N, t, a, b) = \sum_{j=0}^{t} e_{a+jb}$*. Then the well-distribution measure of* E_N *is defined as*

$$W(E_N) = \max_{a,b,t} \left| \sum_{j=0}^{t} e_{a+jb} \right|,$$

where the maximum is taken over all a, b, t *such that* $a, b, t \in \mathbb{N}$ *and* $1 \le a \le a + tb \le N$*.*

In order to study certain connections of between different elements of the sequence Mauduit and Sárközy [30] introduced the correlation measure:

Definition 2. *For a binary sequence* $E_N = (e_1, \dots, e_N) \in \{-1, +1\}^N$ *of length* N*, and for* $D = (d_1, \dots, d_\ell)$ *with non-negative integers* $0 \le d_1 < \dots < d_\ell$*, write* $V(E_N, M, D) = \sum_{n=1}^{M} e_{n+d_1} \dots e_{n+d_\ell}$*. Then the correlation measure of order* ℓ *of* E_N *is defined as*

$$C_\ell(E_N) = \max_{M,D} \left| \sum_{n=1}^{M} e_{n+d_1} \dots e_{n+d_\ell} \right|,$$

where the maximum is taken over all $D = (d_1, \dots, d_\ell)$ *and* M *such that* $0 \le d_1 < \dots < d_\ell < M + d_\ell \le N$*.*

In [7] Cassaigne et al. formulated the following principle: "The sequence E_N is considered a "good" pseudorandom sequence if these measures $W(E_N)$ and $C_\ell(E_N)$ (at least for "small" ℓ) are "small"." This principle was justified by Cassaigne et al. [8]. They proved that for the majority of the sequences $E_N \in \{-1, +1\}^N$ the measures $W(E_N)$ and $C_\ell(E_N)$ are around $N^{1/2}$ (up to some logarithmic factors). Later Alon et al. [4] improved on these bounds.

It is also important that we will be able to present constructions for which these pseudorandom measures are small. First Mauduit and Sárközy [30] studied the following construction:

Construction A. *Let p be a prime number, $N=p-1$ and define the Legendre-sequence $E_N=(e_1,e_2,\dots,e_N)\in\{-1,+1\}^N$ by*

$$e_n=\left(\frac{n}{p}\right),$$

where $\left(\frac{\cdot}{p}\right)$ denotes the Legendre symbol.

Then by Theorem 1 in [30] for the sequence E_N defined in Construction A we have $W(E_N)\ll N^{1/2}\log N$ and $C_\ell(E_N)\ll N^{1/2}\log N$.

After their first paper [30] on pseudorandomness, Mauduit and Sárközy continued it by a series of papers and later many people continued the work started by them. Since then numerous constructions have been given by several authors.

First for fixed N most constructions produced only a single sequence of length N, however, in many applications one needs many pseudorandom binary sequences. In 2001 Hoffstein and Liemann [27] succeeded in constructing large families of pseudorandom binary sequences based on the Legendre symbol, but they did not prove anything on its pseudorandom properties. Their construction was the following:

Construction B. *Let $K\in\mathbb{N}$, p be a prime number, and denote by $\mathcal{P}_{\le K}$ the set of monic polynomials $f(x)\in\mathbb{F}_p[x]$ of degree k, where $0<k\le K$. For $f\in\mathcal{P}_{\le K}$ define the binary sequence $E_p(f)=(e_1,\dots,e_p)$ by*

$$e_n=\begin{cases}\left(\frac{f(n)}{p}\right) & \text{for } (f(n),p)=1,\\ +1 & \text{for } p\mid f(n).\end{cases}\tag{1.1}$$

Let $\mathcal{F}_{\le K,Legendre}=\{E_p(f):\ f\in\mathcal{P}_{\le K}\}$.

Clearly $\mathcal{F}_{\le K,Legendre}$ is a large family of pseudorandom binary sequences. Goubin et al. [14] proved that, under some not too restrictive conditions on the polynomials f, the sequences $E_p(f)$ have strong pseudorandom properties:

Theorem A. *Let p, $\mathcal{P}_{\le K}$ and $\mathcal{F}_{\le K,Legendre}$ be defined as in Construction B and for $f\in\mathcal{P}_{\le K}$ define $E_p=E_p(f)\in\mathcal{F}_{\le K,Legendre}$ by (1.1). Suppose that f has no multiple root in $\overline{\mathbb{F}}_p$ and denote by k the degree of f. Then*

$$W(E_p)\le 10kp^{1/2}\log p.$$

Moreover, assume that for $\ell\in\mathbb{N}$ one of the following assumptions holds:

(i) $\ell=2$;
(ii) $\ell<p$ *and 2 is a primitive root modulo* p;
(iii) $(4k)^\ell<p$.

Then we also have

$$C_\ell(E_p)\le 10k\ell p^{1/2}\log p.$$

We remark that several important a posteriori tests (indicated by the 1.4-sts. package of the National Institute of Standards and Technology) were checked by Rivat and Sárközy [39] by computer for many sequences generated by Construction B. In each cases they obtained that the sequence passes all these tests. This work was continued by Mérai et al. [36]. After the construction in Theorem A many other constructions of large families of pseudorandom sequences have been given by several authors.

Although many constructions exist, Construction B is one of the best: we have optimally good bounds for the pseudorandom measures and the elements of the sequences can be generated fast. In these constructions it is guaranteed that the individual sequences belonging to the family possess strong pseudorandom properties. However, in many applications it is not enough to know this; it can be much more important to know that the given family has a "rich", "complex" structure, there are many "independent" sequences in it. In order to handle this requirement Ahlswede et al. [1] (see also [2,3,16,32]) introduced the notion of *family complexity* or briefly *f-complexity* (which can be especially useful in cryptography):

Definition 3. *The f-complexity $\Gamma(\mathcal{F})$ of a family $\mathcal{F}$ of binary sequences $E_N \in \{-1,+1\}^N$ is defined as the greatest integer j so that for any* specification

$$e_{i_1} = \varepsilon_1, \dots, e_{i_j} = \varepsilon_j \ (1 \le i_1 < \dots < i_j \le N)$$

(with $\varepsilon_1, \dots, \varepsilon_j \in \{-1,+1\}$) there is at least one $E_N = (e_1, \dots, e_N) \in \mathcal{F}$ which satisfies it. The f-complexity of $\mathcal{F}$ is denoted by $\Gamma(\mathcal{F})$. (If there is no $j \in \mathbb{N}$ with the property above then we set $\Gamma(\mathcal{F}) = 0$.)

Later other properties of large families were studied and other family measures were introduced, see e.g. *collision free* [6,33,40,41], *avalanche effect* or a variant of Hamming-distance called in the case of pseudorandom binary sequences as *distance-minimum* [6,11,29,40,41]. These measures have multi-dimensional analogues (see the papers [19,20]) and in Sect. 1.2 these multi-dimensional versions of family measures will be presented.

In Sect. 3 of this paper I will introduce and focus on a new very general measure, the *cross-combined measure*. This new measure will be a natural extension of the one-dimensional cross-correlation measure defined by Gyarmati et al. in [21]:

Definition 4. *Let $N \in \mathbb{N}$, $\ell \in \mathbb{N}$, and for any ℓ binary sequences $E_N^{(1)}, \dots, E_N^{(\ell)}$ with*

$$E_N^{(i)} = \left(e_1^{(i)}, \dots, e_N^{(i)}\right) \in \{-1,+1\}^N \ (\text{for } i = 1, 2, \dots, \ell)$$

and any $M \in \mathbb{N}$ and ℓ-tuple $D = (d_1, \dots, d_\ell)$ of non-negative integers with $0 \le d_1 \le \dots \le d_\ell < M + d_\ell \le N$, write

$$V_\ell\left(E_N^{(1)}, \dots, E_N^{(\ell)}, M, D\right) = \sum_{n=1}^{M} e_{n+d_1}^{(1)} \cdots e_{n+d_\ell}^{(\ell)}$$

Let

$$\widetilde{C}_\ell\left(E_N^{(1)},\ldots,E_N^{(\ell)}\right)=\max_{M,D}\left|V_\ell\left(E_N^{(1)},\ldots,E_N^{(\ell)},M,D\right)\right|$$

where the maximum is taken over all $D=(d_1,\ldots,d_\ell)$ *and* $M\in\mathbb{N}$ *satisfying* $0\le d_1\le\cdots\le d_\ell<M+d_\ell\le N$ *with the additional restriction that if* $E_N^{(i)}=E_N^{(j)}$ *for some* $i\ne j$*, then we must not have* $d_i=d_j$*. Then the* cross-correlation measure *of order* ℓ *of the family* $\mathcal{F}$ *of binary sequences* $E_N\in\{-1,+1\}^N$ *is defined as*

$$\Phi_\ell(\mathcal{F})=\max\widetilde{C}_\ell\left(E_N^{(1)},\ldots,E_N^{(\ell)}\right)$$

where the maximum is taken over all ℓ*-tuples of binary sequences* $\left(E_N^{(1)},\ldots,E_N^{(\ell)}\right)$ *with* $E_N^{(i)}\in\mathcal{F}$ *for* $i=1,\ldots,\ell$*.*

(Note that other cross-correlation type quantities also occur in [5,13,15].)

In [21] jointly with Mauduit and Sárközy we also studied the main properties and connections of cross-correlation measure to other family measures. Later Mérai studied the average behaviour of the cross-correlation measure. Among others he proved that usually the cross-correlation measure Φ_ℓ of a family of binary lattices $\eta:\ I_N^n\to\{-1,+1\}$ is between two constant factors of $N^{1/2}(\log N)^{1/2}$. For more details see [34,35].

The goal of the present paper is to extend this measure to the multi-dimensional case. The multi-dimensional cross-combined measure will have all advantages of the one-dimensional cross-correlation measure.

1.2 Large Families of Binary Lattices

Before introducing the definition of the multi-dimensional cross-combined measure we will need to present the standard terminology of the multi-dimensional theory of pseudorandomness. This will follow in the next section. In [28] Hubert et al. extended this theory of pseudorandomness to n dimensions.

Denote by I_N^n the set of n-dimensional vectors whose coordinates are integers between 0 and $N-1$:

$$I_N^n=\{\mathbf{x}=(x_1,\ldots,x_n):\ x_i\in\{0,1,\ldots,N-1\}\}.$$

This set is called an *n-dimensional N-lattice* or briefly an *N-lattice*. In [25] this definition was extended to more general lattices in the following way: Let $\mathbf{u_1},\mathbf{u_2},\ldots,\mathbf{u_n}$ be n linearly independent n-dimensional vectors over the field of the real numbers such that the i-th coordinate of $\mathbf{u_i}$ is a positive integer and the other coordinates of $\mathbf{u_i}$ are 0, so that $\mathbf{u_i}$ is of the form $(0,\ldots,0,z_i,0,\ldots,0)$ (with $z_i\in\mathbb{N}$). Let $t_1,t_2,\ldots,t_n$ be integers with $0\le t_1,t_2,\ldots,t_n<N$. Then we call the set

$$B_N^n=\{\mathbf{x}=x_1\mathbf{u_1}+\cdots+x_n\mathbf{u_n}:x_i\in\mathbb{N}\cup\{0\},0\le x_i\left|\mathbf{u_i}\right|\le t_i(<N)\\ \text{for } i=1,\ldots,n\} \tag{1.2}$$

an *n-dimensional box N-lattice* or briefly a *box N-lattice*.

In [28] the definition of binary sequences was extended to more dimensions by considering functions of type

$$\eta(\mathbf{x}):\ I_N^n \to \{-1,+1\}.$$

If $\mathbf{x} = (x_1, \dots, x_n)$ so that $\eta(\mathbf{x}) = \eta((x_1, \dots, x_n))$, then we will simplify the notation slightly by writing $\eta(\mathbf{x}) = \eta(x_1, \dots, x_n)$. Such a function can be visualized as the lattice points of the N-lattice replaced by the two symbols + and −, thus they are called *binary N-lattices.*

In [28] Hubert et al. introduced the following measures of pseudorandomness of binary lattices (here we will present the definition in the same slightly modified but equivalent form as in [25]):

Definition 5. *Let $\eta : I_N^n \to \{-1,+1\}$ be a binary lattice. Define the combined pseudorandom measure of order ℓ of η by*

$$Q_\ell(\eta) = \max_{B,\mathbf{d_1},\dots,\mathbf{d}_\ell} \left| \sum_{\mathbf{x}\in B} \eta(\mathbf{x}+\mathbf{d_1}) \cdots \eta(\mathbf{x}+\mathbf{d}_\ell) \right|,$$

where the maximum is taken over all distinct $\mathbf{d_1}, \dots, \mathbf{d}_\ell \in I_N^n$ and all box N-lattices B such that $B + \mathbf{d_1}, \dots, B + \mathbf{d}_\ell \subseteq I_N^n$.

Note that in the one-dimensional special case $Q_1(\eta)$ is the well-distribution measure W.

η is said to have strong pseudorandom properties, or briefly, it is considered as a good pseudorandom binary lattice if at least for small ℓ's and large N the measures $Q_\ell(\eta)$'s are small (much smaller, than the trivial upper bound N^n). This terminology is justified by the fact that, as it was proved in [28], for a truly random binary lattice defined on I_N^n and for fixed ℓ the measure $Q_\ell(\eta)$ is “small”, more precisely, it is less than $N^{n/2}$ multiplied by a logarithmic factor. As in the one-dimensional case, many papers have been written on pseudorandomness of binary lattices, for further references see e.g. [22–24].

In the application (similarly to the one-dimensional case) it is important that a large family $\mathcal{G}$ of binary lattices has a “rich”, “complex” structure, there are many “independent” sequences, resp. lattices in it which are “far apart”. Thus one needs quantitative measures for these properties of families of binary lattices. In case of binary sequences some of these measures were mentioned in Sect. 1.1.

Next few definitions of family measures of binary lattices introduced by Gyarmati et al. in [21] follow:

Definition 6. *If $\mathcal{G}$ is a family of binary lattices η is of the form*

$$\mathcal{G} = \mathcal{G}(\mathcal{S}) = \{\eta_s :\ s \in \mathcal{S}\}, \tag{1.3}$$

and for any $s \in \mathcal{S}$ changing any element of s changes large proportion of the elements of $\eta_s :\ I_N^n \to \{-1,+1\}$, then we speak about avalanche effect*, and we say that $\mathcal{F} = \mathcal{F}(\mathcal{S})$ possesses the* avalanche property*. If for any $s \in \mathcal{S}$, $s' \in \mathcal{S}$, $s \neq s'$ at least $\left(\frac{1}{2} - o(1)\right) N^n$ elements of η_s and $\eta_{s'}$ are different, then $\mathcal{F}$ is said to possess the* strict avalanche property*.*

Definition 7. *If $N \in \mathbb{N}$, $n \in \mathbb{N}$, $\eta :\ I_N^n \rightarrow \{-1,+1\}$ and $\eta' :\ I_N^n \rightarrow \{-1,+1\}$, then the distance $d(\eta,\eta')$ between η and η' is defined by*

$$d(\eta,\eta') = |\{(x_1,x_2,\dots,x_n) : (x_1,\dots,x_n) \in \mathbb{I}_N^n,\\ \eta(x_1,\dots,x_n) \neq \eta'(x_1,\dots,x_n)\}|.$$

If $\mathcal{G}$ is a family of binary lattices, then the distance minimum *$m(\mathcal{G})$ is defined by*

$$m(\mathcal{G}) = \min_{\substack{\eta,\eta' \in \mathcal{G}\\ \eta \neq \eta'}} d(\eta,\eta').$$

So that $\mathcal{G}$ is collision free if $m(\mathcal{G}) > 0$, and it possesses the strict avalanche property if

$$m(\mathcal{G}) \geq \left(\frac{1}{2} - o(1)\right) N^n. \tag{1.4}$$

2 The Definition of Cross-Combined Measure and Its Connection with other Family Measures

In this Sect. 1 extend the cross-correlation measure to the multi-dimensional case. This new measure will be called cross-combined measure:

Definition 8. *Let $N \in \mathbb{N}$, $\ell \in \mathbb{N}$, and for any ℓ binary sequences $\eta_1,\dots,\eta_\ell$ with*

$$\eta_i :\ I_N^n \rightarrow \{-1,+1\} \quad (i = 1,2,\dots,\ell)$$

and for any B box-lattice of the form (1.2) and ℓ-tuple $D = (\mathbf{d}_1,\dots,\mathbf{d}_\ell)$ with $\mathbf{d}_i \in I_N^n$ $(i = 1,2,\dots,\ell)$ write

$$V_\ell\left(\eta_1,\dots,\eta_\ell,B,D\right) = \sum_{\mathbf{x} \in B} \eta_1(\mathbf{x}+\mathbf{d}_1)\cdots\eta_\ell(\mathbf{x}+\mathbf{d}_\ell). \tag{2.1}$$

Let

$$\widetilde{Q}_\ell\left(\eta_1,\dots,\eta_\ell\right) = \max_{B,D} \left|V_\ell\left(\eta_1,\dots,\eta_\ell,B,D\right)\right| \tag{2.2}$$

where the maximum is taken over all $D = (\mathbf{d}_1,\dots,\mathbf{d}_\ell)$ and B box-lattice satisfying $B+\mathbf{d}_1, B+\mathbf{d}_2,\dots,B+\mathbf{d}_\ell \subseteq I_N^n$ with the additional restriction that if $\eta_i = \eta_j$ for some $i \neq j$, then we must not have $\mathbf{d}_i = \mathbf{d}_j$. Then the cross-combined measure of order ℓ *of the family $\mathcal{G}$ of binary lattices $\eta \in \{-1,+1\}^N$ is defined as*

$$\Phi_\ell(\mathcal{G}) = \max \widetilde{Q}_\ell\left(\eta_1,\dots,\eta_\ell\right) \tag{2.3}$$

where the maximum is taken over all ℓ-tuples of binary lattices $(\eta_1,\dots,\eta_\ell)$ with

$$\eta_i \in \mathcal{G} \text{ for } i = 1,\dots,\ell.$$

By the definition of $\widetilde{Q}_\ell$, we have $\widetilde{Q}_\ell\left(\eta,\dots,\eta\right) = Q_\ell(\eta)$, thus it follows from (2.3) that

Proposition 1. *We have*

$$\Phi_\ell(\mathcal{G}) \geq \max_{\eta\in\mathcal{G}} Q_\ell(\eta).$$

This means that if we have a "good" upper bound for $\Phi_\ell(\mathcal{G})$, then this guarantees that *all lattices in* $\mathcal{G}$ *possess strong pseudorandom properties.*

Next in this Sect. 1 will study the connection of cross-combined measure with other family measures. As a multi-dimensional analog of Proposition 2.2 in [21] now we obtain:

Proposition 2. *If* $N, n \in \mathbb{N}$ *and* $\mathcal{G}$ *is a family of binary lattices* $\eta : I_N^n \rightarrow \{-1,+1\}$, *then for* $\eta_1, \eta_2 \in \mathcal{G}$ *we have*

$$\left| d(\eta_1,\eta_2) - \frac{N^n}{2} \right| \leq \frac{1}{2}\tilde{Q}_2(\eta_1,\eta_2) \leq \frac{1}{2}\Phi_2(\mathcal{G}). \tag{2.4}$$

Proof. Clearly we have

$$d(\eta_1,\eta_2) = \sum_{\mathbf{x}\in I_N^n} \frac{(\eta_1(\mathbf{x}) - \eta_2(\mathbf{x}))^2}{4} = \frac{N^n}{2} - \frac{1}{2}\sum_{\mathbf{x}\in I_N^n} \eta_1(\mathbf{x})\eta_2(\mathbf{x})$$

whence, by (2.1), (2.2) and (2.3),

$$\left| d(\eta_1,\eta_2) - \frac{N^n}{2} \right| = \frac{1}{2}\left| \sum_{\mathbf{x}\in I_N^n} \eta_1(\mathbf{x})\eta_2(\mathbf{x}) \right| \leq \frac{1}{2}\tilde{Q}_2(\eta_1,\eta_2) \leq \Phi_2(\mathcal{G})$$

which proves (2.4).

If the cross-combined measure of order 2 of a family $\mathcal{G}$ of n-dimensional binary lattices is $o(N^n)$ then it follows from Definition 7 and (2.4) that

$$m(\mathcal{G}) = \min_{\substack{\eta,\eta'\in\mathcal{F} \\ \eta\neq\eta'}} d(\eta_1,\eta_2) \geq \frac{N^n}{2} - \frac{1}{2}\Phi_2(\mathcal{G}) = \frac{N^n}{2} - o(N^n)$$

so that (1.4) holds. This proves

Proposition 3. *If* $N, n \in \mathbb{N}$, $\mathcal{G}$ *is a large family of binary lattices* $\eta : I_N^n \rightarrow \{-1,+1\}$ *and* $\Phi_2(\mathcal{G}) = o(N^n)$ *then the family* $\mathcal{G}$ *possesses the strict avalanche property.*

3 Cross-Combined Measure of a Family of Binary Lattices Constructed by Using Quadratic Characters

Mauduit and Sárközy [31] constructed a large family of binary lattices with strong pseudorandom properties by using quadratic characters of finite fields (this construction generalizes the one dimensional constructions in [14,30]). They proved the following theorem:

Theorem B. *Assume that $q = p^n$ is the power of an odd prime, $f(x) \in \mathbb{F}_q[x]$ has degree k with*

$$0 < k < p.$$

Denote the quadratic character of $\mathbb{F}_q$ by γ (setting also $\gamma(0) = 0$). Consider the linear vector space formed by the elements of $\mathbb{F}_q$ over $\mathbb{F}_p$, and let $v_1, \dots, v_n$ be a basis of this vector space (i.e., assume that $v_1, v_2, \dots, v_n$ are linearly independent over $\mathbb{F}_p$). Define the n dimensional binary p-lattice $\eta :\ I_p^n \to \{-1, +1\}$ by

$$\eta(\mathbf{x}) = \eta((x_1, \dots, x_n)) = \begin{cases} \gamma(f(x_1 v_1 + \dots + x_n v_n)) \textit{ for} \\ \qquad\qquad\qquad f(x_1 v_1 + \dots + x_n v_n) \neq 0 \\ +1 \textit{ for } f(x_1 v_1 + \dots + x_n v_n) = 0. \end{cases} \tag{3.1}$$

Assume that $f(x)$ has no multiple zero in $\overline{\mathbb{F}}_q$, $\ell \in \mathbb{N}$ and

$$4^{n(k+\ell)} < p.$$

Then we have

$$Q_\ell(\eta) < k\ell(q^{1/2}(1 + \log p)^n + 2).$$

Indeed this is a combination of Theorems 1 and 2 in [31].

Throughout this section p, n and $q = p^n$ will be fixed. We will denote the construction of Theorem B by $\mathcal{G}_{\le K, quadratic}$:

Construction C. *Denote by $\mathcal{P}_{\le K}$ the set of monic polynomials $f \in \mathbb{F}_q[x]$ with degree $0 < \deg\ f \le K$. Let $\mathcal{G}_{\le K, quadratic}$ denote the family of the binary lattices η defined by (3.1) assigned to polynomials $f \in \mathcal{P}_{\le K}$.*

It is clear that all lattices $\eta \in G_{\le K, quadratic}$ satisfying the conditions of Theorem B possess strong pseudorandom properties.

In order to simplify the notations we will introduce a function $\tau : \mathbb{F}_p^n \to \mathbb{F}_q$. We may assume that I_p^n represents the elements of $\mathbb{F}_p^n$ and thus we may also use τ as a function $\tau :\ I_p^n \to \mathbb{F}_q$. Let $v_1, v_2, \dots, v_n$ be the basis of the vector space $\mathbb{F}_q$ over $\mathbb{F}_p$ defined in Theorem B. (Here $q = p^n$.) For an $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{F}_p^n$ let

$$\tau(\mathbf{x}) = x_1 v_1 + x_2 v_2 + \dots + x_n v_n.$$

Then τ is a bijection. We also have for $\mathbf{a}, \mathbf{b} \in \mathbb{F}_p^n$ that $\tau(\mathbf{a} + \mathbf{b}) = \tau(\mathbf{a}) + \tau(\mathbf{b})$. Then (3.1) in Theorem B can be written in the equivalent form

$$\eta(\mathbf{x}) = \begin{cases} \gamma(f(\tau(\mathbf{x}))) & \text{for } f(\tau(\mathbf{x})) \neq 0 \\ +1 & \text{for } f(\tau(\mathbf{x})) = 0. \end{cases} \tag{3.2}$$

In [19] jointly with Mauduit and Sárközy we proved that the family measure of $\mathcal{G}_{\le K, quadratic}$ is optimal. The distance minimum was also estimated in [19] and If $K < \frac{1}{2} q^{1/2}$, then $\mathcal{G}_{\le K, quadratic}$ is collision free. Moreover if $q \to \infty$, $K = o(q^{1/2})$, then $\mathcal{G}_{\le K, quadratic}$ possesses the strict avalanche property.

Unfortunately, it turned out that for $K \ge 2$, our new measure, the cross-combined measure of $\mathcal{G}_{\le K, quadratic}$ is very bad:

Proposition 4. *For $K \geq 2$ we have $\Phi_3(G_{\leq K,quadratic}) \geq q-2$.*

Proof. Consider the following 3 polynomials: $f_1(x) = x, f_2(x) = x+1, f_3(x) = x(x+1) \in \mathbb{F}_q[x]$. Let η_i be the binary lattice defined by (3.1) with f_i in place of f for $i = 1, 2, 3$. Then using (3.2) we get:

$$\begin{aligned}\Phi_3(\mathcal{G}_{\leq K,quadratic}) &\geq \widetilde{Q}_3(\eta_1,\eta_2,\eta_3) \geq V_3(\eta_1,\eta_2,\eta_3,I_p^n,(0,0,0)) = \sum_{\mathbf{x}\in I_p^n} \eta_1(\mathbf{x})\eta_2(\mathbf{x})\eta_3(\mathbf{x}) \\ &= \sum_{\substack{\tau(\mathbf{x})\in I_p^n \\ \tau(\mathbf{x})(\tau(\mathbf{x})+1)\neq 0}} \gamma(\tau(\mathbf{x}))\gamma(\tau(\mathbf{x})+1)\gamma(\tau(\mathbf{x})(\tau(\mathbf{x})+1)) + \gamma(1) + \gamma(-1) \\ &= \sum_{\substack{y\in\mathbb{F}_q \\ y(y+1)\neq 0}} \gamma(y^2(y+1)^2) + \gamma(1) + \gamma(-1) \geq q-2.\end{aligned}$$

Clearly Proposition 4 can be easily extended to cross-combined measures of higher order.

Thus we need to restrict the family $\mathcal{G}_{\leq K,quadratic}$ to a large subfamily of it such that this subfamily has a good cross-combined measure. In the one-dimensional case jointly with Mauduit and Sárközy [21] we have the following idea:

Construction D. *Consider the set of monic irreducible polynomials of the form $f(x) = x^k + a_{k-2}x^{k-2} + a_{k-3}x^{k-3} + \cdots + a_0$ (so that the coefficient $a_{k-1} = 0$) with degree $0 < k \leq K$ and let $\mathcal{F}_{\leq K,irreducible,Legendre}$ ($\mathcal{F}_{\leq K,Legendre}$) the set of all binary sequences defined by (1.1) where the used monic irreducible polynomial f are in this form.*

Then by [21] the family $\mathcal{F}_{\leq K,irreducible,Legendre}$ has optimal cross-correlation measure:

Theorem C

$$\Phi_\ell(\mathcal{F}_{\leq K,irreducible,Legendre}) \leq 10K\ell p^{1/2}\log p.$$

(This is Theorem 1 in [21]). Here the family $\mathcal{F}_{\leq K,irreducible,Legendre}$ is almost as large as $\mathcal{F}_{\leq K,Legendre}$, and so far this is the only method to construct *very large* family of binary sequences with optimal cross-correlation measure. In Sect. 5 I will show another type of construction of a *very large* family of binary sequences for which the cross-correlation measure is nearly optimal.

Next I return to the cross-combined measure and the multi-dimensional case.

Construction 1. *Let $\mathcal{G}_{\leq K,irreducible,quadratic}$ denote the following subfamily of $\mathcal{G}_{\leq K,quadratic}$: consider those $\eta \in \mathcal{G}_{\leq K,quadratic}$ for which the used polynomials f in (3.1) are monic irreducible and of the form $f(x) = x^k + a_{k-2}x^{k-2} + a_{k-3}x^{k-3} + \cdots + a_0$ (so that the coefficient $a_{k-1} = 0$) with degree $0 < k \leq K$ and let $\mathcal{G}_{\leq K,irreducible,quadratic}$ the set of all binary lattices obtained in this way. Clearly $G_{\leq K,irreducible,quadratic} \subset G_{\leq K,quadratic}$.*

Next I prove

Theorem 1

$$\Phi_\ell(\mathcal{G}_{\le K,irreducible,quadratic}) < K\ell q^{1/2}(\log p+1)^n + 2\ell.$$

Proof. By the definition of cross-combined measure we have that there exist binary lattices $\eta_1,\eta_2,\dots,\eta_\ell \in \mathcal{G}_{\le K,Legendre}$ $D=(\mathbf{d}_1,\dots,\mathbf{d}_\ell)$ ℓ-tuple (where $\mathbf{d}_i \in \mathbb{I}_p^n$) and B box-lattice satisfying $B+\mathbf{d}_1,\dots,B+\mathbf{d}_\ell \subset I_p^n$ with the additional restriction that if $\eta_i=\eta_j$ for some $i\ne j$, then we must not have $\mathbf{d}_i=\mathbf{d}_j$ such that

$$\Phi_\ell(\mathcal{G}_{\le K,irreducible,quadratic}) = |V_\ell(\eta_1,\dots,\eta_\ell,B,D)| = \left|\sum_{\mathbf{x}\in B}\eta_1(\mathbf{x}+\mathbf{d}_1)\cdots\eta_\ell(\mathbf{x}+\mathbf{d}_\ell)\right| \tag{3.3}$$

Clearly by (3.2) there exists monic irreducible polynomials f_i $(i=1,2,\dots,\ell)$ such that all f_i can be written in the form

$$x^k + a_{k-2}x^{k-2} + a_{k-3}x^{k-3} + \cdots + a_1x + a_0 \tag{3.4}$$

for some $0<k\le K$, $a_0,a_1,\dots,a_{k-2}\in\mathbb{F}_q$ (thus the coefficient of $x^{\deg\ f_i-1}$ is always 0) and for the binary lattice η_i $(i=1,2,\dots,\ell)$ we have

$$\eta_i(\mathbf{x}) = \begin{cases}\gamma(f_i(\tau(\mathbf{x}))) & \text{for } f(\tau(\mathbf{x}))\ne 0\\ +1 & \text{for } f_i(\tau(\mathbf{x}))=0.\end{cases} \tag{3.5}$$

By (3.3), (3.5) and since irreducible polynomials may have only one zero (and only in the case of linear polynomials) we have

$$\begin{aligned}\Phi_\ell(\mathcal{G}_{\le K,irreducible,quadratic}) &\le \left|\sum_{\mathbf{x}\in B}\gamma(f_1(\tau(\mathbf{x}+\mathbf{d}_1)))\cdots\gamma(f_\ell(\tau(\mathbf{x}+\mathbf{d}_\ell)))\right| + 2\ell\\ &= \left|\sum_{\mathbf{y}\in\tau(B)}\gamma(f_1(y+\tau(\mathbf{d}_1))\cdots f_\ell(y+\tau(\mathbf{d}_\ell)))\right| + 2\ell\end{aligned} \tag{3.6}$$

where the set $\tau(B)$ is defined by $\tau(B) \overset{\text{def}}{=} \{\tau(\mathbf{x}) : \mathbf{x}\in B\}$. Next we use Winterhof's Lemma [43]:

Lemma 1. *Let χ be a non-trivial multiplicative character of order d over $\mathbb{F}_q$ and $g\in\mathbb{F}_q[x]$ of a polynomial with s distinct zeros in $\overline{\mathbb{F}}_q$ and which is not of the form $ch(x)^d$ with $c\in\mathbb{F}_q$ and $h(x)\in\mathbb{F}_q[x]$. Then for $1\le t_i<p$ $(i=1,2,\dots,n)$ and for a set C defined by*

$$C = C(t_1,t_2,\dots,t_n) = \{x_1v_1+x_2v_2+\cdots+x_nv_n :\ 0\le x_i\le t_i \text{ for } i=1,2,\dots,n\} \tag{3.7}$$

we have

$$\left|\sum_{y\in C}\chi(g(x))\right| < sq^{1/2}(1+\log p)^n \le \deg\ g\ q^{1/2}(1+\log p)^n.$$

This is Theorem 2 in [43]. (The main tool in the proof is Weil's theorem [42].)

Clearly the set $\tau(B)$ is a set of the form (3.7). We will use Lemma 1 with the quadratic character γ in place of χ and with the polynomial $g(y) \overset{\text{def}}{=} f_1(y+\tau(\mathbf{d}_1))\cdots f_\ell(y+\tau(\mathbf{d}_\ell))$. In order to use this lemma first we need to show $g(y)$ is not of the form $ch(y)^2$. If for some $1\le i<j\le \ell$ we have $f_i(y)\ne f_j(y)$, then

$$f_i(y+\tau(\mathbf{d}_i)) \ne f_j(y+\tau(\mathbf{d}_j)) \tag{3.8}$$

also holds since if

$$f_i(y+\tau(\mathbf{d}_i)) = f_j(y+\tau(\mathbf{d}_j)), \tag{3.9}$$

then deg f_i = deg $f_j = k$. Then the coefficient of the term x^{k-1} are the same both in $f_i(y+\tau(\mathbf{d}_i))$ and $f_j(y+\tau(\mathbf{d}_j))$ and by the special form of these polynomials (see (3.4)) we also have that (3.9) holds only if $\tau(\mathbf{d}_i)=\tau(\mathbf{d}_j)$. Since τ is a bijection then we have $\mathbf{d}_i=\mathbf{d}_j$. Writing this in (3.9) we get the polynomials f_i and f_j are the same, but then the lattices η_i and η_j are also the same. In the definition of cross-combined measure we have the additional restriction that if $\eta_i=\eta_j$, then we must have $\mathbf{d}_i\ne\mathbf{d}_j$, which is a contradiction. Thus we proved (3.8). By (3.8) we get $g(y)$ is a product of different irreducible polynomials thus it cannot be of the form $ch(y)^2$. So we may use Lemma 1 for the character sum in (3.6) and we obtain

$$\varPhi_\ell(\mathcal{G}_{\le K, irreducible, quadratic}) < K\ell q^{1/2}(\log p+1)^n + 2\ell$$

which was to be proved.

4 Cross-Combined Measure of a Family of Binary Lattices Constructed by Using Legendre Symbol

Next I study a natural construction of families of two-dimensional binary lattices based on Legendre symbol introduced by Gyarmati et al. in [25,26]. In the case of this construction we will have slightly weaker upper bounds both for the pseudorandom measures of the binary lattices and for the cross-combined measure of the family than the optimal. The reason for this is that in order to estimate the necessary character sums we would need the two-dimensional analogue of Weil's theorem [42]. The multi-dimensional analogue of Weil's theorem was studied by Delinge [9,10], and later Fouvry and Katz [12] simplified the requirements. Still an inconvenient assumption of nonsingularity is required in order to reach the optimal bounds, which in our cases are not applicable. However in the case of this construction we have weaker upper bounds for the pseudorandom measures, on the other hand the lattices of the family can be generated very fast, which makes the implementation easy. Our starting point is the following construction defined by Sárközy, Stewart and myself in [25]:

Construction E. *Let p be an odd prime. Denote by $\mathcal{R}_{\leq K}$ the set of polynomials $f \in \mathbb{F}_p[x_1, x_2]$ with degree $0 < \deg f \leq K$. Let $\mathcal{G}_{\leq K, Legendre}$ denote the family all binary lattices $\eta : I_p^2 \to \{-1, +1\}$ which can be written in the form defined by*

$$\eta(x_1, x_2) = \begin{cases} \left(\frac{f(x_1,x_2)}{p}\right) & \text{if } (f(x_1,x_2), p) = 1, \\ 1 & \text{if } p \mid f(x_1, x_2). \end{cases} \tag{4.1}$$

with a polynomial $f \in R_{\leq K}$.

In [25,26] jointly with Sárközy and Stewart we proved that under some not too restrictive conditions on the polynomial f or the prime p we have:

$$Q_\ell(\eta) \leq 11k\ell p^{3/2} \log p.$$

Similarly to Sect. 3, it turned out that for $K \geq 2$, the cross-combined measure of $\mathcal{G}_{\leq K, Legendre}$ is very bad:

Proposition 5. *For $K \geq 2$ we have $\Phi_3(G_{\leq K, Legendre}) \geq p^2 - 2$.*

The proof of Proposition 5 is similar to Proposition 4 thus we leave the details to the reader. Thus again we need to restrict $G_{\leq K, Legendre}$ to a proper large subfamily which has good cross-correlation measure. Again we have the idea using irreducible polynomials. Here we need the following special case of Theorem 1 in [25]:

Theorem D. *Let p be an odd prime, $f \in \mathbb{F}_p[x_1, x_2]$ be an irreducible polynomial in two variables of degree k. Define $\eta : I_p^2 \to \{-1, +1\}$ by (4.1). If $f(x_1, x_2)$ is not of the form*

$$f(x_1, x_2) = \varphi(\gamma x_1 + \delta x_2) \tag{4.2}$$

with $\gamma, \delta \in \mathbb{F}_p$ and $\varphi \in \mathbb{F}_p[x]$.

Then for the binary p-lattice defined by (4.1) we have

$$Q_\ell(\eta) \leq 11k\ell p^{3/2} \log p.$$

Is the condition that $f(x_1, x_2)$ is not of the form (4.2) necessary? The answer is affirmative since by Theorem 2 in [26] we have

Theorem E. *Let p be an odd prime, $f \in \mathbb{F}_p[x_1, x_2]$ be a polynomial in two variables of degree k. Define $\eta : I_p^2 \to \{-1, +1\}$ by (4.1). If $f(x_1, x_2)$ is of the form $f(x_1, x_2) = \varphi(\gamma x_1 + \delta x_2)$ with some $\gamma, \delta \in \mathbb{F}_p$ and $\varphi \in \mathbb{F}_p[x]$, then for the binary p-lattice defined by (4.1) we have*

$$Q_2(\eta) \geq p^2 - 4p^{3/2} - 8kp.$$

By Theorems D and E we have the idea of studying the following subfamily of $\mathcal{G}_{\leq K, Legendre}$:

Construction 2. *Let* $\mathcal{G}_{\leq K,irreducible,Legendre}$ *denote the following subfamily of* $\mathcal{G}_{\leq K,Legendre}$*: consider those* $\eta \in \mathcal{G}_{\leq K,quadratic}$ *for which the used polynomials* f *in* (4.1) *are irreducible and they are of the form*

$$f(x_1,x_2) = x_1^k + x_2^{k-1} + s(x_1,x_2) \tag{4.3}$$

where $\deg s \leq k-3$*. Clearly* $G_{\leq K,irreducible,quadratic} \subset G_{\leq K,quadratic}$*.*

The cross-combined measure of this family is relatively small:

Theorem 2

$$\Phi_\ell(\mathcal{G}_{\leq K,irreducible,Legendre}) < 11K\ell p^{3/2}\log p.$$

Proof. The theorem is trivial for the cases $p \leq 7$ and $p \leq K$, thus throughout the proof we may assume $p \geq 11$ and $K < p$. Let $\eta_1,\eta_2,\dots,\eta_\ell \in \mathcal{G}_{\leq K,irreducible,Legendre}$ binary lattices, $D = (\mathbf{d}_1,\dots,\mathbf{d}_\ell)$ and B box-lattice satisfying $B+\mathbf{d}_1,\dots,B+\mathbf{d}_\ell \subset I_p^n$ with the additional restriction that if $\eta_i = \eta_j$ for some $i \neq j$, then we must not have $\mathbf{d}_i = \mathbf{d}_j$ for which we have

$$\Phi_\ell(\mathcal{G}_{\leq K,irreducible,Legendre}) = |V_\ell(\eta_1,\dots,\eta_\ell,B,D)| = \left|\sum_{\mathbf{x}\in B}\eta_1(\mathbf{x}+\mathbf{d}_1)\cdots\eta_\ell(\mathbf{x}+\mathbf{d}_\ell)\right|$$

Clearly by (4.1) there exists irreducible polynomials f_i $(i=1,2,\dots,\ell)$ which are of the form (4.3) and

$$\eta_i(\mathbf{x}) = \begin{cases} \left(\frac{f_i(\mathbf{x})}{p}\right) & \text{for } f(\mathbf{x}) \neq 0 \\ +1 & \text{for } f_i(\mathbf{x}) = 0. \end{cases}$$

Since for fixed x_1 the polynomial $f(\mathbf{x}) = f(x_1,x_2)$ has at most K zeros in x_2, we have $f(\mathbf{x})$ has at most Kp zeros in $\mathbf{x}$. Then similarly to (3.6) we get

$$\Phi_\ell(\mathcal{G}_{\leq K,irreducible,Legendre}) \leq \left|\sum_{\mathbf{x}\in B}\left(\frac{f_1(\mathbf{x}+\mathbf{d}_1)\cdots f_\ell(\mathbf{x}+\mathbf{d}_\ell)}{p}\right)\right| + 2K\ell p. \tag{4.4}$$

Here we would like to use the following lemma which was proved in [25]:

Lemma 2. *Let* $p \geq 5$ *be a prime and* χ *be a multiplicative character of order* d*. Suppose that* $h(x_1,x_2) \in \mathbb{F}_p[x_1,x_2]$ *is not of the form* $cg(x_1,x_2)^d$ *with* $c \in \mathbb{F}_p$*,* $g(x_1,x_2) \in \mathbb{F}_p[x_1,x_2]$*. Let* k *be the degree of* $h(x_1,x_2)$*. Then we have*

$$\sum_{\mathbf{x}\in B}\chi\left(h(\mathbf{x})\right) < 10kp^{3/2}\log p$$

for every 2 *dimensional box* p*-lattice* $B \subseteq I_p^2$*.*

In order to prove Theorem 2 first we mention that the irreducible polynomials $f_1(\mathbf{x}+\mathbf{d_1}),\dots,f_\ell(\mathbf{x}+\mathbf{d}_\ell)$ are different. Indeed if

$$f_i(\mathbf{x}+\mathbf{d}_i)=f_j(\mathbf{x}+\mathbf{d}_j), \tag{4.5}$$

then writing $\mathbf{d}=\mathbf{d}_j-\mathbf{d}_i$ we get

$$f_i(\mathbf{x}+\mathbf{d})=f_j(\mathbf{x}).$$

Write $\mathbf{x}=(x_1,x_2)$ and $\mathbf{d}=(d_1,d_2)$. Then using that f is of the form (4.3) we get

$$(x_1+d_1)^k+(x_2+d_2)^{k-1}+s(x_1+d_1,x_2+d_2)=x_1^k+x_2^{k-1}+s(x_1,x_2). \tag{4.6}$$

Here the coefficients of x_1^{k-1} and x_2^{k-2} must be identical in the left and right-hand side of (4.6) but since $\deg s\le k-3$ this is possible only for $d_1=0$ and $d_2=0$. Then $\mathbf{d}=(d_1,d_2)=\mathbf{0}$, so $\mathbf{d}_i=\mathbf{d}_j$. Then from (4.5) we get $f_i=f_j$ and thus $\eta_i=\eta_j$. But in the beginning of the proof we assumed that if $\eta_i=\eta_j$ for some $i\ne j$, then we must not have $\mathbf{d}_i=\mathbf{d}_j$ which is a contradiction.

Since the irreducible polynomials $f_1(\mathbf{x}+\mathbf{d_1}),\dots,f_\ell(\mathbf{x}+\mathbf{d}_\ell)$ are different, the product polynomial $g(\mathbf{x})=f_1(\mathbf{x}+\mathbf{d_1})\cdots f_\ell(\mathbf{x}+\mathbf{d}_\ell)$ cannot be of the form $cg(\mathbf{x})^2$. By (4.4) and using Lemma 2 we get

$$\Phi_\ell(\mathcal{G}_{\le K,irreducible,Legendre})<10K\ell p^{3/2}\log p+2K\ell p<11K\ell p^{3/2}\log p.$$

which was to be proved.

Corollary 1. *For all subfamily $\mathcal{G}_0$ of $\mathcal{G}_{\le K,irreducible,Legendre}$ we have*

$$\Phi_\ell(\mathcal{G}_0)<11K\ell p^{3/2}\log p.$$

This corollary is trivial and at first sight not very interesting. The important feature of it is that while the construction of one-variable irreducible polynomials is slightly complicated (see e.g. the Handbook of Finite Fields [37]), then there is an easy way to construct two-variable irreducible polynomials over finite fields using the Schöneman-Eisenstein criteria:

Lemma 3. *Let $k\ge 5$ and $f\in\mathbb{F}_p[x_1,x_2]$ be a polynomial of the form*

$$f(x_1,x_2)=x_1^k+x_1x_2g(x_1,x_2)+x_2h(x_2) \tag{4.7}$$

with $g\in\mathbb{F}_p[x_1,x_2]$, $\deg\ g\le k-5$, $h\in\mathbb{F}_p[x_2]$ is a monic polynomial such that $\deg\ h=k-2$, $x_2\nmid h(x_2)$ and the coefficient of x_2^{k-3} in $h(x_2)$ is 0.

Then $f(x_1,x_2)$ is irreducible and it is of the form (4.3).

It is clear that the polynomial f is of the form (4.3). The irreducibility of the polynomial f was also stated and used in [26] in the proof of Theorem 3 (there the conditions on g and h were slightly weaker than here, e.g. it is enough to assume $\deg g\le k-3$ etc.). In [26] the proof of the irreducibility of f was deduced from Theorem 282 in the book of Rédei [38].

Using polynomials of form (4.7) we can construct a large family of binary lattices such that its implementation is easy and fast:

Construction 3. *Let* $\mathcal{G}_{\leq K,Sch-Eis,Legendre}$ *denote the following subfamily of* $\mathcal{G}_{\leq K,irreducible,Legendre}$*: consider those* $\eta \in \mathcal{G}_{\leq K,quadratic}$ *for which the used polynomials* f *in* (4.1) *are of the form* (4.7)*. Clearly* $\mathcal{G}_{\leq K,Sch-Eis,Legendre} \subset G_{\leq K,irreducible,Legendre} \subset G_{\leq K,quadratic}$.

Using Corollary 1 we immediately get.

Corollary 2

$$\varPhi_\ell(\mathcal{G}_{\leq K,Sch-Eis,Legendre}) < 11K\ell p^{3/2}\log p.$$

Thus the family $\mathcal{G}_{\leq K,Sch-Eis,Legendre}$ has nearly optimal cross-combined measure, clearly is is very large (it contains more than $p^{K(K-1)/2}$ different binary lattices) and the binary lattices in it can be generated easily and very fast. In the next section we will show how it is possible to generate a very large families of pseudorandom binary sequences with optimal or nearly optimal cross-correlation measure using these families of binary lattices.

5 Constructions of Binary Sequences with Optimal or Nearly Optimal Cross-Correlation Measures Based on Lattices and Multi-dimensional Theory

In [18] jointly with Mauduit and Sárközy we reduced the two dimensional case to the one-dimensional one by the following way: To any 2-dimensional binary N-lattice

$$\eta : \ I_N^2 \rightarrow \{-1,+1\} \tag{5.1}$$

we may assign a unique binary sequence $E_{N^2} = E_{N^2}(\eta) = \{e_1, e_2, \dots, e_{N^2}\} \in \{-1,+1\}^N$ by taking the first (from the bottom) row of the lattice then we continue the binary sequence by taking the second row of the lattice, then the third row follows, etc.; in general, we set

$$e_{iN+j} = \eta((j-1,i)) \text{ for } i = 0,1,\dots,N^2-1, \ j = 1,2,\dots,N. \tag{5.2}$$

We will denote the sequence defined by this way by $\overline{E}(\eta)$. In [18] with Mauduit and Sárközy we asked if it is true that if $\overline{E}(\eta)$ is a “good” pseudorandom binary *sequence* then η is a “good” pseudorandom 2-dimensional lattice? The answer to this question is negative; in [18] it is showed that it may occur that the pseudorandom measures of the sequence $E_{N^2}(\eta)$ are small, however, the corresponding pseudorandom measures of the lattice η are large. On the other hand, in [17] I proved the following: if the lattice η has small combined measure, then the corresponding $\overline{E}(\eta)$ sequence has small correlation measure as well.

Theorem F. *Let* η *be an arbitrary binary lattice. Then*

$$C_\ell(\overline{E}(\eta)) \leq (\ell+2)Q_\ell(\eta).$$

Here I generalize this result to families of binary sequences and lattices and the cross-correlation and cross-combined measure.

Definition 9. *Let $\mathcal{F}$ be a two-dimensional family of binary lattices $\eta : I_N^2 \rightarrow \{-1, +1\}$. Define the family $\overline{E}(\mathcal{G})$ of binary sequences of length N^2 by*

$$\overline{E}(\mathcal{G}) \overset{\text{def}}{=} \{\overline{E}(\eta) : \eta \in \mathcal{G}\}.$$

Next I will prove that if a family $\mathcal{G}$ of two-dimensional binary lattices has good cross-combined measure then the family of binary sequences $\overline{E}(\mathcal{G})$ also has good cross-correlation measure. The proof of this fact will be very similar to the proof of Theorem F (see [17]).

Theorem 3. *Let $\mathcal{G}$ be a family of two-dimensional binary lattices $\eta : I_N^2 \rightarrow \{-1, +1\}$. Then*

$$\Phi_\ell(\overline{E}(\mathcal{G})) \leq (\ell + 2)\Phi_\ell(\mathcal{G})$$

Proof. By the definition of the cross-correlation measure we have that there exist binary sequences $\overline{E}(\eta_1), \overline{E}(\eta_2), \dots, \overline{E}(\eta_\ell) \in \overline{E}(\mathcal{G})$ (where $\eta_1, \eta_2, \dots, \eta_\ell \in \mathcal{G}$), $M \in \mathbb{N}$ and ℓ-tuple $D = (d_1, d_2, \dots, d_\ell)$ of non-negative integers with $0 \leq d_1 \leq d_2 \leq \dots \leq d_\ell < M + d_\ell$ with the additional restriction that if $\overline{E}(\eta_i) = \overline{E}(\eta_j)$ (in other words $\eta_i = \eta_j$) for some $i \neq j$, then we must not have $d_i = d_j$ and for which

$$\Phi_\ell(\overline{E}(\mathcal{G})) = \left| V_\ell(\overline{E}(\eta_1), \dots, \overline{E}(\eta_\ell), M, D) \right|. \tag{5.3}$$

Write $\overline{E}(\eta_i)$ of the form $\overline{E}(\eta_i) = (e_1^{(i)}, e_2^{(i)}, \dots, e_{N^2}^{(i)})$ for $i = 1, 2, \dots, \ell$. Then by (5.3)

$$\Phi_\ell(\overline{E}(\mathcal{G})) = \left| \sum_{n=1}^{M} e_{n+d_1}^{(1)} \cdots e_{n+d_\ell}^{(\ell)} \right|. \tag{5.4}$$

Next few definitions will follow: For $x \in \mathbb{Z}$ let

$$x = r_N(x)N + m_N(x)$$

where $m_N(x) \equiv x \pmod{N}$, $0 \leq m_N(x) \leq N - 1$ and $r_N(x) = \left[\frac{x}{N}\right]$.

By definition $e_{yN+x+1}^{(i)} = \eta_i(x, y)$ for $0 \leq x \leq N-1, 0 \leq y \leq N-1$ and $i = 1, \dots, \ell$ and thus

$$e_n^{(i)} = \eta_i(m_N(n-1), r_N(n-1)).$$

Then for $1 \leq i \leq \ell$

$$e_{n+d_i}^{(i)} = \eta(m_N(n + d_i - 1), r_N(n + d_i - 1)). \tag{5.5}$$

Here

$$n + d_i - 1 = (r_N(n-1) + r_N(d_i))N + m_N(n-1) + m_N(d_i).$$

Thus if $0 \leq m_N(n-1) + m_N(d_i) \leq N - 1$, then

$$r_N(n + d_i - 1) = r_N(n-1) + r_N(d_i), \quad m_N(n + d_i - 1) = m_N(n-1) + m_N(d_i)$$

and if $N \leq m_N(n-1) + m_N(d_i)$, then

$$r_N(n + d_i - 1) = r_N(n-1) + r_N(d_i) + 1, \quad m_N(n + d_i - 1) = m_N(n-1) + m_N(d_i) - N.$$

Thus we get that there exists an $a_i \overset{\text{def}}{=} N-1-m_N(d_i)$ such that for $m_N(n-1) \leq \ a_i$

$$r_N(n+d_i-1)=r_N(n-1)+r_N(d_i), \quad m_N(n+d_i-1)=m_N(n-1)+m_N(d_i) \tag{5.6}$$

and for $a_i+1 \leq m_N(n-1)$

$$r_N(n+d_i-1)=r_N(n-1)+r_N(d_i)+1, \quad m_N(n+d_i-1)=m_N(n-1)+m_N(d_i)-N. \tag{5.7}$$

Then $\{1, a_1+1, a_2+1, \dots, a_\ell+1, m_N(M-1)+1, N\}$ is a multiset which contains integers $1=c_1<c_2<\dots<c_m \leq N$ where $m \leq \ell+3$. By (5.6) and (5.7) we get that for $c_j \leq n \leq c_{j+1}-1$ there exist numbers $b_{i,j}$ and $f_{i,j}$ such that

$$r_N(n+d_i-1)=r_N(n)+r_N(d_i-1)+b_{i,j}, \quad m_N(n+d_i-1)=m_N(n)+m_N(d_i-1)-f_{i,j} \tag{5.8}$$

where $b_{i,j} \in \{0,1\}$ and $f_{i,j} \in \{0,N\}$. Moreover, if $b_{i,j}=0$, then $f_{i,j}=0$ and if $b_{i,j}=1$, then $f_{i,j}=N$. Now

$$\begin{aligned}
&[1,M]= \\
&=\{n=TN+x+1: \ T=0,1,\dots,\left[\frac{M-1}{N}\right], x=0,1,\dots,m_N(M-1)\} \\
&\quad \cup \{n=TN+x+1: \ T=0,1,\dots,\left[\frac{M-1}{N}\right]-1, x=m_N(M-1)+1, \\
&\quad \dots, N-1\}.
\end{aligned}$$

Thus

$$\begin{aligned}
[1,M]=\cup_{j=1}^{m-1}\{n: \ n=r_N(N-1)N+m_N(n-1)+1, \\
c_j \leq m_N(n-1) \leq c_{j+1}-1, r_N(n-1) \in \{0,1,2,\dots,T_j\}\}
\end{aligned} \tag{5.9}$$

where $T_j=\left[\frac{M-1}{N}\right]$ if $c_{j+1} \leq m_N(M-1)+1$ and $T_j=\left[\frac{M-1}{N}\right]-1$ if $m_N(M-1)+1 \leq c_j$. (Since $m_N(M-1)+1 \in \{c_1,c_2,\dots,c_m\}$ and $c_1<c_2<\dots<c_m$ thus $c_j<m_N(M-1)+1<c_{j+1}$ is not possible.) By this, (5.4), (5.5) and (5.6)

$$\begin{aligned}
\Phi_\ell(\overline{E}(\mathcal{G}))&=\sum_{n=1}^{M} e_{n+d_1}^{(1)} \cdots e_{n+d_\ell}^{(\ell)}=\sum_{j=1}^{m-1} \sum_{\substack{c_j \leq m_N(n-1) \leq c_{j+1}-1 \\ 1 \leq n \leq M}} e_{n+d_1}^{(1)} \dots e_{n+d_\ell}^{(\ell)} \\
&=\sum_{j=1}^{m-1} \sum_{\substack{c_j \leq m_N(n-1) \leq c_{j+1}-1 \\ 1 \leq n \leq M}} \\
&\quad \prod_{i=1}^{\ell} \eta_i(m_N(n-1)+m_N(d_i)-f_{i,j}, r_N(n-1)+r_N(d_i)+b_{i,j})
\end{aligned} \tag{5.10}$$

By (5.9)

$$\{(m_N(n-1), r_N(n-1)) :\ 1 \leq n \leq M \text{ and } c_j \leq m_N(n-1) \leq c_{j+1}-1\} = \{(x,y) :\ 0 \leq x \leq T_j \text{ and } c_j \leq y \leq c_{j+1}-1\}.$$

Using this, (5.8) and (5.10) we get

$$\Phi_\ell(\overline{E}(\mathcal{G})) = \sum_{j=1}^{m-1} \sum_{x=0}^{T_j} sum_{y=c_j}^{c_{j+1}-1} \prod_{i=1}^{\ell} \eta_i(x + m_N(d_i) - f_{i,j}, y + r_N(d_i) + b_{i,j}) \leq (m-1)\Phi_\ell(\mathcal{G}) \leq (\ell+2)\Phi_\ell(\mathcal{G})) \tag{5.11}$$

which was to be proved. Let us see whether the pairs $(m_N(d_i) - f_{i,j}, r_N(d_i) + b_{i,j})$ are different for fixed j as i runs over $1, 2, \dots, \ell$. Indeed if for fixed j there exist i_1 and i_2 with

$$(m_N(d_{i_1}) - f_{i_1,j}, r_N(d_{i_1}) + b_{i_1,j}) = (m_N(d_{i_2}) - f_{i_2,j}, r_N(d_{i_2}) + b_{i_2,j}),$$

then

$$N(r_N(d_{i_1}) + b_{i_1,j}) + m_N(d_{i_1}) - f_{i_1,j} = N(r_N(d_{i_2}) + b_{i_2,j}) + m_N(d_{i_2}) - f_{i_2,j}.$$

Since if $b_{i,j} = 0$, then $f_{i,j} = 0$ and if $b_{i,j} = 1$, then $f_{i,j} = N$, from this we get

$$Nr_N(d_{i_1}) + m_N(d_{i_1}) = Nr_N(d_{i_2}) + m_N(d_{i_2})$$
$$d_{i_1} = d_{i_2}$$

By the definition of cross-correlation measure $d_{i_1} = d_{i_2}$ is possible only if $\overline{E}(\eta_{i_1}) \neq \overline{E}(\eta_{i_2})$. Then clearly we have $\eta_{i_1} \neq \eta_{i_2}$, so indeed

$$\left| \sum_{x=0}^{T_j} \sum_{y=c_j}^{c_{j+1}-1} \prod_{i=1}^{\ell} \eta_i(x + m_N(d_i) - f_{i,j}, y + r_N(d_i) + b_{i,j}) \right|$$

can be estimated by $\Phi_\ell(\mathcal{G})$ in (5.11). This completes the proof of Theorem 3.

Using Theorems 1, 2, Corollary 2 and Theorem 3 we immediately get the following:

Corollary 3. *Let $q = p^2$ where p is a prime and define $\mathcal{G}_{\leq K, irreducible, quadratic}$ as in Construction 1 Then*

$$\Phi_\ell(\overline{E}(\mathcal{G}_{\leq K, irreducible, quadratic})) < K\ell(\ell+2)p(\log p + 1)^n + 2\ell.$$

Corollary 4. *Let p be a prime and define $\mathcal{G}_{\leq K, irreducible, Legendre}$ as in Construction 2. Then*

$$\Phi_\ell(\overline{E}(\mathcal{G}_{\leq K, irreducible, Legendre})) < 11K\ell(\ell+2)p^{3/2}\log p.$$

Corollary 5. *Let p be a prime and define $\mathcal{G}_{\le K,Sch-Eis,Legendre}$ as in Construction 3. Then*

$$\Phi_\ell(\overline{E}(\mathcal{G}_{\le K,Sch-Eis,Legendre})) < 11K\ell(\ell+2)p^{3/2}\log p.$$

Thus each family of binary sequences in Corollaries 3, 4 and 5 have optimal or nearly optimal cross-combined measure. Between them we were able to prove the strongest bound for cross-correlation measure in the case of the family of $\overline{E}(\mathcal{G}_{\le K,irreducible,quadratic})$. The weak point of this construction is that it is based on one-variable irreducible polynomials over $\mathbb{F}_{p^2}$ and however there are relatively fast algorithms to construct one-variable irreducible polynomials (see e.g. the Handbook of Finite Fields [37]), still in certain applications these algorithms are too complicated or are not fast enough (e.g. we need several binary sequences or lattices used them as a key-streams in Vernam-cipher). Using binary lattices based on two-variable irreducible polynomials and Legendre symbol this problem can be avoided, however a slightly weaker upper bound is obtained for the cross-correlation measure than in the original construction. But, contrary to one-variable polynomials, using Schöneman-Eisenstein criteria it is very fast and easy to construct two-variable irreducible polynomials over $\mathbb{F}_p$ (e.g. see Lemma 3). Indeed by Construction 3 the binary lattices in $\mathcal{G}_{\le K,Sch-Eis,Legendre}$ can be implemented easily and fast, and thus the binary sequences in $\overline{E}(\mathcal{G}_{\le K,Sch-Eis,Legendre})$ also can be implemented easily and fast. However we do not have the strongest bound $cp\log p$, we have only $cK\ell^2p^{3/2}\log p$ for the cross-correlation measure of this family, it is much better than the trivial bound p^2. Moreover, the family $\overline{E}(\mathcal{G}_{\le K,Sch-Eis,Legendre})$ is very big, it contains more than $p^{K(K-1)/2}$ pieces of binary sequences, which is also important in the applications.

Acknowledgement. I would like to thank the referee for his careful reading and valuable advice concerning Theorem 1.

References

1. Ahlswede, R., Khachatrian, L.H., Mauduit, C., Sárközy, A.: A complexity measure for families of binary sequences. Period. Math. Hung. **46**, 107–118 (2003)
2. Ahlswede, R., Mauduit, C., Sárközy, A.: Large families of pseudorandom sequences of k symbols and their complexity – part I. In: Ahlswede, R., Bäumer, L., Cai, N., Aydinian, H., Blinovsky, V., Deppe, C., Mashurian, H. (eds.) General Theory of Information Transfer and Combinatorics. LNCS, vol. 4123, pp. 293–307. Springer, Heidelberg (2006). https://doi.org/10.1007/11889342_16
3. Ahlswede, R., Mauduit, C., Sárközy, A.: Large families of pseudorandom sequences of k symbols and their complexity – part II. In: Ahlswede, R., Bäumer, L., Cai, N., Aydinian, H., Blinovsky, V., Deppe, C., Mashurian, H. (eds.) General Theory of Information Transfer and Combinatorics. LNCS, vol. 4123, pp. 308–325. Springer, Heidelberg (2006). https://doi.org/10.1007/11889342_17
4. Alon, N., Kohayakawa, Y., Mauduit, C., Moreira, C.G., Rödl, V.: Measures of pseudorandomness for finite sequences: typical values. Proc. Lond. Math. Soc. **95**, 778–812 (2007)

5. Anantharam, V.: A technique to study the correlation measures of binary sequences. Discret. Math. **308**, 6203–6209 (2008)
6. Bérczes, A., Ködmön, J., Pethő, A.: A one-way function based on norm form equations. Period. Math. Hung. **49**, 1–13 (2004)
7. Cassaigne, J., Ferenczi, S., Mauduit, C., Rivat, J., Sárközy, A.: On finite pseudorandom binary sequences III: the Liouville function, I. Acta Arith. **87**, 367–384 (1999)
8. Cassaigne, J., Mauduit, C., Sárközy, A.: On finite pseudorandom binary sequences VII: the measures of pseudorandomness. Acta Arith. **103**, 97–118 (2002)
9. Delinge, P.: La conjecture de Weil, I. Publications Mathématiques de l'Institut des Hautes Études Scientifiques **43**, 273–307 (1974)
10. Delinge, P.: La conjecture de Weil, II. Publications Mathématiques de l'Institut des Hautes Études Scientifiques **43**, 137–250 (1980)
11. Feistel, H., Notz, W.A., Smith, J.L.: Some cryptographic techniques for machine-to-machine data communications. Proc. IEEE **63**, 1545–1554 (1975)
12. Fouvry, E., Katz, N.: A general stratification theorem for exponential sums, and applications. Journal für die reine und angewandte Mathematik **540**, 115–166 (2001)
13. Gong, G.: Character sums and polyphase sequence families with low correlation, discrete fourier transform (DFT), and ambiguty. In: Pascale, C., et al. (eds.) Finite Fields and Their Applications. Radon Series on Computational and Applied Mathematics, vol. 11, pp. 1–42. de Gruyter, Berlin (2013)
14. Goubin, L., Mauduit, C., Sárközy, A.: Construction of large families of pseudorandom binary sequences. J. Number Theory **106**, 56–69 (2004)
15. Gyarmati, K.: Concatenation of pseudorandom binary sequences. Period. Math. Hung. **58**, 99–120 (2009)
16. Gyarmati, K.: On the complexity of a family related to the Legendre symbol. Period. Math. Hung. **58**, 209–215 (2009)
17. Gyarmati, K.: On the correlation of subsequences. Unif. Distrib. Theory **7**, 169–195 (2012)
18. Gyarmati, K., Mauduit, C., Sárközy, A.: Pseudorandom binary sequences and lattices. Acta Arith. **135**, 181–197 (2008)
19. Gyarmati, K., Mauduit, C., Sárközy, A.: Measures of pseudorandomness of finite binary lattices, I (The measures Q_k, normality.). Acta Arith. **144**, 295–313 (2010)
20. Gyarmati, K., Mauduit, C., Sárközy, A.: Measures of pseudorandomness of finite binary lattices, III (Q_k, correlation, normality, minimal values.). Unif. Distrib. Theory **5**, 183–207 (2010)
21. Gyarmati, K., Mauduit, C., Sárközy, A.: The cross-correlation measure for families of binary sequences. In: Larcher, G., Pillichshammer, F., Winterhof, A., Xing, C. (eds.) Applications of Algebra and Number Theory (Lectures on the occasion of Harald Niederreiter's 70th Birthday) (2014)
22. Gyarmati, K., Mauduit, C., Sárközy, A.: Measures of pseudorandomness of finite binary lattices, I. (The measures Q_k, normality.). Acta Arith. **144**, 295–313 (2010)
23. Gyarmati, K., Mauduit, C., Sárközy, A.: Measures of pseudorandomness of finite binary lattices, II. (The symmetry measures.). Ramanujan J. **25**, 155–178 (2011)
24. Gyarmati, K., Mauduit, C., Sárközy, A.: Measures of pseudorandomness of finite binary lattices, III. (Q_k, correlation, normality, minimal values). Unif. Distrib. Theory **5**, 183–207 (2010)
25. Gyarmati, K., Sárközy, A., Stewart, C.L.: On Legendre symbol lattices. Unif. Distrib. Theory **4**, 81–95 (2009)

26. Gyarmati, K., Sárközy, A., Stewart, C.L.: On Legendre symbol lattices, II. Unif. Distrib. Theory **8**, 47–65 (2013)
27. Hoffstein, J., Lieman, D.: The distribution of the quadratic symbol in function fields and a faster mathematical stream cipher. In: Lam, K.Y., Shparlinski, I., Wang, H., Xing, C. (eds.) Cryptography and Computational Number Theory. PCS, vol. 20, pp. 59–68. Birkhäuser Verlag, Basel (2001)
28. Hubert, P., Mauduit, C., Sárközy, A.: On pseudorandom binary lattices. Acta Arith. **125**, 51–62 (2006)
29. Kam, J., Davida, G.: Structured design of substitution-permutation encryption networks. IEEE Trans. Comput. **28**, 747–753 (1979)
30. Mauduit, C., Sárközy, A.: On finite pseudorandom binary sequences I: measures of pseudorandomness, the Legendre symbol. Acta Arith. **82**, 365–377 (1997)
31. Mauduit, C., Sárközy, A.: On large families of pseudorandom binary lattices. Unif. Distrib. Theory **2**, 23–37 (2007)
32. Mauduit, C., Sárközy, A.: Family complexity and VC-dimension. In: Aydinian, H., Cicalese, F., Deppe, C. (eds.) Information Theory, Combinatorics, and Search Theory. LNCS, vol. 7777, pp. 346–363. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36899-8_15
33. Menezes, A., van Oorschot, P.C., Vanstone, S.: Handbook of Applied Cryptography. CRS Press, Boca Raton (1997)
34. Mérai, L.: On the typical values of the cross-correlation measure. Monatsh. Math. **180**(1), 83–99 (2016)
35. Mérai, L.: The cross-correlation measure of families of finite binary sequences: limiting distributions and minimal values. Discret. Appl. Math. **214**, 153–168 (2016)
36. Mérai, L., Rivat, J., Sárközy, A.: The measures of pseudorandomness and the NIST tests. In: Kaczorowski, J., et al. (eds.) NuTMiC 2017. LNCS, vol. 10737, pp. 197–216. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-319-76620-1_12
37. Mullen, G.L., Panario, D.: Handbook of Finite Fields. Chapman and Hall/CRC, Boca Raton (2013)
38. Rédei, L.: Algebra. Pergamon Press, Oxford/New York/Toronto (1967)
39. Rivat, J., Sárközy, A.: On pseudorandom sequences and their application. In: Ahlswede, R., Bäumer, L., Cai, N., Aydinian, H., Blinovsky, V., Deppe, C., Mashurian, H. (eds.) General Theory of Information Transfer and Combinatorics. LNCS, vol. 4123, pp. 343–361. Springer, Heidelberg (2006). https://doi.org/10.1007/11889342_19
40. Tóth, V.: Collision and avalanche effect in families of pseudorandom binary sequences. Period. Math. Hung. **55**, 185–196 (2007)
41. Tóth, V.: The study of collision and avalanche effect in a family of pseudorandom binary sequences. Period. Math. Hung. **59**, 1–8 (2009)
42. Weil, A.: Sur les courbes algébriques et les variétés qui s'en déduisent, Act. Sci. Ind. **1041** (1948)
43. Winterhof, A.: Some estimates for character sums and applications. Des. Codes Crypt. **22**, 123–131 (2001)

Algebraic Structures and Analysis

The Cube Attack on Courtois Toy Cipher

Janusz Szmidt(✉)

Military Communication Institute, 05-130 Zegrze, Poland
j.szmidt@wil.waw.pl

Abstract. The cube attack has been introduced by Dinur and Shamir [8] as a known plaintext attack on symmetric primitives. The attack has been applied to reduced variants of stream ciphers Trivium [3,8] and Grain-128 [2], a reduced to three rounds variant of the block cipher Serpent [9] and a reduced version of the keyed hash function MD6 [3]. In another form the attack appeared in the Vielhaber ePrint articles [13,14], where it was named AIDA (*Algebraic Initial Value Differential Attack*) and applied to reduced variants of Trivium. We applied the cube attack to the reduced variant of Courtois Toy Cipher (CTC) consisting of four rounds and 120-bit key. After that we extended the attack to five rounds of CTC by applying the *4 + 1* cryptanalytic principle. The article also presents experimental results of recovering the key.

Keywords: Cube attack · Symmetric primitives
Boolean polynomials · CTC · The *4 + 1* cryptanalytic principle

1 Introduction

In recent years there have been developed the methods of algebraic cryptanalysis of symmetric primitives, i.e. block and stream ciphers, hash and MAC functions. The idea is to represent the investigated algorithm as a system of multivariate polynomials involving the plaintext and ciphertext bits, the initial value bits and the key bits as their variables. As a result, to break the cryptosystem (to find the secret key) one must solve such a complicated system of algebraic equations. For ciphers used in practice such systems are too large to be solved with the computational capabilities currently available. Thus usually reduced and simplified versions of symmetric algorithms are considered to investigate the applicability of algebraic cryptanalysis.

In this paper, we apply recently introduced by Dinur and Shamir [8] cube attack, as an example of algebraic technique in cryptanalysis. The method involves also some probabilistic tools. The linear tests were applied to test the approximation of complicated Boolean functions of several variables. If this approximation is possible with probability close to one, then the cube attack is applicable.

The cube attack were applied to reduced variants of stream ciphers Trivium [3,8] and Grain-128 [2], a reduced to three rounds variant of the block cipher

J. Kaczorowski et al. (Eds.): NuTMiC 2017, LNCS 10737, pp. 241–253, 2018.
https://doi.org/10.1007/978-3-319-76620-1_14

Serpent [9] and a reduced version of the keyed hash function MD6 [3]. In another form the attack appeared in the Vielhaber ePrint articles [13,14], where it was named AIDA (*Algebraic Initial Value Differential Attack*) and applied to reduced versions of Trivium. Vielhaber considered Disjunctive Normal Form (DNF) or Algebraic Normal Form (ANF) of Boolean Functions and Inclusion-Exclusion-Principle to represent sums over cubes as linear terms of key bits. Dinur and Shamir in their paper [8] first time applied linear tests to extract linear terms of key bits. Their method does not depend on particular form of investigated Boolean Functions. It seems that linear tests are more suitable in application to block ciphers.

The CTC was designed by Courtois [5] to apply methods of algebraic cryptanalysis. The security of this cipher was analysed by Courtois [5], Albrecht [1], and Dunkelman and Keller [10]. The modification of CTC named CTC2 [6] was investigated by Courtois [6] and Dunkelman and Keller [11]. Algebraic, differential and linear cryptanalysis provides attacks below the exhaustive key search complexity (see [10,11]). Although, CTC and CTC2 are not ciphers used in practice, cryptanalysts have payed the attention to them.

Our contribution is an application of the cube attack to a version of the Courtois Toy Cipher with four rounds and 120-bit key and the extension of the original cube attack by combining it with the *4 + 1* cryptanalytic principle, where we add one round more. In our attack we assume that during the *preprocessing* phase an attacker can encrypt the chosen plaintexts and investigate the sums of the ciphertexts bits as a function of the key bits. The main task of this phase is to find linear (or affine) functions using the linear tests [4]. During the *preprocessing* phase the attacker collects many linear expressions in key bits and chooses linearly independent ones; we have used here the MAGMA [15] package to do the needed calculations.

The *on-line* phase is a chosen plaintext attack, where one round is added to the cipher. The key is secret now and an attacker encrypts the plaintexts (obtained from the cubes found in the previous phase) and collects the ciphertexts after the added round. Now the attacker has no access to partial ciphertexts after the previous round. The *4 + 1* phase compares the right hand sides of the linear expressions obtained during the *preprocessing* phase (but without explicit calculation of them, as it was done in the original Dinur and Shamir cube attack) with the sums of bits obtained after inverting the last round of the cipher. The task is realized using the explicit formules for output bits of the inversion of the last round.

The *4 + 1* attack was practically realized for the five round CTC with 120-bit block and key size. In the experiments, randomly chosen keys were retrieved during the *on-line* phase of the attack. It is worth to mention, that using the *BooleanPolynomial* class from the SAGE [17] package and the code written in Python [16] it is possible to perform the necessary calculations with quadratic Boolean functions depending on 240 binary variables.

2 The Cube Attack

We shall not distinguish at the moment between secret and public variables. Let p be a polynomial of n variables $x_1, \ldots, x_n$ over the field $GF(2)$. For a fixed subset of indices $I = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$ let us take a monomial $t_I = x_{i_1} \ldots x_{i_k}$. Then we have a decomposition

$$p(x_1, \ldots, x_n) = t_I \cdot p_{S(I)} + q(x_1, \ldots, x_n),$$

where the polynomial $p_{S(I)}$ does not depend on the variables $x_{i_1}, \ldots, x_{i_k}$.

Definition 1. The maxterm of the polynomial p we call the monomial t_I, such that $deg(p_{S(I)}) = 1$, it means that the polynomial $p_{S(I)}$ corresponding to the subset of indices I is a linear one, which is not a constant.

The set of indices I defines the k-dimensional Boolean cube C_I, where on the place of each of the indices we put 0 or 1. A given vector $v \in C_I$ defines the derived polynomial p_v depending on $n - k$ variables, where in the basic polynomial p we put the values corresponding to the vector v. Summing over all vectors in the cube C_I we obtain the polynomial

$$p_I = \sum_{v \in C_I} p_v.$$

Theorem 1. Let p be a polynomial over the field $GF(2)$ and $I \subset \{1, \ldots, n\}$ the subset of indices. Then we have

$$p_I = p_{S(I)},$$

where the polynomials are equal modulo 2.

Let us consider a cryptosystem described by the polynomial

$$p(v_1, \ldots, v_m, x_1, \ldots, x_n)$$

depending on m public variables $v_1, \ldots, v_m$ (the initial value or plaintext) and on n secret variables $x_1, \ldots, x_n$ (the key). The value of the polynomial represents a ciphertext bit. In general, the polynomial p is not explicitelly known; it can be a *black box*. We will consider the known plaintext attack, where at the *preprocessing* phase the attacker has also access to secret variables.

The attack has two phases. In the *preprocessing* one the attacker can change the values of public and secret variables. The task is to obtain a system of linear equations on secret variables. In the second *on-line* phase of the attack the key is secret and the attacker can change the values of public variables. He adds the output bits, where the inputs run over some multi-dimensional cube. The task is to obtain the right hand sides of linear equations. The system of linear equations can be solved giving some bits of the key.

The first task is to fix the dimension of the cube and the public variables over which we will sum up; they are called *the tweakable* variables, and the other public variables are equal to zero. If we know the degree d of the basic polynomial, we fix the cube dimension to $d-1$. We sum up over a fixed cube for several values of secret variables and collect the obtained values. We do the linear tests for the obtained function of secret variables and store it when it is linear. The linear test for a function $f(x)$ depending on a collection x of binary variables requires checking the condition

$$f(x \oplus x') = f(x) \oplus f(x') \oplus f(0)$$

for some randomly chosen arguments x, x'. If the function f passes the linear test for a few hundreds of pairs x, x' and it is not a constant function (equal to zero or to one), then we can put the hypothesis that it is a linear (or affine) function. The theoretical explanation for these tests has been elaborated in the paper [4].

The next task is to calculate the coefficients explicit values of the obtained linear function of secret variables. The free term of the linear function we obtain fixing all its arguments as equal to zero. The coefficient of the variable x_i is equal 1 if and only if the change of this variable implies the change of value of the function. The coefficient of the variable x_i is equal to 0 if and only if the change of this variable does not imply the change of value of the function. The task of the *preprocessing* phase of attack is to collect possibly many independent linear terms - they constitute the system of linear equations on secret variables. This system of linear equations will be used in the *on-line* phase of attack. The *preprocessing* procedure is done only once in cryptanalysis of the algorithm.

In the *on-line* phase an attacker has access only to public variables (the plaintexts for block ciphers, the initial values for stream ciphers), which he can change and then he calculates the corresponding bits of the ciphertext under the unknown value of secret variables. The task of this phase of attack is to find some bits of secret key with complexity, which is lower than that of the exhaustive search in the brute force attack. The attacker uses the derived system of linear equations for secret variables (the unknown bits of the key), where the right hand sides of these equations are sums of bits values of ciphertexts obtained after summation over the same cubes as in the *preprocessing* phase, but now the key is not known.

The cube attack is applicable to symmetric ciphers for which the polynomials describing the system have relatively low degree. Then one can eventually find some bits of unknown key; the remaining bits of the key may be found by brute force search. After successful *preprocessing*, the *on-line* phase of the attack can be done many times for different unknown keys. In general, the cube attack is applicable to cryptosystems without knowing their inner structure. The attacker must have the possibility to realize the preprocessing phase and in the *on-line* one has an access to the implementation of the algorithm (to perform the summation over cubes under unknown key).

3 The Courtois Toy Cipher

3.1 The Specification

The CTC has been designed by Courtois [5,6] to apply algebraic cryptanalysis methods. It is an SPN network with a scalable number of rounds, block and key size. Each round performs the same operations on the input data, except that a different round key is added each time. The number of rounds is denoted by N_r. The output of round $i-1$ is the input to round i. Each round consists of parallel application of B S-boxes (S), the application of the linear diffusion layer (D), and a final key addition of the round key (K_i). The round key K_0 is added to the plaintext block before the first round.

The plaintext bits $p_0 \dots p_{Bs-1}$ are identified with $Z_{0,0} \dots Z_{0,Bs-1}$ and the ciphertext bits $c_0 \dots c_{Bs-1}$ are identified with $X_{N_r+1,0} \dots X_{N_r+1,Bs-1}$ to have an uniform notation ($s = 3$ is the size of the S-box). The S-box was chosen as the permutation

$$[7, 6, 0, 4, 2, 5, 1, 3].$$

It has $2^3 = 8$ inputs and 8 outputs. The output bits are quadratic Boolean functions of the input bits which can be expressed as

$$y_0 = x_0x_1 + x_0 + x_1 + x_2 + 1,$$

$$y_1 = x_0x_2 + x_1 + 1,$$

$$y_2 = x_0x_1 + x_0x_2 + x_1x_2 + x_1 + x_2 + 1,$$

and for the inverse S-box:

$$x_0 = y_0y_1 + y_2,$$

$$x_1 = y_0y_1 + y_0y_2 + y_1 + 1.$$

$$x_2 = y_0y_1 + y_1y_2 + y_0 + y_1.$$

The explicit form of these functions will be used when we apply *the meet-in-the-middle* method.

The diffusion layer (D) is defined as

$$Z_{i,257modBs} = Y_{i,0},$$

for $i = 1, \dots, N_r$, and

$$Z_{i,(1987j+257)modBs} = Y_{i,j} + Y_{i,(j+137)modBs}$$

for $j \neq 0$ and all i, where $Y_{i,j}$ represent input bits and $Z_{i,j}$ represent output bits.

The key schedule is a simple permutation of bits:

$$K_{i,j} = K_{0,(i+j)modBs}$$

for all i and j, where K_0 is the main key. Key addition is performed bit-wise:

$$X_{i+1,j} = Z_{i,j} + K_{i,j}$$

for all $i = 1, \dots, N_r$ and $j = 0, 1, \dots, Bs - 1$, where $Z_{i,j}$ represent output bits of the previous diffusion layer, $X_{i+1,j}$ the input bits of the next round, and $K_{i,j}$ the bits of the current round key (Fig. 1).

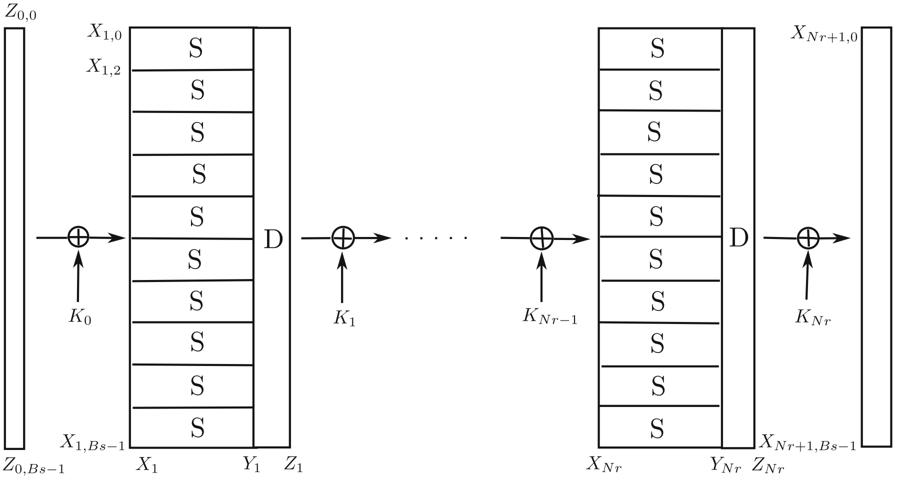


Fig. 1. CTC overview for B = 10.

3.2 The Cube Attack on CTC

We have applied the cube attack to the version of Courtois Toy Cipher with four rounds and 120-bit block and key size. We have found the maxterms corresponding to four dimensional cubes and we have collected 120 r linearly independent linear polynomials related to them which are given in Table 1 in the Appendix. The table contains the indices of the cubes, the corresponding linear expressions and the output bits after four rounds of the CTC for which there were found these expressions after summation over the cubes. There were performed 1000 linear tests for each expression to test its linearity. In fact, these 120 linearly independent functions were chosen among 610 linear ones generated during the *preprocessing* phase. It is difficult to estimate explicitelly the complexity of this phase. The first task was to find, for which dimension of cubes there appear the maxterms with corresponding linear polynomials. This phenomenon has appeared for four dimensional cubes after four rounds of 120-bit CTC. Each round of CTC can be described by quadratic Boolean functions, hence the output bits of four round CTC are described by Boolean polynomials of degree $2^4 = 16$ regarded as a function of the plaintext bits and the key bits. According to general principles, there should exist linear expressions corresponding to 15-dimensional cubes, but we have not found any up to now; probably the probability to detect any of them is very low. The existence of linear terms for four dimensional cubes in this case may be related to poor diffusion.

The complexity of the *on-line* phase is equal to $120 \times 2^4 \approx 2^{11}$ encryptions of the four round CTC. In this phase the attacker has the derived system of linear equations and calculates (by summing up over the cubes the ciphertext bits) the right hand sides of these equations. The solution of this system gives the key. We have performed the experiment for several randomly chosen keys and obtained their exact values. All the calculations involving linear algebra, e.g. solving the systems of linear equations over binary field, were done with Magma [15] computational system.

Below there are presented the results obtained for the variant of CTC with six rounds and 15-bit block and key size. The maxterms with corresponding linear polynomials have appeared after summation over 14-dimensional cubes. Here is the system of 15 linearly independent equations with the right hand sides representing the sums of ciphertext bits after six rounds of this variant of CTC. The eight linear equations obtained for the cube $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}$:

$$\begin{aligned}
x0 + x1 + x2 + x3 + x4 &= c0\\
x2 + x3 + x9 &= c1\\
x2 + x3 + x10 + x11 + x12 + x13 &= c2\\
1 + x2 + x3 + x6 + x7 + x8 + x9 + x11 + x13 &= c3\\
x0 + x1 + x2 + x4 + x7 + x8 + x10 + x11 + x12 &= c4\\
x0 + x3 + x6 + x9 + x11 &= c6\\
x5 + x11 &= c7\\
1 + x0 + x3 + x6 + x7 + x8 + x10 + x12 + x13 &= c10
\end{aligned}$$

The seven linear equations obtained for the cube $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14\}$:

$$\begin{aligned}
1 + x3 + x4 + x7 + x8 + x9 + x11 + x13 &= c0\\
x1 + x3 + x5 + x10 + x11 + x12 + x13 + x14 &= c1\\
1 + x0 + x2 + x3 + x5 + x6 + x8 + x9 + x14 &= c2\\
x1 + x2 + x3 + x5 + x6 + x8 + x9 + x10 + x11 + x12 + x13 + x14 &= c3\\
1 + x0 + x1 + x2 + x3 + x4 + x6 + x9 + x10 + x13 + x14 &= c4\\
x0 + x2 + x3 + x7 + x10 + x13 + x14 &= c6\\
1 + x0 + x1 + x2 + x4 + x5 + x6 + x7 + x9 + x14 &= c7
\end{aligned}$$

This example will be continued in the next point, where we will add one round more to extend the attack.

4 The Cube Attack and the *4 + 1* Cryptanalytic Principle

We assume now that in the *preprocessing* phase the attacker has access to keys and encryption data after four rounds of CTC (the variant with 120-bit block and key size) and then he collects the linear expressions in key bits which are given in Table 1 of the Appendix. Now in the *on-line* phase we assume that the attacker can encrypt the plaintexts corresponding to the previously selected cubes and can collect the ciphertexts only after five rounds of the CTC. The task of this phase is to obtain the linear equations for unknown bits of the key.

We invert the last fifth round of the cipher and obtain the exact formules expressing the output bits as quadratic Boolean functions of the ciphertext bits (after five rounds) and the bits of the key. Summing up these output bits over ciphertexts belonging to the given cube we obtain linear expression in unknown bits of the key: there is an even number of ciphertexts corresponding to the cube and the quadratic terms in key bits are canceled. Now we make them equal to linear expressions given in Table 1 (obtained in *preprocessing* phase after four rounds) having this way the system of linear equations for the bits of the key. In fact, we compare the sums of the bits after four rounds with the sums of the bits obtained after decryption of the fifth round, but we do not collect the exact values of bits in the meeting point (these bits are equal and hence their sums are equal too). The exact formules for the inverted last round are not presented here since they are to complicated. The main reason is, that the inversion of the diffusion layer has not a simple form. We have generated them using the suitable program and they are included in the execution files (see below for the simplest case). The Fig. 2 depicts the *4 + 1* attack.

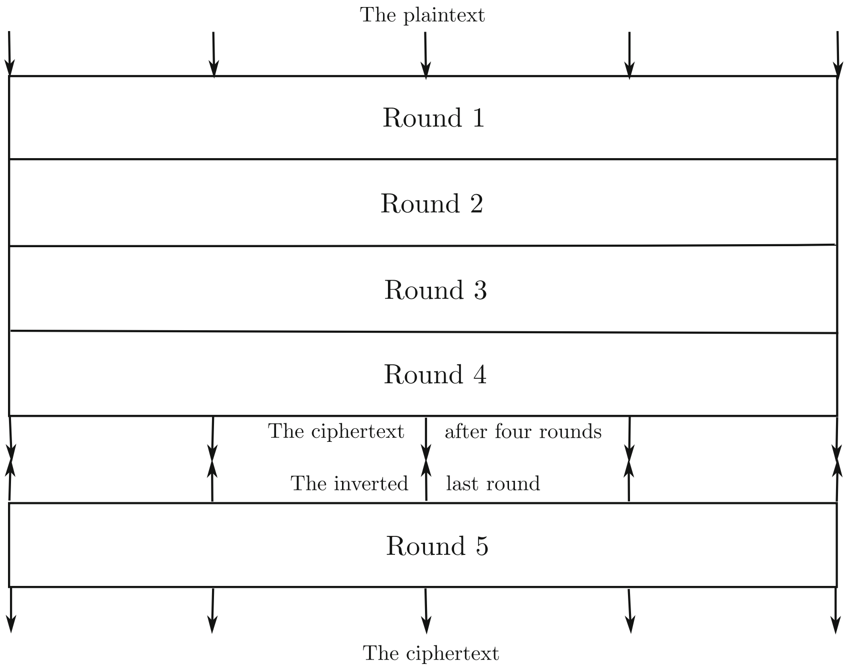


Fig. 2. The *4 + 1* attack.

As an example, the formules for the inversion of diffusion layer in the case of CTC with five S-boxes (i.e., 15-bit plaintext and key size) are given below. Here

$z0, \ldots, z14$ are the inputs bits and $y0, \ldots, y14$ the output bits of the diffusion layer.

$$\begin{aligned}
y0 &= z2 \\
y1 &= z2 + z3 + z4 + z5 + z6 + z7 + z8 + z9 \\
y2 &= z0 + z1 + z2 + z3 + z4 + z5 + z6 + z7 + z8 + z9 + z10 + z11 + z12 \\
&\quad + z13 + z14 \\
y3 &= z2 + z3 + z4 + z5 + z6 + z7 + z8 \\
y4 &= z0 + z2 + z3 + z4 + z5 + z6 + z7 + z8 + z9 + z10 + z11 + z12 + z13 \\
&\quad + z14 \\
y5 &= z2 + z3 + z4 + z5 + z6 + z7 \\
y6 &= z2 + z3 + z4 + z5 + z6 + z7 + z8 + z9 + z10 + z11 + z12 + z13 + z14 \\
y7 &= z2 + z3 + z4 + z5 + z6 \\
y8 &= z2 + z3 + z4 + z5 + z6 + z7 + z8 + z9 + z10 + z11 + z12 + z13 \\
y9 &= z2 + z3 + z4 + z5 \\
y10 &= z2 + z3 + z4 + z5 + z6 + z7 + z8 + z9 + z10 + z11 + z12 \\
y11 &= z2 + z3 + z4 \\
y12 &= z2 + z3 + z4 + z5 + z6 + z7 + z8 + z9 + z10 + z11 \\
y13 &= z2 + z3 \\
y14 &= z2 + z3 + z4 + z5 + z6 + z7 + z8 + z9 + z10
\end{aligned}$$

We have performed the described above *4 + 1* phase of the attack with the inverted seventh round of the 15-bit CTC and here there are the obtained linear equations.

$$\begin{aligned}
x0 + x1 + x2 + x3 + x4 + x9 &= 0 \\
x2 + x3 + x9 &= 1 \\
x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11 + x12 + x13 &= 0 \\
x0 + x2 + x3 + x6 + x7 + x8 + x10 + x12 + x14 &= 0 \\
x1 + x2 + x4 + x7 + x8 + x9 + x13 + x14 &= 1 \\
x1 + x2 + x4 + x5 + x9 + x11 + x14 &= 0 \\
x5 + x6 + x9 + x10 + x12 + x13 &= 1 \\
x1 + x2 + x4 + x6 + x7 + x8 + x10 + x14 &= 0 \\
x0 + x1 + x3 + x4 + x7 + x8 + x9 + x10 + x12 + x14 &= 0 \\
x1 + x2 + x4 + x6 + x7 + x8 + x9 + x10 + x11 + x12 + x13 + x14 &= 0 \\
x0 + x4 + x7 + x14 &= 1 \\
x4 + x7 + x8 + x9 + x10 + x11 + x12 + x13 + x14 + 1 &= 0 \\
x0 + x5 + x7 + x11 + x12 &= 0 \\
x0 + x2 + x3 + x7 + x10 + x13 + x14 &= 0 \\
x0 + x1 + x2 + x4 + x5 + x6 + x7 + x9 + x14 &= 0
\end{aligned}$$

The solution of these equations is the key:

$$(x0, x1, \ldots, x14) = (1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1),$$

randomly chosen at the beginning of the experiment.

We have performed the same experiment for the five round CTC with 120-bit block and key size. We have exploited the linear expressions (Appendix, Table 1) obtained after four rounds during the *preprocessing* phase. In fact, each of them corresponds here with another 4-dimensional cube. In the *on-line* phase which is now the *4 + 1* attack, we have collected the ciphertexts after five rounds of 120-bit CTC obtained after encryptions with the key (which is assumed to be unknown in the experiment) referring to the same cubes. The system of 120 linear equations is too large to write it down here (see the extended version [12] of this article). All manipulations with the Boolean polynomials depending on $120+120=240$ binary variables were done in SAGE package [17] and the related program was written using the Python language. It appears that the rank of this system of linear equations is equal to 119, hence one bit of the key must be guessed. The performed experiments have confirmed the correctness of the method for several randomly chosen 120-bit keys. The complexity of the *on-line* phase here is the 2^{11} encryptions of five round CTC and storage of the 2^{11} 120-bit ciphertexts. The complexity of the linear part of caculations is negligible.

In general, this *4 + 1* extension of extension work in the situation when we are able to realize successfully the *preprocessing* phase of the cube attack for n rounds of block cipher and the invertion of $n + 1$ round leads to a system of equations which could be solved.

5 Conclusion

After investigating five round and 120-bit key CTC we have applied the same methods to five round and 255-bit key and plaintext block CTC2 (the experimental results will be given in another note). We have tried to find linear terms referring to cubes for versions of CTC and CTC2 with five rounds, but it was only possible for plaintext blocks up to 57 bits. It seems that the probability to find such linear terms is very low for bigger plaintext blocks. An appearance of linear terms for these ciphers with a small number of rounds can be interpreted as an effect of poor diffusion. We see a possibility to apply similar attacks for six and more rounds of Courtois Toy Ciphers in combining the cube attack, the *4+ r* (r the number of last rounds) method and description of two or more last r rounds as a system of quadratic equations in key bits and auxiliary variables. Then a success of attack depends on possibility to solve obtained systems of equations.

Appendix

Table 1. The linear expressions for CTC with 4 rounds and 120-bit key.

Cube idices	Expression	Out. bit	Cube indices	Expression	Out. bit
$\{78, 84, 86, 113\}$	x80	c69	$\{26, 28, 63, 118\}$	x64+x65	c69
$\{71, 85, 107, 116\}$	1 + x69 + x70	c21	$\{5, 46, 86, 103\}$	1 + x84 + x85	c37
$\{32, 63, 64, 77\}$	1 + x30 + x31	c17	$\{22, 84, 110, 113\}$	x86	c76
$\{10, 11, 12, 50\}$	x13 + x14	c17	$\{49, 62, 68, 113\}$	1 + x48 + x50	c87
$\{25, 74, 100, 101\}$	1 + x24	c102	$\{32, 65, 73, 89\}$	1 + x87 + x88	c93
$\{49, 76, 85, 86\}$	1 + x75	c102	$\{4, 20, 32, 84\}$	x85 + x86	c39
$\{36, 37, 110, 115\}$	x108	c99	$\{18, 20, 62, 73\}$	1 + x60 + x61	c63
$\{20, 23, 112, 114\}$	x116	c106	$\{1, 8, 64, 77\}$	1 + x63 + x65	c53
$\{0, 13, 61, 92\}$	x2	c117	$\{37, 38, 91, 115\}$	1 + x90 + x92	c99
$\{41, 56, 78, 110\}$	x79 + x80	c51	$\{37, 67, 97, 109\}$	1 + x66	c93
$\{14, 20, 46, 51\}$	x53	c96	$\{18, 20, 47, 114\}$	x115 + x116	c3
$\{38, 53, 79, 80\}$	x36	c81	$\{40, 45, 98, 119\}$	1 + x46	c31
$\{7, 11, 47, 52\}$	1 + x6 + x8	c43	$\{13, 59, 92, 101\}$	x99	c113
$\{25, 46, 83, 104\}$	1 + x45 + x47	c16	$\{3, 12, 14, 97\}$	1 + x4	c30
$\{0, 2, 94, 98\}$	1 + x93 + x95	c93	$\{10, 58, 70, 101\}$	1 + x99 + x100	c54
$\{11, 23, 79, 92\}$	1 + x78	c70	$\{5, 26, 59, 97\}$	x57	c48
$\{11, 35, 43, 118\}$	1 + x33 + x34	c1	$\{34, 75, 87, 89\}$	x76 + x77	c17
$\{0, 52, 98, 112\}$	x1 + x2	c32	$\{5, 56, 58, 104\}$	1 + x57 + x59	c35
$\{12, 14, 56, 89\}$	1 + x54 + x55	c117	$\{7, 35, 53, 70\}$	1 + x51 + x52	c1
$\{61, 62, 89, 102\}$	x104	c48	$\{41, 82, 83, 94\}$	x39	c84
$\{24, 28, 53, 107\}$	x26	c19	$\{21, 41, 49, 77\}$	x22 + x23	c114
$\{17, 49, 81, 101\}$	x82 + x83	c54	$\{28, 74, 88, 98\}$	1 + x96 + x97	c8
$\{3, 97, 98, 101\}$	x4 + x5	c35	$\{38, 69, 100, 101\}$	x71	c114
$\{62, 97, 113, 117\}$	1 + x118	c94	$\{22, 27, 82, 107\}$	x29	c19
$\{8, 75, 83, 115\}$	1 + x76	c39	$\{18, 26, 58, 71\}$	1 + x19	c102
$\{26, 80, 95, 102\}$	x103 + x104	c117	$\{67, 88, 95, 106\}$	1 + x87 + x89	c5
$\{30, 72, 73, 85\}$	x32	c75	$\{29, 34, 35, 112\}$	1 + x27 + x28	c36
$\{4, 23, 50, 92\}$	1 + x3 + x5 + x21	c9	$\{17, 67, 68, 103\}$	1 + x102 + x104	c108
$\{39, 43, 68, 71\}$	x41	c34	$\{17, 59, 100, 113\}$	1 + x15 + x16	c30
$\{76, 105, 118, 119\}$	x106 + x107	c0	$\{48, 77, 106, 107\}$	x50	c18
$\{76, 77, 104, 108\}$	1 + x109	c35	$\{17, 46, 86, 105\}$	x107	c42
$\{33, 49, 71, 80\}$	x34 + x35	c46	$\{12, 20, 89, 101\}$	x14	c20
$\{11, 40, 41, 42\}$	x44	c72	$\{13, 20, 41, 70\}$	1 + x69	c4
$\{13, 29, 73, 107\}$	x27	c48	$\{46, 79, 82, 104\}$	1 + x45	c79
$\{16, 50, 76, 90\}$	x92	c111	$\{8, 56, 91, 107\}$	1 + x6 + x7	c27
$\{11, 43, 80, 107\}$	1 + x105 + x106	c102	$\{1, 26, 85, 93\}$	1 + x0 + x2 + x94	c119

(*continued*)

Table 1. (*continued*)

Cube idices	Expression	Out. bit	Cube indices	Expression	Out. bit
$\{34, 59, 91, 111\}$	1 + x112	c25	$\{16, 73, 104, 109\}$	1 + x108 + x110	c9
$\{51, 65, 97, 104\}$	1 + x52	c104	$\{77, 92, 116, 118\}$	1 + x114 + x115	c33
$\{21, 38, 79, 92\}$	1 + x22	c33	$\{70, 73, 90, 101\}$	1 + x91	c67
$\{14, 36, 40, 119\}$	x37 + x38	c9	$\{7, 54, 55, 74\}$	x72	c117
$\{77, 113, 117, 119\}$	x111	c102	$\{35, 41, 66, 73\}$	x68	c117
$\{17, 19, 61, 86\}$	1 + x18	c52	$\{7, 23, 35, 44\}$	x21	c19
$\{11, 38, 114, 116\}$	1 + x36 + x37	c39	$\{10, 11, 83, 88\}$	x81	c72
$\{15, 16, 43, 89\}$	1 + x42 + x44	c30	$\{11, 14, 36, 38\}$	1 + x9 + x10	c9
$\{11, 62, 77, 117\}$	x119	c72	$\{7, 28, 94, 115\}$	1 + x93 + x114	c18
$\{6, 13, 72, 74\}$	x8	c117	$\{20, 51, 53, 82\}$	1 + x81 + x83	c96
$\{8, 26, 48, 50\}$	1 + x24 + x25	c33	$\{57, 65, 91, 97\}$	1 + x58	c21
$\{5, 9, 74, 80\}$	1 + x10	c36	$\{1, 32, 64, 98\}$	1 + x0 + x2	c39
$\{17, 83, 95, 115\}$	x15	c36	$\{11, 31, 73, 80\}$	1 + x30	c64
$\{4, 8, 65, 77\}$	x63	c18	$\{24, 26, 39, 85\}$	1 + x40	c69
$\{35, 37, 116, 119\}$	1 + x117 + x118	c57	$\{49, 58, 86, 111\}$	x112 + x113	c49
$\{47, 50, 73, 116\}$	1 + x72 + x74	c82	$\{24, 25, 96, 103\}$	x98	c87
$\{40, 41, 66, 110\}$	1 + x67	c27	$\{50, 70, 71, 87\}$	x89	c17
$\{10, 22, 28, 101\}$	1 + x9 + x11	c18	$\{26, 52, 53, 96\}$	1 + x97	c54
$\{34, 56, 76, 88\}$	1 + x33	c12	$\{42, 47, 88, 113\}$	1 + x43	c79
$\{23, 56, 58, 80\}$	x54	c55	$\{37, 38, 71, 72\}$	1 + x73	c99
$\{17, 19, 49, 97\}$	1 + x18 + x20	c93	$\{32, 50, 55, 56\}$	1 + x48 + x49	c57
$\{50, 60, 82, 103\}$	1 + x61	c3	$\{20, 61, 68, 99\}$	x101	c52
$\{15, 41, 76, 82\}$	x17	c12	$\{8, 13, 59, 97\}$	1 + x12 + x14	c33
$\{25, 35, 61, 86\}$	1 + x60 + x62	c59	$\{31, 32, 54, 59\}$	x55 + x56	c18

References

1. Albrecht, M.: Algebraic attacks on the courtois toy cipher. Master thesis. Department of Computer Science. University of Bremen (2006)
2. Aumasson, J.-P., Dinur, I., Henzen, L., Meier, W., Shamir, A.: Efficient FPGA implementations of high-dimensional cube teters on the stream cipher grain-128. IACR Cryptology ePrint Archive, 2009/218
3. Aumasson, J.-P., Dinur, I., Meier, W., Shamir, A.: Cube testers and key recovery attacks on reduced-round MD6 and trivium. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 1–22. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03317-9_1
4. Blum, M., Luby, M., Rubinfeld, R.: Self-testing/correcting with applications to numerical problems. J. Comput. Syst. Sci. **47**, 549–595 (1993)
5. Courtois, N.: How fast can be algebraic attacks on block ciphers? IACR Cryptology ePrint Archive, 2006/168
6. Courtois, N.: CTC2 and fast algebraic attacks on block ciphers revisited. IACR Cryptology ePrint Archive, 2007/152

7. De Canniere, C., Preneel, B.: Trivium: ecrypt stream cipher project. http://www.ecrypt.eu.org/stream/
8. Dinur, I., Shamir, A.: Cube attacks on tweakable black box polynomials. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 278–299. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01001-9_16
9. Dinur, I., Shamir, A.: Side channel cube attacks on block ciphers. IACR Cryptology ePrint Archive, 2009/127
10. Dunkelman, O., Keller, N.: Linear cryptanalysis of CTC. IACR Cryptology ePrint Archive, 2006/250
11. Dunkelman, O., Keller, N.: Cryptanalysis of CTC2. In: Fischlin, M. (ed.) CT-RSA 2009. LNCS, vol. 5473, pp. 226–239. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00862-7_15
12. Mroczkowski, P., Szmidt, J.: The cube attack on courtois toy cipher. IACR Cryptology ePrint Archive, 2009/497
13. Vielhaber, M.: Breaking one. fivium by AIDA an algebraic IV differential attack, IACR Cryptology ePrint Archive, 2007/413
14. Vielhaber, M.: AIDA braeks BIVIUM (A and B) in 1 Minute dual core CPU time. IACR Cryptology ePrint Archive, 2009/402
15. Magma V2.14-17: Computational Algebra Group. School of Mathematics and Statistics. University of Sydney. http://magma.maths.usyd.edu.au
16. Python Programming Language. http://www.python.org
17. SAGE Mathematical Software. Version 2.6. http://www.sagemath.org, http://polybori.sourceforge.net/

Near Butson-Hadamard Matrices and Nonlinear Boolean Functions

Sibel Kurt and Oğuz Yayla(✉)

Department of Mathematics, Hacettepe University Beytepe, 06800 Ankara, Turkey
sibelkurt3211@gmail.com, oguz.yayla@hacettepe.edu.tr

Abstract. A Hadamard matrix is a square matrix with entries ±1 whose rows are orthogonal to each other. Hadamard matrices appear in various fields including cryptography, coding theory, combinatorics etc. This study takes an interest in γ near Butson-Hadamard matrix that is a generalization of Hadamard matrices for $\gamma \in \mathbb{R} \cap \mathbb{Z}[\zeta_m]$. These matrices are examined in this study. In particular, the unsolvability of certain equations is studied in the case of cyclotomic number fields. Winterhof et al. considered the equations for $\gamma \in \mathbb{Z}$, and by the authors for $\gamma \in \mathbb{R} \cap \mathbb{Z}[\zeta_m]$. In this study, we obtain another method for checking the nonexistence cases of these equations, which uses the tool of norm from algebraic number theory. Then, the direct applications of these results to γ near Butson-Hadamard matrices are obtained. In the second part of this study, the connection between nonlinear Boolean cryptographic functions and γ near Butson-Hadamard matrices having small $|\gamma|$ is established. In addition, a computer search is done for checking the cases which are excluded by our results and for obtaining new examples of existence parameters.

Keywords: Butson-Hadamard matrices · Cyclotomic fields
Cryptographic functions

1 Introduction

Hadamard matrices are used in computational mathematics and quantum computer science. They have also been used in many practical areas e.g. cryptographic function design, telecommunication satellites, modern cell phones and wireless networks. Modern CDMA based cell phones use Hadamard matrices to modulate the signals and to minimize the interference between signals arriving the base station. Information hiding in wireless networks, optical telecommunication, neuroscience and pattern recognition are other practical areas where Hadamard matrices are used. In addition, Hadamard matrices are directly applied in computer science, for example, Hadamard codes (known as best error correcting codes) and Hadamard gates (used in quantum gates), see [4,5] for details and other applications.

In this study, a class of Butson-Hadamard matrices is studied and its application to cryptographic Boolean functions is investigated. Very recently, new

J. Kaczorowski et al. (Eds.): NuTMiC 2017, LNCS 10737, pp. 254–266, 2018.
https://doi.org/10.1007/978-3-319-76620-1_15

properties of m-ary γ near Butson-Hadamard matrices for $\gamma \in \mathbb{Z}$ are studied in [7,13]. We study m-ary γ near Butson-Hadamard matrices for $\gamma \in (\mathbb{Z}[\zeta_m] \cap \mathbb{R})$, and look for new examples of near Butson-Hadamard matrix and their existence requirements, where $m \in \mathbb{Z}^+$ and ζ_m is a primitive m-th root of unity. We use the methods in algebraic number theory and results in [13] to find new results on m-ary γ near Butson-Hadamard matrices. Moreover, these new results on Hadamard matrices are used in the investigation of new applications in cryptography.

A *Hadamard matrix* is a square matrix with entries ±1 whose rows are orthogonal to each other. First generalization of Hadamard matrices was made by Butson in 1962. Butson [3] handled complex m-th root of unity for entries of Hadamard matrices, instead of second root of unity. γ *near Butson-Hadamard matrices* are similar to Butson-Hadamard matrices, except inner product of a row with a complex conjugate of another row is γ. The most common result for Butson-Hadamard matrices is presented by [2,7,13].

Winterhof et al. [13] reduce the existence condition of a γ near Butson Hadamard matrix to an equation over ring of integers of a cyclotomic field having class number $h_m > 1$. Namely, they consider the solutions $\alpha \in \mathbb{Z}[\zeta_m]$ of the following equation

$$\alpha\bar{\alpha} = ((\gamma + 1)v - \gamma)(v - \gamma)^{v-1}, \tag{1}$$

where $v \in \mathbb{Z}^+$ is the dimension of the γ near Butson Hadamard matrix and $\gamma \in \mathbb{Z}$. Then they consider the principal ideal factorization of $D = ((\gamma+1)v-\gamma)(v-\gamma)^{v-1}$ and deal with the unsolvability conditions of (1). They only consider integer γ for ideal factorization of D. Then, the authors in [7] extended this method for $\gamma \in \mathbb{Z}[\zeta_m] \cap \mathbb{R}$. It was shown that the norm of nonprincipal part of D is to be relatively prime to the norm of principal part of D for the unsolvability of (1) in case $\gamma \in \mathbb{Z}[\zeta_m] \cap \mathbb{R}$ (see Theorem 2 herein).

In Sect. 3, a new method is built up for checking the cases in which a near Butson-Hadamard matrix does not exist. This method does not depend on the class number h_m. We use the fact that the norm of a prime ideal dividing α in (1), also divides $\bar{\alpha}$. Therefore, for any prime ideal $\mathfrak{p}$ dividing D, if the norm of D divided by the norm of $\mathfrak{p}$ is relatively prime to the norm of $\mathfrak{p}$, then (1) has no solution (see Theorem 3). In particular, if the norm of D is square-free then it is clear that (1) has no solution (see Corollary 1). In addition, we perform an exhaustive computer search by using MAGMA [1] on the set $v \in \{2, 3, \ldots, 100\}$ for fixed m and ζ and for the nonexistence of the equation (1) to see the strength of Theorems 2 and 3. It is seen that Theorems 2 and 3 exclude the existence of many values, on the other hand, we see that they do not cover each other.

We applied our new result (Theorem 3) to γ near Butson-Hadamard matrices in Sect. 4. Namely, we look for dimension $v \in \mathbb{Z}$ and $\gamma \in \mathbb{Z}[\zeta_m] \cap \mathbb{R}$ for which a γ near Butson-Hadamard matrix does not exist. This is equivalent to finding necessary conditions for solvability of $\alpha\bar{\alpha} = ((\gamma + 1)v - \gamma)(v - \gamma)^{v-1}$ for some $\alpha \in \mathbb{Z}[\zeta_m]$. Hence, by using our main theorems we obtain nonexistence results for γ near Butson-Hadamard matrices (see Corollary 2).

Finally, in this study, the relationship between a Butson-Hadamard matrix and a cryptographic function is investigated. In cryptography, secrecy (or confidentiality) is satisfied by using block ciphers which confuses a message into a ciphertext via a nonlinear *Boolean function.* A nonlinear Boolean function attaining the maximum nonlinearity is called a *bent function.* It is known that a Butson-Hadamard matrix is equivalent to cryptographic bent function (see [6] or Theorem 6 in this study). By using this equivalence, we convert a γ near Butson-Hadamard matrix into a Boolean function in Sect. 5. It is seen that one can find a highly nonlinear Boolean function via circulant γ near Butson-Hadamard matrices having very small $|\gamma|$ values. The existence cases of circulant γ near Butson-Hadamard matrices for small $|\gamma|$ is considered in this study. Hence, we perform an exhaustive computer search by using MAGMA [1] on dimension $v \in \{1, 2, \ldots, 11\}$ and alphabet $m \in \{1, 2, \ldots, 11\}$, and look for γ (see Table 2).

The outline of this study is as follows. In Sect. 2, previous studies, the methods based on self conjugacy condition and principal ideal factorization are presented. In Sect. 3, a new result for deciding the nonexistence of a solution to (1) is given. Then, in Sect. 4, the consequences of the results given in Sect. 3 are applied to Hadamard matrices. Next, a cryptographic application of the results given Sect. 4 is presented in Sect. 5.

2 Previous Results

We first give definitions of the *norm of an element* in a number field and *norm of an ideal* of the ring of integers of a number field.

Definition 1 [11, p. 49]. *Let $K = \mathbb{Q}(\theta)$ be a number field of degree n and let $\sigma_1, \ldots, \sigma_n$ be monomorphisms $K \to \mathbb{C}$. $\alpha \in K$ is an algebraic integer. For any $\alpha \in K$, we define the norm.*

$$N_K(\alpha) = \prod_{i=1}^{n} \sigma_i(\alpha)$$

Definition 2 [11, p. 115]. *Let $\mathcal{O}_K$ be the ring of unit of a number field K and I be non-zero ideal of $\mathcal{O}_K$, we define the norm of I by*

$$N(I) = |\mathcal{O}_{\mathcal{K}}|.$$

We note that if $\mathfrak{a} = \langle a \rangle$ is a principal ideal then $N(\mathfrak{a}) = \langle N(a) \rangle$ [11, Corollary 5.10]. If $\mathfrak{a}|\mathfrak{b}$ then $N(\mathfrak{a})|N(\mathfrak{b})$ [11, Theorem 5.12]. For an ideal $\mathfrak{a}$, its conjugate ideal is $\bar{\mathfrak{a}} := \{\bar{\alpha} : \alpha \in \mathfrak{a}\}$. It can be seen that $N(\mathfrak{a}) = N(\bar{\mathfrak{a}})$ and if $\mathfrak{a}$ is a prime ideal, then $\bar{\mathfrak{a}}$ is also prime ideal.

We now present non-existence results on γ near Butson-Hadamard matrices based on results of Brock [2], see also [8], for $\gamma \in \mathbb{Z}$.

Let p be a prime and m a positive integer with $\gcd(p, m) = 1$. We say that p is *self-conjugate* modulo m if the order f of p modulo m is even and $p^{f/2} \equiv -1 \bmod m$.

Table 1. The class number h_m of $\mathbb{Q}(\zeta_m)$ for $m \leq 70$ [12].

m	h_m	m	h_m	m	h_m	m	h_m	m	h_m	m	h_m	m	h_m
1	1	11	1	21	1	31	9	41	121	51	5	61	76301
2	1	12	1	22	1	32	1	42	1	52	3	62	9
3	1	13	1	23	3	33	1	43	211	53	48891	63	7
4	1	14	1	24	1	34	1	44	1	54	1	64	17
5	1	15	1	25	1	35	1	45	1	55	10	65	64
6	1	16	1	26	1	36	1	46	3	56	2	66	1
7	1	17	1	27	1	37	37	47	695	57	9	67	853513
8	1	18	1	28	1	38	1	48	1	58	8	68	8
9	1	19	1	29	8	39	2	49	43	59	41421	69	69
10	1	20	1	30	1	40	1	50	1	60	1	70	1

Theorem 1 [2, Theorem 3.1]. *For a positive integer w there exists no solution α to the equation $\alpha\overline{\alpha} = w$ over $\mathbb{Q}[\zeta_m]$ if the square-free part of w is divisible by a prime which is self-conjugate modulo m.*

We now consider the equation $D = \alpha\bar{\alpha}$ over $\mathbb{Z}[\zeta_m]$ for some $m \in \mathbb{Z}^+$ and $D \in \mathbb{Z}[\zeta_m] \cap \mathbb{R}$. Winterhof et. al. [13] presented a condition for the non-existence of a solution $\alpha \in \mathbb{Z}[\zeta_m]$ to this equation for $D \in \mathbb{Z}$, then it was extended to $D \in \mathbb{Z}[\zeta_m] \cap \mathbb{R}$ in [7]. In particular, we consider

$$D = ((\gamma + 1)v - \gamma)(v - \gamma)^{v-1}$$

for some $m, v \in \mathbb{Z}^+$ and $\gamma \in \mathbb{Z}[\zeta_m] \cap \mathbb{R}$, which we get in case of proving nonexistence of some Butson-Hadamard matrices in Sect. 5.

The main theorem of [7] that there is no solution on $D = \alpha\bar{\alpha} \in \mathbb{Z}[\zeta_m]$ for some $\gamma \in \mathbb{Z}[\zeta_m]$ is given below. For completeness, the outline of the proof is given below. Let h_m denote the class numbers of cyclotomic number field $\mathbb{Q}(\zeta_m)$. We list h_m for $m \leq 70$ in Table 1.

Theorem 2 [7]. *Let $D \in \mathbb{Z}[\zeta_m] \cap \mathbb{R}$ such that $D = tq^{2e+1}$ where $q, t \in \mathbb{Z}[\zeta_m]$ and q is square-free, provided that every prime ideal $\mathfrak{t} \lhd \mathbb{Z}[\zeta_m]$ with $\mathfrak{t}|(t)$ is principal, $(q) = \mathfrak{q}_1\mathfrak{q}_2$ where $\mathfrak{q}_1$ and $\mathfrak{q}_2$ are non-principal prime ideals of $\mathbb{Z}[\zeta_m]$, $e > 0$ be rational integer, $gcd(2e + 1 - 2k, h_m) = 1$ for $0 \leq k \leq e - 1$ and $gcd(N(q), N(t)) = 1$. Then, there exists no $\alpha \in \mathbb{Z}[\zeta_m]$ satisfying $D = \alpha\bar{\alpha}$.*

Proof. We first assume that there exists $\alpha \in \mathbb{Z}[\zeta_m]$ such that $\alpha\bar{\alpha} = tq^{2e+1}$ such that

$$(\alpha) = \mathfrak{t}_1\mathfrak{q}_1^{2e+1-k}\mathfrak{q}_2^k$$

$$(\bar{\alpha}) = \mathfrak{t}_2\mathfrak{q}_1^k\mathfrak{q}_2^{2e+1-k}$$

for some $\mathfrak{t} \lhd \mathbb{Z}[\zeta_m]$. We have

$$(\alpha) = \mathfrak{t}_1\mathfrak{q}_1^{2e+1-k}\mathfrak{q}_2^k = \mathfrak{t}_1\mathfrak{q}_1^{2e+1-2k}q^k$$

We know that t_1 and q are principal ideals of $\mathbb{Z}[\zeta_m]$ but $\mathfrak{q}_1^{2e+1-2k}$ is nonprincipal since $gcd(2e+1-2k, h_m) = 1$. Hence we get a contradiction. Next, we assume that $\alpha = \mathfrak{t}_1 q^s$, $\bar{\alpha} = \mathfrak{t}_2 q^{2e+1-s}$ for some principal ideals $\mathfrak{t}_1, \mathfrak{t}_2 \lhd \mathbb{Z}[\zeta_m]$ and $s \in \mathbb{Z}^+ \cup \{0\}$, $s \leq e$. Then, $q^{2e+1-2s} | \mathfrak{t}_1$. However, this contradicts to $gcd(N(q), N(\mathfrak{t})) = 1$. □

We now give an example of Theorem 2.

Example 1. Let $D = ((-\zeta_{23} - \zeta_{23}^{22})5 + 1 + \zeta_{23} + \zeta_{23}^{22})(6 + \zeta_{23} + \zeta_{23}^{22})^4 \in \mathbb{Z}[\zeta_{23}]$ be obtained by setting $v = 5$, $m = 23$, $\gamma = -1 - \zeta_{23} - \zeta_{23}^{22}$. D has two non-principal prime ideals such that $D = \mathfrak{p}_1^4 \mathfrak{p}_2 \mathfrak{p}_3^4 \mathfrak{q}_4 \mathfrak{q}_5$ where $\mathfrak{p}_1, \mathfrak{p}_2, \mathfrak{p}_3 \lhd \mathbb{Z}[\zeta_{23}]$ are principal prime ideals and $\mathfrak{q}_4, \mathfrak{q}_5 \in \mathbb{Z}[\zeta_{23}]$ are the non-principal prime ideals. By Theorem 2 we say that there is no $\alpha \in \mathbb{Z}[\zeta_m]$ satisfying $D = \alpha\bar{\alpha}$.

We note that the method given in Theorem 2 to the case that q has more than two non-principal ideals factors does not work.

Example 2. Let $D = ((-\zeta_{23} - \zeta_{23}^{22})46 + 1 + \zeta_{23} + \zeta_{23}^{22})(47 + \zeta_{23} + \zeta_{23}^{22})^{45} \in \mathbb{Z}[\zeta_{23}]$ be obtained by setting $v = 46$, $m = 23$, $\gamma = -1 - \zeta_{23} - \zeta_{23}^{22}$. D has four non-principal prime ideals such that $D = \mathfrak{p}_1 \mathfrak{p}_2^{45} \mathfrak{p}_3^{45} \mathfrak{q}_4 \mathfrak{q}_5 \mathfrak{q}_6 \mathfrak{q}_7$ where $\mathfrak{p}_1, \mathfrak{p}_2, \mathfrak{p}_3 \lhd \mathbb{Z}[\zeta_{23}]$ are principal prime ideals and $\mathfrak{q}_4, \mathfrak{q}_5, \mathfrak{q}_6, \mathfrak{q}_7 \lhd \mathbb{Z}[\zeta_{23}]$ are the non-principal ideals. The methodology in Example 1 does not work for this example. Note that $(\alpha) = \mathfrak{t}_1 \mathfrak{q}_{10} \mathfrak{q}_{12}^{38}$ is a principal ideal and satisfies $D = \alpha\bar{\alpha}$ for a convenient principal ideal $\mathfrak{t}_1 \lhd \mathbb{Z}[\zeta_{23}]$ such that $t_1 \mid D$.

Example 3. Let $D = ((-\zeta_{23} - \zeta_{23}^{22})39 + 1 + \zeta_{23} + \zeta_{23}^{22})(40 + \zeta_{23} + \zeta_{23}^{22})^{38} \in \mathbb{Z}[\zeta_{23}]$ be obtained by setting $v = 39$, $m = 23$, $\gamma = -1 - \zeta_{23} - \zeta_{23}^{22}$. D has four prime non-principal ideals such that $D = \mathfrak{p}_1 \mathfrak{p}_2 \mathfrak{p}_3 \mathfrak{p}_4 \mathfrak{p}_5^2 \mathfrak{p}_6^{38} \mathfrak{p}_7^{38} \mathfrak{p}_8^{38} \mathfrak{q}_9 \mathfrak{q}_{10} \mathfrak{q}_{11}^{38} \mathfrak{q}_{12}^{38}$ where $\mathfrak{p}_1, \mathfrak{p}_2, \mathfrak{p}_3, \mathfrak{p}_4, \mathfrak{p}_5, \mathfrak{p}_6, \mathfrak{p}_7, \mathfrak{p}_8 \lhd \mathbb{Z}[\zeta_{23}]$ are principal ideals and $\mathfrak{q}_9, \mathfrak{q}_{10}, \mathfrak{q}_{11}, \mathfrak{q}_{12} \lhd \mathbb{Z}[\zeta_{23}]$ are non-principal ideals. Note that $(\alpha) = \mathfrak{t}_1 \mathfrak{q}_{10} \mathfrak{q}_{12}^{38}$ is a principal ideal and satisfies $D = \alpha\bar{\alpha}$ for a convenient principal ideal $\mathfrak{t}_1 \lhd \mathbb{Z}[\zeta_{23}]$ such that $t_1 \mid D$.

In order to speak of the non-existence of a solution to the equation $D = \alpha\bar{\alpha}$ for $\alpha \in \mathbb{Z}[\zeta_m]$ with D is divisible by more than two non-principal ideals, one can consider principal parts produced by the non-principal ones. We remark this method below.

Remark 1. If D is divisible by four non-principal prime ideals which are distinct and relatively prime to each other, then there exists no solution $\alpha \in \mathbb{Z}[\zeta_m]$ satisfying $D = \alpha\bar{\alpha}$. In other words, let $q_1, q_2, q_3, q_4 \lhd \mathbb{Z}[\zeta_m]$ be non-principal prime ideals of $\mathbb{Z}[\zeta_m]$ dividing D. Assume that $q_1 q_2, q_3 q_4, q_1 q_3, q_2 q_4$ are all principal in $\mathbb{Z}[\zeta_m]$. If $gcd(N(q_1 q_2), N(q_3 q_4)) = 1$, $gcd(N(q_1 q_3), N(q_2 q_4)) = 1$, then we can conclude that there exists no solution.

3 Norm Method

The method presented in Theorem 2 works only for $h_m > 1$, where h_m denotes the class numbers of cyclotomic number field $\mathbb{Q}(\zeta_m)$. In this section, we present a new method for deciding an existence of a solution $\alpha \in \mathbb{Z}[\zeta_m]$ to the equation $D = \alpha\bar{\alpha}$ where $m \in \mathbb{Z}^+$ and $D \in \mathbb{Z}[\zeta_m] \cap \mathbb{R}$, which does not depend on h_m. In particular, we consider

$$D = ((\gamma + 1)v - \gamma)(v - \gamma)^{v-\gamma}$$

for some $m, v \in \mathbb{Z}^+$ and $\gamma \in \mathbb{Z}[\zeta_m] \cap \mathbb{R}$, which we get in case of proving nonexistence of some Butson-Hadamard matrices in Sect. 5.

Theorem 3. *Let $p \lhd \mathbb{Z}[\zeta_m]$ be a prime ideal with $\mathfrak{p}|D$ and $gcd(N(D)/N(\mathfrak{p}), N(\mathfrak{p})) = 1$. Then there is no solution $\alpha \in \mathbb{Z}[\zeta_m]$ satisfying $D = \alpha\bar{\alpha}$.*

Proof. Assume $\alpha \in \mathbb{Z}[\zeta_m]$ is a solution of $D = \alpha\bar{\alpha}$ and $\mathfrak{p} \lhd \mathbb{Z}[\zeta_m]$ is a prime ideal factor of α. We know that if $\mathfrak{p} \mid D$, then $N(\mathfrak{p}) \mid N(D)$. We have $N(\mathfrak{p}) \nmid \frac{N(D)}{N(\mathfrak{p})}$ since $gcd(N(D)/N(\mathfrak{p}), N(\mathfrak{p})) = 1$. By $N(\mathfrak{p}) = N(\bar{\mathfrak{p}})$, we have $N(\bar{\mathfrak{p}}) \nmid \frac{N(D)}{N(\mathfrak{p})}$. Hence, $N(\mathfrak{p})N(\bar{\mathfrak{p}}) \nmid N(D)$. This is a contradiction to $D = \alpha\bar{\alpha}$. □

There is an immediate consequence of Theorem 3.

Corollary 1. *If the norm of non-principal part of D is square-free, then there exists no $\alpha \in \mathbb{Z}[\zeta_m]$ satisfying $D = \alpha\bar{\alpha}$.*

Next, we give an example of Theorem 3. Below, we consider $D = ((\gamma+1)v - \gamma)(v - \gamma)^{v-1}$ for some $m, v \in \mathbb{Z}^+$ and $\gamma \in \mathbb{Z}[\zeta_m] \cap \mathbb{R}$.

Example 4. Let be $v = 30$, $m = 23$, $\gamma = -1 - \zeta_{23} - \zeta_{23}^{22}$. Then $D = ((-\zeta_{23} - \zeta_{23}^{22})39 + 1 + \zeta_{23} + \zeta_{23}^{22})(40 + \zeta_{23} + \zeta_{23}^{22})^{38} \in \mathbb{Z}[\zeta_{23}]$ has four non-principal prime ideal factors, such that $D = \mathfrak{p}_1\mathfrak{p}_2^{29}\mathfrak{q}_3\mathfrak{q}_4\mathfrak{q}_5^{29}\mathfrak{q}_6^{29}$ where $\mathfrak{p}_1, \mathfrak{p}_2 \lhd \mathbb{Z}[\zeta_{23}]$ are principal prime ideals and $\mathfrak{q}_1, \mathfrak{q}_2, \mathfrak{q}_3, \mathfrak{q}_4 \lhd \mathbb{Z}[\zeta_{23}]$ are non-principal prime ideals. Then,

$$N(D) = 47^{58} \cdot 229^2 \cdot 63276304836881^2 \cdot 517725371091023^2, N(p_1) = 229^2$$

and

$$gcd(\frac{47^{58} \cdot 229^2 \cdot 63276304836881^2 \cdot 517725371091023^2}{229^2}, 229^2) = 1.$$

Hence, we say that there is no $\alpha \in \mathbb{Z}[\zeta_{23}]$ satisfying $D = \alpha\bar{\alpha}$ by Theorem 3.

We note that Theorem 3 does not completely cover Theorem 2 and vice versa. For $\gamma = -1 - \zeta_{23} - \zeta_{23}^{22}$ and $m = 23$, the existence of a solution to the equation $D = \alpha\bar{\alpha}$ over $\mathbb{Z}[\zeta_{23}]$ for $v \in \{8, 26\}$ can be excluded by Theorem 2, but Theorem 3. On the other hand, the existence of a solution to the equation $D = \alpha\bar{\alpha}$ over $\mathbb{Z}[\zeta_{23}]$ for $v \in \{9, 10, 11, 12, 13, 14\}$ can be excluded by Theorem 3, but Theorem 2. Therefore, the two theorems do not cover each other, but they intersect.

4 Application to Butson-Hadamard Matrices

In this section, we give the definition of a Butson-Hadamard matrix, and apply the results of the previous sections to this kind of matrices.

A *Hadamard matrix* is an $(v \times v)$ square matrix with entries 1 or -1 satisfying $H\overline{H}^T = vI$. Two examples of Hadamard matrices are given below.

$$A = \begin{bmatrix} 1 & 1 \\ 1 & - \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & - & - \\ 1 & - & 1 & - \\ 1 & - & - & 1 \end{bmatrix}$$

A square matrix $H = (h_{ij})$ of order v is called *circulant* if $h_{i+1 \bmod v, j+1 \bmod v} = h_{i,j}$ for all $0 \leq i, j < v$. An example of a circulant matrix H is given below.

$$H = \begin{bmatrix} 1 & 1 & - & - & - \\ - & 1 & 1 & - & - \\ - & - & 1 & 1 & - \\ - & - & - & 1 & 1 \\ 1 & - & - & - & 1 \end{bmatrix}$$

For an integer $m \geq 2$, let ζ_m denote a primitive complex m-th root of unity and let $\mathcal{E}_m = \{1, \zeta_m, \zeta_m^2, \ldots, \zeta_m^{m-1}\}$. The identity matrix is denoted by I and all one matrix is denoted by J.

Definition 3. *A Butson-Hadamard matrix is a square matrix H of order v with entries in $\mathcal{E}_m$ such that $H\overline{H}^T = vI$. It is denoted by* $\mathrm{BH}(v, m)$. $\mathrm{BH}(v, 2)$ *is so called* Hadamard matrix *of order v. In general, a γ near Butson-Hadamard matrix is a square matrix H of order v with entries in $\mathcal{E}_m$ such that $H\overline{H}^T = (v - \gamma)I + \gamma J$ for a $\gamma \in \mathbb{R} \cap \mathbb{Z}[\zeta_m]$. Similarly, it is denoted by* $\mathrm{BH}_\gamma(v, m)$.

Two examples on the existence of γ near Butson-Hadamard matrices are presented below.

Example 5. $\mathrm{BH}_\gamma(5{,}5)$ exists for $\gamma \in \{-\xi_5^3 - \xi_5^2 + 2, 0, 5, \xi_5^3 + \xi_5^2 + 3\}$ with $|\gamma| \in \{1.38, 0, 5, 3.61\}$, respectively. For instance, the matrix H has $\gamma = -\xi_5^3 - \xi_5^2 + 2$ with $|\gamma| = 1.38$

$$H = \begin{bmatrix} 1 & 1 & -\xi_5^2 & 1 & 1 \\ 1 & 1 & 1 & -\xi_5^2 & 1 \\ 1 & 1 & 1 & 1 & -\xi_5^2 \\ -\xi_5^2 & 1 & 1 & 1 & 1 \\ 1 & -\xi_5^2 & 1 & 1 & 1 \end{bmatrix}.$$

Example 6. Similarly, we obtained by an exhaustive search that $\mathrm{BH}_\gamma(8{,}5)$ exists for $\gamma \in \{-\xi_5^3 - \xi_5^2 + 5, -\xi_5^3 - \xi_5^2, 8, \xi_5^3 + \xi_5^2 + 1, \xi_5^3 + \xi_5^2 + 6\}$ with $|\gamma| \in$

$\{6.61, 1.61, 8, 0.61, 4.38\}$, respectively. In particular, the matrix H has $\gamma = -\xi_5^3 - \xi_5^2 + 2$ with $|\gamma| = 0.61$

$$H = \begin{bmatrix} 1 & 1 & \zeta_5^2 & \zeta_5^3 & 1 & \zeta_5^3 & \zeta_5 & 1 \\ 1 & 1 & 1 & \zeta_5^2 & \zeta_5^3 & 1 & \zeta_5^3 & \zeta_5 \\ \zeta_5 & 1 & 1 & 1 & \zeta_5^2 & \zeta_5^3 & 1 & \zeta_5^3 \\ \zeta_5^3 & \zeta_5 & 1 & 1 & 1 & \zeta_5^2 & \zeta_5^3 & 1 \\ 1 & \zeta_5^3 & \zeta_5 & 1 & 1 & 1 & \zeta_5^2 & \zeta_5^3 \\ \zeta_5^3 & 1 & \zeta_5^3 & \zeta_5 & 1 & 1 & 1 & \zeta_5^2 \\ \zeta_5^2 & \zeta_5^3 & 1 & \zeta_5^3 & \zeta_5 & 1 & 1 & 1 \\ 1 & \zeta_5^2 & \zeta_5^3 & 1 & \zeta_5^3 & \zeta_5 & 1 & 1 \end{bmatrix}.$$

We now investigate a property that a γ near Butson-Hadamard matrix H satisfy. It is clear that $\det(H) \in \mathbb{Z}[\zeta_m]$ and we have the following equalities:

$$\begin{aligned} H\overline{H}^T &= (v-\gamma)I + \gamma J, \\ \det(H\overline{H}^T) &= \det((v-\gamma)I + \gamma J) \\ \det(H)\det(\overline{H}) &= ((\gamma+1)v - \gamma)(v-\gamma)^{v-1}. \end{aligned}$$

Therefore, a $\mathrm{BH}_\gamma(v,m)$ exists then the following equation has a solution $\alpha \in \mathbb{Z}[\zeta_m]$

$$\alpha\overline{\alpha} = ((\gamma+1)v - \gamma)(v-\gamma)^{v-1}. \tag{2}$$

Now, we apply Theorem 3 to (2), we get a criterion for the non-existence of $\mathrm{BH}_\gamma(v,m)$:

Corollary 2. *Let $v, m \in \mathbb{Z}^+$ and $\gamma \in \mathbb{Z}[\zeta_m] \cap \mathbb{R}$ such that $D = ((\gamma+1)v-\gamma)(v-\gamma)^{v-1}$ and $p \lhd \mathbb{Z}[\zeta_m]$ be a prime ideal with $\mathfrak{p}|D$ and gcd$(N(D)/N(\mathfrak{p}), N(\mathfrak{p})) = 1$. Then, there exists no* $\mathrm{BH}_\gamma(v,m)$.

We give an example illustrating the results above.

Example 7. Consider $\mathrm{BH}_\gamma(39,23)$, $\gamma = -1 - \zeta_{23} - \zeta_{23}^{22}$, $v = 39$ and $m = 23$. Hence we have

$$\alpha\overline{\alpha} = ((-\zeta_{23} - \zeta_{23}^{22})39 + 1 + \zeta_{23} + \zeta_{23}^{22})(40 + \zeta_{23} + \zeta_{23}^{22})^{38}.$$

Then we conclude that $\mathrm{BH}_\gamma(39,23)$ does not exist by Corollary 2 and Example 4.

Remark 2. We performed an exhaustive computer search by using MAGMA [1] to check the cases for which Corollary 2 excludes the existence of a $\mathrm{BH}_\gamma(v,m)$. We fixed $m = 23$, $\gamma = -1 - \zeta_{23} - \zeta_{23}^{22}$ and searched on the set $v \in \{2, 3, \ldots, 100\}$. We obtained that Corollary 2 excludes the existence of a $\mathrm{BH}_\gamma(v,m)$ for all $v = \{2, 3, \ldots, 100\}$ except $\{6, 8, 15, 16, 26, 44, 49, 62, 67, 75, 84, 85, 88, 94\}$.

5 Nonlinear Boolean Functions

There is a close relationship between the family of Hadamard matrices and cryptography. For instance there is a class of functions called bent function used in block cipher cryptosystems, and they can be constructed via Butson-Hadamard matrices.

Functions used in block cipher design have to satisfy some properties in order to resist attacks. Two of them are balancedness and nonlinearity. A function is said to be balanced if each value in its image set is attained by the same probability. And, a function's nonlinearity is measured by its minimum distance to all linear functions.

The family of bent functions is a branch of the Boolean functions. Their Walsh spectrum coefficients allow us to examine their non-linearity. Hence, we start with the definition of a Boolean function.

Definition 4. *A function $f : (\mathbb{Z}_2)^n \rightarrow \mathbb{Z}_2$ is a Boolean function of n variables. Let B_n be the set of all Boolean functions of n variables. A function $f \in B_n$ is represented with a vector of length 2^n having values $f(x)$ for all $x \in (Z_2)^n$ where x values are in lexicographic order.*

Definition 5. *For any $f \in B_n$, define $(-1)^f$ to be the function $F : (\mathbb{Z}_2)^n \rightarrow \{-1, 1\}$ such that $F(x) = (-1)^{f(x)}$ for all $x \in (\mathbb{Z}_2)^n$.*

For cryptographic systems, the method of confusion and diffusion is used as a fundamental technique to achieve security [10]. Confusion is satisfied by including a highly nonlinear function into the cryptosystem. These functions simultaneously have maximum distance to affine functions and maximum distance to linear structures, as well. So they are called as strong functions, i.e. not weak. A function is considered weak whenever it can be turned into a cryptographically weak function by means of simple (linear or affine) transformations as a minimum correlation to affine functions [9, p. 549].

The nonlinearity of a function can be calculated by using the Walsh transform, one of the important tools in cryptography. The definition of Walsh transform and its properties are given below. After that, a method for computing the nonlinearity will be demonstrated.

The inner product of two vectors $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n) \in (\mathbb{Z}_2)^n$ is $x \cdot y = \sum_{i=1}^{n} x_i y_i \mod 2$.

Definition 6. *Let F be any real-valued function defined on $(\mathbb{Z}_2)^n$. The Walsh transform of F is the function $\hat{F} : (Z_2)^n \rightarrow \mathbb{R}$ defined by the following formula. For all $x \in (Z_2)^n$, $\hat{F}(x) = \sum_{y \in (\mathbb{Z}_2)^n} (-1)^{x \cdot y} F(y)$.*

Nonlinearity of a Boolean function is the minimum distance of a Boolean function f to the set of all linear functions

$$nl(f) = min\{d(f, A_n)\},$$

where A_n is the set of all affine functions in B_n. Below we consider $F(x) = (-1)^{f(x)}$.

$$\begin{aligned}\hat{F}(x) &= \sum_{y\in(\mathbb{Z}_2)^n} (-1)^{x\cdot y}(-1)^{f(x)} \\ &= \sum_{f(x)=x\cdot y} 1 - \sum_{f(x)\neq x\cdot y} 1, \\ &= 2^n - 2d(f, x\cdot y).\end{aligned}$$

Then, $d(f, x\cdot y) = 2^{n-1} - \frac{1}{2}\hat{F}(x)$ is the distance between $f(x)$ and $l_y(x) = x\cdot y$.

Theorem 4. *The nonlinearity of a Boolean function f on $\mathbb{Z}_2^n$ can be expressed by $nl(f) = 2^{n-1} - \frac{1}{2}max\{|\hat{F}(x)| : x \in \mathbb{Z}_2^n\}$.*

Theorem 5. *For any function f on $\mathbb{Z}_2^n$, the nonlinearity of f satisfies $nl(f) \leq 2^{n-1} - 2^{\frac{n}{2}-1}$.*

A function f on $\mathbb{Z}_2^n$ attains the upper bound of nonlinearity $2^{n-1} - 2^{\frac{n}{2}-1}$ is called a *bent function*. It is clear that a function $f \in B_n$ is a bent function if $\hat{F}(x) = \pm 2^{n/2}$ for all $x \in \mathbb{Z}_2^n$. Maximal nonlinearity is hence attained by bent functions, with only even n. For instance, let $P(x)$ be a function from $\mathbb{Z}_2$ to $\mathbb{Z}_2$. $P(x)$ is bent if all Walsh coefficients of $(-1)^{P(x)}$ are ±1. This definition of a bent function over $\mathbb{Z}_2$ can be directly extended to functions on $\mathbb{Z}_q$. First the Walsh transform is extended to the functions on $\mathbb{Z}_q$.

Definition 7 [6, p. 339]. *Suppose $F : (\mathbb{Z}_q)^n \to \mathbb{C}$ and let $\omega = e^{2i\pi/q}$. The Walsh transform of F is the function $\hat{F} : (\mathbb{Z}_q)^n \to \mathbb{C}$ defined for all $\boldsymbol{x} \in (\mathbb{Z}_q)^n$ by the formula:*

$$\hat{F}(x) = \sum_{\boldsymbol{y}\in(\mathbb{Z}_q)^n} \omega^{\boldsymbol{x}\cdot\boldsymbol{y}} F(\boldsymbol{y}).$$

Then a generalized bent function is defined similarly.

Definition 8 [6]. *Suppose $f : (\mathbb{Z}_q)^n \to \mathbb{Z}_q$ and define $F : (\mathbb{Z}_q) \to \mathbb{C}$ by the rule $F(\boldsymbol{x}) = \omega^{f(\boldsymbol{x})}$ for all $\boldsymbol{x} \in (\mathbb{Z}_q)^n$, where $\omega = e^{2i\pi/q}$. Then f is a generalized bent function if $|\hat{F}(\boldsymbol{x})| = q^{n/2}$ for all $\boldsymbol{x} \in (\mathbb{Z}_q)^n$.*

The connection between Hadamard matrices and generalized bent functions is given in Theorem 6.

Theorem 6 [6]. *Suppose f and F are defined as above. Define the matrix $H_f = (h_{\boldsymbol{x},\boldsymbol{y}})$, where $h_{\boldsymbol{x},\boldsymbol{y}} = F(\boldsymbol{x} - \boldsymbol{y})$ for all $\boldsymbol{x}, \boldsymbol{y} \in (\mathbb{Z}_q)^n$. Then f is a generalized bent function if and only if H_f is a Butson-Hadamard matrix.*

We give a well known result on the existence of a generalized bent function.

Theorem 7 [6, p. 96]. *Suppose that n is even or $q \equiv 2 \mod 4$. Then there exists a generalized bent function $f : (\mathbb{Z}_q)^n \to \mathbb{Z}_q$.*

Therefore, we see that there is a one to one correspondence between generalized bent functions and Butson-Hadamard matrices. We give an example below.

Example 8. $f : \mathbb{Z}_3^2 \to \mathbb{Z}_3$ and $f(x_1, x_2) = x_1 x_2$. The matrix H corresponding to the bent function f is given below. The entries of the Hadamard matrix forms a power of 3-th of unity ω.

$$H = \begin{bmatrix} w^{f(0,0)} & w^{f(0,2)} & w^{f(0,1)} & w^{f(2,0)} & w^{f(2,2)} & w^{f(2,1)} & w^{f(1,0)} & w^{f(1,2)} & w^{f(1,1)} \\ w^{f(0,1)} & w^{f(0,0)} & w^{f(0,2)} & w^{f(2,1)} & w^{f(2,0)} & w^{f(2,2)} & w^{f(1,1)} & w^{f(1,0)} & w^{f(1,2)} \\ w^{f(0,2)} & w^{f(0,1)} & w^{f(0,0)} & w^{f(2,2)} & w^{f(2,1)} & w^{f(2,0)} & w^{f(1,2)} & w^{f(1,1)} & w^{f(1,0)} \\ w^{f(1,0)} & w^{f(1,2)} & w^{f(1,1)} & w^{f(0,0)} & w^{f(0,2)} & w^{f(0,1)} & w^{f(2,0)} & w^{f(2,2)} & w^{f(2,1)} \\ w^{f(1,1)} & w^{f(1,0)} & w^{f(1,2)} & w^{f(0,1)} & w^{f(0,0)} & w^{f(0,2)} & w^{f(2,1)} & w^{f(2,0)} & w^{f(2,2)} \\ w^{f(1,2)} & w^{f(1,1)} & w^{f(1,0)} & w^{f(0,2)} & w^{f(0,1)} & w^{f(0,0)} & w^{f(2,2)} & w^{f(2,1)} & w^{f(2,0)} \\ w^{f(2,0)} & w^{f(2,2)} & w^{f(2,1)} & w^{f(1,0)} & w^{f(1,2)} & w^{f(1,1)} & w^{f(0,0)} & w^{f(0,2)} & w^{f(0,1)} \\ w^{f(2,1)} & w^{f(2,0)} & w^{f(2,2)} & w^{f(1,0)} & w^{f(1,1)} & w^{f(1,2)} & w^{f(0,1)} & w^{f(0,0)} & w^{f(0,2)} \\ w^{f(2,2)} & w^{f(2,1)} & w^{f(2,0)} & w^{f(1,2)} & w^{f(1,1)} & w^{f(1,0)} & w^{f(0,2)} & w^{f(0,1)} & w^{f(0,0)} \end{bmatrix}$$

$$= \begin{bmatrix} w^0 & w^2 & w^1 & w^6 & w^8 & w^7 & w^3 & w^5 & w^4 \\ w^1 & w^{f0} & w^2 & w^7 & w^6 & w^8 & w^4 & w^3 & w^5 \\ w^2 & w^1 & w^0 & w^8 & w^7 & w^6 & w^5 & w^4 & w^3 \\ w^3 & w^5 & w^4 & w^0 & w^2 & w^1 & w^6 & w^8 & w^7 \\ w^4 & w^3 & w^5 & w^1 & w^0 & w^2 & w^7 & w^6 & w^8 \\ w^5 & w^4 & w^3 & w^2 & w^1 & w^0 & w^8 & w^7 & w^6 \\ w^6 & w^8 & w^7 & w^3 & w^5 & w^4 & w^0 & w^2 & w^1 \\ w^7 & w^6 & w^8 & w^4 & w^3 & w^5 & w^1 & w^0 & w^2 \\ w^8 & w^7 & w^6 & w^5 & w^4 & w^3 & w^2 & w^1 & w^0 \end{bmatrix}$$

On the other hand, we can show an example for the other direction of Theorem 6. The matrix H is a Butson-Hadamard matrix.

$$H = \begin{bmatrix} \omega^0 & \omega^2 & \omega^0 & \omega^0 \\ \omega^0 & \omega^0 & \omega^2 & \omega^0 \\ \omega^0 & \omega^0 & \omega^0 & \omega^2 \\ \omega^2 & \omega^0 & \omega^0 & \omega^0 \end{bmatrix}$$

Then, $f : \mathbb{Z}_4 \to \mathbb{Z}_4$, as follows $f(0) = 0$, $f(1) = 0$, $f(2) = 0$, $f(3) = 2$.

We now investigate the functions corresponding to γ near Butson Hadamard matrices. We start with a circulant γ near Butson Hadamard matrix H and convert the first row of H into a truth table of a function f as in Theorem 6 and Example 8. Then the Walsh transform of f is calculated by Definition 7. We apply this conversion for the examples obtained by exhaustive search on m and v by using MAGMA [1]. We tabulate our results in Table 2.

Looking at Table 2, it is seen that the smaller gamma values, the more flat Walsh spectrum and so the higher nonlinearity. Therefore one can obtain new families of nonlinear functions by searching matrices $\mathrm{BH}_\gamma(v, m)$ for non integer $\gamma \in \mathbb{Z}[\zeta_m]$ having small absolute value.

Table 2. Samples of Walsh spectrum of some γ near Butson Hadamard matrices

m	v	γ	$\lvert\gamma\rvert$	*truthtable*	$\lvert\hat{F}\rvert$
5	5	$\zeta_5^3 + \zeta_5^2 + 3$	1.38	$(0, 2, 0, 0, 0)$	$(3.24, 1.90, 1.90, 1.90, 1.90)$
6	6	-1	1	$(6, 2, 0, 2, 6, 1)$	$(3.60, 1, 1, 4.35, 1, 1)$
7	7	$2\zeta_7^4 + 2\zeta_7^3 + 3$	0.60	$(2, 3, 3, 2, 3, 2, 2)$	$(6.32, 1.22, \ldots, 1.22)$
8	8	0	0	$(5, 7, 1, 5, 1, 7, 5, 5)$	$(2.82, \ldots, 2.82)$
9	9	$\zeta_9^5 + \zeta_9^4 + 7$	5.12	$(6, 2, 6, 6, 6, 6, 6, 6, 6)$	$(7.06, 1.97, \ldots, 1.97)$
10	10	$\zeta_{10}^3 - \zeta_{10}^2 + 7$	6.38	$(0, 6, 7, 3, 5, 2, 5, 3, 7, 6, 0)$	$(3.55, 3.23, 1.32, 2.55, 4.30$ $3.59, 4.29, 2.55, 1.32, 3.23)$
11	11	$3\zeta_{11}^6 + 3\zeta_{11}^5 + 5,$	0.75	$(0, 6, 6, 6, 0, 6, 0, 0, 6, 0, 0)$	$(1.85, 3.42, \ldots, 3.42)$
11	11	$\zeta_{11}^6 + \zeta_{11}^5 + 9$	7.08	$(0, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0)$	$(9.044, 1.979, \ldots, 1.979)$
11	11	0	0	$(0, 6, 7, 3, 5, 2, 5, 3, 7, 6, 0)$	$(3.31, \ldots, 3.31)$

6 Conclusion

In this paper, we studied the γ near Butson-Hadamard matrices and their cryptographic applications. We studied the existence cases of γ near Butson-Hadamard matrices for $\gamma \in (\mathbb{Z}[\zeta_m] \cap \mathbb{R}) \backslash \mathbb{Z}$ by using the tools from algebraic number theory.

Firstly, we converted the existence condition of a γ near Butson-Hadamard matrix to an equation over a ring of integers of an cyclotomic number field. Then we obtained a novel result stating necessary conditions for the nonexistence of this equation. Then the direct applications of these results to γ near Butson-Hadamard matrices were shown. We presented examples of nonexistence cases in details and obtained existence examples by computer search.

Next, the connection of γ near Butson-Hadamard matrices to cryptographic functions was drawn. Cryptographers look for nonlinear Boolean (multivariate) functions on residue rings. These functions are used in block ciphers to provide confidentiality of the message between two parties. In this study, it was shown that a γ near Butson-Hadamard matrix can be converted to a Boolean function whose nonlinearity is proportional with the value $|\gamma|$. And, the examples of nonlinear functions obtained from γ near Butson-Hadamard matrices were presented.

Acknowledgement. The authors are supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under Project No: 116R026.

References

1. Bosma, W., Cannon, J., Playoust, C.: The Magma algebra system I: the user language. J. Symbolic Comput. **24**(3–4), 235–265 (1997). https://doi.org/10.1006/jsco.1996.0125. Computational Algebra and Number Theory (London, 1993)
2. Brock, B.W.: Hermitian congruence and the existence and completion of generalized Hadamard matrices. J. Comb. Theory Ser. A **49**(2), 233–261 (1988)

3. Butson, A.: Generalized Hadamard matrices. Proc. Am. Math. Soc. **13**(6), 894–898 (1962)
4. Colbourn, C.J., Dinitz, J.H.: Handbook of Combinatorial Designs. CRC Press, Boca Raton (2006)
5. Horadam, K.J.: Hadamard Matrices and Their Applications. Princeton University Press, Princeton (2007)
6. Kumar, P.V., Scholtz, R.A., Welch, L.R.: Generalized bent functions and their properties. J. Comb. Theory Ser. A **40**(1), 90–107 (1985)
7. Kurt, S., Yayla, O.: Nearly perfect sequences and cryptographic functions. In: Workshop on Practical and Theoretical Aspects of Cryptography and Information Security, Tbilisi (2017)
8. de Launey, W.: On the nonexistence of generalised weighing matrices. Ars Comb. **17**(A), 117–132 (1984)
9. Meier, W., Staffelbach, O.: Nonlinearity criteria for cryptographic functions. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 549–562. Springer, Heidelberg (1990). https://doi.org/10.1007/3-540-46885-4_53
10. Shannon, C.E.: A mathematical theory of cryptography. Memorandum MM 45–110–02 (1945)
11. Stewart, I., Tall, D.: Algebraic Number Theory and Fermat's Last Theorem. CRC Press, Boca Raton (2015)
12. Washington, L.C.: Introduction to Cyclotomic Fields. Springer Science & Business Media, Heidelberg (1997). https://doi.org/10.1007/978-1-4612-1934-7
13. Winterhof, A., Yayla, O., Ziegler, V.: Non-existence of some nearly perfect sequences, near Butson-Hadamard matrices, and near conference matrices. arXiv preprint arXiv:1407.6548 (2014)

Multi-secret Sharing Scheme for Level-Ordered Access Structures

Appala Naidu Tentu[1](✉), Abdul Basit[2], K. Bhavani[2], and V. Ch. Venkaiah[2]

[1] CR Rao Advanced Institute of Mathematics, Statistics, and Computer Science,
University of Hyderabad Campus, Hyderabad 500046, India
naidunit@gmail.com

[2] School of Computer and Information Science, University of Hyderabad,
Hyderabad 500046, India

Abstract. The secret sharing scheme by Dileep et al. [19] uses Level ordered access structure which is missing in the existing access structures. In their scheme, sequential reconstruction of the secret is achieved by adding a virtual player at all the levels except at the first level. In this paper, we propose a variation of sequential secret sharing scheme for level ordered access structure (LOAS) [19], where multisecrets are distributed to multilevels each corresponding to a level by using the concepts of quadratic residues and discrete logarithm problem. The method consists of sharing of m secrets in m levels, each corresponding to a level. The distribution of secrets is based on quadratic residues concept and that of the discrete logarithm problem. The reconstruction of secrets is such that players of different levels find their respective level secrets individually only after they get their immediate higher level permission. Verification phase is also added at all the levels which guarantees the correctness of the shares in the presence of any cheater. The comparison of the proposed secret sharing scheme with existing secret sharing schemes, time complexity of the scheme and security analysis of the scheme for passive adversary model are discussed.

Keywords: Hierarchical · Compartmented · Sequential
Quadratic residue · Discrete logarithm problem

1 Introduction

Secret sharing is a method for sharing a secret among a group of participants, i.e., each participant receives a share or part of the secret. Reconstruction of secret is possible only when authorized participants pool their shares together; with individual shares, secret cannot be recovered. Ideality and perfectness are the necessary characteristics for a secret sharing scheme concerning of efficiency and security respectively.

A secret sharing scheme is said to be perfect if any subset of players that is not in the defined access structure (i.e. unauthorized subset) cannot determine

J. Kaczorowski et al. (Eds.): NuTMiC 2017, LNCS 10737, pp. 267–278, 2018.
https://doi.org/10.1007/978-3-319-76620-1_16

any information regarding the secret. A secret sharing scheme is known to be ideal if each participant receives exactly one share and the domain of both the secrets and the shares are the same.

Shamir [1] and Blackely [2] are the first to propose secret sharing in which Shamir has used the standard Lagrange polynomial interpolation, and Blakley has used linear projective geometry for constructions.

Definition 1 (Access Structure). *Let $U = \{U_1, \ldots, U_n\}$ be a set of n players. A collection $\Gamma \subseteq 2^U$ of non-empty subsets of U is monotone if for any $A, B \subseteq U$, $A \in \Gamma$ and $A \subseteq B$ imply that $B \in \Gamma$. An access structure Γ over U is a collection $\Gamma \subseteq 2^U$ such that it is monotone. Sets in Γ are called authorized, and sets not in Γ are called unauthorized.*

Many access structures are suggested in the literature. Some of these are the generalised access structure, the (t, n) access structure and the multipartite access structure. Threshold (t, n) access structure is an important class of access structure which consists of n shareholders where, an authorized set consists of any t or more than t players. Less than t players belong to unauthorized set and they cannot reconstruct the secret. Multipartite access structures may further be classified into Hierarchical and Compartmented access structures. Hierarchical access structure is again classified into Disjunctive and Conjunctive access structures.

Simmons [3] has first proposed the disjunctive multilevel secret sharing scheme.

Definition 2. *Disjunctive hierarchical access structure [7] is a multipartite access structure in which each level U_i is associated with a threshold t_i, $1 \leq i \leq m$, and the secret can be reconstructed when, for some i, there are at least t_i shareholders who all belong to levels smaller than or equal to L_i. Mathematically,*

$$\Gamma = \{V \subseteq U : |V \cap (\cup_{j=1}^{i} U_j)| \geq t_i, \ \textit{for some } i \in \{1, 2, \cdots, m\}\}$$

where U denotes the set of participants.

Tassa [4] has proposed a conjunctive hierarchical secret sharing scheme which is a variation of disjunctive hierarchical secret sharing scheme.

Definition 3. *Conjunctive hierarchical access structure [7] is a multipartite access structure in which each level U_i is associated with a threshold t_i for $1 \leq i \leq m$, and the secret can be reconstructed when, for every i, there are at least t_i shareholders who all are associated to levels smaller than or equal to U_i. Mathematically,*

$$\Gamma = \{V \subseteq U : |V \cap (\cup_{j=1}^{i} U_j)| \geq t_i, \ \textit{for every } i \in \{1, 2, \cdots, m\}\}$$

where U denotes the set of participants.

Compartmented access structure Γ is defined as follows.

Definition 4. *Compartmented access structure [7] is a multipartite access structure in which each compartment is associated with a threshold t_i, $1 \leq i \leq m$, and the reconstruction of the secret is possible only when, for every i, there are at least t_i shareholders from U_i and a total of at least t_0 participants from all the compartments. Mathematically,*

$$\Gamma = \{V \subseteq U : |V \cap U_i| \geq t_i, \textit{ for every } i \in \{1, 2, \cdots, m\} \textit{ and } |V| \geq t_0\}$$

where U denotes the set of participants.

The proposed scheme can be applied to scenarios such as an organisation consisting of different departments in which each department is having a secret and for reconstruction of secret of a particular department, the immediate above department has to give permission.

A simple application can be in banking scenarios where, the managers, assistant managers, cashiers, and clerks are in different authority levels. For a clerk to perform any action, he is not supposed to seek the permission of managers or assistant managers, but he has to take the permission of his direct higher officer cashier. To cater this type of requirement Level Ordered Access structure was proposed by Dileep et al. [19]. They also gave a secret sharing scheme that realizes this access structure. Here we propose yet another scheme for this access structure. Our scheme makes use of the concept residues and the well known discrete logarithm problem.

1.1 Related Work

An ideal secret sharing scheme based on Birkhoff interpolation was proposed by Tassa. Constructions of the scheme are given in [4]. Later, Tassa and Dyn have proposed ideal secret sharing scheme for multipartite access structures like hierarchical and compartmented. These constructions, which are described in [5] are based on bivariate interpolation. These schemes have a drawback of either requiring a large finite field or having some restrictions on the identities assigned to the participants or the scheme being perfect in a probabilistic manner. A scheme is said to be perfect in a probabilistic manner if either a set of unauthorised participants can recover the secret or set of authorised participants might not be able to recover the secret with some probability.

Computationally perfect conjuctive and disjunctive secret sharing schemes based on error correcting codes were proposed by Tentu et al. in [7,8]. Simmons [3] has introduced the concept of compartmented secret sharing. He has considered secret sharing in compartmented and multilevel access structures. Ghodosi has proposed a scheme [6] for a compartmented access structures by applying Shamir secret sharing scheme two times, first to get partial secrets and second to combine them into the required secret. A clear description of two cases of global threshold being equal to the summation of all local thresholds and global threshold being greater than the summation of all local thresholds are provided in the paper.

A multistage secret sharing scheme that uses one-way function was proposed by [12] He and Dawson in 1994 to share multiple secrets. Public shift techniques were used to obtain the true shadows and in order to recover the secrets stage-by-stage in a predefined manner, successive applications of one-way function is used. In this scheme, the dealer publishes pn public values. An alternate scheme was proposed by Harn [13] reducing the number of public values to $p(n-t)$. Both the schemes proposed by He and Dawson and Harns are claimed to be multi-use. These are later proved to be of not multiuse by Chang et al. in [14] by showing that the dealer fails to recover the secrets in the predefined order. In order to remove the drawbacks Chang has refined the He and Dawson scheme which using one-way function [14]. Bidyapati has proposed [15] where multiple secrets are shared among the participants. A practically verifiable multistage secret sharing scheme based on the YCH scheme that uses the intractability of the discrete logarithm has been proposed in [16]. we use some of these concepts in our proposed scheme. A new hierarchical sequential secret sharing scheme is proposed by Mehrdad Nojoumian [18] which is different from the existing hierarchical secret sharing schemes.

1.2 Motivation and Contribution

All the existing schemes that realize hierarchical access structures involve sharing of a single secret among the participants who are in different levels of authority. Also there is no concept of ordering among the levels. There can be certain applications which may require ordering among the levels.

Dileep et al. has proposed a level ordered access structure in [19] which enforces ordering required in some of the applications as against the existing access structures in the literature. They also gave a secret sharing scheme realizing the level ordered access structure in [18,19]. Here, we are proposing yet another scheme that realizes this level ordered access structure. Our scheme is based on the concepts of quadratic residue and discrete logarithm problem. Our scheme is applicable for the case of multi-secrets that are to be shared among the multi-levels; each corresponding to a level. Reconstruction of a particular level secret requires its immediate higher level permission (enforcing ordering concept in the recovery of secrets). In our scheme a verification phase has also been added at each and every level to check whether the shares submitted by the shareholders during reconstruction are true or not.

Organization of the paper: The rest of the paper is organized as follows: Sect. 2 presents some of the preliminaries used in the construction. In Sect. 3, we presented our proposed Level ordered secret sharing scheme. Comparison and security analysis of the scheme is presented in Sects. 4 and 5 respectively. Conclusions of the proposed scheme are discussed in Sect. 6.

2 Preliminaries

Discrete Logarithm Problem: Discrete logarithm problem is used in variety of applications in the field of cryptography. It is significant because of its computational difficulty. There are many cryptosystems whose security is based on the computational difficulty of discrete logarithm problem.

Definition 5. *Given a and g there exists G, where G is a multiplicative group G, find an integer x, if it exists, such that $g^x = a$. The number x is the discrete logarithm of a to the base g, which can also be written as $x = \log_g(a)$.*

Quadratic Residues: An integer $a \in Z$ is said to be a quadratic residue modulo n, if there exists a $b \in Z$ such that the congruence $b^2 \equiv a \mod n$ is satisfied, i.e., if a is a perfect square modulo n. If there is no such b satisfying the above congruence, then it is called a non quadratic residue.

Computing a square root of an integer is generally a difficult problem. But if the integer is from a finite field, it can be computed as follows.

Definition 6. *Square root of a quadratic residue in a finite field $\mathbb{F}_P$ can be calculated in polynomial time.*

Given a and P, where a is a quadratic residue modulo P and P is a prime number such that $P \equiv 3 \mod 4$, then $\sqrt{a} = a^{\frac{P+1}{4}}$.

$\sqrt{a}$ can be found by formula $a^{\frac{P+1}{4}}$.

Secret Sharing Scheme was first introduced by Shamir [1]. The scheme relies on Lagrange interpolation. In this scheme, the dealer divides the secret S into n shares so that at least any t number of participants (authorized set of participants) can recover the secret S and less than t participants (unauthorized set of participants) can never recover the secret S.

2.1 Level Ordered Access Structure

A new access structure that is different from the existing multipartite and multistage access structures is proposed by Dileep et al. in [19]. LOAS imposes an ordering concept which is missing in the existing secret sharing schemes.

Definition 7. *Let U be a set of n participants and let $U_1, U_2, \cdots U_m$ be a partition of the set U. Also let b_i be a boolean variable, which we call the activation index associated with the i^{th} level U_i, $1 \leq i \leq m$. Define S_i, recursively, to be an authorized set corresponding to the i^{th} level.*

1. *$S_i \subseteq U_i$ and $|S_i| \geq t_i$,*
2. *$\exists$ an authorized set (S_{i-1}) whose activation index (b_{i-1}) is True (T), where $b_0 = T$ and $S_0 = \emptyset$*

Dileep et al. [19] proposed a secret sharing scheme that realizes this access structure in that the participants are divided into different levels of authority where each level is assigned with a partial secret s_i $(1 \leqslant i \leqslant m)$. The master secret is the partial secret at the last level i.e. s_m. Except at the first level, each and every level contains two parts, the first part containing the set of players at level i, and the second part containing the virtual player at level i. At all the levels except at first level, Shamir's (t, n) secret sharing scheme is used for the distribution of partial secrets among the set of participants in the first part and the partial secret of the immediate below level is assigned to the virtual player who is at the second part. For the participants at the first level, Shamir's (t, n) secret sharing scheme is used for the distribution of partial secret s_1. At each and every level, the partial secrets are recovered by the authorised participants only after the participants at the lower level have first reconstructed their partial secret. For the reconstruction of master secret, the partial secrets are to be recovered based on the specified order which is enforced by adding a virtual player at each and every level except at the last level whose share will be the partial secret at its immediate below level.

We are proposing yet another scheme to realize the access structure based on the quadratic residue concept and the discrete logarithm problem.

3 Proposed Scheme

3.1 Overview

The dealer chooses m secrets $s_1, \cdots, s_m$ one each for the m levels, over $\mathbb{F}_P$ where $P \equiv 3 \mod 4$. The dealer then computes partial secrets from the above original secrets using the quadratic residue concept and the discrete logarithm problem and shares one each in the respective levels using Shamir's (t, n) secret sharing scheme. For reconstructing the secret of a particular level i, the set of authorized participants at its immediate higher level i.e. $i-1$ level are required to find s'_{i-1} and pass it to level i as a permission, which can be used to recover the secret at level i. Without the cooperation of its immediate higher level, no level can reconstruct its secret.

In the following, the details about the scheme are given, which divides n players into m levels, where U_i denotes the players at level i for $i \in (1, m)$, and n_i denotes the number of participants in that particular level. Threshold t_i, $1 \leq i \leq m$ are assigned to respective levels, and m secrets are chosen from $\mathbb{F}_P$. where P is a prime such that $P \equiv 3 \mod 4$.

3.2 Initialization

Let $U = \{P_{ij} | 1 \leqslant i \leqslant m, 1 \leqslant j \leqslant n_i\}$ (i is the level number and j is the participant number in that level)

n_i is the number of players at the i^{th} level and m is the number of levels.
$U = \{U_1, U_2, \cdots, U_m\}$ be the m levels.
$t = \{t_1, t_2, \cdots, t_m\}$ be the set of thresholds corresponding to the m levels.
Let n_i represent the set of players at level U_i, $1 \leq i \leq m$ and $n = \sum_{i=1}^{m} |n_i|$
Initialization phase is done in 3 steps.

Dealer:

- Selects m secrets $s_1, \cdots, s_m$ from $\mathbb{F}_p$ where P is prime such that $P \equiv 3 \mod 4$.
- Chooses two primes, p and q, $N = p * q$. Both p and q should be so safe that no one can factor N efficiently.
- Randomly chooses an integer g from the interval $\left[N^{1/2}, N\right]$ such that g is relatively prime to p and q. Publish $\{g, N\}$.

Players:

- Each participant P_{ij} in U_i randomly chooses an integer s_{ij} from the interval $[2, N]$ as her/his own secret shadow (share) where $i = 1, \cdots, m$, and $j = 1, \cdots, n_i$ and computes $R_{ij} = g^{s_{ij}} \mod N$.
- Then P_{ij} provides R_{ij} and her/his identity number ID_{ij}, to the dealer D.

Dealer:

- Dealer must ensure that R_{ij}'s are unique in each and every level else, demand the participants to choose different secret shadows (share) until R_{ij}'s are different in the respective levels, for $i = 1, \cdots, m$, and $j = 1, 2, \cdots, n_i$. Publish $\{ID_{ij}, R_{ij}\}$.

3.3 Distribution

Dealer performs the following steps for all the levels.

- Dealer D randomly chooses an integer s_0 from the interval $[2, N]$ such that s_0 is relatively prime to $(p-1)$ and $(q-1)$. Then D computes f so that $s_0 * f = 1 \mod \Phi(N)$, where $\Phi(N)$ is the Euler phi-function.
- Compute $R_0 = g^{^s0} \mod N$ and $I_{ij} = R_{ij}^{^s0} \mod N$ where $j = 1, \cdots, n_i$ and $i = 1, \cdots, m$.
- Publish $\{R_0, f\}$.
- Construct $(t_i - 1)$ degree polynomial
  $h_i(x) = S_i + a_{i1}x^1 + a_{i2}x^2 + \cdots + a_{it_{i-1}}x^{t_i-1} \mod P$
  where $0 < S_i, a_{i1}, a_{i2}, \cdots, a_{it_i-1} < P$.

(1) For $i = 1$, $S_i = s_1$
   - Compute $y_{1j} = h_1(I_{1j}) \mod P$, where $j = 1, \cdots, n_1$ and distribute the share (y_{1j}) via secure channel, for $j = 1, \cdots, n_1$ to the i^{th} level participants.
   - Find $s'_1 = g^{S_i} \mod P$.

(2) For $i = 2, m$.

1. Find $s_i * s'_{i-1}$,

- If $s_i * s'_{i-1}$ is a quadratic residue in $\mathbb{F}_P$, then find partial secret $S_i = \sqrt{s_i * s'_{i-1}}$
  (a) Compute $y_{ij} = h_i(I_{ij}) \mod P$ where $j = 1, \cdots, n_i$ and Securely share (y_{ij}) for $j = 1, \cdots, n_i$ to the i^{th} level participants.
- If $s_i * s'_{i-1}$ is not a quadratic residue,
  Choose a random number R from $\mathbb{F}_P$ such that $s_i * s'_{i-1} * R$ is a quadratic residue modulo P.
  (a) With $S_i = R$, Compute $r_{ij} = h_i(I_{ij}) \mod P$ where $j = 1, \cdots, n_i$ and Securely share (r_{ij}) for $j = 1, \cdots, n_i$. to the participants.
  (b) With $S_i = \sqrt{s_i * s'_{i-1} * R}$, Compute $y_{ij} = h_i(I_{ij}) \mod P$ where $j = 1, \cdots, n_i$ and securely share (y_{ij}) for $j = 1, \cdots, n_i$. to the participants.

2. For $i = 2, m-1$, find $s'_i = g^{S_i} \mod P$.

3.4 Verification

- P_{ij} computes $I_{ij}^{'} = R_0^{s_{ij}} \mod N$ to gain the share, where s_{ij} is the shadow of P_{ij} where $j = 1, 2, \cdots, n_i$.
- Anyone can verify $I_{ij}^{'}$ provided by P_{ij}.

  If $I_{ij}^{'f} = R_{ij} \mod N$ where $j = 1, 2, \cdots, n_i$ then $I_{ij}^{'}$ is true; else $I_{ij}^{'}$ is false and P_{ij} may be a cheater.

3.5 Reconstruction

Players perform the following steps.

- For $i = 1$

  Using Lagrange Polynomial Interpolation, $S_i = s_1$ can be recovered as follows

  The polynomial $h_1(x) \mod P$ can be found uniquely as follows:

  $h_1(x) = \sum_{j=1}^{t_1} y_{ij} \prod_{k=1,k\neq j}^{t_1} \left(\frac{x - I'_{ik}}{I'_{ij} - I'_{ik}}\right)$

  $h_1(x) = S_i + a_{i1}x^1 + a_{i2}x^2 + \cdots + a_{it_{i-1}}x^{t_i - 1} \mod P$.
- Any other level, other than level-1 has to perform the following operations to recover s_i.
  1. Using Lagrange Polynomial Interpolation, S_i can be recovered.

     The polynomial $h_i(x) \mod P$ can be found uniquely as follows:

     $h_i(x) = \sum_{j=1}^{t_i} y_{ij} \prod_{k=1,k\neq j}^{t_i} \left(\frac{x - I'_{ik}}{I'_{ij} - I'_{ik}}\right)$

     $h_i(x) = S_i + a_{i1}x^1 + a_{i2}x^2 + \cdots + a_{it_{i-1}}x^{t_i - 1} \mod P$.
  2. Using Lagrange Polynomial Interpolation, R can be recovered.

     The polynomial $h_i(x) \mod P$ can be found uniquely as follows:

     $h_i(x) = \sum_{j=1}^{t_i} r_{ij} \prod_{k=1,k\neq j}^{t_i} \left(\frac{x - I'_{ik}}{I'_{ij} - I'_{ik}}\right)$

     $h_i(x) = R + a_{i1}x^1 + a_{i2}x^2 + \cdots + a_{it_{i-1}}x^{t_i - 1} \mod P$.
  3. $i - 1$ level finds $s'_{i-1} = g^{S_{i-1}} \mod P$ and passes securely to level i. where S_{i-1} is found by Lagrange Polynomial Interpolation with (t_{i-1}, n_{i-1}).
  4. Find $s_i = \frac{S_i^2}{s'_{i-1} * R} \mod P$.

4 Comparison with Existing Hierarchical Schemes

In the existing hierarchical access structures only a single secret is shared among the set of participants who are divided into different levels of authority. Moreover there is no concept of ordering in the levels during the reconstruction of the secret. In our model, multiple different secrets are shared among the participants in different authority levels, each one in the corresponding level. The access structure for each and every level is in such a way that, if they want to recover their level secret, then they need to take their immediate higher authority level permission. Ordering concept is enforced in the levels during the reconstruction of secrets.

5 Security Analysis

The following security analysis we are discussing is based on the assumptions that the dealer is honest (dealer doesn't involve in any malicious activity) and the channels for communication between any two connecting nodes are secure, so there is no possibility for the information to leak to non-authenticating node. Hence, we discuss the security analysis only if the participants in the scheme can cheat.

Notations used in this scheme are listed below
i : The number of level.
S_i : Partial secret of level i which is private.

$$\left\{ \begin{array}{cc} S_i = s_1 & i = 1 \\ S_i = \sqrt{s_i * s'_{i-1} * R}, & i = \{2, m\} \end{array} \right\}$$

P : Prime number which is public.
g : Generator which is also public.
s_i : Compartment original secret

$$s_i = \frac{S_i^2}{s'_{i-1} * R} \mod P$$

s'_i : $g^{S_i} \mod P$
R : Random number.

Some of the possible attacks are the following:

- Unauthorized set of participants try to recover the secret s_i at level i.
- Participants at the immediate lower level try to obtain the secret s_i of level i.
- Participants at the immediate higher level try to obtain the secret s_i of level i.
- Participants at the higher and lower levels together try to recover the secret s_i of level i.

Lemma 1: Unauthorized set of participants at level i cannot recover the secret s_i.

Proof:

$$s_i = \frac{S_i^2}{s'_{i-1} * R} \mod P$$

Partial secret S_i and a random number R are distributed using Shamir's (t_i, n_i) threshold scheme. Less than t_i participants cannot recover R and the partial secret S_i which are essential to recover s_i. Hence, unauthorized set of participants cannot recover s_i.

Lemma 2: The partial secret S_i of a level i cannot be obtained by the lower level participants.

Proof: At level i,

- The public values are g, P.
- The private value is the partial secret S_i.

$$s'_i = g^{S_i} \mod P$$

is calculated and passed to level $i+1$. Thus s'_i is known to level i and $i+1$ only. Hence the participants at level $i+1$ have the knowledge of s'_i, g, P. But recovering S_i from s'_i, g, and P by the $i+1$ level participants depends on the hardness of discrete logarithm problem which is stated in Sect. 2.

Lemma 3: The secret s_i of a level i cannot be obtained by the higher level participants.

Proof: $i-1$ level participants knows s'_{i-1}

$$s_i = \frac{S_i^2}{s'_{i-1} * R} \mod P$$

S_i and R are essential to recover s_i which can be obtained only by the participants at i^{th} level using Lagrange polynomial interpolation.

Lemma 4: Higher and lower level participants of level i cannot recover the secret s_i.

Proof: At level i,

$$S_i^2 = s_i * s'_{i-1} * R \mod P$$

$$s_i = \frac{S_i^2}{s'_{i-1} * R} \mod P$$

$i-1$ level participants knows s'_{i-1}.

$i+1$ level participants knows $s'_i = g^{S_i} \mod P$, but they cannot get S_i from s'_i, g and P due to discrete logarithm problem which is stated in Sect. 2, Definition 5.

Since S_i and R are essential to recover S_i, the players at higher and lower levels together cannot recover s_i.

5.1 Ideality and Perfectness

Lemma 5: The proposed scheme is not ideal but it is perfect.

Proof: Some of the participants are receiving shares as r_{ij} as well as y_{ij}, so the proposed scheme is not ideal. In the proposed scheme, at each and every level, we apply Shamir's (t, n) secret sharing scheme for share distribution. In Shamir's (t, n) secret sharing scheme, any less than t people cannot reconstruct the secret. Hence, our scheme is also perfect based on Shamir's perfect scheme.

6 Conclusions

A new secret sharing scheme that realizes the level ordered access structure is proposed, where multiple secrets are shared among the multilevels each corresponding to a level. The concepts of quadratic residues and discrete logarithm problem are used for sharing of secrets. An ordering concept is enforced in the levels during reconstruction of secrets. The reconstruction of secrets in each and every level is subjected to the cooperation of their immediate higher level. A verification phase has been added at each and every level to check whether the shares being submitted by the players during the reconstruction phase are correct or not.

References

1. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)
2. Blakley, G.R.: Safeguarding cryptographic keys. AFIPS **48**, 313–317 (1979)
3. Simmons, G.J.: How to (really) share a secret. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 390–448. Springer, New York (1990). https://doi.org/10.1007/0-387-34799-2_30
4. Tassa, T.: Hierarchical threshold secret sharing. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 473–490. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24638-1_26
5. Tassa, T., Dyn, N.: Multipartite secret sharing by bivariate interpolation. J. Cryptol. **22**, 227–258 (2009)
6. Ghodosi, H., Pieprzyk, J., Safavi-Naini, R.: Secret sharing in multilevel and compartmented groups. In: Boyd, C., Dawson, E. (eds.) ACISP 1998. LNCS, vol. 1438, pp. 367–378. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0053748
7. Tentu, A.N., Paul, P., Venkaiah, V.C.: Computationally perfect secret sharing scheme based on error-correcting codes. In: Martínez Pérez, G., Thampi, S.M., Ko, R., Shu, L. (eds.) SNDS 2014. CCIS, vol. 420, pp. 251–262. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54525-2_23
8. Tentu, A.N., Paul, P., Venkaiah, V.C.: Computationally perfect secret sharing schemes based on MDS codes. Int. J. Trust Manag. Comput. Commun. (IJTMCC) **2**(4), 353–378 (2014)
9. Basit, A., Kumar, N.C., Venkaiah, V.C., Moiz, S.A., Tentu, A.N., Naik, W.: Multi-stage Multi-secret sharing scheme for hierarchical access structure. In: 2017 IEEE International Conference on Computing, Communication and Automation (ICCCA), Noida (2017, in press)
10. Iftene, S.: Compartmented secret sharing based on the Chinese remainder theorem, Cryptology ePrint Archive, Report 2005/4 08 (2005)
11. Singh, N., Tentu, A.N., Basit, A., Venkaiah, V.C.: Sequential secret sharing scheme based on Chinese remainder theorem. In: IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1–6 (2016)
12. He, J., Dawson, E.: Multistage secret sharing based on one-way function. Electron. Lett. **30**(19), 1591–1592 (1994)
13. Harn, L.: Comment: multistage secret sharing based on one-way function. Electron. Lett. **31**(4), 262 (1995)
14. Chang, T.-Y., Hwang, M.-S., Yang, W.-P.: A new multi-stage secret sharing scheme using one-way function. Oper. Syst. Rev. **39**(1), 48–55 (2005)

15. Chanu, O.B., Tentu, A.N., Venkaiah, V.C.: Multi-stage multi-secret sharing schemes based on Chinese remainder theorem. In: International Conference on Advanced Research in Computer Science Engineering Technology (ICARCSET 2015), Unnao, India, vol. 17, no. 6 (2015)
16. Zhao, J., Zhang, J., Zhao, R.: A practical verifiable multi-secret sharing scheme. Comput. Stand. Interfaces **29**(1), 138–141 (2007)
17. Tentu, A.N., Mahapatra, B., Venkaiah, V.C., Prasad, V.K.: New secret sharing scheme for multipartite access structures with threshold changeability. In: International Conference on Advances in Computing, Communications and Informatics, ICACCI 2015, Kochi, India, 10–13 Aug 2015
18. Nojoumian, M., Stinson, D.R.: Sequential secret sharing as a new hierarchical access structure. J. Internet Serv. Inf. Secur. (JISIS) **5**(2), 23–31 (2015)
19. Dileep, K.P., Tentu, A.N., Venkaiah, V.C., Apparao, A.: Sequential secret sharing scheme based on level ordered access structure. J. Netw. Secur. **18**(5), 874–881 (2016)

Author Index