KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Optimisations of fully homomorphic encryption

Ilia Iliashenko

Supervisor:
Prof. dr. ir. Bart Preneel
Co-supervisor:
Prof. dr. ir. Frederik Vercauteren

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

May 2019

Optimisations of fully homomorphic encryption

Ilia ILIASHENKO

Examination committee:
Prof. dr. ir. Hugo Hens, chair
Prof. dr. ir. Bart Preneel, supervisor
Prof. dr. ir. Frederik Vercauteren, co-supervisor
Prof. dr. ir. Vincent Rijmen
Dr. Wouter Castryck
Prof. dr. Bruno Crispo
(KU Leuven and University of Trento)
Dr. Joppe W. Bos
(NXP Belgium)
Dr. Léo Ducas
(CWI, the Netherlands)
Prof. dr. Damien Stehlé
(École Normale Supérieure de Lyon)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

May 2019

Acknowledgements

First of all, I want to thank my supervisors, Bart Preneel and Frederik Vercauteren, for the opportunity to work in the truly amazing atmosphere of the COSIC group. Without their constant support and mentoring this thesis would never be finished. In particular, I want to thank Fre for his enthusiasm and dedication even when he had extremely busy and nervous times. I am indebted to you for teaching me how to do research and perceive the work of others.

I would like to thank the jury members for accepting to review this thesis. I appreciate the time they spent on assessing this manuscript and providing valuable feedback.

I would like to express my gratitude to my coauthors. It is a great luck and a great pleasure to work with such smart, witty and friendly people. I am looking forward to future collaborations with you.

I want to thank my colleagues for maintaining a friendly and cheerful atmosphere in COSIC. In particular, I thank Wouter, Carl and Elena for countless discussions on mathematics, crypto and less fascinating parts of our life.

Many thanks to Pela Noë who helped me a lot with transportation to Leuven and administration of my PhD program. I also thank Elsy and Wim for dealing with my inability to remember how to create reimbursement forms.

This work was supported by the European Commission through the ICT programme H2020 and the ERC Advanced Grant. The last year of my PhD program was financially supported by Prof. Nigel Smart. Thank you, Nigel. I would like to become so energetic and enthusiastic as you.

Last but not least, I want to thank my family. Thank you for love, support and relentless efforts to understand what I am doing. Ira, I owe you so much (including several thousands ice cream scoops) for keeping my life in balance and making Belgium my home.

Abstract

Fully homomorphic encryption (FHE) is a class of encryption algorithms that support any computation on encrypted messages without revealing anything about these messages in unencrypted form except for their maximal size. Using FHE, a party that owns private data can securely outsource computations on this data to another party. Due to this functionality, FHE finds many applications both in practice (e.g. cloud computing) and in the design of new cryptographic algorithms.

Since the seminal work of Gentry in 2009, FHE has become an active research area that revolves around the question how to make FHE efficient. In the recent decade, various schemes and optimisations were proposed that gradually decreased the computational overhead of homomorphic function evaluation. Nevertheless, these schemes still remains impractical for general industrial applications.

In this thesis, we propose and analyse several optimisations for FHE schemes.

First, we demonstrate that the security of the most efficient FHE schemes can be compromised by switching from the RLWE problem to a slightly simpler computational problem, called Scaled Canonical Gaussian-LWE (SCG-LWE). We provide an algebraic attack on SCG-LWE and present countermeasures against it.

Second, we design several algorithms that efficiently encode real- and complex-valued data for FHE evaluation. These algorithms endow FHE schemes with native real- and complex-number arithmetic, thus reducing the computational overhead of homomorphic circuits.

Third, we decrease the memory overhead of FHE by generalising the packing technique of Smart and Vercauteren. Using our algorithm, more plaintext messages can be packed into a single ciphertext in comparison to the previous methods.

All the aforementioned results are accompanied by practical examples that illustrate the efficiency gain of the presented techniques.

Beknopte samenvatting

Volledig homomorfe encryptie (FHE) is een klasse van encryptie-algoritmen die elke berekening op versleutelde gegevens ondersteunen zonder iets over deze gegevens in niet-versleutelde vorm bekend te maken, behalve met betrekking tot de maximale grootte. Met behulp van FHE kan een partij die de privégegevens bezit veilig berekeningen op deze gegevens uitbesteden aan een andere partij. Vanwege deze functionaliteit kan FHE worden gebruikt voor vele toepassingen zowel in de praktijk (bijvoorbeeld cloud computing) als in het ontwerp van nieuwe cryptografische algoritmen.

Sinds het baanbrekende werk van Gentry in 2009 is FHE een actief onderzoeksgebied dat draait om de vraag hoe FHE efficiënt kan worden gemaakt. In het recente decennium werden verschillende schema's en optimalisaties voorgesteld die de computationele overheadkosten van homomorfe functie-evaluatie geleidelijk verminderden. Niettemin blijven deze overheadkosten nog steeds onpraktisch voor algemene industriële toepassingen.

In dit proefschrift stellen we verschillende optimalisaties van FHE-schema's en hun analyse voor.

Ten eerste laten we zien dat de beveiliging van de meest efficiënte FHE-schema's kan worden aangetast door over te schakelen van het RLWE-probleem naar een iets eenvoudiger computerprobleem, Scaled Canonical Gaussian-LWE (SCGLWE). We bieden een algebraïsche aanval op SCGLWE aan en presenteren een aantal tegenmaatregelen.

Ten tweede ontwerpen we verschillende algoritmen die op een efficiënte manier reële en complexe gegevens voor FHE-evaluatie coderen. Deze algoritmen voorzien FHE-schema's van berekeningen op reële en complexe getallen, waardoor de computationele overheadkosten van homomorfe circuits worden verminderd.

Ten derde verminderen we de geheugenoverhead van FHE door de verpakkingstechniek van Smart en Vercauteren te generaliseren. Met behulp van ons algoritme kunnen meer klaarteksten in één cijfertekst worden ingepakt in vergelijking met de vorige methoden.

Alle bovengenoemde resultaten worden vergezeld door praktische voorbeelden die de efficiëntieverbetering van de gepresenteerde technieken illustreren.

List of Acronyms

List of Hard Problems

Contents

List of Figures

Part I

Introduction

Chapter 1

General introduction and scope

1.1 Outsourced computation

We live in a highly digitalised era, where personal computers and smartphones have become indispensable. These devices are usually equipped with computational resources that are unable to provide all services the modern user can ask for. Moreover, some of these services demand the deployment of expensive hardware and software infrastructure, which is prohibitively costly for ordinary people or even small-scale companies. To overcome this issue, personal gadgets, which often have constant access to global communication networks, e.g. the Internet, can request additional computational resources from more powerful network members. This rationale gave rise to new business models that shift computational burden from computationally weak personal devices to more powerful commercial computing clusters designed for specific purposes. In other words, computation is *outsourced*.

Formally, by outsourced computation, we understand a protocol where party A sends data X to party B and party B performs computation on X, and returns the result to party A or to another party. Despite its simplicity, this definition covers various scenarios including the one above with a computationally weak user and a powerful server. Furthermore, there can be more than two parties involved in outsourced computation and their roles may vary. For example, in peer-to-peer networks, each party can have a similar computational capability and be interested in sharing a computational burden to jointly provide a service.

The variety of outsourced computation problems results in a variety of methods to solve them. For instance, one major class of solutions refers to so-called "cloud" providers that mainly offer storage capacities for user data and additional functionality, e.g. statistical analysis. Many well-known companies such as Microsoft, Amazon, Google, and Apple are present in this market. In addition to cloud providers, outsourced computation includes numerous service operators, who provide access to specific software or service, e.g. search engines, social networks, messengers, games etc. It is therefore easy to observe that outsourced computation is ubiquitous in everyday life.

In the majority of outsourced computation scenarios data owners wish to send their sensitive and private information to the party that performs the computation. In this case, security concerns inevitably arise as this data needs to remain private during information transfer and computation. Secure information transfer is not an issue in practice as it can be solved by various well-known public and private cryptographic solutions. However, once the data has been securely sent to the computing party, computation must be performed securely as well.

One solution to the secure computation problem is to allow the computation provider to decrypt private data and compute on it in unencrypted form. In fact, this is what modern commercial and public systems offer. However, this solution assumes that the data owners trust the provider. They should believe that the cloud infrastructure prevents any security breach, data loss and misuse. In practice, these security issues are impossible to avoid as the cloud provider can behave dishonestly. Therefore, it is extremely hard to convince customers to outsource highly sensitive data to the cloud. As a result, the data owners may wish stronger security guarantees such as keeping their data secret even from the computing party.

An alternative solution is to empower the computation provider or a group of users with cryptographic techniques that allow computing on encrypted data without decrypting it. This solution is less efficient than the first one as it requires a larger bandwidth and any operation on encrypted data may have a high computational overhead. However, no trust in the computing party is necessary in this case apart from the guarantee that computation is done correctly.

1.2 Homomorphic encryption

This work is dedicated to homomorphic encryption, a cryptographic technique for secure outsourced computation. First, we need to understand what the

term "encryption" means. Encryption is the process of encoding data from its original form, called a *plaintext*, into another form, called a *ciphertext*, such that the plaintext can be recovered from the ciphertext in reasonable time only by authorized parties. The inverse operation that reveals the plaintext to authorized parties is called *decryption*. To perform encryption and decryption, communicating parties require auxiliary pieces of information called *keys*, which must be generated prior to encryption. A triple of encryption, decryption and key generation algorithms constitutes an *encryption scheme*. If both parties use the same key to encrypt and decrypt messages, the encryption scheme is referred to as *symmetric*. In a *public-key* scheme, the party that wishes to decrypt messages generates two keys, which are mathematically related. The first key is *public* such that everyone who is willing to communicate with the key owner encrypts messages using this key. Given these ciphertexts, the key owner uses the second key, which is kept *secret*, to decrypt them. The mathematical relation between the public and the secret keys makes it computationally hard for an adversary to decrypt a ciphertext using only the public key.

Homomorphic encryption (HE) is a class of encryption schemes that allow computations on encrypted messages without exposing them in unencrypted form, i.e. decrypting them. In particular, given a ciphertext $\mathtt{ct}(x)$ encrypting a message x, and a function f from a predefined class of functions on plaintexts, any party can compute a ciphertext $f'(\mathtt{ct}(x)) = \mathtt{ct}(f(x))$ where f' operates on ciphertexts. This implies that, besides the encryption, decryption and key generation functions, an HE scheme should have the ciphertext analogues of basic functions on plaintext messages. For instance, to add two plaintext messages $\mathtt{msg}_1$ and $\mathtt{msg}_2$, an HE scheme must have a function `Add` that, given two ciphertexts $\mathtt{ct}(\mathtt{msg}_1)$ and $\mathtt{ct}(\mathtt{msg}_2)$, outputs a ciphertext $\mathtt{ct}(\mathtt{msg}_1 + \mathtt{msg}_2)$. Depending on the number and the type of supported basic functions, HE schemes are divided into three classes: partially, somewhat and fully homomorphic encryption schemes.

Partially homomorphic encryption (PHE) schemes support only homomorphic addition or only homomorphic multiplication. The examples of PHE schemes have been known since Rivest et al. [106], who introduced the idea of homomorphic encryption. Even though four out of the five examples in [106] were broken [28], the remaining scheme, the well-known RSA cryptosystem [107], is still considered secure against classical adversaries. RSA and the ElGamal cryptosystem [52] are *multiplicatively* homomorphic, i.e. they support multiplication of encrypted messages. Another example is the famous Paillier cryptosystem [96], which is *additively* homomorphic. In certain use cases, PHE schemes turn out to be a useful cryptographic primitive, e.g. in secure voting protocols [38, 14, 72, 78].

Currently known PHE schemes are unable to compute a complete set of

Boolean operators, i.e. the NAND operator. Instead, they support either addition or multiplication in a commutative ring, which contains the Boolean set $\{0, 1\}$. However, the NAND operator requires both homomorphic addition and multiplication as can be seen in the following expression

$$x_1 \texttt{ NAND } x_2 = 1 - x_1 \cdot x_2, \quad x_1, x_2 \in \{0, 1\}.$$

Therefore, the functionality of PHE schemes is fundamentally limited.

If an HE scheme supports a limited number of both homomorphic additions and multiplications, it is referred to as *somewhat* homomorphic (SHE). The first encryption schemes that can be considered SHE are the so-called "Poly-Cracker" schemes introduced in the early 1990s [55, 13, 80]. These schemes are based on the ideal remainder problem over multivariate rings. Here, encryption is performed via masking a plaintext with a random element of a publicly known ideal. Unfortunately, almost all such schemes lack security guarantees [86, 5]. The SHE scheme by Sander et al. in 1999 [108], employs an additive PHE scheme to evaluate arbitrary circuits. Similarly to the Poly-Cracker schemes, the drawback of this scheme is that the ciphertext size increases exponentially with the depth of the circuit. The next significant improvement of SHE was given by Boneh et al. in 2005 [18]. Based on the Paillier scheme and bilinear maps, this scheme is compact such that the ciphertext size remains the same throughout homomorphic computations. Unfortunately, this scheme allows only one homomorphic multiplication, which can be used to compute quadratic polynomials.

In 2009, Gentry published his seminal thesis [60] introducing a new SHE scheme based on the ideal coset problem over ideal lattices. Each ciphertext of this scheme contains a special noisy term that must be small enough, otherwise decryption fails. This noise increases after every homomorphic operation. Therefore, during homomorphic function evaluation it can reach the so-called decryption threshold such that the resulting ciphertext is no longer decryptable. Hence, the noise must be somehow made smaller, or "refreshed", by the computation provider, which has no access to the secret key. Gentry proposed to encrypt a ciphertext that needs to be refreshed under the public key and then homomorphically evaluate the decryption function using the secret key encrypted under the same public key. This encryption of the secret key is called a *bootstrapping key*. It is assumed that the bootstrapping key leaks no information about the secret key, which is referred to as the *circular security* assumption. Gentry called this "refreshing" method *bootstrapping* and the schemes able to do such computations *bootstrappable*. The result of bootstrapping is an encryption of the original message. As the decryption function removes all the noise of the input ciphertext, the noise level of the bootstrapping result is equal to the noise introduced solely by the bootstrapping function.

Gentry showed that bootstrappable SHE schemes result in *fully* homomorphic encryption (FHE) schemes that support any arithmetic circuit. If the noise introduced by bootstrapping is small enough, one more `NAND` operator can be evaluated before the next bootstrapping. Hence, the bootstrappable scheme can compute any number of `NAND` gates, so it becomes unbounded fully homomorphic.

Even though Gentry's scheme was recognized as a profound theoretical breakthrough, it is highly inefficient due to large parameters as was shown in the subsequent implementations [112, 61]. Following Gentry's blueprint, many researchers have been trying to develop more practical FHE schemes. The outcome of these efforts can be partitioned into three generations.

Gentry's scheme and the scheme of van Dijk et al. [115] belong to the first generation of FHE schemes. The main drawback of these schemes is the rapid noise growth: the product of two ciphertexts with noise magnitude E results in a ciphertext with noise magnitude roughly E^2. As a result, evaluating a degree-d polynomial leads to noise magnitude of E^d. Such noise growth is prohibitively fast to evaluate the decryption function. Therefore, the bootstrapping algorithm involves auxiliary operations that require ad hoc hardness assumptions. Even for small circuits that can be evaluated without bootstrapping, these schemes demand large parameters, which are not feasible in practice.

The second generation schemes such as Brakerski-Vaikuntanathan [26], Brakerski [22], Brakerski et al. [24], made important progress in efficiency. These schemes introduced new techniques that efficiently suppress the noise growth after homomorphic multiplication: the noise size increases logarithmically rather than linearly in the polynomial degree of the homomorphic function. Furthermore, it is possible to significantly decrease the ciphertext-to-plaintext expansion ratio by encrypting several messages with a single ciphertext [23]. The security of these schemes is based on the Learning with Errors (LWE) problem [105], which revolves around solving systems of noisy linear equations. LWE is as hard to solve as well-established lattice problems.

The "golden" generation 2+ includes the well-known Brakerski-Gentry-Vaikuntanathan [24], Fan-Vercauteren [54] and HEAAN [35] schemes. Here, the security is based on the variant of LWE over polynomial rings, called RLWE. The algebraic structure of RLWE shrinks the size of secret and public keys, speeds up homomorphic operations via the number theoretic transform and also leads to new optimizations such as new packing algorithms [113, 35], various bootstrapping improvements [62, 8, 69, 34]. Given these optimizations, the computational overhead of the RLWE schemes was shown to be only poly-logarithmic in the security parameter [63]. The SHE parts of these schemes are considered the most efficient for arithmetic circuits, which is supported by their success in public competitions [74].

The third generation [65, 49, 36] is characterized by the use of matrix arithmetic on LWE and RLWE ciphertexts. Although these schemes are less efficient than RLWE-based schemes for arithmetic circuits, their bootstrapping functions are much faster. In addition, some of these schemes support a look-up table functionality that allows to evaluate non-polynomial homomorphic functions.

We also mention FHE schemes based on the NTRU problem [73] such as the LTV scheme [88] and YASHE [19]. The LTV scheme introduced the concept of multi-key FHE, where the computation provider can compute on ciphertexts encrypted by several data owners. In addition to NTRU, these schemes rely on the Decisional Small Polynomial Ratio assumption, whose hardness was compromised by the attack of Albrecht et al. [4].

Another interesting but impractical proposal is the FHE scheme of Doröz et al. [46], which relies on the subset sum problem and the new problem of finding finite field isomorphisms.

In addition to the obvious usefulness for outsourced computation, FHE can be applied in a wide range of applications as indicated below:

- program execution [60, 27],
- private information retrieval [82, 60, 117],
- indistinguishability obfuscation [57],
- multi-party computation [88, 95, 101],
- verifiable computation [59],
- zero-knowledge proofs [68],
- verification of quantum computations [93].

1.3 Secure computation methods

FHE provides a universal tool to address many problems in secure outsourced computation. However, a variety of other cryptographic techniques exists that compete with FHE schemes in some cryptographic applications. Below we outline these techniques and compare their functionality with FHE.

Secure multi-party computation, or MPC [116], solves the problem of joint computation by a group of users, who want to keep their inputs private. In MPC, each participant of a secure computation protocol encrypts his/her input

data and actively participates in the computation. Any participant can decrypt the result of this computation. Therefore, MPC contrasts with FHE, in which only the data owner but not the computing party can encrypt and decrypt information. MPC requires the on-line presence of the users, which involves an extensive on-line communication to obtain the evaluation result. On the contrary, FHE needs only two communication rounds and does not demand the data owner to be on-line during the computation phase.

The most famous technique for MPC with two parties is *Yao's garbled circuit* (GC). GC is considered as one of the fastest MPC techniques as its computation involves symmetric cryptographic primitives during the circuit evaluation phase. Since GC is applicable only for evaluating Boolean circuits, GC demands communication complexity proportional to the size of the Boolean circuit, which is also proportional to the size of the input data. FHE requires much smaller communication complexity as only encrypted input data and encrypted secret keys must be transferred.

Introduced by Dwork et al. in 2005 [50], *differential privacy* deals with the scenario where one party releases some statistics on a private database to the public. The goal of this party is to ensure that the released statistics does not leak any information about individual records of the database. In other words, any adversary cannot derive private information from the published statistical results. In comparison to FHE, differential privacy operates directly on unencrypted data and does not provide cryptographic security.

Private set intersection (PSI) is a class of cryptographic protocols that allows two parties to compute the intersection of their sets without exposing anything except the intersection. A vast number of PSI realizations exist (e.g. [103, 81]), most of which require significant communication overhead. Recently, this overhead was significantly decreased by using SHE [30].

Private information retrieval (PIR) [37], retrieves a unique item which matches a given query in the outsourced data. The server that keeps the outsourced data must remain oblivious to the query and the retrieval results. It is assumed that the query has at most one unique match in the outsourced data. If no such item exists, the encryption of "none" is returned. PIR realisation using FHE was already given in the work of Gentry [60], who also demonstrated how FHE reduces communication costs of PIR protocols in comparison to other solutions.

Searchable symmetric encryption (SSE) [114, 44] allows encryption of the outsourced data before being sent to the server, while having the ability to search over it. This functionality can be also realised using FHE as demonstrated by Gentry [60]. In contrast to FHE, the existing SSE solutions employ symmetric-key block or stream ciphers to encrypt the data, which results in minimal

ciphertext expansion. Hence, the outsourced data and the query must be encrypted under the same secret key. To compare two encrypted values, it is enough to compare their encryptions bit by bit. As a result, SSE assumes "minimal leakage" of private information which includes positions of successfully matched patterns, their number and size, and the size of outsourced data. FHE avoids such leakage by randomising encryptions.

1.4 Motivation for this work

Despite the advantages presented in the previous section, FHE schemes suffer from several drawbacks outlined below.

FHE does not guarantee the integrity of encrypted data. Assume that the data owner sends his encrypted data to the server, which can evaluate a function f. After the data owner receives the encrypted results, they may ask whether these ciphertexts are a real outcome of the function f. In fact, such verification is impossible in existing FHE schemes without using additional cryptographic primitives, e.g. homomorphic signatures [67]. This problem is the subject of another very active research area known as *verifiable computation*.

Bootstrapping is still too inefficient for practical applications. As we mentioned in Section 1.2, several SHE schemes exist that evaluate arithmetic circuits with only polylogarithmic overhead. Unfortunately, these schemes are accompanied by inefficient bootstrapping algorithms, which are rarely used in practice. To avoid bootstrapping, one can scale the parameters of these schemes according to the noise growth inside encrypted messages. In this work, we mainly focus on possible optimisations of FHE schemes to reduce their encryption parameters and postpone bootstrapping.

Redefinition of standard computational problems for FHE leads to efficient attacks. The most efficient FHE schemes are based on the RLWE problem, which must be properly instantiated to provide reasonable security guarantees. Given a target security level, one can find corresponding standard parameter sets in the literature. However, some researchers [51, 53, 31, 32] slightly redefine the RLWE problem with non-standard parameters, thus switching to another computational problem. This problem leads to schemes that deviate from the original security assumptions and thus can be broken.

Large ciphertext-to-plaintext expansion ratio. In the existing FHE schemes, encryption is performed via masking the plaintext by large integers, large-degree polynomials with big coefficients or matrices of large dimension. This implies that the encryption of one bit can expand up to several kilobytes. Since homomorphic operations are performed on ciphertexts, this expansion directly affects the performance of homomorphically evaluated circuits. As discussed in Section 1.2, several packing techniques were introduced to solve this issue. In particular, packing allows combining several plaintexts into one that is fed to the encryption function. Another method to reduce the expansion overhead is encoding numerical data directly into the plaintext space of an FHE scheme.

Lack of encoding algorithms for real and complex numbers. By definition, FHE can perform the complete set of Boolean operations, thus allowing (in theory) evaluation of any computable function. However, the inherent operations of FHE schemes are addition and multiplication in commutative rings. Hence, it would be more natural to homomorphically evaluate arithmetic circuits. Furthermore, arithmetic circuits of certain functions have much smaller multiplicative depth than their Boolean circuit counterparts. Arithmetic circuits significantly reduce encryption parameters and can avoid costly bootstrapping.

While working with arithmetic circuits, it is desirable to deviate from the binary-bit representation of data and encode numerical values (real, complex numbers) directly into the plaintext space. For this purpose, Dowlin et al. [47] suggested to use the ternary expansion of a real number $a = \sum a_i 3^i$ with $a_i \in \{-1, 0, 1\}$ and then replace powers of 3 by powers of the variable X. Encoding of complex numbers can be done via the cyclotomic integer approximation [40]. However, both methods inefficiently consume the plaintext space and require large encryption parameters to compute deep homomorphic circuits. A more efficient technique was proposed in HEAAN [35], which allows encoding of several complex numbers into a single plaintext. The main idea is to allow the ciphertext noise to mix with lower bits of encoded complex numbers. Hence, exact arithmetic is only possible if the noise is small enough, which is hard to achieve in reasonably deep arithmetic circuits.

In this work we address the last three issues.

1.5 Contributions

We analyse the security of a variant of the RLWE problem, called SCG-LWE. This variant simplifies the error generation at the cost of skewing the error distribution in relation to the power basis of the underlying number field. We

show that this skewness can be efficiently exploited in a simple algebraic attack on the search SCG-LWE problem. This attack can be avoided by using larger parameters; however, it is still an open question whether the SCG-LWE problem is hard in this case.

We demonstrate that the plaintext space R_t is significantly underused by the existing encoding algorithms for real numbers: encoded data is concentrated in a small number of polynomial coefficients, which causes faster coefficient growth. To accommodate enough space for this growth, the plaintext modulus must be large, which leads to faster noise growth, bigger encryption parameters and, as a result, less efficient implementations of homomorphic functions. To remedy this issue, we design a new encoding algorithm that significantly reduces the coefficient growth. It leads to much smaller encryption parameters such that homomorphic evaluation becomes several times faster in some applications. Furthermore, this encoding algorithm is flexible in a way that the size of encodings can be controlled to fit a given plaintext space.

We show that the new encoding algorithm proves to be useful for packing several messages, which are different in magnitude. Our technique generalises the SIMD method [113] and can increase the packing capacity of the plaintext space R_t manyfold. Nevertheless, our encoding method, as well as its precursor, highlights that the algebraic structure of the ring R_t is inherently restrictive when using real- or complex-number arithmetic.

We design a variant of the FV scheme, whose plaintext space embeds Gaussian integers modulo a large number. Hence, complex number arithmetic can be natively performed in this scheme. The main idea of this scheme is to take a polynomial $X^m + b$ as the plaintext modulus. The noise growth of this scheme is so slow that it can evaluate much deeper homomorphic circuits than the original FV scheme. In comparison to the methods which separately encode the imaginary and the real parts of a complex number, our scheme consumes half the memory and requires less homomorphic operations to compute complex multiplication. However, the packing capacity of our scheme is very low, which poses a question whether other polynomial plaintext moduli can increase it.

1.6 Outline

The main results of this work are presented in Part II in the form of peer-reviewed and published papers. In particular, Chapters 6 and 7 present algebraic attacks on a simplification of the RLWE problem. Chapter 8 is dedicated to an application of SHE to secure energy forecasting. This work highlighted practical issues with the existing encoding methods for real numbers, which were partially

addressed by a new encoding algorithm given in Chapter 9. Chapter 10 presents a generalisation of the SIMD packing technique [113]. Chapter 11 is dedicated to a new efficient SHE scheme that is built specifically for complex number arithmetic.

Part I introduces the necessary background to understand and critically assess the contributions. Chapter 2 contains basic mathematical concepts used throughout this work. Chapter 3 introduces hard computational problems on which the security of the most efficient FHE schemes is based. Chapter 4 describes the most efficient SHE schemes based on the RLWE problem. Finally, Chapter 5 summarises the contributions of this thesis and suggests potential research topics.

Chapter 2

Preliminaries

This chapter introduces the mathematical notation and the basic mathematical facts that will be used throughout this thesis.

2.1 Basic notation

For an integer a, let $[a]_q$ denote the unique integer in $[-q/2, q/2)$ with $[a]_q \equiv a \mod q$. The quotient ring of integers modulo a is denoted $\mathbb{Z}_a$. The remainder of $a \mod q$ in $[0, q-1]$ is denoted by $|a|_q$.

For a natural number a, we denote the set of integers $\{1, \ldots, a\}$ by $[a]$. Similarly, for any $a, b \in \mathbb{Z}, a \leqslant b$, the integer set $\{a, a+1, \ldots, b-1, b\}$ is denoted by $[a, b]$.

Vectors and matrices are denoted by boldface lower- and upper-case letters, respectively. Vectors are written in column form. The coefficient-wise product of two vectors $\mathbf{v}$ and $\mathbf{w}$ is denoted by $\mathbf{v} \circ \mathbf{w}$.

For a vector $\mathbf{v} = (v_1, \ldots, v_n)^\intercal \in \mathbb{C}^n$, the Euclidean norm is defined as $\|\mathbf{v}\| = \left(\sum_{i=1}^n |v_i|^2\right)^{1/2}$. Its infinity norm is equal to $|\mathbf{v}|_\infty = \max_{i=1}^n |v_i|$.

The expression $a \leftarrow \mathcal{D}$ means "a is sampled from a distribution $\mathcal{D}$". For a set S, the expression $a \xleftarrow{\$} S$ means that "a is sampled uniformly random from S".

For any two integers a and b, their greatest common divisor is denoted by (a, b).

2.2 Number fields and rings

A subset $\mathcal{I}$ of a commutative ring $\mathcal{R}$ is called an ideal if it is an additive subgroup of $\mathcal{R}$ closed under multiplication by elements of $\mathcal{R}$, i.e. $a\mathcal{I} \subset \mathcal{I}$ for any $a \in \mathcal{R}$. If an ideal $\mathcal{I}$ is generated by some $u \in \mathcal{R}$, i.e. $\mathcal{I} = u\mathcal{R}$, this ideal is called *principal* and denoted as $\mathcal{I} = \langle u \rangle$.

Let $f \in \mathbb{Z}[X]$ be a monic irreducible polynomial of degree n. The quotient ring $K = \mathbb{Q}[X]/\langle f \rangle$ is a number field of degree n and f is its *defining polynomial*.

Let $R \subset K$ denote the ring of integers of K, i.e. the set of all algebraic integers that are contained in K. Further, we will use the quotient ring $R_q = R/\langle q \rangle$ with $q \in \mathbb{Z}$.

If f is such that $R = \mathbb{Z}[X]/\langle f \rangle$, then K is called a *monogenic* number field and f is a monogenic polynomial. In this case, both K and R have a *power basis* $1, X, \ldots, X^{n-1}$. An element $a = \sum a_i X^i \in K$ can be viewed as a vector of its coefficients $\mathbf{a} = (a_0, \ldots, a_{n-1})^\intercal \in \mathbb{Q}^n$ with relation to the power basis of K. Note that an element in R_q can be viewed as an element from R with its coefficient vector reduced by the coefficientwise $[\cdot]_q$ operation. We define the Euclidean norm of $a \in K$ as the Euclidean norm of its coefficient vector, namely $\|a\| = \|\mathbf{a}\|$. By analogy, the infinity norm of $a \in K$ is defined as $|a|_\infty = |\mathbf{a}|_\infty$. We call the quantity

$$\delta_R = \sup \left\{ \frac{|ab|_\infty}{|a|_\infty |b|_\infty} : a, b \in R \right\}$$

the *expansion factor* of R.

An important class of monogenic number fields is cyclotomic fields.

Let m be a positive integer and $\zeta_m = \exp(2\pi i/m)$ be a primitive complex mth root of unity. We define the mth *cyclotomic polynomial* $\Phi_m(X)$ as

$$\Phi_m(X) = \prod_{i:(i,m)=1} (X - \zeta_m^i).$$

For instance, if $m = 2^k$ then $\Phi_m(X) = X^{m/2} + 1$. It is well known that Φ_m is an irreducible polynomial in $\mathbb{Z}[X]$ of degree $n = \phi(m)$ where $\phi(x)$ is the Euler totient function. We can therefore define a *cyclotomic field* $K = \mathbb{Q}[X]/\langle \Phi_m(X) \rangle$. All cyclotomic fields are monogenic. If m is a power of 2, the expansion factor of the ring of integers of K satisfies $\delta_R = n$ [45].

2.3 Canonical embedding

Recall that n is the degree of $K/\mathbb{Q}$. Let s_1 be the number of real roots of $f(X)$, then $n - s_1 = 2s_2$ is its number of its complex roots. Therefore, the field $K = \mathbb{Q}[X]/\langle f \rangle$ has exactly s_1 real embeddings into $\mathbb{R}$ and $2s_2$ complex embeddings into $\mathbb{C}$ denoted by σ_i such that

$$\sigma_i(a(X)) \mapsto a(\alpha_i)\,,$$

where α_i is a root of f. Since the complex roots come in conjugate pairs, we can order them such that $\overline{\alpha_{s_1+k}} = \alpha_{s_1+s_2+k}$ for any $k \in [s_2]$. We define the *trace* of an element $x \in K$ as the sum of its embeddings $\mathrm{Tr}(x) = \sum_{i=1}^{n} \sigma_i(x)$. Note that Tr is an additive homomorphism from K to $\mathbb{Z}$.

The *canonical embedding* (also known as the Minkowski embedding) $\sigma : K \to \mathbb{C}^n$ is then defined as follows:

$$\sigma(a) = (\sigma_1(a), \dots, \sigma_{s_1}(a), \sigma_{s_1+1}(a), \dots, \sigma_{s_1+s_2}(a), \overline{\sigma_{s_1+1}}(a), \dots, \overline{\sigma_{s_1+s_2}}(a))\,.$$

In number fields, the canonical embedding maps the power basis $1, X, \dots, X^{n-1}$ of K to the columns of the Vandermonde matrix $\mathbf{\Sigma} = (\alpha_i^{j-1})_{i,j}$ with $i, j \in [n]$.

The image of σ is contained in the following space

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

This space is a ring with relation to component-wise addition and multiplication. Since $\sigma_i(a)\sigma_i(b) = \sigma_i(ab)$ and $\sigma_i(a) + \sigma_i(b) = \sigma_i(a+b)$ for any i, the canonical embedding is a ring homomorphism from K to H. Hence, the elements of K can be endowed with the *canonical norm* equal to $\|a\|^{\mathrm{can}} = |\sigma(a)|_\infty$. For any $a, b \in R$, it holds

$$\|ab\|^{\mathrm{can}} \leqslant \|a\|^{\mathrm{can}} \|b\|^{\mathrm{can}}\,.$$

As shown in [45], in cyclotomic number fields of order m we have $|a|_\infty \leqslant C_m \cdot \|a\|^{\mathrm{can}}$ for some constant C_m. In particular, $C_m = 1$ if m is a power of 2.

Looking at the basis of H over $\mathbb{R}^n$ given by the columns of the unitary matrix

$$\mathbf{B} = \begin{pmatrix} \mathbf{I}_{s_1 \times s_1} & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}}\mathbf{I}_{s_2 \times s_2} & \frac{\mathbf{i}}{\sqrt{2}}\mathbf{I}_{s_2 \times s_2} \\ 0 & \frac{1}{\sqrt{2}}\mathbf{I}_{s_2 \times s_2} & -\frac{\mathbf{i}}{\sqrt{2}}\mathbf{I}_{s_2 \times s_2} \end{pmatrix},$$

it is easy to see that H is isomorphic to $\mathbb{R}^n$ with relation to the Hermitian inner product. As a result, the transformation from $\mathbb{R}^n$ represented in its standard basis to K represented by its polynomial power basis is defined via the matrix product $\mathbf{\Sigma}^{-1} \cdot \mathbf{B}$.

2.4 Lattices

An n-dimensional *lattice* of rank $k \leqslant n$ is a discrete additive subgroup of $\mathbb{R}^n$, namely

$$\mathcal{L}(\mathbf{B}) = \{\mathbf{Bx} : \mathbf{x} \in \mathbb{Z}^k\}, \mathbf{B} \in \mathbb{R}^{n \times k}\,,$$

where the columns of matrix $\mathbf{B}$ constitute a basis of $\mathcal{L}(\mathbf{B})$ over $\mathbb{R}$. In this work, we focus on lattices of full rank $k = n$.

For any lattice $\mathcal{L}$, the *dual lattice* of $\mathcal{L}$ is defined as $\mathcal{L}^* = \{\mathbf{x} \in \mathbb{R}^n : \langle \mathcal{L}, \mathbf{x} \rangle \subseteq \mathbb{Z}\}$. If $\mathbf{B}$ is the basis of the lattice $\mathcal{L}$, then $\mathbf{D} = (\mathbf{B}^\intercal)^{-1}$ is the *dual basis* of $\mathbf{B}$. Any dual basis of a lattice $\mathcal{L}$ is a basis of its dual $\mathcal{L}^*$.

The *length* of a lattice vector is equal to its Euclidean norm. The length of a shortest non-zero lattice vector of $\mathcal{L}$ is called the *minimum distance* of $\mathcal{L}$ and denoted by $\lambda_1(\mathcal{L})$. More generally, for $i \in [n]$ the real number

$$\lambda_i(\mathcal{L}) = \inf\{d : \mathcal{L} \text{ has } i \text{ linearly independent vectors of length at most } d\}$$

is called the *ith successive minimum* of $\mathcal{L}$. See Figure 2.1 for illustration.

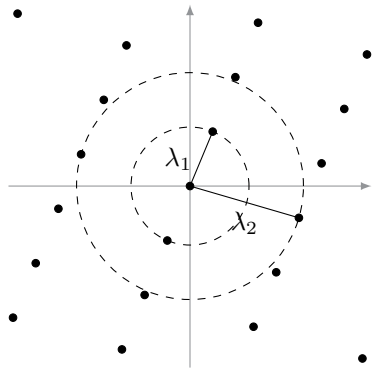


Figure 2.1: The 2-dimensional lattice generated by $(0.5, 1.2)^\intercal$ and $(2.4, -0.7)^\intercal$ with the minimal distance $\lambda_1 = 1.3$ and the second successive minimum $\lambda_2 = 2.5$.

Banaszczyk [12] proved the following connection between lattices and their duals, called the first transference theorem. For any rank-n lattice $\mathcal{L}$ it holds

$$1 \leqslant \lambda_1(\mathcal{L}) \cdot \lambda_n(\mathcal{L}^*) \leqslant n\,.$$

2.5 Fractional ideals

A non-zero ideal of the ring of integers R is called *integral*. The *(absolute) norm* of an integral ideal $\mathcal{I}$ is defined as the cardinality of the quotient ring of R modulo $\mathcal{I}$, i.e. $N(\mathcal{I}) = |R/\mathcal{I}|$.

A *fractional ideal* is a subset $\mathcal{I} \subset K$ such that $d\mathcal{I}$ is an integral ideal of R for some $d \in R$. If a fractional ideal $\mathcal{I}$ is generated by some $u \in K$, i.e. $\mathcal{I} = uR$, this ideal is called *principal* and denoted as $\mathcal{I} = \langle u \rangle$. Every fractional ideal $\mathcal{I}$ is a free $\mathbb{Z}$-module of rank n, and therefore $\mathcal{I} \otimes \mathbb{Q} = K$. Hence, its image $\sigma(\mathcal{I})$ under the canonical embedding can be viewed as a full-rank lattice in the space H, which is called an *ideal lattice*.

The sum of two (fractional or integral) ideals $\mathcal{I}$ and $\mathcal{J}$ is the set $\mathcal{I} + \mathcal{J} = \{x + y : x \in I, y \in J\}$. Note that $\mathcal{I} + \mathcal{J}$ is the smallest (fractional or integral) ideal containing $\mathcal{I}$ and $\mathcal{J}$. Two integral ideals $\mathcal{I}$ and $\mathcal{J}$ of R are said to be *co-prime* if $\mathcal{I} + \mathcal{J} = \langle 1 \rangle$, i.e. $\mathcal{I} + \mathcal{J} = R$. The product of $\mathcal{I}$ and $\mathcal{J}$ is defined as $\mathcal{I}\mathcal{J} = \{\sum x_i y_i : x_i \in \mathcal{I}, y_i \in \mathcal{J}\}$. For integral ideals, $N(\mathcal{I}\mathcal{J}) = N(\mathcal{I})\, N(\mathcal{J})$.

The *dual ideal* of a fractional ideal $\mathcal{I}$ is defined as

$$\mathcal{I}^\vee = \{x \in K : \mathrm{Tr}(x\mathcal{I}) \subseteq \mathbb{Z}\}\,.$$

Since $\mathrm{Tr}(xy) = \sum_i \sigma_i(x)\sigma_i(y) = \langle \sigma(x), \overline{\sigma(y)} \rangle$, the ideal lattice of $\mathcal{I}^\vee$ is equal to the complex conjugate of the dual of the ideal lattice $\sigma(\mathcal{I})$, i.e. $\sigma(\mathcal{I}^\vee) = \overline{\sigma(\mathcal{I})^*}$.

The ring of integers R is also a fractional ideal. Its dual $R^\vee$ is called the *codifferent*. If K is monogenic, i.e. $R = \mathbb{Z}[X]/\langle f \rangle$, the codifferent is generated by $1/f'(\alpha)$ where α is a root of f. In $2n$th cyclotomic number fields with n a power of two, the defining polynomial is $f(X) = X^n + 1$. Thus, for a primitive $2n$th root of unity ζ, we obtain $f'(\zeta) = n\zeta^{n-1}$. Since ζ^{n-1} is a unit in R, the ideal $(1/f'(\zeta))R$ is equal to $n^{-1}R$, thus leading to $R^\vee = \left\langle n^{-1} \right\rangle$. We denote the quotient $R^\vee/\langle q \rangle$ by $R_q^\vee$.

2.6 Discrete Gaussian distribution

Let Γ_r be the normal Gaussian distribution on $\mathbb{R}$ with mean 0 and standard deviation $r/\sqrt{2\pi}$ for some *width parameter* $r \in \mathbb{R}^+$ such that $\Gamma_r(x) = r^{-1} \exp(-\pi x^2/r^2)$.

We can extend Γ_r to the real vector space $\mathbb{R}^n$ by sampling $\mathbf{x} = (x_1, \ldots, x_n)$ where each x_i is independently drawn from the one-dimensional distribution Γ_r. This defines a *spherical* Gaussian distribution with probability density

function $\Gamma_r(\mathbf{x}) = r^{-1}\exp(-\pi\|\mathbf{x}\|/r^2)$. For any countable set S, we define $\Gamma_r(S) = \sum_{\mathbf{x}\in S}\Gamma_r(\mathbf{x})$.

If $(x_1,\ldots,x_n)$ is sampled from $\Gamma_{r_1}\times\ldots\times\Gamma_{r_n}$ with $r_{j+s_1+s_2} = r_{j+s_1}$ for each $j \in [s_2]$, this distribution is called an *elliptical* Gaussian distribution. We denote it by $\Gamma_{\mathbf{r}}$ with $\mathbf{r} = (r_1,\ldots,r_n) \in (\mathbb{R}^+)^n$ a vector of the distribution parameters.

We can view $\Gamma_{\mathbf{r}}$ (and Γ_r) as a distribution on H through the isomorphism with $\mathbb{R}^n$. Namely, $\Gamma_{\mathbf{r}}$ produces samples of the form

$$\mathbf{B}\cdot(x_1,\ldots,x_n)^\intercal,$$

where $x_i \leftarrow \Gamma_{r_i}$. For a lattice $\mathcal{L}\subset H$ and positive real $\varepsilon > 0$, the *smoothing parameter* $\eta_\varepsilon(\mathcal{L})$ is defined to be the smallest r such that $\Gamma_{1/r}(\mathcal{L}^*\backslash\{\mathbf{0}\}) \leqslant \varepsilon$.

The distribution $\Gamma_{\mathbf{r}}$ induces a distribution $\Psi_{\mathbf{r}}$ on $K\otimes\mathbb{R}$ through the inverse of the canonical embedding, in other words

$$\mathbf{\Sigma}^{-1}\cdot\mathbf{B}\cdot(x_1,\ldots,x_n)^\intercal$$

is equal to the coordinates of $(x_1,\ldots,x_n) \leftarrow \Gamma_{\mathbf{r}}$ with respect to the power basis $1, X, X^2, \ldots, X^{n-1}$ of K over $\mathbb{Q}$.

Note that for a 2-power cyclotomic field, the matrix product $\mathbf{\Sigma}^{-1}\cdot\mathbf{B}$ is an orthogonal matrix scaled by n^{-1}. Hence, $\Psi_{\mathbf{r}}$ is a coefficient-wise scaled version of $\Gamma_{\mathbf{r}}$ over $\mathbb{R}^n$.

However, for general cyclotomics one should deal with $\mathbf{\Sigma}^{-1}$, which makes the sampling less efficient. As shown in [48], this problem can be alleviated by sampling in the ring $\mathbb{Q}[X]/\langle\Theta_m\rangle$, where $\Theta_m(X) = X^m - 1$ if m is odd, and $X^{m/2}+1$ if m is even (note that Φ_m divides Θ_m). In particular, the sample coefficients are drawn independently from Γ_r over $\mathbb{R}$. To end up in K, this polynomial sample is reduced modulo Φ_m.

In practice, the distinction between K and $K\otimes\mathbb{R}$ is ignored as long as the latter is approximated by the former with some finite but sufficiently high precision. Hence, the $\Gamma_{\mathbf{r}}$ samples live over $\mathbb{Q}$ rather than $\mathbb{R}$, so that an element sampled from $\Psi_{\mathbf{r}}$ can be truly seen as an element of the field K after pulling back along the canonical embedding.

In Chapter 3, we will need to draw elements from $\mathcal{I}$ for some fixed fractional ideal $\mathcal{I}\subset K$, where $\mathcal{I} = R$ and $\mathcal{I} = R^\vee$ are the main examples.

There are two options here. The first approach is to draw samples from the discrete Gaussian distribution $\mathcal{DG}_{\mathcal{L},r}$ defined directly over the lattice $\mathcal{L} = \sigma(\mathcal{I})$ as

$$\mathcal{DG}_{\mathcal{L},r}(\mathbf{x}) = \frac{\Gamma_r(\mathbf{x})}{\Gamma_r(\mathcal{L})}, \quad \forall\mathbf{x}\in\mathcal{L}.$$

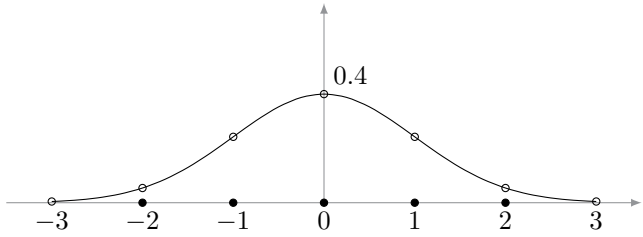


Figure 2.2: The probability density functions of the discrete Gaussian distribution $\mathcal{DG}_{\mathbb{Z},\sqrt{2\pi}}$ with standard deviation $\sigma = 1$ (white circles) and the continuous Gaussian distribution $\Gamma_{\sqrt{2\pi}}$ with the same standard deviation (solid line).

The difference between $\mathcal{DG}_{\mathcal{L},r}$ and Γ_r is illustrated in Figure 2.2. In general, sampling from $\mathcal{DG}_{\mathcal{L},r}$ is at least as hard as standard lattice problems (see in Section 3.2).

In the second approach, we round the Gaussian distribution $\Gamma_{\mathbf{r}}$ to the ideal lattice $\sigma(\mathcal{I})$. This rounding can be efficiently implemented by rounding coordinates with respect to a given $\mathbb{Z}$-module basis of $\sigma(\mathcal{I})$ as can be seen in [90, 92]. The resulting distribution is denoted by $\lfloor \Psi_{\mathbf{r}} \rceil$.

2.7 Scaled canonical Gaussian distribution

As mentioned in Section 2.5, it holds that $R = \theta R^\vee$ with $\theta = f'(\alpha)$ if f is monogenic. More generally, the former equality holds if and only if R is a complete intersection, or in the form $\mathbb{Z}[X_1, \ldots, X_n]/(f_1, \ldots, f_n)$. In this case, $\theta = \left|(\partial f_i / \partial X_j)_{i,j}\right|$.

Every sample $x \in R^\vee$ drawn from $\Psi_{\mathbf{r}}$ over the ideal lattice $\sigma(R^\vee)$ can be mapped to $\theta x \in R$. The element θ is referred to as the *tweaking factor* [43]. With respect to the power basis of K, the element θx has the following form

$$\mathbf{A}_\theta \cdot \mathbf{\Sigma}^{-1} \cdot \mathbf{B} \cdot (x_1, \ldots, x_n)^\intercal , \tag{2.1}$$

where $\mathbf{A}_\theta$ is the matrix of multiplication by θ. The resulting *canonical Gaussian* distribution on R, denoted by $\theta \cdot \Psi_{\mathbf{r}}$, is a scaling of $\Psi_{\mathbf{r}}$ by $\sigma_i(\theta)$ in the ith coordinate of the canonical embedding. Therefore, if $\Psi_{\mathbf{r}}$ is a spherical Gaussian, this scaling by θ may transform it to an elliptical Gaussian. On average the factor $\mathbf{A}_\theta$ expands $x \leftarrow \Psi_{\mathbf{r}}$, because $|\det \mathbf{A}_\theta| = \Delta$ where $\Delta = |\Delta_K|$ is the

absolute value of the discriminant of K; see [56]. In particular, $\Delta = |\mathrm{disc} f(X)|$ if K is monogenic.

To sample more efficiently, one may wish to replace the matrix $\mathbf{A}_\theta$ by a real scalar, say λ, such that (2.1) turns into

$$\lambda \cdot \mathbf{\Sigma}^{-1} \cdot \mathbf{B} \cdot (x_1, \ldots, x_n)^\intercal.$$

This expression defines the *scaled canonical Gaussian* (SCG) distribution on R, denoted by $\lambda \cdot \Psi_{\mathbf{r}}$. As shown in Section 2.5, we have $\theta = n$ in 2-power cyclotomic fields. Hence, if $\lambda = n$, the SCG distribution coincides with the canonical Gaussian distribution on R.

2.8 Canonical norm of random polynomials

To construct FHE schemes we need to generate random polynomials in the cyclotomic number field $K = \mathbb{Q}[X]/\langle X^n + 1\rangle$ where each coefficient is sampled independently from the following zero-mean distributions:

- a discrete Gaussian distribution $\mathcal{DG}_{\mathbb{Z},\sigma\sqrt{2\pi}}$ with standard deviation σ,
- the uniform distribution $\mathcal{U}_3$ over the ternary set $\{-1, 0, 1\}$,
- the uniform distribution $\mathcal{U}_q$ over $\mathbb{Z}_q$,
- the uniform distribution $\mathcal{U}_{\mathrm{rnd}}$ over the interval $(-1/2, 1/2]$.

To analyse the noise growth of FHE schemes, we heuristically estimate the canonical norm of these polynomials. This approach yields better estimations of encryption parameters as illustrated in [64, A.5].

We use the following fact: given two independent random variables drawn from zero-mean distributions with variances V_1 and V_2, the variance of their product is equal to $V_1 V_2$ and the variance of their sum is equal to $V_1 + V_2$.

We start by computing the variance of $\|a\|^{\mathrm{can}}$ for a random polynomial $a \in K$. For brevity, let $\zeta = \zeta_{2n}$. By the definition of the canonical norm, it is sufficient to compute the variance of any $a(\zeta^i)$. Let σ^2 be the variance of each coefficient of a in the power basis. The evaluation $a(\zeta^i)$ is the inner product between the coefficient vector of $a = (a_0, \ldots, a_n)$ and the fixed vector $\left(1, \zeta^i, \ldots, \zeta^{i(n-1)}\right)$, which has Euclidean norm $\sqrt{n}$. Hence, the random variable $a(\zeta^i)$ has variance $V = \sigma^2 n$.

Given this equality, we compute the variance of $a(\zeta^i)$ for the four coefficient distributions above:

$$V_{\mathcal{DG}} = \sigma^2 n, \quad a_i \leftarrow \mathcal{DG}_{\mathbb{Z},\sigma\sqrt{2\pi}},$$

$$V_3 = \tfrac{2}{3}n, \quad a_i \leftarrow \mathcal{U}_3,$$

$$V_q \simeq \tfrac{q^2}{12}n, \quad a_i \leftarrow \mathcal{U}_q,$$

$$V_{\text{rnd}} = \tfrac{1}{12}n, \quad a_i \leftarrow \mathcal{U}_{\text{rnd}}.$$

Since $a(\zeta^i)$ is the sum of independent and identically distributed variables, by the central limit theorem it is distributed similarly to a Gaussian random variable of variance V. Hence, given that $\text{erfc}(6) \simeq 2^{-55}$, we can use $6\sqrt{V}$ as a high-probability bound on $\left|a(\zeta^i)\right|$. Since in practice $n \geqslant 2^{12}$, this bound is good enough to claim that $\|a\|^{\text{can}} \leqslant 6\sqrt{V}$ with very high probability. For the above distributions, the canonical norm of a is thus bounded as follows

$$\|a\|^{\text{can}} \leqslant 6\sigma\sqrt{n}, \quad a_i \leftarrow \mathcal{DG}_{\mathbb{Z},\sigma\sqrt{2\pi}},$$

$$\|a\|^{\text{can}} \leqslant 2\sqrt{6n}, \quad a_i \leftarrow \mathcal{U}_3,$$

$$\|a\|^{\text{can}} \leqslant q\sqrt{3n}, \quad a_i \leftarrow \mathcal{U}_q,$$

$$\|a\|^{\text{can}} \leqslant \sqrt{3n}, \quad a_i \leftarrow \mathcal{U}_{\text{rnd}}.$$

In some cases we need to bound the canonical norm of a product of two or three random polynomials.

Let $a, b \in R$ be two random polynomials, whose coefficients are sampled independently from two distributions with variances V_a and V_b, respectively. The product $g = ab \mod (X^n + 1)$ has the following coefficients for any $k \in [0, n-1]$

$$g_k = \sum_{i=0}^{k} a_i b_{k-i} - \sum_{i=k+1}^{n-1} a_i b_{k+n-i}.$$

The variance of each coefficient g_k is equal to nV_aV_b. Hence, the random variable $g(\zeta^i)$ has variance $n^2V_aV_b$ and therefore

$$\|ab\|^{\text{can}} \leqslant 6n\sqrt{V_aV_b}.$$

2.9 Chinese Remainder Theorem

Theorem 2.9.1 (The Chinese Remainder Theorem (CRT))**.** *Let $\mathcal{I}_1, \ldots, \mathcal{I}_k$ be k integral and pairwise co-prime ideals of R. Let $\mathcal{I}$ be the product of $\mathcal{I}_1, \ldots, \mathcal{I}_k$, then the following ring homomorphism holds*

$$\mathsf{CRT} : R/\mathcal{I} \rightarrow R/\mathcal{I}_1 \times \ldots \times R/\mathcal{I}_k$$

This isomorphism can be inverted in polynomial time by computing a so-called "CRT basis", i.e. elements $b_1, \ldots, b_k \in R$ such that $b_i \equiv 1 \mod \mathcal{I}_i$ and $b_i \equiv 0 \mod \mathcal{I}_j$ if $i \neq j$. For any given $\mathbf{a} = (a_1, \ldots, a_k) \in \prod_{i=1}^{k} R/\mathcal{I}_i$, the element $a = \sum_{i=1}^{k} a_i b_i \mod \mathcal{I}$ is the unique pre-image of a in $R/\mathcal{I}$ under CRT.

The application of the CRT isomorphism is twofold.

On the one hand, since $N(\mathcal{I}) = \prod_{i=1}^{k} N(\mathcal{I}_i)$, the quotient ring $R/\mathcal{I}$ can be viewed as a direct product of smaller quotient rings. Therefore, computations in $R/\mathcal{I}$ are translated to independent computations in these smaller rings, which can be performed more efficiently and in parallel.

On the other hand, we can encode information into each of the rings $\{R/\mathcal{I}_i\}_i$ and then transform these encodings to a single element of $R/\mathcal{I}$ via CRT^{-1}. This transformation is called *packing*.

Chapter 3

Hard lattice problems

In general, all public-key encryption schemes base their security on hard computational problems. Namely, given the public information produced by the encryption scheme (i.e. parameters, ciphertexts, public keys, evaluation keys etc.), the adversary needs to solve an instance of a highly non-trivial mathematical problem in order to derive information about the plaintext or the secret key. This mathematical problem is called *hard* if no algorithm is known that solves the problem in time polynomial in the input size. The number of operations needed to solve this mathematical problem defines the *security level* of the related encryption scheme.

A common approach to prove the hardness of a problem A consists in showing that an algorithm solving A also solves another problem B which is believed to be hard. In this case, we say that the problem B *reduces* to the problem A.

As explained in Section 1.2, the most efficient FHE schemes rely on computational problems over lattices: LWE and RLWE. In this chapter, we give an overview of these problems and discuss their hardness.

3.1 Shortest Vector Problems

We start with the Shortest Independent Vectors problem (SIVP_γ), which is defined as follows.

Definition 3.1.1 (SIVP_γ)**.** *For an approximation factor $\gamma = \gamma(n) \geqslant 1$ and a basis $\mathbf{B}$ of a lattice $\mathcal{L}$, output n linearly independent lattice vectors of length at most $\gamma(n) \cdot \lambda_n(\mathcal{L})$.*

For constant factor γ, it was proven that SIVP_γ is NP-hard [16], i.e. a right solution can be found in exponential time and verified in polynomial time. However, if γ is exponential in the lattice dimension n, SIVP_γ can be efficiently solved by the LLL algorithm [84]. The state-of-the-art polynomial-time algorithms [1] break SIVP_γ only for approximation factors $\gamma \in O\left(2^{n \log\log n / \log n}\right)$.

Another important lattice problem for FHE is the decision Shortest Vector problem (GapSVP_γ).

Definition 3.1.2 (GapSVP_γ)**.** *For an approximation factor $\gamma = \gamma(n) \geqslant 1$, a basis $\mathbf{B}$ of a lattice $\mathcal{L}$ and a real number $d > 0$, return*

- *YES, if $\lambda_1(\mathcal{L}) \leqslant d$*
- *NO, if $\lambda_1(\mathcal{L}) > \gamma d$*

Note that if an instance $(\mathbf{B}, d)$ satisfies neither of the two conditions, then the behaviour of the GapSVP_γ algorithm is unspecified.

The reduction from $\mathsf{GapSVP}_{n\gamma}$ to SIVP_γ is established through Banaszczyk's transference theorem. Given the basis $\mathbf{B}$ and $d \in \mathbb{R}$, we compute the dual basis $\mathbf{B}^*$ and feed it to the SIVP_γ oracle. The SIVP_γ oracle outputs linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ of length ℓ in the interval $[\lambda_n(\mathcal{L}^*), \gamma \cdot \lambda_n(\mathcal{L}^*)]$. This implies

$$1/\ell \leqslant 1/\lambda_n(\mathcal{L}^*) \leqslant \gamma/\ell.$$

Due to Banaszczyk's transference theorem, $1 \leqslant \lambda_1(\mathcal{L})\,\lambda_n(\mathcal{L}^*) \leqslant n$, therefore $\lambda_1(\mathcal{L}) \in [1/\ell, n\gamma/\ell]$.

If $d < 1/\ell$, then $\lambda_1(\mathcal{L}) > d$, which implies $\lambda_1(\mathcal{L}) > n\gamma d$ and the "NO" outcome of $\mathsf{GapSVP}_{n\gamma}$. If $d \geqslant 1/\ell$, then $\lambda_1(\mathcal{L}) \leqslant \gamma n d$, which implies $\lambda_1(\mathcal{L}) \leqslant d$ and the "YES" outcome of $\mathsf{GapSVP}_{n\gamma}$.

Importantly, these problems with small enough approximation factors remain hard in any model of quantum computation, where quantum-mechanical phenomena are used to perform computations. As a result, SIVP_γ and GapSVP_γ constitute a reliable basis for post-quantum cryptographic primitives.

3.2 Discrete Gaussian Sampling problem

The Discrete Gaussian Sampling problem (DGS_γ) is defined as follows.

Definition 3.2.1 (DGS_γ)**.** *For a function γ that maps lattices to non-negative real numbers, given a lattice $\mathcal{L}$ and a parameter $r \geqslant \gamma(\mathcal{L})$, output an independent sample from a distribution that is within negligible statistical distance from $\mathcal{DG}_{\mathcal{L},r}$.*

For any $\gamma \leqslant 1$, the reduction from $\mathsf{GapSVP}_{100\sqrt{n}\gamma}$ to $\mathsf{DGS}_{\sqrt{n}\gamma/\lambda_1(\mathcal{L}*)}$ is shown by Regev [105].

The special case of this problem, called K-DGS_γ, deals only with ideal lattices in K. As shown in [90], the K-DGS_γ problem, where $\gamma = \eta_\varepsilon(\mathcal{L})$ for some negligible $\varepsilon > 0$, is at least as hard as SIVP_ξ for $\xi \in O(\sqrt{n \log n})$.

3.3 Learning with Errors

In 2005 Regev introduced the Learning with Errors (LWE) problem [105], which relies on the following distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

Definition 3.3.1 (LWE distribution)**.** *Let n, q be positive integers, χ_e be a probability distribution on $\mathbb{Z}$ and $\mathbf{s}$ be a secret vector in $\mathbb{Z}_q^n$. Sample $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$ and $e \leftarrow \chi_e$. Let $\mathsf{LWE}_{\mathbf{s},\chi_e}$ be the probability distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by returning*

$$(\mathbf{a}, b) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e \mod q) \in \mathbb{Z}_q^n \times \mathbb{Z}_q.$$

Given the LWE distribution we can define two variants of the LWE problem.

Definition 3.3.2 (Search LWE)**.** *Search-LWE (S-LWE) is the problem to recover $\mathbf{s}$ from $(\mathbf{a}, b) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e \bmod q) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ sampled according to $\mathsf{LWE}_{\mathbf{s},\chi_e}$.*

Definition 3.3.3 (Decision LWE)**.** *Decision-LWE (D-LWE) is the problem to decide whether pairs $(\mathbf{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ are sampled according to $\mathsf{LWE}_{\mathbf{s},\chi_e}$ or the uniform distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$.*

These problems are equivalent for any modulus q as was shown in [25].

Hardness. The hardness of S-LWE with $\chi_e = \mathcal{DG}_{\mathbb{Z},r>2\sqrt{n}}$ was proven by Regev using a quantum polynomial-time reduction from DGS_γ to LWE [105]. In 2009, Peikert [98] showed that D-LWE with exponential modulus is at least as hard as the GapSVP_γ problem via a classical polynomial-time reduction, which was improved for a polynomial q by Brakerski et al. [25]. Unfortunately, the latter reduction results in a quadratic loss in the lattice dimension.

Due to the connection to hard lattice problems and functional flexibility, LWE became an extremely fruitful basis of various cryptographic primitives, including FHE schemes [26, 22, 24].

To perform more homomorphic operations, FHE schemes often use error distributions with low width parameters, which are not supported by the aforementioned security reductions. This increases the risk of successful attacks which are unrelated to standard lattice problems. A particular example of such attacks is the algorithm of Arora and Ge [10] that solves LWE with the distribution width $r = n^{\varepsilon} < \sqrt{n}$. The total time and space complexity of this algorithm $2^{\tilde{O}(n^{\varepsilon})}$, which is sub-exponential if $\varepsilon < 1/2$.

However, to compute the concrete security level of LWE with given parameters, LWE samples are transformed into an instance of a standard lattice problem. Given m LWE instances generated by the same secret vector $\mathbf{s}$, we can represent them in matrix notation as

$$(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \mod q) ,$$

where $\mathbf{A}$ is an $m \times n$ matrix and $\mathbf{b}, \mathbf{e} \in \mathbb{Z}_q^m$.

From this representation, the S-LWE problem can be viewed as an average-case *Bounded Distance Decoding* (BDD) problem on a certain family of q-ary lattices of dimension m. Namely, given the following lattice

$$\mathcal{L}(\mathbf{A}) = \{\mathbf{A}\mathbf{x} : \mathbf{x} \in \mathbb{Z}_q^n\} + q\mathbb{Z}^m$$

and the vector $\mathbf{b}$, which is close to exactly one vector $\mathbf{v}$ of $\mathcal{L}(\mathbf{A})$, return $\mathbf{v}$. This task can be also reformulated in terms of the *Unique Shortest Vector* uSVP problem. Given the lattice $\mathcal{L} = \{[\mathbf{b}| - \mathbf{A}]\mathbf{x} : \mathbf{x} \in \mathbb{Z}_q^{n+1}\} + q\mathbb{Z}^m$, find a unique shortest vector $\mathbf{e}$, which is guaranteed to be small by the definition of $\mathsf{LWE}_{\mathbf{s},\chi_e}$.

The D-LWE can be transformed into an instance of the *Shortest Integer Solution* (SIS) problem. Given $t < q$, return $\|\mathbf{y}\| \leqslant t$ such that

$$\mathbf{y}^{\mathsf{T}}\mathbf{A} \equiv \mathbf{0} \mod q.$$

If a tuple $(\mathbf{A}, \mathbf{b})$ is uniformly random, so is $\langle \mathbf{y}, \mathbf{b} \rangle$. However, if $(\mathbf{A}, \mathbf{b})$ is drawn from $\mathsf{LWE}_{\mathbf{s},\chi_e}$ then $\mathbf{y}$ satisfies

$$\langle \mathbf{y}, \mathbf{b} \rangle = \langle \mathbf{y}^{\mathsf{T}}\mathbf{A}, \mathbf{s} \rangle + \langle \mathbf{y}, \mathbf{e} \rangle \equiv \langle \mathbf{y}, \mathbf{e} \rangle \mod q.$$

Since $\mathbf{e}$ is short, $\langle \mathbf{y}, \mathbf{e} \rangle$ is small.

To solve the above problems, many algorithms rely on lattice reduction methods. The analysis of lattice reduction algorithms is a vast topic, which is beyond

the scope of this thesis, but we mention that most algorithms exploit LLL [84], BKZ [109], BKZ 2.0 [33] and BKW [17] along with enumeration, sieving and Voronoi cell computation methods. There is no consensus that a certain lattice attack on LWE outperforms the others. This is why it is necessary to test several lattice algorithms to get a concrete security estimation, which can be done via specialised tools [7].

3.4 Learning with Errors over Rings

The Learning with Errors over Rings (RLWE) problem is an adaptation of the LWE problem over algebraic rings.

Let K be a degree n number field with ring of integers R.

There are two interpretations of the RLWE problem: dual and non-dual. Depending on the interpretation, choose a fractional ideal $\mathcal{I} \subset K$. In the *dual* RLWE problems take $\mathcal{I} = R^\vee$, while in the *non-dual* RLWE problems set $\mathcal{I} = R$.

Note that $\mathcal{I} \otimes \mathbb{R} = K \otimes \mathbb{R}$, so the distribution $\Psi_{\mathbf{r}}$ on $K \otimes \mathbb{R}$ can be viewed as a distribution on $\mathcal{I} \otimes \mathbb{R}$. Let $\mathcal{I}_q$ denote $\mathcal{I}/q\mathcal{I}$ and write $\mathcal{I}_{q,\mathbb{R}}$ for the torus $(\mathcal{I} \otimes \mathbb{R})/q\mathcal{I}$. We define the following distribution, which resembles $\mathsf{LWE}_{\mathbf{s},\chi_e}$.

Definition 3.4.1 (RLWE distribution)**.** *For $s \in \mathcal{I}_q$ and $\mathbf{r} \in (\mathbb{R}^+)^n$, a sample from the RLWE distribution $\mathsf{RLWE}_{s,\mathbf{r}}$ over $R_q \times \mathcal{I}_{q,\mathbb{R}}$ is generated by choosing $a \xleftarrow{\$} R_q, e \leftarrow \Psi_{\mathbf{r}}$ and returning $(a, b = as + e \mod q\mathcal{I})$.*

Accordingly, we define the search and decision versions of RLWE.

Definition 3.4.2 (Search RLWE)**.** *For a random but fixed choice of $s \xleftarrow{\$} \mathcal{I}_q$, the* search *RLWE problem (S-RLWE) is to recover s with non-negligible probability from arbitrarily many independent samples from $\mathsf{RLWE}_{s,\mathbf{r}}$.*

Definition 3.4.3 (Decision RLWE)**.** *The* decision *RLWE problem (D-RLWE) is to distinguish, for a random but fixed choice of $s \xleftarrow{\$} \mathcal{I}_q$, with non-negligible advantage between arbitrarily many independent samples from $\mathsf{RLWE}_{s,\mathbf{r}}$ and the same number of uniformly random samples from $R_q \times \mathcal{I}_{q,\mathbb{R}}$.*

The reduction from D-RLWE to S-RLWE is obvious. If K is an mth cyclotomic number field and $q \equiv 1 \mod m$, then there exists the reduction from S-RLWE to D-RLWE [90, 89], which increases the error distribution width by a factor of about $n^{1/4}$.

Hardness. To serve as a cryptographic primitive, $\mathsf{RLWE}_{s,\mathbf{r}}$ must generate pseudo-random samples. In other words, D-RLWE must be hard.

The hardness reduction of D-RLWE was already provided in the paper of Lyubashevsky et al. [90], which introduced RLWE. This result boils down to the sequence of two reductions. The first is a quantum reduction from the worst-case approximate Shortest Independent Vectors Problem (SIVP_γ) on ideal lattices in a given ring to S-RLWE over that same ring. The second is a classical reduction from S-RLWE to D-RLWE, which works only for cyclotomic number fields of order m and a prime modulus q such that $q \equiv 1 \mod m$. The last condition was eliminated in [83].

For any number field and any modulus q, the hardness of D-RLWE was shown by Peikert et al. [100]. Actually, this result deals with a generalisation of RLWE defined over the following family of distributions Υ_α. Fix an arbitrary $g(n) = \omega(\log n)$. For $\alpha > 0$, a distribution sampled from Υ_α is an elliptical Gaussian $\Gamma_{\mathbf{r}}$, where $\mathbf{r}$ is sampled as follows: for $i = [s_1]$, sample $x_i \leftarrow \Gamma_1$ and set $r_i^2 = \alpha^2(x^2 + g^2(n))/2$. For $i = [s_1 + 1, s_1 + s_2]$, sample $x_i, y_i \leftarrow \Gamma_{1/\sqrt{2}}$ and set $r_i^2 = r_{i+s_2}^2 = \alpha^2(x_i^2 + y_i^2 + g^2(n))/2$. According to [100], the D-RLWE hardness is stated as follows.

Theorem 3.4.1. *Let K be an arbitrary number field of degree n and R be its ring of integers. Let $\alpha = \alpha(n) \in (0,1)$, and let $q = q(n) \geqslant 2$ be an integer such that $\alpha q \geqslant 2 \cdot \omega(1)$. There is a polynomial-time quantum reduction from K-DGS$_\gamma$ to (average-case, decision) dual $\mathsf{RLWE}_{q,\Upsilon_\alpha}$ for any*

$$\gamma = \max\{\eta_\varepsilon(\mathcal{I}) \cdot \sqrt{2}/\alpha \cdot \omega(1), \sqrt{2n}/\lambda_1(\mathcal{I}^\vee)\}\,.$$

Note that this theorem holds for the dual RLWE problem. However, in practice, it is more convenient to work with the non-dual RLWE as only polynomials with integer coefficients are sampled. Note that it is easy to transform a dual RLWE sample to a non-dual one. From Section 2.5, we know that $R^\vee = R/\theta$, if K is monogenic. Hence, given a sample (a, b) from the dual distribution $\mathsf{RLWE}_{s,\mathbf{r}}$, we can re-write its second term in the following form

$$\frac{b'}{\theta} = a\frac{s'}{\theta} + e \mod qR^\vee\,,$$

where $s', b' \in R_q$ and e is sampled from $\Psi_{\mathbf{r}}$ over $R^\vee_{q,\mathbb{R}}$. Then, multiplication by the tweaking factor θ results in

$$b' = as' + \theta e \mod qR\,, \tag{3.1}$$

which is an instance of the non-dual distribution $\mathsf{RLWE}_{s',\mathbf{r}'}$ with width parameter vector $\mathbf{r}' = \sigma(\theta) \circ \mathbf{r}$. As a result, the two forms of RLWE distributions are equivalent up to the choice of the error distribution.

The existing attacks transform RLWE samples to LWE ones and attack those instances. This transformation is done as follows. Given a RLWE sample (a, b), fix a $\mathbb{Z}$-basis of $\mathcal{I}$, which is also a $\mathbb{Z}_q$-basis of $\mathcal{I}_q$, and rewrite (a, b) with relation to this basis as

$$\begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} = \mathbf{A}_a \cdot \begin{pmatrix} s_1 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ \vdots \\ e_n \end{pmatrix},$$

where $\mathbf{A}_a$ is the matrix of multiplication by a. Each b_i can be seen as an LWE sample since $b_i = \langle \mathbf{A}_a[i], \mathbf{s} \rangle + e_i$ where $\mathbf{A}_a[i]$ is the ith row of matrix $\mathbf{A}_a$. Nevertheless, in general e_i's and the rows of $\mathbf{A}_a$ preserve the algebraic structure of $\mathcal{I}_q$. As a result, the b_i's are not independent LWE samples. Despite considerable efforts, no algorithm has been discovered that exploits the algebraic structure of RLWE samples.

Optimisations for FHE. Starting with the BGV scheme [24], the non-dual RLWE problem became a basis of FHE schemes.

The main advantage of RLWE over LWE boils down to compressing n LWE samples into one RLWE sample. Namely, instead of an LWE sample $(\mathbf{A}, \mathbf{b})$, which demands $(n^2 + n) \log q$ bits of memory, we can use a RLWE sample (a, b) of size $2n \log q$.

However, it is highly desirable to go further and observe what other optimisations can be performed for RLWE instances.

As we will show in Chapter 4, the size of the secret s directly affects the noise growth in RLWE-based FHE schemes. Therefore, it is desirable to take s as small as possible, for example, by sampling it from the discrete Gaussian distribution or drawing its coefficients from the ternary set $\{-1, 0, 1\}$. This trick is partially justified by Applebaum et al. [9], who proved that LWE remains secure even if s is drawn from the error distribution χ_e at the loss of n samples. For binary secrets, it was shown in [25] that LWE in dimension $n \log q$ is as hard as LWE in dimension n with uniformly random secrets. These results demonstrate that RLWE becomes less secure when one switches to small secrets without changing other parameters of the problem. This security loss was heuristically observed in attacks [3, 6] on the parameter sets of popular FHE libraries such as HElib [70], SEAL [111] and HEAAN [71], where binary secret keys are used.

Another type of optimisations revolves around simplifications of the error distribution $\Psi_{\mathbf{r}}$. As shown in Section 2.6, if K is a general cyclotomic field, then sampling from $\Psi_{\mathbf{r}}$ can be done via coefficient-wise Gaussian sampling in $\mathbb{Q}[X]/\langle \Theta_m \rangle$ followed by reduction modulo Φ_m. For general rings, $\Psi_{\mathbf{r}}$ can be

replaced by the SCG-LWE distribution, which results in a new computational problem.

3.5 Scaled Canonical Gaussian-LWE

In this section, we discuss another variant of the LWE problem defined over algebraic rings. This problem can be easily confused with non-dual RLWE.

As discussed in the previous section, a dual RLWE instance with error distribution $\Psi_{\mathbf{r}}$ can be efficiently converted to a non-dual RLWE instance with error distribution $\theta \cdot \Psi_{\mathbf{r}}$, which we called the canonical Gaussian distribution in Section 2.7. The canonical Gaussian distribution is a simple scaling of $\Psi_{\mathbf{r}}$ in 2-power cyclotomic fields as shown in Section 2.5.

However, in general, θ introduces a non-trivial transformation of error coordinates rewritten with respect to a $\mathbb{Z}$-basis of R according to Equation (3.1)

$$(b_1, \ldots, b_n)^\intercal = \mathbf{A}_a \cdot (s_1, \ldots, s_n)^\intercal + \mathbf{A}_\theta \cdot \mathbf{\Sigma}^{-1} \cdot \mathbf{B} \cdot (e_1, \ldots, e_n)^\intercal$$

where $\mathbf{A}_a$ and $\mathbf{A}_\theta$ are the matrices of multiplication by a and θ, respectively.

To avoid multiplication by $\mathbf{A}_\theta$, we can replace it by a real constant λ and thus obtain

$$(b_1, \ldots, b_n)^\intercal = \mathbf{A}_a \cdot (s_1, \ldots, s_n)^\intercal + \lambda \cdot \mathbf{\Sigma}^{-1} \cdot \mathbf{B} \cdot (e_1, \ldots, e_n)^\intercal.$$

As a result, the error distribution turns into the SCG distribution $\lambda \cdot \Psi_{\mathbf{r}}$ and the resulting sample (a, b) belongs to the SCG-LWE distribution defined below.

Definition 3.5.1 (SCG-LWE distribution)**.** *For $s \in R_q$ and $\mathbf{r} \in (\mathbb{R}^+)^n$, a sample from the SCG-LWE distribution SCG-LWE$_{s,\lambda,\mathbf{r}}$ over $R_q \times R_{q,\mathbb{R}}$ is generated by choosing $a \xleftarrow{\$} R_q, e \leftarrow \lambda \cdot \Psi_{\mathbf{r}}$ and returning $(a, b = a \cdot s + e \mod qR)$.*

As for RLWE, we define the search and the decision SCG-LWE problems.

Definition 3.5.2 (Search SCG-LWE)**.** *For a random but fixed choice of $s \xleftarrow{\$} R_q$, the* search *SCG-LWE problem is to recover s with non-negligible probability from arbitrarily many independent samples from SCG-LWE$_{s,\lambda,\mathbf{r}}$.*

Definition 3.5.3 (Decision SCG-LWE)**.** *The* decision *SCG-LWE problem is to distinguish, for a random but fixed choice of $s \xleftarrow{\$} R_q$, with non-negligible advantage between arbitrarily many independent samples from SCG-LWE$_{s,\lambda,\mathbf{r}}$ and the same number of uniformly random samples from $R_q \times R_{q,\mathbb{R}}$.*

Hardness. As shown in Section 2.7, the SCG distribution with $\lambda = n$ over a 2-power cyclotomic field coincides with the canonical Gaussian distribution $\theta \cdot \Psi_{\mathbf{r}}$. As a result, the SCG-LWE$_{s,\lambda,\mathbf{r}}$ distribution and the non-dual form of RLWE$_{s,\mathbf{r}}$ are essentially the same, which implies the equivalence of the related search and decision problems. Hence, the security reductions from the standard lattice problems to RLWE from Section 3.4 are also valid for SCG-LWE in 2-power cyclotomic fields. More generally, both problems become equivalent if the matrix $\mathbf{A}_\theta$ is a scaled orthongonal matrix, i.e. $\theta \in \mathbb{Z}$, and $\lambda = \theta$.

However, in general replacing the tweaking factor θ by a small scalar λ results in an SCG error distribution which may be not "well-spread" across a $\mathbb{Z}$-basis of R. In particular, this geometrical distortion of a narrow enough error distribution may lead to an error distribution with extremely small width along certain vectors of the $\mathbb{Z}$-basis of R such that the rounding to the ideal lattice $\sigma(R)$ produces zeros in the corresponding coordinates with very high probability. Moreover, the same situation might happen in RLWE samples if the error distribution is scaled down by a big enough factor. This observation was exploited in efficient attacks on both the search and the decision variants of SCG-LWE [51, 53, 31, 32, 99]. These insecure SCG-LWE and RLWE instances are also the subject of this work. More details can be found in Chapters 6 and 7.

As a countermeasure for the aforementioned attacks, the parameter λ can be taken big enough in order to compensate the skewness of the SCG distribution. We discuss in Chapter 7 that the most natural choice is $\lambda = |\Delta|^{1/n}$ where Δ is the discriminant of K. In this case, it is still an open question what the security level of the SCG-LWE problems is and whether there exist a security reduction from standard lattice problems to SCG-LWE.

Chapter 4

RLWE-based SHE schemes

This chapter is dedicated to RLWE-based SHE schemes of the "golden" 2+ generation such as `BGV`, `FV` and `HEAAN`. As mentioned in Section 1.2, these schemes are considered the most efficient for practical purposes as they are able to compute arithmetic circuits with polylogarithmic computational overhead [63]. We describe a general framework for constructing such schemes and also elaborate the `FV` scheme.

We note that the FHE schemes of the third generation have RLWE-based instantiations [65, 77]. However, these schemes are not compatible with the performance optimisations that make the "golden" generation so efficient. Therefore, the third-generation schemes have higher computational and memory overheads. In this chapter, we omit these schemes from consideration.

4.1 General framework

RLWE-based SHE schemes follow a similar outline both in the symmetric and the public-key modes. Without loss of generality, we describe here the public-key mode.

4.1.1 Basic encryption scheme

Parameters. Let q and n be integers chosen according to a given security level. The ciphertext space is the ring R_q^2 where R is a cyclotomic ring of integers of

dimension n. The plaintext space is a ring R_t with $t \leqslant q$. The size of plaintexts is bounded by the plaintext modulus t such that $|\mathtt{pt}|_\infty \leqslant t/2$. Hence, any plaintext $\mathtt{pt}$ can be trivially identified with an element of the ciphertext space $[\mathtt{pt}]_q$. In BGV and FV, t is taken much smaller than q, whereas $t = q$ in HEAAN. Let $\Delta \in \mathbb{Z}$ be the scaling parameter.

Let χ_k be the key distribution such that any sample $u \leftarrow \chi_\mathrm{k}$ is a uniformly random polynomial with ternary coefficients, i.e. $|u|_\infty \leqslant 1$. We define the error distribution χ_e as the discretised spherical distribution $\lfloor \Psi_r \rceil$ (FV, HEAAN) or $t \cdot \lfloor \Psi_r \rceil$ (BGV) over R.

Key generation. The secret key s is sampled from χ_k. It is then used to generate the public key, which is a RLWE pair

$$\mathtt{pk} = \left([-as + e]_q, a\right)$$

with $a \xleftarrow{\$} R_q$ and $e \leftarrow \chi_\mathrm{e}$. According to the hardness of D-RLWE stated in Theorem 3.4.1, this pair looks uniformly random and thus hides any information about the secret key.

Encryption. To encrypt a message $\mathtt{pt} \in R_t$, we mask it using the public key $\mathtt{pk} = (b, a)$ as follows. First, we generate a polynomial $u \leftarrow \chi_\mathrm{k}$ and two errors $e_1, e_2 \leftarrow \chi_\mathrm{e}$. Then, these polynomials are used to re-randomise the public key

$$b' = [bu + e_1]_q, \quad a' = [au + e_2]_q .$$

Again due to Theorem 3.4.1, the RLWE pairs (a', a) and (b', b) are computationally indistinguishable from uniformly random tuples in R_q^2. Hence, (b', a') is also pseudo-random, so we can use this pair as an additive mask to encrypt $\mathtt{pt}$ as

$$\mathtt{ct} = \left(\left[\Delta \cdot \mathtt{pt} + b'\right]_q, a'\right) .$$

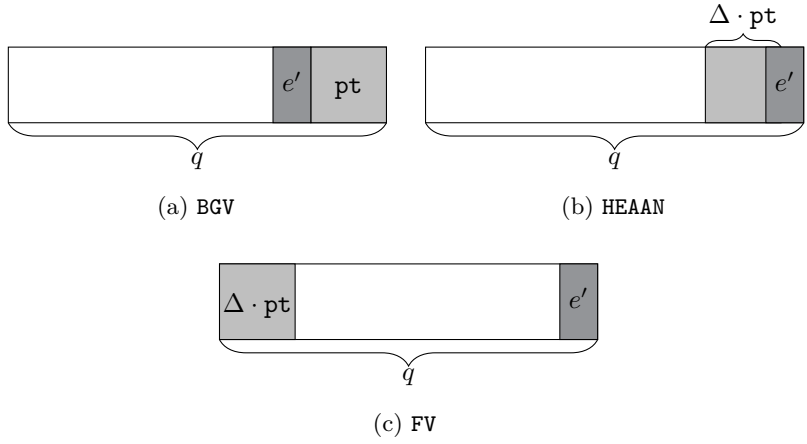
Decryption. Given a ciphertext $\mathtt{ct} = (c_0, c_1)$, we remove a part of its random mask using the secret key

$$[c_0 + c_1 s]_q = \left[\Delta \cdot \mathtt{pt} + b' + a's\right]_q \equiv \Delta \cdot \mathtt{pt} + e' \mod q, \tag{4.1}$$

where $e' \in R_q$ has the smallest possible infinity norm. The term e' is called the *inherent noise* of $\mathtt{ct}$. If $\mathtt{ct}$ is a "fresh" outcome of the encryption function, then $e' = eu + e_1 + e_2 s$.

If we aim to decrypt the exact message $\mathtt{pt}$, then we recover it from $\Delta \cdot \mathtt{pt} + e'$ assuming that $|e'|_\infty$ is small enough. If we are interested in an approximation of $\mathtt{pt}$, we can view e' as a part of the message.

Decryption can be also viewed as the evaluation of the polynomial $\mathtt{ct}(Y) = c_0 + c_1 Y$ at the secret key s modulo q. Depending on the form of the decryption output $[\mathtt{ct}(s)]_q$, the existing RLWE-based schemes can be separated into three families as depicted in Figure 4.1.



(a) BGV (b) HEAAN

(c) FV

Figure 4.1: The decryption structure $[\mathtt{ct}(s)]_q$ of RLWE-based FHE schemes.

In BGV-type schemes (Figure 4.1a), the scaling parameter Δ is simply 1 and error terms are drawn from $t \cdot \lfloor \Psi_r \rceil$ such that $e \equiv 0 \mod t$ for each $e \leftarrow \chi_{\mathrm{e}}$. As a result, the plaintext message is located in the least significant bits of $[\mathtt{ct}(s)]_q$ followed by the inherent noise e' in higher bits. To remove e', it is sufficient to cut the bits higher than the plaintext bits by reducing $[\mathtt{ct}(s)]_q$ modulo t.

In HEAAN-like schemes (Figure 4.1b), the inherent noise is placed in the least significant bits. Moreover, the noise is allowed to mix with the message bits also located in the lower bits of $[\mathtt{ct}(s)]_q$. To reduce the number of message bits covered by the noise, Δ is taken big enough. To decrypt, we scale down $[\mathtt{ct}(s)]_q$ by Δ and round.

In FV-type schemes (Figure 4.1c), the highest bits of $[\mathtt{ct}(s)]_q$ are occupied by the message using $\Delta = \lfloor q/t \rfloor$, while the lowest bits contain the inherent noise. To complete decryption, we scale down $[\mathtt{ct}(s)]_q$ by t/q, round the result and reduce it modulo t.

4.1.2 Homomorphic operations

To study how homomorphic operations work, we analyse their effect on the decryption structure.

Addition. To add two ciphertexts $\mathtt{ct}_1 = (c_0, c_1)$ and $\mathtt{ct}_2 = (d_0, d_1)$, we sum their tuples component-wise.

$$\mathtt{ct}_{\mathtt{Add}} = \left([c_0 + d_0]_q, [c_1 + d_1]_q\right).$$

Hence, the resulting decryption structure is equal to

$$\begin{aligned}\left[\mathtt{ct}_{\mathtt{Add}}(s)\right]_q &= \left[\mathtt{ct}_1(s) + \mathtt{ct}_2(s)\right]_q \\ &= \left[\Delta \cdot \mathtt{pt}_1 + e_1' + \Delta \cdot \mathtt{pt}_2 + e_2'\right]_q .\end{aligned}$$

Since $\mathtt{pt}_1 + \mathtt{pt}_2 = \left[\mathtt{pt}_1 + \mathtt{pt}_2\right]_t + gt$ where $g \in R$ and $|g|_\infty \leqslant 1$, we obtain

$$\left[\mathtt{ct}_{\mathtt{Add}}(s)\right]_q \equiv \Delta \cdot \left[\mathtt{pt}_1 + \mathtt{pt}_2\right]_t + e_1' + e_2' + \Delta g t \mod q$$

The infinity norm of the last term is bounded by $|\Delta g t|_\infty \leqslant \Delta t$. Since this bound is much smaller than q in `BGV`- and `HEAAN`-type schemes, the inherent noise is equal to $e_1' + e_2' + \Delta g t$. In `FV`-type schemes, $\Delta t = q - |q|_t$. Hence, the invariant noise is equal to $e_1' + e_2' - g\,|q|_t$ with $|g\,|q|_t|_\infty \leqslant t$. As a result, the inherent noise growth after addition is additive for all the schemes.

Multiplication. First, we look what happens when two decryption structures are multiplied:

$$\begin{aligned}\left[\mathtt{ct}_1(s) \cdot \mathtt{ct}_2(s)\right]_q &= \left[(c_0 + c_1 s)(d_0 + d_1 s)\right]_q \\ &= \left[c_0 d_0 + (c_0 d_1 + c_1 d_0) s + c_1 d_1 s^2\right]_q \\ &= \left[c_0' + c_1' s + c_2' s^2\right]_q .\end{aligned}$$

Using Equation (4.1), we can rewrite the above expression as

$$\begin{aligned}\left[\mathtt{ct}_1(s) \cdot \mathtt{ct}_2(s)\right]_q &= \left[(c_0 + c_1 s)(d_0 + d_1 s)\right]_q \\ &= \left[(\Delta \cdot \mathtt{pt}_1 + e_1')(\Delta \cdot \mathtt{pt}_2 + e_2')\right]_q \\ &= \left[\Delta^2 \cdot \mathtt{pt}_1 \cdot \mathtt{pt}_2 + \Delta(\mathtt{pt}_1 \cdot e_2' + \mathtt{pt}_2 \cdot e_1') + e_1' e_2'\right]_q \\ &\equiv \Delta^2 \cdot \mathtt{pt}_1 \cdot \mathtt{pt}_2 + e_{\mathtt{Mul}} \mod q .\end{aligned}$$

Note that the triple (c'_0, c'_1, c'_2) is an encryption of $\left[\Delta^2 \cdot \mathtt{pt}_1 \cdot \mathtt{pt}_2 + e_{\mathtt{Mul}}\right]_q$ under the key (s, s^2).

We encounter two problems.

First, there is an additional factor Δ that scales up the resulting plaintext and boosts the invariant noise. As a solution, we can scale (c'_0, c'_1, c'_2) by some constant $C \simeq \Delta^{-1}$ and round. This approach, which is usually applied in FV-type schemes, has an interesting consequence that $\mathtt{pt}_1 \cdot e'_2 + \mathtt{pt}_2 \cdot e'_1$ becomes the leading term of the invariant noise as will be shown in Section 4.2. Since the infinity norm of plaintexts is bounded by $t/2$, the noise grows linearly after multiplication.

In BGV and HEAAN, the scaling parameter Δ is small, thus resulting in the leading term $e'_1 e'_2$ of the invariant noise and, consequently, a multiplicative noise growth. To suppress the noise, these schemes resort to another method that involves switching to a smaller ciphertext modulus p. This method will be explained later in this section.

The second problem consists in the expansion of the resulting ciphertext from dimension 2 to 3. This problem can be addressed by relinearisation techniques described below.

Relinearisation. Given a ciphertext $\mathtt{ct} = (c_0, c_1, c_2) \in R_q^3$ of dimension 3 encrypting a plaintext message $\mathtt{pt}$ under the secret key (s, s^2), we wish to switch to a ciphertext $\mathtt{ct}_{\mathtt{Relin}} = (c'_0, c'_1)$ of dimension 2 that encrypts the same message under the secret key s. The decryption structure of $\mathtt{ct}$ is equal to

$$\left[\mathtt{ct}(s)\right]_q = \left[c_0 + c_1 s + c_2 s^2\right]_q . \tag{4.2}$$

Let $\mathtt{rlk}$ be a publicly known masked version of s^2 under the secret key s, namely $\mathtt{rlk} = (\left[-a_0 s + e_0 + s^2\right]_q, a_0)$ where $a_0 \xleftarrow{\$} R_q$ and $e_0 \leftarrow \chi_{\mathrm{e}}$. This element is usually called a *relinearisation key*. We assume that $\mathtt{rlk}$ leaks no information about the secret key, which requires a circular security assumption.

Note that $\left[\mathtt{rlk}[0] + \mathtt{rlk}[1] \cdot s\right]_q = s^2 + e_0$. We can thus rewrite (4.2) as

$$\begin{aligned}\left[\mathtt{ct}(s)\right]_q &= \left[c_0 + c_1 s + c_2(\mathtt{rlk}[0] + \mathtt{rlk}[1] \cdot s - e_0)\right]_q \\ &= \left[(c_0 + c_2 \cdot \mathtt{rlk}[0]) + (c_1 + c_2 \cdot \mathtt{rlk}[1])s - c_2 e_0)\right]_q .\end{aligned}$$

As a result, the ciphertext $\mathtt{ct}_{\mathtt{Relin}} = (\left[c_0 + c_2 \cdot \mathtt{rlk}[0]\right]_q, c_1 + c_2 \cdot \mathtt{rlk}[1])$ encrypts the same message as $\mathtt{ct}$ at the cost of the invariant noise increased by $c_2 e_0$. Since c_2 is a pseudo-random element in R_q, this error term is huge, namely, $|c_2 e|_\infty \leqslant (q/2) \cdot B$ where $|e|_\infty < B$ for any $e \leftarrow \chi_{\mathrm{e}}$ with very high probability.

The relinearisation noise can be decreased by decomposing c_2 in some base $w \geqslant 2$ such that $c_2 = \sum_{i=0}^{\ell} w^i c_{2,i}$ where $\ell = \lfloor \log_w q \rfloor$. Given this decomposition, the product $c_2 s^2$ can be written as

$$c_2 s^2 = \sum_{i=0}^{\ell} w^i c_{2,i} s^2 = \langle c_{2,i}, (w^i s^2)_i \rangle .$$

Now, instead of masking s^2, we create ℓ masked versions of $w^i s^2$ for any $i \in [0, \ell]$:

$$\texttt{rlk}[i] = (\left[-a_i s + e_i + w^i s^2\right]_q, a_i) .$$

Since $\left[\texttt{rlk}[i][0] + \texttt{rlk}[i][1] \cdot s\right]_q = w^i s^2 + e_i$, we obtain

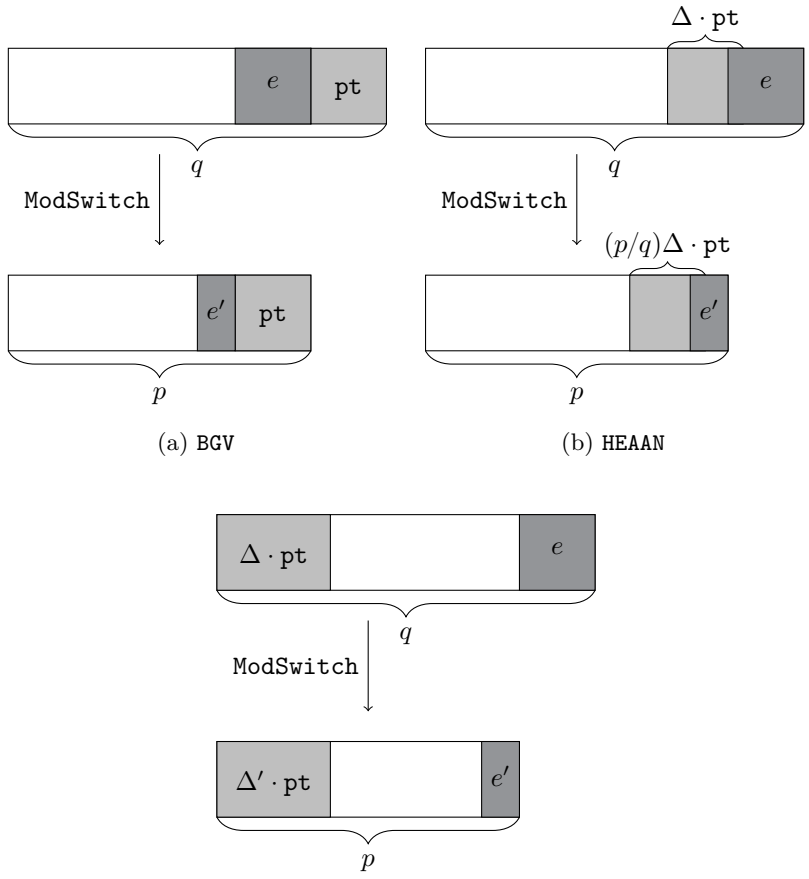
$$\begin{aligned}
\left[\texttt{ct}(s)\right]_q &= \left[c_0 + c_1 s + c_2 \cdot s)\right]_q \\
&= \left[c_0 + c_1 s + \sum_{i=0}^{\ell} c_{2,i} w^i s^2\right]_q \\
&= \left[c_0 + c_1 s + \sum_{i=0}^{\ell} c_{2,i} (\texttt{rlk}[i][0] + \texttt{rlk}[i][1] \cdot s)\right]_q \\
&= \left[c_0 + \sum_{i=0}^{\ell} c_{2,i} \cdot \texttt{rlk}[i][0] + \left(c_1 + \sum_{i=0}^{\ell} c_{2,i} \cdot \texttt{rlk}[i][1]\right) s - \sum_{i=0}^{\ell} c_{2,i} e_i\right]_q \\
&= \left[c'_0 + c'_1 \cdot s - \sum_{i=0}^{\ell} c_{2,i} e_i\right]_q .
\end{aligned}$$

Hence, the ciphertext $\texttt{ct} = (c'_0, c'_1)$ encrypts the same message as $\texttt{ct}$ with the additional noise $e_{\texttt{Relin}} = \sum_{i=0}^{\ell} c_{2,i} e_i$. Since $|e_{\texttt{Relin}}|_\infty \leqslant (\ell + 1)(w/2)B$, the relinearisation noise gets smaller as w decreases, but at the cost of additional relinearisation keys and thus additional operations to compute c'_0 and c'_1. In the extreme case $w = 2$, the relinearisation noise is bounded by $(\lfloor \log_2 q \rfloor + 1)B$, which is much smaller than the bound $(q/2) \cdot B$ we had with a single encryption $\texttt{rlk}$ of s^2.

Modulus switching. Given a ciphertext $\texttt{ct} = (c_0, c_1)$ and a modulus $p < q$, the modulus switching function computes

$$\texttt{ct}' = \left(\left\lfloor \frac{p}{q} \cdot c_0 \right\rceil, \left\lfloor \frac{p}{q} \cdot c_1 \right\rceil\right) .$$

Both BGV and HEAAN use modulus switching to change their ciphertext modulus size after homomorphic operations. If p is sufficiently small, then the modulus switching function decreases the inherent noise of its input ciphertext as shown in Figure 4.2.



(a) BGV (b) HEAAN

(c) FV

Figure 4.2: The effect of modulus switching on the decryption structures of BGV, HEAAN and FV.

As mentioned earlier, modulus switching plays a crucial role in reducing the inherent noise in BGV and HEAAN. In particular, the inherent noise in these schemes grows multiplicatively after homomorphic multiplication, from size B to B^2. If p is taken such that $p/q \simeq B$, then modulus switching after

multiplication outputs noise of size about B. Thus, we can keep the inherent noise within a certain range while performing homomorphic operations.

In HEAAN, modulus switching also changes the scaling parameter Δ. This can be useful to align ciphertext messages with different scaling parameters, if $q/p \simeq \Delta$.

Despite its functionality, modulus switching introduces additional complexity as ciphertexts dynamically change their sizes and thus must be appropriately aligned before being sent to the next arithmetic operator. For example, we want to homomorphically compute $x^2 + x$. First, we square a given input $\mathtt{ct} \in R_q^2$, relinearise it and switch to modulus p. The resulting ciphertext $\mathtt{ct}' \in R_p^2$ must be then added to ct, which is impossible as both ciphertexts lie in different ciphertext spaces. To align the ciphertexts, we do one more modulus switching in BGV. In HEAAN, we multiply ct by an encryption of 1 (HEAAN), thus increasing its noise. Then we relinearise the result and finally switch to the modulus p. So instead of having one multiplication and one addition as in the plaintext space, we perform more operations in both schemes.

In FV-type schemes, modulus switching is not needed; this is why such schemes are called *scale-invariant*. However, modulus switching can be used as an additional technique to control the noise and speed up computations as ciphertexts become smaller after this operation. Note that the scaling parameter Δ should also change in order to decrypt resulting messages as shown in Figure 4.2c.

4.2 Fan-Vercauteren scheme

The FV scheme was introduced in an unpublished manuscript of Fan and Vercauteren [54]. This scheme is a RLWE version of the Brakerski scheme [22], which is based on the LWE problem.

Despite its "unofficial" status, FV is considered among the most efficient FHE schemes due to its noise-controlling techniques. In the previous section we mentioned that FV is a scale-invariant scheme that makes it more user-friendly in comparison to BGV and HEAAN, where modulus switching is necessary. Therefore, FV was implemented and actively used in several popular HE libraries such as SEAL [111], FV-NFLlib [85] and Palisade [104].

In this section, we provide a thorough analysis of the internal operations in FV.

4.2.1 Basic scheme

We start by describing the basic encryption operations. Note that a similar encryption scheme without homomorphic functionality was firstly mentioned as an application of RLWE in the full version of [91]. In addition, the secret key of this scheme is a uniformly random element of R_q, whereas it is ternary in FV.

Parameters. Given a security level λ, we choose the ciphertext modulus q, the standard deviation σ and the ring dimension n, which is a power of 2. Define a $2n$-th cyclotomic ring of integers $R = \mathbb{Z}[X]/\langle X^n + 1\rangle$. Let R_q^2 be the ciphertext space and R_t be the plaintext space with $t \ll q$.

Let χ_{e} be the error distribution equal to the discretisation of a normal spherical Gaussian distribution $\lfloor \Gamma_r \rceil$ over R with standard deviation σ. Let χ_{k} be the key distribution that samples polynomials with random ternary coefficients drawn from $\mathcal{U}_3$.

Set the scaling parameter $\Delta = \lfloor q/t \rfloor$ and the relinearisation base $w \leqslant 2$ and compute $\ell = \lfloor \log_w q \rfloor$.

As in Section 4.1.1, we define the key generation, the encryption and the decryption functions.

Key generation.

- $\texttt{FV.SecretKeyGen}(1^\lambda)$: sample $s \leftarrow \chi_{\mathrm{k}}$ and output s.
- $\texttt{FV.PublicKeyGen}(s)$: sample $a \xleftarrow{\$} R_q, e \leftarrow \chi_{\mathrm{e}}$, and output
$$\texttt{pk} = \left(\left[-as + e \right]_q, a \right).$$

Encryption and decryption.

- $\texttt{FV.Encrypt}(\texttt{pk}, \texttt{pt})$: to encrypt message $\texttt{pt} \in R_t$, sample $u \leftarrow \chi_{\mathrm{k}}$ and $e_1, e_2 \leftarrow \chi_{\mathrm{e}}$. Return
$$\texttt{ct} = \left(\left[\Delta \cdot \texttt{pt} + u \cdot \texttt{pk}[0] + e_1 \right]_q, \left[u \cdot \texttt{pk}[1] + e_2 \right]_q \right).$$
- $\texttt{FV.Decrypt}(\texttt{ct}, s)$: given a ciphertext $\texttt{ct}$ encrypting the message $\texttt{pt}$, compute
$$\texttt{pt}' = \left[\left\lfloor \frac{t}{q} \left[\texttt{ct}[0] + \texttt{ct}[1] \cdot s \right]_q \right\rceil \right]_t.$$

If $\mathtt{pt}' = \mathtt{pt}$, then decryption is successful.

Every ciphertext message contains a noisy component. The *invariant noise* of a ciphertext $\mathtt{ct}$ is an element $v \in K$ of the smallest infinity norm such that

$$\frac{t}{q} \cdot [\mathtt{ct}[0] + \mathtt{ct}[1] \cdot s]_q = \mathtt{pt} + v + tg$$

for some $g \in R$. The decryption function outputs a correct plaintext message as long as $\|v\|^{\mathrm{can}} < 1/2$ according to the following theorem.

Theorem 4.2.1 (Decryption noise)**.** *Let* $\mathtt{ct} = (c_0, c_1)$ *be an encryption of the plaintext element* $\mathtt{pt} \in R_t$ *such that its invariant noise* v *satisfies* $\|v\|^{\mathrm{can}} < 1/2$*. Then* $\mathtt{FV.Decrypt}(\mathtt{ct}, s)$ *outputs* $\mathtt{pt}' = \mathtt{pt}$*.*

Proof. Computing $\mathtt{FV.Decrypt}(\mathtt{ct}, s)$, we have for some polynomials $g \in R$

$$\begin{aligned} \mathtt{pt}' &= \left\lfloor \frac{t}{q} [\mathtt{ct}[0] + \mathtt{ct}[1] \cdot s]_q \right\rceil \\ &= \lfloor \mathtt{pt} + v + tg \rceil \\ &= \mathtt{pt} + \lfloor v \rceil + tg\,. \end{aligned}$$

Since $|v|_\infty \leqslant \|v\|^{\mathrm{can}} < 1/2$, rounding in the last line removes v. Thus, after reduction modulo t we obtain the message $\mathtt{pt}' = \mathtt{pt}$.

□

Now, we estimate the size of the noise introduced by encryption. For this purpose, we use the heuristic approach of Gentry et al. [64]. This approach relies on the average distributional analysis, which estimates the expected size of the invariant noise in the canonical embedding norm.

For convenience, the ciphertext of the message $\mathtt{pt}$ with invariant noise v is denoted by $\mathtt{ct}(\mathtt{pt}, v)$.

Encryption noise heuristic. Let $\mathtt{ct}$ be a fresh ciphertext $\mathtt{ct} = \mathtt{FV.Encrypt}(\mathtt{pk}, \mathtt{pt})$. Set $c_0 = \mathtt{ct}[0]$, $c_1 = \mathtt{ct}[1]$, and $p_0 = \mathtt{pk}[0], p_1 = \mathtt{pk}[1]$. For some $g \in R$ it holds

$$\frac{t}{q} \cdot [c_0 + c_1 \cdot s]_q = \frac{t}{q} \cdot (\Delta \cdot \mathtt{pt} + p_0 u + e_0 + p_1 us + e_1 s + qg)\,.$$

Since $\Delta t = q - |q|_t$, the right-hand side results in

$$\mathtt{pt} - \frac{|q|_t}{q} \cdot \mathtt{pt} + \frac{t}{q}(p_0 u + e_0 + p_1 us + e_1 s) + tg\,.$$

Using the formula of the public key, we obtain for some $h \in R$

$$\begin{aligned}\mathtt{pt} - \frac{|q|_t}{q} \cdot \mathtt{pt} + \frac{t}{q}((-as + e + hq)u + aus + e_1 s) + tg \\ = \mathtt{pt} - \frac{|q|_t}{q} \cdot \mathtt{pt} + \frac{t}{q}(eu + e_1 + e_2 s) + t(g + hu)\,.\end{aligned}$$

Here, the noise term is $v = (-|q|_t \cdot \mathtt{pt} + t(eu + e_1 + e_2 s))/q$. Given the results of Section 2.8, the variance of the Gaussian variable $\|eu + e_1 + e_2 s\|^{\mathrm{can}}$ is equal to $\sigma\sqrt{4n^2/3 + n}$. Hence, the noise is bounded by

$$\|v\|^{\mathrm{can}} \leqslant \frac{t}{q}\left(\frac{n(t-1)}{2} + 2\sigma\sqrt{12n^2 + 9n}\right)$$

with overwhelming probability.

Combining this result with Theorem 4.2.1, it follows that the scheme parameters should be set such that

$$\frac{t}{q}\left(\frac{n(t-1)}{2} + 2\sigma\sqrt{12n^2 + 9n}\right) < \frac{1}{2}$$

to guarantee the decryption correctness with very high probability.

4.2.2 Homomorphic operations

Addition.

As shown in Section 4.1, addition of two ciphertexts is done component-wise. $\mathtt{FV.Add}\,(\mathtt{ct}_1, \mathtt{ct}_2)$: return

$$\mathtt{ct}_{\mathtt{Add}} = \left([\mathtt{ct}_1[0] + \mathtt{ct}_2[0]]_q\,, [\mathtt{ct}_1[1] + \mathtt{ct}_2[1]]_q\right)\,.$$

The noise after `FV.Add` grows additively as shown below.

Addition noise. Given two ciphertexts $\mathtt{ct}_1 = \mathtt{ct}(\mathtt{pt}_1, v_1)$ and $\mathtt{ct}_2 = \mathtt{ct}(\mathtt{pt}_2, v_2)$, the function $\mathtt{FV.Add}(\mathtt{ct}_1, \mathtt{ct}_2)$ returns a ciphertext

$$\mathtt{ct}_{\mathtt{Add}} = \mathtt{ct}([\mathtt{pt}_1 + \mathtt{pt}_2]_t, v_{\mathtt{Add}}).$$

Let $\mathtt{ct}_1 = (c_0, c_1)$ and $\mathtt{ct}_2 = (d_0, d_1)$. Performing the decryption steps before rounding, we obtain

$$\begin{aligned}\frac{t}{q}\left[\mathtt{ct}_{\mathtt{Add}}(s)\right]_q &= \frac{t}{q}\left[\mathtt{ct}_1(s) + \mathtt{ct}_2(s)\right]_q \\ &= \frac{t}{q}([\mathtt{ct}_1(s)]_q + [\mathtt{ct}_2(s)]_q + qg) \\ &= (\mathtt{pt}_1 + v_1 + tg_1) + (\mathtt{pt}_2 + v_2 + tg_2) + tg \\ &= [\mathtt{pt}_1 + \mathtt{pt}_2]_t + (v_1 + v_2) + t(g_1 + g_2 + g + h)\end{aligned}$$

for some $g_1, g_2, g, h \in R$. Assuming that $\mathtt{ct}_1$ and $\mathtt{ct}_2$ are decryptable, i.e. $\|v_1\|^{\mathrm{can}}, \|v_2\|^{\mathrm{can}} < 1/2$, it follows that $|v_1 + v_2| < t$. Hence, $v_{\mathtt{Add}} = v_1 + v_2$ is the invariant noise of $\mathtt{ct}_{\mathtt{Add}}$.

Multiplication.

By homomorphic multiplication, we define a two-step algorithm consisting of basic multiplication and relinearisation.

Basic multiplication. Given two ciphertexts $\mathtt{ct}_1 = (c_0, c_1)$ and $\mathtt{ct}_2 = (d_0, d_1)$, basic multiplication computes their convolution as described in the general framework. To reduce the number of R_q-products, we can resort to the Karatsuba algorithm [76] and compute

$$z_0 = c_0 d_0, \quad z_1 = (c_0 + d_0)(c_1 + d_1), \quad z_2 = c_1 d_1\,.$$

Then, the convolution result is given by a triple $(z_0, z_1 - z_0 - z_2, z_2)$. Note that before reduction modulo q, the convolution components must be scaled down by t/q to remove an excessive factor of Δ. As a result, the basic multiplication function is defined as

$\mathtt{FV.BasicMul}(\mathtt{ct}_1, \mathtt{ct}_2)$: compute

$$c_0 = \left[\left\lfloor \frac{t}{q} c_0 d_0 \right\rceil\right]_q, \quad c_1 = \left[\left\lfloor \frac{t}{q}\left(c_0 d_1 + c_1 d_0\right) \right\rceil\right]_q, \quad c_2 = \left[\left\lfloor \frac{t}{q} c_1 d_1 \right\rceil\right]_q$$

and return $\mathtt{ct}_{\mathtt{BasicMul}} = (c_0, c_1, c_2)$.

Noise heuristic after basic multiplication. Given two ciphertexts $\mathtt{ct}_1 = \mathtt{ct}(\mathtt{pt}_1, v_1)$ and $\mathtt{ct}_2 = \mathtt{ct}(\mathtt{pt}_2, v_2)$, the function $\mathtt{FV.BasicMul}(\mathtt{ct}_1, \mathtt{ct}_2)$ returns a triple $\mathtt{ct}_{\mathtt{BasicMul}} = (c_0, c_1, c_2)$ such that

$$\frac{t}{q}\left[c_0 + c_1 s + c_2 s^2\right]_q = \left[\mathtt{pt}_1 \cdot \mathtt{pt}_2\right]_t + v_{\mathtt{BasicMul}} + th$$

for some $h \in R$.

According to the description of $\mathtt{FV.BasicMul}$, every component c_i of $\mathtt{ct}_{\mathtt{BasicMul}}$ contains a rounding error r_i, $|r_i|_\infty \leqslant 1/2$. Thus, decrypting $\mathtt{ct}_{\mathtt{BasicMul}}$ leads to the following equality

$$\frac{t}{q}\left[c_0 + c_1 s + c_2 s^2\right]_q = \frac{t^2}{q^2}(\mathtt{ct}_1[0] + \mathtt{ct}_1[1]\cdot s)(\mathtt{ct}_2[0] + \mathtt{ct}_2[1]\cdot s) + r + tg\,, \quad (4.3)$$

where $r = t(r_0 + r_1 s + r_2 s^2)/q$ and $g \in R$. Since $\mathtt{ct}_1$ and $\mathtt{ct}_2$ are pseudo-random, we can view the coefficients of r_i being sampled from $\mathcal{U}_{\mathrm{rnd}}$. Hence, the variance of $\left\|r_0 + r_1 s + r_2 s^2\right\|^{\mathrm{can}}$ is equal to $n/12 + n^2/18 + n^3/27$. It follows

$$\begin{aligned}\|r\|^{\mathrm{can}} &\leqslant \frac{t}{q} 6\sqrt{n/12 + n^2/18 + n^3/27} \\ &= \frac{t}{q}\sqrt{3n + 2n^2 + 4n^3/3}\,.\end{aligned}$$

Let $\mathtt{pt}_1 \cdot \mathtt{pt}_2 - \left[\mathtt{pt}_1 \cdot \mathtt{pt}_2\right]_t = tg_0$. Expanding (4.3), we obtain for some $g_1, g_2 \in R$

$$\begin{aligned}\frac{t}{q}\left[c_0 + c_1 \cdot s + c_2 \cdot s^2\right]_q &= (\mathtt{pt}_1 + v_1 + tg_1)(\mathtt{pt}_2 + v_2 + tg_2) + r + tg \\ &= \left[\mathtt{pt}_1 \cdot \mathtt{pt}_2\right]_t + v_2(\mathtt{pt}_1 + tg_1) + v_1(\mathtt{pt}_2 + tg_2) \\ &\quad + v_1 v_2 + r \\ &\quad + t(g_0 + \mathtt{pt}_1 \cdot g_2 + \mathtt{pt}_2 \cdot g_1 + g + tg_1 g_2) \\ &= \left[\mathtt{pt}_1 \cdot \mathtt{pt}_2\right]_t + v_{\mathtt{BasicMul}} + th\,.\end{aligned}$$

As $\mathtt{ct}_i[0], \mathtt{ct}_i[1] \in R_q$ are indistinguishable from uniformly random, the term $\mathtt{ct}_i[0] + \mathtt{ct}_i[1] \cdot s$ has variance $q^2 n/12 + q^2 n^2/18$. It follows

$$\begin{aligned}\|\mathtt{pt}_i + tg_i\|^{\mathrm{can}} &= \left\|\frac{t}{q}\left(\mathtt{ct}_i[0] + \mathtt{ct}_i[1]\cdot s\right) - v_i\right\|^{\mathrm{can}} \\ &\leqslant t\sqrt{3n + 2n^2} + \|v_i\|^{\mathrm{can}}\,.\end{aligned}$$

Hence, the invariant noise $v_{\texttt{BasicMul}}$ is bounded by

$$\begin{aligned}\|v_{\texttt{BasicMul}}\|^{\text{can}} &\leqslant \|v_2\|^{\text{can}} \left(t\sqrt{3n+2n^2} + \|v_1\|^{\text{can}}\right) + \|v_1\|^{\text{can}} \left(t\sqrt{3n+2n^2} + \|v_2\|^{\text{can}}\right) \\ &\quad + \|v_1\|^{\text{can}} \|v_2\|^{\text{can}} + \frac{t}{q}\sqrt{3n+2n^2+4n^3/3} \\ &= t\sqrt{3n+2n^2}\left(\|v_1\|^{\text{can}} + \|v_2\|^{\text{can}}\right) + 3\|v_1\|^{\text{can}} \|v_2\|^{\text{can}} \\ &\quad + \frac{t}{q}\sqrt{3n+2n^2+4n^3/3}\end{aligned}$$

with very high probability.

Remarkably, the leading term of $v_{\texttt{BasicMul}}$ is the first one as $\|v_1\|^{\text{can}}, \|v_2\|^{\text{can}} < 1/2$ assuming that the input ciphertexts can be decrypted correctly. As a result, the noise after basic multiplication grows linearly, while in `BGV` and `HEAAN` the noise growth is multiplicative after this operation.

Relinearisation. To reduce the dimension of the `FV.BasicMul` outcome from 3 to 2, we apply relinearisation.

First, we generate relinearisation keys, which are the masked versions of the decomposition of s^2 in base w.

`FV.RelinKeyGen`(s): sample $a_i \xleftarrow{\$} R_q$ and $e_i \leftarrow \chi_{\text{e}}$, output

$$\texttt{rlk} = \left\{ \left(\left[w \cdot s^2 - a \cdot s + e_i \right]_q, a_i \right) : i \in [0, \ell] \right\} .$$

Given a 3-component ciphertext `ct` and a relinearisation key `rlk`, we perform relinearisation as follows.

`FV.Relin`$(\texttt{ct}, \texttt{rlk})$: expand $\texttt{ct}[2]$ in base w, i.e. $\texttt{ct}[2] = \sum_{i=0}^{\ell} d_i w^i$, then compute

$$c_0 = \left[\texttt{ct}[0] + \sum_{i=0}^{\ell} \texttt{rlk}[i][0] \cdot d_i \right]_q ,$$

$$c_1 = \left[\texttt{ct}[1] + \sum_{i=0}^{\ell} \texttt{rlk}[i][1] \cdot d_i \right]_q ,$$

and return $\mathtt{ct}_{\mathtt{Mul}} = (c_0, c_1)$.

As shown in Section 4.1, relinearisation introduces additional inherent noise equal to $\sum_{i=0}^{\ell} e_i d_i$. In the following lemma, we estimate how the invariant noise changes after this operation.

Noise heuristic after relinearisation. Given a triple $\mathtt{ct} = (c_0, c_1, c_2)$ encrypting a plaintext $\mathtt{pt}$ and containing noise v, the relinearisation function returns a ciphertext $\mathtt{ct}_{\mathtt{Relin}} = \mathtt{ct}(\mathtt{pt}, v_{\mathtt{Relin}})$.

Let $\mathtt{ct}_{\mathtt{Relin}} = (c_0', c_1')$. We obtain for some $g, g_0, \ldots, g_\ell \in R$

$$\begin{aligned}
\frac{t}{q}\left[c_0' + c_1' s\right]_q &= \frac{t}{q}\left(c_0 + \sum_{i=0}^{\ell} \mathtt{rlk}[i][0] \cdot d_i + c_1 s + \sum_{i=0}^{\ell} \mathtt{rlk}[i][1] \cdot d_i s\right) + tg \\
&= \frac{t}{q}\left(c_0 + c_1 s + \sum_{i=0}^{\ell} \left(-a_i s + e_i + w^i s^2 + g_i q + s a_i\right) d_i\right) + tg \\
&= \frac{t}{q}\left(c_0 + c_1 s + \sum_{i=0}^{\ell} e_i d_i + s^2 \sum_{i=0}^{\ell} d_i w^i\right) + t\left(\sum_{i=0}^{\ell} g_i d_i + g\right).
\end{aligned}$$

Recall that by definition $\sum_i d_i w^i = c_2$. Thus, replacing $\sum_i g_i d_i + g$ by h, we obtain

$$\frac{t}{q}\left[c_0' + c_1' s\right]_q = \frac{t}{q}\left(c_0 + c_1 s + c_2 s^2\right) + \frac{t}{q}\sum_{i=0}^{\ell} e_i d_i + th$$

Since $\mathtt{ct}$ encrypts the message $\mathtt{pt}$ and has invariant noise v, the right-hand side can be rewritten for some $h' \in R$ as

$$\mathtt{pt} + v + \frac{t}{q}\sum_{i=0}^{\ell} e_i d_i + t(h' + h).$$

As a result, $v_{\mathtt{Relin}} = v + \frac{t}{q}\sum_{i=0}^{\ell} e_i d_i$. Given that d_i's look uniformly random in R_w, the variance of $\left\|\sum_{i=0}^{\ell} e_i d_i\right\|^{\mathrm{can}}$ is equal to $(\ell+1)(w\sigma n)^2/12$. Hence, the canonical norm of this term is bounded by $w\sigma n\sqrt{3(\ell+1)}$. It follows that

$$\begin{aligned}
\|v_{\mathtt{Relin}}\|^{\mathrm{can}} &\leqslant \|v\|^{\mathrm{can}} + \frac{t}{q}\left\|\sum_{i=0}^{\ell} e_i d_i\right\|^{\mathrm{can}} \\
&\leqslant \|v\|^{\mathrm{can}} + \frac{t}{q} w\sigma n\sqrt{3(\ell+1)}
\end{aligned}$$

with very high probability.

Now, we have all the components to compute homomorphic multiplication.

$\texttt{FV.Mul}(\texttt{ct}_1, \texttt{ct}_2, \texttt{rlk})$: return

$$\texttt{FV.Relin}(\texttt{FV.BasicMul}(\texttt{ct}_1, \texttt{ct}_2), \texttt{rlk}) .$$

Noise heuristic after multiplication. The total invariant noise growth is given by combining the above noise heuristics. Given two ciphertexts $\texttt{ct}_1 = \texttt{ct}(\texttt{pt}_1, v_1)$ and $\texttt{ct}_2 = \texttt{ct}(\texttt{pt}_2, v_2)$, the function $\texttt{FV.Mul}(\texttt{ct}_1, \texttt{ct}_2, \texttt{rlk})$ outputs a ciphertext $\texttt{ct}_{\texttt{Mul}} = \texttt{ct}([\texttt{pt}_1 \cdot \texttt{pt}_2]_t, v_{\texttt{Mul}})$ where

$$\|v_{\texttt{Mul}}\|^{\text{can}} \leqslant t\sqrt{3n + 2n^2}\,(\|v_1\|^{\text{can}} + \|v_2\|^{\text{can}}) + 3\,\|v_1\|^{\text{can}}\,\|v_2\|^{\text{can}}$$

$$+ \frac{t}{q}\left(\sqrt{3n + 2n^2 + 4n^3/3} + w\sigma n\sqrt{3(\ell + 1)}\right)$$

with very high probability.

4.2.3 Optimisations

All computations in the FV scheme are performed on polynomials in the cyclotomic ring R_q, which is defined by two parameters: q and n. These parameters and the noise standard deviation should be large enough to support a certain security level of the corresponding RLWE problem. In addition, the ciphertext modulus q should be large to accommodate all the noise introduced by homomorphic operations.

In practice, n is taken within the range $[2^{10}, 2^{15}]$ accompanied by q having hundreds of bits. This results in huge polynomials in R_q, which causes two problems. First, arithmetic operations on these polynomials are heavy, meaning that thousands of large polynomial coefficients must be handled in memory. Second, given that the plaintext modulus is much smaller than the ciphertext modulus, we obtain large ciphertext-to-plaintext expansion ratio. For example, to encrypt and perform one multiplication on bits, we need at least 32 kB of memory. To reduce this computational and memory overhead, the following optimisations were proposed.

CRT for faster arithmetic

Residue number system. As we know from Section 2.9, CRT can be used to represent any quotient of R as a direct product of smaller quotient rings. Let

the ciphertext modulus q be a composite integer such that it has a co-prime factorisation with k factors: $q = \prod_{i=1}^{k} q_i$. For the ring R_q, CRT yields the following ring isomorphism

$$R_q \to R_{q_1} \times \ldots \times R_{q_k} ,$$

$$a \mapsto \mathbf{a} = \left([a]_{q_1} , \ldots , [a]_{q_k} \right) .$$

Hence, arithmetic operations on elements $a, b \in R_q$ can be performed by mapping these elements to vectors $\mathbf{a}, \mathbf{b} \in \prod_i R_{q_i}$ and performing coefficient-wise arithmetic operations on these vectors:

$$[a+b]_q \mapsto \mathbf{a} + \mathbf{b} = \left([a+b]_{q_1} , \ldots , [a+b]_{q_k} \right) ,$$

$$[ab]_q \mapsto \mathbf{a} \circ \mathbf{b} = \left([ab]_{q_1} , \ldots , [ab]_{q_k} \right) .$$

This numerical representation, called the *residue number system* (RNS), was introduced independently by Valach [94] and Garner [58].

Note that RNS arithmetic is done in rings with much smaller moduli than q. For example, q can be chosen such that its co-prime factors fit one machine word. This decreases the hardware latency of arithmetic modulo q_i, thus resulting in better performance. Furthermore, arithmetic at each coordinate can be done independently from the results of other coordinates. As a result, the vector operations can be efficiently parallelised.

In [11], Bajard et al. implemented all operations of the `FV` scheme in RNS. They demonstrated that the use of RNS improves the scheme's performance at the cost of slightly increased noise.

Number theoretic transform. In RNS, the CRT isomorphism is constructed with relation to a co-prime factorisation of q. Alternatively, we can look at the factorisation of the defining polynomial $f = X^n + 1$ into co-prime components modulo q. Let us consider the extreme case when f factors into linear components. This is possible when $q \equiv 1 \mod 2n$. Let $f(X) = \prod_{i=1}^{n} (X - \alpha_i)$ be such factorisation where $\alpha_1, \ldots, \alpha_n$ are the roots of f modulo q. Then the CRT isomorphism yields

$$R_q \to R_q / \langle X - \alpha_1 \rangle \times \ldots \times R_q / \langle X - \alpha_n \rangle \cong \mathbb{Z}_q^n ,$$

$$a \mapsto \mathbf{a} = \left([a(\alpha_1)]_q , \ldots , [a(\alpha_k)]_q \right) .$$

This isomorphism, called the number theoretic transform (NTT), resembles a finite-field analogue of the canonical embedding and the discrete Fourier transform. Hence, NTT and its inverse can be computed with $O(n \log n)$ multiplications in $\mathbb{Z}_q$ by existing FFT algorithms, for instance by the Cooley-Tukey method [39].

NTT can be exploited to speed up integer multiplication as shown by Schönhage and Strassen [110]. Their work can be easily applied to multiplication in R_q. To multiply two elements $a, b \in R_q$, we map them via NTT to $\mathbb{Z}_q^n$, multiply coefficient-wise and send back to R_q through the inverse of NTT. All these operations take $O(n \log n)$ multiplications, whereas schoolbook polynomial multiplication in R_q requires $O(n^2)$ multiplications. Hence, NTT results in a significant speed up of polynomial arithmetic. Furthermore, in 2-power cyclotomic rings NTT can be further optimised [87], but not asymptotically.

CRT for packing

It is easy to see that the aforementioned CRT techniques can be applied to the plaintext space R_t as well. Let $t = \prod_{i=1}^{k} t_i$ be a factorisation of the plaintext modulus into co-prime factors. Hence, CRT results in the following isomorphism

$$R_t \to R_{t_1} \times \ldots \times R_{t_k} \,,$$

$$a \mapsto \mathbf{a} = \left([a]_{t_1}, \ldots, [a]_{t_k}\right).$$

We can go further and apply CRT to every ring R_{t_i} using factorisations of the defining polynomial. Modulo each t_i, the defining polynomial f factors in ℓ_i irreducible polynomials $f = \prod_{j=1}^{\ell_i} f_{ij} \mod t_i$. As a result, the following isomorphism holds for every $i \in [k]$

$$R_{t_i} \to R_{t_i}/\langle f_{i1} \rangle \times \ldots \times R_{t_i}/\langle f_{i\ell_i} \rangle .$$

Now, we change our view on these CRT isomorphisms by starting from the rings $R_{t_i}/\langle f_{ij} \rangle$, which are called *slots*. We can encode different data values to polynomials in these slots and then combine these polynomials using the inverses of the CRT isomorphisms above. The resulting plaintext `pt` then contains, or *packs*, all the data we have encoded.

Every arithmetic operation on `pt` results in the same operation performed simultaneously on all slots. In other words, we can homomorphically perform SIMD (single-instruction multiple-data) operations. This is why this packing algorithm is often referred to as the SIMD packing.

The SIMD packing significantly decreases the ciphertext-to-plaintext expansion ratio. Assuming for simplicity that f splits into ℓ factors modulo any t_i, each

plaintext can accommodate $k\ell$ encoded values and thus reduce the memory overhead by a factor of $k\ell$. In Chapter 10, we further discuss functional capabilities of this technique and demonstrate how to encode data.

Exotic plaintext spaces

In the original description of the `FV` scheme the plaintext space is given by the polynomial ring R_t, whose elements are polynomials of degree at most $n-1$ with coefficients taken from the interval $[-t/2, t/2)$. The plaintext modulus t can be just 2, which is enough to encode binary bits and compute Boolean arithmetic. However, Boolean arithmetic involves more homomorphic operations to perform computational tasks on numerical data than arithmetic circuits. For instance, addition of two k-bit integers needs $O(k)$ bitwise `AND` operations and thus a Boolean circuit of multiplicative depth $O(k)$. It is thus desirable to encode numerical types such as integer, rational, real and complex numbers directly into the plaintext space in order to have homomorphic operations resulting in arithmetic operations on these types.

As shown in Chapters 8 and 9, the algebraic structure of R_t is not a great match with numerical data types. The size of R_t should be large enough to preserve the decoding correctness, thus leading to faster noise growth and reduced efficiency.

Instead of R_t, we can employ another plaintext space which is isomorphic to some large ring containing numerical values, e.g. $\mathbb{Z}_p$ with large p. To construct such a ring, Brakerski et al. [24] proposed to replace the ideal $\langle t \rangle$ by a principal ideal $\mathcal{I} = \langle g \rangle \in R$. For instance, if $g = X - b$, then $R/\mathcal{I}$ is isomorphic to $\mathbb{Z}_{b^n+1}$. To instantiate the `FV` scheme with this new plaintext space, the plaintext modulus should be replaced by g in `FV.Decrypt` and `FV.BasicMul`. The scaling parameter Δ must be computed as follows

$$\Delta = \left\lfloor qg^{-1} \mod (X^n + 1) \right\rceil .$$

In Chapter 11, we show how to use this approach to encode complex number directly into the plaintext space. In addition, we demonstrate that the noise growth of the new `FV` scheme depends on the norm of g, so the new plaintext modulus must be chosen short.

Chapter 5

Conclusions and future work

5.1 Conclusions

When we started this research in 2015, FHE schemes were mainly considered as a theoretical concept. Despite its versatile applicability, the computational overhead of FHE was incompatible with practical tasks. The existing implementations of homomorphic algorithms demonstrated that secure computation on numerical data demands large parameters even for modest functionality.

In recent years, this situation has changed. Various optimisations and even new FHE schemes appeared, significantly improving the performance of homomorphic computing. This progress was possible due to the use of the rich algebraic structure of the underlying computational problems on which the security of FHE schemes is based. This structure offers numerous optimisations, which can lead to an efficiency gain.

In line with these achievements, our contribution consists in expanding the set of tools that allow optimisation of FHE computation on numerical data without compromising security. In addition, we provide security analysis of new variants of the underlying computational problems that can lead to insecure instantiations of FHE. As a result, we can draw the following conclusions.

The underlying computational problems of FHE remain secure, but their variants can be efficiently attacked. We demonstrated that tweaking RLWE, the standard hard problem in FHE, leads to more powerful attacks than previously described. However, these attacks work for specific parameter sets

which are not used in practice. Moreover, our results do not compromise the security of RLWE with standard parameters. Therefore, the security of FHE schemes should be based on the original formulation of this problem.

FHE can efficiently compute on numerical data. Prior to our work, very few algorithms existed that allow encoding numerical data for FHE. Moreover, these algorithms inefficiently use the large capacity of FHE plaintext spaces, thus demanding large encryption parameters. We significantly optimised these algorithms and constructed new ones such that much smaller encryption parameters are sufficient. Our contributions include:

- A new encoding algorithm of real numbers that evenly spreads numerical data over the plaintext space. As a result, the plaintext space is used more efficiently in comparison to previously described methods and requires significantly smaller encryption parameters.
- The plaintext space of the fastest FHE schemes has a larger packing capacity than previously used. We achieved this capacity by generalising the existing SIMD packing method. Moreover, our technique can be exploited in combination with the above encoding algorithm, thus decreasing the ciphertext-to-plaintext expansion ratio.
- Non-standard plaintext spaces can be used to encode numerical data in a more natural way. We designed a new variant of the FV scheme especially for complex number arithmetic. The main idea is to use a new plaintext space which contains large complex numbers. The structure of the plaintext space dictates the design of the internal operations of this FHE scheme, which remarkably results in a better performance. Using this scheme, we can compute much deeper arithmetic circuits in comparison to the classical FHE schemes.

FHE is getting closer to practice, but at the cost of a tedious optimisation procedure. We proposed several techniques that can help reduce the overhead of FHE. Given a computational task and its input data, these methods can be combined and adjusted in order to homomorphically process as much data as possible. However, the parameters of these methods constitute a vast space such that the search of optimal parameters becomes a non-trivial problem.

Despite some limitations, FHE is applicable in a wide range of use cases. In recent years, FHE demonstrated reasonable performance in a great variety of use cases: image classification [66, 21], statistics [79], bioinformatics [15],

database search [2] etc. Our contribution consists in the first homomorphic forecasting algorithm for energy consumption, whose efficiency was significantly improved by our optimisation techniques. Nevertheless, there are some fundamental limitations of FHE in practice. First, FHE does not provide the integrity of computation results by default. This functionality requires additional cryptographic primitives, e.g. homomorphic signatures [67]. Second, bootstrapping still remains a prohibitively heavy operation in many FHE schemes. Therefore, most FHE applications avoid it by taking large enough encryption parameters.

5.2 Future work

Future research directions include the following tasks.

Security analysis of the hard computational problems in FHE. The most efficient FHE schemes are based on the RLWE problem, which is a structured variant of LWE over algebraic rings. It is still an open question whether the algebraic structure of RLWE can be exploited and can lead to more efficient attacks in comparison to LWE. It was recently shown in [41, 42, 102] that this structure leads to better attacks on SVP-type problems. Another undecided issue revolves around an important optimisation of RLWE which uses binary and sparse secret keys. The existing literature [25, 3, 6] demonstrate that switching to such keys while fixing the other encryption parameters decreases the security level of LWE. However, these results do not provide tight theoretical bounds on how much easier LWE and RLWE become in comparison to their original formulations.

FHE based on new computational problems. Most FHE schemes are based on hard lattice problems, which work with large integer vectors or polynomials. The implementation of these mathematical objects is cumbersome, which impedes the use of FHE in practice. It would be interesting to study new cryptographic primitives that allow homomorphic computation and simultaneously lead to simpler FHE schemes.

FHE for other data types. In this thesis we demonstrated a new FHE scheme whose plaintext space natively supports complex number arithmetic. Similar ideas were also applied for large integer arithmetic [29]. Matrix and tensor arithmetic can be realised as well [97, 75]. A promising direction for future

work would be to build new FHE schemes that have an innate capability to deal with other data types, e.g. floating points, strings. These schemes might potentially reduce the overhead of FHE when working with these data types.

Benchmarking homomorphic implementations of classical algorithms. Homomorphic computation introduces several computational constraints. For instance, homomorphic algorithms cannot terminate a loop as they are unable to read encrypted logical variables. Given these constraints, it would be interesting to compare the performance of the existing classical algorithms (e.g. sorting, pattern-matching) implemented homomorphically. This comparison might highlight the difference between the real and the encrypted worlds. It can also provide important insights into the design of homomorphic algorithms.

Conversion between different FHE schemes. Even though all FHE schemes can compute arbitrary circuits, some of those schemes demonstrate better efficiency for certain computational tasks than the others. For instance, `FV` and `BGV` are faster for integer arithmetic, while `TFHE` is more suitable for non-linear Boolean gates. Therefore, it is desirable to leverage the best qualities of several FHE schemes, which can be done via conversion between ciphertext and plaintext representations of these schemes. The first step in this line of research was done by Boura et al. [20], who showed how to switch between `FV` and `HEAAN` via `TFHE`. Further optimisation of these methods and new conversion algorithms are appealing research tasks.

Bibliography

[1] Ajtai, M., Kumar, R., and Sivakumar, D. A sieve algorithm for the shortest lattice vector problem. In *33rd Annual ACM Symposium on Theory of Computing* (July 2001), ACM Press, pp. 601–610.

[2] Akavia, A., Feldman, D., and Shaul, H. Secure search on encrypted data via multi-ring sketch. In *ACM CCS 18: 25th Conference on Computer and Communications Security* (Oct. 2018), D. Lie, M. Mannan, M. Backes, and X. Wang, Eds., ACM Press, pp. 985–1001.

[3] Albrecht, M. R. On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL. In *Advances in Cryptology – EUROCRYPT 2017, Part II* (Apr. / May 2017), J. Coron and J. B. Nielsen, Eds., vol. 10211 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 103–129.

[4] Albrecht, M. R., Bai, S., and Ducas, L. A subfield lattice attack on overstretched NTRU assumptions - cryptanalysis of some FHE and graded encoding schemes. In *Advances in Cryptology – CRYPTO 2016, Part I* (Aug. 2016), M. Robshaw and J. Katz, Eds., vol. 9814 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 153–178.

[5] Albrecht, M. R., Farshim, P., Faugère, J.-C., and Perret, L. Polly cracker, revisited. In *Advances in Cryptology – ASIACRYPT 2011* (Dec. 2011), D. H. Lee and X. Wang, Eds., vol. 7073 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 179–196.

[6] Albrecht, M. R., Göpfert, F., Virdia, F., and Wunderer, T. Revisiting the expected cost of solving uSVP and applications to LWE. In *Advances in Cryptology – ASIACRYPT 2017, Part I* (Dec. 2017), T. Takagi and T. Peyrin, Eds., vol. 10624 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 297–322.

[7] Albrecht, M. R., Player, R., and Scott, S. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology 9*, 3 (2015), 169–203.

[8] Alperin-Sheriff, J., and Peikert, C. Practical bootstrapping in quasilinear time. In *Advances in Cryptology – CRYPTO 2013, Part I* (Aug. 2013), R. Canetti and J. A. Garay, Eds., vol. 8042 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 1–20.

[9] Applebaum, B., Cash, D., Peikert, C., and Sahai, A. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *Advances in Cryptology – CRYPTO 2009* (Aug. 2009), S. Halevi, Ed., vol. 5677 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 595–618.

[10] Arora, S., and Ge, R. New algorithms for learning in presence of errors. In *ICALP 2011: 38th International Colloquium on Automata, Languages and Programming, Part I* (July 2011), L. Aceto, M. Henzinger, and J. Sgall, Eds., vol. 6755 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 403–415.

[11] Bajard, J.-C., Eynard, J., Hasan, M. A., and Zucca, V. A full RNS variant of FV like somewhat homomorphic encryption schemes. In *SAC 2016: 23rd Annual International Workshop on Selected Areas in Cryptography* (Aug. 2016), R. Avanzi and H. M. Heys, Eds., vol. 10532 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 423–442.

[12] Banaszczyk, W. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen 296*, 1 (1993), 625–635.

[13] Barkee, B., Can, D. C., Ecks, J., Moriarty, T., and Ree, R. Why you cannot even hope to use Gröbner bases in public key cryptography: an open letter to a scientist who failed and a challenge to those who have not yet failed. *Journal of Symbolic Computation 18*, 6 (1994), 497–501.

[14] Benaloh, J. C. Secret sharing homomorphisms: Keeping shares of a secret sharing. In *Advances in Cryptology – CRYPTO'86* (Aug. 1987), A. M. Odlyzko, Ed., vol. 263 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 251–260.

[15] Blatt, M., Gusev, A., Polyakov, Y., Rohloff, K., and Vaikuntanathan, V. Optimized homomorphic encryption solution for secure genome-wide association studies. Cryptology ePrint Archive, Report 2019/223, 2019. `https://eprint.iacr.org/2019/223`.

[16] Blömer, J., and Seifert, J.-P. On the complexity of computing short linearly independent vectors and short bases in a lattice. In *31st Annual ACM Symposium on Theory of Computing* (May 1999), ACM Press, pp. 711–720.

[17] Blum, A., Kalai, A., and Wasserman, H. Noise-tolerant learning, the parity problem, and the statistical query model. In *32nd Annual ACM Symposium on Theory of Computing* (May 2000), ACM Press, pp. 435–440.

[18] Boneh, D., Goh, E.-J., and Nissim, K. Evaluating 2-DNF formulas on ciphertexts. In *TCC 2005: 2nd Theory of Cryptography Conference* (Feb. 2005), J. Kilian, Ed., vol. 3378 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 325–341.

[19] Bos, J. W., Lauter, K., Loftus, J., and Naehrig, M. Improved security for a ring-based fully homomorphic encryption scheme. In *14th IMA International Conference on Cryptography and Coding* (Dec. 2013), M. Stam, Ed., vol. 8308 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 45–64.

[20] Boura, C., Gama, N., and Georgieva, M. Chimera: a unified framework for B/FV, TFHE and HEAAN fully homomorphic encryption and predictions for deep learning. Cryptology ePrint Archive, Report 2018/758, 2018. `https://eprint.iacr.org/2018/758`.

[21] Bourse, F., Minelli, M., Minihold, M., and Paillier, P. Fast homomorphic evaluation of deep discretized neural networks. In *Advances in Cryptology – CRYPTO 2018, Part III* (Aug. 2018), H. Shacham and A. Boldyreva, Eds., vol. 10993 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 483–512.

[22] Brakerski, Z. Fully homomorphic encryption without modulus switching from classical GapSVP. In *Advances in Cryptology – CRYPTO 2012* (Aug. 2012), R. Safavi-Naini and R. Canetti, Eds., vol. 7417 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 868–886.

[23] Brakerski, Z., Gentry, C., and Halevi, S. Packed ciphertexts in LWE-based homomorphic encryption. In *PKC 2013: 16th International Conference on Theory and Practice of Public Key Cryptography* (Feb. / Mar. 2013), K. Kurosawa and G. Hanaoka, Eds., vol. 7778 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 1–13.

[24] Brakerski, Z., Gentry, C., and Vaikuntanathan, V. (Leveled) fully homomorphic encryption without bootstrapping. In *ITCS 2012: 3rd*

Innovations in Theoretical Computer Science (Jan. 2012), S. Goldwasser, Ed., Association for Computing Machinery, pp. 309–325.

[25] Brakerski, Z., Langlois, A., Peikert, C., Regev, O., and Stehlé, D. Classical hardness of learning with errors. In *45th Annual ACM Symposium on Theory of Computing* (June 2013), D. Boneh, T. Roughgarden, and J. Feigenbaum, Eds., ACM Press, pp. 575–584.

[26] Brakerski, Z., and Vaikuntanathan, V. Efficient fully homomorphic encryption from (standard) LWE. In *52nd Annual Symposium on Foundations of Computer Science* (Oct. 2011), R. Ostrovsky, Ed., IEEE Computer Society Press, pp. 97–106.

[27] Brenner, M., Wiebelitz, J., Von Voigt, G., and Smith, M. Secret program execution in the cloud applying homomorphic encryption. In *Proceedings of the 5th IEEE International Conference on Digital Ecosystems and Technologies Conference (DEST)* (2011), IEEE Computer Society, pp. 114–119.

[28] Brickell, E. F., and Yacobi, Y. On privacy homomorphisms (extended abstract). In *Advances in Cryptology – EUROCRYPT'87* (Apr. 1988), D. Chaum and W. L. Price, Eds., vol. 304 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 117–125.

[29] Chen, H., Laine, K., Player, R., and Xia, Y. High-precision arithmetic in homomorphic encryption. In *Topics in Cryptology – CT-RSA 2018* (2018), N. P. Smart, Ed., vol. 10808 of *Lecture Notes in Computer Science*, Springer, Heidelberg. To appear.

[30] Chen, H., Laine, K., and Rindal, P. Fast private set intersection from homomorphic encryption. In *ACM CCS 17: 24th Conference on Computer and Communications Security* (Oct. / Nov. 2017), B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, Eds., ACM Press, pp. 1243–1255.

[31] Chen, H., Lauter, K., and Stange, K. E. Attacks on search RLWE. Cryptology ePrint Archive, Report 2015/971, 2015. `http://eprint.iacr.org/2015/971`.

[32] Chen, H., Lauter, K., and Stange, K. E. Vulnerable Galois RLWE families and improved attacks. Cryptology ePrint Archive, Report 2016/193, 2016. `http://eprint.iacr.org/2016/193`.

[33] Chen, Y., and Nguyen, P. Q. BKZ 2.0: Better lattice security estimates. In *Advances in Cryptology – ASIACRYPT 2011* (Dec. 2011), D. H. Lee and X. Wang, Eds., vol. 7073 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 1–20.

[34] Cheon, J. H., Han, K., Kim, A., Kim, M., and Song, Y. Bootstrapping for approximate homomorphic encryption. In *Advances in Cryptology – EUROCRYPT 2018, Part I* (Apr. / May 2018), J. B. Nielsen and V. Rijmen, Eds., vol. 10820 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 360–384.

[35] Cheon, J. H., Kim, A., Kim, M., and Song, Y. S. Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology – ASIACRYPT 2017, Part I* (Dec. 2017), T. Takagi and T. Peyrin, Eds., vol. 10624 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 409–437.

[36] Chillotti, I., Gama, N., Georgieva, M., and Izabachène, M. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In *Advances in Cryptology – ASIACRYPT 2016, Part I* (Dec. 2016), J. H. Cheon and T. Takagi, Eds., vol. 10031 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 3–33.

[37] Chor, B., Goldreich, O., Kushilevitz, E., and Sudan, M. Private information retrieval. In *36th Annual Symposium on Foundations of Computer Science* (Oct. 1995), IEEE Computer Society Press, pp. 41–50.

[38] Cohen, J. D., and Fischer, M. J. A robust and verifiable cryptographically secure election scheme (extended abstract). In *26th Annual Symposium on Foundations of Computer Science* (Oct. 1985), IEEE Computer Society Press, pp. 372–382.

[39] Cooley, J. W., and Tukey, J. W. An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation 19*, 90 (1965), 297–301.

[40] Costache, A., Smart, N. P., and Vivek, S. Faster homomorphic evaluation of discrete Fourier transforms. In *FC 2017: 21st International Conference on Financial Cryptography and Data Security* (Apr. 2017), A. Kiayias, Ed., vol. 10322 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 517–529.

[41] Cramer, R., Ducas, L., Peikert, C., and Regev, O. Recovering short generators of principal ideals in cyclotomic rings. In *Advances in Cryptology – EUROCRYPT 2016, Part II* (May 2016), M. Fischlin and J.-S. Coron, Eds., vol. 9666 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 559–585.

[42] Cramer, R., Ducas, L., and Wesolowski, B. Short Stickelberger class relations and application to ideal-SVP. In *Advances in Cryptology*

– *EUROCRYPT 2017, Part I* (Apr. / May 2017), J. Coron and J. B. Nielsen, Eds., vol. 10210 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 324–348.

[43] Crockett, E., and Peikert, C. $\Lambda \circ \lambda$ functional lattice cryptography. In *ACM CCS 16: 23rd Conference on Computer and Communications Security* (Oct. 2016), E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, Eds., ACM Press, pp. 993–1005.

[44] Curtmola, R., Garay, J. A., Kamara, S., and Ostrovsky, R. Searchable symmetric encryption: improved definitions and efficient constructions. In *ACM CCS 06: 13th Conference on Computer and Communications Security* (Oct. / Nov. 2006), A. Juels, R. N. Wright, and S. Vimercati, Eds., ACM Press, pp. 79–88.

[45] Damgård, I., Pastro, V., Smart, N. P., and Zakarias, S. Multiparty computation from somewhat homomorphic encryption. In *Advances in Cryptology – CRYPTO 2012* (Aug. 2012), R. Safavi-Naini and R. Canetti, Eds., vol. 7417 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 643–662.

[46] Doröz, Y., Hoffstein, J., Pipher, J., Silverman, J. H., Sunar, B., Whyte, W., and Zhang, Z. Fully homomorphic encryption from the finite field isomorphism problem. In *PKC 2018: 21st International Conference on Theory and Practice of Public Key Cryptography, Part I* (Mar. 2018), M. Abdalla and R. Dahab, Eds., vol. 10769 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 125–155.

[47] Dowlin, N., Gilad-Bachrach, R., Laine, K., Lauter, K. E., Naehrig, M., and Wernsing, J. Manual for using homomorphic encryption for bioinformatics. *Proceedings of the IEEE 105*, 3 (2017), 552–567.

[48] Ducas, L., and Durmus, A. Ring-LWE in polynomial rings. In *PKC 2012: 15th International Conference on Theory and Practice of Public Key Cryptography* (May 2012), M. Fischlin, J. Buchmann, and M. Manulis, Eds., vol. 7293 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 34–51.

[49] Ducas, L., and Micciancio, D. FHEW: Bootstrapping homomorphic encryption in less than a second. In *Advances in Cryptology – EUROCRYPT 2015, Part I* (Apr. 2015), E. Oswald and M. Fischlin, Eds., vol. 9056 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 617–640.

[50] Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *TCC 2006: 3rd Theory of Cryptography Conference* (Mar. 2006), S. Halevi and T. Rabin, Eds., vol. 3876 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 265–284.

[51] Eisenträger, K., Hallgren, S., and Lauter, K. E. Weak instances of PLWE. In *SAC 2014: 21st Annual International Workshop on Selected Areas in Cryptography* (Aug. 2014), A. Joux and A. M. Youssef, Eds., vol. 8781 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 183–194.

[52] ElGamal, T. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory 31* (1985), 469–472.

[53] Elias, Y., Lauter, K. E., Ozman, E., and Stange, K. E. Provably weak instances of ring-LWE. In *Advances in Cryptology – CRYPTO 2015, Part I* (Aug. 2015), R. Gennaro and M. J. B. Robshaw, Eds., vol. 9215 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 63–92.

[54] Fan, J., and Vercauteren, F. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144, 2012. `http://eprint.iacr.org/2012/144`.

[55] Fellows, M., and Koblitz, N. Combinatorial cryptosystems galore! *Finite Fields: Theory, Applications, and Algorithms 168* (1994), 51–61.

[56] Fröhlich, A., and Taylor, M. J. *Algebraic number theory*, vol. 27. Cambridge University Press, 1993.

[57] Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., and Waters, B. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual Symposium on Foundations of Computer Science* (Oct. 2013), IEEE Computer Society Press, pp. 40–49.

[58] Garner, H. L. The residue number system. In *Papers presented at the the March 3-5, 1959, western joint computer conference* (1959), ACM, pp. 146–153.

[59] Gennaro, R., Gentry, C., and Parno, B. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In *Advances in Cryptology – CRYPTO 2010* (Aug. 2010), T. Rabin, Ed., vol. 6223 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 465–482.

[60] Gentry, C. *A Fully Homomorphic Encryption Scheme*. PhD thesis, Stanford University, 2009.

[61] Gentry, C., and Halevi, S. Implementing Gentry's fully-homomorphic encryption scheme. In *Advances in Cryptology – EUROCRYPT 2011* (May 2011), K. G. Paterson, Ed., vol. 6632 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 129–148.

[62] Gentry, C., Halevi, S., and Smart, N. P. Better bootstrapping in fully homomorphic encryption. In *PKC 2012: 15th International Conference on Theory and Practice of Public Key Cryptography* (May 2012), M. Fischlin, J. Buchmann, and M. Manulis, Eds., vol. 7293 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 1–16.

[63] Gentry, C., Halevi, S., and Smart, N. P. Fully homomorphic encryption with polylog overhead. In *Advances in Cryptology – EUROCRYPT 2012* (Apr. 2012), D. Pointcheval and T. Johansson, Eds., vol. 7237 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 465–482.

[64] Gentry, C., Halevi, S., and Smart, N. P. Homomorphic evaluation of the AES circuit. In *Advances in Cryptology – CRYPTO 2012* (Aug. 2012), R. Safavi-Naini and R. Canetti, Eds., vol. 7417 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 850–867.

[65] Gentry, C., Sahai, A., and Waters, B. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology – CRYPTO 2013, Part I* (Aug. 2013), R. Canetti and J. A. Garay, Eds., vol. 8042 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 75–92.

[66] Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., and Wernsing, J. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning* (2016), pp. 201–210.

[67] Gorbunov, S., Vaikuntanathan, V., and Wichs, D. Leveled fully homomorphic signatures from standard lattices. In *47th Annual ACM Symposium on Theory of Computing* (June 2015), R. A. Servedio and R. Rubinfeld, Eds., ACM Press, pp. 469–477.

[68] Halevi, S. Homomorphic encryption. In *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich* (2017), Y. Lindell, Ed., Springer, pp. 219–276.

[69] Halevi, S., and Shoup, V. Bootstrapping for HElib. In *Advances in Cryptology – EUROCRYPT 2015, Part I* (Apr. 2015), E. Oswald and M. Fischlin, Eds., vol. 9056 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 641–670.

[70] Halevi, S., and Shoup, V. HElib: An implementation of homomorphic encryption (1.0.0 beta). `https://github.com/shaih/HElib`, Jan. 2019. IBM.

[71] HEAAN library (v2.1). `https://github.com/snucrypto/HEAAN`, Jan. 2019.

[72] Hirt, M., and Sako, K. Efficient receipt-free voting based on homomorphic encryption. In *Advances in Cryptology – EUROCRYPT 2000* (May 2000), B. Preneel, Ed., vol. 1807 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 539–556.

[73] Hoffstein, J., Pipher, J., and Silverman, J. H. NTRU: A ring-based public key cryptosystem. In *Algorithmic Number Theory, Third International Symposium, ANTS-III* (1998), J. Buhler, Ed., Springer, Heidelberg, pp. 267–288.

[74] iDASH: secure genome analysis competition. `http://www.humangenomeprivacy.org/2018/`.

[75] Jiang, X., Kim, M., Lauter, K. E., and Song, Y. Secure outsourced matrix computation and application to neural networks. In *ACM CCS 18: 25th Conference on Computer and Communications Security* (Oct. 2018), D. Lie, M. Mannan, M. Backes, and X. Wang, Eds., ACM Press, pp. 1209–1222.

[76] Karatsuba, A. A., and Ofman, Y. P. Multiplication of many-digital numbers by automatic computers. In *Doklady Akademii Nauk* (1962), vol. 145, Russian Academy of Sciences, pp. 293–294.

[77] Khedr, A., Gulak, G., and Vaikuntanathan, V. Shield: scalable homomorphic implementation of encrypted data-classifiers. *IEEE Transactions on Computers 65*, 9 (2016), 2848–2858.

[78] Kiayias, A., and Yung, M. The vector-ballot e-voting approach. In *FC 2004: 8th International Conference on Financial Cryptography* (Feb. 2004), A. Juels, Ed., vol. 3110 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 72–89.

[79] Kim, A., Song, Y., Kim, M., Lee, K., and Cheon, J. H. Logistic regression model training based on the approximate homomorphic encryption. *BMC medical genomics 11*, 4 (2018), 83.

[80] Koblitz, N., Menezes, A. J., Wu, Y.-H., and Zuccherato, R. J. *Algebraic Aspects of Cryptography.* Springer, Heidelberg, 1998.

[81] Kolesnikov, V., Kumaresan, R., Rosulek, M., and Trieu, N. Efficient batched oblivious PRF with applications to private set intersection. In *ACM CCS 16: 23rd Conference on Computer and Communications Security* (Oct. 2016), E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, Eds., ACM Press, pp. 818–829.

[82] Kushilevitz, E., and Ostrovsky, R. Replication is NOT needed: SINGLE database, computationally-private information retrieval. In *38th Annual Symposium on Foundations of Computer Science* (Oct. 1997), IEEE Computer Society Press, pp. 364–373.

[83] Langlois, A., and Stehlé, D. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptography 75*, 3 (June 2015), 565–599.

[84] Lenstra, A. K., Lenstra, H. W., and Lovász, L. Factoring polynomials with rational coefficients. *Mathematische Annalen 261*, 4 (1982), 515–534.

[85] Lepoint, T. FV-NFLlib: library implementing the Fan-Vercauteren homomorphic encryption scheme. `https://github.com/CryptoExperts/FV-NFLlib`, 2016.

[86] Levy-dit Vehel, F., Marinari, M. G., Perret, L., and Traverso, C. *A Survey on Polly Cracker Systems*. Springer, Heidelberg, 2009.

[87] Longa, P., and Naehrig, M. Speeding up the number theoretic transform for faster ideal lattice-based cryptography. In *CANS 16: 15th International Conference on Cryptology and Network Security* (Nov. 2016), S. Foresti and G. Persiano, Eds., vol. 10052 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 124–139.

[88] López-Alt, A., Tromer, E., and Vaikuntanathan, V. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *44th Annual ACM Symposium on Theory of Computing* (May 2012), H. J. Karloff and T. Pitassi, Eds., ACM Press, pp. 1219–1234.

[89] Lyubashevsky, V. Search to decision reduction for the learning with errors over rings problem. In *2011 IEEE Information Theory Workshop, ITW 2011* (October 2011), IEEE, pp. 410–414.

[90] Lyubashevsky, V., Peikert, C., and Regev, O. On ideal lattices and learning with errors over rings. In *Advances in Cryptology – EUROCRYPT 2010* (May / June 2010), H. Gilbert, Ed., vol. 6110 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 1–23. Full version of the paper available on https://eprint.iacr.org/2012/230.

[91] Lyubashevsky, V., Peikert, C., and Regev, O. On ideal lattices and learning with errors over rings. *J. ACM 60*, 6 (Nov. 2013), 43:1–43:35.

[92] Lyubashevsky, V., Peikert, C., and Regev, O. A toolkit for ring-LWE cryptography. In *Advances in Cryptology – EUROCRYPT 2013* (May 2013), T. Johansson and P. Q. Nguyen, Eds., vol. 7881 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 35–54.

[93] Mahadev, U. Classical verification of quantum computations. In *59th Annual Symposium on Foundations of Computer Science* (Oct. 2018), M. Thorup, Ed., IEEE Computer Society Press, pp. 259–267.

[94] Miroslav, V. Vznik kodu a ciselne soustavy zbytkovych trid. *Stroje Na Zpracovani Informaci 3* (1955).

[95] Mukherjee, P., and Wichs, D. Two round multiparty computation via multi-key FHE. In *Advances in Cryptology – EUROCRYPT 2016, Part II* (May 2016), M. Fischlin and J.-S. Coron, Eds., vol. 9666 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 735–763.

[96] Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology – EUROCRYPT'99* (May 1999), J. Stern, Ed., vol. 1592 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 223–238.

[97] Pedrouzo-Ulloa, A., Troncoso-Pastoriza, J. R., and Pérez-González, F. Multivariate lattices for encrypted image processing. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2015), IEEE, pp. 1707–1711.

[98] Peikert, C. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *41st Annual ACM Symposium on Theory of Computing* (May / June 2009), M. Mitzenmacher, Ed., ACM Press, pp. 333–342.

[99] Peikert, C. How (not) to instantiate ring-LWE. In *SCN 16: 10th International Conference on Security in Communication Networks* (Aug. / Sept. 2016), V. Zikas and R. De Prisco, Eds., vol. 9841 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 411–430.

[100] Peikert, C., Regev, O., and Stephens-Davidowitz, N. Pseudorandomness of ring-LWE for any ring and modulus. In *49th Annual ACM Symposium on Theory of Computing* (June 2017), H. Hatami, P. McKenzie, and V. King, Eds., ACM Press, pp. 461–473.

[101] Peikert, C., and Shiehian, S. Multi-key FHE from LWE, revisited. In *TCC 2016-B: 14th Theory of Cryptography Conference, Part II* (Oct. / Nov. 2016), M. Hirt and A. D. Smith, Eds., vol. 9986 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 217–238.

[102] Pellet-Mary, A., Hanrot, G., and Stehlé, D. Approx-SVP in ideal lattices with pre-processing. In *Advances in Cryptology – EUROCRYPT 2019* (May 2019), Lecture Notes in Computer Science, Springer, Heidelberg. to appear.

[103] Pinkas, B., Schneider, T., and Zohner, M. Scalable private set intersection based on OT extension. *ACM Transactions on Privacy and Security (TOPS) 21*, 2 (2018), 7.

[104] Polyakov, Y., Rohloff, K., and Ryan, G. W. Palisade: Lattice cryptography library. `https://git.njit.edu/palisade/PALISADE`, 2017.

[105] Regev, O. On lattices, learning with errors, random linear codes, and cryptography. In *37th Annual ACM Symposium on Theory of Computing* (May 2005), H. N. Gabow and R. Fagin, Eds., ACM Press, pp. 84–93.

[106] Rivest, R. L., Adleman, L., and Dertouzos, M. L. On data banks and privacy homomorphisms. *Foundations of secure computation 4*, 11 (1978), 169–180.

[107] Rivest, R. L., Shamir, A., and Adleman, L. M. A method for obtaining digital signature and public-key cryptosystems. *Communications of the Association for Computing Machinery 21*, 2 (1978), 120–126.

[108] Sander, T., Young, A., and Yung, M. Non-interactive cryptocomputing for NC1. In *40th Annual Symposium on Foundations of Computer Science* (Oct. 1999), IEEE Computer Society Press, pp. 554–567.

[109] Schnorr, C.-P., and Euchner, M. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical programming 66*, 1-3 (1994), 181–199.

[110] Schönhage, A., and Strassen, V. Schnelle multiplikation grosser zahlen. *Computing 7*, 3-4 (1971), 281–292.

[111] Simple Encrypted Arithmetic Library (release 3.1.0). `https://github.com/Microsoft/SEAL`, Dec. 2018. Microsoft Research, Redmond, WA.

[112] Smart, N. P., and Vercauteren, F. Fully homomorphic encryption with relatively small key and ciphertext sizes. In *PKC 2010: 13th International Conference on Theory and Practice of Public Key Cryptography* (May 2010), P. Q. Nguyen and D. Pointcheval, Eds., vol. 6056 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 420–443.

[113] Smart, N. P., and Vercauteren, F. Fully homomorphic SIMD operations. *Des. Codes Cryptography 71*, 1 (Apr. 2014), 57–81.

[114] Song, D. X., Wagner, D., and Perrig, A. Practical techniques for searches on encrypted data. In *2000 IEEE Symposium on Security and Privacy* (May 2000), IEEE Computer Society Press, pp. 44–55.

[115] van Dijk, M., Gentry, C., Halevi, S., and Vaikuntanathan, V. Fully homomorphic encryption over the integers. In *Advances in Cryptology – EUROCRYPT 2010* (May / June 2010), H. Gilbert, Ed., vol. 6110 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 24–43.

[116] Yao, A. C.-C. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science* (Nov. 1982), IEEE Computer Society Press, pp. 160–164.

[117] Yi, X., Kaosar, M. G., Paulet, R., and Bertino, E. Single-database private information retrieval from fully homomorphic encryption. *IEEE Transactions on Knowledge and Data Engineering 25*, 5 (2013), 1125–1134.

Part II

Publications

Chapter 6

Provably Weak Instances of Ring-LWE Revisited.

Publication data

Castryck, W., Iliashenko, I., and Vercauteren, F. Provably weak instances of Ring-LWE revisited. In *Advances in Cryptology – EUROCRYPT 2016, Part I* (May 2016), M. Fischlin and J.-S. Coron, Eds., vol. 9665 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 147–167.

Provably Weak Instances of Ring-LWE Revisited

Wouter Castryck[1,2], Ilia Iliashenko[1], and Frederik Vercauteren[1]

[1]KU Leuven ESAT/COSIC and iMinds
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
firstname.lastname@esat.kuleuven.be

[2]Vakgroep Wiskunde, Universiteit Gent
Krijgslaan 281/S22, B-9000 Gent, Belgium

Abstract. In CRYPTO 2015, Elias, Lauter, Ozman and Stange described an attack on the non-dual *decision* version of the ring learning with errors problem (RLWE) for two special families of defining polynomials, whose construction *depends on the modulus* q that is being used. For particularly chosen error parameters, they managed to solve non-dual decision RLWE given 20 samples, with a success rate ranging from 10% to 80%. In this paper we show how to solve the *search* version for the same families and error parameters, using only 7 samples with a success rate of 100%. Moreover our attack works for *every modulus* q' instead of the q that was used to construct the defining polynomial. The attack is based on the observation that the RLWE error distribution for these families of polynomials is very skewed in the directions of the polynomial basis. For the parameters chosen by Elias et al. the smallest errors are negligible and simple linear algebra suffices to recover the secret. But enlarging the error paremeters makes the largest errors wrap around, thereby turning the RLWE problem unsuitable for cryptographic applications. These observations also apply to dual RLWE, but do not contradict the seminal work by Lyubashevsky, Peikert and Regev.

1 Introduction

Hard problems on lattices have become popular building blocks for cryptographic primitives mainly because of two reasons: firstly, lattice based cryptography appears to remain secure even in the presence of quantum computers, and secondly, the security of the primitives can be based on worst-case hardness assumptions. Although it seems appealing to use classical hard lattice problems such as the shortest vector problem or closest vector problem for cryptographic applications, the learning with errors problem (LWE) has proven much more versatile. This problem was introduced by Regev [12, 13] who showed that an efficient algorithm for LWE results in efficient quantum algorithms for approximate lattice problems. The *decision* version of LWE can be defined informally as the problem of distinguishing noisy linear equations from truly random ones. More precisely, let $n \geq 1$ be an integer dimension and $q \geq 2$ an integer modulus, then the problem is to distinguish polynomially many pairs of the form $(\mathbf{a}_i, b_i \approx \langle \mathbf{a}_i, \mathbf{s} \rangle)$ from uniformly random and independent pairs. The vectors $\mathbf{a}_i$ are chosen uniformly

random in $\mathbb{Z}_q^n$, the vector $\mathbf{s}$ is secret and the same for all pairs, and the element b_i is computed as $b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i$ where e_i is a random error term drawn from an error distribution on $\mathbb{Z}_q$, such as a discretized Gaussian. The *search* version of LWE asks to recover the secret vector $\mathbf{s}$. The hardness of the LWE problem has been analyzed in [12, 13, 11, 8, 3].

The main downside of LWE is that it is not very practical, basically due to the fact that each new $\mathbf{a}_i$ only gives rise to one element b_i (and not a vector of n elements as one could hope). The result is that the public key size and the computation time of LWE-based cryptosystems are typically quadratic in the security parameter. Lyubashevsky, Peikert and Regev [9] solved this issue by introducing the Ring-LWE (RLWE) problem and showing its hardness under wost-case assumptions on ideal lattices. Its flavour is distantly similar to that of NTRU [7]. Informally, the secret key space $\mathbb{Z}_q^n$ is replaced by $R_q = R/qR$ where R is the ring of integers in a number field $K = \mathbb{Q}[x]/(f)$ with f a monic irreducible integral polynomial of degree n and $q \geq 2$ an integer modulus. The inner product on $\mathbb{Z}_q^n$ is replaced by the ring product in R_q. In its *non-dual* form the *decision* version of RLWE is then roughly defined as follows: distinguish polynomially many samples of the form $(\mathbf{a}_i, \mathbf{b}_i \approx \mathbf{a}_i \cdot \mathbf{s})$ from uniformly random and independent pairs. Here the $\mathbf{a}_i \in R_q$ are uniformly random and independent, $\mathbf{s} \in R_q$ is a fixed random secret, and $\mathbf{b}_i$ is computed as $\mathbf{b}_i = \mathbf{a}_i \cdot \mathbf{s} + \mathbf{e}_i$ where $\mathbf{e}_i \in R_q$ is a short random error term that is drawn from a specific error distribution ψ on R_q. The *search* version of the problem is to recover the secret $\mathbf{s}$ from the list of samples. We stress that the actual problem described and analyzed in [9] is the *dual* RLWE problem, in which the secret and the error term are taken from the reduction modulo q of a certain fractional ideal of K, denoted by $R_q^\vee$; see Section 2 for more details.

As explained in [9], the search and decision problems are equivalent when K is Galois and q is a prime number that splits into prime ideals with small norm (polynomial in n). In general, no such reduction is known and it is easy to see that search RLWE is at least as hard as decision RLWE.

The definition of the error distribution ψ on R_q (or on $R_q^\vee$) plays a crucial role in RLWE and is obtained by pulling back a near-spherical Gaussian distribution under the canonical embedding of the number field. An alternative problem [5] is called Polynomial-LWE (PLWE) and uses an error distribution on R_q where each coordinate of the error term with respect to the polynomial basis $1, x, x^2, \ldots, x^{n-1}$ is drawn independently from a fixed one-dimensional Gaussian distribution. Again we refer to Section 2 for more details.

In [5], Eisentraeger, Hallgren and Lauter presented families of defining polynomials $f \in \mathbb{Z}[x]$ and moduli q such that the *decision* version of PLWE is weak. The attack can be described in a nutshell as follows: assume that $f(1) \equiv 0 \bmod q$, then evaluation at 1 defines a ring homomorphism ϕ from R_q to $\mathbb{Z}_q$. Applying ϕ to the PLWE samples results in equations of the form $\mathbf{a}_i(1) \cdot \mathbf{s}(1) + \mathbf{e}_i(1) = \mathbf{b}_i(1)$. Therefore, if the images $\mathbf{e}_i(1)$ of the error terms can be distinguished from uniform, one can simply loop through all possibilities for $\mathbf{s}(1) \in \mathbb{Z}_q$ and determine if the corresponding $\mathbf{e}_i(1)$ are uniform on $\mathbb{Z}_q$ or not. So as long as q is small

enough (such that one can exhaustively run through $\mathbb{Z}_q$), $f(1) \equiv 0 \bmod q$, and the images $\mathbf{e}_i(1)$ do not wrap around too much modulo q, this attack breaks decision PLWE.

In [6], Elias, Lauter, Ozman and Stange extended this attack to the *decision* version of non-dual RLWE, rather than PLWE, by showing that for defining polynomials of the form

$$f_{n,a,b} = x^n + ax + b \in \mathbb{Z}[x]$$

where n, a, b are specifically chosen parameters such that i.a. $f_{n,a,b}(1) \equiv 0 \bmod q$, the distortion introduced by pulling back the Gaussian error terms through the canonical embedding is small enough such that the attack on PLWE still applies. This attack was executed for three parameter sets n, a, b, r where given 20 samples, non-dual decision RLWE could be solved with success rates ranging from 10% to 80% depending on the particular family considered [6, Section 9]. Here the parameter r determines the width of the Gaussian that is being pulled back, which Elias et al. chose to be spherical.

Our contributions in this paper are as follows. Firstly, we explain how to solve the *search* version of non-dual RLWE, which one might expect to be a harder problem than the decision version (due to the fact that the corresponding number fields are not Galois), for the same parameter sets, using only 7 samples with a success rate of 100%. The attack invokes simple linear algebra to recover the secret element $\mathbf{s}$ and does not use that $f_{n,a,b}(1) \equiv 0 \bmod q$: in fact, for the same defining polynomial and the same error parameter r our attack works for *every* modulus q'. Secondly, we show that if one tries to adjust r in order to obtain a hard instance of non-dual RLWE, the first few components of the noise wrap around modulo q and become indistinguishable from uniform, thereby obstructing certain cryptographic applications. Thirdly, we show that our observations also apply to the *dual* RLWE problem when set up for the same number fields: either the errors wrap around or linear algebra can be used to reveal the secret. The latter situation only occurs for error widths that are way too small for the hardness results of Lyubashekvsy, Peikert and Regev [9] to be applicable. Therefore neither the results from [6] nor our present attack seem to form a threat on RLWE, at least when set up along the guidelines in [9, 10].

Our observations are easiest to explain for $a = 0$, a case which covers two of the three parameter sets. From $f_{n,a,b}(1) = b + 1 \equiv 0 \bmod q$ and $f_{n,a,b}(1) \neq 0$ (by irreducibility) it follows that the roots of $f_{n,a,b}$ lie on a circle with radius

$$\rho \geq \sqrt[n]{q-1} > 1.$$

With respect to the polynomial basis $1, x, x^2, \ldots, x^{n-1}$, the canonical embedding matrix is essentially the Vandermonde matrix generated by these roots, whose column norms grow geometrically as

$$\sqrt{n}, \sqrt{n}\rho, \ldots, \sqrt{n}\rho^{n-1}.$$

This simple observation has major implications for the distortion introduced by the inverse of the canonical embedding: the distribution of the error terms will

be extremely stretched at the terms of low degree, whereas they will be squashed at the terms of high degree. For the parameter sets attacked by Elias et al., the latter are so small that after rounding they simply become zero, thereby resulting in exact linear equations in the coefficients of the secret element $\mathbf{s}$. Given enough samples (in the cases considered, between 4 and 7 samples suffice), the secret $\mathbf{s}$ can be recovered using elementary linear algebra. Furthermore, since the ratio between the maximal and the minimal distortion is roughly $\rho^n \geq q - 1$, it is impossible to increase the width of the Gaussians used without causing the errors at the terms of low degree to wrap around modulo q.

The remainder of the paper is organized as follows: in Section 2 we recall the definition of PLWE and of dual and non-dual RLWE, with particular focus on the error distributions involved. Section 3 reviews the attacks on decision PLWE by Eisentraeger, Hallgren and Lauter and non-dual decision RLWE by Elias, Lauter, Ozman and Stange. Section 4 describes our attack on non-dual search RLWE by analyzing the singular value decomposition of the canonical embedding. We also report on an implementation of our attack in Magma [2], which shows that we can indeed easily break the families considered in [6] using less samples, with a higher success probability, and for every choice of modulus q' (instead of just the q that was used to define $f_{n,a,b}$). We also discuss how switching to dual RLWE affects these observations. In Section 5 we study the effect of increasing the error parameter as an attempt to counter our attack, and compare with the hardness results from [9]. Section 6 concludes the paper.

2 Preliminaries

In this section we briefly recall the necessary background on number fields, the canonical embedding and Gaussian distributions to give proper definitions of PLWE and dual and non-dual RLWE.

2.1 Number fields and the canonical embedding

Let $f \in \mathbb{Z}[x]$ be a monic irreducible polynomial of degree n and consider the number field $K = \mathbb{Q}[x]/(f)$ it defines. Let $R \subset K$ denote the ring of integers of K, i.e. the set of all algebraic integers that are contained in K. If f can be taken such that $R = \mathbb{Z}[x]/(f)$, then K is called a *monogenic* number field and f a monogenic polynomial.

The field K has exactly n embeddings into $\mathbb{C}$ denoted by $\sigma_i : K \to \mathbb{C}$ for $i = 1, \ldots, n$. These n embeddings correspond precisely to evaluation in each of the n distinct roots α_i of f, i.e. an element $a(x) \in K$ is mapped to $\sigma_i(a(x)) = a(\alpha_i) \in \mathbb{C}$. Assume that f has s_1 real roots and $n - s_1 = 2s_2$ complex conjugate roots and order the roots such that $\overline{\alpha_{s_1+k}} = \alpha_{s_1+s_2+k}$ for $k = 1, \ldots, s_2$. The *canonical embedding* (also known as the Minkowski embedding) $\sigma : K \to \mathbb{C}^n$ is then defined as:

$$\sigma(a) = (\sigma_1(a), \ldots, \sigma_{s_1}(a), \sigma_{s_1+1}(a), \ldots, \sigma_{s_1+s_2}(a), \overline{\sigma_{s_1+1}}(a), \ldots, \overline{\sigma_{s_1+s_2}}(a)) .$$

It is easy to see that the canonical embedding maps into the space $H \subset \mathbb{R}^{s_1} \times \mathbb{C}^{2s_2}$ given by

$$H = \{(x_1, \ldots, x_n) \in \mathbb{R}^{s_1} \times \mathbb{C}^{2s_2} : \overline{x_{s_1+j}} = x_{s_1+s_2+j}\,, \forall j \in [1 \ldots s_2]\}\,.$$

The space H is isomorphic to $\mathbb{R}^n$ as an inner product space by considering the orthonormal basis for H given by the columns of

$$B = \begin{pmatrix} I_{s_1 \times s_1} & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} I_{s_2 \times s_2} & \frac{\mathbf{i}}{\sqrt{2}} I_{s_2 \times s_2} \\ 0 & \frac{1}{\sqrt{2}} I_{s_2 \times s_2} & -\frac{\mathbf{i}}{\sqrt{2}} I_{s_2 \times s_2} \end{pmatrix}\,.$$

With respect to this basis, the coordinates of $\sigma(a)$ are given by a real vector

$$(\tilde{a}_1, \ldots, \tilde{a}_n) := (\sigma_1(a), \ldots, \sigma_{s_1}(a), \sqrt{2}\,\mathfrak{Re}(\sigma_{s_1+1}(a)), \ldots, \sqrt{2}\,\mathfrak{Re}(\sigma_{s_1+s_2}(a)), \\ \sqrt{2}\,\mathfrak{Im}(\sigma_{s_1+1}(a)), \ldots, \sqrt{2}\,\mathfrak{Im}(\sigma_{s_1+s_2}(a))).$$

Note that in [6] the authors did not include the factor $\sqrt{2}$, but we choose to keep it since it makes B unitary.

In summary, an element $a(x) \in K$ can be represented in the polynomial basis as $(a_0, \ldots, a_{n-1})$ where $a(x) = \sum_{i=0}^{n-1} a_i x^i$ but also by a real vector $(\tilde{a}_1, \ldots, \tilde{a}_n)$ where the canonical embedding of a is given by:

$$\sigma(a) = B \cdot (\tilde{a}_1, \ldots, \tilde{a}_n)^t\,.$$

Let M_f denote the Vandermonde matrix $(\alpha_i^{j-1})_{i,j}$ for $i, j = 1, \ldots, n$, then the polynomial basis representation is related to the (real) canonical embedding representation by the following transformation

$$(a_0, \ldots, a_{n-1})^t = M_f^{-1} \cdot B \cdot (\tilde{a}_1, \ldots, \tilde{a}_n)^t\,.$$

Since M_f^{-1} will play a crucial role in the following, we denote it with N_f. Later on, to ease notation we will just write M_f instead of $M_{f_{n,a,b}}$, and similarly for N_f.

2.2 Ideals of the ring of integers and their dual

An *integral ideal* $I \subseteq R$ is an additive subgroup of R closed under multiplication by elements of R, i.e. $rI \subset I$ for any $r \in R$. A *fractional ideal* $I \subset K$ is a set such that $dI \subseteq R$ is an integral ideal for some $d \in R$. A *principal* (fractional or integral) ideal I is one that is generated by some $u \in K$, i.e. $I = uR$; we denote it as $I = \langle u \rangle$. The sum $I + J$ of two (fractional or integral) ideals is the set of all $x + y$ with $x \in I, y \in J$ and the product $I \cdot J$ is the smallest (fractional or integral) ideal containing all products $x \cdot y$ with $x \in I, y \in J$. The set of non-zero fractional ideals forms a group under multiplication; this is not true for integral ideals. The inverse of a non-zero fractional ideal is denoted by I^{-1}. Every fractional ideal I is a free $\mathbb{Z}$-module of rank n, and therefore $I \otimes \mathbb{Q} = K$.

Its image $\sigma(I)$ under the canonical embedding is a lattice of rank n inside the space H.

The trace $\mathrm{Tr} = \mathrm{Tr}_{K/\mathbb{Q}} : K \to \mathbb{Q}$ maps an element x to the sum of its embeddings $\mathrm{Tr}(x) = \sum_{i=1}^{n} \sigma_i(x)$ and defines an additive homomorphism from R to $\mathbb{Z}$. The norm $\mathrm{No} = \mathrm{No}_{K/\mathbb{Q}} : K \to \mathbb{Q}$ takes the product of all embeddings $\mathrm{No}(x) = \prod_{i=1}^{n} \sigma_i(x)$ and is multiplicative.

For a fractional ideal I, its dual $I^\vee$ is defined as

$$I^\vee = \{x \in K : \mathrm{Tr}(xI) \subseteq \mathbb{Z}\}.$$

It is easy to see that $(I^\vee)^\vee = I$ and that $I^\vee$ is also a fractional ideal. (Under the canonical embedding, this corresponds to the usual notion of dual lattice, modulo complex conjugation.) Furthermore, for any fractional ideal I, its dual is $I^\vee = I^{-1}R^\vee$. The factor $R^\vee$ is a fractional ideal called the *codifferent* and its inverse is called the *different ideal* which is integral. For a monogenic defining polynomial f, i.e. $R = \mathbb{Z}[x]/(f)$ we have that $R^\vee = \langle 1/f'(\alpha) \rangle$ where α is a root of f. Applying this fact to the cyclotomic number field of degree $n = 2^k$ with defining polynomial $f(x) = x^n + 1$, we get that $f'(\xi_{2n}) = n\xi_{2n}^{n-1}$ with ξ_{2n} a primitive $2n$-th root of unity. Thus $R^\vee = \langle n^{-1} \rangle$, since ξ_{2n}^{n-1} is a unit.

2.3 Gaussian distributions and discretization

Denote by Γ_r the normal Gaussian distribution on $\mathbb{R}$ with mean 0 and parameter r given by $\Gamma_r(x) = r^{-1}\exp(-\pi x^2/r^2)$. Note that we have $r = \sqrt{2\pi}\rho$ with ρ the standard deviation. We can define an elliptical Gaussian distribution $\Gamma_{\mathbf{r}}$ on H as follows: let $\mathbf{r} = (r_1, \ldots, r_n) \in (\mathbb{R}^+)^n$ be a vector of n positive real numbers, then a sample of $\Gamma_{\mathbf{r}}$ is given by $B \cdot (x_1, \ldots, x_n)^t$ where each x_i is sampled independently from Γ_{r_i} on $\mathbb{R}$. Note that via the inverse of the canonical embedding this also defines a distribution $\Psi_{\mathbf{r}}$ on $K \otimes \mathbb{R}$, in other words

$$N_f \cdot B \cdot (x_1, \ldots, x_n)^t$$

gives us the coordinates of $\Gamma_{\mathbf{r}} \leftarrow (x_1, \ldots, x_n)$ with respect to the polynomial basis $1, x, x^2, \ldots, x^{n-1}$.

In practice we sample from the continuous distribution $\Gamma_{\mathbf{r}}$ modulo some finite but sufficiently high precision (e.g. using the Box-Muller method). In particular our samples live over $\mathbb{Q}$ rather than $\mathbb{R}$, so that an element sampled from $\Psi_{\mathbf{r}}$ can be truly seen as an element of the field K. For use in RLWE one even wants to draw elements from I for some fixed fractional ideal $I \subset K$, where $I = R$ (non-dual RLWE) and $I = R^\vee$ (dual RLWE) are the main examples. In this case one should discretize the Gaussian distribution $\Gamma_{\mathbf{r}}$ to the lattice $\sigma(I)$. There are several ways of doing this, e.g. by rounding coordinates with respect to some given $\mathbb{Z}$-module basis; see [10, 9] and the references therein. But for our conclusions this discretization is not relevant, and because it would needlessly complicate things we will just omit it.

2.4 The Polynomial-LWE and Ring-LWE problem

In this section we provide formal definitions of PLWE [5] and RLWE [9, 4], both in its dual and its non-dual version [6]. We stress that it is the dual version of RLWE that was introduced in [9] and for which certain hardness results are available, one of which is recalled in Theorem 1 below.

Let $f \in \mathbb{Z}[x]$ be a monic irreducible polynomial of degree n and let $q \geq 2$ be an integer modulus. Consider the quotient ring $P = \mathbb{Z}[x]/(f)$ and denote with P_q the residue ring P/qP. Denote with Γ_r^n the spherical Gaussian on $\mathbb{R}^n$ with parameter r and interpret this as a distribution on $P \otimes \mathbb{R}$ by mapping the standard basis of $\mathbb{R}^n$ to the polynomial basis $1, x, x^2, \ldots, x^{n-1}$ of P. In particular, elements $\mathbf{e}(x) = \sum_{i=0}^{n-1} e_i x^i \leftarrow \Gamma_r^n$ have each coefficient e_i drawn independently from Γ_r. Let $\mathfrak{U}(P_q)$ denote the uniform distribution on P_q and let $\mathfrak{U}(P_{q,\mathbb{R}})$ be the uniform distribution on the torus $P_{q,\mathbb{R}} = (P \otimes \mathbb{R})/qP$.

With these ingredients we can define the decision and search PLWE problems.

Definition 1 (PLWE distribution). *For $\mathbf{s}(x) \in P_q$ and $r \in \mathbb{R}^+$, a sample from the* PLWE *distribution $A_{\mathbf{s}(x),r}$ over $P_q \times P_{q,\mathbb{R}}$ is generated by choosing $\mathbf{a}(x) \leftarrow \mathfrak{U}(P_q)$, choosing $\mathbf{e}(x) \leftarrow \Gamma_r^n$ and outputting $(\mathbf{a}(x), \mathbf{b}(x) = \mathbf{a}(x) \cdot \mathbf{s}(x) + \mathbf{e}(x) \bmod qP)$.*

Definition 2 (Decision PLWE). *The decision* PLWE *problem is to distinguish, for a random but fixed choice of $\mathbf{s}(x) \leftarrow \mathfrak{U}(P_q)$, with non-negligible advantage between arbitrarily many independent samples from $A_{\mathbf{s}(x),r}$ and the same number of independent samples from $\mathfrak{U}(P_q) \times \mathfrak{U}(P_{q,\mathbb{R}})$.*

Definition 3 (Search PLWE). *For a random but fixed choice of $\mathbf{s}(x) \leftarrow \mathfrak{U}(P_q)$, the search* PLWE *problem is to recover $\mathbf{s}(x)$ with non-negligible probability from arbitrarily many independent samples from $A_{\mathbf{s}(x),r}$.*

To define the dual and non-dual RLWE problems we require a degree n number field K with ring of integers R. We also fix a fractional ideal $I \subset K$, for which two choices are available: in the *dual* RLWE problems we let $I = R^\vee$, while in the *non-dual* RLWE problems we take $I = R$. Note that $I \otimes \mathbb{R} = K \otimes \mathbb{R}$, so we can view the distribution $\Psi_{\mathbf{r}}$ from the previous section as a distribution on $I \otimes \mathbb{R}$. We let I_q denote I/qI and write $I_{q,\mathbb{R}}$ for the torus $(I \otimes \mathbb{R})/qI$. As before we let $\mathfrak{U}(I_q)$ denote the uniform distribution on I_q and let $\mathfrak{U}(I_{q,\mathbb{R}})$ be the uniform distribution on $I_{q,\mathbb{R}}$.

Definition 4 (RLWE distribution). *For $\mathbf{s}(x) \in I_q$ and $\mathbf{r} \in (\mathbb{R}^+)^n$, a sample from the* RLWE *distribution $A_{\mathbf{s}(x),\mathbf{r}}$ over $R_q \times I_{q,\mathbb{R}}$ is generated by choosing $\mathbf{a}(x) \leftarrow \mathfrak{U}(R_q)$, choosing $\mathbf{e}(x) \leftarrow \Psi_{\mathbf{r}}$ and returning $(\mathbf{a}(x), \mathbf{b}(x) = \mathbf{a}(x) \cdot \mathbf{s}(x) + \mathbf{e}(x) \bmod qI)$.*

Definition 5 (Decision RLWE). *The decision* RLWE *problem is to distinguish, for a random but fixed choice of $\mathbf{s}(x) \leftarrow \mathfrak{U}(I_q)$, with non-negligible advantage between arbitrarily many independent samples from $A_{\mathbf{s}(x),\mathbf{r}}$ and the same number of independent samples from $\mathfrak{U}(R_q) \times \mathfrak{U}(I_{q,\mathbb{R}})$.*

Definition 6 (Search RLWE). *For a random but fixed choice of* $\mathbf{s}(x) \leftarrow \mathfrak{U}(I_q)$*, the search* RLWE *problem is to recover* $\mathbf{s}(x)$ *with non-negligible probability from arbitrarily many independent samples from* $A_{\mathbf{s}(x),\mathbf{r}}$*.*

A hardness statement on the search RLWE problem in its dual form (i.e. with $I = R^\vee$) was provided by Lyubashevsky, Peikert and Regev. For proof-technical reasons their result actually deals with a slight variant called the search $\mathrm{RLWE}_{\leq r}$ problem, where $r \in \mathbb{R}^+$. In this variant each sample is taken from $A_{\mathbf{s}(x),\mathbf{r}}$ for a new choice of $\mathbf{r}$, chosen uniformly at random from $\{(r_1, \ldots, r_n) \in (\mathbb{R}^+)^n \mid r_i \leq r \text{ for all } i\}$. Think of this parameter r and the modulus $q \geq 2$ as quantities that vary with n, and let ω be a superlinear function. Then Lyubashevsky et al. proved:

Theorem 1 ([9, Theorem 4.1]). *If* $r \geq 2\omega(\sqrt{\log n})$ *then for some negligible* ε *(depending on* n*) there is a probabilistic polynomial-time quantum reduction from* KDGS_γ *to* $\mathrm{RLWE}_{\leq r}$*, where*

$$\gamma : I \mapsto \max\left\{\eta_\varepsilon(I) \cdot (\sqrt{2}q/r) \cdot \omega(\sqrt{\log n}), \sqrt{2n}/\lambda_1(I^\vee)\right\}.$$

Here $\eta_\varepsilon(I)$ *is the smoothing parameter of* $\sigma(I)$ *with threshold* ε*, and* $\lambda_1(I^\vee)$ *is the length of a shortest vector of* $\sigma(I^\vee)$*.*

In the above statement KDGS_γ refers to the *discrete Gaussian sampling problem*, which is about producing samples from a spherical Gaussian in H with parameter r', discretized to the lattice $\sigma(I)$, for any given non-zero ideal $I \subset R$ and any $r' \geq \gamma(I)$. As discussed in [9] there are easy reductions from certain standard lattice problems to the discrete Gaussian sampling problem.

As an intermediate step in their proof Lyubashevsky et al. obtain a classical (i.e. non-quantum) reduction from an instance of the bounded distance decoding problem in ideal lattices to $\mathrm{RLWE}_{\leq r}$; see [9, Lemma 4.5].

In contrast, Elias, Lauter, Ozman and Stange [6] study RLWE in its non-dual version, and for the sake of comparison our main focus will also be on that setting, i.e. we will mostly take $I = R$. In Section 4.3 we will look at the effect of switching to the dual case where $I = R^\vee$, and in Section 5 we will include the above hardness result in the discussion. Moreover, again as in [6], the noise parameter $\mathbf{r} = (r_1, \ldots, r_n)$ will usually be taken fixed and spherical, i.e. $r_1 = \cdots = r_n = r$.

3 Provably weak instances of non-dual decision RLWE

In [5], Eisentraeger, Hallgren and Lauter presented families of defining polynomials $f \in \mathbb{Z}[x]$ such that the *decision* version of PLWE is weak. This attack was later extended to non-dual decision RLWE [6] by Elias, Lauter, Ozman and Stange. In this section we recall the attack, first for PLWE and then how it transfers to non-dual RLWE. We provide a detailed analysis of the singular value decomposition of the matrix N_f for these polynomial families, since this will play an instructive role in our exposition.

3.1 Attack on decision PLWE

The simplest form of the attack on decision PLWE requires that the defining polynomial f of P and the modulus q satisfy the relation $f(1) \equiv 0 \bmod q$. This implies that evaluation at 1 induces a ring homomorphism $\phi : P_q \to \mathbb{Z}_q : a(x) \mapsto a(1) \bmod q$. By applying ϕ to the PLWE samples $(\mathbf{a}_i, \mathbf{b}_i = \mathbf{a}_i \cdot \mathbf{s} + \mathbf{e}_i)$ we obtain tuples in $\mathbb{Z}_q^2$ namely $(\phi(\mathbf{a}_i), \phi(\mathbf{a}_i) \cdot \phi(\mathbf{s}) + \phi(\lfloor \mathbf{e}_i \rceil))$. Here $\lfloor \mathbf{e}_i \rceil$ denotes the polynomial obtained by rounding each coefficient of $\mathbf{e}_i$ to the nearest integer (with ties broken upward, say).

Assuming that the images of the error terms $\mathbf{e}_i$ under the homomorphism ϕ can be distinguished from uniform with sufficiently high probability, one obtains the following straightforward attack: for each guess $s \in \mathbb{Z}_q$ for the value of $\phi(\mathbf{s}) = \mathbf{s}(1) \bmod q$, compute the corresponding image of the (rounded) error term $\phi(\lfloor \mathbf{e}_i \rceil)$ as $\phi(\lfloor \mathbf{b}_i \rceil) - \phi(\mathbf{a}_i)s$, assuming that the guess is correct. If there exists an s such that the corresponding images $\phi(\lfloor \mathbf{e}_i \rceil)$ are more or less distributed like a discretized Gaussian, rather than uniform, the samples were indeed likely to be actual PLWE samples and the secret $\mathbf{s}$ satisfies $\mathbf{s}(1) = s$. If no such guess is found, the samples were likely to be uniform samples. The attack succeeds if the following three conditions are met:

1. $f(1) \equiv 0 \bmod q$,
2. q is small enough that $\mathbb{Z}_q$ can be enumerated,
3. $\phi(\lfloor \Gamma_r^n \rceil)$ is distinguishable from uniform $\mathfrak{U}(\mathbb{Z}_q)$.

Note that if $\mathbf{e}_i$ is sampled from Γ_r^n, then the $\mathbf{e}_i(1)$ are also Gaussian distributed but with parameter $\sqrt{n} \cdot r$. Therefore, as long as $\sqrt{n} \cdot r$ is sufficiently smaller than q, it should be possible to distinguish $\phi(\lfloor \Gamma_r^n \rceil)$ from uniform.

3.2 Attack on non-dual decision RLWE

The attack of Elias et al. on non-dual decision RLWE basically works by interpreting the RLWE samples as PLWE samples and then executing the above attack. For this approach to work, two requirements need to be fulfilled. Firstly, the ring of integers R of the number field K should be a quotient ring of the form $R = \mathbb{Z}[x]/(f)$, i.e. the number field should be monogenic.

The second condition deals with the difference between the error distributions of PLWE and non-dual RLWE. For PLWE one simply uses a spherical Gaussian Γ_r^n on $R \otimes \mathbb{R}$ with respect to the polynomial basis $1, x, x^2, \ldots, x^{n-1}$, whereas the RLWE distribution $\Psi_{\mathbf{r}}$ is obtained by pulling back a near-spherical Gaussian distribution on H through the canonical embedding σ. With respect to the polynomial basis one can view $\Psi_{\mathbf{r}}$ as a near-spherical Gaussion that got distorted by $N_f \cdot B$. Since B is a unitary transformation, the only actual distortion comes from N_f.

The maximum distortion of N_f is captured by its spectral norm $s_1(N_f)$, i.e. its largest singular value. The other singular values are denoted by $s_i(N_f)$ ordered by size such that $s_n(N_f)$ denotes its smallest singular value. A spherical Gaussian distribution on H of parameter $\mathbf{r} = (r, r, \ldots, r)$ will therefore be

transformed into an elliptical Gaussian distribution on $R \otimes \mathbb{R} = K \otimes \mathbb{R}$ where the maximum parameter will be given by $s_1(N_f) \cdot r$. The attack on non-dual decision RLWE then proceeds by considering the samples with errors coming from $\Psi_{\mathbf{r}}$ as PLWE samples where the error is bounded by a spherical Gaussian with deviation $s_1(N_f) \cdot r$, with $r = \max(\mathbf{r})$.

For the attack to succeed we therefore need the following four conditions:

1. K is monogenic,
2. $f(1) \equiv 0 \bmod q$,
3. q is small enough that $\mathbb{Z}_q$ can be enumerated,
4. $r' = s_1(N_f) \cdot r$ is small enough such that $\phi(\lfloor \Gamma_{r'}^n \rceil)$ can be distinguished from uniform.

Again note that if $\mathbf{e}_i$ is bounded by Γ_r^n then the $\mathbf{e}_i(1)$ are bounded by a Gaussian with parameter $\sqrt{n} \cdot r' = \sqrt{n} \cdot s_1(N_f) \cdot r$. So the requirement is that the latter quantity is sufficiently smaller than q. In fact this is a very rough estimate, and indeed Elias et al. empirically observe in [6, §9] that their attack works more often than this bound predicts. We will explain this observation in Section 4.1.

In [6] the authors remark that given a parameter set (n, q, r) for PLWE, one cannot simply use the same parameter set for non-dual RLWE since the canonical embedding of the ring R into H might be very sparse, i.e. the covolume (volume of a fundamental domain) of $\sigma(R)$ in H might be very large. They therefore propose to scale up the parameter r by a factor of $|\det(M_f B)|^{1/n} = |\det(M_f)|^{1/n}$, which is the n-th root of the covolume. Thus given a PLWE parameter set (n, q, r), their corresponding RLWE parameter set reads $(n, q, \tilde{r})$ with $\tilde{r} = r \cdot |\det(M_f)|^{1/n}$.

3.3 Provably weak number fields for non-dual decision RLWE

The first type of polynomials to which the attack of [6] was applied are polynomials of the form $f_{n,a,b}$ with $a = 0$. More precisely they considered

$$f_{n,q} := f_{n,0,q-1} = x^n + q - 1\,,$$

where $n \geq 1$ and q is a prime. Note that the roots of these polynomials are simply the primitive $2n$-th roots of unity scaled up by $(q-1)^{1/n}$. These polynomials satisfy $f_{n,q}(1) \equiv 0 \bmod q$ and are irreducible by Eisenstein's criterion whenever $q - 1$ has a prime factor with exponent one. As shown in [6, Proposition 3], the polynomials $f_{n,q}$ are monogenic whenever $q - 1$ is squarefree, n is a power of a prime ℓ, and $\ell^2 \nmid ((1-q)^n - (1-q))$. In particular it is easy to construct examples for $n = 2^k$.

The final missing ingredient is a bound on the spectral norm $s_1(N_f)$. In [6], a slightly different matrix M_f is used (it is a real matrix containing the real and imaginary parts of the roots of f). For use further down, we adapt the proof of [6, Proposition 4] to derive *all* singular values $s_i(N_f)$. Due to its practical importance we will only deal with the case where n is even, since we are particularly interested in the case where $n = 2^k$.

Proposition 1 (Adapted from [6, Proposition 4]). *Assume that $f_{n,q}$ is irreducible and that n is even, then the singular values $s_i(N_f)$ are given by*

$$s_i(N_f) = \frac{1}{\sqrt{n}(q-1)^{(i-1)/n}}\,.$$

Proof: The roots of $f_{n,q}$ are given by $a \cdot \xi_{2n}^j$ for $0 < j < 2n$ and j odd, with $a = (q-1)^{1/n} \in \mathbb{R}^+$ and ξ_{2n} a primitive $2n$-th root of unity. To derive the singular values of $N_f = M_f^{-1}$ it suffices to derive the singular values of M_f. Recall that the u-th column of M_f (counting from 0) is given by

$$a^u \cdot (\xi_{2n}^u, \xi_{2n}^{3u}, \ldots, \xi_{2n}^{(2n-1)u})^t\,.$$

The (Hermitian) inner product of the u-th and v-th column is therefore given by

$$S = a^{u+v} \cdot \sum_{k=0}^{n-1} \xi_{2n}^{(2k+1)(u-v)}\,.$$

Since $\xi_{2n}^{2n+1} = \xi_{2n}$, we obtain that $\xi_{2n}^{2(u-v)} S = S$. For $u \neq v$ we have that $\xi_{2n}^{2(u-v)} \neq 1$, which implies that $S = 0$. For $u = v$ we obtain $S = na^{2u}$. This shows that the matrix M_f has columns that are orthogonal. The singular values of M_f can be read off from the diagonal of $\overline{M_f}^t \cdot M_f$, in particular $s_i(M_f) = \sqrt{n}a^{n-i}$ for $i = 1, \ldots, n$. This also shows that $s_i(N_f) = 1/(\sqrt{n}a^{i-1})$ for $i = 1, \ldots, n$. One finishes the proof by using that $a^n = q - 1$. □

The above proposition gives $s_1(N_f) = 1/\sqrt{n}$ which is small enough for the attack described in Section 3.2 to apply. In [6, Section 9], two examples of this family were attacked, giving the following results:

$f_{n,q}$	q	r	$\tilde{r}$	samples per run	successful runs	time per run
$x^{192} + 4092$	4093	8.87	5440.28	20	1 of 10	25 sec
$x^{256} + 8190$	8191	8.35	8399.70	20	2 of 10	44 sec

Recall that $\tilde{r}$ is simply r scaled up by a factor $|\det(M_f)|^{1/n}$. We remark, as do Elias et al. [6, §9], that these two examples unfortunately do *not* satisfy that $q - 1$ is squarefree. As a consequence the RLWE problem is not set up in the full ring of integers of the number field $K = \mathbb{Q}[x]/(f)$. We will nevertheless keep using these examples for the sake of comparison; it should be clear from the exposition below that this is not a crucial issue.

As a second instance, the authors of [6] considered polynomials of the form $f_{n,a,b} = x^n + ax + b$ with $a \approx b$, again chosen such that $f_{n,a,b}(1) \equiv 0$ modulo q, which is assumed to be an odd prime. More precisely, they let $a = (q-1)/2 + \Delta$ and $b = (q-1)/2 - \Delta - 1$, or $a = q + \Delta$ and $b = q - \Delta - 1$, for a small value of Δ. Heuristically these polynomials also result in weak instances of non-dual

decision RLWE, even though the analysis cannot be made as precise as in the foregoing case. In particular, no explicit formula is known for the spectral norm $s_1(N_f)$, but in [6] a heuristic perturbation argument is given that implies that it is bounded by $\sqrt{\max(a,b)} \cdot \det(N_f)^{1/n}$ infinitely often. They ran their attack for the particular case where $q = 524287$, $\Delta = 1$, $a = q + \Delta$ and $b = q - \Delta - 1$:

$f_{n,a,b}$	q	r	$\tilde{r}$	samples per run	successful runs	time per run
$x^{128} + 524288x + 524285$	524287	8.00	45540	20	8 of 10	24 sec

4 A simple attack on search RLWE

We derive a very simple attack on *search* RLWE for the families and parameter sets considered by Elias, Lauter, Ozman and Stange in [6]. The attack is based on two observations.

Firstly, a unit ball in the H-space gets severely deformed when being pulled back to $K \otimes \mathbb{R}$ along the canonical embedding. With respect to the polynomial basis $1, x, x^2, \ldots, x^{n-1}$ we end up with an ellipsoid whose axes have lengths $s_1(N_f), \ldots, s_n(N_f)$. For the first family of polynomials (i.e. where $a = 0$) this is a geometrically decreasing sequence, while for the second family this statement remains almost true. In particular a spherical Gaussian distribution $\Gamma_{\mathbf{r}}$ with $\mathbf{r} = (r, \ldots, r)$ on the H-space will result in a very skew elliptical Gaussian distribution on $K \otimes \mathbb{R}$ with parameters $s_1(N_f) \cdot r, \ldots, s_n(N_f) \cdot r$. For the choices of r (or in fact $\tilde{r}$) made by Elias et al., the errors along the shortest axes of the ellipsoid are so small that after rounding they become zero.

The second observation is that the axes of the error distribution ellipsoid coincide almost perfectly with the polynomial basis. Again for the first family this is exactly the case, while for the second family the distribution is consistent enough, in the sense that the axes do not line up perfectly, but the coordinates of the error samples with respect to $1, x, x^2, \ldots, x^{n-1}$ still tend to go down geometrically. The result is that the directions that get squashed simply correspond to the coefficients of the higher powers of x in the error terms $\mathbf{e}(x)$.

To make these statements precise we will compute the singular value decomposition of the whole transformation matrix $N_f \cdot B$. Recall that the singular value decomposition of an $n \times n$ matrix M is given by

$$M = U \Sigma \overline{V}^t,$$

where U, V are $n \times n$ unitary matrices and Σ is an $n \times n$ matrix with non-negative real numbers on the diagonal, namely the singular values. The image of a unit sphere under M will therefore result in an ellipsoid where the axes are given by the columns of U, with lengths equal to the corresponding singular values.

4.1 Singular value decomposition and error distribution

For the first family of polynomials $f_{n,q}$ everything can be made totally explicit:

Proposition 2. *The singular value decomposition of $N_f \cdot B$ is*

$$I_{n\times n} \cdot \Sigma \cdot \overline{V}^t, \qquad \text{where} \quad V = \overline{B}^t \cdot M_f \cdot \Sigma$$

and Σ is the diagonal matrix containing the singular values of N_f.

Proof: Recall from the proof of Proposition 1 that the Vandermonde matrix M_f has mutually orthogonal columns, where the ith column has norm $\sqrt{n}a^{i-1}$. Thus the normalized matrix

$$M_f \cdot \Sigma \qquad \text{where } \Sigma = \operatorname{diag}\left(1/(\sqrt{n}a^{i-1})\right)_i = \operatorname{diag}\left(s_i(N_f)\right)_i$$

is unitary. But then so is $V = \overline{B}^t \cdot M_f \cdot \Sigma$, and since $\Sigma = \Sigma^2 \cdot \Sigma^{-1} = N_f \overline{N_f}^t \cdot \Sigma^{-1}$, we see that

$$N_f \cdot B = I_{n\times n} \cdot \Sigma \cdot \overline{V}^t$$

is the singular value decomposition of our transformation matrix $N_f \cdot B$. □

The factor $I_{n\times n}$ implies that the axes of our ellipsoid match perfectly with the polynomial basis $1, x, x^2, \ldots, x^{n-1}$. In other words, if we start from a spherical error distribution $\Gamma_{\mathbf{r}}$ on H, $\mathbf{r} = (r, r, \ldots, r)$, then the induced error distribution $\Psi_{\mathbf{r}}$ on $K \otimes \mathbb{R}$ in the ith coordinate (coefficient of x^{i-1}) is a Gaussian with parameter

$$s_i(N_f) \cdot r = \frac{r}{\sqrt{n} \cdot (q-1)^{(i-1)/n}}$$

by Proposition 1. This indeed decreases geometrically with i.

As a side remark, note that this implies that for $\mathbf{e}(x) \leftarrow \Psi_{\mathbf{r}}$ the evaluation $\mathbf{e}(1)$ is sampled from a Gaussian with parameter

$$\left(\sum_{i=1}^{n} s_i(N_f)^2\right)^{1/2} \cdot r = s_1(N_f)\sqrt{\frac{(q-1)^2 - 1}{(q-1)^2 - (q-1)^{2(n-1)/n}}} \cdot r.$$

This is considerably smaller than $\sqrt{n} \cdot s_1(N_f) \cdot r$ and explains why the attack from [6] works better than what their theory predicts [6, §9].

To illustrate the geometric behavior of the coordinates of the errors $\mathbf{e}(x)$ with respect to the polynomial basis, we have plotted the average and standard deviation of their high order coefficients for the second example $x^{256} + 8190$ from [6] in Figure 1 (the results for the first example are totally similar), using the error parameter that they used to attack non-dual decision RLWE. The plot shows that for the given parameter set, the highest $\lceil n/7 \rceil$ error coefficients in the polynomial basis of $K \otimes \mathbb{R}$ are all extremely likely to be smaller than $1/2$ (indicated by the dashed line) in absolute value and therefore become zero after rounding.

For the second family of polynomials $f_{n,a,b}$ with $a \neq 0$, we were not able to derive the singular value decomposition in such an explicit form. To get a handle on them, we have computed it explicitly for $f = x^{128} + 524288x + 524285$. For

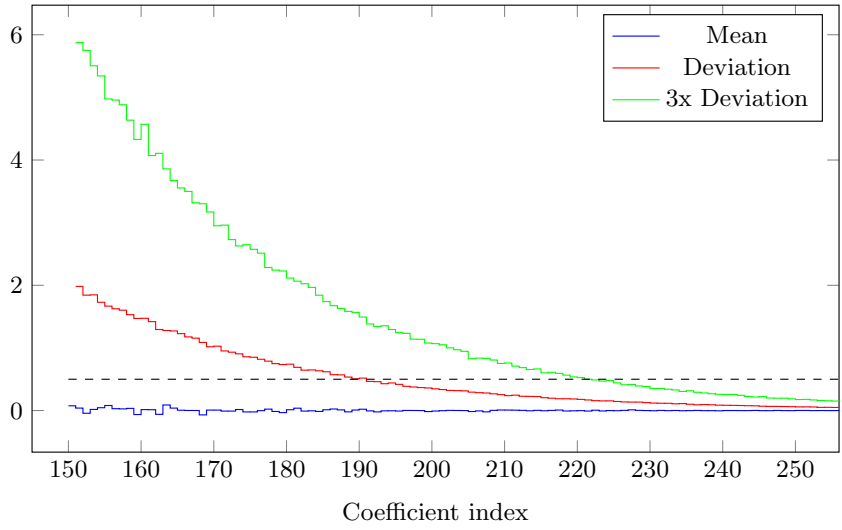


Fig. 1. Distribution of the error terms in the polynomial basis for $f = x^{256} + 8190$

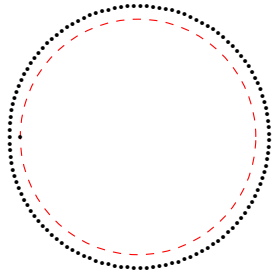


Fig. 2. Zeroes in $\mathbb{C}$ of $x^{128} + 524288x + 524285$, along with the unit circle (dashed)

this particular example, the roots of $f_{n,a,b}$ again lie roughly on a circle (except for the real root close to -1): see Figure 2. So through the Vandermonde matrix we again expect geometric growth of the singular values, as is confirmed by the explicit numerics in Figure 3, which shows a plot of their logarithms. There is only one outlier, caused by the real root of f close to -1.

The heat map in Figure 4 plots the norms of the entries in the U-matrix of the singular value decomposition of $N_f \cdot B$ and shows that U is close to being diagonal, implying that the axes of the ellipsoid are indeed lining up almost perfectly with the polynomial basis. Finally Figure 5 contains a similar plot as Figure 1, namely, the distribution of the errors terms (highest powers only) for the polynomial $f = x^{128} + 524288x + 524285$. Again we conclude that with very high probability, the last $\lceil n/6 \rceil$ coefficients of the error terms in the polynomial basis will be smaller than $1/2$, and therefore they become zero after rounding.

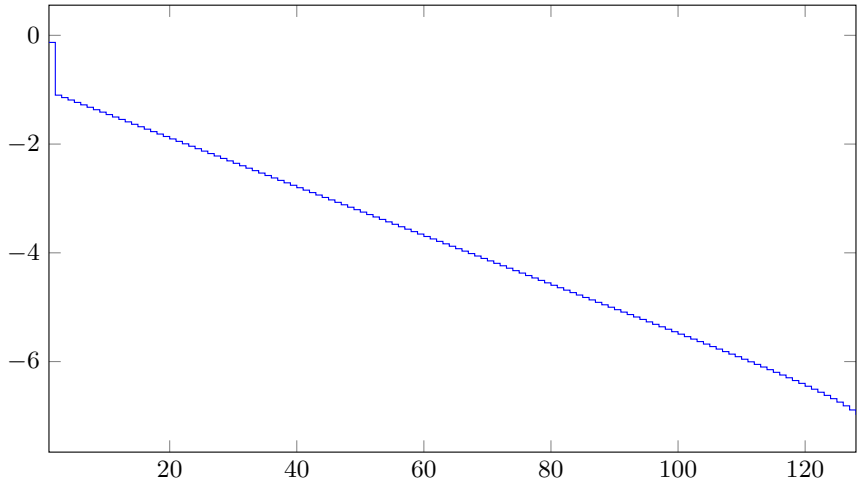


Fig. 3. $\log_{10}$ of the singular values of N_f for $f = x^{128} + 524288x + 524285$

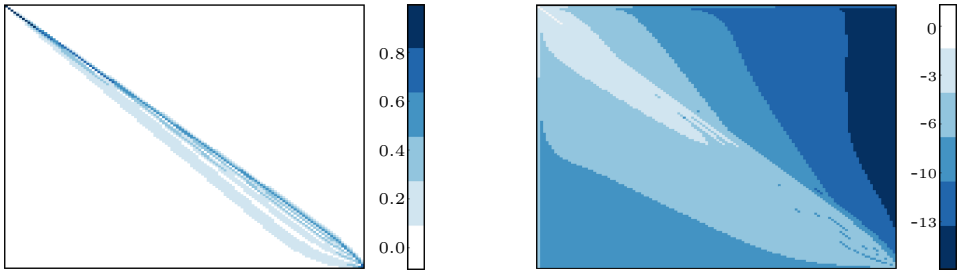


Fig. 4. Heat maps of the norms of the entries of U (left) and $\log_{10}$ of the norms of the entries of $U\Sigma$ (right), where $U\Sigma\overline{V}^t$ is the singular value decomposition of $N_f B$

4.2 Linear algebra attack on non-dual search RLWE

Turning the above observations into an attack on non-dual search RLWE for these families is straightforward. Recall that the samples are of the form $(\mathbf{a}, \mathbf{b} = \mathbf{a}\cdot\mathbf{s}+\mathbf{e} \bmod q)$ where the errors were sampled from the distribution $\Psi_{\mathbf{r}}$ on $K\otimes\mathbb{R}$. Since $\mathbf{a}$ is known, we can express multiplication by $\mathbf{a}$ as a linear operation, i.e. we can compute the $n \times n$ matrix $M_{\mathbf{a}}$ that corresponds to multiplication by $\mathbf{a}$ with respect to the polynomial basis $1, x, x^2, \ldots, x^{n-1}$. Each RLWE sample can therefore be written as a linear algebra problem as follows:

$$M_{\mathbf{a}} \cdot (s_0, s_1, \ldots, s_{n-1})^t = (b_0, b_1, \ldots, b_{n-1})^t - (e_0, e_1, \ldots, e_{n-1})^t \qquad (1)$$

where the s_i (resp. b_i, e_i) are the coefficients of $\mathbf{s}$ (resp. $\mathbf{b}$ and $\mathbf{e}$) with respect to the polynomial basis. By rounding the coefficients of the right-hand side, we effectively remove the error terms of high index, which implies that the last equations in the linear system become *exact* equations in the unknown coefficients of

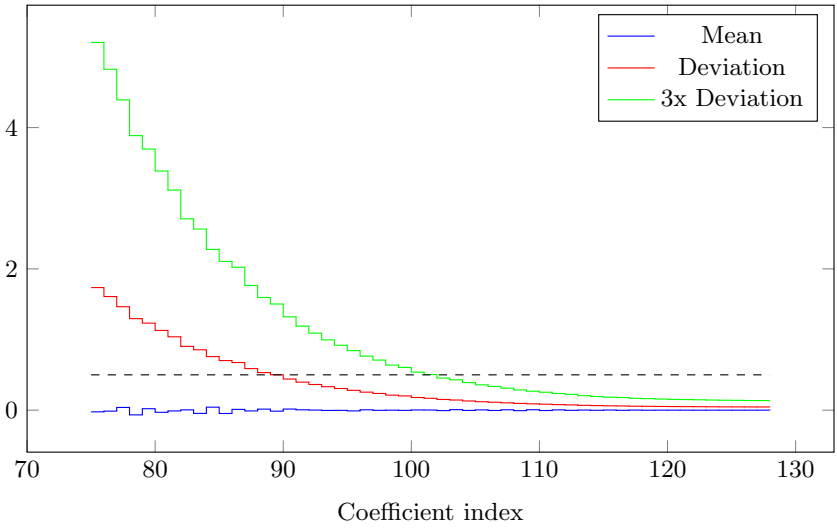


Fig. 5. Distribution of errors of high index for $f = x^{128} + 524288x + 524285$

s. Assuming that the highest $\lceil n/k \rceil$ error terms round to zero, we only require k samples to recover the secret **s** using simple linear algebra with a 100% success rate.

We have implemented this attack in Magma [2] with the following results.

$f_{n,a,b}$	q	r	$\tilde{r}$	samples per run	successful runs	time per run
$x^{192} + 4092$	4093	8.87	5440	7	10 of 10	8.37 sec
$x^{256} + 8190$	8191	8.35	8390	6	10 of 10	17.2 sec
$x^{128} + 524288x + 524285$	524287	8.00	45540	4	10 of 10	1.96 sec

We note that using less samples per run is also possible, but results in a lower than 100% success rate. A more elaborate strategy would construct several linear systems of equations by discarding some of the equations of lower index (which are most likely to be off by 1) and running exhaustively through the kernel of the resulting underdetermined system of equations. However, we did not implement this strategy since it needlessly complicates the attack.

In fact for errors of the above size one can also use the linearization technique developed by Arora and Ge [1, Theorem 3.1] to retrieve $\mathbf{s}(x)$, but this requires a lot more samples.

We stress that our attack does not use that $f(1) \equiv 0 \bmod q$. For the above defining polynomials our attack works modulo *every* modulus q', as long as the same error parameters are used (or smaller ones).

4.3 Modifications for dual search RLWE

In this section we discuss how switching from non-dual RLWE (i.e. from $I = R$) to dual RLWE (where one takes $I = R^\vee$) affects our observations. Recall that in the case of a monogenic defining polynomial f, the codifferent $R^\vee$ is generated as a fractional ideal by $1/f'(\alpha)$ with $\alpha \in \mathbb{C}$ a root of f. We will again work with respect to the polynomial basis $1, x, x^2, \ldots, x^{n-1}$ of $K = \mathbb{Q}[x]/(f)$ over $\mathbb{Q}$, which is also a basis of R over $\mathbb{Z}$, and take $\alpha = x$. For technical reasons we will only do the analysis for the first family of polynomials, namely those of the form

$$f_{n,q} = f_{n,0,q-1} = x^n + q - 1,$$

where one has $f'_{n,q} = nx^{n-1}$. Since

$$1 = \frac{1}{q-1} f_{n,q} - \frac{x}{n(q-1)} f'_{n,q}$$

we find that

$$R^\vee = R\frac{x}{n(q-1)}.$$

Proposition 3. *The elements*

$$\frac{1}{n}, \frac{x}{n(q-1)}, \frac{x^2}{n(q-1)}, \frac{x^3}{n(q-1)}, \ldots, \frac{x^{n-1}}{n(q-1)} \tag{2}$$

form a $\mathbb{Z}$-basis of $R^\vee$.

Proof. It is immediate that

$$\frac{x}{n(q-1)}, \frac{x^2}{n(q-1)}, \frac{x^3}{n(q-1)}, \ldots, \frac{x^{n-1}}{n(q-1)}, \frac{x^n}{n(q-1)}$$

form a $\mathbb{Z}$-basis. But modulo $f_{n,q}$ the last element is just $-1/n$.

Thus we can think of our secret $\mathbf{s}(x) \in R_q^\vee$ as a $\mathbb{Z}$-linear combination of the elements in (2), where the coefficients are considered modulo q. A corresponding RLWE-sample is then of the form $(\mathbf{a}(x), \mathbf{a}(x)\cdot\mathbf{s}(x) + \mathbf{e}(x) \bmod qR^\vee)$ with $\mathbf{e}(x) \in R^\vee \otimes \mathbb{R} = K \otimes \mathbb{R}$ sampled from $\Psi_{\mathbf{r}}$ for an appropriate choice of $\mathbf{r} \in (\mathbb{R}^+)^n$. To make a comparison with our attack in the non-dual case, involving the parameters from [6], we have to make an honest choice of $\mathbf{r}$, which we again take spherical. Note that the lattice $\sigma(R^\vee)$ is much denser than $\sigma(R)$: the covolume gets scaled down by a factor

$$\left|\mathrm{No}(f'_{n,q}(\alpha))\right| = n^n(q-1)^{n-1}.$$

Therefore, in view of the discussion concluding Section 3.2, we scale down our scaled-up error parameter $\tilde{r}$ by a factor

$$\sqrt[n]{n^n(q-1)^{n-1}} \approx n(q-1).$$

Let us denote the result by $\tilde{r}^\vee$.

It follows that the dual setting is essentially just a scaled version of its non-dual counterpart: both the errors and the basis elements become divided by a factor of roughly $n(q-1)$. In particular, for the same choices of r we again find that with near certainty the highest $\lceil n/7 \rceil$ error coefficients are all smaller than

$$\frac{1}{2} \cdot \frac{1}{n(q-1)}$$

in absolute value, and therefore become zero after rounding to the nearest multiple of $1/(n(q-1))$. This then again results in exact equations in the coefficients of the secret $\mathbf{s}(x) \in R_q^\vee$ with respect to the basis (2), that can be solved using linear algebra.

Here too, the attack does not use that $f(1) \equiv 0 \bmod q$ so it works for whatever choice of modulus q' instead of q, as long as the same error parameters are used (or smaller ones).

5 Range of applicability

One obvious way of countering our attack is by modifying the error parameter. In principle the skewness of $N_f \cdot B$ could be addressed by using an equally distorted elliptical Gaussian rather than a near-spherical one, but that conflicts with the philosophy of RLWE (as opposed to PLWE), namely that the more natural way of viewing a number field is through its canonical embedding. So we will not discuss this option and stick to spherical distributions. Then the only remaining way out is to enlarge the width of the distribution. Again for technical reasons we will restrict our discussion to the first family of polynomials, namely those of the form $f_{n,q} = x^n + q - 1$; the conclusions for the second family should be similar.

In the non-dual case we see that a version of the attack works as long as a sample drawn from a univariate Gaussian with parameter $s_n(N_f) \cdot \tilde{r}$ has absolute value less than $1/2$ with non-negligible probability: then by rounding one obtains at least one exact equation in the unknown secret $\mathbf{s}(x)$. For this one needs that

$$s_n(N_f) \cdot \tilde{r} \leq \frac{C}{2}$$

for some absolute constant $C > 0$ that quantifies what it means to be 'non-negligible'.

Remark 1. In order to recover the *entire* secret, one even wants a non-negligible probability for n consecutive samples to be less than $1/2$, for which one should replace $s_n(N_f) \cdot \tilde{r}$ by $s_n(N_f) \cdot \tilde{r} \cdot \sqrt{\log n}$ (roughly). In fact a slightly better approach is to find the optimal $1 \leq k \leq n$ for which $s_{n-k+1}(N_f) \cdot \tilde{r}$ is likely to be less than $1/2$, thereby yielding at least k exact equations at once, for $\lceil n/k \rceil$ consecutive times.

Let us take $C = 1$ in what follows: for this choice meeting the upper bound corresponds to a chance of about 98.78% of recovering at least one exact equation. Using Proposition 1 this can be rewritten as

$$\tilde{r} \leq \frac{1}{2} \cdot \sqrt{n} \cdot (q-1)^{1-1/n}. \tag{3}$$

For our two specific polynomials $x^{192} + 4092$ and $x^{256} + 8190$ the right-hand side reads 27148.39 and 63253.95 whereas Elias et al. took $\tilde{r}$ to be 5440.28 and 8399.70, respectively.

Note that the bound in (3) does not depend on the modulus q' that is being used: the q that appears there is just part of the data defining our number field. In other words, whenever $\tilde{r}$ satisfies (3) then for every choice of modulus q' we are very likely to recover at least one exact equation in the coefficients of the secret $\mathbf{s}(x)$.

Unfortunately the bound (3) does not allow for an immediate comparison with the hardness result of Lyubashevsky, Peikert and Regev (see Theorem 1), which was formulated for dual RLWE only. But for dual RLWE one can make a similar analysis. From Section 4.3 it follows that we want error coefficients that are smaller than $1/(2n(q-1))$ with a non-negligible probability. The same discussion then leads to the bound

$$\tilde{r}^{\vee} \leq \frac{1}{2} \cdot \frac{1}{\sqrt{n} \cdot (q-1)^{\frac{1}{n}}} \tag{4}$$

which is highly incompatible with the condition $\tilde{r}^{\vee} \geq 2\omega(\sqrt{\log n})$ from Theorem 1. Thus we conclude that it is impossible to enlarge the error parameter up to a range where our attack would form an actual threat to RLWE, as defined in [9, §3].

Another issue with modifying the error parameter is decodability. In the non-dual case, from (3) we see that $s_n(N_f) \cdot \tilde{r} \gg 1$ is needed to avoid being vulnerable to our skewness attack. But it automatically follows that $s_1(N_f) \cdot \tilde{r} \gg q$. Indeed, this is implied by the fact that the condition number $k(N_f) := s_1(N_f)/s_n(N_f)$ equals

$$(q-1)^{1-1/n} \approx q$$

by Proposition 1. This causes the errors at the terms of low degree to wrap around modulo q. In the dual case the same observation applies, where now the error terms of low degree tend to wrap around modulo multiples of $q \cdot 1/(n(q-1))$. In both cases the effect is that several of these terms become indistinguishable from uniform, requiring more samples for the RLWE problem to become information theoretically solvable. This obstructs, or at least complicates, certain cryptographic applications.

So overall, the conclusion is that the defining polynomials $f_{n,a,b}$ are just not well-suited for use in RLWE: either the error parameter is too small for the RLWE problem to be hard, or the error parameter is too large for the problem to be convenient for use in cryptography. But we stress once more that neither the attack from [6] nor our attack form a genuine threat to RLWE, as it was defined in [9, §3].

6 Conclusions

In this paper we have shown that non-dual *search* RLWE can be solved efficiently for the families of polynomials and parameter sets from [6] which were shown to be weak for the *decision* version of the problem. The central reason for this weakness lies in the (exponential) skewness of the canonical embedding transformation. We analyzed the singular value decomposition of this transformation and showed that the singular values form an (approximate) geometric series. Furthermore, we also showed that the axes of the error ellipsoid are consistent with the polynomial basis, allowing us to readily identify very small noise coefficients. The attack applies to wider ranges of moduli, and also applies to the dual version, but does not contradict any statement in the work of Lyubashevsky, Peikert and Regev [9].

It is worth remarking that while we used the language of singular value decomposition, for our skewness attack it merely suffices that $N_f \cdot B$ has a very short row, so that the corresponding error coefficient e_i vanishes after rounding and (1) provides an exact equation in the coefficients of the secret. For general number fields this is a strictly weaker condition than having a very small singular value whose corresponding axis lines up perfectly with one of the polynomial basis vectors. But for the particular families of [6] the singular value decomposition turned out to be a convenient tool in proving this, and in visualizing how the RLWE errors are transformed under pull-back along the canonical embedding.

Acknowledgments

This work was supported by the European Commission through the ICT programme under contract H2020-ICT-2014-1 644209 HEAT and contract H2020-ICT-2014-1 645622 PQCRYPTO. We would like to thank Ron Steinfeld and the anonymous referees for their valuable comments.

References

1. Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In *Automata, languages and programming. Part I*, volume 6755 of *Lecture Notes in Comput. Sci.*, pages 403–415. Springer, Heidelberg, 2011.
2. Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).
3. Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Symposium on Theory of Computing Conference, STOC'13*, pages 575–584. ACM, 2013.
4. Léo Ducas and Alain Durmus. Ring-LWE in polynomial rings. In *Public Key Cryptography - PKC 2012*, volume 7293 of *Lecture Notes in Computer Science*, pages 34–51. Springer, 2012.
5. Kirsten Eisenträger, Sean Hallgren, and Kristin E. Lauter. Weak instances of PLWE. In *Selected Areas in Cryptography - SAC 2014*, volume 8781 of *Lecture Notes in Computer Science*, pages 183–194. Springer, 2014.

6. Yara Elias, Kristin E. Lauter, Ekin Ozman, and Katherine E. Stange. Provably weak instances of Ring-LWE. In *Advances in Cryptology - CRYPTO 2015 - Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 63–92. Springer, 2015.
7. Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *Algorithmic Number Theory, ANTS-III*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer, 1998.
8. Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In *Topics in Cryptology - CT-RSA 2011*, volume 6558 of *Lecture Notes in Computer Science*, pages 319–339. Springer, 2011.
9. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23. Springer, 2010.
10. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for Ring-LWE cryptography. In *Advances in Cryptology - EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 35–54. Springer, 2013.
11. Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *Symposium on Theory of Computing, STOC 2009*, pages 333–342. ACM, 2009.
12. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Symposium on Theory of Computing*, pages 84–93. ACM, 2005.
13. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.

Chapter 7

On Error Distributions in Ring-based LWE.

Publication data

Castryck, W., Iliashenko, I., and Vercauteren, F. On error distributions in ring-based LWE. *LMS Journal of Computation and Mathematics* 19, A (2016), 130–145.

On Error Distributions in Ring-based LWE

W. Castryck, I. Iliashenko and F. Vercauteren

Abstract

Since its introduction in 2010 by Lyubashevsky, Peikert and Regev, the Ring Learning With Errors problem (Ring-LWE) has become a popular building block for cryptographic primitives, due to its great versatility and its hardness proof consisting of a (quantum) reduction from ideal lattice problems. But for a given modulus q and degree n number field K, generating Ring-LWE samples can be perceived as cumbersome, because the secret keys have to be taken from the reduction mod q of a certain fractional ideal $\mathcal{O}_K^\vee \subset K$ called the codifferent or 'dual', rather than from the ring of integers $\mathcal{O}_K$ itself. This has led to various non-dual variants of Ring-LWE, in which one compensates for the non-duality by scaling up the errors. We give a comparison of these versions, and revisit some unfortunate choices that have been made in the recent literature, one of which is scaling up by $|\Delta_K|^{1/2n}$ with Δ_K the discriminant of K. As a main result, we provide for any $\varepsilon > 0$ a family of number fields K for which this variant of Ring-LWE can be broken easily as soon as the errors are scaled up by $|\Delta_K|^{(1-\varepsilon)/n}$.

1. *Introduction: Ring-based versions of LWE*

About a decade ago Regev [**22**] proposed a new hard problem for use in public-key cryptography, namely the learning with errors problem (LWE), which informally stated is about solving an approximate linear system

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} = A \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_m \end{pmatrix}$$

for an unknown secret $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ over $\mathbb{Z}/q\mathbb{Z}$, with q some integer modulus. The entries of A are selected independently and uniformly at random in $\mathbb{Z}/q\mathbb{Z}$ and the e_i are small error terms, obtained by sampling from a fixed Gaussian with mean 0 and standard deviation $\rho \geq \sqrt{2n/\pi}$, and reducing the outcome mod q. These errors are elements of $\mathbb{R}/q\mathbb{Z}$, but in practice they can be rounded to the nearest element of $\mathbb{Z}/q\mathbb{Z}$. To recover $\mathbf{s}$ uniquely, the system has to be overdetermined, i.e. $m > n$. In fact in Regev's model an attacker is allowed to ask for new equations indefinitely, in the (conjecturally vain) hope of learning more information about $\mathbf{s}$: hence the terminology learning with errors.

The LWE problem is being acclaimed for three reasons. Firstly it enjoys a hardness proof in the form of a reduction from worst-case forms of certain well-established lattice problems [**22**, **20**, **3**], providing security guarantees that are lacking for classical hard problems such as integer factorization or discrete logarithm computation. Secondly, it seems that LWE would remain hard in a post-quantum world, unlike the classical problems [**23**]. Thirdly, LWE has proven to be very versatile for use in cryptography, enabling applications that were not known before,

---

2000 *Mathematics Subject Classification* 11T71(primary), 11R04, 11R11, 11R18, 11T22(secondary).

This work was supported by the European Commission through the ICT programme under contracts H2020-ICT-2014-1 644209 HEAT and H2020-ICT-2014-1 645622 PQCRYPTO. The authors would like to thank the anonymous reviewers for their highly valuable comments, which led to several improvements.

such as homomorphic encryption [**5**, **2**]. Its major drawback however is that the key sizes of the resulting cryptosystems are impractically large, because typically one needs the entire $(m \times n)$-matrix A.

One idea to address this [**17**, **19**] is to use a ring structure $R_q = \mathbb{Z}[x]/(q, f(x))$ for some monic degree n polynomial $f(x) \in \mathbb{Z}[x]$ through the isomorphism (a priori of modules)

$$\varphi : \left(\frac{\mathbb{Z}}{q\mathbb{Z}}\right)^n \to R_q : (s_1, s_2, \ldots, s_n) \mapsto s_1 + s_2 x + \cdots + s_n x^{n-1}.$$

Each block of n rows of A is replaced with the matrix $A_{\mathbf{a}}$ of multiplication by a random ring element $\mathbf{a}(x)$, say with respect to the polynomial basis $1, x, \ldots, x^{n-1}$, in order to obtain an approximate linear system of the form

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}, \tag{1.1}$$

which using φ can be rewritten as $\mathbf{b}(x) = \mathbf{a}(x) \cdot \mathbf{s}(x) + \mathbf{e}(x)$. When storing $\mathbf{a}(x)$ rather than $A_{\mathbf{a}}$ one gains a factor n, thereby addressing the key size issue. The general setup above is called **Ring-based LWE** (not to be confused with Ring-LWE) and the terminology allows for any error distribution Ψ on $\mathbb{R}_q$ for the error terms $\mathbf{e}(x)$. In the remainder of the article, we will consider three variants, all of which sample the error terms as a linear transformation of an n-dimensional spherical Gaussian Γ_r^n, i.e. there exists a matrix T such that $(e_1, \ldots, e_n)^t = T \cdot \Gamma_r^n$. The different choices for T are summarized in Table 1 and how these T arise is explained in detail in the next paragraphs.

Poly-LWE can be considered the most straightforward generalisation of LWE, in that the errors e_i are drawn independently from the same Gaussian with mean 0 and small standard deviation ρ. In particular, the matrix T is simply the identity matrix. For the sake of analogy one could again impose $\rho \geq \sqrt{2n/\pi}$, but for our needs it suffices that ρ depends on n only, in a non-negligible and polynomially bounded way. In joint terms one sees that $(e_1, e_2, \ldots, e_n)$ is being drawn from a spherical Gaussian distribution on $\mathbb{R}^n$ centered around the origin and reduced modulo $q\mathbb{Z}^n$. Unfortunately restricting to multiplication matrices comes at the cost of giving up on the uniform randomness, thereby invalidating the mentioned hardness proof, and in fact it is possible to cook up instances of the problem having certain flaws. For example if $f(1) \equiv 0 \bmod q$ then $\mathbf{b}(1) \equiv \mathbf{a}(1) \cdot \mathbf{s}(1) + \mathbf{e}(1) \bmod q$, which can in certain special cases be exploited to obtain information about the secret [**13**], thereby mimicking an attack on early versions of NTRU that use arithmetic modulo $f(x) = x^n - 1$, see [**16**]. This concern is partly addressed by restricting to irreducible $f(x) \in \mathbb{Z}[x]$, which we do from now on.

Ring-LWE was introduced by Lyubashevsky, Peikert and Regev in [**19**] and admits a hardness proof akin to the one for general LWE. The main difference is that the error terms are generated in a way that is canonical for the underlying number field K defined by $f(x)$ and in particular, does not depend on the choice of the defining polynomial $f(x)$ itself, but only on K (unlike Poly-LWE). For the purpose of this introduction, it suffices to think of Ring-LWE samples as above, except that the error vector $(e_1, e_2, \ldots, e_n)$ is being transformed

	Poly-LWE	Ring-LWE	SCG Ring-based LWE
T	$I_{n\times n}$	$A_{f'(x)} \cdot B^{-1}$	$\lambda \cdot B^{-1}$

TABLE 1. *Noise distributions $T \cdot \Gamma_r^n$ in three main instantiations of Ring-based LWE, with B the (real) canonical embedding matrix, $A_{f'(x)}$ matrix of multiplication by $f'(x)$ and λ a constant*

in the following specific way:

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = A_{\mathbf{a}} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + A_{f'(x)} \cdot B^{-1} \cdot \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}. \tag{1.2}$$

Here $B \in \mathbb{R}^{n \times n}$ is the Vandermonde matrix $(\alpha_i^{j-1})_{i,j}$ generated by the roots $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{C}$ of $f(x)$, turned into a real matrix using an easy unitary transformation. The factor B^{-1} expresses that Ring-LWE errors are actually generated in the codomain of the canonical embedding of the number field $K = \mathbb{Q}[x]/(f(x))$. On the other hand $A_{f'(x)}$ is the matrix of multiplication by the derivative $f'(x)$ of our defining polynomial. It compensates for the fact that we sampled the secret $\mathbf{s}(x)$ from the reduction mod q of $R = \mathbb{Z}[x]/(f(x))$, rather than from the reduction mod q of a certain fractional ideal $R^\vee \subset K$, called the *dual* of R. It is convenient to think of $A_{f'(x)}$ as an *integral* matrix, i.e. as the matrix of multiplication by $f'(x)$ in R with respect to the $\mathbb{Z}$-basis $1, x, \ldots, x^{n-1}$. As in Poly-LWE the vector $(e_1, e_2, \ldots, e_n)$ is sampled from a spherical Gaussian centered at the origin with non-negligible standard deviation $\rho \in \mathrm{poly}(n)$. But the reduction modulo $q\mathbb{Z}^n$ only happens *after* multiplication by $A_{f'(x)} \cdot B^{-1}$.

The matrix $A_{f'(x)} \cdot B^{-1}$ transforms our spherical distribution into an ellipsoidal one. In particular the errors in certain coordinates might be systematically much larger than those in others, and they might no longer be independent. But it is crucial to observe that the error coordinates are being *scaled up* on average, in the sense of the geometric mean. Indeed, one can show that $\left|\det A_{f'(x)}\right| = \Delta$ and that $|\det B| = \sqrt{\Delta}$, where Δ denotes the absolute value of the discriminant of $f(x)$. Thus

$$\left|\det\left(A_{f'(x)} \cdot B^{-1}\right)\right| = \sqrt{\Delta},$$

meaning that on average the errors tend to grow by a factor $\Delta^{1/2n}$.

SCG ring-based LWE where SCG stands for Scaled Canonical Gaussian, was analyzed in a series of papers [**14**, **7**, **8**] where it was called non-dual Ring-LWE. In this version, one considers samples of the form

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = A_{\mathbf{a}} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \lambda \cdot B^{-1} \cdot \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}, \tag{1.3}$$

where $\lambda \geq 1$ denotes a fixed real number. This variant basically replaces the matrix $A_{f'(x)}$ in Ring-LWE by a scalar matrix. The authors called this variant non-dual Ring-LWE since the matrix $A_{f'(x)}$ corresponds to the factor coming from working with the dual. However, we do not use this terminology since it should refer to the variant that is equivalent with Ring-LWE but avoids explicit use of the dual by mutliplying by $f'(x)$ (as was done exactly as in (1.2)).

Note that one cannot simply remove $A_{f'(x)}$ (i.e. take $\lambda = 1$), since the remaining factor B^{-1} has determinant $1/\sqrt{\Delta}$, which typically scales down the errors to a point where they become negligible, leading to *exact* equations in $s_1, s_2, \ldots, s_n$ that can be solved using linear algebra. This is of course highly undesirable, and to remedy this the authors of [**14**, **7**, **8**] used $\lambda = \sqrt{\Delta}^{1/n}$, in order to undo the factor B^{-1} determinant-wise.

This choice of scalar indeed takes back the errors to a reasonable size, but only on average. If the ellipsoidal distribution induced by B^{-1} is extremely skew then there might be error coordinates that remain negligibly small after scaling. The following example, introduced in [**14**] and revisited in [**6**], illustrates this: for $f(x) = x^{256} + 8190$ the successive radii of the corresponding 256-dimensional ellipsoid go down geometrically, as is illustrated in Figure 1. It

turns out that with

$$\rho = 8.35 \qquad \text{and} \qquad \lambda = \sqrt{\Delta}^{1/256} \approx 1422.72,$$

the coordinates at the highest 45 indices become zero after rounding, with overwhelming probability. Thus each sample yields 45 exact equations in $s_1, s_2, \ldots, s_n$, and about six samples suffice to recover the entire secret. If one would take $\lambda = 1$ the effect is even more pronounced since then over 240 errors are negligible, and one only requires 2 samples. In general, for this attack to work it is enough that B^{-1} admits a very short $\mathbb{Z}$-linear combination of its rows. See [**21**] for a more thorough analysis of all this.

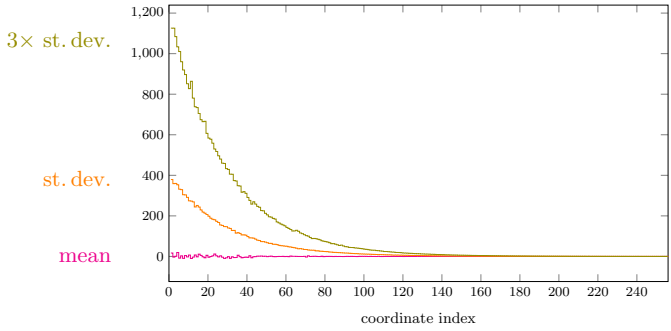


FIGURE 1. *Coordinate-wise error distributions for* $f(x) = x^{256} + 8190$, $\rho = 8.35$, *and* $\lambda = \sqrt{\Delta}^{1/256}$.

To us it seems more natural to take $\lambda = \Delta^{1/n}$: in this way one compensates determinant-wise for the removal of $A_{f'(x)}$. For this choice of scalar we are unaware of any attacks on SCG ring-based LWE and it would be interesting to know whether a variant of the hardness proof of [**19**] applies here.

The main result of this article is that $\Delta^{1/n}$ is a lower bound for λ, in the following sense: for each $\varepsilon > 0$ we provide a family of irreducible polynomials $f(x) \in \mathbb{Z}[x]$ of increasing degree n, for which $O(n)$ SCG samples of the form (1.3) with $\lambda = \Delta^{(1-\varepsilon)/n}$ are sufficient to recover the entire secret using standard linear algebra.

In fact, as we will see, the analogous result also applies to Ring-LWE, for the same families of polynomials. In other words, as soon as one scales *down* the right-most term in (1.2) by $\Delta^{\varepsilon/n}$ then the corresponding samples leak exact equations, again allowing one to find the entire secret easily. However, as suggested by a reviewer, in this case the statement admits an easier proof, based on the trivial fact that $1 \in R$.

The article is organized as follows. In Section 2 we give a more formal introduction to Ring-LWE, while Section 3 is devoted to the SCG Ring-based LWE version that was studied in [**14**, **7**, **8**]. Apart from providing more details, these descriptions will differ slightly from the one given in the introduction: instead of $\mathbb{Z}[x]/(f(x))$ we will work in the potentially larger ring of integers $\mathcal{O}_K$ of K. Then in Section 4 we state a rigorous version of our tightness result on the scaling factor λ, and provide a proof. Finally in Section 5 we make some additional comments from the point of view of Galois theory.

2. *Ring-LWE set up formally*

The actual Ring-LWE problem is formulated using the ring of integers $R = \mathcal{O}_K$ of a given degree n number field K, which one considers along with a modulus $q \in \mathbb{Z}$. A central role is played by the codifferent $R^\vee$ of K, which is defined as the inverse (fractional) ideal of the

different ideal $\partial \subset R$. Alternatively it can be viewed as the dual of R with respect to the trace pairing:

$$R^\vee = \{x \in K \mid \mathrm{Tr}_{K/\mathbb{Q}}(xR) \subset \mathbb{Z}\}. \tag{2.1}$$

The reductions of R and $R^\vee$ modulo qR resp. $qR^\vee$ are denoted by R_q and $R_q^\vee$, respectively. The Ring-LWE problem is then about finding a fixed secret $\mathbf{s} \in R_q^\vee$ from an arbitrary number of approximate equations of the form

$$\mathbf{b} = \mathbf{a} \cdot \mathbf{s} + \mathbf{e}, \tag{2.2}$$

where $\mathbf{a} \in R_q$ is chosen uniformly at random and $\mathbf{e}$ is a small error term sampled from a distribution that will be described in the next paragraph. Recall that everything is to be interpreted modulo $qR^\vee$. After agreeing upon a $\mathbb{Z}$-basis of $R^\vee$ this can be rewritten as

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_1 \\ \vdots \\ e_n \end{pmatrix}$$

where the s_i are the coordinates of $\mathbf{s}$, the b_i are the coordinates of $\mathbf{b}$, and $A_{\mathbf{a}}$ is the matrix of multiplication by $\mathbf{a}$ with respect to the chosen $\mathbb{Z}$-basis, all considered modulo q.

As for the error distribution, the main role is played by the canonical embedding

$$\sigma : K \to \mathbb{C}^n : \alpha \mapsto (\sigma_1(\alpha), \ldots, \sigma_n(\alpha)),$$

where $\sigma_1, \ldots, \sigma_s$ are the real ring monomorphisms from K to $\mathbb{R}$ and $\sigma_{s+1}, \ldots, \sigma_{s+2t}$ are the complex ring monomorphisms from K to $\mathbb{C}$ (so that $n = s + 2t$), ordered such that $\sigma_{s+i} = \tau \circ \sigma_{s+t+i}$ for $i = 1, \ldots, t$, where $\tau : \mathbb{C} \to \mathbb{C} : z \mapsto \overline{z}$ denotes complex conjugation. Thus σ takes values in

$$H = \{\, (z_1, \ldots, z_n) \in \mathbb{C}^n \mid z_1, \ldots, z_s \in \mathbb{R} \text{ and } \overline{z}_{s+i} = z_{s+t+i} \text{ for } i = 1, \ldots, t \,\},$$

which when equipped with the Hermitian inner product $\langle \cdot, \cdot \rangle$ is seen to be isomorphic to the standard inner product space $\mathbb{R}^n$, by considering the basis given by the columns of the unitary matrix

$$U = \begin{pmatrix} I_{s\times s} & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} I_{t\times t} & \frac{\mathbf{i}}{\sqrt{2}} I_{t\times t} \\ 0 & \frac{1}{\sqrt{2}} I_{t\times t} & -\frac{\mathbf{i}}{\sqrt{2}} I_{t\times t} \end{pmatrix}.$$

It is well-known that under this identification of H with $\mathbb{R}^n$, the image $\sigma(I)$ of a fractional ideal $I \subset K$ is a lattice of rank n, and that $\sigma(R^\vee)$ is the complex conjugate of the dual lattice

$$\sigma(R)^* := \{\, \alpha \in H \mid \langle \alpha, \sigma(R) \rangle \subset \mathbb{Z} \,\},$$

as is immediate from (2.1); more generally $\sigma(I)^* = \tau(\sigma(I^\vee))$ where $I^\vee = (\partial I)^{-1}$. Now consider a spherical Gaussian on $\mathbb{R}^n$, say with distribution function

$$\Gamma_r^n(\mathbf{x}) = \frac{1}{r^n} \exp\left(-\pi \frac{\|\mathbf{x}\|^2}{r^2}\right),$$

where we note that Γ_r^1 is a univariate Gaussian distribution with mean 0 and standard deviation $r/\sqrt{2\pi}$, and that

$$\Gamma_r^n = \Gamma_r^1 \times \Gamma_r^1 \times \cdots \times \Gamma_r^1.$$

We view Γ_r^n as a distribution on H through the above identification with $\mathbb{R}^n$. Pulling it back along the canonical embedding and reducing it mod $qR^\vee$ results in a distribution Ψ_r on the torus

$$(R^\vee \otimes_{\mathbb{Z}} \mathbb{R})/qR^\vee,$$

from which the errors are to be sampled.

We can now formulate the Ring-LWE problem precisely. Let $\mathfrak{U}(R_q)$ and $\mathfrak{U}(R_q^\vee)$ denote the uniform distributions on R_q and $R_q^\vee$, respectively. For $\mathbf{s} \in R_q^\vee$ and $r \in \mathbb{R}_{>0}$ we let $A_{\mathbf{s},r}$ be the distribution over

$$R_q \times (R_q^\vee \otimes_{\mathbb{Z}} \mathbb{R})/qR^\vee$$

obtained by sampling $\mathbf{a} \leftarrow \mathfrak{U}(R_q)$, $\mathbf{e} \leftarrow \Psi_r$ and returning $(\mathbf{a}, \mathbf{a} \cdot \mathbf{s} + \mathbf{e})$.

Definition 1 (Ring-LWE over a number field K with error parameter r). For a random but fixed choice of $\mathbf{s} \leftarrow \mathfrak{U}(R_q^\vee)$ the (search) *Ring-LWE* problem is to recover $\mathbf{s}$ with non-negligible probability from arbitrarily many independent samples from $A_{\mathbf{s},r}$.

Here it is understood that $r \geq 2\omega(\sqrt{\log n})$ for some superlinear function $\omega = \omega(n)$. It may seem surprising that this bound is less restrictive than in standard LWE, where one assumes $r = \sqrt{2\pi}\rho \geq 2\sqrt{n}$. But this is only superficial: the lattice of possible products $\mathbf{a} \cdot \mathbf{s}$ is much denser because $\mathbf{s}$ was sampled from $R_q^\vee$, and relative to this the Ring-LWE bound is considerably larger.

In their seminal paper [**19**] Lyubashevsky, Peikert and Regev provided the following hardness result. They actually deal with a slight variant called the Ring-LWE$_{\leq r}$ problem, where each sample is taken from $A_{\mathbf{s},\mathbf{r}}$ for some arbitrary fixed $\mathbf{r}$ taken from

$$\{(r_1, \ldots, r_n) \in (\mathbb{R}^+)^n \mid r_i \leq r \text{ for all } i = 1, \ldots, s \text{ and } r_{s+i} = r_{s+t+i} \leq r \text{ for all } i = 1, \ldots, t\}.$$

The distribution $A_{\mathbf{s},\mathbf{r}}$ is defined in roughly the same way as $A_{\mathbf{s},r}$, the main difference being that the spherical Gaussian Γ_r^n is to be replaced by the ellipsoidal Gaussian $\Gamma_{r_1}^1 \times \Gamma_{r_2}^1 \times \cdots \times \Gamma_{r_n}^1$. If we think of the error width $r \geq 2\omega(\sqrt{\log n})$ and the modulus $q \geq 2$ as quantities that vary with n, then the hardness result [**19**, Theorem 4.1] reads:

Theorem 2.1. *For some negligible $\varepsilon = \varepsilon(n)$ there is a probabilistic polynomial-time quantum reduction from* DGS$_\gamma$ *to* Ring-LWE$_{\leq r}$, *where*

$$\gamma : I \mapsto \max\left\{\eta_\varepsilon(I) \cdot (\sqrt{2}q/r) \cdot \omega(\sqrt{\log n}), \sqrt{2n}/\lambda_1(I^\vee)\right\}.$$

Here $\eta_\varepsilon(I)$ is the smoothing parameter of $\sigma(I)$ with threshold ε, and $\lambda_1(I^\vee)$ is the length of a shortest vector of $\sigma(I^\vee)$.

The statement involves the discrete Gaussian sampling problem DGS$_\gamma$, which is about producing samples from a spherical Gaussian in H with parameter r', restricted to the lattice $\sigma(I)$, for any given non-zero ideal $I \subset R$ and any $r' \geq \gamma(I)$. As discussed in [**19**] there are easy reductions from standard lattice problems to the discrete Gaussian sampling problem.

3. *SCG Ring-based LWE*

To allow for a common framework for Poly-LWE and Ring-LWE, from now on we restrict ourselves to number fields K for which the different ideal ∂ is principal, say generated by $\theta \in R$, so that $R^\vee = R/\theta$. This restriction is mainly for convenience: in general one can replace θ in the discussion below by a so-called *tweaking* factor, see [**9**, **21**]. But principality holds in most cases of interest. For instance if K is monogenic, meaning that the ring of integers R is of the form $\mathbb{Z}[x]/(f(x))$, then one can take $\theta = f'(x)$. More generally ∂ is principal if and only if R is a so-called complete intersection, i.e. of the form $\mathbb{Z}[x_1, x_2, \ldots, x_n]/(f_1, f_2, \ldots, f_n)$, in which case one can take $\theta = |(\partial f_i/\partial x_j)_{i,j}|$; see [**11**].

Without loss of generality we can rewrite our sample (2.2) as

$$\mathbf{a} \cdot \frac{\mathbf{s}}{\theta} = \frac{\mathbf{b}}{\theta} + \mathbf{e},$$

where now $\mathbf{s} \in R_q$ and $\mathbf{e}$ sampled from Ψ_r. Multiplying by θ then gives $\mathbf{b} = \mathbf{a} \cdot \mathbf{s} + \theta \cdot \mathbf{e}$. After fixing a $\mathbb{Z}$-basis $\alpha_1, \alpha_2, \ldots, \alpha_n$ of R we obtain

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + A_\theta \cdot B^{-1} \cdot \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}, \tag{3.1}$$

where the s_i are the coordinates of $\mathbf{s}$, the b_i are the coordinates of $\mathbf{b}$, $A_{\mathbf{a}}$ is the matrix of multiplication by $\mathbf{a}$, A_θ is the matrix of multiplication by θ, and $B = U^{-1} \cdot \Sigma$ with Σ the matrix of the canonical embedding σ, all expressed with respect to the basis $\alpha_1, \alpha_2, \ldots, \alpha_n$. Note that Σ is just the complex matrix having $\sigma(\alpha_1), \sigma(\alpha_2), \ldots, \sigma(\alpha_n)$ as its columns. The e_i are sampled independently from the univariate Gaussian Γ_r^1. The formula (3.1) is to be considered modulo q, but note that in the case of the subexpression $B^{-1} \cdot (e_1\, e_2\, \ldots\, e_n)^t$ it only makes sense to do so *after* elaborating the product. On average the factor $A_\theta \cdot B^{-1}$ causes the errors to expand, because $|\det A_\theta| = \Delta$ and $|\det B| = |\det \Sigma| = \mathrm{covol}(\sigma(R)) = \sqrt{|\Delta|}$, where $\Delta = |\Delta_K|$ is the absolute value of the discriminant of K; see [**15**].

Remark 1. In the monogenic case we can take $\theta = f'(x)$ and work with respect to the basis $1, x, \ldots, x^{n-1}$. For these choices we exactly recover (1.2), and we enter the discussion from the introduction. Note that $\Delta = |\mathrm{disc}\, f(x)|$ in this case.

Taking another generator θ of ∂ boils down to replacing the right-most term in (3.1), i.e. the vector of coordinates of the error term $\theta \cdot \mathbf{e}$, by

$$M \cdot A_\theta \cdot B^{-1} \cdot \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}$$

for some matrix $M \in \mathrm{GL}_n(\mathbb{Z})$. The same remark applies to switching to another basis of R, in which case M arises as the corresponding matrix of base change. In particular, if for one choice of basis a certain error coordinate is negligible, then for another choice of basis a certain non-trivial $\mathbb{Z}$-linear combination of the error coordinates will be negligible, and conversely.

Example 1. Let $\beta_1, \beta_2, \ldots, \beta_n$ be the basis of $R^\vee$ that is dual to our given basis $\alpha_1, \alpha_2, \ldots, \alpha_n$ of R with respect to the trace pairing. In other words $\sigma(\beta_1), \sigma(\beta_2), \ldots, \sigma(\beta_n) \in \mathbb{C}^n$ are the columns of $(\Sigma^t)^{-1}$. But then $\theta \cdot \beta_1, \theta \cdot \beta_2, \ldots, \theta \cdot \beta_n$ is also a basis of R, so we can change bases. In this case one verifies that the matrix of base change M is $\Sigma^t \cdot \Sigma \cdot A_\theta^{-1}$, with respect to which our Ring-LWE samples becomes

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + B^t \cdot U^t \cdot U \cdot \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}.$$

It is easy to check that $U^t \cdot U$ is a permutation matrix. Hence, we can rearrange the basis $\theta \cdot \beta_1, \theta \cdot \beta_2, \ldots, \theta \cdot \beta_n$ to obtain the following expression

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + B^t \cdot \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}. \tag{3.2}$$

If we would express the Ring-LWE samples directly in terms of the basis $\beta_1, \beta_2, \ldots, \beta_n$ of $R^\vee$ then we would find the same formula. Thus in some sense (3.2) is more in the actual spirit of [**19**] than (3.1), but it is less suited for discussing the SCG Ring-based LWE version from [**14**, **7**, **8**]. □

Recall from the introduction that the SCG Ring-based LWE from [**14**, **7**, **8**] leaves out the multiplication-by-θ step, and compensates for it by a scalar. Formally, one considers samples of the form

$$\mathbf{b} = \mathbf{a} \cdot \mathbf{s} + \lambda \cdot \mathbf{e}$$

where $\mathbf{s}$ is now taken from R_q rather than $R_q^\vee$. As before let $\mathbf{a} \leftarrow \mathfrak{U}(R_q)$ and $\mathbf{e} \leftarrow \Psi_r$, and let $\lambda \geq 1$ be a fixed real scalar. Let $A^\lambda_{\mathbf{s},r}$ be the resulting distribution over

$$R_q \times (R_q \otimes_{\mathbb{Z}} \mathbb{R})/qR$$

returning $(\mathbf{a}, \mathbf{a} \cdot \mathbf{s} + \lambda \cdot \mathbf{e})$. Then:

Definition 2 (SCG Ring-based LWE with scalar λ). For a random but fixed choice of $\mathbf{s} \leftarrow \mathfrak{U}(R_q)$ the problem is to recover $\mathbf{s}$ with non-negligible probability from arbitrarily many independent samples from $A^\lambda_{\mathbf{s},r}$.

When expressed with respect to a basis $\alpha_1, \alpha_2, \ldots, \alpha_n$ of R, such a sample converts into an expression of the form

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \lambda \cdot B^{-1} \cdot \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}, \tag{3.3}$$

Equivalently, one can also just remove the scalar λ and sample the errors e_i from $\Gamma_{\lambda \cdot r}$ instead of Γ_r. Here too switching to another basis amounts to multiplying the right-most factor from the left with a matrix $M \in \mathrm{GL}_n(\mathbb{Z})$.

As mentioned in the introduction, the authors of [**14**, **7**, **8**] took $\lambda = \Delta^{1/2n}$, while to us the most natural choice of scalar seems $\lambda = |\Delta|^{1/n}$, in order to compensate determinant-wise for the removal of A_θ. It would be interesting to know whether the latter choice allows for a hardness statement similar to Theorem 2.1. If A_θ happens to be a scalar matrix itself then both problems are of course equivalent. For instance this is the case if K is the 2^m-th cyclotomic field for some $m \geq 2$, where one can take $\lambda = \theta = 2^{m-1} = n$.

Example 2. To illustrate these different flavors of ring-based LWE, we analyze a simple example that will act as one of the building blocks in our main theorem. Let $d \equiv 1 \bmod 4$ be a positive squarefree integer and consider the real quadratic field $K = \mathbb{Q}(\sqrt{d})$. It has discriminant d and its ring of integers $R = \mathbb{Z}[(1 + \sqrt{d})/2]$ admits the integral basis $1, (1 + \sqrt{d})/2$. The different ideal ∂ is the principal ideal generated by $\theta = \sqrt{d}$. With respect to this basis one

has

$$A_\theta = \begin{pmatrix} -1 & \frac{-1+d}{2} \\ 2 & 1 \end{pmatrix}, \qquad \Sigma^{-1} = \frac{1}{\sqrt{d}} \begin{pmatrix} \frac{-1+\sqrt{d}}{2} & \frac{1+\sqrt{d}}{2} \\ 1 & -1 \end{pmatrix}, \qquad U = I_{2\times 2}.$$

So a Ring-LWE sample reads

$$\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} + A_\theta \cdot B^{-1} \cdot \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} + \begin{pmatrix} \frac{-1+\sqrt{d}}{2} & \frac{-1-\sqrt{d}}{2} \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} e_1 \\ e_2 \end{pmatrix},$$

while a SCG Ring-based LWE sample reads

$$\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} + \sqrt{d} \cdot B^{-1} \cdot \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} = A_{\mathbf{a}} \cdot \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} + \begin{pmatrix} \frac{-1+\sqrt{d}}{2} & \frac{1+\sqrt{d}}{2} \\ 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} e_1 \\ e_2 \end{pmatrix},$$

for scaling factor $\lambda = |\Delta|^{1/n} = \sqrt{d}$. □

For the sake of completeness let us conclude with the setting where one considers (3.1) with the *entire* matrix product $A_\theta \cdot B^{-1}$ replaced by a real scalar $\lambda \geq 1$. In the monogenic case $R = \mathbb{Z}[x]/(f(x))$ where one takes $\lambda = 1$ and works with respect to the basis $1, x, \ldots, x^{n-1}$, one recovers the Poly-LWE problem from the introduction. Note that in order to compensate for the removal of $A_\theta \cdot B^{-1}$ determinant-wise it is more natural to take $\lambda = \Delta^{1/2n}$ (here too it would be interesting to know whether the resulting problem enjoys a hardness proof). The more aggressive choice for $\lambda = 1$ may be motivated by the error bound in Regev's original work on LWE [**22**] where there is no number field at play, and by NTRU where the errors are bounded by a small constant. Taking smaller errors has advantages towards the efficiency of the resulting cryptosystems, but the security risks of doing so are not fully understood.

4. *Main theorem*

THEOREM 4.1. *Let $\rho : \mathbb{N} \to \mathbb{R}_{>0}$ be in* poly(n)*, let $(q_n)_{n\in\mathbb{N}}$ be any sequence of integer moduli, and let $\varepsilon \in \mathbb{R}_{>0}$ be fixed. Then there exists a family of number fields $(K_\ell)_{\ell\in\mathbb{N}}$ such that the following properties are satisfied:*

- *Each K_ℓ is Galois over $\mathbb{Q}$.*
- *The degree $n_\ell := [K_\ell : \mathbb{Q}]$ tends to infinity as ℓ does.*
- *Over K_ℓ the SCG Ring-based LWE problem with scalar $|\Delta_{K_\ell}|^{(1-\varepsilon)/n_\ell}$, error parameter $r = \rho(n_\ell)$ and modulus q_{n_ℓ} can be solved in time* poly$(n_\ell \cdot \log q_{n_\ell})$ *using $O(n_\ell)$ samples.*

The same statement is true for actual Ring-LWE as soon as one scales down the errors by a factor $|\Delta_{K_\ell}|^{\varepsilon/n_\ell}$.

REMARK 2. We certainly do not claim that *all* number fields become vulnerable after scaling inappropriately: the fields K_ℓ that will be constructed below are very special, in the sense that the lattices $\sigma(\mathcal{O}_{K_\ell})$ and $\sigma(\mathcal{O}_{K_\ell}^\vee)$ are extremely 'skew', in that they have widely varying successive minima. In particular our findings do not seem to apply to cyclotomic number fields, which are the main candidates for making their way to daily-life cryptography. Therefore the practical impact of Theorem 4.1 is limited.

PROOF OF THEOREM 4.1: Fix an $\ell \geq 2$ and pick prime numbers $p_1, \ldots, p_\ell$ congruent to 1 mod 4 such that

$$m_\ell := p_1 p_2 \cdots p_\ell \geq \left(2\sqrt{n_\ell}\rho(n_\ell)\sqrt{\log n_\ell}\right)^{2/\varepsilon}. \tag{4.1}$$

For each p_i consider the corresponding quadratic field $K_{\ell,i} = \mathbb{Q}(\sqrt{p_i})$. It has discriminant p_i and ring of integers $R_{\ell,i} = \mathbb{Z}[(1+\sqrt{p_i})/2]$, which we equip with the basis $\alpha_{i,1} = 1$, $\alpha_{i,2} = (1+\sqrt{p_i})/2$. We will analyze Ring-LWE in the field compositum

$$K_\ell = \mathbb{Q}(\sqrt{p_1}, \sqrt{p_2}, \ldots, \sqrt{p_\ell}) \cong K_{\ell,1} \otimes_{\mathbb{Q}} K_{\ell,2} \otimes_{\mathbb{Q}} \cdots \otimes_{\mathbb{Q}} K_{\ell,\ell},$$

which is clearly of degree $n_\ell := 2^\ell$. Because the discriminants p_i of $\mathbb{Q}(\sqrt{p_i})$ are mutually coprime this tensor structure carries over to the integral elements [**24**, Thm. 2.6], i.e. the ring R_ℓ of integers in K_ℓ reads

$$R_\ell = \mathbb{Z}[(1+\sqrt{p_1})/2, (1+\sqrt{p_2})/2, \ldots, (1+\sqrt{p_\ell})/2] \cong R_{\ell,1} \otimes_{\mathbb{Z}} R_{\ell,2} \otimes_{\mathbb{Z}} \cdots \otimes_{\mathbb{Z}} R_{\ell,\ell}.$$

Please do not confuse this notation with our previous notation R_q for the reduction of R mod q (in fact the modulus will not play an important role in the proof). Note that R_ℓ is a complete intersection, so the different ideal $\partial_\ell \subset R_\ell$ is generated by $\theta_\ell = \sqrt{p_1}\sqrt{p_2}\cdots\sqrt{p_\ell} = \sqrt{m_\ell}$. Therefore the codifferent reads

$$R_\ell^\vee = \frac{1}{\sqrt{m_\ell}}\mathbb{Z}[(1+\sqrt{p_1})/2, (1+\sqrt{p_2})/2, \ldots, (1+\sqrt{p_\ell})/2] \cong R_{\ell,1}^\vee \otimes_{\mathbb{Z}} R_{\ell,2}^\vee \otimes_{\mathbb{Z}} \cdots \otimes_{\mathbb{Z}} R_{\ell,\ell}^\vee,$$

i.e. it is again naturally compatible with the tensor structure of K_ℓ.

We begin with actual Ring-LWE, where we assume that the samples are expressed with respect to the product basis

$$\{\alpha_{1,i_1}\alpha_{2,i_2}\cdots\alpha_{\ell,i_\ell}\}_{\iota\in\{1,2\}^\ell}, \tag{4.2}$$

where ι abbreviates $(i_1, i_2, \ldots, i_\ell)$. With respect to this basis a Ring-LWE sample reads:

$$(b_\iota)_\iota^t = \mathbf{A}_\mathbf{a} \cdot (s_\iota)_\iota^t + A_{\theta_\ell} \cdot B^{-1} \cdot (e_\iota)_\iota^t. \tag{4.3}$$

Here $A_\mathbf{a}$ and A_{θ_ℓ} are the matrices of multiplication by $\mathbf{a}$ resp. $\theta_\ell = \sqrt{m_\ell}$ and $B^{-1} = \Sigma^{-1}$ is the inverse of the canonical embedding matrix; note that $U = I_{n_\ell \times n_\ell}$ because K_ℓ is totally real. We think of the e_ι's as being sampled independently from Γ_r^1 with $r = \rho(n_\ell)/|\Delta_{K_\ell}|^{\varepsilon/n_\ell}$, and the whole expression is considered modulo q_{n_ℓ}.

Because we work with respect to the product basis, the matrix $A_{\theta_\ell} \cdot B^{-1}$ arises as the Kronecker product of the corresponding matrices for the quadratic fields $K_{\ell,i}$, which by Example 2 are given by

$$\begin{pmatrix} \frac{-1+\sqrt{d}}{2} & \frac{-1-\sqrt{d}}{2} \\ 1 & 1 \end{pmatrix}.$$

Note that

$$\begin{pmatrix} 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \frac{-1+\sqrt{d}}{2} & \frac{-1-\sqrt{d}}{2} \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \end{pmatrix}, \tag{4.4}$$

so through the Kronecker product we find that

$$\begin{pmatrix} 0 & 0 & \ldots & 1 \end{pmatrix} \cdot A_{\theta_\ell} \cdot B^{-1} = \begin{pmatrix} 1 & 1 & \ldots & 1 \end{pmatrix},$$

where the row vector on the left has 0's everywhere, except at index $\iota = (2, 2, \ldots, 2)$ where it has a 1.

Thus given a Ring-LWE sample (4.3), we can multiply both sides from the left by the row vector $(0\ 0\ \ldots\ 1)$ in order to end up with a single linear equation in the secret $\mathbf{s} = (s_\iota)_\iota$, perturbed by an error of the form

$$\begin{pmatrix} 1 & 1 & \ldots & 1 \end{pmatrix} \cdot (e_\iota)_\iota^t,$$

which behaves as if it were sampled from a univariate Gaussian $\Gamma_{r'}^1$ with $r' = \sqrt{n_\ell} \cdot r$. Now our primes p_i have been chosen in such a way that this error is most likely negligible. More

precisely, our bound (4.1) on m_ℓ implies that

$$r' = \frac{\sqrt{n_\ell} \cdot \rho(n_\ell)}{|\Delta_{K_\ell}|^{\varepsilon/n_\ell}} = \frac{\sqrt{n_\ell} \cdot \rho(n_\ell)}{\sqrt{m_\ell}^{\varepsilon}} \leq \frac{1}{2\sqrt{\log n_\ell}},$$

whose absolute value is less than $1/2$ with overwhelming probability, so a mere rounding results in an *exact* linear equation in the secret. In fact by the lemma below, with very high probability we can successfully repeat this during n_ℓ consecutive rounds, to end up with an exact linear system of n_ℓ equations in the n_ℓ unknowns s_ι. This system is likely to have full rank (if not we can simply query a few more samples), so that the secret can be recovered using standard linear algebra over $\mathbb{Z}/q_{n_\ell}\mathbb{Z}$. This concludes the proof in the case of proper Ring-LWE.

To obtain the analogous result for the SCG Ring-based LWE using scaling factor $|\Delta_{K_\ell}|^{(1-\varepsilon)/n_\ell}$, one repeats the foregoing reasoning with $A_{\theta_\ell} \cdot B^{-1}$ replaced by $|\Delta_{K_\ell}|^{1/n} \cdot B^{-1}$. The analogue of (4.4) reads

$$\begin{pmatrix} 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \frac{-1+\sqrt{d}}{2} & \frac{1+\sqrt{d}}{2} \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & -1 \end{pmatrix},$$

leading to

$$\begin{pmatrix} 0 & 0 & \dots & 1 \end{pmatrix} \cdot |\Delta_{K_\ell}|^{1/n} \cdot B^{-1} = \left((-1)^{\eta(\iota)} \right)_\iota,$$

where $\eta(\iota)$ denotes the number of 2's appearing in $\iota \in \{1,2\}^\ell$. The right-hand side is again a norm $\sqrt{n_\ell}$ vector, which is the main ingredient needed for the rest of the proof to apply. □

LEMMA 4.2. *Let P_n denote the probability that n independent samples from the univariate Gaussian $\Gamma^1_{1/2\sqrt{\log n}}$ are all at most $1/2$ in absolute value. Then $P_n \to 1$ as $n \to \infty$.*

Proof. Write $r = 1/2\sqrt{\log n}$ and let z be sampled from Γ^1_r. Then P_n equals

$$\left(1 - 2P\left(z > \frac{1}{2}\right)\right)^n = \left(1 - \frac{2}{r}\int_{1/2}^{\infty} \exp\left(-\pi\frac{x^2}{r^2}\right)\right)^n \geq \left(1 - \frac{2}{r}\int_{1/2}^{\infty} 2x \exp\left(-\pi\frac{x^2}{r^2}\right)\right)^n$$

so

$$P_n \geq \left(1 - \frac{\exp\left(-\pi \log n\right)}{\pi\sqrt{\log n}}\right)^n,$$

where the right hand side is seen to converge to 1 using l'Hôpital's rule. □

REMARK 3. The fields K_ℓ that were constructed in the above proof are totally real, but this is not essential. Indeed, if we would also allow primes $p_i \equiv 3 \bmod 4$ and instead consider the field

$$K_\ell = \mathbb{Q}(\sqrt{p_1^*}, \sqrt{p_2^*}, \dots, \sqrt{p_\ell^*}),$$

where

$$p_i^* = (-1)^{\frac{p_i-1}{2}} p_i,$$

then the same conclusions would have followed.

As was pointed out to us by a reviewer, the part of Theorem 4.1 that deals with actual Ring-LWE admits an easier and more broadly applicable proof. Just pick number fields having large enough discriminants, such as the ones constructed in the above proof, and apply the following observation:

THEOREM 4.3. *Let $(K_n)_{n\in\mathbb{N}}$ be a family of number fields of increasing degree n, let $\rho : \mathbb{N} \to \mathbb{R}_{>0}$ be in* poly(n)*, and let $(q_n)_{n\in\mathbb{N}}$ be any sequence of integer moduli. Then the Ring-LWE problem in K_n with error parameter $r = \rho(n)$ can be solved in time* poly$(n \cdot \log q_n)$ *using $O(n)$ samples as soon as the errors are being scaled down by at least $2\rho(n)\sqrt{n \log n}$.*

Proof. This is based on the simple fact that $1 \in R$, which implies that $\sigma(R)$ always contains the vector $(1, 1, \ldots, 1)$. Thus there always exists a $\mathbb{Z}$-linear combination of the column vectors of $\Sigma = U \cdot B$ having norm $\sqrt{n}$. As a consequence the same $\mathbb{Z}$-linear combination of the rows of B^t is of said norm, meaning that given a Ring-LWE sample as in (3.2), one can extract from it a linear equation in the coordinates of the secret $s_1, s_2, \ldots, s_n$ that is perturbed by an error sampled from $\Gamma^1_{\rho(n)\cdot\sqrt{n}}$. As soon as one scales this down by a factor of size at least $2\rho(n)\sqrt{n \log n}$, by the previous lemma about $O(n)$ samples suffice to recover $\mathbf{s}$. □

5. *A cyclotomic point of view*

The fields K_ℓ constructed in the previous section are abelian, more precisely they are Galois with Galois group

$$\mathrm{Gal}(K_\ell/\mathbb{Q}) \cong C_2 \times C_2 \times \cdots \times C_2,$$

where C_2 denotes the group of order two. So by the Kronecker-Weber theorem it should be a subfield of some cyclotomic field. The following lemma shows that it is a subfield of $K := \mathbb{Q}(\zeta_{m_\ell})$. We identify the Galois group $\mathrm{Gal}(K/\mathbb{Q})$ with $G := (\mathbb{Z}/(m_\ell))^\times$, where $a \in G$ acts on K as $\zeta_{m_\ell} \mapsto \zeta^a_{m_\ell}$.

LEMMA 5.1. *Let G^2 be the subgroup of squares in G. Then K_ℓ is the subfield of K fixed by G^2.*

Proof. Denote the subfield of K fixed by G^2 as K^{G^2}. For each $c \in G/G^2$ consider

$$w_c = \mathrm{Tr}_{K/K^{G^2}}(\zeta^c_{m_\ell}) = \sum_{h\in G^2} \zeta^{hc}_{m_\ell} \ \in \ K^{G^2}.$$

By the Chinese remainder theorem (CRT) we have the isomorphism

$$G \cong \mathbb{F}^\times_{p_1} \times \mathbb{F}^\times_{p_2} \times \cdots \times \mathbb{F}^\times_{p_\ell},$$

according to which the w_c's can be decomposed as follows:

$$w_c = \sum_{h\in G^2} \zeta^{hc}_{m_\ell} = \sum_{\substack{h_1\in(\mathbb{F}^\times_{p_1})^2 \\ \cdots \\ h_\ell\in(\mathbb{F}^\times_{p_\ell})^2}} \zeta^{h_1 c}_{p_1}\zeta^{h_2 c}_{p_2}\cdots\zeta^{h_\ell c}_{p_\ell} = \prod_{i=1}^{\ell} \sum_{h\in(\mathbb{F}^\times_{p_i})^2} \zeta^{hc}_{p_i}. \tag{5.1}$$

Every sum in the last product is a so-called Gaussian period, where the exponents run through either the quadratic residues or the quadratic non-residues modulo p_i. As all p_i's are congruent to 1 modulo 4, such sums result in

$$\beta_{i,1} := \frac{-1+\sqrt{p_i}}{2}, \quad \text{resp.} \quad \beta_{i,-1} := \frac{-1-\sqrt{p_i}}{2}$$

(see [**10**]). One sees that $\{w_c\}_c$ is the product basis of K_ℓ obtained by equipping the $R_{\ell,i}$'s with the $\mathbb{Z}$-bases $\beta_{i,1}, \beta_{i,-1}$ rather than $\alpha_{i,1}, \alpha_{i,2}$. In particular the w_c's generate K_ℓ, so $K_\ell \subset K^{G^2}$ and the lemma follows by comparing degrees. □

As a byproduct of the above proof, we obtain that the w_c's form a $\mathbb{Z}$-basis of R_ℓ, which is a special case of a more general statement [**18**, Prop. 6.1]. This kind of 'trace basis' is also used in the recent work on SCG Ring-based LWE by Chen, Lauter and Stange [**7**], an example of which we will analyze later in this section. It is interesting to have a quick look at our proof of Theorem 4.1, where now we express the samples with respect to the basis $\{w_c\}_c$, instead of (4.2). Here the factors in the Kronecker product decomposition of $A_{\theta_\ell} \cdot B^{-1}$ read

$$\begin{pmatrix} \frac{-1-p_i}{2} & \frac{-1+p_i}{2} \\ \frac{1-p_i}{2} & \frac{1+p_i}{2} \end{pmatrix} \cdot \frac{1}{\sqrt{p_i}} \begin{pmatrix} \frac{1-\sqrt{p_i}}{2} & \frac{-1-\sqrt{p_i}}{2} \\ \frac{-1-\sqrt{p_i}}{2} & \frac{1-\sqrt{p_i}}{2} \end{pmatrix} = \begin{pmatrix} \frac{1-\sqrt{p_i}}{2} & \frac{1+\sqrt{p_i}}{2} \\ \frac{-1-\sqrt{p_i}}{2} & \frac{-1+\sqrt{p_i}}{2} \end{pmatrix}.$$

One sees that

$$\begin{pmatrix} 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} \frac{1-\sqrt{p_i}}{2} & \frac{1+\sqrt{p_i}}{2} \\ \frac{-1-\sqrt{p_i}}{2} & \frac{-1+\sqrt{p_i}}{2} \end{pmatrix} = \begin{pmatrix} 1 & 1 \end{pmatrix}.$$

So expanding the Kronecker product gives

$$(J(\iota))_\iota \cdot A_{\theta_\ell} \cdot B^{-1} = \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix}, \tag{5.2}$$

where ι runs over all tuples $(i_1, i_2, \dots, i_\ell) \in \{1, -1\}^\ell$ and

$$J(\iota) = J(i_1, i_2, \dots, i_\ell) = \prod_{j=1}^{\ell} i_j$$

(this formula explains why we indexed the β_i's by ± 1 rather than $1, 2$). The row vector $(1\ 1\ \dots\ 1)$ on the right-hand side of (5.2) has norm $\sqrt{n_\ell}$, so as before this can be used to obtain linear equations in the coordinates of the secret $\mathbf{s}$ that carry negligible error terms, allowing one to recover $\mathbf{s}$ by means of simple linear algebra.

Remark 4. As before, the same claims apply to non-dual Ring-LWE and/or to the setting where we allow primes $p_i \equiv 3 \bmod 4$, upon replacement of every appearance of $\sqrt{p_i}$ by $\sqrt{p_i^*}$.

Remark 5. The letter J refers to the Jacobi-symbol. Indeed, through the CRT we have

$$G/G^2 \cong \frac{\mathbb{F}_{p_1}^\times}{(\mathbb{F}_{p_1}^\times)^2} \times \frac{\mathbb{F}_{p_2}^\times}{(\mathbb{F}_{p_2}^\times)^2} \times \dots \times \frac{\mathbb{F}_{p_\ell}^\times}{(\mathbb{F}_{p_\ell}^\times)^2} = \{\pm 1\} \times \{\pm 1\} \times \dots \times \{\pm 1\},$$

where if $c \in G/G^2$ corresponds to $\iota = (i_1, i_2, \dots, i_\ell) \in \{1, -1\}^\ell$, then $w_c = \beta_{1,i_1}\beta_{2,i_2}\cdots\beta_{\ell,i_\ell}$ and $J(\iota) = (c/m_\ell)$. Thus if we prefer to think of the rows and columns of the matrices A_θ and M as being indexed by $c \in G/G^2$ rather than $\iota \in \{1, -1\}^\ell$, then (5.2) becomes

$$\left(\left(\tfrac{c}{m_\ell}\right)\right)_c \cdot A_{\theta_\ell} \cdot M^{-1} = \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix},$$

an identity which we found remarkable at first sight.

To conclude this article, we note that more generally, the presence of factors of the form $\mathbb{Z}[(1+\sqrt{d})/2]$ for some $d \equiv 1 \bmod 4$ may lead to unexpectedly short linear combinations of the rows of B^{-1} and $A_\theta \cdot B^{-1}$, and thus to weaker instances of ring-based LWE than one might hope (for an aggressive choice of scaling factor).

For instance, let us analyze the first example listed in [**7**, §5.1]; the other examples admit a similar analysis. Here Chen et al. let $m = 2805 = 3 \cdot 5 \cdot 11 \cdot 17$ and they consider the fixed field $K^{G'}$ of $K = \mathbb{Q}(\zeta_m)$ under the action of

$$G' := \langle 1684, 1618 \rangle \subset G = \mathrm{Gal}(K/\mathbb{Q}) = (\mathbb{Z}/(m))^\times.$$

Under the CRT decomposition $(\mathbb{Z}/(m))^\times \cong \mathbb{F}_3^\times \times \mathbb{F}_{11}^\times \times (\mathbb{Z}/(85))^\times$ this subgroup corresponds to $\{1\} \times \{1\} \times G'_{85}$ where G'_{85} denotes the index two subgroup of elements having Jacobi symbol 1. We again work with respect to the trace basis

$$w_c = \sum_{h \in G'} \zeta_m^{hc} = \zeta_3^c \cdot \zeta_{11}^c \cdot \sum_{h \in G'_{85}} \zeta_{85}^{hc},$$

where $c \in G/G'$. The latter sum equals $\beta_1 := (1 + \sqrt{85})/2$ or $\beta_{-1} := (1 - \sqrt{85})/2$ depending on whether $\left(\frac{c}{85}\right) = 1$ or not. So we conclude similarly as before that the ring of integers equals

$$R := \mathcal{O}_{K^{G'}} = \mathbb{Z}[\zeta_3, \zeta_{11}, (1 + \sqrt{85})/2] \cong \mathbb{Z}[\zeta_3] \otimes_{\mathbb{Z}} \mathbb{Z}[\zeta_{11}] \otimes_{\mathbb{Z}} \mathbb{Z}[(1 + \sqrt{85})/2]$$

and that $\{w_c\}_c$ is the product basis

$$\{\zeta_3^i \zeta_{11}^j \beta_k\}_{\substack{i=1,2 \\ j=1,2,\ldots,10 \\ k=1,-1}}.$$

As in [**7**], let us have a look at SCG Ring-based LWE with scaling factor $|\Delta|^{1/2n}$, where $\Delta = \Delta_{K^{G'}} = (-3) \cdot (-11^9) \cdot 85$ and $n = [K^{G'} : \mathbb{Q}] = 40$. Let M denote the matrix of the canonical embedding of $K^{G'}$ with respect to the above basis. Then the last Kronecker factor of $|\Delta|^{1/2n} \cdot M^{-1} = |\Delta|^{1/80} \cdot M^{-1}$ is given by

$$\frac{1}{\sqrt[4]{85}} \cdot \begin{pmatrix} \frac{1+\sqrt{85}}{2} & \frac{-1+\sqrt{85}}{2} \\ \frac{-1+\sqrt{85}}{2} & \frac{1+\sqrt{85}}{2} \end{pmatrix}.$$

So multiplying from the left by $(1 \ -1)$ leads to the row vector $(1 \ 1)/\sqrt[4]{85}$ of norm ≈ 0.4658, which is 'unexpectedly short'. The other Kronecker factors correspond to cyclotomic fields and have less surprising behavior. Here taking the first row (for instance) of each factor leads to norms $\sqrt{2}/\sqrt[4]{3} \approx 1.0746$ and $\sqrt{10}/\sqrt[20]{11^9} \approx 1.0750$, respectively. Thus multiplying $|\Delta|^{1/80} \cdot B^{-1}$ from the left by

$$(1, 0) \otimes (1, 0, 0, 0, 0, 0, 0, 0, 0, 0) \otimes (1, -1)$$

yields a row vector of norm $\approx 1.0746 \cdot 1.0750 \cdot 0.4658 \approx 0.5381$. Since Chen et al. let $r = 1$, this results in a linear equation in the secret $\mathbf{s}$ carrying an error term sampled from $\Gamma^1_{0.5381}$, roughly. By taking other rows of the cyclotomic parts one in fact finds 20 independent such equations. This is insufficient to break this concrete instance of SCG Ring-based LWE using mere rounding (a substantial number of equations will carry an error that exceeds $1/2$ in absolute value), but it is tight, so it provides an explanation why this was indirectly helpful for Chen et al. to successfully apply their χ^2-analysis.

6. *Conclusion*

In this paper we explained that *if* one wishes to set up a SCG Ring-based LWE in a degree n number field K, as was done in [**14**, **7**, **8**] in the context of potential attacks involving evaluation at 1, then it is natural to scale up the errors by $|\Delta_K|^{1/n}$. More precisely we proved that for each $\varepsilon > 0$ scaling up by $|\Delta_K|^{(1-\varepsilon)/n}$ may indeed be insufficient, in the sense that there exist number fields for which the corresponding problem is easily broken. These observations also apply to proper Ring-LWE, in the sense that scaling down by $|\Delta_K|^{\varepsilon/n}$ leads to vulnerable families for any $\varepsilon > 0$. Some of our families implicitly exploit the structure of the Galois group, which raises the question to what extent the structure of the Galois group can be exploited further in the analysis of the hardness of Ring-LWE.

In any case we stress that the families constructed in this paper are very special. In particular it is unlikely that they will ever be used in a cryptographic context. Our main aim is to help delimit the room of flexibility there is in tweaking the parameters (or even the definition) of

Ring-LWE as it was introduced in [**19**]. We refer the interested reader to the recent work of Peikert [**21**] that has appeared in the meantime, in which a unifying framework is given.

References

1. J. Bos, K. Lauter, J. Loftus, M. Naehrig, 'Improved security for a ring-based fully homomorphic encryption scheme', 14*th IMA Conference on Cryptography and Coding* (2013)
2. Z. Brakerski, C. Gentry, V. Vaikunthanathan, '(Leveled) Fully Homomorphic Encryption without Bootstrapping', *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference – ITCS '12*, pp. 309-325 (2012)
3. Z. Brakerski, A. Langlois, C. Peikert, O. Regev and D. Stehlé, 'Classical hardness of learning with errors', *ACM Symposium on the Theory of Computing – STOC '13*, pp. 575-584 (2013)
4. Z. Brakerski, V. Vaikunthanathan, 'Fully homomorphic encryption from Ring-LWE and security for key dependent messages', *Advances in Cryptology – CRYPTO '11*, Lecture Notes in Computer Science 6841, pp. 505-524 (2011)
5. Z. Brakerski, V. Vaikunthanathan, 'Efficient Fully Homomorphic Encryption from (Standard) LWE', *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science – FOCS '11*, pp. 97-106 (2011)
6. W. Castryck, I. Iliashenko, F. Vercauteren, 'Provably weak instances of Ring-LWE revisited', *Advances in Cryptology – EUROCRYPT 2016*, Lecture Notes in Computer Science 9665(1), pp. 147-167 (2016)
7. H. Chen, K. Lauter, K. Stange, 'Attacks on search RLWE', Cryptology ePrint Archive 2015/971 (2015)
8. H. Chen, K. Lauter, K. Stange, 'Vulnerable Galois RLWE families and improved attacks', Cryptology ePrint Archive 2016/193 (2016)
9. E. Crockett, C. Peikert, '$\Lambda \circ \lambda$: A functional library for lattice cryptography', Cryptology ePrint Archive 2015/1134 (2015)
10. H. Davenport, *Multiplicative number theory*, 2nd edition (revised by H. Montgomery), Graduate Texts in Mathematics 74 (Springer, 2000)
11. B. de Smit, 'A differential criterion for complete intersections', *Journées Arithmétiques* 1995, Collectanea Mathematica 48 (1-2), pp. 85-96 (1997)
12. L. Ducas, A. Durmus, 'Ring-LWE in polynomial rings', *Public Key Cryptography – PKC '12*, Lecture Notes in Computer Science 7293, pp. 34-51 (2012)
13. K. Eisenträger, S. Hallgren, K. Lauter, 'Weak instances of PLWE', *Selected Areas in Cryptography – SAC 2014*, Lecture Notes in Computer Science 8781, pp. 183-194 (2014)
14. Y. Elias, K. Lauter, E. Ozman, K. Stange, 'Provably weak instances of Ring-LWE', *Advances in Cryptology – CRYPTO '15*, Lecture Notes in Computer Science 9215, pp. 63-92 (2015)
15. A. Fröhlich, M. Taylor, *Algebraic number theory*, Cambridge Studies in Advances Mathematics 27, (Cambridge University Press, 1991)
16. C. Gentry, 'Key recovery and message attacks on NTRU-Composite', *EUROCRYPT '01*, Lecture Notes in Computer Science 2045, pp. 182-194 (2001)
17. J. Hoffstein, J. Pipher, J. H. Silverman, 'NTRU: A Ring-Based Public Key Cryptosystem', *Proceedings of the Third International Symposium on Algorithmic Number Theory – ANTS-III*, Lecture Notes in Computer Science 1423, pp. 267-288 (1998)
18. H. Johnston, 'Notes on Galois modules', Notes accompanying the course 'Galois Modules' given in Cambridge (2011)
19. V. Lyubashevsky, C. Peikert, O. Regev, 'On ideal lattices and learning with errors over rings', *Journal of the ACM* 60(6), article 43, 35 pp. (2013)
20. C. Peikert, 'Public-key cryptosystems from the worst-case shortest vector problem', *ACM Symposium on the Theory of Computing – STOC '09*, pp. 333-342 (2009)
21. C. Peikert, 'How (not) to instantiate Ring-LWE', Cryptology ePrint Archive 2016/351 (2016)
22. O. Regev, 'On lattices, learning with errors, random linear codes, and cryptography', *Journal of the ACM* 56(6), article 34, 40 pp. (2009)
23. P. Shor, 'Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer', *SIAM Journal of Computing* 26(5), pp. 1484-1509 (1997)
24. L. Washington, *Introduction to cyclotomic fields*, Graduate Texts in Mathematics 83 (Springer, 1982)

Chapter 8

Privacy-Friendly Forecasting for the Smart Grid Using Homomorphic Encryption and the Group Method of Data Handling

Publication data

Bos, J. W., Castryck, W., Iliashenko, I., and Vercauteren, F. Privacy-friendly forecasting for the smart grid using homomorphic encryption and the group method of data handling. In *AFRICACRYPT 17: 9th International Conference on Cryptology in Africa* (May 2017), M. Joye and A. Nitaj, Eds., vol. 10239 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 184–201.

Privacy-friendly Forecasting for the Smart Grid using Homomorphic Encryption and the Group Method of Data Handling

Joppe W. Bos[1], Wouter Castryck[2,3], Ilia Iliashenko[2], and Frederik Vercauteren[2,4]

[1] NXP Semiconductors
[2] imec-Cosic, Dept. Electrical Engineering, KU Leuven
[3] Laboratoire Paul Painlevé, Université de Lille-1
[4] Open Security Research

Abstract. While the smart grid has the potential to have a positive impact on the sustainability and efficiency of the electricity market, it also poses some serious challenges with respect to the privacy of the consumer. One of the traditional use-cases of this privacy sensitive data is the usage for forecast prediction. In this paper we show how to compute the forecast prediction such that the supplier does not learn any individual consumer usage information. This is achieved by using the Fan-Vercauteren somewhat homomorphic encryption scheme. Typical prediction algorithms are based on artificial neural networks that require the computation of an activation function which is complicated to compute homomorphically. We investigate a different approach and show that Ivakhnenko's group method of data handling is suitable for homomorphic computation.
Our results show this approach is practical: prediction for a small apartment complex of 10 households can be computed homomorphically in less than four seconds using a parallel implementation or in about half a minute using a sequential implementation. Expressed in terms of the mean absolute percentage error, the prediction accuracy is roughly 21%.

1 Introduction

One of the promising solutions to cope with current and future challenges of electricity supply is the *smart grid*. With the prospect of having a positive impact on the sustainability, reliability, flexibility, and efficiency many countries around the world are investing significantly in such smart grid solutions. The deployment of smart meters is already well underway. For example, in the United Kingdom the large energy suppliers were operating over $400,000$ smart gas and electricity meters, representing 0.9 percent of all the domestic meters operated by the large

This work was supported by the European Commission under the ICT programme with contract H2020-ICT-2014-1 644209 HEAT, and through the European Research Council under the FP7/2007-2013 programme with ERC Grant Agreement 615722 MOTMELSUM.

suppliers in 2014 [9]. This development is expected to continue and intensify: the EU third energy package has as an objective to replace at least 80 percent of electricity meters with smart meters by 2020 [15]. This change will fundamentally re-engineer the (electricity) service industry.

The replacement of the classical meters with their smart variants has advantages for both the consumer and industry. Some of the key benefits include giving consumers the information to gain control over their energy consumption, lowering the cost for managing the supply of energy across industry, and producing detailed consumption information data from these smart meters which in turn enable a wide range of services [9]. It is expected that the meters have an update rate of every 15 minutes at least [14]. When generating such a large amount of consumer data a lot of privacy sensitive information is being disclosed. There are various initiatives (e.g. [32,37]) which stress and outline the importance of having solutions for the smart grid where privacy protecting mechanisms are already built-in by design.

This work is concerned with enhancing the privacy of the smart meter readings in the setting of *forecast prediction*: energy suppliers need to forecast in order to buy energy generation contracts that cover their clients. Moreover, to ensure network capacity the network operators require longer term forecasting [23,37,10]. This forecasting is typically done by taking as input the (aggregated) data from a number of households. Based on this consumption data, together with other variables such as the date and the current temperature and weather, a forecast is computed to predict the short, medium, or long term consumption. The energy providers or network operators only need to know the desired forecast information based on their (potentially proprietary) forecasting algorithm and model. There is no need to observe the individual consumer data.

We investigate the potential of *fully homomorphic encryption* (FHE) to realize this goal. The notion of FHE was introduced in the late 1970s [34] and a concrete instantiation was found in 2009 by Gentry [19]. FHE allows an untrusted party to carry out arbitrary computation on *encrypted data* without learning anything about the content of this data. Currently, the Fan-Vercauteren (FV) FHE scheme [16] is regarded as the best choice with respect to security and practical performance. See Section 4 for a more detailed description of the FV scheme. Additively homomorphic encryption schemes [31] and other tools have been proposed to enhance the privacy in the setting of computing detailed billing in the context of the smart grid [33,30,18,25,13,24]. However, these approaches cannot be directly used in the setting of prediction algorithms since these more complex algorithms need to compute both additions and multiplications.

One popular class of algorithms which are used for prediction are based on artificial neural networks. One of the main ingredients in these forecasting algorithms is the computation of the so-called activation function, in practice it is common to use a sigmoid function where the logistic function $t \mapsto 1/(1+e^{-t})$ is a popular choice. However, computing such a sigmoid function homomorphically is far from practical. One possible way to proceed is to simply ignore the sigmoidality requirement and to proceed with a truncated Taylor series ap-

proximating this function or, more generally, to use any non-linear polynomial function which is *simple*. This was investigated by Livni et al. [26] regardless of cryptographic applications. Recent work by Xie et al. [39] and Dowlin et al. [12] suggests to apply the same approach to homomorphically encrypted data. However, by computing artificial neural networks in this fashion it becomes just an organized manner of fitting a polynomial through the given data set. In this paper we investigate an older tool for realizing this goal. Namely, we show that Ivakhnenko's group method of data handling (GMDH) which was proposed back in 1970 [22] is a perfect match for being computed homomorphically. Moreover, a recent comparison analysis between different forecasting methods [36] showed that GMDH produced significantly more accurate results compared to the other methods considered.

We show that GMDH can be implemented homomorphically using the recent fixed point approach from [11,6]. Using a five-layered network (one input layer, three hidden layers and an output node) we are able to homomorphically predict the next half-hour energy consumption for an apartment complex of 10 households. Our software implementation results indicate that this requires less than four seconds using a parallel implementation or about half a minute using a sequential implementation while the prediction accuracy expressed using the mean absolute percentage error (MAPE, see Section 3 for a definition) is only 21 percent. This shows that privacy preserving forecasting using homomorphic encryption is indeed practical.

2 The Smart Grid and Privacy Concerns

The authors of [35] define the smart grid as "*an electricity network that can cost efficiently integrate the behavior and actions of all users connected to it – generators, consumers and those that do both – in order to ensure economically efficient, sustainable power system with low losses and high levels of quality and security of supply and safety*". This paper is concerned with the cryptographic solutions to privacy concerns within the smart grid. Within this scope we assume that the meters are protected against various types of side-channel attacks such that no secret data can be retrieved from the device when it is operating (e.g. key extraction). Moreover, we assume that the smart metering device acts honestly in accordance with the implementation or protocol given to it. These assumptions avoid the usual security threats and leave us with the privacy related concerns which we aim to address.

In the early 1990s, Hart showed a non-intrusive approach where by monitoring the electric load one can observe the individual appliances turning on and off [20]. Hence, detailed smart meter readings, which are expected to be generated at least every 15 minutes in the context of the smart grid (cf. [14]), can be used to derive various privacy sensitive information about a house-hold or even an apartment complex. In order to grasp where the main privacy challenges are in smart metering it is good to understand how and when the meter readings are used in practice by the various parties involved. As identified by the survey

paper [23], which in turn has collected this information from the privacy impact assessment by NIST [37] and the enumeration of data uses by the consultation of the British Department of Energy and Climate Change [10], the key usages of smart meter readings include the usage for *load monitoring and forecasting* and *smart billing*.

There has been a significant amount of work related to privacy-preserving smart billing solutions for the smart grid. One line of research allows complex non-linear tariff policies where the bill is computed and sent along with a zero-knowledge proof to ensure that the computations are correct [33,30]. Another approach is based on privacy-friendly aggregation schemes (e.g. using additively homomorphic encryption schemes such as the Paillier scheme [31]) where one can compute a function on the ciphertexts which corresponds to adding the plaintexts [18,25,13,24]. Such approaches heavily rely on the fact that only aggregation of the results is required. As soon as more complex operations need to be computed (such as a large number of multiplications) one has to look for other solutions.

One example where more complex operations are performed is in the setting of load monitoring and forecasting. There are many different forecasting approaches (see e.g. the survey paper [21] on this topic and the references therein). One of the popular and well-studied techniques is using artificial neural networks (see e.g. [1,17]). In the next section we describe how such neural networks operate, analyze the challenges they pose when being evaluated in the encrypted domain, and discuss how this naturally leads to considering the group method of data handling as an alternative forecasting tool.

3 Neural Networks versus The Group Method of Data Handling

Over time, artificial neural networks (ANNs) have manifested themselves among the most popular and reliable prediction tools for various purposes, including load forecasting. For our preliminary discussion, it suffices to think of an ANN as a real-valued function $f : \mathbf{R}^{n_0} \to \mathbf{R}$ that arises as the composition of a number of 'neurons' $\nu_{ij} : \mathbf{R}^{n_{i-1}} \to \mathbf{R}$, organized in layers $i = 1, \ldots, r$, as depicted in Figure 1. Each neuron is of the form

$$\nu_{ij} : \mathbf{R}^{n_{i-1}} \to \mathbf{R} : (x_1, x_2, \ldots, x_{n_{i-1}}) \mapsto g\left(\sum_{k=1}^{n_{i-1}} w_{ijk} x_k - b_{ij}\right)$$

for weights and biases $w_{ijk}, b_{ij} \in \mathbf{R}$ and some fixed sigmoidal activation function $g : \mathbf{R} \to \mathbf{R}$, such as the logistic function $t \mapsto 1/(1 + e^{-t})$. The global shape of the network is decided in advance, and the goal is to determine the weights w_{ijk} and the biases b_{ij} such that f approximates an unknown target function $\tilde{f} : \mathbf{R}^{n_0} \to \mathbf{R}$, in our case load prediction, as well as possible. This is done during a so-called supervised learning phase. One starts from a reasonable guess, after

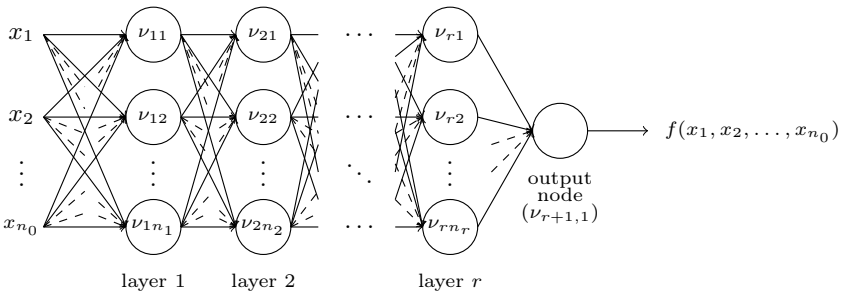


Fig. 1. Design of an Artificial Neural Network (ANN).

which the network's performance is assessed by feeding to it a number of input-output pairs of $\tilde{f}$, taken from a given data set, and measuring the error. During a process called backpropagation, which is based on the chain rule for derivation, the weights and biases are then modified repeatedly, in the hope of converging to values that minimize the error.

The backpropagation method requires the activation function g to have a nice and easy derivative, while at the same time it should be sigmoidal, i.e. its graph should have the typical step-like activation shape, allowing the ANNs to do what they were designed for: to simulate computation in (an area of) the human brain. Unfortunately, the class of such functions does not contain examples that are easy to evaluate homomorphically. A natural attempt would be to use a Taylor approximation to the logistic function or to one of its known alternatives, but such approximations become highly non-sigmoidal away from the origin.

One way out is simply to ignore the sigmoidality requirement and to proceed with this truncated Taylor series, or more generally to replace g by any simple non-linear polynomial function, the easiest choice being $t \mapsto t^2$. This has been investigated by Livni et al. [26] for reasons of computational efficiency, regardless of cryptographic applications. Recent work by Xie et al. [39] and Dowlin et al. [12] suggested to apply the same approach to homomorphically encrypted data. The resulting neural networks were named 'crypto-nets'.

However in this way the ANN just becomes an organized way of fitting a polynomial through the given data set. There exist older and simpler prediction tools that do this. In this paper we study one of the oldest such tools, namely Ivakhnenko's group method of data handling (GMDH) from 1970 [22]. Besides being suited for applications using homomorphic encryption, one particular feature is that its performance in the context of load forecasting enjoys a large amount of existing literature, at times even with results that are superior to ANNs. Indeed, a comparison analysis between different forecasting methods from 2008 [36] showed that GMDH produced significantly more accurate results compared to the other methods considered.

The basic version of GMDH works as follows, although many variations are possible (and seem to deserve a further analysis). The goal is to approximate

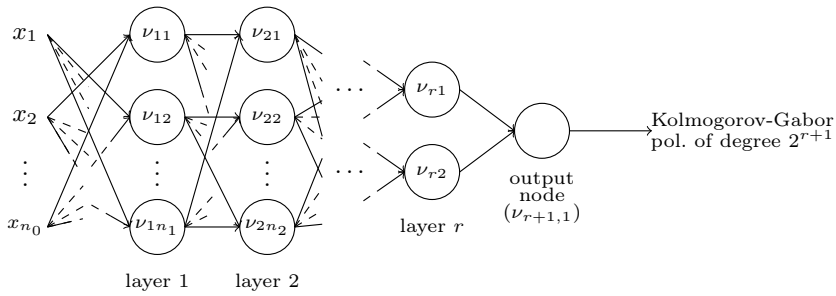


Fig. 2. Network-like illustration of the Group Method of Data Handling.

our target function $\tilde{f} : \mathbf{R}^{n_0} \to \mathbf{R}$ with a truncated Wiener series

$$a_0 + \sum_{i=1}^{n_0} a_i x_i + \sum_{i=1}^{n_0} \sum_{j=i}^{n_0} a_{ij} x_i x_j + \sum_{i=1}^{n_0} \sum_{j=i}^{n_0} \sum_{k=j}^{n_0} a_{ijk} x_i x_j x_k + \ldots,$$

which is also called a Kolmogorov-Gabor polynomial. The idea is to approach this by a finite superposition of quadratic polynomials

$$\nu_{ij} : \mathbf{R}^2 \to \mathbf{R} : (x, y) \mapsto b_{ij0} + b_{ij1} x + b_{ij2} y + b_{ij3} xy + b_{ij4} x^2 + b_{ij5} y^2$$

along a diagram of the kind depicted in Figure 2. One can think of this as some sort of ANN, and indeed the diagram is sometimes called a 'polynomial neural network'. As a first main difference, however, note that the wiring is incomplete: each neuron has two inputs only.

Also the learning phase is quite different from the one in conventional ANNs. Here the goal is to determine the coefficients b_{ijk} of the quadratic polynomials ν_{ij}, but also the concrete structure of the network, which is not fixed in advance. Indeed, one decides beforehand on the number of layers r and the number of neurons n_i in each layer, but the wiring between these is defined during the learning process. Recall that each node can have only two inputs, so the following constraint should be satisfied: $n_i \leq \binom{n_{i-1}}{2}$. In order to prevent exponential growth of the number of neurons, the left hand side will in general be much smaller than the right hand side. As to *which* combinations end up being chosen, one first considers all possible combinations and then removes the $\binom{n_{i-1}}{2} - n_i$ worst neurons with respect to their error performance, in the sense explained below, while at the same time determining the coefficients b_{ijk} of the surviving neurons. One then proceeds with the next layer. In particular, there is no backpropagation. The node with the smallest error performance will be assigned as an output for the whole network; this may in fact be different from what was initially foreseen to become the output neuron. One sometimes applies the rule that if at some point all nodes in layer i perform worse than the best performing node in layer $i-1$, then the algorithm stops, and the latter node is assigned as the output.

To assess the error performance of a neuron, while at the same time determining the coefficients of the corresponding quadratic polynomial, one uses a

given data set of correct input-output pairs for $\tilde{f}$. Additionally, an error (or loss) function should be set up beforehand. Throughout this paper we use the Mean Squared Error (MSE) function

$$\mathrm{MSE}((y_1^{\mathrm{forecast}},\ldots,y_n^{\mathrm{forecast}}),(y_1^{\mathrm{actual}},\ldots,y_n^{\mathrm{actual}})) = \frac{1}{n}\sum_{i=1}^{n}(y_i^{\mathrm{forecast}} - y_i^{\mathrm{actual}})^2,$$

but there are a couple of other standard choices, such as the Mean Absolute Error (MAE) and the Mean Absolute Percentage Error (MAPE):

$$\frac{1}{n}\sum_{i=1}^{n}\left|y_i^{\mathrm{forecast}} - y_i^{\mathrm{actual}}\right| \qquad \text{resp.} \qquad \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i^{\mathrm{forecast}} - y_i^{\mathrm{actual}}}{y_i^{\mathrm{actual}}}\right|.$$

For each neuron ν_{ij} the data set is randomly split into a learning set and a test set. This is done to avoid overfitting, where the network learns too much about the inherent noise always being present in real-world data. The learning set is used to determine the coefficients b_{ijk}, by choosing them such that the error is as small as possible. In the case of MSE this can be achieved by linearization of the quadratic polynomial and applying the least squares method. The test set is then used to assess the performance of the neuron.

4 The Fan-Vercauteren SHE scheme

In this section we briefly describe a simplified version of the FV scheme [16], which we will present in its *somewhat* homomorphic encryption (SHE) form, meaning that it is suitable only for computations up to a given depth, thereby avoiding very expensive noise reduction operations (i.e. bootstrapping). It concerns a scale-invariant SHE scheme based on the hardness of the ring version of the learning with errors problem (RLWE) [27]. It works in the polynomial ring $R = \mathbf{Z}[X]/(f(X))$ with $f(X) = X^d + 1$ and $d = 2^n$. For an integer N we denote with R_N the reduction of R modulo N. Abusing notation, elements of R will often be identified with their unique representant in $\mathbf{Z}[X]$ of degree at most $d-1$, and similarly elements of R_N are identified with their unique representant inside

$$\left\{\, \alpha_{d-1}X^{d-1} + \alpha_{d-2}X^{d-2} + \ldots + \alpha_0 \;\middle|\; \alpha_i \in (-N/2, N/2] \text{ for all } i \right\},$$

but this should cause no confusion. For an element $a \in R$ or $a \in \mathbf{Z}[X]$ we write $[a]_N$ do denote its reduction inside the above set of representants.

The plaintext space in the FV scheme is given by the ring R_t for some small integer modulus $t > 1$, while a ciphertext is given by a pair of ring elements in R_q where $q > 1$ is a much larger modulus. The key generation and the encryption operations in the FV scheme require sampling from two probability distributions defined on R, denoted χ_{key} and χ_{err}. The security of the scheme is determined by the degree d of f, the size of q, and by the probability distributions. Typically χ_{key} and χ_{err} are coefficient-wise discrete Gaussian distributions centered around

0 and having a small standard deviation, but in practice one often samples the coefficients of the key from a uniform distribution on a narrow set like $\{-1, 0, 1\}$. We remark that the errors are sampled coefficient-wise because R is a ring of 2-power cyclotomic integers: for more general rings one should proceed with the more complicated joint distribution described in [28]. The RLWE distribution on $R_q \times R_q$ is then constructed as follows: first choose a fixed element $s \leftarrow \chi_{\text{key}}$, and then generate samples of the form (a, b) with $a \leftarrow R_q$ uniformly random and $b = [-(as + e)]_q$ with $e \leftarrow \chi_{\text{err}}$. (The minus sign is not standard but makes a better fit with the discussion below.) The decision RLWE problem is then to distinguish between the RLWE distribution and the uniform distribution on $R_q \times R_q$. The search RLWE problem is to retrieve s from polynomially many samples. Both problems are believed to be very hard for an appropriate choice of parameters.

By construction, for a RLWE sample (a, b) we have that $e = -[as + b]_q$ and therefore that the right-hand side has small coefficients, with overwhelming probability. Furthermore note that the sample can be easily re-randomized without knowledge of s as follows: choose $u \leftarrow \chi_{\text{key}}$ and $e_1, e_2 \leftarrow \chi_{\text{err}}$ and form the new sample as $(ua + e_1, ub + e_2)$. In the encryption scheme below, the public key will consist of a single RLWE sample, which will be re-randomized during encryption. The new RLWE sample will then be used as an additive mask to encrypt a message $m \in R_t$. Before we present the FV scheme, we first describe some subroutines that are required in the algorithm:

- $\texttt{WordDecomp}_{w,q}(a)$: This function is used to decompose a ring element $a \in R_q$ in base w by splicing each coefficient of a. For $u = \lfloor \log_w(q) \rfloor$, it returns $a_i \in R$ with coefficients in $(-w/2, w/2]$, such that $a = \sum_{i=0}^{u} a_i w^i$.
- $\texttt{PowersOf}_{w,q}(a)$: This function scales an element $a \in R_q$ by the different powers of w. It is defined as $\texttt{PowersOf}_{w,q}(a) = (aw^i)_{i=0}^{u}$.

These two functions can be used to perform a polynomial multiplication in R_q through an inner product: $\langle \texttt{WordDecomp}_{w,q}(a), \texttt{PowersOf}_{w,q}(b) \rangle = a \cdot b$. This expression has advantage in reducing the noise during homomorphic multiplications, as the first vector contains small elements only.

The FV scheme consists of an encryption scheme augmented with additional functions `Add`, `Mult`, and `ReLin` to compute homomorphically on encrypted data.

1. $\texttt{ParamsGen}(\lambda)$: For a given security parameter λ, choose a degree $d = 2^n$ and thus a polynomial $f(X) = X^d + 1$, moduli q and t and distributions χ_{err} and χ_{key}. Also choose the base w for $\texttt{WordDecomp}_{w,q}(\cdot)$. Return the system parameters $(d, q, t, \chi_{\text{err}}, \chi_{\text{key}}, w)$.
2. $\texttt{KeyGen}(d, q, t, \chi_{\text{err}}, \chi_{\text{key}}, w)$: Sample the secret key $s \leftarrow \chi_{\text{key}}$, sample $a \leftarrow R_q$ uniformly at random, and sample $e \leftarrow \chi_{\text{err}}$. Compute $b = [-(as + e)]_q$. The public key is the pair $\mathbf{pk} = (b, a)$ and the secret key is $\text{sk} = s$. The scheme uses another key $\mathbf{rlk}$ called *relinearization key* in the function `ReLin` below. Define $\ell = u + 1 = \lfloor \log_w(q) \rfloor + 1$, sample a vector $\mathbf{a} \leftarrow R_q^\ell$ uniformly at random, sample $\mathbf{e} \leftarrow \chi_{\text{err}}^\ell$, and let $\mathbf{rlk} = ([\texttt{PowersOf}_{w,q}(s^2) - (\mathbf{e} + \mathbf{a} \cdot s)]_q, \mathbf{a}) \in R_q^\ell \times R_q^\ell$.

3. $\texttt{Encrypt}(\mathbf{pk}, m)$: First encode the input message $m \in R_t$ into a polynomial $\Delta m \in R_q$ with $\Delta = \lfloor q/t \rfloor$. Next sample the error polynomials $e_1, e_2 \leftarrow \chi_{\text{err}}$, sample $u \leftarrow \chi_{\text{key}}$, and compute the two polynomials $c_0 = \Delta m + bu + e_1 \in R_q$ and $c_1 = au + e_2 \in R_q$. The ciphertext is the pair of polynomials $\mathbf{c} = (c_0, c_1)$.
4. $\texttt{Decrypt}(\text{sk}, \mathbf{c})$: First compute the polynomial $\tilde{m} = [c_0 + sc_1]_q$. Then recover the plaintext message m by a decoding the coefficients of $\tilde{m}$ by scaling down by Δ and rounding.
5. $\texttt{Add}(\mathbf{c_1}, \mathbf{c_2})$: For two ciphertexts $\mathbf{c_1} = (c_{1,0}, c_{1,1})$ and $\mathbf{c_2} = (c_{2,0}, c_{2,1})$, return $\mathbf{c} = (c_{1,0} + c_{2,0}, c_{1,1} + c_{2,1}) \in R_q \times R_q$.
6. $\texttt{Mult}(\mathbf{c}_1, \mathbf{c}_2, \mathbf{rlk})$: Compute $\tilde{\mathbf{c}}_{\text{mult}} = (c_0, c_1, c_2)$ where $c_0 = \lfloor \frac{t}{q} \cdot c_{1,0} \cdot c_{2,0} \rceil$, $c_1 = \lfloor \frac{t}{q} \cdot (c_{1,0} \cdot c_{2,1} + c_{1,1} \cdot c_{2,0}) \rceil$, and $c_2 = \lfloor \frac{t}{q} \cdot c_{1,1} \cdot c_{2,1} \rceil$ and apply relinearization.
7. $\texttt{ReLin}(\tilde{\mathbf{c}}_{\mathbf{mult}}, \mathbf{rlk})$: Write $\mathbf{rlk} = (\mathbf{b}, \mathbf{a})$ and $\tilde{\mathbf{c}}_{\text{mult}} = (c_0, c_1, c_2)$, then compute a relinearized ciphertext as $\mathbf{c}' = (c_0', c_1')$ as $([c_0 + \langle \texttt{WordDecomp}_{w,q}(c_2), \mathbf{b} \rangle]_q, [c_1 + \langle \texttt{WordDecomp}_{w,q}(c_2), \mathbf{a} \rangle]_q)$.

Given an FV ciphertext $\mathbf{c} = (c_0, c_1)$, we can write $[c_0 + c_1 s]_q = \Delta m + e$, where e is called the noise inside the ciphertext. Every operation on ciphertexts causes the noise to increase. It is clear that when the noise gets too large, in particular if $\|e\|_\infty > \Delta/2$, correct decryption will fail, where $\|\cdot\|_\infty$ denotes the maximal absolute value of the coefficients.

From now on we assume that χ_{err} is a coefficient-wise discrete Gaussian with standard deviation σ and that χ_{key} samples the coefficients uniformly from $\{-1, 0, 1\}$. With overwhelming probability $B_{\text{err}} = 6\sigma$ and $B_{\text{key}} = 1$ are upper bounds on the absolute values of the coefficients of their respective samples. Therefore we can use $V = B_{\text{err}}(1 + 2dB_{\text{key}}) = B_{\text{err}}(1 + 2d)$ as an upper bound on the noise of the input ciphertexts. When doing arithmetic the noise is affected in the following way. Firstly, adding ciphertexts $\mathbf{c}_1$ and $\mathbf{c}_2$ corresponds to adding the noises, potentially augmented by a carryover γ satisfying $\|\gamma\|_\infty < t$, as explained in [16]. Secondly, multiplying a ciphertext $\mathbf{c}$ by an unencrypted scalar $(\Delta\alpha, 0)$ for some $\alpha \in R_t$ corresponds to multiplying the noise by α, again with some carryover γ. For use below, fix an integer $\lambda \geq 1$ and assume that the coefficients of α are in $\{-1, 0, 1\}$ with at most λ of them being non-zero. Then in a similar way one sees that $\|\gamma\|_\infty < \lfloor \lambda/2 \rfloor \cdot t$. Thirdly, multiplying two ciphertexts $\mathbf{c}_1$ and $\mathbf{c}_2$ whose noise coefficients are bounded by E results in a ciphertext whose noise coefficients are at most

$$2 \cdot E \cdot t \cdot d \cdot (d + 1) + 8 \cdot t^2 \cdot d^2 + \ell \cdot B_{\text{err}} \cdot w \cdot d/2$$

in absolute value, by [16, Lem. 2 & Lem. 3].

Now assume that we wish to evaluate a GMDH network $f : R_t^{n_0} \to R_t$ having r hidden layers in a fresh component-wise encryption of an n_0-tuple $(x_1, x_2, \ldots, x_{n_0}) \in R_t^{n_0}$. For the moment just think of this as a Kolmogorov-Gabor polynomial that we evaluate in the encrypted domain along a diagram of the kind depicted in Figure 2; the purpose of this will become clear in the next section. The network parameters b_{ijk} are assumed to be small public scalars along the lines mentioned above: the coefficients are in $\{-1, 0, 1\}$ and at most λ

of them are non-zero. Define $A_1 = 6 \cdot \lambda \cdot t \cdot d \cdot (d+1) + 2 \cdot \lambda$ and

$$A_2 = 3/2 \cdot \lambda \cdot \ell \cdot B_{\mathrm{err}} \cdot w \cdot d + 24 \cdot t^2 \cdot d^2 + 5 \cdot (\lfloor \lambda/2 \rfloor + 1) \cdot t.$$

One verifies that homomorphically evaluating a node $\nu_{1j} : R_t^{n_0} \to R_t^{n_1}$ from the first layer causes the noise coefficients to grow to at most $A_1 \cdot V + A_2$. Recursively applying this formula yields the upper bound $A_1^{r+1} \cdot V + (A_1^{r+1} - 1) \cdot A_2/(A_1 - 1)$ on the absolute values of the noise coefficients that are present in the output of the entire network f.

The parameters of the FV scheme are not only determined by the noise growth, but also by the security requirements. It is easy to see that when d and σ/q grow, amounting to larger polynomials and more noise in the ciphertexts, then RLWE becomes harder. A precise security analysis is beyond the scope of this paper, but to derive our security estimates we closely follow the work by Albrecht, Player and Scott [3] and the open source LWE-estimator implemented by Albrecht [2]. In particular, the LWE-estimator allows one to estimate the concrete hardness of the LWE problem given the dimension d, the modulus q and the standard deviation σ. Note that the actual tool takes as input the parameter $\alpha = \sqrt{2\pi}\sigma/q$, instead of σ directly.

For the design reasons explained in Section 6 we will take $r = 3$, $\lambda = 9$, while for compatibility reasons with the software library `FV-NFLlib` [7] we wish to take $w = 2^{32}$ and $\log_2 q$ an integral multiple of 62. Targetting a security level of 80 bits, we can address the restrictions coming from both the noise growth and the security considerations by using the parameter set $d = 4096$, $q \simeq 2^{186}$ and $\sigma = 102$ (corresponding to $\alpha = 256/q$). These parameters will be used throughout the remainder of the paper and allow for usage of all plaintext moduli $t \leq 396$. Note that one ciphertext takes up 186kB space.

5 Representing fixed-point numbers in plaintext space

Our final goal is to evaluate a trained GMDH network in the encrypted domain using the FV scheme. As explained in the previous section, the plaintext space is of the form R_t, which is the reduction modulo a certain integer $t > 1$ of $R = \mathbf{Z}[X]/(X^d + 1)$, where $d = 2^n$ for some $n \in \mathbf{Z}_{>0}$. Therefore an important task is to encode the input values $x_1, x_2, \ldots, x_{n_0} \in \mathbf{R}$, as well as the coefficients $b_{ijk} \in \mathbf{R}$, as elements of R_t. This should be done in such a way that real additions and multiplications agree with the corresponding operations in the ring R_t, up to a certain depth of computation. Dowlin et al. [11] proposed two ways of addressing this issue, which were revisited in a recent paper by Costache et al. [6], who showed them to be essentially equivalent, and also provided lower bounds on t and d guaranteeing that the arithmetic in $\mathbf{R}$ is indeed compatible with that in R_t to the extent desired. We briefly recall their main conclusions, adapted to our setting.

On the real number side, we use fixed-point arithmetic. We assume that the x_i's and the b_{ijk}'s are given in balanced ternary expansion to some finite

precision, that is, they are of the form

$$b_{\ell_1-1}b_{\ell_1-2}\ldots b_0\,.\,b_{-1}b_{-2}\ldots b_{-\ell_2} \tag{1}$$

with $b_i \in \{-1,0,1\}$ for $i = -\ell_2,\ldots,\ell_1-1$. This should be read as

$$b_{\ell_1-1}3^{\ell_1-1} + b_{\ell_1-2}3^{\ell_1-2} + \ldots + b_0 3^0 + b_{-1}3^{-1} + b_{-2}3^{-2} + \ldots + b_{-\ell_2}3^{-\ell_2}.$$

As usual we say that (1) has ℓ_1 integral digits and ℓ_2 fractional digits; throughout we assume that $\ell_1 \geq 1$ and $\ell_2 \geq 0$. In order to encode (1) as an element of R_t one simply replaces the base 3 by X. This yields

$$b_{\ell_1-1}X^{\ell_1-1} + b_{\ell_1-2}X^{\ell_1-2} + \ldots + b_0X^0 + b_{-1}X^{-1} + b_{-2}X^{-2} + \ldots + b_{-\ell_2}X^{-\ell_2}, \tag{2}$$

which one can rewrite as

$$b_{\ell_1-1}X^{\ell_1-1} + b_{\ell_1-2}X^{\ell_1-2} + \ldots + b_0X^0 + b_{-1}X^{d-1} + b_{-2}X^{d-2} + \ldots + b_{-\ell_2}X^{d-\ell_2},$$

using the relation $X^d \equiv -1$.

To decode a given element of R_t one first considers its unique representant inside $\left\{ \alpha_{d-1}X^{d-1} + \alpha_{d-2}X^{d-2} + \ldots + \alpha_0 \;\middle|\; \alpha_i \in (-t/2, t/2] \text{ for all } i \right\}$, after which one replaces all suitably high powers X^i by $-X^{i-d}$, and one evaluates the resulting Laurent polynomial at 3. The outcome is a rational number whose denominator is a power of 3, so it can be easily rewritten in balanced ternary expansion. For simplicity we think of 'suitably high' as $i > d/2$, although to improve the bound on d in Lemma 1 below, a more careful (but easy) estimation should be made, that takes into account the lengths of the integral and fractional parts of the fixed-point numbers involved.

Clearly, the ring operations in R_t are compatible with fixed-point arithmetic on the real number side as long as they do not involve 'wrapping around' modulo t and/or modulo X^d+1. (In the latter case this means that neither the terms of high degree nor the terms of low degree are allowed to cross the separation point $X^{d/2}$.) Thus t and d should be taken large enough to ensure this, for which Costache et al. elaborated concrete lower bounds. We will not explicitly rely on these bounds, but rather apply the underlying ideas to obtain a more implicit statement. For all integers $\ell \geq 0, \lambda \geq 0, r \geq -1$ we define $d_{\ell,\lambda,r} := 2^{r+1}\ell + (2^{r+1}-1)\lambda$. Moreover for all $\ell_1 \geq 1, \lambda_1 \geq 1, \ell_2 \geq 0, \lambda_2 \geq 0, r \geq -1$ we introduce a polynomial $D_{\ell_1,\lambda_1,\ell_2,\lambda_2,r}(X) \in \mathbf{Z}[X]$, which is recursively defined by putting $D_{\ell_1,\lambda_1,\ell_2,\lambda_2,-1}(X) = 1 + X + X^2 + \ldots + X^{\ell_1+\ell_2-1}$ and for $r \geq 0$ letting $D_{\ell_1,\lambda_1,\ell_2,\lambda_2,r}(X)$ be $X^{2d_{\ell_2,\lambda_2,r-1}} + 2X^{d_{\ell_2,\lambda_2,r-1}}D_{\ell_1,\lambda_1,\ell_2,\lambda_2,r-1}(X) + 3D_{\ell_1,\lambda_1,\ell_2,\lambda_2,r-1}(X)^2$ multiplied with $1 + X + X^2 + \ldots + X^{\lambda_1+\lambda_2-1}$. We then define $c_{\ell_1,\lambda_1,\ell_2,\lambda_2,r} = \|D_{\ell,\lambda,r}(X)\|_\infty$ where as before $\|\cdot\|_\infty$ denotes the maximal absolute value of the coefficients. Note that $\deg D_{\ell,\lambda,r}(X) = d_{\ell_1+\ell_2-1,\lambda_1+\lambda_2-1,r}$. This all looks a bit cumbersome but the idea underlying these definitions should become apparent from the proof below.

Lemma 1. *Suppose that the input values $x_1, x_2, \ldots, x_{n_0}$ resp. the coefficients b_{ijk} are given by balanced ternary expansions of at most ℓ_1 resp. λ_1 integral*

digits and ℓ_2 resp. λ_2 fractional digits. Let x_{out} be the evaluation of our GMDH network at the x_i's, obtained by using fixed-point arithmetic. Let $\phi(X) \in R_t$ be the evaluation of our GMDH network at the encodings of the x_i's (using the encodings of the b_{ijk}'s as coefficients), obtained by using the respective ring operations in R_t. If

$$t \geq 2 \cdot c_{\ell_1,\lambda_1,\ell_2,\lambda_2,r} \quad \text{and} \quad d \geq 2 \cdot \max\{d_{\ell_1+\ell_2-1,\lambda_1+\lambda_2-1,r}, d_{\ell_2,\lambda_2,r} + 1\}$$

then $\phi(X)$ decodes to x_{out}.

Proof. Consider the evaluation of our GMDH network when carried out in $\mathbf{Z}[X, X^{-1}]$, using encodings of the form (2). We claim that the outcome is of the form $X^{-m}g(X)$ with $m \leq d_{\ell_2,\lambda_2,r}$ and $g(X) \in \mathbf{Z}[X]$ of degree at most $d_{\ell_1+\ell_2-1,\lambda_1+\lambda_2-1,r}$ and having coefficients bounded (in absolute value) by $c_{\ell_1,\lambda_1,\ell_2,\lambda_2,r}$. This claim clearly implies the lemma.

The key observation is that if one replaces all inputs by $X^{-\ell_2} + X^{-\ell_2+1} + X^{-\ell_2+2} + \ldots + X^{\ell_1-1}$ while replacing all encoded b_{ijk}'s by $X^{-\lambda_2} + X^{-\lambda_2+1} + X^{-\lambda_2+2} + \ldots + X^{\lambda_1-1}$ then these quantities can only increase, by the triangle inequality for the absolute value. By induction on r, it is easy to show that the corresponding evaluation is precisely $X^{-d_{\ell_2,\lambda_2,r}} \cdot D_{\ell_1,\lambda_1,\ell_2,\lambda_2,r}(X)$, from which the claim follows. ∎

These bounds are easy to compute in practice, using a computer algebra package. For example with $\ell_1 = 4$, $\ell_2 = 1$, $\lambda_1 = 1$, $\lambda_2 = 8$ and $r = 3$, we obtain the bounds

$$t \geq 93659577705415581454099599864654 \approx 2^{106.207} \qquad \text{and} \qquad d \geq 368. \quad (3)$$

This concrete choice of parameters will reoccur later in the paper.

One sees that the obtained bound on t is very large, which is problematic for a direct application of the FV scheme: remember from the previous section that we need $t \leq 396$. To address this issue we follow an idea mentioned in [4, §5.5], namely to decompose the plaintext space using the Chinese Remainder Theorem (CRT). That is, if one lets t be a large enough product of small mutually coprime numbers $t_1, t_2, \ldots, t_m$ then we have the well-known ring isomorphism

$$R_t \to R_{t_1} \times R_{t_2} \times \ldots \times R_{t_m} : g(X) \mapsto (g(X) \bmod t_1, \ldots, g(X) \bmod t_m).$$

Instead of evaluating our GMDH network directly in R_t, we can work in each of the R_{t_i}'s separately, simply by reducing things modulo t_i. The outcomes can then be combined very efficiently in order to end up in R_t again. As a consequence it suffices to carry out the FV scheme using the much smaller plaintext spaces R_{t_i}, although one needs to do it for each i separately. For the above example, the 13 mutually coprime numbers 269, 271, 277, 281, 283, 285, 286, 287, 289, 293, 307, 311 and 313 multiply together to $t = 950594835330878124611715152760 10 \approx 2^{106.229}$, which indeed satisfies the bound from (3). Thus it suffices to work with R_{269}, R_{271}, …, R_{313}.

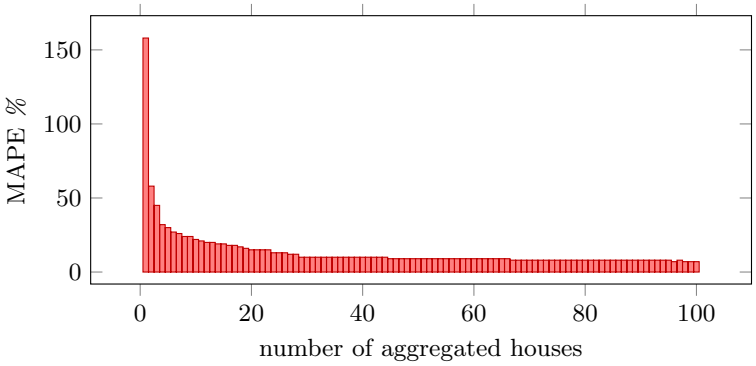


Fig. 3. The MAPE when forecasting the power consumption for the next half hour when using a varying number of aggregated households.

6 Prediction Approach for the Smart Grid

6.1 Prediction Model: Apartment Complexes

It is known that it is intrinsically difficult to make accurate short-term predictions based on data from one household when using an artificial neural network [38], and the same volatile behaviour is to be expected when following a GMDH approach. In order to confirm this, we designed and trained for each value of $n = 1, \ldots, 100$ a GMDH network that predicts the energy consumption during the next half hour for n aggregated households. This was done along the design criteria (and using the data set) described in Section 6.2 below. The observed prediction qualities, expressed in terms of the mean absolute percentage error (MAPE), are given in Figure 3. One sees that the results for one household are particularly bad, showing a MAPE of over 158 percent. However, the results start to improve significantly when using aggregated measurements of 10 households: here the MAPE is slightly above 20 percent, while it drops to 7 percent for $n = 100$. These observations are well in line with the ones for ANNs [38]. Due to this volatile nature we decided to aim for *aggregated* prediction, albeit for a *low* number of households. More precisely, we chose $n = 10$, which matches small apartment complexes in rural areas.

The cryptographic setting we have in mind is that the individual meter readings are homomorphically encrypted by the smart meter or gateway, and then sent to a third party who will perform homomorphic computations. Our security assumption is that the third party is honest but curious: it runs the protocol and computations as specified (i.e. it evaluates the GMDH network), but it will try to learn as much as possible about its inputs and outputs. The third party has received the concrete parameters (such as the coefficients b_{ijk}) of a trained GMDH network from the final party who wants to know the consumption prediction (e.g. the electricity supplier or the network operator). After homomorphically aggregating the data per 10 households, the third party obtains the encrypted inputs

$x_1, x_2, \ldots, x_{n_0}$ on which the GMDH network is evaluated homomorphically. The result is an encrypted forecast, which is then forwarded to the final party, who is able to decrypt using the cryptographic key corresponding to the one installed in the smart meters. The second security assumption we have to make is that the third party does not collude with the final party, since otherwise the third party could simply forward the encrypted individual meter readings.

6.2 Design of the Network

As explained in Section 3 the exact layout of our GMDH network is determined during a learning phase, for which we need access to some real smart meter data. We used the data that was collected through the Irish smart metering electricity customer behaviour trials [5] which ran in 2009 and 2010 with over 5,000 Irish homes and businesses participating. The data consists of electricity consumed during 30 minute intervals (in kW). Per household there are $25,728$ electricity measurements for a total of 536 days. We use the measurements of the first year as training data and the remaining half year to validate and measure how good the network is performing.

An important balancing act is to find a network layout that minimizes the number of layers (and therefore the multiplicative depth of the prediction algorithm) while at the same time preserving a reasonable prediction accuracy, preferably comparable to [38]. Through some trial and error the simplest GMDH network we found to meet these requirements consists of $r = 3$ hidden layers with $n_1 = 8$, $n_2 = 4$ and $n_3 = 2$ nodes, respectively. As input layer a set of $n_0 = 51$ nodes is used, where 48 nodes represent the half hour measurements that were made during the previous 24 hours. The remaining 3 inputs correspond to the temperature, the month, and the day of the week. The single output node $\nu_{4,1}$ then returns the predicted electricity consumption for the next half hour.

Let $\tilde{f} : \mathbf{R}^{51} \to \mathbf{R}$ denote the function that we want to approximate, for which a set of m input-output pairs $((x_{i1}, x_{i2}, \ldots, x_{in_0}), y_i^{\text{actual}})_{i=1,\ldots,m}$, with $y_i^{\text{actual}} = \tilde{f}(x_{i1}, x_{i2}, \ldots, x_{in_0})$, is given through the Irish data set. As explained in Section 3 these are used to inductively determine the coefficients b_{ijk}, while at the same time selecting the best performing nodes. Assuming that layer $i-1$ was dealt with, for node ν_{ij} this is done by minimizing the quantity

$$\text{MSE}\left(\left(f_{ij}(x_{11}, \ldots, x_{1n_0}), \ldots, f_{ij}(x_{m1}, \ldots, x_{mn_0}) \right), \left(y_1^{\text{actual}}, \ldots, y_m^{\text{actual}} \right) \right),$$

where $f_{ij} : \mathbf{R}^{51} \to \mathbf{R}$ denotes the function obtained from the network by temporarily considering ν_{ij} as an output node. The minimization can be done using standard linear regression. The useful feature of this approach is that one can apply L2-regularization and kill two birds with one stone. On the one hand regularization helps to avoid the *overfitting* problem, while on the other hand it allows to control the magnitude of the b_{ijk}'s. In this way one can achieve that ν_{ij} is a quadratic polynomial function with small coefficients and a reasonable MSE. We would like to point out that while we use MSE in the learning phase, the quality of the eventually resulting GMDH network is measured in terms

Table 1. The time (in ms) to compute the various basic (homomorphic) operations for our selected parameters.

op	enc	dec	key gen	add	mul	scalar mul
ms	2.1	5.8	77	0.1	33	29

of MAPE, in order to allow for a meaningful comparison with the forecasting results reported upon in the scientific literature.

As outlined in Section 5 we carry out fixed-point arithmetic using balanced ternary expansions, rather than binary expansions. To represent the input values $x_1, x_2, \ldots, x_{n_0}$ we use 1 fractional digit and, since the maximal data value is 27.265, at most 4 integral digits. The coefficients b_{ijk} are represented using 1 integral and 8 fractional digits. With these choices we attain basically the same average MAPE around 21 percent as in the floating point setting: a further increase of the precision does not give any significant improvement, although it gradually makes the fixed-point MSE converge to the floating-point one.

6.3 Benchmark Results

In order to assess the practical performance and verify the correctness of our selected parameters we implemented the privacy-preserving homomorphic forecasting approach as introduced in this paper. Our implementation (which will be made publicly avaiable soon) uses the FV-NFLlib software library [7] which implements the FV homomorphic encryption scheme which in turn uses the NFLlib software library (as described in [29] and released at [8]) for computing polynomial arithmetic. Our presented benchmark figures are obtained when running the implementation on an average laptop equipped with an Intel Core i5-3427U CPU (running at 1.80GHz).

Let us recall and summarize the exact forecasting setting and the parameters we selected for the implementation. It is our goal to predict the energy consumption for the next half hour of an apartment complex of 10 households while not revealing any energy consumption information to the party computing on this data using the GMDH approach as outlined in Section 3. Inherent to this approach we expect a MAPE which is slightly over 20 percent (see Section 6.1). In order to work efficiently with real numbers we use the fixed-point representation with the parameters as outlined in Section 5, using the CRT approach for decomposing plaintext space. We use the FV scheme for the homomorphic computation with the parameters as presented in Section 4. Hence, we target a security level of 80 bits and use the ring $R_{2^{186}} = \mathbf{Z}_{2^{186}}[X]/(X^{2^{12}} + 1)$ along with a standard deviation of 102. This means a ciphertext size of 186kB. Recall that the coefficients b_{ijk} are not being encrypted, which limits the noise growth when carrying out scalar multiplications.

As outlined in Section 6.2 the layout of our network consists of an input layer of 51 nodes, three hidden layers of 8, 4 and 2 nodes respectively and a single output node. Remember that when building a new layer the learning algorithm

excludes nodes corresponding to node pairs from the previous layer. So not all nodes of the resulting GMDH network affect on the final output and thus can be ignored during evaluation. Each node performs 8 multiplications out of which 5 are by polynomial coefficients and 5 additions. Since there are at most 15 nodes being evaluated this means computing 120 multiplications (out of which 75 by polynomial coefficients) and 75 additions. Table 1 summarizes the performance cost (expressed in milliseconds) for the various basic building blocks used in our homomorphic prediction algorithm. As can be seen from this table, and this is confirmed by running the entire forecasting algorithm in practice, the average computation of the prediction over 100 aggregated datasets is around 2.5 seconds depending on the node wiring. However, as explained in Section 5, this process has to be repeated 13 times for the CRT approach. In practice, the entire forecasting can be computed in half a minute. Due to the embarrassingly parallel nature of the CRT approach, a parallel implementation can compute this in less than 4 seconds or 2.5 seconds on average.

7 Conclusions and Future Work

We have shown that Ivakhnenko's group method of data handling from the 1970s is very suitable for homomorphic computation. This seems to be a better method with respect to the applicability to implement prediction homomorphically compared to the related artificial neural network based approaches in this cryptographic setting. We have studied this prediction approach in the setting of enhancing the privacy of the consumer for forecasting in the smart grid. Our privacy-preserving implementation of this approach to homomorphically forecast for 10 households shows is that this can be computed in less than four seconds for parallel and in half a minute for a sequential implementation.

We would like to point out that this approach has applications beyond the scope of just the smart grid. Other areas which need reliable prediction algorithms but work with privacy sensitive data can directly benefit as well. Examples include computing on financial data or biometric data.

References

1. A. Ahmad, M. Hassan, M. Abdullah, H. Rahman, F. Hussin, H. Abdullah, and R. Saidur. A review on applications of ANN and SVM for building electrical energy consumption forecasting. *Renewable and Sustainable Energy Reviews*, 33:102 – 109, 2014.
2. M. Albrecht. Complexity estimates for solving LWE. `https://bitbucket.org/malb/lwe-estimator/raw/HEAD/estimator.py`, 2000–2004.
3. M. R. Albrecht, R. Player, and S. Scott. On the concrete hardness of learning with errors. *J. Mathematical Cryptology*, 9(3):169–203, 2015.
4. J. W. Bos, K. Lauter, J. Loftus, and M. Naehrig. Improved security for a ring-based fully homomorphic encryption scheme. In M. Stam, editor, *Cryptography and Coding 2013*, volume 8308 of *LNCS*, pages 45–64. Springer, 2013.

5. Commission for Energy Regulation. Electricity smart metering customer behaviour trials (CBT) findings report. Technical Report CER11080a, 2011. http://www.cer.ie/docs/000340/cer11080(a)(i).pdf.
6. A. Costache, N. P. Smart, S. Vivek, and A. Waller. Fixed point arithmetic in SHE schemes. In *SAC 2016*, LNCS. Springer, 2016.
7. CryptoExperts. FV-NFLlib. https://github.com/CryptoExperts/FV-NFLlib, 2016.
8. CryptoExperts, INP ENSEEIHT, and Quarkslab. NFLlib. https://github.com/quarkslab/NFLlib, 2016.
9. Department of Energy & Climate Change. Smart metering implementation programme. Technical Report Third Annual Report on the Roll-out of Smart Meters, 2014. https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/384190/smip_smart_metering_annual_report_2014.pdf.
10. Department of Energy and Climate Change. Smart metering implementation programme – data access and privacy. https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/43043/4933-data-access-privacy-con-doc-smart-meter.pdf.
11. N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing. Manual for using homomorphic encryption for bioinformatics. Technical report, Technical report MSR-TR-2015-87, Microsoft Research, 2015.
12. N. Dowlin, R. Gilad-Bachrach, K. Laine, K. E. Lauter, M. Naehrig, and J. Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In M. Balcan and K. Q. Weinberger, editors, *International Conference on Machine Learning*, volume 48, pages 201–210. JMLR.org, 2016.
13. Z. Erkin and G. Tsudik. Private computation of spatial and temporal power consumption with smart meters. In F. Bao, P. Samarati, and J. Zhou, editors, *ACNS*, volume 7341 of *LNCS*, pages 561–577. Springer, 2012.
14. European Commission. Commission recommendation of 9 March 2012 on preparations for the roll-out of smart metering systems. Official Journal of the European Union http://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:32012H0148, March 2012.
15. European Commission. Benchmarking smart metering deployment in the EU-27 with a focus on electricity. Technical Report 365, June 2014. http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52014DC0356&from=EN.
16. J. Fan and F. Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012:144, 2012.
17. B. g. Koo, S. W. Lee, W. Kim, and J. H. Park. Comparative study of short-term electric load forecasting. In *Conference on Intelligent Systems, Modelling and Simulation*, pages 463–467, Jan 2014.
18. F. D. Garcia and B. Jacobs. Privacy-friendly energy-metering via homomorphic encryption. In J. Cuéllar, J. Lopez, G. Barthe, and A. Pretschner, editors, *STM*, volume 6710 of *LNCS*, pages 226–238. Springer, 2011.
19. C. Gentry. Fully homomorphic encryption using ideal lattices. In *ACM Symposium on Theory of Computing – STOC 2009*, STOC '09, pages 169–178. ACM, 2009.
20. G. W. Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, 1992.
21. L. Hernandez, C. Baladron, J. M. Aguiar, B. Carro, A. J. Sanchez-Esguevillas, J. Lloret, and J. Massana. A survey on electric power demand forecasting: Future trends in smart grids, microgrids and smart buildings. *IEEE Communications Surveys Tutorials*, 16(3):1460–1495, 2014.

22. A. Ivakhnenko. Heuristic self-organization in problems of engineering cybernetics. *Automatica*, 6(2):207 – 219, 1970.
23. M. Jawurek, F. Kerschbaum, and G. Danezis. Privacy technologies for smart grids - a survey of options. Technical Report MSR-TR-2012-119, November 2012. `http://research.microsoft.com/apps/pubs/default.aspx?id=178055`.
24. K. Kursawe, G. Danezis, and M. Kohlweiss. Privacy-friendly aggregation for the smart-grid. In S. Fischer-Hübner and N. Hopper, editors, *Privacy Enhancing Technologies – PETS*, volume 6794 of *LNCS*, pages 175–191. Springer, 2011.
25. F. Li, B. Luo, and P. Liu. Secure information aggregation for smart grids using homomorphic encryption. In *Smart Grid Comm.*, pages 327–332. IEEE, 2010.
26. R. Livni, S. Shalev-Shwartz, and O. Shamir. On the computational efficiency of training neural networks. In *Advances in Neural Information Processing Systems*, pages 855–863, 2014.
27. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, Heidelberg, May 2010.
28. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. *J. ACM*, 60(6):Art. 43, 35, 2013.
29. C. A. Melchor, J. Barrier, S. Guelton, A. Guinet, M. Killijian, and T. Lepoint. NFLlib: NTT-based fast lattice library. In K. Sako, editor, *CT-RSA 2016*, volume 9610 of *LNCS*, pages 341–356. Springer, 2016.
30. A. Molina-Markham, P. J. Shenoy, K. Fu, E. Cecchet, and D. E. Irwin. Private memoirs of a smart meter. In A. G. Ruzzelli, editor, *Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 61–66. ACM, 2010.
31. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In J. Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 223–238. Springer, Heidelberg, May 1999.
32. Recommendation to the European Commission. Essential regulatory requirements and recommendations for data handling, data safety, and consumer protection. Technical Report version 1.0, 2011. `https://ec.europa.eu/energy/sites/ener/files/documents/Recommendations%20regulatory%20requirements%20v1.pdf`.
33. A. Rial and G. Danezis. Privacy-preserving smart metering. In *Workshop on Privacy in the Electronic Society*, WPES '11, pages 49–60. ACM, 2011.
34. R. L. Rivest, L. Adleman, and M. L. Dertouzos. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.
35. Smart Grid Coordination Group. Smart grid information security. `http://ec.europa.eu/energy/sites/ener/files/documents/xpert_group1_security.pdf`, November 2012.
36. D. Srinivasan. Energy demand prediction using GMDH networks. *Neurocomputing*, 72(1):625–629, 2008.
37. The Smart Grid Interoperability Panel – Smart Grid Cybersecurity Committee. Guidelines for smart grid cybersecurity: Volume 1 - smart grid cybersecurity strategy, architecture, and high-level requirements. Technical Report NISTIR 7628 Rev 1, 2014. `http://nvlpubs.nist.gov/nistpubs/ir/2014/NIST.IR.7628r1.pdf`.
38. A. Veit, C. Goebel, R. Tidke, C. Doblander, and H.-A. Jacobsen. Household electricity demand forecasting: benchmarking state-of-the-art methods. In *Conference on future energy systems*, pages 233–234. ACM, 2014.
39. P. Xie, M. Bilenko, T. Finley, R. Gilad-Bachrach, K. E. Lauter, and M. Naehrig. Crypto-nets: Neural networks over encrypted data. *CoRR*, abs/1412.6181, 2014.

Chapter 9

Faster Homomorphic Function Evaluation Using Non-integral Base Encoding

Publication data

Bonte, C., Bootland, C., Bos, J. W., Castryck, W., Iliashenko, I., and Vercauteren, F. Faster homomorphic function evaluation using non-integral base encoding. In *Cryptographic Hardware and Embedded Systems - CHES 2017* (Sept. 2017), W. Fischer and N. Homma, Eds., vol. 10529 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 579–600.

Faster Homomorphic Function Evaluation using Non-Integral Base Encoding

Charlotte Bonte[1], Carl Bootland[1], Joppe W. Bos[2], Wouter Castryck[1,3], Ilia Iliashenko[1], and Frederik Vercauteren[1,4]

[1] imec-Cosic, Dept. Electrical Engineering, KU Leuven
[2] NXP Semiconductors
[3] Laboratoire Paul Painlevé, Université de Lille-1
[4] Open Security Research

Abstract. In this paper we present an encoding method for real numbers tailored for homomorphic function evaluation. The choice of the degree of the polynomial modulus used in all popular somewhat homomorphic encryption schemes is dominated by security considerations, while with the current encoding techniques the correctness requirement allows for much smaller values. We introduce a generic encoding method using expansions with respect to a non-integral base, which exploits this large degree at the benefit of reducing the growth of the coefficients when performing homomorphic operations. This allows one to choose a smaller plaintext coefficient modulus which results in a significant reduction of the running time. We illustrate our approach by applying this encoding in the setting of homomorphic electricity load forecasting for the smart grid which results in a speed-up by a factor 13 compared to previous work, where encoding was done using balanced ternary expansions.

1 Introduction

The cryptographic technique which allows an untrusted entity to perform arbitrary computation on encrypted data is known as fully homomorphic encryption. The first such construction was based on ideal lattices and was presented by Gentry in 2009 [24]. When the algorithm applied to the encrypted data is known in advance one can use a *somewhat homomorphic encryption* (SHE) scheme which only allows to perform a limited number of computational steps on the encrypted data. Such schemes are significantly more efficient in practice.

In all popular SHE schemes, the plaintext space is a ring of the form $R_t = \mathbb{Z}_t[X]/(f(X))$, where $t \geq 2$ is a small integer called the coefficient modulus, and $f(X) \in \mathbb{Z}[X]$ is a monic irreducible degree d polynomial called the polynomial modulus. Usually one lets $f(X)$ be a cyclotomic polynomial, where for reasons of

This work was supported by the European Commission under the ICT programme with contract H2020-ICT-2014-1 644209 HEAT, and through the European Research Council under the FP7/2007-2013 programme with ERC Grant Agreement 615722 MOTMELSUM. The second author is also supported by a PhD fellowship of the Research Foundation - Flanders (FWO).

performance the most popular choices are the power-of-two cyclotomics $X^d + 1$ where $d = 2^k$ for some positive integer k, which are maximally sparse. In this case arithmetic in R_t can be performed efficiently using the fast Fourier transform, which is used in many lattice-based constructions (e.g. [8,9,10,34]) and most implementations (e.g. [3,6,7,25,26,29,32]).

One interesting problem relates to the *encoding* of the input data of the algorithm such that it can be represented as elements of R_t and such that one obtains a meaningful outcome after the encrypted result is decrypted and decoded. This means that addition and multiplication of the input data must agree with the corresponding operations in R_t up to the depth of the envisaged SHE computation. An active research area investigates different such encoding techniques, which are often application-specific and dependent on the type of the input data. For the sake of exposition we will concentrate on the particularly interesting and popular setting where the input data consists of finite precision real numbers θ, even though our discussion below is fairly generic. The main idea, going back to Dowlin et al. [19] (see also [20,27,31]) and analyzed in more detail by Costache et al. [16], is to expand θ with respect to a base b

$$\theta = a_r b^r + a_{r-1} b^{r-1} + \cdots + a_1 b + a_0 + a_{-1} b^{-1} + a_{-2} b^{-2} + \cdots + a_{-s} b^{-s} \quad (1)$$

using integer digits a_i, after which one replaces b by X to end up inside the Laurent polynomial ring $\mathbb{Z}[X^{\pm 1}]$. One then reduces the digits a_i modulo t and applies the ring homomorphism to R_t defined by

$$\iota : \mathbb{Z}_t[X^{\pm 1}] \to R_t : \begin{cases} X & \mapsto X, \\ X^{-1} & \mapsto -g(X) \cdot f(0)^{-1}, \end{cases}$$

where we write $f(X) = Xg(X) + f(0)$ and it is assumed that $f(0)$ is invertible modulo t; this is always true for cyclotomic polynomials, or for factors of them. The quantity $r + s$ will sometimes be referred to as the *degree* of the encoding (where we assume that $a_r a_{-s} \neq 0$). For power-of-two-cyclotomics the homomorphism ι amounts to letting $X^{-1} \mapsto -X^{d-1}$, so that the encoding of (1) is given by[5] $a_r X^r + a_{r-1} X^{r-1} + \cdots + a_1 X + a_0 - a_{-1} X^{d-1} - a_{-2} X^{d-2} - \cdots - a_{-s} X^{d-s}$.

Decoding is done through the inverse of the restriction $\iota|_{\mathbb{Z}_t[X^{\pm 1}]_{[-\ell,m]}}$ where

$$\mathbb{Z}_t[X^{\pm 1}]_{[-\ell,m]} = \{\, a_m X^m + a_{m-1} X^{m-1} + \ldots + a_{-\ell} X^{-\ell} \mid a_i \in \mathbb{Z}_t \text{ for all } i \,\}$$

is a subset of Laurent polynomials whose monomials have bounded exponents. If $\ell + m + 1 = d$ then this restriction of ι is indeed invertible as a $\mathbb{Z}_t$-linear map. The precise choice of ℓ, m depends on the data encoded. After applying this inverse, one replaces the coefficients by their representants in $\{-\lfloor (t-1)/2 \rfloor, \ldots, \lceil (t-1)/2 \rceil\}$ to end up with an expression in $\mathbb{Z}[X^{\pm 1}]$, and evaluates the result at $X = b$. Ensuring that decoding is correct to a given computational depth places constraints on the parameters t and d, in order to avoid ending up outside the box depicted in Figure 1 if the computation were to be carried out directly in

[5] In fact in [16] it is mentioned that inverting X is only possible in the power-of-two cyclotomic case, but this seems to be overcareful.

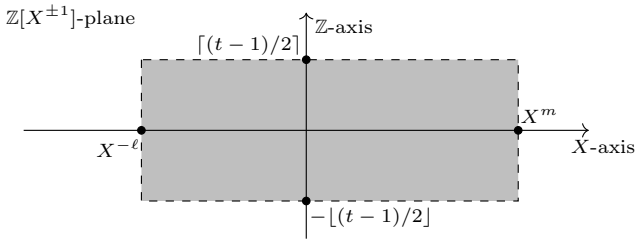


Fig. 1. Box in which to stay during computation, where $\ell + m + 1 = d$.

$\mathbb{Z}[X^{\pm 1}]$. In terms of R_t we will often refer to this event as the 'wrapping around' of the encoded data modulo t or $f(X)$, although we note that this is an abuse of language. In the case of power-of-two cyclotomics, ending up above or below the box does indeed correspond to wrapping around modulo t, but ending up at the left or the right of the box corresponds to a mix-up of the high degree terms and the low degree terms.

The precise constraints on t and d not only depend on the complexity of the computation, but also on the type of expansion (1) used in the encoding. Dowlin et al. suggest to use balanced b-ary expansions with respect to an odd base $b \in \mathbb{Z}_{\geq 3}$, which means that the digits are taken from $\{-(b-1)/2, \ldots, (b-1)/2\}$. Such expansions have been used for centuries going back at least to Colson (1726) and Cauchy (1840) in the quest for more efficient arithmetic.

If we fix a precision, then for smaller b the balanced b-ary expansions are longer but the coefficients are smaller, this implies the need for a larger d but smaller t. Similarly for larger bases the expansions become shorter but have larger coefficients leading to smaller d but larger t. For the application to somewhat homomorphic encryption considered in [6,16] the security requirements ask for a very large d, so that the best choice is to use as small a base as possible, namely $b = 3$, with digits in $\{\pm 1, 0\}$. Even for this smallest choice the resulting lower bound on t is very large and the bound on d is much smaller than that coming from the cryptographic requirements. To illustrate this, we recall the concrete figures from the paper [6], which uses the Fan-Vercauteren (FV) somewhat homomorphic encryption scheme [23] for privacy-friendly prediction of electricity consumption in the setting of the smart grid. Here the authors use $d = 4096$ for cryptographic reasons, which is an optimistic choice that leads to 80-bit security only (and maybe even a few bits less than that [1]). On the other hand using balanced ternary expansions, correct decoding is guaranteed as soon as $d \geq 368$, which is even a conservative estimate. This eventually leads to the huge bound $t \gtrapprox 2^{107}$, which is overcome by decomposing R_t into 13 factors using the Chinese Remainder Theorem (CRT). This is then used to homomorphically forecast the electricity usage for the next half hour for a small apartment complex of 10 households in about half a minute, using a sequential implementation.

The discrepancy between the requirements coming from correct decoding and those coming from security considerations suggests that other possible expan-

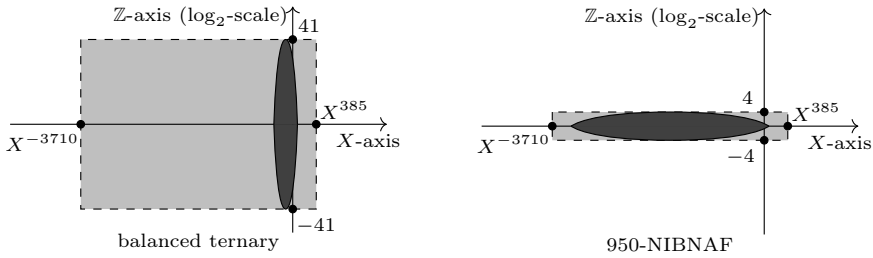


Fig. 2. Comparison of the amount of plaintext space which is actually used in the setting of [6], where $d = 4096$. More precise figures to be found in Section 4.

sions may be better suited for use with SHE. In this paper we introduce a generic encoding technique, using very sparse expansions having digits in $\{\pm 1, 0\}$ with respect to a *non-integral* base $b_w > 1$, where w is a sparseness measure. These expansions will be said to be of 'non-integral base non-adjacent form' with window size w, abbreviated to w-NIBNAF. Increasing w makes the degrees of the resulting Laurent polynomial encodings grow and decreases the growth of the coefficients when performing operations; hence lowering the bound on t. Our encoding technique is especially useful when using finite precision real numbers, but could also serve in dealing with finite precision complex numbers or even with integers, despite the fact that b_w is non-integral (this would require a careful precision analysis which is avoided here).

We demonstrate that this technique results in significant performance increases by re-doing the experiments from [6]. Along with a more careful precision analysis which is tailored for this specific use case, using 950-NIBNAF expansions we end up with the dramatically reduced bound $t \geq 33$. It is not entirely honest to compare this to $t \gtrapprox 2^{107}$ because of our better precision analysis; as explained in Section 4 it makes more sense to compare the new bound to $t \gtrapprox 2^{42}$, but the reduction remains huge. As the reader can see in Figure 2 this is explained by the fact that the data is spread more evenly across the plaintext space during computation. As a consequence we avoid the need for CRT decomposition and thus reduce the running time by a factor 13, showing that the same homomorphic forecasting can be done in only 2.5 seconds.

Remark. An alternative recent proposal for encoding using a non-integral base can be found in [15], which targets efficient evaluation of the discrete Fourier transform on encrypted data. Here the authors work exclusively in the power-of-two cyclotomic setting $f(X) = X^d + 1$, and the input data consists of complex numbers θ which are expanded with respect to the base $b = \zeta$, where ζ is a primitive $2d$-th root of unity, i.e. a root of $f(X)$; a similar idea was used in [12]. One nice feature of this approach is that the correctness of decoding is not affected by wrapping around modulo $f(X)$. To find a sparse expansion they use the LLL algorithm [28], but for arbitrary complex inputs the digits become rather large when compared to w-NIBNAF.

2 Encoding data using w-NIBNAF

Our approach in reducing the lower bound on the plaintext modulus t is to use encodings for which many of the coefficients are zero. In this respect, a first improvement over balanced ternary expansions is obtained by using the non-adjacent form (NAF) representations which were introduced by Reitweisner in 1960 for speeding up early multiplication algorithms [33]. We note that independent work by Cheon et al. [11] also mentions the advantages of using NAF encodings.

Definition 1. *The* non-adjacent form (NAF) representation *of a real number θ is an expansion of θ to the base $b = 2$ with coefficients in $\{-1, 0, 1\}$ such that any two adjacent coefficients are not both non-zero.*

The NAF representation has been generalized [13]: for an integer $w \geq 1$ (called the 'window size') one can ensure that in any window of w consecutive coefficients at most one of them is non-zero. This is possible to base $b = 2$ but for $w > 2$ one requires larger coefficients.

Definition 2. *Let $w \geq 1$ be an integer. A* w-NAF representation *of a real number θ is an expansion of θ with base 2 and whose non-zero coefficients are odd and less than 2^{w-1} in absolute value such that for every set of w consecutive coefficients at most one of them is non-zero.*

We see that NAF is just the special case of w-NAF for $w = 2$. Unfortunately, due to the fact that the coefficients are taken from a much larger set, using w-NAF encodings in the SHE setting actually gives larger bounds on both t and d for increasing w. Therefore this is not useful for our purposes.

Ideally, we want the coefficients in our expansions to be members of $\{\pm 1, 0\}$ with many equal to 0, as this leads to the slowest growth in coefficient sizes, allowing us to use smaller values for t. This would come at the expense of using longer encodings, but remember that we have a lot of manoeuvring space on the d side. One way to achieve this goal is to use a *non-integral* base $b > 1$ when computing a non-adjacent form. We first give the definition of a non-integral base non-adjacent form with window size w (w-NIBNAF) representation and then explain where this precise formulation comes from.

Definition 3. *A sequence $a_0, a_1, \ldots, a_n, \ldots$ is a* w-balanced ternary sequence *if it has $a_i \in \{-1, 0, 1\}$ for $i \in \mathbb{Z}_{\geq 0}$ and satisfies the property that each set of w consecutive terms has no more than one non-zero term.*

Definition 4. *Let $\theta \in \mathbb{R}$ and $w \in \mathbb{Z}_{>0}$. Define b_w to be the unique positive real root of the polynomial $F_w(x) = x^{w+1} - x^w - x - 1$. A w-balanced ternary sequence $a_r, a_{r-1}, \ldots, a_1, a_0, a_{-1}, \ldots$ is a w-NIBNAF representation of θ if*

$$\theta = a_r b_w^r + a_{r-1} b_w^{r-1} + \cdots + a_1 b_w + a_0 + a_{-1} b_w^{-1} + \cdots \ .$$

Below we will show that every $\theta \in \mathbb{R}$ has at least one such w-NIBNAF representation and provide an algorithm to find such a representation. But let us first state a lemma which shows that b_w is well-defined for $w \geq 1$.

Lemma 1. *For an integer $w \geq 1$ the polynomial $F_w(x) = x^{w+1} - x^w - x - 1$ has a unique positive real root $b_w > 1$. The sequence $b_1, b_2, \ldots$ is strictly decreasing and $\lim_{w\to\infty} b_w = 1$. Further, $(x^2+1) \mid F_w(x)$ for $w \equiv 3 \bmod 4$.*

The proof is straightforward and given in Appendix A. The first few values of b_w are as follows

$$\begin{aligned} b_1 &= 1 + \sqrt{2} \approx 2.414214, & b_2 &\approx 1.839287, \\ b_3 &= \tfrac{1}{2}(1+\sqrt{5}) \approx 1.618034, & b_4 &\approx 1.497094, \end{aligned}$$

where we note that b_3 is the golden ratio ϕ.

Since we are using a non-integral base, a w-NIBNAF representation of a fixed-point number has infinitely many non-zero terms in general. To overcome this one approximates the number by terminating the w-NIBNAF representation after some power of the base. We call such a terminated sequence an *approximate w-NIBNAF representation*. There are two straightforward ways of deciding where to terminate: either a fixed power of the base is chosen so that any terms after this are discarded giving an easy bound on the maximal possible error created, or we choose a maximal allowed error in advance and terminate after the first power which gives error less than or equal to this value.

Algorithm 1 below produces for every $\theta \in \mathbb{R}$ a w-NIBNAF representation in the limit as ϵ tends to 0, thereby demonstrating its existence. It takes the form of a greedy algorithm which chooses the closest signed power of the base to θ and then iteratively finds a representation of the difference. Except when θ can be written as $\theta = h(b_w)/b_w^q$, for some polynomial h with coefficients in $\{\pm 1, 0\}$ and $q \in \mathbb{Z}_{\geq 0}$, any w-NIBNAF representation is infinitely long. Hence, we must terminate Algorithm 1 once the iterative input is smaller than some pre-determined precision $\epsilon > 0$.

We now prove that the algorithm works as required.

Lemma 2. *Algorithm 1 produces an approximate w-NIBNAF representation of θ with an error of at most ϵ.*

Proof. Assuming that the algorithm terminates, the output clearly represents θ to within an error of at most size ϵ. First we show that the output is w-NIBNAF. Suppose that the output, on input θ, b_w, ϵ, has at least two non-zero terms, the first being a_d. This implies either that $b_w^d \leq |\theta| < b_w^{d+1}$ and $b_w^{d+1} - |\theta| > |\theta| - b_w^d$ or $b_w^{d-1} < |\theta| \leq b_w^d$ and $b_w^d - |\theta| \leq |\theta| - b_w^{d-1}$. These conditions can be written as $b_w^d \leq |\theta| < \frac{1}{2} b_w^d (1 + b_w)$ and $\frac{1}{2} b_w^{d-1}(1+b_w) \leq |\theta| \leq b_w^d$ respectively, showing that

$$||\theta| - b_w^d| < \max\left\{ b_w^d - \tfrac{1}{2} b_w^{d-1}(1+b_w),\ \tfrac{1}{2} b_w^d (1+b_w) - b_w^d \right\} = \tfrac{1}{2} b_w^d (b_w - 1) \ .$$

The algorithm subsequently chooses the closest power of b_w to this smaller value, suppose it is b_w^ℓ. By the same argument with θ replaced by $|\theta| - b_w^d$ we have that

Algorithm 1: `GreedyRepresentation`

Input: θ – the real number to be represented,
b_w – the w-NIBNAF base to be used in the representation,
ϵ – the precision to which the representation is determined.
Output: An approximate w-NIBNAF representation $a_r, a_{r-1}, \ldots$ of θ with error less than ϵ, where $a_i = 0$ if not otherwise specified.

$\sigma \leftarrow \operatorname{sgn}(\theta)$
$t \leftarrow |\theta|$
while $t > \epsilon$ **do**
  $r \leftarrow \lceil \log_{b_w}(t) \rceil$
  if $b_w^r - t > t - b_w^{r-1}$ **then**
    $r \leftarrow r - 1$
  $a_r \leftarrow \sigma$
  $\sigma \leftarrow \sigma \cdot \operatorname{sgn}(t - b_w^r)$
  $t \leftarrow |t - b_w^r|$
Return $(a_i)_i$.

either $b_w^\ell \leq \left||\theta| - b_w^d\right|$ or $\frac{1}{2}b_w^{\ell-1}(1+b_w) \leq \left||\theta| - b_w^d\right|$ and since b_w^ℓ is larger than $\frac{1}{2}b_w^{\ell-1}(1+b_w)$ the maximal possible value of ℓ, which we denote by $\ell_w(d)$, satisfies

$$\ell_w(d) = \max\left\{\ell \in \mathbb{Z} \;\middle|\; \tfrac{1}{2}b_w^{\ell-1}(1+b_w) < \tfrac{1}{2}b_w^d(b_w-1)\right\} \quad .$$

The condition on ℓ can be rewritten as $b_w^\ell < b_w^{d+1}(b_w-1)/(b_w+1)$ which implies that $\ell < d + 1 + \log_{b_w}((b_w-1)/(b_w+1))$ and thus

$$\ell_w(d) = d + \left\lceil \log_{b_w}\left(\frac{b_w-1}{b_w+1}\right)\right\rceil \quad ,$$

so that the smallest possible difference is independent of d and equal to

$$s(w) := d - \ell_w(d) = -\left\lceil \log_{b_w}\left(\frac{b_w-1}{b_w+1}\right)\right\rceil = \left\lfloor \log_{b_w}\left(\frac{b_w+1}{b_w-1}\right)\right\rfloor \quad .$$

We thus need to show that $s(w) \geq w$. As w is an integer this is equivalent to

$$\log_{b_w}\left(\frac{b_w+1}{b_w-1}\right) \geq w \iff b_w^w \leq \frac{b_w+1}{b_w-1} \iff b_w^{w+1} - b_w^w - b_w - 1 \leq 0$$

which holds for all w since $F_w(b_w) = 0$. Note that our algorithm works correctly and deterministically because when $|\theta|$ is exactly half-way between two powers of b_w we pick the larger power. This shows that the output is of the desired form.

Finally, to show that the algorithm terminates we note that the k'th successive difference is bounded above by $\frac{1}{2}b_w^{d-(k-1)s(w)}(b_w-1)$ and this tends to 0 as k tends to infinity. Therefore after a finite number of steps (at most $\left\lceil \left(d - \log_{b_w}\left(2\epsilon/(b_w-1)\right)\right)/s(w)\right\rceil + 1$) the difference is smaller than or equal to ϵ and the algorithm terminates. □

The process of encoding works as described in the introduction, i.e. we follow the approach from [16,19] except we use an approximate w-NIBNAF representation instead of the balanced ternary representation. Thus to encode a real number θ we find an approximate w-NIBNAF representation of θ with small enough error and replace each occurrence of b_w by X, after which we apply the map ι to end up in plaintext space R_t. Decoding is almost the same as well, only that after inverting ι and lifting the coefficients to $\mathbb{Z}$ we evaluate the resulting Laurent polynomial at $X = b_w$ rather than $X = 3$, computing the value only to the required precision. Rather than evaluating directly it is best to reduce the Laurent polynomial modulo $F_w(X)$ (or modulo $F_w(X)/(X^2+1)$ if $w \equiv 3 \bmod 4$) so that we only have to compute powers of b_w up to w (respectively $w-2$).

Clearly we can also ask Algorithm 1 to return $\sum_i a_i X^i \in \mathbb{Z}_t[X^{\pm 1}]$, this gives an encoding of θ with maximal error ϵ. Since the input θ of the algorithm can get arbitrarily close to but larger than ϵ, the final term can be $\pm X^h$ where $h = \lfloor \log_{b_w}(2\epsilon/(1+b_w)) \rfloor + 1$. If we are to ensure that the smallest power of the base to appear in any approximate w-NIBNAF representation is b_w^s then we require that if b_w^{s-1} is the nearest power of b_w to the input θ then $|\theta| \leq \epsilon$ so that we must have $\frac{1}{2} b_w^{s-1}(1+b_w) \leq \epsilon$ which implies the smallest precision we can achieve is $\epsilon = b_w^{s-1}(1+b_w)/2$. In particular if we want no negative powers of b_w then the best precision possible using the greedy algorithm is $(1+b_w^{-1})/2 < 1$.

Remark. If one replaces b_w by a smaller base $b > 1$ then Algorithm 1 still produces a w-NIBNAF expansion to precision ϵ: this follows from the proof of Lemma 2. The distinguishing feature of b_w is that it is maximal with respect to this property, so that the resulting expansions become as short as possible.

3 Analysis of coefficient growth during computation

After encoding the input data it is ready for homomorphic computations. This increases both the number of non-zero coefficients as well as the size of these coefficients. Since we are working in the ring R_t there is a risk that our data wraps around modulo t as well as modulo $f(X)$, in the sense explained in the introduction, which we should avoid since this leads to erroneous decoding. Therefore we need to understand the coefficient growth more thoroughly. We simplify the analysis in this section by only considering multiplications and what constraint this puts on t, it is then not hard to generalize this to include additions.

Worst case coefficient growth for w-NIBNAF encodings. Here we analyze the maximal possible size of a coefficient which could occur from computing with w-NIBNAF encodings. Because fresh w-NIBNAF encodings are just approximate w-NIBNAF representations written as elements of R_t we consider finite w-balanced ternary sequences and the multiplication endowed on them from R_t. Equivalently, we consider multiplication in the $\mathbb{Z}[X^{\pm 1}]$-plane depicted in Figure 1. As we ensure in practice that there is no wrap around modulo $f(X)$ this can be ignored in our analysis.

To start the worst case analysis we have the following lower bound; note that the d we use here is *not* that of the degree of $f(X)$.

Lemma 3. *The maximal absolute size of a term that can appear in the product of p arbitrary w-balanced ternary sequences of length $d+1$ is at least*

$$B_w(d,p) := \sum_{k=0}^{\lfloor\lfloor p\lfloor d/w\rfloor/2\rfloor/(\lfloor d/w\rfloor+1)\rfloor} (-1)^k \binom{p}{k}\binom{p-1+\lfloor p\lfloor d/w\rfloor/2\rfloor - k\lfloor d/w\rfloor - k}{p-1} \ .$$

A full proof of this lemma is given in Appendix A but the main idea is to look at the largest coefficient of m^p where m has the maximal number of non-zero coefficients, $\lfloor d/w\rfloor+1$, all being equal to 1 and with exactly $w-1$ zero coefficients between each pair of adjacent non-zero coefficients. The (non-zero) coefficients of m^p are variously known in the literature as extended (or generalized) binomial coefficients or ordinary multinomials; we denote them here by $\binom{p}{k}_n$ defined via

$$\left(1+X+X^2+\ldots+X^{n-1}\right)^p = \sum_{k=0}^{\infty} \binom{p}{k}_n X^k \ ,$$

[22,18,35,21]. In particular the maximal coefficient is the (or a) central one and we can write $B_w(d,p) = \binom{p}{k}_n$ where $k = \lfloor p\lfloor d/w\rfloor/2\rfloor$ and $n = \lfloor d/w\rfloor + 1$.

We note that the w-NIBNAF encoding, using the greedy algorithm with precision $\frac{1}{2}$, of $b_w^{d+w-(d \bmod w)}(b_w-1)/2$ is m so in practice this lower bound is achievable although highly unlikely to occur.

We expect that this lower bound is tight, indeed we were able to prove the following lemma, the proof is also given in Appendix A.

Lemma 4. *Suppose w divides d, then $B_w(d,p)$ equals the maximal absolute size of a term that can be produced by taking the product of p arbitrary w-balanced ternary sequences of length $d+1$.*

We thus make the following conjecture which holds for all small values of p and d we tested and which we assume to be true in general.

Conjecture 1 *The lower bound $B_w(d,p)$ given in Lemma 3 is exact for all d, that is the maximal absolute term size which can occur after multiplying p arbitrary w-balanced ternary sequences of length $d+1$ is $B_w(d,p)$.*

This conjecture seems very plausible since as soon as one multiplicand does not have non-zero coefficients exactly w places apart the non-zero coefficients start to spread out and decrease in value.

To determine $B_w(d,p)$ for fixed p define $n := \lfloor d/w\rfloor + 1$, then we can expand the expression for $B_w(d,p)$ as a 'polynomial' in n of degree $p-1$ where the coefficients depend on the parity of n, see [5] for more details. The first few are:

$$B_w(d,1) = 1, \qquad B_w(d,2) = n,$$
$$B_w(d,3) = \tfrac{1}{8}(6n^2+1) - \tfrac{(-1)^n}{8}, \qquad B_w(d,4) = \tfrac{1}{3}(2n^3+n),$$
$$B_w(d,5) = \tfrac{1}{384}(230n^4+70n^2+27) - \tfrac{(-1)^n}{384}(30n^2+27),$$
$$B_w(d,6) = \tfrac{1}{20}(11n^5+5n^3+4n).$$

Denoting the coefficient of n^{p-1} in these expressions by ℓ_p, it can be shown (see [2] or [5]) that $\lim_{p\to\infty} \sqrt{p}\ell_p = \sqrt{6/\pi}$ and hence we have

$$\lim_{p\to\infty} \log_2(B_w(d,p)) - (p-1)\log_2(n) + \tfrac{1}{2}\log_2\left(\tfrac{\pi p}{6}\right) = 0$$

or equivalently $B_w(d,p) \sim_p \sqrt{6/\pi p}n^{p-1}$. Thus we have the approximation

$$\log_2(B_w(d,p)) \approx (p-1)\log_2(n) - \tfrac{1}{2}\log_2\left(\tfrac{\pi p}{6}\right)$$

which for large enough n (experimentally we found for $n > 1.825\sqrt{p-1/2}$) is an upper bound for $p > 2$. For a guaranteed upper bound we refer to Mattner and Roos [30] where they state, for $n, p \in \mathbb{Z}_{>0}$ with $n \geq 2$, if $p \neq 2$ or $n \in \{2,3,4\}$ then $B_w(d,p) \leq \sqrt{6/(\pi p(n^2-1))}n^p$. This upper bound is in fact a more precise asymptotic limit than that above which only considers the leading coefficient.

Statistical analysis of the coefficient growth. Based on the w-NIBNAF encodings of random numbers in $N \in \left[-2^{40}, 2^{40}\right]$, we try to get an idea of the amount of zero and non-zero coefficients in a fresh encoding without fractional part, obtained by running Algorithm 1 to precision $(1+b_w^{-1})/2$. We also analyze how these proportions change when we perform multiplications. We plot this for different values of w to illustrate the positive effects of using sparser encodings. As a preliminary remark note that the w-NIBNAF encodings produced by Algorithm 1 applied to $-N$ and N are obtained from one another by changing all the signs, so the coefficients -1 and 1 are necessarily distributed evenly.[6]

We know from the definition of a w-NIBNAF expansion that at least $w-1$ among each block of w consecutive coefficients of the expansion will be 0, so we expect for big w that the 0 coefficient occurs a lot more often than ± 1. This is clearly visible in Figure 3. In addition we see an increasing number of 0 coefficients and decreasing number of ± 1 coefficients for increasing w. Thus both the absolute and the relative sparseness of our encodings increase as w increases.

Since the balanced ternary encoding of [16,19] and the 2-NAF encoding [33], only have coefficients in $\{0, \pm 1\}$ it is interesting to compare them to 1-NIBNAF and 2-NIBNAF respectively. We compare them by computing the percentage of zero and non-zero coefficients, in 10 000 encodings of random integers N in $\left[-2^{40}, 2^{40}\right]$. We compute this percentage up to an accuracy of 10^{-2} and consider for our counts all coefficients up to and including the leading coefficient, further zero coefficients are not counted. When we compare the percentages of zero and non-zero coefficients occurring in 1-NIBNAF and balanced ternary in Table 1 we see that for the balanced ternary representation, the occurrences of 0, 1 and -1 coefficients are approximately the same, while for 1-NIBNAF the proportion of 0

[6] This is a desirable property leading to the maximal amount of cancellation during computation. While this does not affect our worst case analysis, in practice where the worst case is extremely unlikely this accounts for a considerable reduction of the size of the coefficient modulus t. If in some application the input encodings happen to be biased towards 1 or -1 then one can work with respect to the *negative* base $-b_w < -1$, by switching the signs of all the digits appearing at an odd index.

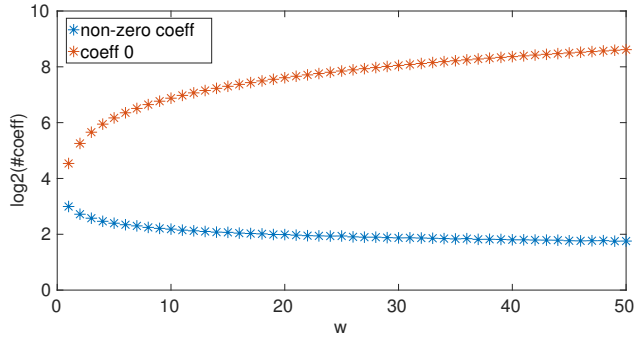


Fig. 3. Plot of $\log_2(\#\text{coeff})$ on the vertical axis against w on the horizontal axis averaged over 10 000 w-NIBNAF encodings of random integers in $\left[-2^{40}, 2^{40}\right]$.

	balanced ternary	1-NIBNAF	2-NAF	2-NIBNAF
zero coefficients	32.25%	48.69%	65.23%	70.46%
non-zero coefficients	67.76%	51.31%	34.77%	29.54%

Table 1. Comparison between the previous encoding techniques and w-NIBNAF

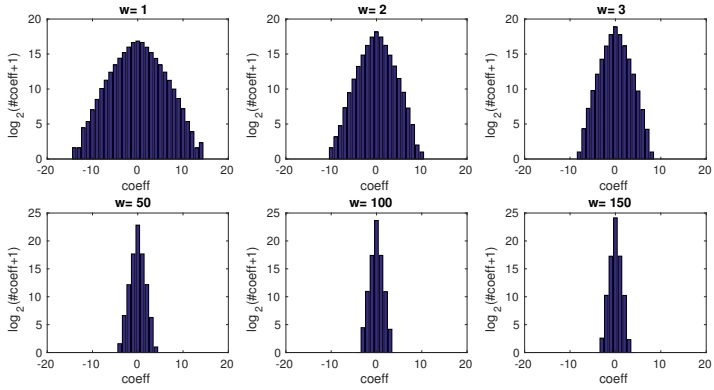


Fig. 4. Plot of $\log_2(\#\text{coeff}+1)$ on the vertical axis against the respective value of the coefficient on the horizontal axis for the result of 10 000 multiplications of two w-NIBNAF encodings of random numbers between $\left[-2^{40}, 2^{40}\right]$.

coefficients is larger than that of 1 or -1. Hence we can conclude that 1-NIBNAF encodings will be sparser than the balanced ternary encodings even though the window size is the same. For 2-NIBNAF we also see an improvement in terms of sparseness of the encoding compared to 2-NAF.

The next step is to investigate what happens to the coefficients when we multiply two encodings. From Figure 4 we see that when w increases the max-

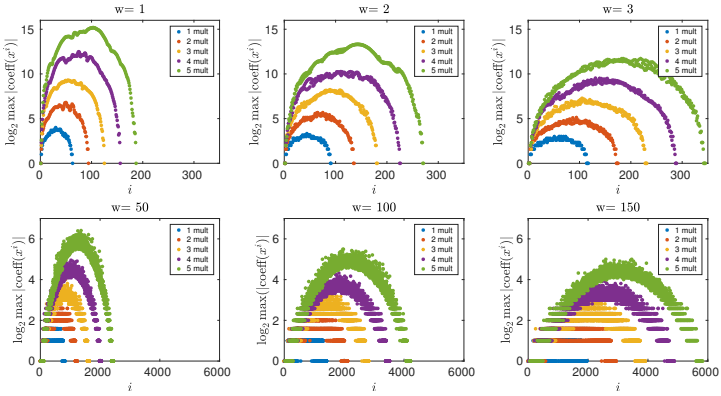


Fig. 5. $\log_2$ of the maximum absolute value of the coefficient of x^i seen during 10 000 products of two w-NIBNAF encodings of random numbers in $\left[-2^{40}, 2^{40}\right]$ against i.

imal size of the resulting coefficients becomes smaller. So the plots confirm the expected result that sparser encodings lead to a reduction in the size of the resulting coefficients after one multiplication. Next, we investigate the behaviour for an increasing amount of multiplications. In Figure 5 one observes that for a fixed number of multiplications the maximum coefficient, considering all coefficients in the resulting polynomial, decreases as w increases and the maximum degree of the polynomial increases as w increases. This confirms that increasing the degree of the polynomial, in order to make it more sparse, has the desirable effect of decreasing the size of the coefficients. Figure 5 also shows that based on the result of one multiplication we can even estimate the maximum value of the average coefficients of x^i for a specific number of multiplications by scaling the result for one multiplication.

To summarize, we plot the number of bits of the maximum coefficient of the polynomial that is the result of a certain fixed amount of multiplications as a function of w in Figure 6. From this figure we clearly see that the maximal coefficient decreases when w increases and hence the original encoding polynomial is sparser. In addition we see that the effect of the sparseness of the encoding on the size of the resulting maximal coefficient is bigger when the amount of multiplications increases. However the gain of sparser encodings decreases as w becomes bigger. Furthermore, Figure 6 shows that the bound given in Lemma 3 is much bigger than the observed upper bound we get from 10 000 samples.

4 Practical impact

We encounter the following constraints on the plaintext coefficient modulus t while homomorphically computing with polynomial encodings of finite precision real numbers. The first constraint comes from the correctness requirement of the SHE scheme: the noise inside the ciphertext should not exceed a certain level during the computations, otherwise decryption fails. Since an increase of the

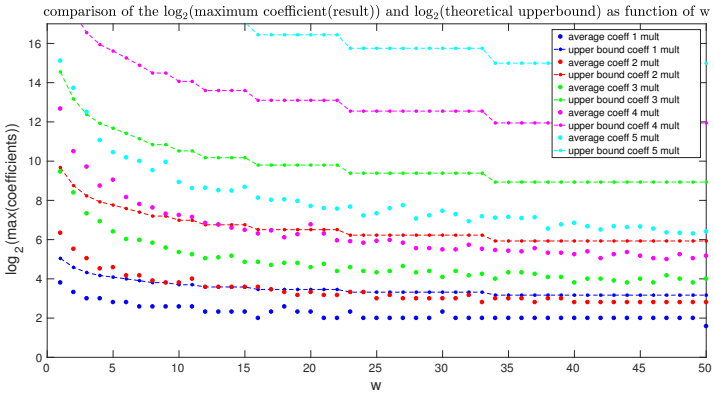


Fig. 6. $\log_2$ of the observed and theoretical maximum absolute coefficient of the result of multiplying w-NIBNAF encodings of random numbers in $\left[-2^{40}, 2^{40}\right]$ against w.

plaintext modulus expands the noise this places an upper bound on the possible t which can be used. The second constraint does not relate to SHE but to the circuit itself. After any arithmetic operation the polynomial coefficients tend to grow. Given that fact, one should take a big enough plaintext modulus in order to prevent or mitigate possible wrapping around modulo t. This determines a lower bound on the range of possible values of t. In practice, for deep enough circuits these two constraints are incompatible, i.e. there is no interval from which t can be chosen. However, the plaintext space R_t can be split into smaller rings $R_{t_1}, \ldots, R_{t_k}$ with $t = \prod_{i=1}^{k} t_i$ using the Chinese Remainder Theorem (CRT). This technique [8] allows us to take the modulus big enough for correct evaluation of the circuit and then perform k threads of the homomorphic algorithm over $\{R_{t_i}\}_i$. These k output polynomials will then be combined into the final output, again by CRT. This approach needs k times more memory and time than the case of a single modulus. Thus the problem is mostly about reducing the number of factors of t needed.

An a priori lower bound on t can be derived using the worst case scenario in which the final output has the maximal possible coefficient, which was analyzed in Section 3. If we use w-NIBNAF encodings for increasing values of w then this lower bound will decrease, eventually leading to fewer CRT factors; here a concern is not to take w too large to prevent wrapping around modulo $f(X)$. In practice though, we can take t considerably smaller because the worst case occurs with a negligible probability, which even decreases for circuits having a bigger multiplicative depth. Moreover, we can allow the least significant coefficients of the fractional part to wrap around modulo t with no harm to the final results.

In this section we revisit the homomorphic method for electricity load forecasting described in [6] and demonstrate that by using w-NIBNAF encodings, by ignoring the unlikely worst cases, and by tolerating minor precision losses we can reduce the number of CRT factors from $k = 13$ to $k = 1$, thereby enhancing its practical performance by a factor 13. We recall that [6] uses the Fan-Vercauteren

SHE scheme [23], along with the group method of data handling (GMDH) as a prediction tool; we refer to [6, §3] for a quick introduction to this method. Due to the fact that 80 percent of electricity meter devices in the European Union should be replaced with smart meters by 2020, this application may mitigate some emerging privacy and efficiency issues.

Experimental setup. For comparison's sake we mimic the treatment in [6] as closely as possible. In particular we also use the real world measurements obtained from the smart meter electricity trials performed in Ireland [14]. This dataset [14] contains observed electricity consumption of over 5000 residential and commercial buildings during 30 minute intervals. We use aggregated consumption data of 10 buildings. Given previous consumption data with some additional information, the GMDH network has the goal of predicting electricity demand for the next time period. Concretely, it requires 51 input parameters: the 48 previous measurements plus the day of the week, the month and the temperature. There are three hidden layers with 8, 4, 2 nodes, respectively. A single output node provides the electricity consumption prediction for the next half hour. Recall that a node is just a bivariate quadratic polynomial evaluation.

The plaintext space is of the form $R_t = \mathbb{Z}_t[X]/(X^{4096}+1)$, where the degree $d = 4096$ is motivated by the security level of 80 bits which is targetted in [6]; recent work by Albrecht [1] implies that the actual level of security is slightly less than that. Inside R_t the terms corresponding to the fractional parts and those corresponding to the integral parts come closer together after each multiplication. Wrapping around modulo $X^{4096}+1$, i.e. ending up at the left or at the right of the box depicted in Figure 1, means that inside R_t these integer and fractional parts start to overlap. In this case it is no longer possible to decode correctly. We encode the input data using approximate w-NIBNAF representations with a fixed number of integer and fractional digits. When increasing the window size w one should take into account that the precision of the corresponding encodings changes as well. To maintain the same accuracy of the algorithm it is important to keep the precision fixed, hence for bigger w's the smaller base b_w should result in an increase of the number of coefficients used by an encoding. Starting with the balanced ternary expansion (BTE), for any $w > 2$, the numbers $\ell(w)_i$ and $\ell(w)_f$ of integer and fractional digits should be expanded according to $\ell(w)_i = (\ell(\mathrm{BTE})_i - 1) \cdot \log_{b_w} 3 + 1$, $\ell(w)_f = -\lfloor \log_{b_w} e_f \rfloor$, where e_f is the maximal error of an approximate w-NIBNAF representation such that the prediction algorithm preserves the same accuracy. Empirically we found that the GMDH network demonstrates reasonable absolute and relative errors when $\ell(\mathrm{BTE})_i^{\mathrm{inp}} = 4$ and $e_f^{\mathrm{inp}} = 1$ for the input and $\ell(\mathrm{BTE})_i^{\mathrm{pol}} = 2$ and $e_f^{\mathrm{pol}} = 0.02032$ for the coefficients of the nodes (quadratic polynomials).

Results. The results reported in this section are obtained running the same software and hardware as in [6]: namely, FV-NFLlib software library [17] running on a laptop equipped with an Intel Core i5-3427U CPU (running at 1.80GHz). We performed 8560 runs of the GMDH algorithm with BTE, NAF and 950-NIBNAF. The last expansion is with the maximal possible w such that the resulting output

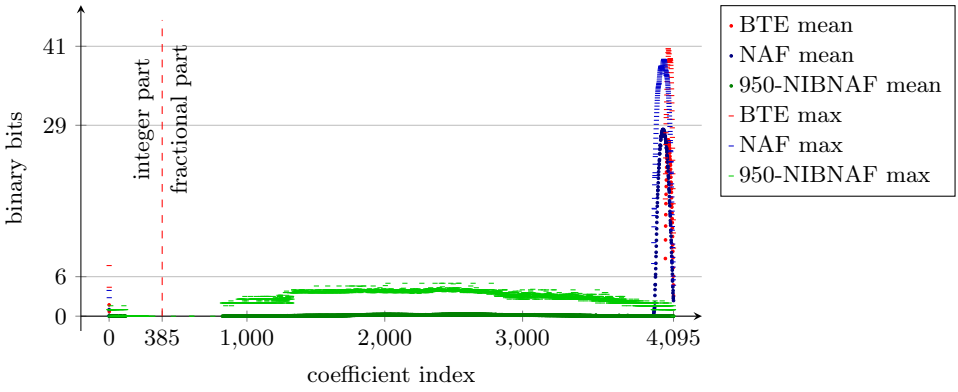


Fig. 7. The mean and the maximal size per coefficient of the resulting polynomial.

polynomial still has discernible integer and fractional parts. Correct evaluation of the prediction algorithm requires the plaintext modulus to be bigger than the maximal coefficient of the resulting polynomial. This lower bound for t can be deduced either from the maximal coefficient (in absolute value) appearing after any run or, in case of known distribution of coefficient values, from the mean and the standard deviation. In both cases increasing window sizes reduce the bound as depicted in Figure 7. Since negative encoding coefficients are used, 950-NIBNAF demands a plaintext modulus of 7 bits which is almost 6 times smaller than for BTE and NAF.

As expected, w-NIBNAF encodings have longer expansions for bigger w's and that disrupts the decoding procedure in [6,16]. Namely, they naively split the resulting polynomial into two parts of equal size. As one can observe in Figure 7, using 950-NIBNAF, decoding in this manner will not give correct results. Instead, the splitting index i_s should be shifted towards zero, i.e. to 385. To be specific [6, Lem. 1] states that i_s lies in the interval $(d_i + 1, d - d_f)$ where $d_i = 2^{r+1}(\ell(w)_i^{\text{inp}} + \ell(w)_i^{\text{pol}}) - \ell(w)_i^{\text{pol}}$ and $d_f = 2^{r+1}(\ell(w)_f^{\text{inp}} + \ell(w)_f^{\text{pol}}) - \ell(w)_f^{\text{pol}}$. Indeed, this is the worst case estimation which results in the maximal $w = 74$ for the current network configuration.

However the impact of the lower coefficients of the fractional part can be much smaller than the precision required by an application. In our use case the prediction value should be precise up to $e_f^{\text{inp}} = 1$. We denote the aggregated sum of lower coefficients multiplied by corresponding powers of the w-NIBNAF base as $L(j) = \sum_{i=j-1}^{i_s} a_i b_w^{-i}$. Then the omitted fractional coefficients a_i should satisfy $|L(i_c)| < 1$, where i_c is the index after which coefficients are ignored.

To find i_c we computed $L(j)$ for every index j of the fractional part and stored those sums for each run of the algorithm. For fixed j the distribution of $L(j)$ is bimodal with mean $\mu_{L(j)}$ and standard deviation $\sigma_{L(j)}$ (see Figure 8). Despite the fact that this unknown distribution is not normal, we naively approximate the prediction interval $[\mu_{L(j)} - 6\sigma_{L(j)}, \mu_{L(j)} + 6\sigma_{L(j)}]$ that will contain the future observation with high probability. It seems to be a plausible guess in

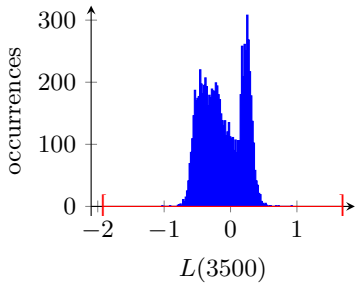


Fig. 8. The distribution of $L(3500)$ over 8560 runs of the GMDH algorithm and an approximation of its prediction interval in red.

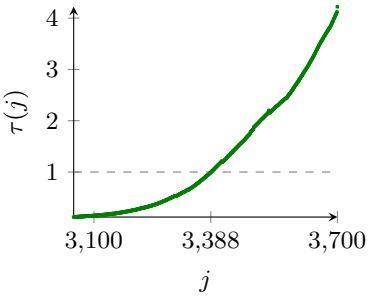


Fig. 9. The expected precision loss after ignoring fractional coefficients less than j.

	t	CRT factors	timing for one run
950-NIBNAF	$2^{5.044}$	1	2.57 s
BTE (this paper)	$2^{41.627}$	5	12.95 s
BTE [6]	$2^{103.787}$	13	32.5 s

Table 2. GMDH implementation with 950-NIBNAF and BTE [6]

this application because all observed $L(j)$ fall into that region with a big overestimate according to Figure 8. Therefore i_c is equal to the maximal j that satisfies $\tau(j) < 1$, where $\tau(j) = \max(|\mu_{L(j)} - 6\sigma_{L(j)}|, |\mu_{L(j)} + 6\sigma_{L(j)}|)$.

As Figure 9 shows, i_c is equal to 3388. Thus, the precision setting allows an overflow in any fractional coefficient a_j for $j < 3388$. The final goal is to provide the bound on t which is bigger than any a_j for $j \geq 3388$. Since the explicit distributions of coefficients are unknown and seem to vary among different indices, we rely in our analysis on the maximal coefficients occurring among all runs. Hence, the plaintext modulus should be bigger than $\max_{j \geq 3388}\{a_j\}$ over all resulting polynomials. Looking back at Figure 7, one can find that $t = 33$ suffices.

As mentioned above t is constrained in two ways: from the circuit and from the SHE correctness requirements. In our setup the ciphertext modulus is $q \approx 2^{186}$ and the standard deviation of noise is $\sigma = 102$, which together impose that $t \leq 396$ [6]. This is perfectly compatible with $t = 33$, therefore 950-NIBNAF allows us to omit the CRT trick and work with a single modulus, reducing the sequential timings by a factor 13. In the parallel mode it means that 13 times less memory is needed.

Additionally, these plaintext moduli are much smaller than the worst case estimation from Section 3. For 950-NIBNAF we take $d \in [542, 821]$ according to the encoding degrees of input data and network coefficients. Any such encoding contains only one non-zero coefficient. Consequently, any product of those encodings has only one non-zero coefficient which is equal to ± 1. When all mono-

mials of the GMDH polynomial result in an encoding with the same index of a non-zero coefficient, the maximal possible coefficient of the output encoding will occur. In this case the maximal coefficient is equal to the evaluation of the GMDH network with all input data and network coefficients being just 1. It leads to $t = 2 \cdot 6^{15} \simeq 2^{39.775}$.

One further consequence of smaller t is that one can reconsider the parameters of the underlying SHE scheme. Namely, one can take smaller q and σ that preserve the same security level and require a smaller bound on t instead of 396 taken above. Given $t = 33$ from above experiments, q reduces to 2^{154} together with $\sigma \approx 5$ that corresponds to smaller sizes of ciphertexts and faster SHE routines, where σ is taken the minimal possible to prevent the Arora-Ge attack [4] as long as each batch of input parameters is encrypted with a different key. Unfortunately, it is not possible to reduce the size of q by 32 bits in our implementation due to constraints of the FV-NFLlib library.

5 Conclusions

We have presented a generic technique to encode real numbers using a non-integral base. This encoding technique is especially suitable for use when evaluating homomorphic functions since it utilizes the large degree of the defining polynomial imposed by the security requirements. This leads to a considerably smaller growth of the coefficients and allows one to reduce the size of the plaintext modulus significantly, resulting in faster implementations. We show that in the setting studied in [6], where somewhat homomorphic function evaluation is used to achieve a privacy-preserving electricity forecast algorithm, the plaintext modulus can be reduced from about 2^{103} when using a balanced ternary expansion encoding, to $33 \simeq 2^{5.044}$ when using the encoding method introduced in this paper (non-integral base non-adjacent form with window size w), see Table 2. This smaller plaintext modulus means a factor 13 decrease in the running time of this privacy-preserving forecasting algorithm: closing the gap even further to making this approach suitable for industrial applications in the smart grid.

References

1. M. R. Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in helib and SEAL. In J. Coron and J. B. Nielsen, editors, *EUROCRYPT 2017*, volume 10211 of *LNCS*, pages 103–129, 2017.
2. I. Aliev. Siegel's lemma and sum-distinct sets. *Discrete Comput. Geom.*, 39(1-3):59–66, 2008.
3. E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. Post-quantum key exchange – a new hope. In *USENIX Security Symposium*. USENIX Association, 2016.
4. S. Arora and R. Ge. New algorithms for learning in presence of errors. In L. Aceto, M. Henzinger, and J. Sgall, editors, *ICALP 2011, Part I*, volume 6755 of *LNCS*, pages 403–415. Springer, Heidelberg, July 2011.
5. C. Bootland. Central Extended Binomial Coefficients and Sums of Powers. In preparation.

6. J. W. Bos, W. Castryck, I. Iliashenko, and F. Vercauteren. Privacy-friendly forecasting for the smart grid using homomorphic encryption and the group method of data handling. In M. Joye and A. Nitaj, editors, *AFRICACRYPT 2017*, volume 10239 of *LNCS*, pages 184–201, 2017.
7. J. W. Bos, C. Costello, M. Naehrig, and D. Stebila. Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In *IEEE S&P*, pages 553–570. IEEE Computer Society, 2015.
8. J. W. Bos, K. Lauter, J. Loftus, and M. Naehrig. Improved security for a ring-based fully homomorphic encryption scheme. In M. Stam, editor, *Cryptography and Coding 2013*, volume 8308 of *LNCS*, pages 45–64. Springer, 2013.
9. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In S. Goldwasser, editor, *ITCS 2012*, pages 309–325. ACM, Jan. 2012.
10. Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 505–524. Springer, Heidelberg, Aug. 2011.
11. J. H. Cheon, J. Jeong, J. Lee, and K. Lee. Privacy-preserving computations of predictive medical models with minimax approximation and non-adjacent form. In *Proceedings of WAHC 2017*, LNCS, 2017.
12. J. H. Cheon, A. Kim, M. Kim, and Y. Song. Homomorphic encryption for arithmetic of approximate numbers. Cryptology ePrint Archive, Report 2016/421, 2016. `http://eprint.iacr.org/2016/421`.
13. H. Cohen, A. Miyaji, and T. Ono. Efficient Elliptic Curve Exponentiation Using Mixed Coordinates. In K. Ohta and D. Pei, editors, *Advances in Cryptology – ASIACRYPT '98*, volume 1514 of *LNCS*, pages 51–65. Springer, 1998.
14. Commission for Energy Regulation. Electricity smart metering customer behaviour trials (CBT) findings report. Technical Report CER11080a, 2011. `http://www.cer.ie/docs/000340/cer11080(a)(i).pdf`.
15. A. Costache, N. P. Smart, and S. Vivek. Faster homomorphic evaluation of Discrete Fourier Transforms. *IACR Cryptology ePrint Archive*, 2016.
16. A. Costache, N. P. Smart, S. Vivek, and A. Waller. Fixed point arithmetic in SHE schemes. In *SAC 2016*, LNCS. Springer, 2016.
17. CryptoExperts. FV-NFLlib. `https://github.com/CryptoExperts/FV-NFLlib`, 2016.
18. A. de Moivre. *The doctrine of Chances*. Woodfall, 1738.
19. N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing. Manual for using homomorphic encryption for bioinformatics. Technical report, MSR-TR-2015-87, Microsoft Research, 2015.
20. N. Dowlin, R. Gilad-Bachrach, K. Laine, K. E. Lauter, M. Naehrig, and J. Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In M. Balcan and K. Q. Weinberger, editors, *International Conference on Machine Learning*, volume 48, pages 201–210. JMLR.org, 2016.
21. S. Eger. Stirling's Approximation for Central Extended Binomial Coefficients. *The American Mathematical Monthly*, 121:344–349, 2014.
22. L. Euler. De evolutione potestatis polynomialis cuiuscunque $(1 + x + x^2 + x^3 + x^4 + \text{etc.})^n$. *Nova Acta Academiae Scientarum Imperialis Petropolitinae*, 12:47–57, 1801.
23. J. Fan and F. Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012:144, 2012.

24. C. Gentry. Fully homomorphic encryption using ideal lattices. In M. Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009.
25. N. Göttert, T. Feller, M. Schneider, J. Buchmann, and S. A. Huss. On the design of hardware building blocks for modern lattice-based encryption schemes. In E. Prouff and P. Schaumont, editors, *CHES 2012*, volume 7428 of *LNCS*, pages 512–529. Springer, Heidelberg, Sept. 2012.
26. T. Güneysu, T. Oder, T. Pöppelmann, and P. Schwabe. Software speed records for lattice-based signatures. In P. Gaborit, editor, *PQCrypto 2013*, volume 7932 of *LNCS*, pages 67–82. Springer, 2013.
27. K. E. Lauter, A. López-Alt, and M. Naehrig. Private computation on encrypted genomic data. In D. F. Aranha and A. Menezes, editors, *LATINCRYPT 2014*, volume 8895 of *LNCS*, pages 3–27. Springer, Heidelberg, Sept. 2015.
28. A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *MATH. ANN*, 261:515–534, 1982.
29. V. Lyubashevsky, D. Micciancio, C. Peikert, and A. Rosen. SWIFFT: A modest proposal for FFT hashing. In K. Nyberg, editor, *FSE 2008*, volume 5086 of *LNCS*, pages 54–72. Springer, Heidelberg, Feb. 2008.
30. L. Mattner and B. Roos. Maximal probabilities of convolution powers of discrete uniform distributions. *Statistics & Probability Letters*, 78(17):2992 – 2996, 2008.
31. M. Naehrig, K. E. Lauter, and V. Vaikuntanathan. Can homomorphic encryption be practical? In C. Cachin and T. Ristenpart, editors, *ACM Cloud Computing Security Workshop – CCSW*, pages 113–124. ACM, 2011.
32. T. Pöppelmann and T. Güneysu. Towards practical lattice-based public-key encryption on reconfigurable hardware. In T. Lange, K. Lauter, and P. Lisonek, editors, *SAC 2013*, volume 8282 of *LNCS*, pages 68–85. Springer, Heidelberg, Aug. 2014.
33. G. W. Reitwiesner. *Binary Arithmetic*, volume 1 of *Advances in Computers*, pages 231–308. Academic Press, 1960.
34. D. Stehlé and R. Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 27–47. Springer, Heidelberg, May 2011.
35. J. W. Swanepoel. On a generalization of a theorem by Euler. *Journal of Number Theory*, 149:46–56, 2015.

A Proofs

Lemma 1 *For an integer $w \geq 1$ the polynomial $F_w(x) = x^{w+1} - x^w - x - 1$ has a unique positive root $b_w > 1$. The sequence $b_1, b_2, \ldots$ is strictly decreasing and $\lim_{w\to\infty} b_w = 1$. Further, $(x^2 + 1) \mid F_w(x)$ for $w \equiv 3 \mod 4$.*

Proof. For $w \geq 1$, $F'_w(x) = (w+1)x^w - wx^{w-1} - 1 = (x-1)((w+1)x^{w-1} + x^{w-2} + \cdots + 1)$ so that for $x \geq 0$ there is only one turning point of $F_w(x)$, at $x = 1$. Further, $F''_w(x) = (w+1)wx^{w-1} - w(w-1)x^{w-2}$, which takes the value $2w > 0$ at $x = 1$, so the turning point is a minimum. Since $F_w(0) = -1$ and $\lim_{x\to\infty} F_w(x) = \infty$ we conclude that there is a unique positive root of $F_w(x)$, $b_w > 1$, for any $w \geq 1$. Further, we have that $F_{w+1}(x) = xF_w(x) + x^2 - 1$ so that $F_{w+1}(b_w) = b_w^2 - 1 > 0$ so that $b_{w+1} < b_w$ and hence the sequence b_w is strictly decreasing and bounded below by 1 so must converge

to some limit, say $b_\infty \geq 1$. If $b_\infty > 1$ then as b_w is the positive solution to $x - 1 = (x+1)/x^w$ and, for $x \geq b_\infty > 1$, $\lim_{w\to\infty}(x+1)/x^w = 0$ we see that $b_\infty = \lim_{w\to\infty} b_w = 1$, a contradiction. Hence $b_\infty = 1$ as required. Finally we see that $F_w(x) = x(x-1)(x^{w-1}+1) - (x^2+1)$ and for $w = 4k+3$ that $x^{w-1}+1 = 1-(-x^2)^{2k+1} = (x^2+1)\sum_{i=0}^{2k}(-x^2)^i$ and hence $(x^2+1) \mid F_{4k+3}(x)$. □

Recall that to find a lower bound on the maximal absolute coefficient size we consider w-balanced ternary sequences and to each sequence (a_i) we have the corresponding polynomial $\sum_i a_i X^i$ in R_t. As we only look at the coefficients and their relative distances we can simply assume that to each w-balanced ternary sequence $c_0, c_1, \ldots, c_d$ of length $d+1$ we have the associated polynomial $c_0 + c_1 X + \ldots + c_d X^d$ of degree d. Multiplication of polynomials thus gives us a way of multiplying (finite) w-balanced ternary sequences. In the rest of this appendix we use the polynomial and sequence notation interchangeably.

Lemma 3 *The maximal absolute size of a term that can appear in the product of p arbitrary w-balanced ternary sequences of length $d+1$ is at least*

$$B_w(d,p) := \sum_{k=0}^{\lfloor\lfloor p\lfloor d/w\rfloor/2\rfloor/(\lfloor d/w\rfloor+1)\rfloor} (-1)^k \binom{p}{k}\binom{p-1+\lfloor p\lfloor d/w\rfloor/2\rfloor - k\lfloor d/w\rfloor - k}{p-1} .$$

Proof. Consider the product of p sequences all of which are equal to $m = 10\cdots010\cdots010\cdots0$ of length $d+1$, having $n := \lfloor d/w\rfloor + 1$ non-zero terms (all being 1) and between each pair of adjacent non-zero terms there are exactly $w-1$ zero terms. Note that n is the maximal number of non-zero terms possible. As polynomials we have that $m = \sum_{i=0}^{n-1} X^{iw} = \frac{1-X^{nw}}{1-X^w}$, and hence we have

$$\begin{aligned} m^p &= \left(\frac{1-X^{nw}}{1-X^w}\right)^p = (1-X^{nw})^p \cdot (1-X^w)^{-p} \\ &= \left(\sum_{i=0}^{p}(-1)^i\binom{p}{i}X^{inw}\right)\left(\sum_{j=0}^{\infty}\binom{p-1+j}{p-1}X^{jw}\right) \\ &= \sum_{\ell=0}^{\infty}\left(\sum_{k=0}^{\lfloor \ell/n\rfloor}(-1)^k\binom{p}{k}\binom{p-1+\ell-kn}{p-1}\right)X^{\ell w} , \end{aligned}$$

where we have used the substitution $(i,j) \to (k,\ell) = (i, in+j)$. Since we know that m^p has degree $p(n-1)w$ we can in fact change the infinite sum over ℓ to a finite one from $\ell = 0$ to $p(n-1)$. To give the tightest lower bound we look for the maximal coefficient of m^p. It is well known that this maximal coefficient occurs as the central coefficient, namely of x^ℓ where ℓ is any nearest integer to $p(n-1)/2$ and this gives us $B_w(d,p)$. □

Lemma 4 *Suppose w divides d, then $B_w(d,p)$ equals the maximal absolute size of a term that can be produced by taking the product of p arbitrary w-balanced ternary sequences of length $d+1$.*

Proof. Let $S_w(d,p)$ be the set of all sequences that are the product of p arbitrary w-balanced ternary sequences of length $d+1$. To prove the lemma we bound all the terms of any sequence in $S_w(d,p)$. For $i=0,\ldots,pd$ define

$$m_w(d,p,i) = \max\{\, |a_i| \;\mid\; a_i \text{ is the } i\text{'th term of a sequence in } S_w(d,p)\,\}\ .$$

Define $B_w(d,p,\ell) := \sum_{k=0}^{\lfloor \ell/n \rfloor} (-1)^k \binom{p}{k} \binom{p-1+\ell-kn}{p-1}$, the coefficient of $X^{\ell w}$ in m^p. We will prove by induction on p that $m_w(d,p,i) \le B_w(d,p,\lfloor i/w \rfloor)$. We will use the notation $C_i(f)$ for a polynomial f to denote the coefficient of X^i in $f(X)$; this is defined to be zero if $i > \deg(f)$ or $i<0$. Thus in this notation $B_w(d,p,\ell) = C_{\ell w}\left((1-X^{nw})^p/(1-X^w)^p\right)$. The base case $p=1$ is straight forward, all the $m_w(d,p,i)$ are equal to 1 by the definition of a w-balanced ternary sequence. We therefore suppose that $m_w(d,p-1,i) \le B_w(d,p-1,\lfloor i/w \rfloor)$ for $0 \le i \le (p-1)d$. Consider a product of p w-balanced ternary sequences of length $d+1$. It can be written as $f(X)e(X)$ where $f(X) \in S_w(d,p-1)$ and $e(X) \in S_w(d,1)$. We know that if $f(X) = \sum_{i=0}^{(p-1)d} a_i X^i$ then $|a_i| \le m_w(d,p-1,i)$ and if $e(X) = \sum_{j=0}^{d} \alpha_j X^j$ that $(fe)(X) = f(X)e(X) = \sum_{k=0}^{pd} \left(\sum_{i=\max(0,k-d)}^{\min((p-1)d,k)} a_i \alpha_{k-i} \right) X^k$, and due to the form of $e(X)$ we see that $|C_k(fe)| \le \sum_{j=1}^{n_k} |a_{i_j}| \le \sum_{j=1}^{n_k} m_w(d,p-1,i_j)$ for some $n_k \le n$, $\max(0,k-d) \le i_1 < i_2 < \cdots < i_{n_k} \le \min((p-1)d,k)$ and $i_{j+1} - i_j \ge w$ for $j=1,\ldots,n_k-1$.

The final condition on the i_j implies that the $\lfloor i_j/w \rfloor$ are distinct and since $m_w(d,p-1,i)$ is bounded above by $B_w(d,p-1,\lfloor i/w \rfloor)$, which depends only on $\lfloor i/w \rfloor$, we can recast this as

$$|C_k(fe)| \le \sum_{j=1}^{n_k} B_w(d,p-1,\ell_j) = \sum_{j=1}^{n_k} C_{\ell_j w}\left(\left(\frac{1-X^{nw}}{1-X^w} \right)^{p-1} \right)$$

where $\max(0, \lfloor k/w \rfloor - (n-1)) \le \ell_1 < \ell_2 < \cdots < \ell_{n_k} \le \min((p-1)(n-1), \lfloor k/w \rfloor)$ where we have used that $d/w = n-1$ is an integer.

Since $\lfloor k/w \rfloor - (\lfloor k/w \rfloor - (n-1)) + 1 = n$ we see that to make n_k as large as possible the ℓ_j must be the (at most n) consecutive integers in this range subject also to $0 \le \ell_1$ and $\ell_{n_k} \le (p-1)(n-1)$. Thus taking a maximum over all possible f and e we have

$$\begin{aligned} m_w(d,p,k) &\le \sum_{\ell=\lfloor k/w \rfloor-(n-1)}^{\lfloor k/w \rfloor} C_{\ell w}\left(\left(\frac{1-X^{nw}}{1-X^w} \right)^{p-1} \right) \\ &= \sum_{j=0}^{n-1} C_{\lfloor k/w \rfloor w}\left(\left(\frac{1-X^{nw}}{1-X^w} \right)^{p-1} X^{w(n-1-j)} \right) \\ &= C_{\lfloor k/w \rfloor w}\left(\left(\frac{1-X^{nw}}{1-X^w} \right)^{p} \right) = B_w(d,p,\lfloor k/w \rfloor)\ , \end{aligned}$$

which proves the inductive step. To finish the proof we note as before that the maximal value of $B_w(d,p,\lfloor k/w \rfloor)$ for $0 \le k \le pd$ is reached, for example, when $\lfloor k/w \rfloor = \lfloor p \lfloor d/w \rfloor /2 \rfloor$ and in this case we have $B_w(d,p)$ as required. □

Chapter 10

Homomorphic SIM^{2}D Operations: Single Instruction Much More Data.

Publication data

Castryck, W., Iliashenko, I., and Vercauteren, F. Homomorphic SIM^{2}D operations: Single instruction much more data. In *Advances in Cryptology – EUROCRYPT 2018, Part I* (Apr. / May 2018), J. B. Nielsen and V. Rijmen, Eds., vol. 10820 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 338–359.

Homomorphic SIM^{2}D Operations: Single Instruction Much More Data

Wouter Castryck, Ilia Iliashenko, and Frederik Vercauteren

imec-Cosic, Dept. Electrical Engineering, KU Leuven
firstname.lastname@esat.kuleuven.be

Abstract. In 2014, Smart and Vercauteren introduced a packing technique for homomorphic encryption schemes by decomposing the plaintext space using the Chinese Remainder Theorem. This technique allows to encrypt multiple data values simultaneously into one ciphertext and execute Single Instruction Multiple Data operations homomorphically. In this paper we improve and generalize their results by introducing a flexible Laurent polynomial encoding technique and by using a more fine-grained CRT decomposition of the plaintext space. The Laurent polynomial encoding provides a convenient common framework for all conventional ways in which input data types can be represented, e.g. finite field elements, integers, rationals, floats and complex numbers. Our methods greatly increase the packing capacity of the plaintext space, as well as one's flexibility in optimizing the system parameters with respect to efficiency and/or security.

1 Introduction

Homomorphic encryption allows to perform arithmetic operations on encrypted data without decryption. The idea stems from [26] where the authors introduced so-called 'privacy homomorphisms' from plaintext space to ciphertext space. In 2009, Gentry [21] presented the first fully homomorphic encryption scheme (FHE) using ideal lattices. This breakthrough result was followed by several variants and improvements [8,9,6,7,20,23] all using the same blueprint. One first constructs a *somewhat* homomorphic encryption (SHE) scheme that can homomorphically evaluate arithmetic circuits of limited depth and then turns this into a fully homomorphic scheme using a bootstrapping procedure. The security of these schemes relies on the presence of a noise component in the ciphertexts. This noise grows during arithmetic operations and eventually reaches a threshold beyond which the ciphertext can no longer be decrypted correctly. The bootstrapping procedure basically reduces the inherent noise by executing

This work was supported by the European Commission under the ICT programme with contract H2020-ICT-2014-1 644209 HEAT, and through the European Research Council under the FP7/2007-2013 programme with ERC Grant Agreement 615722 MOTMELSUM. The first author thanks Ghent University for its hospitality. The authors also thank the anonymous referees for some helpful remarks.

the decryption circuit homomorphically. Despite considerable effort in making bootstrapping more efficient [19,13,11,2], full fledged FHE is still rather slow, so implementers typically resort to using SHE schemes for practical applications.

The efficiency of homomorphic encryption schemes can be improved significantly by a judicious choice of plaintext space and encoding techniques for the common data types such as finite field elements, integers, rationals, floats and complex numbers. Concretely, throughout this paper we assume that the plaintext space is a ring of the form

$$R_t = \mathbb{Z}_t[X]/(\overline{f}(X))$$

where $t \geqslant 2$ is an integer called the plaintext modulus, and $\overline{f}(X)$ is the reduction modulo t of a monic irreducible polynomial $f(X) \in \mathbb{Z}[X]$ of degree $d \geqslant 1$. This setting is valid for most SHE schemes whose security relies on the Ring-LWE problem.[1] The degree d together with the ciphertext modulus q and the standard deviation σ of the initial noise distribution are the main security parameters, and these are typically determined by the required security level. The noise growth is influenced by d, q, σ, but also by the plaintext modulus t. A first optimization to decrease the noise growth is therefore to use a smaller plaintext space. Several encoding techniques [25,18,14,12,3,10] have been proposed whose goal is to 'spread out' the numerical input data as evenly as possible over the whole plaintext space, allowing for a smaller value of t. A second optimization, which can be combined with the first, is to decompose the plaintext space into smaller pieces using the Chinese Remainder Theorem (CRT) and run several computations in parallel [27,4]. Smart and Vercauteren [27] described how to carry out SIMD calculations in an SHE context by viewing R_t as the CRT composition of

$$\mathbb{Z}_t[X]/(\overline{f}_1(X)) \times \mathbb{Z}_t[X]/(\overline{f}_2(X)) \times \cdots \times \mathbb{Z}_t[X]/(\overline{f}_r(X)),$$

where $\overline{f}_1(X)\overline{f}_2(X)\cdots\overline{f}_r(X)$ is a factorization of $\overline{f}(X)$ into coprime factors. In fact, they concentrate on the case $t = 2$, but the above immediate generalization is discussed in [22]. We will refer to this decomposition of R_t as a *vertical* slicing of the plaintext space.

Contributions. Our first contribution is an improvement of the above SIMD approach by utilizing a more fine-grained CRT decomposition of the plaintext space. We do this by also taking into account factorizations of the plaintext modulus t. We will refer to the CRT decomposition

$$R_t \;\cong\; \mathbb{Z}[X]/(t_1, f(X)) \times \mathbb{Z}[X]/(t_2, f(X)) \times \cdots \times \mathbb{Z}[X]/(t_s, f(X)),$$

[1] A recent adaptation of the FV scheme due to Chen et al. [10] uses as plaintext modulus a linear polynomial $x - a$ instead of an integer t. The resulting plaintext space $R_{x-a} = \mathbb{Z}[X]/(X^n + 1, x - a) \cong \mathbb{Z}/(a^n + 1)$ has various nice features, both in terms of noise growth and in terms of packing capacity. However, the algebraic structure of R_{x-a} becomes more restrictive for CRT decomposition, so rings of this type will not be considered in this paper.

corresponding to a factorization $t = t_1 t_2 \cdots t_s$ into coprime factors, as a *horizontal* slicing of the plaintext space. The flexibility of our method stems partly from the fact that factorisations modulo the various t_i do not imply a global factorisation modulo t. This alternative type of slicing for SIMD purposes is not new (see e.g. [4]). However, by *combining* horizontal and vertical slicing as explained in Section 4, the plaintext space becomes subdivided in 'bricks' as depicted in Figure 4. In our SIMD approach, which we call SIM^{2}D, each data slot corresponds to a set of such bricks (called a block) rather than one vertical or horizontal slice as considered in previous works. This results in a much more flexible but, at the same time, denser packing as described in Section 5. In Section 6 we provide several tools that can help in making an optimal choice of blocks. This includes slight alterations to t and/or $f(X)$ that lead to more fine-grained decompositions.

Our second contribution is a novel encoding technique for Laurent polynomials into a plaintext space of the form $R_t = \mathbb{Z}_t[X]/(\overline{f}(X))$ that works for general $\overline{f}$ (under the mild assumption that $\overline{f}(0)$ is an invertible element of $\mathbb{Z}_t$). Previous work [15] could only deal with the very special case of 2-power cyclotomic polynomials, due to concerns of mixing of integral and fractional parts. Our encoding technique is explained in Section 3. Encoding elements of the Laurent polynomial ring $\mathbb{Z}[X^{\pm 1}]$ serves as a convenient common framework for all customary encoding techniques: indeed, under $X \mapsto b$ the Laurent polynomials specialize to b-ary expansions for any choice of base $b \in \mathbb{C}\backslash\{0\}$. This framework allows to encode common data types such as finite field elements, integers, rationals, floats and complex numbers. Furthermore, we show that choosing different bases b for different blocks can be useful in optimizing the data packing (see Section 6).

Our algorithms for encoding, packing, unpacking and decoding are easy to implement (pseudo-code is provided) and extremely flexible to use. The overall goal is to provide a set of tools which together can be used to perform SIMD in an optimal way, given the constraints on the plaintext space imposed by security, efficiency and correctness requirements.

2 Preliminaries

2.1 Basic notation

Vectors are denoted by bold letters such as $\mathbf{a}$ and when the individual coordinates are required, we write a row vector as $(a_1, \ldots, a_k)$. For a natural number r, we denote the set $\{1, \ldots, r\}$ by $[r]$. Similarly, for any $\ell, m \in \mathbb{Z}, \ell \leqslant m$, the set $\{\ell, \ell+1, \ldots, m-1, m\}$ is denoted by $[\ell, m]$. The quotient ring of integers modulo a natural number t is denoted $\mathbb{Z}_t$.

2.2 Laurent polynomials

Most common numerical types (integers, rational, real or complex numbers) are represented as (finite) power series expansions in a certain base $b \in \mathbb{C}\backslash\{0\}$, using

digits that are taken from some given subset of $\mathbb{Z}$. These expansions naturally correspond to Laurent polynomials with integral coefficients, i.e. elements of the ring $\mathbb{Z}[X^{\pm 1}]$.

Most frequently, an integral base $b > 1$ with digit set $\{0, \ldots, b-1\}$ is used in practice, such as binary $b = 2$ or ternary $b = 3$. For use in SHE schemes, several variations [18,14,12,3] have been proposed. For the purposes of this paper we mention the non-integral base non-adjacent form (NIBNAF) from [3] which is a very sparse expansion with respect to a real base $b \in (1, 2)$ and using the digit set $\{-1, 0, 1\}$. All of these expansions can be thought of as the evaluations at $X = b$ of a Laurent polynomial with integral coefficients.

Example 1. The real number 2.3 can be approximated in base $b = 2$ using digits in $\{0, 1\}$ as

$$2.3 \simeq 1 \cdot 2 + 1 \cdot 2^{-2} + 1 \cdot 2^{-5} + 1 \cdot 2^{-6},$$

which is the evaluation of the Laurent polynomial

$$1 \cdot X + 1 \cdot X^{-2} + 1 \cdot X^{-5} + 1 \cdot X^{-6} \in \mathbb{Z}[X^{\pm 1}]$$

at $X = b = 2$.

Recall that in general, any Laurent polynomial $a(X) \in \mathbb{Z}[X^{\pm 1}]$ can be written as

$$a(X) = a_\ell X^\ell + \cdots + a_{m-1} X^{m-1} + a_m X^m \tag{1}$$

where $a_i \in \mathbb{Z}$ for every $i \in [\ell, m]$, $a_\ell, a_m \neq 0$ and $\ell \leqslant m$. For a modulus t (which will be clear from the context) we write $\overline{a}(X)$ for the Laurent polynomial in $\mathbb{Z}_t[X^{\pm 1}]$ obtained by reducing all coefficients.

Definition 1. *For an integral Laurent polynomial $a(X) \in \mathbb{Z}[X^{\pm 1}]$ represented as in Equation* (1)*, we define the* bounding box *of $a(X)$ as the tuple (w, h) with $w = m - \ell$ and $h = \log_2(\max_i a_i - \min_i a_i + 1)$ the sizes of the exponent and the coefficient ranges of $a(X)$.*

We represent the bounding box graphically with a rectangle of width w and height h.

Fig. 1. The bounding box of a polynomial.

2.3 Plaintext space

Most SHE schemes utilize quotient rings of the form

$$R = \mathbb{Z}[X]/(f(X))$$

where $f(X) \in \mathbb{Z}[X]$ is a monic irreducible polynomial of degree d. The plaintext space is typically represented as a quotient ring $R_t = \mathbb{Z}_t[X]/(\overline{f}(X))$ for an integral plaintext modulus t. Similarly, the ciphertext space is defined as $R_q = \mathbb{Z}_q[X]/(\overline{f}(X))$ where $q \gg t$. Another important parameter is the standard deviation σ of the discretized Gaussian distribution from which the SHE encryption scheme samples its noise, which is embedded into the ciphertexts.

Typically, one first sets the parameters q, d and σ, primarily as functions of the security level, in order to prevent all known attacks on the underlying lattice problems [1]. Afterwards, the plaintext modulus t is selected, subject to two constraints. Firstly, it is bounded from above, which stems from the fact that the embedded noise grows during arithmetic operations up to a critical threshold above which ciphertexts can no longer be decrypted. Since the plaintext modulus directly affects the noise growth in ciphertexts, one can find a maximal t for which the decryption remains correct while evaluating a given arithmetic circuit $\mathcal{C}$. We denote this bound by $t_{\mathcal{C}}^{\max}$. If it is impossible to satisfy this bound then one can use the Chinese Remainder Theorem to split the computation into smaller parts, as explained in Remark 3; see also [4]. Secondly, as explained in the next section, the plaintext modulus t is bounded from below by some value $t_{\mathcal{C}}^{\min}$ which depends on the input data and on the way the latter is encoded, and which ensures correct decoding.

Remark 1. The values of q, d, σ are not uniquely determined by the security level. Therefore, one can try to use the remaining freedom to target a specific value of $t_{\mathcal{C}}^{\max}$. In the remainder of the paper, we will assume that $t_{\mathcal{C}}^{\max}$ is given, and our aim is to utilize the available plaintext space in an optimal way. One motivation for targeting maximal flexibility here is that it is not clear whether preselecting a precise value of $t_{\mathcal{C}}^{\max}$ is always possible in practice (e.g., for a fixed degree and security level it turns out that the value of $t_{\mathcal{C}}^{\max}$ stabilizes as $q \to \infty$). This is further impeded by the fact that concrete implementations often do not allow q and d to be picked from some continuous-like range (e.g., the `FV-NFLlib` [16] and the `SEAL` [24] libraries require that d is a power of 2 and that $\log_2 q$ is a multiple of some integer). A second motivation is that it can be desirable to use a single SHE implementation for encrypting batches of data of largely varying sizes. The plaintext space should be chosen to fit the largest data, and the methods presented below can then be used to optimize the handling of the smaller data.

The most common choice for $f(X)$ is a cyclotomic polynomial. The nth cyclotomic polynomial $\Phi_n(X) \in \mathbb{Z}[X]$ is the minimal polynomial of a primitive nth root of unity in $\mathbb{C}$

$$\Phi_n(X) = \prod_{0<k<n\,,\,(k,n)=1} (X - \zeta_n^k),$$

where $\zeta_n = e^{2\pi i/n}$. The degree of $\Phi_n(X)$ is equal to $\phi(n)$, where $\phi(n)$ is the totient function. It is always irreducible over $\mathbb{Z}$ and, additionally, $\Phi(0) = 1$ for $n \geqslant 3$.

Cyclotomic polynomials are often used by SHE implementers since they have very nice arithmetic properties such as fast modular reduction and simple Galois groups, which can be used to move data values in between data slots.

3 Plaintext encoding/decoding of Laurent polynomials

In this section we consider the problem of encoding an integral Laurent polynomial in the plaintext space and the reverse operation of decoding. We also give necessary conditions on the 'size' of the plaintext space such that a given circuit $\mathcal{C}$ can be evaluated correctly.

3.1 Encoding

Assume that the input data (integers, rationals, reals, ...) has been represented as a Laurent polynomial $a(X) \in \mathbb{Z}[X^{\pm 1}]$. Encoding such a Laurent polynomial in the plaintext space R_t has been considered in a series of recent works [18,14,12,3]. However, it was emphasized in [14] that the plaintext space should only be defined modulo a 2-power cyclotomic polynomial $f(X) = X^{2^k} + 1$ for some k. The reason for this restriction is that the authors required a small and sparse representation for X^{-1}, which in this case is given by $X^{-1} \equiv -X^{2^k-1} \bmod f(X)$.

Here we propose a very general way of encoding Laurent polynomials which works for almost all defining polynomials f. Let $\overline{f}(X)$ denote the reduction modulo t of $f(X)$ and assume that $f(0)$ is co-prime with t, so $\overline{f}(0)$ is invertible in $\mathbb{Z}_t$. Define $\overline{g}(X)$ by writing $\overline{f}(X) = \overline{g}(X)X + \overline{f}(0)$, then it is obvious that modulo $\overline{f}(X)$ we have that $X^{-1} \equiv -\overline{g}(X)\overline{f}(0)^{-1}$.

The encoding map $\mathsf{Encd}_{\overline{f}}$ is then given by the sequence of ring homomorphisms

$$\mathbb{Z}[X^{\pm 1}] \xrightarrow{\bmod t} \mathbb{Z}_t[X^{\pm 1}] \xrightarrow{\eta_{\overline{f}}} R_t$$

with

$$\eta_{\overline{f}} : \begin{array}{ll} X & \mapsto X \\ X^{-1} & \mapsto -\overline{g}(X)\overline{f}(0)^{-1}\,. \end{array}$$

Example 2. In the case of the 2-power cyclotomic polynomial $f(X) = X^{2^k} + 1$, the above map replaces negative powers X^{-j} by $-X^{d-j}$, which coincides with the approach from [14]: when expressed in terms of the basis $1, X, X^2, \ldots, X^{d-1}$ of R_t, the map $\eta_{\overline{f}}$ places the positive exponents at the low end of this range, and the negative exponents are placed at the high end.

3.2 Decoding

The crux of the construction relies on the fact that the above encoding map $\mathsf{Encd}_{\overline{f}}$ defines an isomorphism when restricted to a subset of Laurent polynomials. Indeed, if we choose a subset of $\mathbb{Z}_t[X^{\pm 1}]$ of the form

$$\mathbb{Z}_t[X^{\pm 1}]_\ell^m = \left\{ \sum_{i=\ell}^{m} \overline{a}_i X^i | \overline{a}_i \in \mathbb{Z}_t \right\}$$

with ℓ and m chosen such that $m - \ell + 1 = d$, then the restriction of $\eta_{\overline{f}}$ to $\mathbb{Z}_t[X^{\pm 1}]_\ell^m$ is an isomorphism between two free $\mathbb{Z}_t$-modules of rank d. The inverse of this map, denoted $\theta_{\overline{f},\ell,m}$, is easy to compute in practice, since it simply corresponds to a matrix inversion.

Thus, $\theta_{\overline{f},\ell,m}$ determines the decoding algorithm from R_t to Laurent polynomials over $\mathbb{Z}_t$. In the final step, one has to lift a Laurent polynomial from $\mathbb{Z}_t[X^{\pm 1}]$ to $\mathbb{Z}[X^{\pm 1}]$ by choosing a representative for each coefficient in a non-empty subset A of $\mathbb{Z}$ of size t. For simplicity we will always take $A = [z, z + t - 1]$ for some $z \in \mathbb{Z}$, common choices being $A = [-\lfloor (t-1)/2 \rfloor, \lceil (t-1)/2 \rceil]$ or $A = [0, t-1]$. But any set A of representatives would be possible, and in fact it can even depend on the coefficient under consideration. Together these two steps define the decoding map $\mathsf{Decd}_{\overline{f},\ell,m,A}$.

3.3 Correctness conditions

Since homomorphic encryption aims to perform arithmetic operations on ciphertexts, one usually deals with a ciphertext being the outcome of an arithmetic circuit involving only multiplications and additions. By the homomorphic property this ciphertext corresponds to a plaintext which is the result of the same operations in the plaintext space. Given a circuit $\mathcal{C}$, the result of its evaluation on encodings of Laurent polynomials $\mathbf{a} = (a_1(X), \ldots, a_k(X))$ is denoted by $\mathcal{C}(\mathsf{Encd}_{\overline{f}}(\mathbf{a})) \in R_t$.

To guarantee correctness of circuit evaluation, one has to make sure that there exist $\ell, m \in \mathbb{Z}$ such that $m - \ell + 1 = d$ and some non-empty set $A \subsetneq \mathbb{Z}$ of size at most t such that

$$\mathsf{Decd}_{\overline{f},\ell,m,A}(\mathcal{C}(\mathsf{Encd}_{\overline{f}}(\mathbf{a}))) = \mathcal{C}(\mathbf{a}),$$

where $\mathcal{C}(\mathbf{a})$ is the result of the same circuit evaluation in $\mathbb{Z}[X^{\pm 1}]$. This implies that the bounding box (w, h) of $\mathcal{C}(\mathbf{a})$ has to satisfy $w \leqslant m - \ell + 1 = d$ and $h \leqslant \log_2 |A| = \log_2 t$. In this case, we say that *the plaintext space covers the bounding box of* $\mathcal{C}(\mathbf{a})$ as shown on the figure below

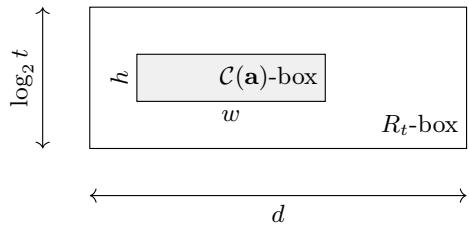


Fig. 2. The bounding box of $\mathcal{C}(\mathbf{a})$ is covered by the plaintext space R_t.

If the bounding box (w, h) of a Laurent polynomial has larger height than the plaintext space, i.e. $h > \log_2 t$, then we say that the computation *overflows modulo* t. If we end up with $w > d$ then we say that it *overflows modulo* $\overline{f}(X)$.

The parameters t and d should therefore be taken large enough to satisfy the above requirement. In practice, d is usually fixed by the security requirements of the SHE scheme. The choice for t, however, strongly depends on the arithmetic circuit $\mathcal{C}$ one is trying to evaluate. Initially, the input data of the circuit is encoded by Laurent polynomials whose bounding boxes are of height $h \leqslant \log_2(|\{b\text{-base digits}\}|)$. During arithmetic operations the height (typically) grows to the height of the bounding box of the outcome. For a given circuit $\mathcal{C}$, this defines a lower bound for t to guarantee correct decoding, which we denote $t_{\mathcal{C}}^{\min}$. Combined with the upper bound on t from Section 2.3, one obtains a range for t, namely $[t_{\mathcal{C}}^{\min}, t_{\mathcal{C}}^{\max}]$.

Example 3. To illustrate encoding and decoding, we take $R_t = \mathbb{Z}_7[X]/(\overline{f}(X))$ where $\overline{f}(X) = X^9 + 4X^7 + 1$. Thus, $\overline{g}(X) = X^8 + 4X^6$ and $\mathsf{Encd}_{\overline{f}}$ maps X^{-1} to $6X^8 + 3X^6$. Let us multiply two rational numbers, $\frac{182}{243}$ and 1476. Their base-3 expansions are as follows

$$\frac{182}{243} = 2 \cdot 3^{-5} + 2 \cdot 3^{-3} + 2 \cdot 3^{-1}, \qquad 1476 = 2 \cdot 3^2 + 2 \cdot 3^6$$

or as Laurent polynomials

$$a = 2X^{-5} + 2X^{-3} + 2X^{-1}, \qquad b = 2X^2 + 2X^6.$$

Applying $\mathsf{Encd}_{\overline{f}}$ we get encodings of a and b in R_t, namely, $\overline{a} = 6X^2 + 4X^4 + 4X^6 + 5X^8$ and $\overline{b} = 2X^2 + 2X^6$. Their product is equal to

$$\overline{c} = X + 4X^3 + 5X^4 + 4X^5 + X^6 + 3X^8.$$

We take $\ell = -3$, $m = 5$ and $A = \{4, 5, \ldots, 10\}$ in order to keep the product inside the box. Now we can define $\mathsf{Decd}_{\overline{f},-3,5,A}$. The first step is to construct a linear operator $\theta_{\overline{f},-3,5}$ using the inverse of the matrix defining the restriction of

$\eta_{\overline{f}}$ on $\mathbb{Z}_7[X^{\pm 1}]^5_{-3}$:

$$\begin{bmatrix} 0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&1&0&0 \\ 3&0&0&0&0&0&0&1&0 \\ 0&3&0&0&0&0&0&0&1 \\ 6&0&3&0&0&0&0&0&0 \\ 0&6&0&0&0&0&0&0&0 \\ 0&0&6&0&0&0&0&0&0 \end{bmatrix}^{-1} = \begin{bmatrix} 0&0&0&0&0&0&6&0&4 \\ 3&0&0&0&0&0&0&6&0 \\ 0&0&0&0&0&0&0&0&6 \\ 1&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&3&0&2 \\ 5&0&0&0&0&1&0&3&0 \end{bmatrix} \in \mathbb{Z}_7^{9\times 9}.$$

Then $\overline{c}$ is mapped to a Laurent polynomial $4X^{-3} + 4X^{-1} + X + 4X^3 + 4X^5 \in \mathbb{Z}_7[X^{\pm 1}]$. By looking for representatives of the coefficients in the set A we get $4X^{-3} + 4X^{-1} + 8X + 4X^3 + 4X^5 \in \mathbb{Z}[X^{\pm 1}]$ and evaluate it at $X = 3$

$$4 \cdot 3^{-3} + 4 \cdot 3^{-1} + 8 \cdot 3 + 4 \cdot 3^3 + 4 \cdot 3^5 = \frac{29848}{27},$$

which is the correct product of $\frac{182}{243}$ and 1476.

Remark 2. Note that the above condition for correct decoding *only* depends on the bounding box of the evaluation of the circuit $\mathcal{C}(\mathbf{a})$ and not on the bounding boxes of the individual inputs $a_i(X) \in \mathbb{Z}[X^{\pm 1}]$ nor on those of the intermediate values. Indeed, we always have

$$\mathcal{C}(\mathsf{Encd}_{\overline{f}}(a_1(X)), \ldots, \mathsf{Encd}_{\overline{f}}(a_k(X))) = \mathsf{Encd}_{\overline{f}}(\mathcal{C}(\mathbf{a})), \tag{2}$$

simply because $\mathsf{Encd}_{\overline{f}}$ is a ring homomorphism. This implies that the bounding boxes of the input or intermediate values should not necessarily be contained in the bounding box of the plaintext space, as long as the outcome of evaluation is.

4 Splitting the plaintext space

In this section we recall how the Chinese Remainder Theorem (CRT) can be used to split the plaintext space naturally along two directions: firstly, we will split horizontally for each prime power factor t_i of the plaintext modulus t and secondly, each horizontal slice will be split vertically by factoring $f(X) \bmod t_i$.

4.1 Horizontal splitting

If t is a composite that factors into distinct prime powers $t = t_1 \ldots t_s$ then the ring R_t can be mapped via the CRT to a direct product of R_{t_i}'s resulting in the following ring isomorphism

$$\begin{aligned} \mathsf{CRT}_t : R_t &\to R_{t_1} \times \cdots \times R_{t_s} \\ \overline{a}(X) &\mapsto (\overline{a}(X) \bmod t_1,\ \ldots, \overline{a}(X) \bmod t_s) \end{aligned}$$

whose inverse is easy to compute. For a given index subset $I = \{i_1, \ldots, i_c\} \subseteq [s]$ the map CRT_t induces a surjective morphism

$$\mathsf{CRT}_{t_I} : R_t \to R_{t_{i_1}} \times \cdots \times R_{t_{i_c}},$$

which is well-defined via the projection map

$$\pi_{t_I} : \prod_{i \in [s]} R_{t_i} \to \prod_{i \in I} R_{t_i}$$

so that $\mathsf{CRT}_{t_I} = \pi_{t_I} \cdot \mathsf{CRT}_t$. The CRT_t can be represented as a 'horizontal' splitting of the plaintext space according to the unique factorization of t into distinct prime powers $\{t_i\}_{i \in [s]}$. Each horizontal slice in Figure 3 corresponds to some R_{t_i}.

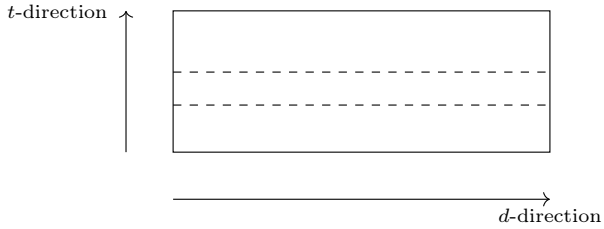


Fig. 3. CRT_t decomposition of R_t

4.2 Vertical splitting

For each factor t_i of t we define $\overline{f}_i(X) \in \mathbb{Z}_{t_i}[X]$ to be the reduction of $f(X)$ modulo t_i. Since $f(0)$ is co-prime with t, it is also co-prime with any t_i and thus, $\overline{f}_i(0)$ is invertible.

The factorization of $\overline{f}_i(X)$ into irreducible factors modulo t_i can be computed as follows: if t_i is prime, then one can simply use factorization algorithms for polynomials over finite fields; for t_i a prime power, one first computes the factorization modulo the prime and then lifts it using Hensel's lemma to a factorization modulo t_i. The result in both cases is that we can easily obtain a factorization

$$\overline{f}_i(X) \equiv \prod_{j=1}^{r_i} \overline{f}_{ij}(X)$$

for monic irreducible polynomials $\overline{f}_{ij}(X) \in \mathbb{Z}_{t_i}[X]$. Note that the constant terms $\overline{f}_{ij}(0)$ are all invertible because their product $\overline{f}_i(0)$ is invertible. Applying the CRT in the polynomial dimension gives the following map for each t_i:

$$\begin{aligned} \mathsf{CRT}_{t_i, \overline{f}_i} : R_{t_i} &\to R_{t_i,1} \times \cdots \times R_{t_i,r_i} \\ \overline{a}(X) &\mapsto (\overline{a}(X) \bmod \overline{f}_{i1}(X), \ldots, \overline{a}(X) \bmod \overline{f}_{ir_i}(X)). \end{aligned}$$

Here the $R_{t_i,j}$ denotes the ring $\mathbb{Z}_{t_i}[X]/(\overline{f}_{ij}(X))$, which corresponds to a 'brick' in Figure 4. The map $\mathsf{CRT}_{t_i,\overline{f}_i}$, whose inverse is again easy to compute, can be thought of as a 'vertical' splitting of R_{t_i}. For simplicity we will usually just write $R_{i,j}$ rather than $R_{t_i,j}$. By analogy with CRT_{tI}, we introduce the surjective ring homomorphism $\mathsf{CRT}_{t_i,\overline{f}_J}$ from R_{t_i} to $\prod_{j\in J} R_{t_i,j}$ where $J = \{j_1, \ldots, j_c\} \subseteq [r_i]$.

5 Improved SIMD encoding

In this section we combine the results of Sections 3 and 4 to derive flexible SIMD encoding and decoding algorithms. Recall that to correctly decode the result of a circuit evaluation $\mathcal{C}(\mathbf{a})$, we require that the bounding box of the plaintext space covers the bounding box of $\mathcal{C}(\mathbf{a})$. We assume that this is indeed the case, and show how to select a minimal number of bricks of R_t to cover the bounding box of $\mathcal{C}(\mathbf{a})$, leaving the other bricks available for doing parallel computations.

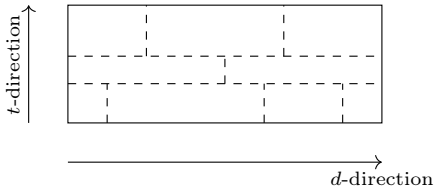


Fig. 4. Decomposition of R_t using factorization of t and $\overline{f}_i$'s

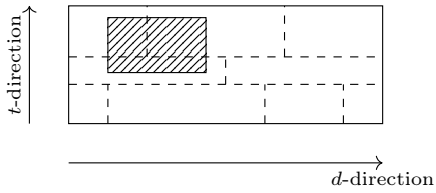


Fig. 5. Encoding of a single Laurent polynomial into R_t.

Recall that each brick corresponds to a ring $R_{i,j}$ in the decomposition

$$R_t \to R_{t_1} \times \cdots \times R_{t_s} \to (R_{1,1} \times \cdots \times R_{1,r_1}) \times \cdots \times (R_{s,1} \times \cdots \times R_{s,r_s}).$$

Each ring $R_{i,j}$ has its own bounding box of size $(d_{ij}, \log_2 t_i)$, where $d_{ij} = \deg \overline{f}_{ij}$. Assuming that the bounding box of $\mathcal{C}(\mathbf{a})$ is given by (w, h), we need to combine enough horizontal slices to cover the height h, and inside each horizontal slice, we need to select enough bricks to cover the width w as illustrated in Figure 5. Any unused bricks can be used to encode other data values, for instance to compute $\mathcal{C}(\mathbf{b})$ for some other input vector $\mathbf{b}$, immediately resulting in SIMD computations.

We formalize this approach by combining bricks into a block structure: we call a *block* a set of tuples $\mathcal{B} = \{(t_i, \overline{f}_{ij})\}_{i\in I(\mathcal{B}), j\in J(\mathcal{B},i)}$ with index sets $I(\mathcal{B}) \subseteq [s]$ and $J(\mathcal{B}, i) \subseteq [r_i]$, where we recall that r_i is the number of irreducible factors of $\overline{f}_i$. We of course think of this as corresponding to the set of $R_{i,j}$'s with $i \in I(\mathcal{B}), j \in J(\mathcal{B}, i)$. Equivalently, through an application of the CRT this corresponds to the set of quotient rings $\{R_{t_i}/(\overline{F}_{i,\mathcal{B}})\}_{i\in I(\mathcal{B})}$ where $\overline{F}_{i,\mathcal{B}} = \prod_{j\in J(\mathcal{B},i)} \overline{f}_{ij}$. Graphically we think of a block as a set of bricks of R_t, which are combined such that the

$R_{i,j}$'s with the same index i are glued column-wise and the resulting rows are placed on top of each other.

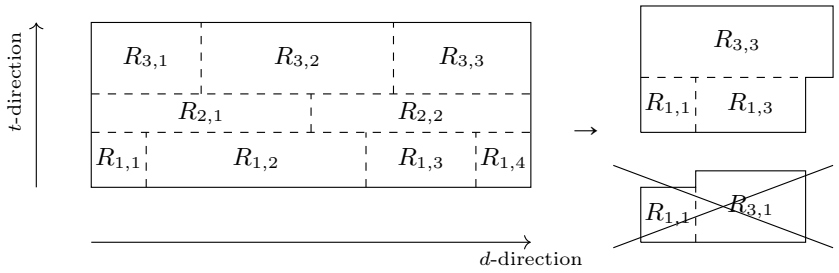


Fig. 6. Example of a block taken from the CRT decomposition of R_t. The bottom combination of 'bricks' is not a block because their first indices do not coincide.

In order for a block $\mathcal{B}$ to be suitable for computing $\mathcal{C}(\mathbf{a})$, whose bounding box we denote by (w, h), we note that the bounding box of $R_{t_i}/(\overline{F}_{i,\mathcal{B}})$ with $i \in I(\mathcal{B})$ is $(w_{i,\mathcal{B}}, \log_2 t_i)$ where

$$w_{i,\mathcal{B}} = \deg \overline{F}_{i,\mathcal{B}} = \sum_{j \in J(\mathcal{B},i)} d_{ij}.$$

If $\min_{i \in I(\mathcal{B})} w_{i,\mathcal{B}} \geqslant w$ and $\sum_{i \in I(\mathcal{B})} \log_2 t_i \geqslant h$ then we say that $\mathcal{B}$ *covers the bounding box* (w, h). As we will see $\mathcal{C}(\mathbf{a})$ will be decoded correctly as soon as an encoding block $\mathcal{B}$ is used that covers its bounding box.

Example 4. We decompose $R_t = \mathbb{Z}_{2761}[X]/(f(X))$ where $f(X) = X^{20} + X^{15} + 1$. The plaintext modulus factors into $t_1 = 11$ and $t_2 = 251$ and

$$\begin{aligned} f(X) &\equiv f_{1,1}(X) \cdot f_{1,2}(X) \\ &\equiv (X^5 + 3)(X^{15} + 9X^{10} + 6X^5 + 4) \bmod 11, \\ f(X) &\equiv f_{2,1}(X) \cdot f_{2,2}(X) \cdot f_{2,3}(X) \\ &\equiv (X^5 + 18)(X^5 + 120)(X^{10} + 114X^5 + 180) \bmod 251. \end{aligned}$$

Accordingly, R_t splits into $(R_{1,1} \times R_{1,2}) \times (R_{2,1} \times R_{2,2} \times R_{2,3})$. Overall we have 5 'bricks' that can be combined into 31 different blocks. For example, one can take a block $\{(11, X^{15}+9X^{10}+6X^5+4), (251, X^5+18), (251, X^5+120)\}$ corresponding to the combination of $R_{1,2}, R_{2,1}$ and $R_{2,2}$ or $\{(11, X^5 + 3), (11, X^{15} + 9X^{10} + 6X^5 + 4)\}$ which simply corresponds to $R_{11} = R_t/(11)$ (see Figure 7).

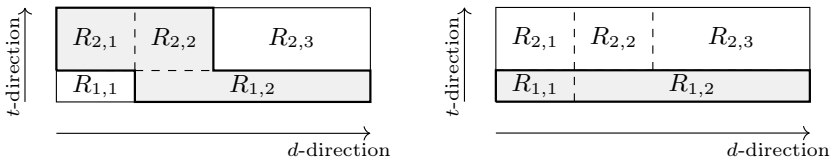


Fig. 7. The block structure of $R_t = \mathbb{Z}[X]/(2651, X^{20}+X^{15}+1)$ with two blocks colored in gray.

The whole plaintext space can be represented by a block as well

$$\mathcal{P} = \bigcup_{i\in[s]} \bigcup_{j\in[r_i]} \{(t_i, \overline{f}_{ij})\}.$$

Therefore, the SIMD packing problem consists in finding a set of disjoint blocks $S = \{\mathcal{B}_1, \ldots, \mathcal{B}_u\}$ such that $\bigcup_{\mathcal{B}\in S} \mathcal{B} = \mathcal{P}$ and every block covers the maximal bounding box among the corresponding output values.

To a partition S of $\mathcal{P}$ there naturally corresponds a factorization of $\overline{f}_i$ for every $i \in [s]$:

$$\overline{f}_i(X) = \prod_{\mathcal{B}\in S, i\in I(\mathcal{B})} \overline{F}_{i,\mathcal{B}}(X).$$

This induces a family of CRT isomorphisms

$$\mathsf{CRT}_{t_i,\overline{f}_i,S} : R_{t_i} \to \prod_{\mathcal{B}\in S, i\in I(\mathcal{B})} R_{t_i}/(\overline{F}_{i,\mathcal{B}}).$$

Now we have all the ingredients to pack a number of data values into one plaintext as described in Algorithm 1.

Algorithm 1: Plaintext packing.

Input : a set of disjoint blocks $S = \{\mathcal{B}_1, \ldots, \mathcal{B}_u\}$ with corresponding data values $a_1, \ldots, a_u \in \mathbb{Z}[X^{\pm 1}]$ such that $\bigcup_{k=1}^{u} \mathcal{B}_k = \mathcal{P}$.

Output: $b \in R_t$

for $k \leftarrow 1$ **to** u **do**
  for $i \in I(\mathcal{B}_k)$ **do**
    $a_{t_i,\overline{F}_{i,\mathcal{B}_k}} \leftarrow \mathsf{Encd}_{\overline{F}_{i,\mathcal{B}_k}}(a_k)$
for $i \leftarrow 1$ **to** s **do**
  $b_i \leftarrow \mathsf{CRT}^{-1}_{t_i,\overline{f}_i,S}(\{a_{t_i,\overline{F}_{i,\mathcal{B}}}\}_{\mathcal{B},i\in I(\mathcal{B})})$
$b \leftarrow \mathsf{CRT}^{-1}_{t}(b_1, \ldots, b_s)$

After packing one can encrypt the output and feed it to an arithmetic circuit (together with other packings in case the circuit takes more than one argument). The resulting plaintext contains multiple evaluations corresponding to each block that can be decoded using Algorithm 2.

Algorithm 2: Plaintext decoding for one block.

Input : a plaintext $\overline{c} \in R_t$, a block $\mathcal{B}$, an exponent range $[\ell, m]$ and a coefficient set $A \in \mathbb{Z}$

Output: a Laurent polynomial $a \in \mathbb{Z}[X^{\pm 1}]$

$t_I \leftarrow 1$
for $i \in I(\mathcal{B})$ **do**
  $t_I \leftarrow t_I \cdot t_i$
  $\overline{c}_i \leftarrow \overline{c} \bmod t_i$
  $\overline{c}_i \leftarrow \overline{c}_i \bmod \overline{F}_{i,\mathcal{B}}$
  $m_i \leftarrow \ell + w_{i,\mathcal{B}} - 1$
  $c_i \leftarrow \theta_{\overline{F}_{i,\mathcal{B}},\ell,m_i}(\overline{c}_i)$
$a \leftarrow$ coefficient-wise CRT^{-1} of $\{c_i\}_{i \in I(\mathcal{B})}$ to $\mathbb{Z}_{t_I}[X^{\pm 1}]$
$a \leftarrow$ selecting coefficient representatives of a from the set A

Algorithm 2 produces correct circuit evaluations for all blocks occurring in Algorithm 1 that satisfy the properties outlined in the next theorem.

Theorem 1. *Let S be a set of disjoint blocks such that $\bigcup_{\mathcal{B} \in S} \mathcal{B} = \mathcal{P}$. Let $\mathcal{C}$ be an arithmetic circuit taking v arguments and for each block $\mathcal{B}$ let $\mathbf{a}_\mathcal{B} = (a_{\mathcal{B},1}, \ldots, a_{\mathcal{B},v})$ be a vector of Laurent polynomials. For each $k = 1, \ldots, v$ let b_k denote the output of Algorithm 1 upon input of $(a_{\mathcal{B},k})_{\mathcal{B} \in S}$. Let $\overline{c} = \mathcal{C}(b_1, \ldots, b_v)$. Then for each block $\mathcal{B}$ we have that if it covers the bounding box of $\mathcal{C}(\mathbf{a}_\mathcal{B})$, then upon input of $\overline{c}$ Algorithm 2 produces $\mathcal{C}(\mathbf{a}_\mathcal{B})$, for an appropriate choice of ℓ, m and A.*

Proof. By our assumption there are ℓ, m such that $\mathcal{C}(\mathbf{a}_\mathcal{B}) = \sum_{i=\ell}^{m} \alpha_i X^i$ where

$$\min_{i \in I(\mathcal{B})} w_{i,\mathcal{B}} \geqslant m - \ell + 1 \qquad \text{and} \qquad \prod_{i \in I(\mathcal{B})} t_i \geqslant |A|, \tag{3}$$

with $A = \{\min_i \alpha_i, \ldots, \max_i \alpha_i\}$. Let a denote the output of Algorithm 2 upon input of $\overline{c}$ using these ℓ, m, and A. Since this is a Laurent polynomial having coefficients in A, by (3) it suffices to prove that the reductions of a and $\mathcal{C}(\mathbf{a}_\mathcal{B})$ modulo t_i are the same for each $i \in I(\mathcal{B})$. Again by (3) these reductions are contained in $\mathbb{Z}_{t_i}[X^{\pm 1}]_\ell^{m_i}$ where $m_i = \ell + w_{i,\mathcal{B}} - 1$, so by injectivity of $\eta_{\overline{F}_{i,\mathcal{B}}}$ it suffices to prove that

$$\mathsf{Encd}_{\overline{F}_{i,\mathcal{B}}}(a) = \mathsf{Encd}_{\overline{F}_{i,\mathcal{B}}}(\mathcal{C}(\mathbf{a}_\mathcal{B})).$$

From Algorithm 2 we see that the left-hand side is just the reduction of $\overline{c}$ into $R_{t_i}/(\overline{F}_{i,\mathcal{B}})$, while the right hand side is

$$\mathcal{C}(\mathsf{Encd}_{\overline{F}_{i,\mathcal{B}}}(a_{\mathcal{B},1}), \ldots, \mathsf{Encd}_{\overline{F}_{i,\mathcal{B}}}(a_{\mathcal{B},v}))$$

because of the homomorphic properties of the encoding map. From Algorithm 1 we clearly see that $\mathsf{Encd}_{\overline{F}_{i,\mathcal{B}}}(a_{\mathcal{B},k})$ is the reduction of b_k into $R_{t_i}/(\overline{F}_{i,\mathcal{B}})$, for all $k = 1, \ldots, v$, so the theorem follows. ▫

Example 5. Using the CRT decomposition of R_t from Example 4 we cube two Laurent polynomials simultaneously using SIMD, namely $u(X) = 7X^3 + 7X^2$ and $v(X) = 8X^5 + 7X$. To encode u^3 we take the block $\mathcal{B}_1$ with rings $R_{1,1}, R_{2,1}$ and the remaining bricks to build the block $\mathcal{B}_2$ to hold the result v^3.

Since only positive exponents are present in the data, all encoding functions $\mathsf{Encd}_{\overline{F}_{i,\mathcal{B}_1}}$ and $\mathsf{Encd}_{\overline{F}_{i,\mathcal{B}_2}}$ map $u(X)$ and $v(X)$ identically to the corresponding $R_{i,j}$'s. Then we get

$$\begin{aligned}
&a_{11,\overline{F}_{1,\mathcal{B}_1}}(X) = 7X^3 + 7X^2 \ \in R_{1,1} = R_{11}/(X^5 + 3),\\
&a_{251,\overline{F}_{2,\mathcal{B}_1}}(X) = 7X^3 + 7X^2 \in R_{2,1} = R_{251}/(X^5 + 18),\\
&a_{11,\overline{F}_{1,\mathcal{B}_2}}(X) = 8X^5 + 7X \ \ \in R_{1,2} = R_{11}/(X^{15} + 9X^{10} + 6X^5 + 4),\\
&a_{251,\overline{F}_{2,\mathcal{B}_2}}(X) = 8X^5 + 7X \ \ \in R_{2,2} \times R_{2,3} \cong R_{251}/(X^{15} + 234X^{10} + 55X^5 + 14).
\end{aligned}$$

Applying $\mathsf{CRT}^{-1}_{t_i,\overline{f}_i,\{\mathcal{B}_1,\mathcal{B}_2\}}$ for each t_i we find

$$\begin{aligned}
b_1 = {} & X^{18} + X^{17} + 10X^{16} + 5X^{15} + 9X^{13} + 9X^{12}\\
& + 2X^{11} + X^{10} + 6X^8 + 6X^7 + 5X^6 + 5X^5 + 4X^3 + 4X^2 + 3X + 9 \in R_{11},\\
b_2 = {} & 162X^{18} + 162X^{17} + 89X^{16} + 213X^{15} + 7X^{13} + 7X^{12}\\
& + 244X^{11} + 144X^{10} + 125X^8 + 125X^7 + 126X^6 + 177X^5\\
& + 9X^3 + 9X^2 + 249X + 221 \in R_{251},
\end{aligned}$$

which finally leads to the following plaintext via CRT_t^{-1}

$$\begin{aligned}
b = {} & 2421X^{18} + 2421X^{17} + 340X^{16} + 1468X^{15} + 2517X^{13} + 2517X^{12}\\
& + 244X^{11} + 144X^{10} + 2635X^8 + 2635X^7 + 126X^6 + 2436X^5\\
& + 2017X^3 + 2017X^2 + 751X + 1978 \in R_{2761}.
\end{aligned}$$

Now we evaluate an arithmetic circuit $z \mapsto z^3$ in b and obtain

$$\begin{aligned}
\overline{c} = {} & 1943X^{19} + 401X^{18} + 745X^{17} + 391X^{16} + 433X^{15}\\
& + 2109X^{14} + 1717X^{13} + 2646X^{12} + 2729X^{11} + 2347X^{10}\\
& + 2198X^9 + 1724X^8 + 234X^7 + 421X^6 + 2683X^5 + 94X^4\\
& + 1188X^3 + 1143X^2 + 1960X + 1906 \in R_{2761},
\end{aligned}$$

which simultaneously encodes u^3 and v^3.

In order to decode the data we apply Algorithm 2 starting with the block $\mathcal{B}_1$ equipped with the exponent range $[6, 9]$ and the coefficient set $A_{\mathcal{B}_1} = [0, 2760]$. At first, we should reduce $\overline{c}$ modulo $\overline{F}_{i,\mathcal{B}_1}$ and t_i for each $i \in I(\mathcal{B}_1)$. As a result, we find

$$\begin{aligned}
&\overline{c}_{1,\mathcal{B}_1} = 5X^4 + 4X^3 + 4X^2 + 5X \in R_{11}/(X^5 + 3),\\
&\overline{c}_{2,\mathcal{B}_1} = 101X^4 + 52X^3 + 52X^2 + 101X \in R_{251}/(X^5 + 18).
\end{aligned}$$

To decode into Laurent polynomials we set $\ell_i = 6$ and $m_i = 10$ for every $i \in I(\mathcal{B}_1)$ because $\deg \overline{F}_{1,\mathcal{B}_1} = \deg \overline{F}_{2,\mathcal{B}_1} = 5$. Then we follow the same procedure as in Example 3 to define $\theta_{\overline{F}_1,6,10}$ and $\theta_{\overline{F}_2,6,10}$ via matrices $M_1 = 7 \cdot M$ and

$M_2 = 237 \cdot M$ where

$$M = \begin{bmatrix} 0\,1\,0\,0\,0 \\ 0\,0\,1\,0\,0 \\ 0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1 \\ 1\,0\,0\,0\,0 \end{bmatrix}.$$

These linear transformations give us two Laurent polynomials modulo 11 and 251, respectively

$$\begin{aligned} c_{1,\mathcal{B}_1} &= 2X^9 + 6X^8 + 6X^7 + 2X^6 \in \mathbb{Z}_{11}[X^{\pm 1}], \\ c_{2,\mathcal{B}_1} &= 92X^9 + 25X^8 + 25X^7 + 92X^6 \in \mathbb{Z}_{251}[X^{\pm 1}]. \end{aligned}$$

Using the coefficient-wise CRT and lifting coefficients in $A_{\mathcal{B}_1}$ we recover the Laurent polynomial

$$a_{\mathcal{B}_1} = 343X^9 + 1029X^8 + 1029X^7 + 343X^6 \in \mathbb{Z}[X^{\pm 1}],$$

which is equal to u^3.

We repeat the same steps for the block $\mathcal{B}_2$ with the exponent range $[3, 15]$ and the same coefficient set A. This block has again the polynomials $\overline{F}_{i,\mathcal{B}_2}$ of the same degree and thus every $m_i = 17$ and $\ell_i = 3$. Executing Algorithm 2 we get the following sequence of calculations

$$\begin{aligned} &\overline{c}_{1,\mathcal{B}_2} = 2X^{11} + X^{10} + 10X^7 + 8X^5 + 2X^3 + 9, \\ &\overline{c}_{2,\mathcal{B}_2} = 89X^{11} + 170X^{10} + 172X^7 + 203X^5 + 92X^3 + 111, \\ &\downarrow \\ &c_{1,\mathcal{B}_2} = 6X^{15} + 2X^{11} + 10X^7 + 2X^3, \\ &c_{2,\mathcal{B}_2} = 10X^{15} + 89X^{11} + 172X^7 + 92X^3, \\ &\downarrow \\ &a_{\mathcal{B}_2} = 512X^{15} + 1344X^{11} + 1176X^7 + 343X^3. \end{aligned}$$

The last polynomial is exactly v^3 so we correctly cubed two Laurent polynomials.

Remark 3. The CRT factorization can also be exploited when a homomorphic algorithm needs a bigger plaintext modulus than the upper bound $t_{\mathcal{C}}^{\max}$ discussed above. Let us denote this modulus with a capital T to emphasize direct incompatibility of this parameter with other SHE parameters, namely, $T > t_{\mathcal{C}}^{\max}$. However, one can find a set of natural numbers $\{T_i \leqslant t_{\mathcal{C}}^{\max}\}$ such that $T \leqslant T' = \prod_i T_i$. Then $R_{T'}$ splits into smaller quotient rings R_{T_i}. A plaintext $a \in R_{T'}$ then maps to a vector whose ith component lies in R_{T_i}. In that case the plaintext space splits into quotient rings with smaller moduli via CRT such that each ring fits the SHE settings according to the following diagram

$$R_{T'} \xrightarrow{\mathsf{CRT}} \begin{cases} R_{T_1} \xrightarrow{\mathsf{CRT}} \prod_{t'|T_1} \prod_{f'|\overline{f} \bmod t'} R_{t',f'} \xrightarrow{\text{Alg 1}} R_{T_1} \\ \dots \\ R_{T_s} \xrightarrow{\mathsf{CRT}} \prod_{t'|T_s} \prod_{f'|\overline{f} \bmod t'} R_{t',f'} \xrightarrow{\text{Alg 1}} R_{T_s} \end{cases}$$

A homomorphic circuit evaluation must then be repeated over each CRT factor T_i. Nevertheless, this gives some freedom of choice for T_i's so as to find $R_{T'}$ with a nice CRT decomposition.

6 Parameter choice

In this section we discuss a set of tools that will allow implementers to benefit from our enhanced SIMD approach as much as possible. There are three parameters that directly affect the packing capacity. We list them below in an order that seems natural for solving any packing problem. Nevertheless, all parameters depend on each other.

Plaintext modulus. Earlier we defined the range $[t_{\mathcal{C}}^{\min}, t_{\mathcal{C}}^{\max}]$ from which the plaintext modulus t is allowed to be chosen. Additionally, at the end of Section 4 we discussed the CRT trick that allows to handle plaintext moduli that are bigger than $t_{\mathcal{C}}^{\max}$. Altogether this gives a designer some freedom to choose t such that it splits into many 'advantageous' t_i's. An 'advantageous' t_i means that the factorization of $\overline{f}_i$ is such that the resulting CRT decomposition can embed as many plaintexts as possible, which is usually facilitated by a finer brick structure as in Figure 8.

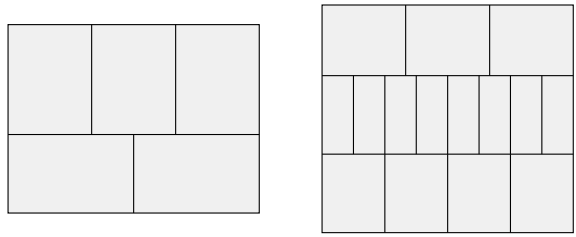


Fig. 8. The CRT decompositions of plaintext spaces corresponding to different t's.

This brick structure is defined by the t_i's and by the degrees of the $\overline{f}_{ij}$'s, namely $d_{i1}, \ldots, d_{ir_i}$ which constitute *a decomposition type* of f modulo t_i. Let G be the Galois group of the splitting field of f over $\mathbb{Q}$. It can be considered as a subgroup of the group S_d of permutations of d elements. Every automorphism σ can be represented as a product of cycle permutations with a corresponding pattern of cycle lengths. Additionally, we say that a set P of prime numbers has density δ if

$$\lim_{x \to \infty} \frac{|\{p \leqslant x : p \in P\}|}{|\{p \leqslant x : p \text{ prime}\}|} = \delta.$$

Then the probability that a desired decomposition type occurs for some random t_i is estimated by the following classical theorem.

Theorem 2 (Frobenius). *The density of the set P of primes modulo which f has a given decomposition type $d_1, d_2, \ldots, d_r$ exists, and it is equal to $1/|G|$ times the number of automorphisms $\sigma \in G$ with cycle pattern $d_1, d_2, \ldots, d_r$.*

An interesting case is where $\overline{f}_i$ splits into linear factors since it gives maximal flexibility to combine blocks. There exists only one $\sigma \in G$ corresponding to such a decomposition which is the identity permutation, so the corresponding probability is $1/|G|$.

Example 6. If $f(X)$ is the nth cyclotomic polynomial then its Galois group G has $d = \phi(n)$ elements and it always splits into irreducible factors of the same degree, i.e. its decomposition type modulo t_i is always $(d', \ldots, d')$ where d' is the order of t_i modulo n; here we implicitly assume that $\gcd(t_i, n) = 1$. Let us take $f(X) = X^{2^k} + 1$. Its Galois group is isomorphic to $\mathbb{Z}^{\times}_{2^{k+1}}$ or to the direct product of two cyclic groups $C_2 \times C_{2^{k-1}}$. It contains 2^k elements with orders shown in the following table:

ord	1	2	4	...	2^{k-1}
$\#\{a \in \mathbb{Z}^{\times}_{2^{k+1}}\}$	1	3	4	...	2^{k-1}

This implies that f splits into $2^{k'}$ irreducible factors of degree $2^{k-k'}$ modulo a random t_i with probability $2^{-k'}$, for any $k' \in \{1, \ldots, k-2, k-1\}$.

In the classical example of a homomorphic application a client encrypts his data and sends it to a third party to perform calculations. Since encryption and decryption are done only on the client side, he therefore has the possibility to tweak the plaintext modulus without re-generation of keys as long as the evaluation (or linearization) key does not depend on t. It is important to note that the plaintext modulus does not affect the security level of an SHE scheme but it does affect the decryption correctness. Hence, t should fit the upper bound $t_{\mathcal{C}}^{\max}$ introduced by the noise growth inside ciphertexts. As a result, one can exploit the same technique as above to find R_t with the most useful decomposition.

Block set. Recall that the plaintext space can be thought of as a set of bricks $\mathcal{P}$. Every block is then a subset of $\mathcal{P}$. The packing problem consists in finding a partition of $\mathcal{P}$ with the maximal number of blocks where each one satisfies Theorem 1. It is clear that the partition search is highly dependent on the data values and the arithmetic operations being performed homomorphically. Therefore the same plaintext space can be used differently for various applications as shown in Figure 9. If $r = \sum_{i=1}^{s} r_i$ is the cardinality of $\mathcal{P}$ then the total number of partitions is equal to the r-th Bell number B_r. That number grows exponentially (see [17]) while r is increasing according to

$$\frac{\ln B_r}{r} \simeq \ln r.$$

As a result a system designer has a lot of flexibility to play with the plaintext space partitions to fit data into some block structure. Obviously, the maximal

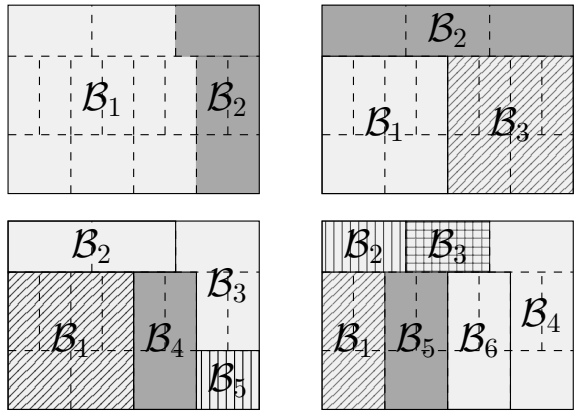


Fig. 9. Different partitions of $\mathcal{P}$.

number of blocks cannot be bigger than r, in which case the blocks are just the singletons $\{R_{i,j}\}$. A plaintext space with many CRT factors is usually easier to handle because it is more flexible for block constructions.

If one does not find a satisfying partition of all of $\mathcal{P}$, it is of course also possible to leave a couple of bricks unused by packing zeros in them (or even random values).

Encoding base. Representing data using Laurent polynomials requires a numerical base b which can be a real or a complex number. The size of b affects the length of a representation as well as the size of its coefficients.

In [3] it was shown that non-integral bases taken from the interval $(1, 2)$ have a simple greedy algorithm that, given a real number, produces a base-b expansion with a ternary set of coefficients. This procedure has the property that smaller bases lead to sparser representations and thus smaller coefficient growth but longer expansions. To illustrate this we resort again to the box representation of a Laurent polynomial.

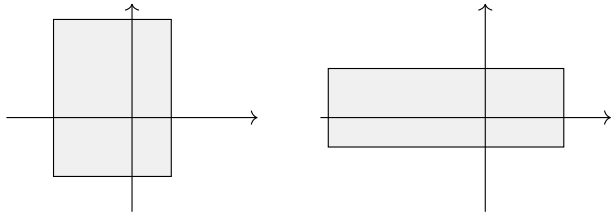


Fig. 10. The examples of bounding boxes corresponding to different encoding bases.

As a result, by changing the encoding base one could play a trade-off game between degree and coefficient size such that the number of plaintexts fitting a block structure is maximal. Furthermore, each block allows to encode data in a different base because neither Algorithm 1 nor Algorithm 2 depends on the choice of b.

Example 7. To illustrate the aforementioned techniques we revisit a medical application of the YASHE homomorphic encryption scheme [4] given in [5]. In this paper the standard logistic function is homomorphically computed to predict the probability of having a heart attack. The algorithm is divided into two steps.
Step 1. One computes the following weighted sum of six encrypted predictive variables

$$z = 0.072 \cdot z_1 + 0.013 \cdot z_2 - 0.029 \cdot z_3 + 0.008 \cdot z_4 - 0.053 \cdot z_5 + 0.021 \cdot z_6,$$

where each $z_i \in [0, 400]$. The multiplicative depth of the corresponding circuit is 1. For this step we take the same YASHE parameters as in [5], i.e. $q \simeq 2^{128}$ and $f(X) = X^{4096} + 1$. Given these parameters we derive $t^{\mathcal{C}}_{\max} = 2097152 \simeq 2^{21}$ using [4, Lem. 9]. Running over all primes less than $t^{\mathcal{C}}_{\max}$ we find that modulo $t_1 = 257$ and modulo $t_2 = 3583$ our polynomial $f(X)$ can be written as a product of 128 coprime factors of degree 32. With $t = t_1$ the conventional SIMD technique allows then to pack at most 128 values into one plaintext. This capacity can be achieved with base-3 balanced ternary expansions that result in an output bounding box of size $(29, \log_2 53)$. However, our approach supports $t = t_1 \cdot t_2$ so one can pack 256 values using the same encoding method.
Step 2. The output of Step 1 is decrypted, decoded to a real number and encoded again to a plaintext. This 'refreshed' encoding is then encrypted and given as input to the following approximation of the logistic function

$$P(x) = \frac{1}{2} + \frac{1}{4}x - \frac{1}{48}x^3 + \frac{1}{480}x^5 - \frac{17}{80640}x^7.$$

In this step the multiplicative depth is 3, $q \simeq 2^{512}$ and $f(X) = X^{16384} + 1$. These parameters lead to $t^{\mathcal{C}}_{\max} \simeq 2^{50}$. Using the previous SIMD technique the maximal plaintext capacity can be achieved with the plaintext modulus $t \simeq 2^{30.54}$ and base-3 balanced ternary encoding. In this case $f(X)$ splits into 8192 quadratic factors and the output bounding box is of size $(229, 29.54)$. We can thus compose 71 blocks with 115 slots and one block with the remaining slots. As a result, one plaintext can contain at most 71 values.

This capacity can be increased with our SIM^{2}D technique. In particular, one can notice that the ratio between $t^{\mathcal{C}}_{\max}$ and the previously mentioned modulus t is around $2^{19.46}$, which implies some part of the plaintext space remains unfilled. We can fill that space setting the plaintext modulus to $t_1 \cdot t_2$ with $t_1 \simeq 2^{30.54}$ and $t_2 = 675071 \simeq 2^{19.36}$. The polynomial $f(X)$ splits into 128 factors of degree 128 modulo t_2. To fit the modulus t_2 we encoded real values with the non-integral base $b = 1.16391$ and obtained the output bounding box $(1684, 19.36)$. Therefore one block should consist of 14 slots, and we can construct 9 such blocks. As a result, we can combine these blocks with the 71 blocks given by the old SIMD technique, which results in a total plaintext capacity of 80 values.

7 Conclusion

In this paper we presented two techniques that make SIMD operations in the setting of homomorphic encryption more flexible and efficient. Our first technique showed how data values that are naturally represented as Laurent polynomials can be encoded into a plaintext space of the form $\mathbb{Z}_t[X]/(\overline{f}(X))$. Furthermore, we also provided sufficient conditions for correct decoding after evaluation of an arithmetic circuit. Our second technique relied on a fine-grained CRT decomposition of the plaintext space resulting in a much denser and thus more efficient data packing compared to the state of the art. Finally, we provided guidelines on how to choose system parameters in order to find the most efficient packing strategy for a particular task.

References

1. Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.
2. Fabrice Benhamouda, Tancrède Lepoint, Claire Mathieu, and Hang Zhou. Optimization of bootstrapping in circuits. In Philip N. Klein, editor, *28th SODA*, pages 2423–2433. ACM-SIAM, 2017.
3. Charlotte Bonte, Carl Bootland, Joppe W. Bos, Wouter Castryck, Ilia Iliashenko, and Frederik Vercauteren. Faster homomorphic function evaluation using non-integral base encoding. In Wieland Fischer and Naofumi Homma, editors, *CHES 2017*, volume 10529 of *LNCS*, pages 579–600. Springer, Heidelberg, 2017.
4. Joppe W. Bos, Kristin Lauter, Jake Loftus, and Michael Naehrig. Improved security for a ring-based fully homomorphic encryption scheme. In Martijn Stam, editor, *14th IMA International Conference on Cryptography and Coding*, volume 8308 of *LNCS*, pages 45–64. Springer, Heidelberg, 2013.
5. Joppe W. Bos, Kristin E. Lauter, and Michael Naehrig. Private predictive analysis on encrypted medical data. *Journal of Biomedical Informatics*, 50:234–243, 2014.
6. Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 868–886. Springer, Heidelberg, 2012.
7. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, *ITCS 2012*, pages 309–325. ACM, 2012.
8. Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd FOCS*, pages 97–106. IEEE Computer Society Press, 2011.
9. Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 505–524. Springer, Heidelberg, 2011.
10. Hao Chen, Kim Laine, Rachel Player, and Yuhou Xia. High-precision arithmetic in homomorphic encryption. In Nigel P. Smart, editor, *CT-RSA 2018*, volume 10808 of *LNCS*. Springer, Heidelberg, 2018. To appear.
11. Jung Hee Cheon, Kyoohyung Han, and Duhyeong Kim. Faster bootstrapping of FHE over the integers. Cryptology ePrint Archive, Report 2017/079, 2017. `http://eprint.iacr.org/2017/079`.

12. Jung Hee Cheon, Jinhyuck Jeong, Joohee Lee, and Keewoo Lee. Privacy-preserving computations of predictive medical models with minimax approximation and non-adjacent form. In Michael Brenner, Kurt Rohloff, Joseph Bonneau, Andrew Miller, Peter Y. A. Ryan, Vanessa Teague, Andrea Bracciali, Massimiliano Sala, Federico Pintore, and Markus Jakobsson, editors, *FC 2017*, volume 10323, pages 53–74. Springer, Heidelberg, 2017.
13. Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 3–33. Springer, Heidelberg, 2016.
14. Anamaria Costache, Nigel P. Smart, and Srinivas Vivek. Faster homomorphic evaluation of discrete Fourier transforms. In Aggelos Kiayias, editor, *FC 2017*, volume 10322 of *LNCS*, pages 517–529, 2017.
15. Anamaria Costache, Nigel P. Smart, Srinivas Vivek, and Adrian Waller. Fixed-point arithmetic in SHE schemes. In Roberto Avanzi and Howard M. Heys, editors, *SAC 2016*, volume 10532 of *LNCS*, pages 401–422. Springer, Heidelberg, 2016.
16. CryptoExperts. FV-NFLlib. `https://github.com/CryptoExperts/FV-NFLlib`, 2016.
17. Nicolaas Govert De Bruijn. *Asymptotic methods in analysis*. Dover, New York, NY, 1958.
18. Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristin E. Lauter, Michael Naehrig, and John Wernsing. Manual for using homomorphic encryption for bioinformatics. *Proceedings of the IEEE*, 105(3):552–567, 2017.
19. Léo Ducas and Daniele Micciancio. FHEW: Bootstrapping homomorphic encryption in less than a second. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 617–640. Springer, Heidelberg, 2015.
20. Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144, 2012. `http://eprint.iacr.org/2012/144`.
21. Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, 2009.
22. Craig Gentry, Shai Halevi, and Nigel P. Smart. Fully homomorphic encryption with polylog overhead. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 465–482. Springer, Heidelberg, 2012.
23. Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 75–92. Springer, Heidelberg, 2013.
24. Zhicong Huang Amir Jalali Hao Chen, Kyoohyung Han and Kim Laine. Simple encrypted arithmetic library — SEAL (v2.3). Technical report, Technical report, Microsoft Research, 2017.
25. Michael Naehrig, Kristin E. Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In Christian Cachin and Thomas Ristenpart, editors, *Proceedings of the 3rd ACM Cloud Computing Security Workshop, CCSW 2011*, pages 113–124. ACM, 2011.
26. Ronald L. Rivest, Len Adleman, and Michael L. Dertouzos. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.
27. Nigel P. Smart and Frederik Vercauteren. Fully homomorphic SIMD operations. *Des. Codes Cryptography*, 71(1):57–81, 2014.

Chapter 11

Efficiently Processing Complex-Valued Data in Homomorphic Encryption.

Publication data

Bootland, C., Castryck, W., Iliashenko, I., and Vercauteren, F. Efficiently processing complex-valued data in homomorphic encryption. *Journal of Mathematical Cryptology* (2019), to be published.

Efficiently Processing Complex-Valued Data in Homomorphic Encryption

Carl Bootland[1], Wouter Castryck[1,2], Ilia Iliashenko[1], and Frederik Vercauteren[1]

[1]imec-COSIC, Dept. Electrical Engineering, KU Leuven
firstname.lastname@esat.kuleuven.be
[2]Department of Mathematics, KU Leuven

Abstract. We introduce a new homomorphic encryption scheme that is natively capable of computing with complex numbers. This is done by generalizing recent work of Chen, Laine, Player and Xia, who modified the Fan–Vercauteren scheme by replacing the integral plaintext modulus t by a linear polynomial $X - b$. Our generalization studies plaintext moduli of the form $X^m + b$. Our construction significantly reduces the noise growth in comparison to the original FV scheme, so much deeper arithmetic circuits can be homomorphically executed.

1 Introduction

The goal of homomorphic encryption is to allow for arbitrary arithmetic operations on encrypted data, such that the decrypted result equals the outcome of the same calculation carried out in the clear. Since the publication of Gentry's seminal Ph.D. work [15], this research area has evolved rapidly and is on the verge of reaching a first degree of maturity, as was recently demonstrated e.g. by practical implementations of privacy-enhanced electricity load forecasting [3, 2], digital image processing [1, 10], and medical data management [12, 18, 7]. Most of the current focus lies on *somewhat* homomorphic encryption (SHE), where the schemes are capable of homomorphically evaluating an arithmetic circuit having a certain predetermined computational depth. The leading proposals for realizing this goal are the Brakerski-Gentry-Vaikunthanathan (BGV) scheme [4] and the Fan-Vercauteren (FV) scheme [13].

In actual applications, the input to the homomorphic evaluation of an arithmetic circuit $\mathcal{C}$ needs to be preprocessed in two steps. The first step is encoding, where one's task is to represent the actual 'real world data' as elements of the plaintext space of the envisaged SHE scheme. This plaintext space is a certain commutative ring, and the encoding should be such that real world arithmetic

The first author is supported by a PhD fellowship of the Research Foundation - Flanders (FWO). The third author has been supported in part by ERC Advanced Grant ERC-2015-AdG-IMPaCT.

agrees with the corresponding ring operations, up to the anticipated computational depth.

In the original descriptions of BGV and FV, the plaintext space is a ring of the form $R_t = \mathbb{Z}[X]/(t, f(X))$ where $t \geq 2$ is an integer and $f(X) \in \mathbb{Z}[X]$ is a monic irreducible polynomial. Throughout this paper we will stick to the common choice of 2-power cyclotomics $f(X) = X^n + 1$, where $n = 2^k$ for some integer $k \geq 1$. Encoding numerical input is typically done by taking an integer-digit expansion with respect to some base b, then replacing b by X and finally reducing the digits modulo t. Decoding then amounts to lifting the coefficients back to $\mathbb{Z}$, for instance by choosing representatives in $(-t/2, t/2]$, and evaluating the result at $X = b$. Thanks to the relation $X^{-1} \equiv -X^{n-1}$ it is possible to allow the expansions to have a fractional part. In this case the decoding step must be preceded by replacing the monomials X^i of degree $i > B$ by $-X^{i-n}$, for some appropriate point of separation B. All these parameters need to be chosen in such a way that the evaluation of $\mathcal{C}$ on the encoded data decodes to the right outcome. At the same time one wants t to be as small as possible, because its size highly affects the efficiency of the resulting SHE computation. Selecting optimal parameters is a tedious application-dependent balancing act to which a large amount of recent literature has been devoted, see e.g. [20, 12, 8, 6, 18, 11, 2].

Because in practice n is of size at least 1024, the plaintext spaces R_t can a priori host an enormous range of data, even for very small values of t. Unfortunately this is hindered by their structure, which is not a great match with numerical input data types like integers, rationals or floats. For example, if $t = 2$ then it is not even possible to add a non-zero element to itself without incorrect decoding. Because of such phenomena, values of t are required that typically consist of dozens of decimal digits, badly affecting the efficiency. An idea to remedy this situation has been around for a while [17, 4, 14] and uses a polynomial plaintext modulus, rather than just an integer. Recently the first detailed instantiation of this idea was given by Chen, Laine, Player and Xia [6], who adapted the FV scheme to plaintext moduli $t = X - b$ for some $b \in \mathbb{Z}_{\geq 2}$. In this case the plaintext space becomes $R_t = \mathbb{Z}[X]/(X - b, X^n + 1) = \mathbb{Z}[X]/(X - b, b^n + 1) \cong \mathbb{Z}_{b^n+1}$, whose structure is a *much* better match with the common numerical input data types. This allows for much smaller plaintext moduli (norm-wise), with beneficial consequences for the efficiency, or for the depth of the circuits $\mathcal{C}$ that can be handled [6, Section 7.2].

This paper further explores the paradigm that the structure of the plaintext space R_t should match the input data type as closely as possible. Concretely, we focus on *complex-valued* data types, such as cyclotomic integers and floating point complex numbers. We study this setting mainly in its own right, but note that complex input data has been considered in homomorphic encryption before, e.g., in the homomorphic evaluation of the Discrete Fourier Transform studied by Costache, Smart and Vivek [10] in the context of digital image processing, where the input consists of cyclotomic integers.

Representing complex numbers. One naive way to encode a complex number z would be to view it as a pair of real numbers, for instance using Cartesian or polar coordinates. These can be fed separately to the SHE scheme, which is now used to evaluate two circuits. A more direct way is to use a complex base b. For instance, one could take $b = e^{\pi \mathbf{i}/n}$, as was done by Cheon, Kim, Kim and Song [8], albeit in a somewhat different context. This choice has the additional feature that $f(b) = 0$, so that wrapping around modulo $f(X) = X^n + 1$ does not lead to incorrect decoding. However, finding an integer-digit base b expansion with small norm which approximates z sufficiently well is an n-dimensional lattice problem, which is practically infeasible. To get around this Costache, Smart and Vivek [10] instead use $b = \zeta := e^{\pi \mathbf{i}/m}$ for some divisor $m \mid n$, which is small enough for finding short base ζ approximations, while preserving the feature that wrapping around modulo $f(X)$ is unharmful. But in their approach, a huge portion of plaintext space is left *unused*. Indeed, the encoding map is

$$\mathbb{Z}[\zeta] \to R_t : z = \sum_{i=0}^{m-1} z_i b^i \mapsto \sum_{i=0}^{m-1} \overline{z}_i Y^i,$$

where $Y = X^{n/m}$, $t \geq 2$ is an integral plaintext modulus and $\overline{z}_i$ is the reduction of z_i mod t, so that all plaintext computations are carried out in the subring $\mathbb{Z}[Y]/(t, Y^m + 1)$, which is of index t^{n-m} in R_t. Our proposal is to resort to a plaintext modulus of the form $t = X^m + b$ for some small integer b, with $|b| \geq 2$. In this case, for $m < n$, we have $R_{X^m+b} = \mathbb{Z}[X]/(X^m + b, X^n + 1) = \mathbb{Z}[X]/(b^{n/m} + 1, X^m + b)$. An additional assumption (which is discussed in more detail in the next section), is that

$$\text{there exists an } \overline{\alpha} \in \mathbb{Z}_{b^{n/m}+1} \text{ such that } \overline{b} = \overline{\alpha}^m, \tag{1}$$

where $\overline{b}$ denotes the reduction of b modulo $b^{n/m} + 1$. Throughout we fix such an $\overline{\alpha}$ and let $\overline{\beta}$ be its multiplicative inverse, which necessarily exists. This implies that $(\overline{\beta} X)^m + 1 = 0$, therefore we have a well-defined ring homomorphism

$$\mathbb{Z}[\zeta] \to R_{X^m+b} : \sum_{i=0}^{m-1} z_i \zeta^i \mapsto \sum_{i=0}^{m-1} \overline{z}_i \overline{\beta}^i X^i \tag{2}$$

which is surjective with kernel $(b^{n/m} + 1)$. In other words, while Costache, Smart and Vivek restrict their computations to an injective copy of $\mathbb{Z}[\zeta]/(t)$ inside R_t, we can view R_{X^m+b} as an *isomorphic* copy of $\mathbb{Z}[\zeta]/(b^{n/m} + 1)$. Essentially, our approach transfers the unused part of the plaintext space coming from the large dimension n into a larger integral modulus, reflected in the exponent n/m.

In the remainder of this paper, we explain how this observation can be used to efficiently process complex-valued input data in homomorphic encryption. First, in Section 2 we explain how to encode and decode elements of the ring $\mathbb{Z}[\zeta]$ of $2m^{\text{th}}$ cyclotomic integers and discuss the assumption (1), with special attention to the case $m = 2$ where $\mathbb{Z}[\zeta] = \mathbb{Z}[\mathbf{i}]$ is the ring of Gaussian integers. Next in Section 3 we explain how this can be used to encode other data types such as

cyclotomic rationals or complex floats, either by resorting to LLL as in [10] or by using Chen et al.'s fractional encoder from [6]. In Section 4 we discuss how to adapt the FV scheme so that it can cope with plaintext spaces of the form R_{X^m+b}. Finally, in Section 6 we discuss the performance of this adaptation in comparison with previous approaches. In short we can reach a depth at least 5 times that of the best approach which directly encrypts encodings of complex numbers [10]. We can also reach very similar depths to the state of the art where one encrypts the real and imaginary parts separately [6]. However, since we natively encrypt complex numbers our ciphertexts are two times smaller and hence our approach is more efficient by roughly a factor two in time and three in space.

2 Encoding and decoding elements of $\mathbb{Z}[\zeta]$

Encoding Encoding an element of $\mathbb{Z}[\zeta]$ happens in two steps. The first step applies the map (2) yielding a polynomial of degree less than m which typically has very large coefficients. The second step is comparable to the *hat encoder* of Chen et al. [6] and switches to another representant by spreading this polynomial across the range $1, X, \ldots, X^{n-1}$ while making the coefficients a lot smaller. The result will then be lifted to $R = \mathbb{Z}[X]/(X^n + 1)$ and fed to our adaptation of the FV scheme, where the smaller coefficients are important to keep the noise growth bounded.

Here is how this second step is carried out in practice: we think of the coefficients $\overline{z}_i\overline{\beta}^i$ as being represented by integers between $-\lfloor b^{n/m}/2 \rfloor$ and $\lceil b^{n/m}/2 \rceil$. We then expand these integers to base b using digits $a_{i,j}$ from the range $-\lfloor b/2 \rfloor$, $\ldots$, $\lfloor b/2 \rfloor$ to find

$$\overline{z}_i\overline{\beta}^i = \overline{a}_{i,n/m-1}\overline{b}^{n/m-1} + \ldots + \overline{a}_{i,1}\overline{b} + \overline{a}_{i,0}.$$

There is a minor caveat here, namely if b is odd then there are more integers modulo $b^{n/m} + 1$ than there are balanced b-ary expansions of length at most n/m. This is easily resolved by allowing the last digit to be one larger. For even b the situation is opposite: since $\overline{z}_i\overline{\beta}^i$ is represented by an integer of size at most $b^{n/m}/2 = b/2 \cdot b^{n/m-1}$ we have a surplus of base-b expansions. Here it makes sense to choose an expansion with the shortest Hamming weight (e.g., if $b = 2$ then we simply pick the non-adjacent form). We denote the maximal number of non-zero coefficients that can appear in a fresh encoding by N_b.

Given such base-b expansions of the coefficients, we replace each occurrence of $\overline{b}$ by $-X^m$ and then substitute the results in the image of (2). We end up with an expansion $\sum_{i=0}^{n-1} \overline{c}_i X^i$ where the $\overline{c}_i$ are represented by integers of absolute value at most $\lfloor b/2 \rfloor$, or in fact $\lfloor (b+1)/2 \rfloor$ if we take into account the caveat.

Decoding In order to decode a given expansion $\sum_{i=0}^{n-1} \overline{c}_i X^i$ we walk through the same steps in reverse order. First we pick another representant by reducing

the expansion modulo $X^m + b$, in order to end up with

$$\sum_{i=0}^{m-1} \overline{c}'_i X^i \in \mathbb{Z}[X]/(b^{n/m}+1, X^m + b).$$

This can be rewritten as $\sum_{i=0}^{m-1} \overline{c}'_i \overline{\alpha}^i \overline{\beta}^i X^i$ so we decode as $\sum_{i=0}^{m-1} z_i \zeta^i \in \mathbb{Z}[\zeta]$ where z_i is a representant of $\overline{c}'_i \overline{\alpha}^i$ taken from the range $-\lfloor b^{n/m}/2 \rfloor, \ldots, \lceil b^{n/m}/2 \rceil$.

On the assumption (1) Usually n and m are determined by security considerations and the concrete application. To apply our encoding method we want to find a small value of b for which condition (1) is met. This is easiest if n/m is small or m is small. If no satisfactory value of b can be found then one can try to enlarge m and view $\mathbb{Z}[\zeta]$ as a subring of a higher degree cyclotomic ring. Below we give two lemmas constraining the possible choices for b given m and n; still assuming we are working with 2-power cyclotomic f.

One choice for b which is always possible is $2^{m/2}$, since defining α as

$$\alpha = 2^{n/8}\left(2^{n/4} - 1\right), \tag{3}$$

then it easy to verify that $\alpha^2 \equiv 2 \bmod 2^{n/2} + 1$ and hence

$$\alpha^m \equiv 2^{m/2} \bmod 2^{\frac{m}{2}\frac{n}{m}} + 1.$$

If m is small then this results in a reasonably slow coefficient growth. On the other hand if m is large compared to n then the modulus $b^{n/m} + 1$ is smaller and it is apparently easier to have condition (1) satisfied, as is confirmed by experiment.

Lemma 1. *Let $n > m > 1$. A necessary condition for* (1) *is that for every odd prime $p \mid b^{n/m} + 1$ we have $2n \mid p - 1$.*

Proof. First we show that b has multiplicative order $2n/m$ in $\mathbb{Z}_{b^{n/m}+1}$. Clearly we have $b^{n/m} \equiv -1 \mod b^{n/m}+1$ so that $b^{2n/m} \equiv 1 \mod b^{n/m}+1$. This shows that the order of b divides $2n/m$ so is a power of 2 and hence it is equal to $2n/m$.

Since $2 \mid n/m$ and $x^2 \equiv 1 \mod 4$ for any odd x we have that if b is odd $b^{n/m} + 1 \equiv 2 \mod 4$ while if b is even $b^{n/m} + 1$ is odd. Thus that we can write

$$b^{n/m} + 1 = 2^\rho p_1^{e_1} \ldots p_j^{e_j}$$

where the p_i, $1 \le i \le j$ are distinct odd primes and $\rho = b \bmod 2$.

Now we can see via the Chinese Remainder Theorem that there exists an α such that $\alpha^m \equiv b \mod b^{n/m} + 1$ if and only if there exist α_i such that $\alpha_i^m \equiv b \bmod p_i^{e_i}$ for every i. Further we must have $b^{n/m} \equiv -1 \mod p_i^{e_i}$ so that b has order $2n/m$ modulo $p_i^{e_i}$. This implies α_i has order $m \cdot 2n/m = 2n$ modulo $p_i^{e_i}$ and since $(\mathbb{Z}/p_i^{e_i}\mathbb{Z})^\times$ is cyclic of order $p_i^{e_i - 1}(p_i - 1)$ we see that $2n \mid (p_i - 1)$ by Lagrange's Theorem for each $1 \le i \le j$.

Lemma 2. *Let g be an element of order n in $\mathbb{Z}_{4n}^{\times}$ and let t be an element of order 2 not in $\langle g \rangle$ so that $\mathbb{Z}_{4n}^{\times} = \langle t \rangle \times \langle g \rangle$. If condition (1) is satisfied for odd $b > 1$ and $m > 1$ then $b \bmod 4n$ is an element of the subgroup $\langle t \rangle \times \langle g^m \rangle$. In particular this implies that $b \equiv \pm 1 \bmod 4m$.*

In fact, one may always take $g = 3$ and $t = -1$ in the above lemma.

Proof. Using Lemma 1 and the notation from its proof we can write each p_i as $2nc_i + 1$ for some natural number c_i. This implies that

$$b^{n/m} + 1 = 2 \prod_{i=1}^{j} (2nc_i + 1)^{e_i} \equiv 2 \bmod 4n$$

and hence $b^{n/m} \equiv 1 \bmod 4n$. Therefore the order of b as an element of $\mathbb{Z}_{4n}^{\times}$ divides n/m.

Now we have $\mathbb{Z}_{4n}^{\times} = \langle t \rangle \times \langle g \rangle$ so that for $b \bmod 4n$ to have an order dividing n/m it must be an element of the subgroup $\langle t \rangle \times \langle g^m \rangle$. This is because this subgroup certainly only contains elements whose order divides n/m. Further, $\mathbb{Z}_{4n}^{\times}$ has exactly $2n/m$ such elements but this is the size of the subgroup so the subgroup is exactly all such elements.

For the final part we note, as stated after the lemma, that $g = 3$ and $t = -1$ can be taken and that $3^m \equiv 1 \bmod 4m$ which gives the desired result. We remark that for any $b \equiv \pm 1 \mod 4m$ it is always the case that $b^{n/m} \equiv 1 \bmod 4n$ so from this condition we cannot determine anything more about b modulo $4m$ but the condition given modulo $4n$ is stronger.

Lemma 3. *Suppose b, n and m satisfy (1), then so does $-b, n, m$.*

Proof. Since $(-b)^{n/m} + 1 = b^{n/m} + 1$ when n is a power of two and $m < n$, we must show that -1 has an mth root modulo $b^{n/m} + 1$; we show that $\alpha^{n/m}$ is such an mth root. We have $(\alpha^{n/m})^m = (\alpha^m)^{n/m} \equiv b^{n/m} \equiv -1 \bmod b^{n/m} + 1$ as required. Hence we see that $(\alpha^{n/m+1})^m = \alpha^n \alpha^m = -1 \cdot b$ as required.

We note that the above proof only required n/m to be even and not equal to a power of two so applies somewhat more generally.

We give some examples of both odd and even b which satisfy Equation (1) in Appendix A. However it seems to be more fruitful to consider the case of even b.

Our method is particularly friendly towards Gaussian integers. Indeed if $m = 2$ then one can always take $b = 2$, as we have seen that $\overline{\alpha}^2 = \overline{2}$ where α is as in (3). The map (2) then defines an isomorphism between R_{X^2+2} and $\mathbb{Z}[\mathbf{i}]/(2^{n/2} + 1)$. If this ring is not large enough to ensure correct decoding, then one can move to slightly larger values of b. The next choice which always works is $b = 4$, where one can simply take $\alpha = 2$. Here the ring becomes $\mathbb{Z}[\mathbf{i}]/(2^n + 1)$.

3 Encoding complex-valued input data

In this section we look at the more general problem of encoding floating point complex numbers. Our approach will be to approximate these complex numbers by suitable cyclotomic rationals and then proceed as in Section 2. We have many choices for such approximations including the choice of m which defines which root of unity we are working with. We also have the choice between using integer or rational coefficients for the approximation. Perhaps the most obvious and straightforward approach is to consider our complex number z written in terms of its real and imaginary parts, say $z = x + y\mathbf{i}$ for some real numbers x and y. We can then approximate x and y by rationals depending on how much precision we require. This leads us to considering the case $m = 2$ and the question then arises of how to encode fractional coefficients.

3.1 Fractional encoding

Here we consider how to encode a rational number into the space $\mathbb{Z}/p\mathbb{Z}$ for some integer p, so that it can then be expanded using the technique in Section 2. This problem was considered by Chen, Laine, Player and Xia in [6, Section 6]. Their approach is to define a finite subset $\mathcal{P}$ of $\mathbb{Q}$ along with an encoding map $\texttt{Enc}\colon \mathcal{P} \to \mathbb{Z}/p\mathbb{Z}$ and a decoding map $\texttt{Dec}\colon \texttt{Enc}(\mathcal{P}) \to \mathcal{P}$. The maps should satisfy, firstly, correctness: $\texttt{Dec}(\texttt{Enc}(x/y)) = x/y$ for $x/y \in \mathcal{P}$ and secondly, $\texttt{Enc}$ should be both additively and multiplicatively homomorphic so long as it still encodes an element of $\mathcal{P}$. The natural choice for the map $\texttt{Enc}$ is $\texttt{Enc}(x/y) = xy^{-1} \bmod p$ where the inverse of y is computed modulo p. Care thus needs to be taken to ensure that y has such an inverse, which is ensured with a careful choice of $\mathcal{P}$.

In our setting the coefficient modulus p is of the form $b^{n/2} + 1$, thus if one wants roughly the same precision for the integer and fractional parts one can take for an odd base b

$$\mathcal{P} = \left\{ c + \frac{d}{b^{n/4}} \ : \ c, d \in \left[-\frac{b^{n/4}-1}{2}, \frac{b^{n/4}-1}{2} \right] \cap \mathbb{Z} \right\};$$

while for even b one can choose

$$\mathcal{P} = \left\{ c + \frac{d}{b^{n/4-\delta}} : |c| \leq \frac{(b^{n/4+\delta-1}-1)b}{2(b-1)}; |d| \leq \frac{(b^{n/4-\delta}-1)b}{2(b-1)}; c, d \in \mathbb{Z} \right\},$$

where $\delta \in \{0, 1\}$ depending on whether you want one more base-b digit in the fractional ($\delta = 0$) or integer ($\delta = 1$) part.

The encoding of an element $e \in \mathcal{P}$ is then computed as $-eb^{n/2} \bmod b^{n/2} + 1$. The important thing to note about using this encoding is that for decoding to work the result of the computations must lie in $\mathcal{P}$. If your input data are complex numbers and you approximate them using $n/4$ fractional b-ary digits then it is likely that after one multiplication the result is no longer in $\mathcal{P}$. Thus one must appropriately choose the precision with which to encode the data, depending

primarily on the depth of the circuit to be evaluated and the final precision required. The only constraint is that the precision should be a divisor of $b^{n/4}$ so that $-eb^{n/2}$ is an integer.

We note that the fractional encoder need not require m to be 2. However in this case there appears to be no straightforward way to find a good rational approximation with small numerators and denominators except when the denominators are all equal, in this case if this denominator is r then we simply require an approximation of rz in $\mathbb{Z}[\zeta]$ subject to some constraint on the coefficients. However, the problem of finding such an approximation to our complex number itself, rather than a scaling, is interesting in its own right as it avoids the need for encoding fractional values and tracking the denominator inherently present in such encodings.

3.2 Integer coefficient approximation

The task of finding a cyclotomic integer closely approximating an arbitrary complex number was considered by Costache, Smart and Vivek in [10]. Here the idea is to solve an instance of the closest vector problem (CVP) in the (scaled) lattice $\mathbb{Z}[\zeta]$, where the power basis is scaled and split into real and complex part, which are approximated by integers. In detail: we choose a scaling constant $C > 0$, and define the constants a_i and b_i for $i = 0, \ldots, m-1$, where $a_i = \lceil \Re(C\zeta^i) \rfloor$ and $b_i = \lceil \Im(C\zeta^i) \rfloor$. The lattice we then consider is given by the m rows of the matrix

$$\begin{pmatrix} 1 & & 0 & a_0 & b_0 \\ & \ddots & & \vdots & \vdots \\ 0 & & 1 & a_{m-1} & b_{m-1} \end{pmatrix}.$$

The target vector in our CVP instance will then be the appropriately scaled real and complex parts of the complex number z we wish to approximate. Concretely, this vector is $(0, \ldots, 0, \lceil \Re(Cz) \rfloor, \lceil \Im(Cz) \rfloor)$.

If $(z_0, \ldots, z_{m-1}, A, B)$ is a solution to the CVP instance then we must have

$$\lceil \Re(Cz) \rfloor \approx A = \sum_{i=0}^{m-1} z_i a_i \approx \Re\left(C \sum_{i=0}^{m-1} z_i \zeta^i \right)$$

and similarly for the imaginary part. We therefore see that $\sum_{i=0}^{m-1} z_i \zeta^i$ is a good approximation to z. Further, C gives some control over the quality of the approximation, larger C gives a finer-grained lattice but also increases the size of the last two coefficients of the basis vectors which may lead to a larger distance between the target vector and the closest lattice point, which in turn makes solving the CVP instance harder and negatively affects the quality of our approximation of Cz.

In [10] the authors solve this CVP instance using the embedding technique. Namely they attempt to solve the shortest vector problem in the lattice spanned

by the rows of

$$\begin{pmatrix} 1 & 0 & a_0 & b_0 & 0 \\ & \ddots & \vdots & \vdots & \vdots \\ 0 & 1 & a_{m-1} & b_{m-1} & 0 \\ 0 & \cdots\ 0 & \lceil \Re(Cz) \rfloor & \lceil \Im(Cz) \rfloor & T \end{pmatrix}$$

for some non-zero constant T. With suitable parameter choices, performing LLL reduction on this lattice will return a basis of short vectors for this lattice, among which at least one has $\pm T$ in the final coordinate. The remaining coefficients then give plus or minus the target vector minus a close vector.

One issue with the embedding technique is that each new instance of the CVP problem requires performing lattice reduction which for large m is rather time-consuming. In typical applications we want to approximate many different complex numbers, using the same C so only the target vector changes. A more efficient approach therefore is to perform lattice reduction on the CVP lattice itself and since this is independent of the target vector it needs only to be done once so we can spend significantly more time in this step to find a good basis of this lattice. We can then apply a technique such as Babai's nearest plane algorithm, or Babai's rounding algorithm, with this reduced basis to find an approximate closest vector.

4 Adapting the Fan-Vercauteren SHE scheme

In this section we construct a variant of the FV scheme [13] with plaintext modulus $X^m + b$ following the blueprint given in [6]. We prove correctness of this scheme and analyze the noise growth induced by homomorphic arithmetic operations.

4.1 Basic scheme

Writing $R = \mathbb{Z}[X]/(X^n + 1)$, the ciphertext space is defined by $R_q = R/(q)$ for some positive integer q, while the plaintext space is $R_{X^m+b} = R/(X^m + b)$. We will assume that $b \ll q$. Recall that in the original FV scheme the plaintext space is $R/(t)$ for some positive integer $t \ll q$. We define the scaling parameter Δ_b as

$$\Delta_b = \left\lfloor \frac{q}{X^m + b} \mod (X^n + 1) \right\rceil = \left\lfloor -\frac{q}{b^{n/m} + 1} \sum_{i=1}^{n/m} (-b)^{i-1} X^{n-im} \right\rceil .$$

Obviously, Δ_b is the analogue of the scalar $\Delta = \lfloor q/t \rfloor$ in the original FV scheme. Other parameters are the error distribution $\chi_e = \mathcal{D}(\sigma^2)$ on R (coefficient-wise with respect to the power basis, with standard deviation σ) and the key distribution $\chi_k = \mathcal{U}_3$ which uniformly generates elements of R with ternary coefficients (with respect to the power basis). We also define the decomposition base w and denote $\ell = \lfloor \log_w q \rfloor$.

The new encryption scheme $\texttt{ComFV}$ is then defined in the same way as $\texttt{FV}$ where t and Δ are replaced by $X^m + b$ and Δ_b, respectively.

- $\texttt{ComFV.KeyGen}(\,)$: Let $s \leftarrow \chi_k$ and $e, e_0, \ldots e_\ell \leftarrow \chi_e$. Uniformly sample random $a, a_0, \ldots, a_\ell \in R_q$ and compute $b_i = \left[-(a_i s + e_i) + w^i s^2\right]_q$. Output the secret key $\texttt{sk} = s$, the public key $\texttt{pk} = \left(\left[-(as+e)\right]_q, a\right)$ and the evaluation key $\texttt{evk} = \{(b_i, a_i)\}_{i=0}^{\ell}$.
- $\texttt{ComFV.Encrypt}(\texttt{pk}, \mathsf{msg})$: Sample $u \leftarrow \chi_k$ and $e_0, e_1 \leftarrow \chi_e$. Set $p_0 = \texttt{pk}[0]$ and $p_1 = \texttt{pk}[1]$, and compute $c_0 = \left[\Delta_b \cdot \mathsf{msg} + p_0 u + e_0\right]_q$ and $c_1 = \left[p_1 u + e_1\right]_q$. Output $\texttt{ct} = (c_0, c_1)$.
- $\texttt{ComFV.Decrypt}(\texttt{sk}, \texttt{ct})$: Return $\mathsf{msg}' = \left\lfloor \frac{X^m+b}{q} \left[c_0 + c_1 s\right]_q \right\rceil \mod (X^m + b)$.

The security of this scheme is based on the same argument as of the original FV scheme. In particular, it is hard to distinguish the public key $\texttt{pk}$ and ciphertext pairs from uniform tuples according to the decision version of the Ring-LWE problem [19]. The evaluation key $\texttt{evk}$ does not leak any information about the secret key as long as a circular security assumption holds [13].

For an element $a \in K := \mathbb{Q}[x]/(f(x))$ the canonical (infinity) norm of a is defined as

$$\|a\|_\infty^{\text{can}} = \left\|\left(a(\zeta), a(\zeta^3), \ldots, a(\zeta^{2n-1})\right)\right\|_\infty .$$

In Appendix A we state some properties of the canonical norm which will be used throughout this section. To verify correctness we use the notion of invariant noise introduced in [6]. The *invariant noise* of a ciphertext $\texttt{ct} = (c_0, c_1)$ encrypting a plaintext $\mathsf{msg} \in R_{X^m+b}$ is an element $v \in K$ with the smallest canonical norm such that

$$\frac{X^m + b}{q} \cdot \left[c_0 + c_1 s\right]_q = \mathsf{msg} + v + g(X^m + b) \tag{4}$$

for some $g \in R$. Then decryption works correctly when $\|v\|_\infty^{\text{can}} < 1/2$ that is supported by the following theorem.

Theorem 1 (Decryption noise). *Let* $\texttt{ct}$ *be an encryption of the plaintext element* $\mathsf{msg} \in R_{X^m+b}$ *such that its invariant noise* v *satisfies* $\|v\|_\infty^{\text{can}} < 1/2$. *Then* $\texttt{ComFV.Decrypt}(\texttt{sk}, \texttt{ct}) = \mathsf{msg}$.

Proof. Computing $\texttt{ComFV.Decrypt}(\texttt{sk}, \texttt{ct})$, we have using the definition of the invariant noise

$$\begin{aligned}
\mathsf{msg}' &= \left\lfloor \frac{X^m + b}{q} \left[\texttt{ct}[0] + \texttt{ct}[1] \cdot s\right]_q \right\rceil \mod (X^m + b) \\
&= \lfloor \mathsf{msg} + v + g(X^m + b) \rceil \mod (X^m + b) \\
&= \mathsf{msg} + \lfloor v \rceil
\end{aligned}$$

for some $g \in R$ and since $\|v\|_\infty \leq \|v\|_\infty^{\text{can}} < 1/2$ we have $\lfloor v \rceil = 0$. Thus $\mathsf{msg}' = \mathsf{msg}$.

To show that v is small enough, we need an upper bound on the initial invariant noise size depending on the scheme parameters. For this purpose, we use the heuristic approach of Gentry et al. [16]. This approach relies on the average distributional analysis, which estimates the expected size of the invariant noise in the canonical embedding norm.

Recall that the Hamming weight of a plaintext $\mathsf{msg} \in R_{X^m+b}$ is bounded by N_b. In addition, $\|\mathsf{msg}\|_\infty \leq b/2$ for even b and $\|\mathsf{msg}\|_\infty \leq (b+1)/2$ for odd b with at most one coefficient reaching this bound. Hence, $\|\mathsf{msg}\|_\infty^{\mathrm{can}} \leq N_b(b+1)/2$. Now, we have all the ingredients to define the scheme parameters supporting correct decryption.

Fresh noise heuristic. Let $\mathtt{ct} = \mathtt{ComFV.Encrypt}(\mathtt{pk}, \mathsf{msg})$ be a fresh ciphertext. Set $c_0 = \mathtt{ct}[0]$, $c_1 = \mathtt{ct}[1]$, and $p_0 = \mathtt{pk}[0], p_1 = \mathtt{pk}[1]$. We have, working modulo $(X^m + b)$, that

$$\frac{X^m + b}{q} \cdot [c_0 + c_1 s]_q = \frac{X^m + b}{q} \cdot (\Delta_b \cdot \mathsf{msg} + p_0 u + e_0 + p_1 us + e_1 s) \tag{5}$$

For some polynomial $g \in K$ with $\|g\|_\infty \leq 1/2$,

$$\frac{\Delta_b(X^m + b)}{q} = \left(\frac{q}{X^m + b} + g\right) \cdot \frac{X^m + b}{q} = 1 + \frac{g(X^m + b)}{q}.$$

Thus we can take $\rho = g(X^m + b) \in K$ and

$$\|\rho\|_\infty^{\mathrm{can}} = \|g(X^m + b)\|_\infty^{\mathrm{can}} \leq (b+1)\sqrt{3n}, \tag{6}$$

where the last inequality holds with very high probability due to $g(X) \leftarrow \mathcal{U}_{\mathrm{rnd}}$; see Appendix A. Now, we can expand (5) as follows

$$\begin{aligned}
\frac{X^m + b}{q} \cdot [c_0 + c_1 s]_q &= \mathsf{msg} \cdot \left(1 + \frac{\rho}{q}\right) + \frac{X^m + b}{q} \cdot (p_0 u + e_0 + p_1 us + e_1 s) \\
&= \mathsf{msg} + \frac{\rho}{q} \cdot \mathsf{msg} + \frac{X^m + b}{q} \cdot ((-as - e)u + aus + e_1 s) \\
&= \mathsf{msg} + \frac{\rho}{q} \cdot \mathsf{msg} + \frac{X^m + b}{q} \cdot (-eu + e_1 + e_2 s)
\end{aligned}$$

Here, the noisy term is $v = (\rho \cdot \mathsf{msg} + (X^m + b) \cdot (-eu + e_1 + e_2 s))/q$. Given (6) and the canonical norm analysis in Appendix A, it follows that

$$\begin{aligned}
\|v\|_\infty^{\mathrm{can}} &\leq \frac{1}{q} \cdot \left((b+1)N_b\sqrt{3n} \cdot \frac{b+1}{2} + 6(b+1)\sqrt{n}\sqrt{\sigma^2(4n/3+1)}\right) \\
&= \frac{b+1}{q} \cdot \left(\frac{\sqrt{3n}}{2} \cdot (b+1)N_b + 2\sigma n\sqrt{12 + \frac{9}{n}}\right).
\end{aligned}$$

4.2 Homomorphic operations

In this section we show how homomorphic addition and multiplication are performed in the new scheme. We prove correctness of these operations and estimate the invariant noise growth. Throughout this section, $\mathtt{Ct}(\mathsf{msg}, v)$ denotes a ciphertext encrypting message $\mathsf{msg} \in R_{X^m+b}$ with invariant noise v.

Addition is the coordinate-wise sum of corresponding ciphertext components:

- $\mathtt{ComFV.Add}(\mathtt{ct}_0, \mathtt{ct}_1)$: Return $\left([\mathtt{ct}_0[0] + \mathtt{ct}_1[0]]_q, [\mathtt{ct}_0[1] + \mathtt{ct}_1[1]]_q\right)$.

It follows immediately from (4) that the invariant noise grows additively as in the lemma below.

Lemma 4 (Addition noise). *Given two ciphertexts* $\mathtt{ct}_1 = \mathtt{Ct}(\mathsf{msg}_1, v_1)$ *and* $\mathtt{ct}_1 = \mathtt{Ct}(\mathsf{msg}_2, v_2)$*, the function* $\mathtt{ComFV.Add}(\mathtt{ct}_1, \mathtt{ct}_2)$ *returns a ciphertext* $\mathtt{ct}_{\mathtt{Add}} = \mathtt{Ct}(\mathsf{msg}_1 + \mathsf{msg}_2, v_{\mathtt{Add}})$ *with* $\|v_{\mathtt{Add}}\|_\infty^{\mathrm{can}} \leq \|v_1\|_\infty^{\mathrm{can}} + \|v_2\|_\infty^{\mathrm{can}}$.

Multiplication consists of two steps. The first one, denoted $\mathtt{ComFV.BMul}$, returns the coefficients of the ciphertext product when expressed as of a polynomial in s, namely of $(\mathtt{ct}_0[0] + \mathtt{ct}_0[1]s)(\mathtt{ct}_1[0] + \mathtt{ct}_1[1]s)$. The second step then maps the degree two term back to degree one using the relinearization technique.

- $\mathtt{ComFV.BMul}(\mathtt{ct}_0, \mathtt{ct}_1)$: Compute $c_0 = \left[\left\lfloor \frac{X^m+b}{q} \cdot \mathtt{ct}_0[0] \cdot \mathtt{ct}_1[0] \right\rceil\right]_q$,
  $c_1 = \left[\left\lfloor \frac{X^m+b}{q} \cdot (\mathtt{ct}_0[0] \cdot \mathtt{ct}_1[1] + \mathtt{ct}_0[1] \cdot \mathtt{ct}_1[0]) \right\rceil\right]_q$
  and $c_2 = \left[\left\lfloor \frac{X^m+b}{q} \cdot \mathtt{ct}_0[1] \cdot \mathtt{ct}_1[1] \right\rceil\right]_q$.
  Return $\mathtt{ct}_{\mathtt{BMul}} = (c_0, c_1, c_2)$.
- $\mathtt{ComFV.Relin}(\mathtt{ct}_{\mathtt{BMul}}, \mathtt{evk})$: Writing $\mathtt{ct}_{\mathtt{BMul}} = (c_0, c_1, c_2)$, expand c_2 in base w, namely $c_2 = \sum_{i=0}^{\ell} c_{2,i} w^i$ with $c_{2,i} \in R_w$. Compute

$$c_0' = \left[c_0 + \sum_{i=0}^{\ell} \mathtt{evk}[i][0] \cdot c_{2,i}\right]_q, \qquad c_1' = \left[c_1 + \sum_{i=0}^{\ell} \mathtt{evk}[i][1] \cdot c_{2,i}\right]_q$$

  and output $c_{\mathtt{Relin}} = (c_0', c_1')$.
- $\mathtt{ComFV.Mul}(\mathtt{ct}_0, \mathtt{ct}_1, \mathtt{evk})$: Return $c_{\mathtt{Mul}} = \mathtt{ComFV.Relin}(\mathtt{ComFV.BMul}(\mathtt{ct}_0, \mathtt{ct}_1), \mathtt{evk})$.

To estimate the noise growth of multiplication, we analyze each step above separately. First, we provide a heuristic upper bound on the noise introduced by $\mathtt{ComFV.BMul}$.

Noise heuristic after $\mathtt{ComFV.BMul}$. Given two ciphertexts $\mathtt{ct}_1 = \mathtt{Ct}(\mathsf{msg}_1, v_1)$ and $\mathtt{ct}_1 = \mathtt{Ct}(\mathsf{msg}_2, v_2)$, the function $\mathtt{ComFV.BMul}(\mathtt{ct}_1, \mathtt{ct}_2)$ returns a triple $\mathtt{ct}_{\mathtt{BMul}} = (c_0, c_1, c_2)$ According to the description of $\mathtt{ComFV.BMul}$, every component c_i of $\mathtt{ct}_{\mathtt{BMul}}$ contains a rounding error r_i, $\|r_i\|_\infty \leq 1/2$. Thus, decrypting $\mathtt{ct}_{\mathtt{BMul}}$ leads to

$$\frac{X^m+b}{q} \cdot \left[c_0 + c_1 s + c_2 s^2\right]_q = \left(\frac{X^m+b}{q}\right)^2 \cdot \mathtt{ct}_1(s) \cdot \mathtt{ct}_2(s) + r + g(X^m+b),$$

where $r = (X^m + b)(r_0 + r_1 s + r_2 s^2)/q$ and $g \in R$. According to Appendix A, the variance of $\left\|r_0 + r_1 s + r_2 s^2\right\|_\infty^{\text{can}}$ is equal to $n/12 + n^2/18 + n^3/27$. It follows that

$$\begin{aligned}\|r\|_\infty^{\text{can}} &\leq \frac{b+1}{q} 6\sqrt{n/12 + n^2/18 + n^3/27} \\ &= \frac{b+1}{q}\sqrt{3n + 2n^2 + 4n^3/3}\end{aligned}$$

Since $(X^m + b) \cdot \mathtt{ct}_i(s)/q = \mathsf{msg}_i + v_i + g_i(X^m + b)$ for some $g_i \in R$, expanding the previous expression results in

$$\begin{aligned}\frac{X^m + b}{q} \cdot \left[c_0 + c_1 s + c_2 s^2\right]_q = \mathsf{msg}_1 \cdot \mathsf{msg}_2 &+ v_2(\mathsf{msg}_1 + g_1(X^m + b)) \\ &+ v_1(\mathsf{msg}_2 + g_2(X^m + b)) \\ &+ v_1 v_2 + r \\ &+ (\mathsf{msg}_1 \cdot g_2 + \mathsf{msg}_2 \cdot g_1 + g)(X^m + b) \\ &+ g_1 g_2 (X^m + b)^2 \\ = \mathsf{msg}_1 \cdot \mathsf{msg}_2 &+ v_{\mathtt{BMul}} + h(X^m + b).\end{aligned}$$

Notice that $\mathtt{ct}_i[0]$ and $\mathtt{ct}_i[1]$ should be indistinguishable from samples generated by $\mathcal{U}_q$ according to the decision Ring-LWE problem. The variance of $\mathtt{ct}_i[0] + \mathtt{ct}_i[1] \cdot s$ is thus $q^2 n/12 + q^2 n^2/18$. Hence, it follows

$$\begin{aligned}\|\mathsf{msg}_i + g_i(X^m + b)\|_\infty^{\text{can}} &= \left\| \frac{X^m + b}{q} \cdot \mathtt{ct}_i(s) - v_i \right\|_\infty^{\text{can}} \\ &\leq \frac{b+1}{q} \cdot q\sqrt{3n + 2n^2} + \|v_i\|_\infty^{\text{can}} \\ &= (b+1)\sqrt{3n + 2n^2} + \|v_i\|_\infty^{\text{can}}.\end{aligned}$$

Hence, the noisy term $v_{\mathtt{BMul}}$ satisfies

$$\begin{aligned}\|v_{\mathtt{BMul}}\|_\infty^{\text{can}} \leq \|v_2\|_\infty^{\text{can}} &\cdot \left((b+1)\sqrt{3n + 2n^2} + \|v_1\|_\infty^{\text{can}}\right) \\ &+ \|v_1\|_\infty^{\text{can}} \cdot \left((b+1)\sqrt{3n + 2n^2} + \|v_2\|_\infty^{\text{can}}\right) \\ &+ \|v_1\|_\infty^{\text{can}} \cdot \|v_2\|_\infty^{\text{can}} + \frac{b+1}{q}\sqrt{3n + 2n^2 + 4n^3/3}.\end{aligned}$$

Finally, we obtain

$$\begin{aligned}\|v_{\mathtt{BMul}}\|_\infty^{\text{can}} \leq (b+1)\sqrt{3n + 2n^2}\left(\|v_1\|_\infty^{\text{can}} + \|v_2\|_\infty^{\text{can}}\right) &+ 3\|v_1\|_\infty^{\text{can}} \cdot \|v_2\|_\infty^{\text{can}} \\ &+ \frac{b+1}{q}\sqrt{3n + 2n^2 + 4n^3/3}\end{aligned} \tag{7}$$

with very high probability.

Next, we provide a heuristic upper bound on the noise introduced after relinearization.

Noise heuristic after `ComFV.Relin`. Given a triple $\mathtt{ct} = (c_0, c_1, c_2)$ encrypting a message msg and containing noise v, the relinearization function returns a ciphertext $\mathtt{ct}_{\mathtt{Relin}} = \mathtt{Ct}(\mathsf{msg}, v_{\mathtt{Relin}})$. As above, we scale down the output of relinearization

$$\begin{aligned}
\frac{X^m+b}{q} \cdot \left[\mathtt{ct}_{\mathtt{Relin}}(s)\right]_q &= \frac{X^m+b}{q} \cdot \left[c_0' + c_1' s\right]_q \\
&= \frac{X^m+b}{q} \cdot \left(c_0 + c_1 s + c_{2,i} \sum_{i=0}^{\ell} \mathtt{evk}[i][0] + \mathtt{evk}[i][1] \cdot s \right) \\
&\quad + g(X^m+b) \\
&= \frac{X^m+b}{q} \cdot \left(c_0 + c_1 s - \sum_{i=0}^{\ell} e_i c_{2,i} + s^2 \sum_{i=0}^{\ell} w^i c_{2,i} \right) \\
&\quad + \left(\sum_{i=0}^{\ell} g_i c_{2,i} + g \right) (X^m+b).
\end{aligned}$$

Recall that by definition $\sum_i w^i c_{2,i} = c_2$. Thus, replacing $\sum_i g_i c_{2,i} + g$ by $\tilde{g}$, we obtain for some $h \in R$

$$\begin{aligned}
\frac{X^m+b}{q} \cdot \left[\mathtt{ct}_{\mathtt{Relin}}(s)\right]_q &= \frac{X^m+b}{q} \cdot \left(c_0 + c_1 s + c_2 s^2 - \sum_{i=0}^{\ell} e_i c_{2,i} \right) + \tilde{g}(X^m+b) \\
&= \mathsf{msg} + v - \frac{X^m+b}{q} \cdot \sum_{i=0}^{\ell} e_i c_{2,i} + (\tilde{g}+h)(X^m+b)
\end{aligned}$$

As a result, $v_{\mathtt{Relin}} = v - \frac{X^m+b}{q} \cdot \sum_{i=0}^{\ell} e_i c_{2,i}$. Given that $c_{2,i}$'s look uniformly random in R_w, the variance of $\sum_{i=0}^{\ell} e_i c_{2,i}$ is equal to $(\ell+1)(w\sigma n)^2/12$. Hence, we obtain

$$\begin{aligned}
\|v_{\mathtt{Relin}}\|_{\infty}^{\mathrm{can}} &\leq \|v\|_{\infty}^{\mathrm{can}} + \frac{b+1}{q} \cdot \sum_{i=0}^{\ell} \|e_i c_{2,i}\|_{\infty}^{\mathrm{can}} \\
&\leq \|v\|_{\infty}^{\mathrm{can}} + \frac{b+1}{q} \cdot 6\sigma n w \sqrt{\frac{\ell+1}{12}}.
\end{aligned}$$

As a result, the relinearization noise satisfies

$$\|v_{\mathtt{Relin}}\|_{\infty}^{\mathrm{can}} \leq \|v\|_{\infty}^{\mathrm{can}} + \frac{b+1}{q} \cdot \sigma n w \sqrt{3(\ell+1)} \tag{8}$$

with very high probability.

Noise heuristic after `ComFV.Mul`. Combining the two previous heuristics (7) and (8), we deduce the total noise growth after homomorphic multiplication.

Given two ciphertexts $\mathtt{ct}_1 = \mathtt{Ct}(\mathtt{msg}_1, v_1)$ and $\mathtt{ct}_1 = \mathtt{Ct}(\mathtt{msg}_2, v_2)$, the function $\mathtt{ComFV.Mul}(\mathtt{ct}_1, \mathtt{ct}_2, \mathtt{evk})$ outputs a ciphertext $\mathtt{ct}_{\mathtt{Mul}} = \mathtt{Ct}(\mathtt{msg}_1 \cdot \mathtt{msg}_2, v_{\mathtt{Mul}})$ with

$$\|v_{\mathtt{Mul}}\|_\infty^{\mathrm{can}}(b+1)\sqrt{3n+2n^2}\,(\|v_1\|_\infty^{\mathrm{can}} + \|v_2\|_\infty^{\mathrm{can}}) + 3\,\|v_1\|_\infty^{\mathrm{can}} \cdot \|v_2\|_\infty^{\mathrm{can}} + \frac{b+1}{q}\sqrt{3n+2n^2+4n^3/3} + \frac{b+1}{q} \cdot \sigma n w \sqrt{3(\ell+1)}$$

with very high probability. We note that the dominating term here is the first term and not the term containing the product of the canonical norms of the multiplicands since the canonical norms are smaller than $1/2$ when the ciphertext can be decrypted correctly.

5 Application to Image Processing

In this section we apply the ComFV scheme to the image processing use case [10]. For this application, as with any other, we need to take into account two constraints regarding computation correctness. Firstly, the coefficients of encrypted encodings can increase in absolute value after arithmetic operations and reach some bound, say, B. To decode these resulting encodings, B must be smaller than $(b^{n/m}+1)/2$ as described in Section 3. Secondly, the invariant noise of encryptions grows as well according to the heuristic estimates of Section 4. To decrypt the resulting output, this noise should be smaller than $1/2$ as shown in Theorem 1.

Homomorphic Discrete Fourier Transform. We calculate the parameters of the new scheme which are compatible with the image processing pipeline given in [10].

The circuit takes input images as 8-bit integer vectors $\mathbf{a} \in \mathbb{Z}^d$ for some $d \mid m$. Then, it performs the discrete Fourier transform (DFT), $\mathcal{F}$, that maps $\mathbf{a} = (a_0, \ldots, a_{d-1})$ to a vector $\mathbf{a}' \in \mathbb{Z}^d$ such that $\mathbf{a}'[j] = \sum_{i=0}^{d-1} a_i \zeta_d^{ij}$, where ζ_d is a primitive d-th root of unity. The resulting vector is then multiplied coordinate-wise by some encrypted 8-bit integers and mapped back to $\mathbb{Z}^d$ via the inverse DFT.

Using the ComFV scheme, decoding is correct as long as $b^{n/m} + 1 > 2^{17} d^2$, for details see [10]. Notably, scalar multiplication by a root of unity is no longer noise preserving as in [10], where ζ_m^i is encoded by some power of X. According to (2), ζ_m^i is mapped to some polynomials $z(X)$ such that $\|z\|_\infty^{\mathrm{can}} \leq bn/2m$. Therefore, the canonical norm of the invariant noise is increasing after every multiplication by ζ_m^i.

Computing $\mathcal{F}$ and $\mathcal{F}^{-1}$, we resort to the mixed Fourier transform (MFT) method that combines both the fast Fourier transform (FFT) and the naive Fourier transform (NFT). In the NFT, the input vector is multiplied by a matrix $F = \left(\zeta_d^{ij}\right)_{i,j}$ that needs $O(d^2)$ multiplications and only one multiplicative level. The FFT method calls recursively smaller size DFT's such that the ith

coordinate of the DFT output is then given as

$$\mathcal{F}(\mathbf{a})[i] = \mathcal{F}(a_0, \ldots, a_{d/2-1}) + \zeta_d^i \cdot \mathcal{F}(a_{d/2}, \ldots, a_{d-1}).$$

The FFT reduces the number of multiplications to $O(d \log d)$ but needs $O(\log d)$ multiplicative levels. Thus, the FFT introduces more noise than the NFT but it is computationally faster. The MFT approach consists in computing the FFT recursion up to some dimension $\tilde{d} \leq d$ and then computing NFT.

We applied the `ComFV` scheme to 6 DFT dimensions d given in [10]. As shown in Table 1, the ciphertext size is reduced in all cases. However, only the FFT method was used in [10] while we resort sometimes to a slower MFT circuit for $d \in 2^8, 2^{12}, 2^{13}$.

Table 1. Ciphertext size comparison between our encoding and [10]. All parameters are taken to be compatible with a d-dimensional DFT circuit and the security level λ.

d	$\tilde{d}$	b	n	$\log q$	λ	`ct` size	`ct` size[10]
2^4	1	30	2^{12}	149	119	149 kB	300 kB
2^6	1	30	2^{12}	149	119	149 kB	300 kB
2^8	2^4	30	2^{13}	147	438	294 kB	300 kB
2^{10}	1	132	2^{13}	222	206	444 kB	768 kB
2^{12}	2^8	472	2^{14}	180	1004	720 kB	768 kB
2^{13}	2^{13}	$\simeq 2^{22}$	2^{14}	172	1082	688 kB	768 kB

6 Comparison with `FV`: regular circuits

To estimate the performance of `ComFV` in a general setting and fairly compare it with the original `FV` scheme and the work of [6], we resort to regular circuits as introduced in [11]. These circuits have already been used in [6] for the same purpose.

A regular circuit consists of D computational levels where each level contains $A \in \{0, 3, 10\}$ addition levels, requiring 2^A inputs, followed by one multiplication. Therefore in total the number of inputs required is $2^{D(A+1)}$. Each circuit input is given by a complex number with real and imaginary parts from $(-U, U)$ for some $U \in \{2^8, 2^{16}, 2^{32}, 2^{64}\}$. We will always use a precision of 16 fractional bits in this paper which in the case of a complex number refers to both the real and complex parts independently.

Our aim is to compare `ComFV` to the previously best known scheme allowing native complex inputs as well as to the state of the art when encoding the real and imaginary parts separately [6]. We will compare this method with our method where we use the same encoding of the complex number as a cyclotomic integer. We chose $m = 4$ as this is the minimal m for which $\mathbb{Z}[\zeta]$ is dense in $\mathbb{C}$ and it allows us to use $b = 4^h$ for some $h \in \mathbb{N}$, taking $\alpha = 2^{h/2}$ if h is even and

$\alpha = 2^{(h(n+4)-4)/8}(2^{hn/4} - 1)$ if h is odd. We also use $m = 4$ when using `FV` and one may wonder if taking a larger m is better. However, we found that using larger m in this case gave the same depths and only increased the time to encode a complex number.

For the current state of the art we use the scheme of Chen et al. [6], which we call CLPX, and encode the real and imaginary parts of our complex number separately. Thus an encryption now consists of two ciphertext pairs and addition is performed component-wise while we use the Karatsuba algorithm to perform multiplication using only three calls to the multiplication algorithm of the underlying scheme. We use the same values for n and q for comparison so that ciphertexts will be twice as large compared to our work. The fractional encoder is used to encode the real and imaginary parts so we use $m = 2$ in this case. For the optimal value of b we restrict our search space to powers of 2, since we require a precision of 2^{-16}, the simplest way to ensure correct decoding at depth D is to require $2^{16D} \mid b^{n/4}$ so taking b a power of two looks a good fit. We again compare this approach with ours, in this case we also use the fractional encoder.

We computed the theoretical and heuristic maximal depth of a regular circuit which can be reached using `FV`, the CLPX approach of using plaintext modulus $X - b$ and our `ComFV` with parameters n, q, σ given in the SEAL library [5] and the relinearization base $w = 2^{32}$. Our results are presented in Tables 2 and 3. In the tables we also give a value for b (or t) which allows one to reach this maximal depth, this b is very often not unique and in this case we give the smallest b for which there is a decryption error at the next level. To find a heuristic estimate of the maximal depth that can be reached in each scheme we take a carefully chosen complex number and use this as the complex number given for all inputs of the circuit. One reason for this can be seen in the table of results, Table 3, where we see that for $A = 10$, depths of 14 can be achieved, this requires $2^{14 \cdot 11} = 2^{154}$ inputs, meaning using different inputs would be completely infeasible in practice. Another good reason for choosing all inputs to be the same is that during addition there is no cancellation occurring, indeed the A levels of addition simply become the worst case of scaling by 2^A. The precise complex number we chose depends on the encoding scheme but essentially one finds one with an encoding which has many large coefficients. If the fractional encoder is used then we take the complex number to be $(U - 2^{-16})(1 + \mathbf{i})$ while when using the cyclotomic integer approximation approach it is a matter of trial and error but this need only be done once for each U and m.

From Table 3 we see that in all cases our methods greatly outperform the best scheme natively encrypting complex numbers. At a minimum we can achieve 5 times the depth and for larger n our method becomes even more efficient as the amount of plaintext space not being efficiently used only grows in the current solution. The CLPX method on the other hand is able to achieve slightly larger depths than our scheme, at most one more for the largest n we consider. Where our method improves is on efficiency, we effectively halve the ciphertext size and are expected to be roughly three times faster due to the fact that we can use one multiplication operation per level whereas the CLPX approach requires three.

Table 2. Maximal theoretical regular circuit depths of `FV` (D_O) with the approximation encoding, the CLPX approach encrypting the real and imaginary parts separately (D_M), `ComFV` with the approximation encoding (D_A) and the fractional encoding (D_F) depending on input size (U), number of additions per level (A), n and q. Corresponding t and b's are provided.

	n	4096			8192			16384			32768		
	$\log q$	116			226			435			889		
	A	0	3	10	0	3	10	0	3	10	0	3	10
$U = 2^8$	D_O	1	0	0	1	1	1	2	2	1	3	3	2
	t_O	2^{34}	–	–	2^{34}	2^{40}	2^{54}	2^{68}	2^{86}	2^{54}	2^{135}	2^{177}	2^{128}
	D_M	4	3	3	9	8	6	12	12	11	15	14	14
	b_M	2	2	2	2^3	2^2	2	2^9	2^9	2^5	2^{33}	2^{17}	2^{17}
	D_A	5	4	3	9	8	6	11	11	10	14	13	12
	b_A	2^2	2^2	2^2	2^6	2^4	2^2	2^{10}	2^{12}	2^{10}	2^{34}	2^{24}	2^{20}
	D_F	5	4	3	9	8	7	11	11	10	14	14	13
	b_F	2	2	2	2^5	2^3	2^2	2^9	2^9	2^8	2^{33}	2^{33}	2^{29}
$U = 2^{16}$	D_O	1	0	0	1	1	1	2	2	1	3	3	2
	t_O	2^{34}	–	–	2^{34}	2^{40}	2^{54}	2^{67}	2^{85}	2^{54}	2^{134}	2^{176}	2^{127}
	D_M	4	3	3	9	8	6	12	12	11	14	14	14
	b_M	2	2	2	2^3	2^2	2	2^9	2^9	2^5	2^{18}	2^{18}	2^{18}
	D_A	5	4	3	9	8	6	11	11	10	14	13	12
	b_A	2^2	2^2	2^2	2^6	2^4	2^2	2^{10}	2^{12}	2^{10}	2^{34}	2^{24}	2^{20}
	D_F	5	4	3	9	8	7	11	11	10	14	13	12
	b_F	2	2	2	2^5	2^3	2^3	2^9	2^{12}	2^{10}	2^{34}	2^{23}	2^{19}
$U = 2^{32}$	D_O	0	0	0	1	1	1	1	1	1	2	2	2
	t_O	–	–	–	2^{65}	2^{71}	2^{85}	2^{65}	2^{71}	2^{85}	2^{130}	2^{148}	2^{190}
	D_M	4	3	3	8	8	6	11	11	10	14	14	13
	b_M	2	2	2	2^3	2^3	2	2^9	2^9	2^5	2^{34}	2^{34}	2^{17}
	D_A	5	4	3	8	8	6	11	10	9	13	13	12
	b_A	2^2	2^2	2^2	2^6	2^6	2^2	2^{18}	2^{10}	2^8	2^{34}	2^{40}	2^{28}
	D_F	5	4	3	8	8	6	11	10	9	13	13	12
	b_F	2^2	2	2	2^5	2^5	2^2	2^{17}	2^{10}	2^7	2^{33}	2^{39}	2^{27}
$U = 2^{64}$	D_O	–	–	–	0	0	0	1	1	1	2	1	1
	t_O	–	–	–	–	–	–	2^{129}	2^{135}	2^{149}	2^{258}	2^{135}	2^{149}
	D_M	4	3	3	8	7	6	10	10	10	13	13	12
	b_M	2	2	2	2^5	2^3	2^2	2^9	2^9	2^9	2^{33}	2^{33}	2^{17}
	D_A	4	4	3	7	7	6	10	10	9	12	12	11
	b_A	2^2	2^2	2^2	2^6	2^6	2^4	2^{18}	2^{18}	2^{12}	2^{34}	2^{36}	2^{22}
	D_F	4	4	3	7	7	6	10	10	9	12	12	11
	b_F	2^2	2^2	2	2^5	2^5	2^3	2^{17}	2^{18}	2^{11}	2^{33}	2^{36}	2^{22}

7 Conclusion

We constructed a new encoding algorithm for complex data values and a corresponding somewhat homomorphic encryption scheme by utilizing a polynomial plaintext modulus of the form $X^m + b$. This choice allows for a much better use of the available plaintext space and much slower noise growth compared to existing solutions encrypting complex numbers. As a result, for the same ciphertext modulus q and degree n, we can homomorphically evaluate between 5 and 12 times deeper circuits compared to existing solutions based on FV and natively encoding complex numbers. In comparison to the state of the art, which encrypts the real and imaginary parts of the complex numbers separately, our

Table 3. Maximal heuristic regular circuit depths of the original `FV` scheme with native complex inputs (D_O), the CLPX approach encrypting the real and imaginary parts separately (D_M), `ComFV` with the approximation encoding (D_A) and the fractional encoding (D_F) depending on input size (U), number of additions per level (A), n and q. A corresponding t or b is provided.

	n	4096			8192			16384			32768		
	$\log q$	116			226			435			889		
	A	0	3	10	0	3	10	0	3	10	0	3	10
$U = 2^8$	D_O	1	1	0	1	1	1	2	2	2	3	3	2
	t_O	2^{35}	2^{41}	2^{18}	2^{35}	2^{41}	2^{55}	2^{70}	2^{88}	2^{130}	2^{164}	2^{182}	2^{202}
	D_M	6	5	4	10	9	8	13	12	11	15	15	14
	b_M	2	2	2	2^5	2^4	2^2	2^{16}	2^{14}	2^{10}	2^{37}	2^{34}	2^{31}
	D_A	6	5	4	9	9	7	12	11	10	14	13	13
	b_A	2^2	2^2	2^2	2^6	2^6	2^6	2^{18}	2^{18}	2^{10}	2^{40}	2^{40}	2^{38}
	D_F	6	5	4	9	9	7	12	12	10	14	14	13
	b_F	2	2	2	2^4	2^4	2^2	2^{16}	2^{15}	2^8	2^{32}	2^{33}	2^{33}
$U = 2^{16}$	D_O	1	1	0	1	1	1	2	2	2	3	3	2
	t_O	2^{35}	2^{41}	2^{18}	2^{35}	2^{41}	2^{55}	2^{70}	2^{88}	2^{130}	2^{164}	2^{173}	2^{201}
	D_M	6	5	4	10	9	7	12	12	11	15	14	13
	b_M	2	2	2	2^5	2^4	2^2	2^{17}	2^{14}	2^{10}	2^{37}	2^{38}	2^{35}
	D_A	6	5	4	9	9	7	12	11	10	14	13	13
	b_A	2^2	2^2	2^2	2^6	2^6	2^6	2^{18}	2^{18}	2^{10}	2^{40}	2^{40}	2^{38}
	D_F	6	5	4	9	9	7	12	11	10	14	13	13
	b_F	2^2	2	2	2^5	2^6	2^3	2^{17}	2^{15}	2^{10}	2^{33}	2^{41}	2^{37}
$U = 2^{32}$	D_O	0	0	0	1	1	1	1	1	1	2	2	2
	t_O	2^{33}	2^{33}	2^{33}	2^{65}	2^{71}	2^{84}	2^{65}	2^{71}	2^{85}	2^{206}	2^{205}	2^{198}
	D_M	5	5	4	9	9	7	12	11	10	14	14	13
	b_M	2^2	2	2	2^7	2^5	2^2	2^{17}	2^{16}	2^{13}	2^{40}	2^{39}	2^{35}
	D_A	5	5	4	8	8	7	11	10	10	13	13	12
	b_A	2^2	2^2	2^2	2^6	2^6	2^6	2^{18}	2^{18}	2^{14}	2^{40}	2^{40}	2^{40}
	D_F	5	5	4	9	8	7	11	10	10	13	13	12
	b_F	2^2	2^2	2	2^9	2^6	2^4	2^{17}	2^{15}	2^{14}	2^{33}	2^{41}	2^{38}
$U = 2^{64}$	D_O	—	—	—	0	0	0	1	1	1	2	1	1
	t_O	—	—	—	2^{65}	2^{65}	2^{65}	2^{129}	2^{135}	2^{149}	2^{258}	2^{266}	2^{262}
	D_M	5	5	4	8	8	7	11	11	10	13	13	12
	b_M	2^2	2^2	2	2^9	2^6	2^3	2^{19}	2^{18}	2^{13}	2^{44}	2^{41}	2^{39}
	D_A	5	4	4	8	7	7	10	10	9	12	12	12
	b_A	2^4	2^4	2^2	2^{10}	2^6	2^6	2^{18}	2^{18}	2^{14}	2^{40}	2^{40}	2^{44}
	D_F	5	5	4	8	8	7	10	10	9	12	12	12
	b_F	2^3	2^3	2^2	2^9	2^9	2^6	2^{17}	2^{18}	2^{14}	2^{33}	2^{41}	2^{43}

method reduces the size of ciphertexts by a factor of 2 making our scheme at least twice as efficient in time and three times more efficient in space.

References

1. Barnett, A., Santokhi, J., Simpson, M., Smart, N.P., Stainton-Bygrave, C., Vivek, S., Waller, A.: Image classification using non-linear support vector machines on encrypted data (2017), cryptology ePrint Archive: Report 2017/857
2. Bonte, C., Bootland, C., Bos, J.W., Castryck, W., Iliashenko, I., Vercauteren, F.: Faster homomorphic function evaluation using non-integral base encoding. In: CHES 2017. LNCS, vol. 10529, pp. 579–600. Springer, Heidelberg (Sep 2017)

3. Bos, J.W., Castryck, W., Iliashenko, I., Vercauteren, F.: Privacy-friendly forecasting for the smart grid using homomorphic encryption and the group method of data handling. In: AFRICACRYPT 17. LNCS, vol. 10239, pp. 184–201. Springer, Heidelberg (May 2017)
4. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. In: ITCS 2012. pp. 309–325. ACM (Jan 2012)
5. Chen, H., Laine, K., Player, R.: Simple encrypted arithmetic library - SEAL v2.1. In: FC 2017. vol. 10323, pp. 3–18. Springer, Heidelberg (2017)
6. Chen, H., Laine, K., Player, R., Xia, Y.: High-precision arithmetic in homomorphic encryption. In: Smart, N.P. (ed.) CT-RSA 2018. LNCS, vol. 10808, pp. 116–136. Springer, Heidelberg (2018)
7. Cheon, J.H., Jeong, J., Lee, J., Lee, K.: Privacy-preserving computations of predictive medical models with minimax approximation and non-adjacent form. In: FC 2017. vol. 10323, pp. 53–74. Springer, Heidelberg (2017)
8. Cheon, J.H., Kim, A., Kim, M., Song, Y.S.: Homomorphic encryption for arithmetic of approximate numbers. In: ASIACRYPT 2017, Part I. LNCS, vol. 10624, pp. 409–437. Springer, Heidelberg (Dec 2017)
9. Costache, A., Smart, N.P.: Which ring based somewhat homomorphic encryption scheme is best? In: CT-RSA 2016. LNCS, vol. 9610, pp. 325–340. Springer, Heidelberg (Feb / Mar 2016)
10. Costache, A., Smart, N.P., Vivek, S.: Faster homomorphic evaluation of discrete Fourier transforms. In: FC 2017. LNCS, vol. 10322, pp. 517–529 (2017)
11. Costache, A., Smart, N.P., Vivek, S., Waller, A.: Fixed-point arithmetic in SHE schemes. In: SAC 2016. LNCS, vol. 10532, pp. 401–422. Springer, Heidelberg (Aug 2016)
12. Dowlin, N., Gilad-Bachrach, R., Laine, K., Lauter, K., Naehrig, M., Wernsing, J.: Manual for using homomorphic encryption for bioinformatics. Tech. rep., MSR-TR-2015-87, Microsoft Research (2015)
13. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144 (2012), `http://eprint.iacr.org/2012/144`
14. Geihs, M., Cabarcas, D.: Efficient integer encoding for homomorphic encryption via ring isomorphisms. In: LATINCRYPT 2014. LNCS, vol. 8895, pp. 48–63. Springer, Heidelberg (Sep 2015)
15. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: 41st ACM STOC. pp. 169–178. ACM Press (May / Jun 2009)
16. Gentry, C., Halevi, S., Smart, N.P.: Homomorphic evaluation of the AES circuit. In: CRYPTO 2012. LNCS, vol. 7417, pp. 850–867. Springer, Heidelberg (Aug 2012)
17. Hoffstein, J., Pipher, J., Silverman, J.H.: Ntru: A ring-based public key cryptosystem. In: Algorithmic Number Theory, Third International Symposium, ANTS-III. pp. 267–288. Springer, Heidelberg (1998)
18. Lauter, K., López-Alt, A., Naehrig, M.: Private computation on encrypted genomic data. In: LATINCRYPT 2014. LNCS, vol. 8895, pp. 3–27 (Sep 2015)
19. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (May 2010)
20. Naehrig, M., Lauter, K.E., Vaikuntanathan, V.: Can homomorphic encryption be practical? In: ACM Cloud Computing Security Workshop – CCSW. pp. 113–124. ACM (2011)

A The canonical norm

This appendix closely follows Appendix A.5 of the ePrint version of [6].

Let $K = \mathbb{Q}[X]/(f(X))$ be a cyclotomic number field where, as usual, $f(X) = X^n + 1$ is the $2n$-cyclotomic polynomial, n a power of two. We denote the ring of integers of K by R, i.e. $R = \mathbb{Z}[X]/(f(X))$. Let R_a be the reduction of R modulo an ideal (a). If a is a natural number $R_a = \mathbb{Z}_a[X]/(f(X))$ and we take representatives of $\mathbb{Z}/a\mathbb{Z}$ from the half-open interval $[-a/2, a/2)$.

For any $a = \sum_i a_i X^i \in K$, *the infinity norm* $\|a\|_\infty$ is defined as $\max_i |a_i|$. We denote by δ_R the upper bound on $\|ab\|_\infty / \|a\|_\infty \cdot \|b\|_\infty$ for any $a, b \in R$. This bound is called *the expansion factor* of R. For a our ring of cyclotomic integers R, the expansion factor is $\delta_R = n$. Let ζ is a complex primitive $2n$-th root of unity. We define *the canonical norm* as

$$\|a\|_\infty^{\text{can}} = \left\| \left(a(\zeta), a(\zeta^3), \ldots, a(\zeta^{2n-1})\right) \right\|_\infty .$$

It is easy to check that the canonical norm satisfies

$$\|a\|_\infty \le \|a\|_\infty^{\text{can}} , \qquad \|a+b\|_\infty^{\text{can}} \le \|a\|_\infty^{\text{can}} + \|b\|_\infty^{\text{can}} , \qquad \|ab\|_\infty^{\text{can}} \le \|a\|_\infty^{\text{can}} \cdot \|b\|_\infty^{\text{can}} .$$

The last inequality implies that the canonical norm leads to tighter bounds than the infinity norm [19].

Canonical norm of random polynomials We will need to bound the canonical norm of random polynomials whose coefficients are generated from a discrete Gaussian or uniform distributions. We follow a heuristic approach given in [16, A.5], which was already used in [9, 5, 6] for an analysis of the FV scheme.

Let $a \in R$ be a polynomial such that its coefficients are chosen independently from some zero-mean distribution with standard deviation σ. For this purpose, we use the following distributions

- a discrete Gaussian distribution $\mathcal{D}(\sigma^2)$ with PMF proportional to $\exp(-\frac{|x|^2}{2\sigma^2})$,
- the uniform distribution $\mathcal{U}_3$ over the ternary set $\{-1, 0, 1\}$,
- the uniform distribution $\mathcal{U}_q$ over $\mathbb{Z}_q$,
- the uniform distribution $\mathcal{U}_{\text{rnd}}$ over the interval $(-1/2, 1/2]$.

By the definition of the canonical norm, we need to compute $a(\zeta_{2n}^i)$. The evaluation $a(\zeta^i)$ is the inner product between the coefficient vector of a and the fixed vector $\left(1, \zeta^i, \ldots, \zeta^{i(n-1)}\right)$, which has Euclidean norm $\sqrt{n}$. Hence, the random variable $a(\zeta_{2n}^i)$ has variance $V = \sigma^2 n$ by the Cauchy-Schwartz inequality.

When $a_i \leftarrow \mathcal{D}(\sigma^2)$ then the coefficients have variance $\simeq \sigma^2$ and thus the variance of $a(\zeta_{2n}^i)$ is $V_{\mathcal{D}} \simeq \sigma^2 n$. If $a_i \leftarrow \mathcal{U}_3$ then the coefficients have variance $2/3$ and thus the total variance is $V_{\mathcal{U}_3} = 2n/3$. By analogy, $V_{\mathcal{U}_q} \lesssim q^2 n/12$ as the a_i has variance roughly $q^2/12$. Finally, the variance of $a_i \leftarrow \mathcal{U}_{\text{rnd}}$ is equal to $1/12$, so $V_{\mathcal{U}_{\text{rnd}}} = n/12$.

Since $a(\zeta_{2n}^i)$ is the sum of independently distributed complex variables, by the law of large numbers it is distributed similarly to a complex Gaussian random

variable of variance V. Therefore, given that $\mathrm{erfc}(6) \simeq 2^{-55}$, we can use $6\sqrt{V}$ as a high-probability bound on $a(\zeta_{2n}^i)$. Since in practice $n \geq 2^{12}$, this bound is good enough to claim that $\|a\|_\infty^{\mathrm{can}} \leq 6\sqrt{V}$ with very high probability. For the distributions above, we get

$$\begin{aligned}
\|a\|_\infty^{\mathrm{can}} &\leq 6\sigma\sqrt{n}, \quad a_i \leftarrow \mathcal{D}(\sigma^2),\\
\|a\|_\infty^{\mathrm{can}} &\leq 2\sqrt{6n}, \quad a_i \leftarrow \mathcal{U}_3,\\
\|a\|_\infty^{\mathrm{can}} &\leq q\sqrt{3n}, \quad a_i \leftarrow \mathcal{U}_q,\\
\|a\|_\infty^{\mathrm{can}} &\leq \sqrt{3n}, \quad a_i \leftarrow \mathcal{U}_{\mathrm{rnd}}.
\end{aligned}$$

We also need to bound the canonical norm of a product of two random polynomials a and b whose coefficients are independently sampled from zero-mean distributions with variances σ_1^2, σ_2^2, respectively. Writing the product $ab \bmod (X^n + 1)$ with relation to the power basis of R, we obtain

$$\begin{pmatrix} a_0 & -a_{n-1} & \dots & -a_1 \\ \vdots & \vdots & & \vdots \\ a_{n-1} & a_{n-2} & \dots & a_0 \end{pmatrix} \begin{pmatrix} b_0 \\ \vdots \\ b_{n-1} \end{pmatrix} = \begin{pmatrix} g_0 \\ \vdots \\ g_{n-1} \end{pmatrix}.$$

Hence, the product coefficients are equal to

$$g_k = \sum_{i=0}^{k} a_i b_{k-i} - \sum_{i=k+1}^{n-1} a_i b_{k+n-i}.$$

for any $k \in [0, \dots, n-1]$. Since the coefficient distributions are independent and have zero mean, the product of any pair a_i, b_j has variance $\sigma_1^2\sigma_2^2$ and zero mean. Hence, the variance of each coefficient g_k is equal to $n\sigma_1^2\sigma_2^2$. Following the above reasoning, the canonical norm of $g(\zeta^i)$ is thus bounded by

$$\|ab\|_\infty^{\mathrm{can}} \leq 6n\sigma_1\sigma_2.$$

This means that the variance of the coefficients of ue where $u \leftarrow \chi_k$ and $e \leftarrow \chi_e$ is approximately $2\sigma^2 n/3$. We can now give the variance of the term appearing in the analysis of the decryption noise.

Let $e, e_1, e_2 \leftarrow \chi_e$ and $u, s \leftarrow \chi_k$. We have just seen that the variance of the coefficients of both $-eu$ and $e_2 s$ is approximately $2\sigma^2 n/3$ while the variance of the coefficients of e_1 is approximately σ^2. Because they are independent, we can sum the variances to obtain $\sigma^2(4n/3 + 1)$.

Curriculum vitae

Ilia Iliashenko was born in Shimanovsk, Russia in 1990. He completed his Specialist degree (similar to Master) at the Immanuel Kant Baltic Federal University (IKBFU), Kaliningrad, Russia with majors in cryptography and mathematics. After graduation he studied algebraic geometric codes as a post-doctoral researcher at IKBFU and worked as a C++ programmer in the video game industry. In 2015 he joined COSIC as a PhD candidate under the supervision of Prof. Bart Preneel and Prof. Frederik Vercauteren.

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
IMEC-COSIC
Kasteelpark Arenberg 10 box 2452
B-3001 Leuven
ilia@esat.kuleuven.be