JOÃO MARCOS DE M. BARGUIL

EFFICIENT METHODS FOR LATTICE-BASED CRYPTOGRAPHY

MÉTODOS EFICIENTES PARA CRIPTOGRAFIA BASEADA EM RETICULADOS

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Ciências.

São Paulo
2015

JOÃO MARCOS DE M. BARGUIL

EFFICIENT METHODS FOR LATTICE-BASED CRYPTOGRAPHY

MÉTODOS EFICIENTES PARA CRIPTOGRAFIA BASEADA EM RETICULADOS

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Ciências.

Área de Concentração:

Engenharia de Computação

Orientador:

Prof. Dr. Paulo Sérgio L. M. Barreto

São Paulo
2015

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 05/10/2015.

Assinatura do autor

Assinatura do orientador

FICHA CATALOGRÁFICA

Barguil, João Marcos de Mattos

Efficient Methods for Lattice-Based Cryptography/ J. M. Barguil. – versão corr. – São Paulo, 2015.

93 p.

Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1. Criptologia. 2. Reticulados. 3. Algoritmos. I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais. II. t.

Para Dona Cida, que me deu uma cartilha e me ensinou a ler,
e Seu Elias, que me deu um geoatlas e me ensinou o mundo.

AGRADECIMENTOS

Agradeço a Deus, que se faz sempre presente nas pequenas e grandes coisas.

A meu caro amigo Paulo Barreto, que tanto nos ensina, orienta e lidera, não apenas mostrando o rumo, mas caminhando junto, passo a passo, e por quem tenho admiração profunda.

Aos meus colegas e amigos do Sembei, nem todos geograficamente próximos, mas sempre presentes, por todas as infindáveis discussões criptográficas e piadas (infames). Ao Yuri, a quem também pertencem vários dos resultados aqui apresentados. Ao apoio do University Research Office da Intel Labs, através do programa *"Energy-Efficient Security for SoC Devices – Asymmetric Cryptography for Embedded Systems"* 2012.

À síndrome de Estocolmo, única explicação plausível para este amor imenso que sinto pela Escola Politécnica da Universidade de São Paulo.

Aos meus pais, pelo amor e apoio incondicionais. Aos meus avós (presentes ou ausentes) e família, pelo exemplo e carinho. À Sabrina, pela grande companhia. Aos meus amigos em tantos continentes, pela inspiração.

Ao Mestre, por tantas coisas que não sei nem por onde começar.

Muito obrigado.

– – –

I thank God, always present in big and small things.

To my dear friend Paulo Barreto, who always teaches, orients and leads us, not only showing the way, but also walking the path, step by step, and for whom I have deep admiration.

To my colleagues and friends from Sembei, not all geographically close, but always present, for all the endless cryptographic discussions and (bad) puns. To Yuri, to whom many of the results also belong. To Intel Labs' support through the University Research Office grant *"Energy-Efficient Security for SoC Devices – Asymmetric Cryptography for Embedded Systems"* 2012.

To Stockholm syndrome, the only plausible explanation for this immense love I feel for the Polytechnic School of the University of São Paulo.

To my parents, for the unconditional love and support. To my grandparents (present or absent) and family, for the example and care. To Sabrina, for the great company. To my friends in many continents, for the inspiration.

To the Master, for so many things I don't even know where to start.

Thank you.

RESUMO

Reticulados têm sido aplicados de diferentes maneiras em criptografia. Inicialmente utilizados para a destruição de criptossistemas, eles foram posteriormente aplicados na construção de novos esquemas, incluindo criptossistemas assimétricos, esquemas de assinatura cega e os primeiros métodos para encriptação completamente homomórfica. Contudo, seu desempenho ainda é proibitivamente lenta em muitos casos. Neste trabalho, expandimos técnicas originalmente desenvolvidas para encriptação homomórfica, tornando-as mais genéricas e aplicando-as no esquema GGH-YK-M, um esquema de encriptação de chave pública, e no esquema LMSV, a única construção homomórfica que não sucumbiu a ataques de recuperação de chaves IND-CCA1 até o momento. Em nossos testes, reduzimos o tamanho das chaves do GGH-YK-M em uma ordem de complexidade, especificamente, de $O(n^2 \lg n)$ para $O(n \lg n)$, onde n é um parâmetro público do esquema. A nova técnica também atinge processamento mais rápido em todas as operações envolvidas em um criptossistema assimétrico, isto é, geração de chaves, encriptação e decriptação. A melhora mais significativa é na geração de chaves, que se torna mais de 3 ordens de magnitude mais rápida que resultados anteriores, enquanto a encriptação se torna por volta de 2 ordens de magnitude mais rápida. Para decriptação, nossa implementação é dez vezes mais rápida que a literatura. Também mostramos que é possível aumentar a segurança do esquema LMSV contra os ataques quânticos de recuperação de chaves recentemente publicados pela agência britânica GCHQ. Isso é feito através da adoção de reticulados não-ciclotômicos baseados em anéis polinomiais irredutíveis quase-circulantes. Em nossa implementação, o desempenho da encriptação é virtualmente idêntico, e a decriptação torna-se ligeiramente inferior, um pequeno preço a se pagar pelo aumento de segurança. A geração de chaves, porém, é muito mais lenta, devido à necessidade de se utilizar um método mais genérico e caro. A existência de métodos dedicados altamente eficientes para a geração de chaves nesta variante mais segura do LMSV permanece como um problema em aberto.

Palavras-chave: Criptografia. Criptografia Pós-Quântica. Reticulados. Encriptação Homomórfica.

ABSTRACT

Lattices have been applied in many different ways in cryptography. Firstly used for the destruction of cryptosystems, they were later applied in the construction of new schemes, including asymmetric cryptosystems, blind signature schemes and the first methods for fully homomorphic encryption. Nonetheless, performance is still prohibitively slow in many cases. In this work, we expand techniques originally devised for homomorphic encryption, making them more general and applying them to the GGH-YK-M cryptosystem, a lattice-based public-key cryptosystem, and to the LMSV scheme, the only known homomorphic scheme that has not succumbed to IND-CCA1 key recovery attacks to this date. In our tests, we reduce public key bandwidth occupation of GGH-YK-M by an order of complexity, specifically, from $O(n^2 \lg n)$ down to $O(n \lg n)$ bits, where n is a public parameter of the scheme. The new technique also attains faster processing in all operations involved in an asymmetric cryptosystem, that is, key generation, encryption, and decryption. The most significant improvement in performance is in key generation, which becomes more than 3 orders of magnitude faster than previous results, while encryption becomes about 2 orders of magnitude faster. For decryption, our implementation is ten times faster than the literature. We also show that it is possible to improve security of LMSV against the quantum key recovery attacks recently published by British GCHQ. We do so by adopting non-cyclotomic lattices based on nearly-circulant irreducible polynomial rings. In our implementation, performance of encryption remains virtually the same, and decryption becomes slightly worse, a small price to pay for the improved security. Key generation, however, is much slower, due to the fact that it is necessary to use a more generic and expensive method. The existence of highly efficient dedicated methods for key generation of this secure variant of LMSV remains as an open problem.

Keywords: Cryptography. Post-Quantum Cryptography. Lattices. Homomorphic Encryption.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AES	Advanced Encryption Standard
CA	Certificate Authority
BKZ	Blockwise Korkine-Zolotarev
CESG	Communications-Electronics Security Group
CRT	Chinese remainder theorem
CCA1	Chosen ciphertext attack
CCA2	Adaptive chosen ciphertext attack
CPA	Chosen plaintext attack
CVP	Closest vector problem
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
FFT	Fast Fourier transform
FHE	Fully Homomorphic Encryption
GCHQ	United Kingdom's Government Communications Headquarters
GCD	Greatest common divisor
GGH	Goldreich, Goldwasser and Halevi
GGH-YK	Goldreich, Goldwasser, Halevi, Yoshino and Kunihiro
GGH-YK-M	Goldreich, Goldwasser, Halevi, Yoshino, Kunihiro and M-matrices
HNF	Hermite normal form

IND-CCA1	Indistinguishability under chosen ciphertext attack
IND-CCA2	Indistinguishability under adaptive chosen ciphertext attack
IND-CPA	Indistinguishability under chosen plaintext attack
JVM	Java Virtual Machine
LLL	Lenstra, Lenstra and Lovász
RELIC	RELIC is an Efficient LIbrary for Cryptography
RSA	Rivest, Shamir and Adleman
SHE	Somewhat Homomorphic Encryption
SIVP	Shortest independent vector problem
SSH	Secure shell
SVP	Shortest vector problem
TLS	Transport Layer Security

1 INTRODUCTION

Since remote times, cryptographic methods have been used for ensuring privacy in many scenarios (D'AGAPEYEFF, 1939), like exchanging information via insecure channels and protecting information stored in media that can potentially be accessed by adversaries. Cryptography is used to protect information even when an attacker is able to intercept communication. Unauthorized access is prevented because to reveal protected information one must possess a secret, dubbed a *key*. Without the correct key, it is highly unlikely that an attacker can expose protected information.

In addition to providing confidentiality, modern cryptography is also used for guaranteeing integrity, authentication and non-repudiation. It is important to note, though, that current methods do not offer absolute protection against violation. Instead, they are designed to minimize the probability of an attack being successful, making it exponentially low in some security parameter.

Presently used methods aim to ensure that attacking a secure cryptographic scheme using brute force, that is, by testing all possible keys, would demand a ludicrously high amount of computation and time, taking much longer than the average human lifespan to be successful (at least, using currently available technology). Thus, accessing the information contained in a certain amount of encrypted data without possession of the correct key is deemed to be infeasible.

1.1 Background and motivation

There is a rising, medium-to-long term concern with the potential technological viability of quantum computers, because traditional cryptosystems based on the assumed hardness of integer factorization or discrete logarithm computation can be successfully attacked with the help of this new kind of equipment (SHOR, 1995). New schemes based on different computational problems are thus necessary to address this concern, leading to the development of purely classical, but believed to be quantum-resistant constructions known as *post-quantum* cryptosystems (BERNSTEIN; BUCHMANN; DAHMEN, 2008).

One of the most popular families of post-quantum cryptosystems is that of schemes based on the theory of lattices. This concept was first applied in cryptology for destructive purposes, being used to effectively break various known schemes (ODLYZKO, 1990; JOUX; STERN, 1998). Nevertheless, their value is not limited to cryptanalysis (NGUYEN; STERN, 2001), as lattices have also been used in the construction of cryptosystems and in a few security proofs. Lattices have permitted advancements in asymmetric cryptography (GOLDREICH; GOLDWASSER; HALEVI, 1997) and play a crucial role in the development of homomorphic schemes (GENTRY, 2009b).

One of the first lattice-based encryption schemes, proposed by Goldreich, Goldwasser and Halevi (GOLDREICH; GOLDWASSER; HALEVI, 1997), or GGH for short, was later broken by Nguyen (NGUYEN, 1999), who proved that the original scheme had structural flaws. For several years GGH was deemed irretrievably broken, until Yoshino and Kunihiro (YOSHINO; KUNIHIRO, 2012) described a variant that prevented all known attacks, named GGH-YK. However, this variant does not allow for the construction of proper parameter sets, and therefore has no practical use. More recently, Barros and Schechter (BARROS; SCHECHTER, 2014) expanded this construction, proposing a modification called GGH-YK-M that effectively yields a suitable

parametrization. The result is very promising, as it brings the simplicity of GGH and GGH-YK back to life. This family of schemes may, once again, be a viable foundation for the construction of other post-quantum systems.

Another relevant cryptosystem based on GGH-style lattices is the pioneering scheme devised by Gentry (GENTRY, 2009a). It is the first known instance of *fully homomorphic encryption*, that is, the first scheme that supports arbitrary arithmetic to be performed over encrypted data while still yielding valid results upon decryption, as if the computation had been done over unencrypted data. It produces, though, very large keys, of the order of Terabytes, and demands Gigabytes of ciphertext to encrypt a single bit. Nevertheless, this work represents a major breakthrough, and prompted the development of many other constructions, some even based on different kinds of lattices. Unfortunately, most of these have already succumbed to cryptanalysis (SZYDLO, 2003; CHENAL; TANG, 2014), with the exception of the method proposed by Loftus *et al.* (LOFTUS et al., 2012), known as LMSV and a direct descendant of Gentry's original scheme. That scheme is *proven* to be IND-CCA1, uniquely among all fully or somewhat homomorphic encryption schemes known today.

Nonetheless, the biggest drawbacks of all cryptosystems based on GGH-style lattices, up to now, are their latent high bandwidth occupation and computational cost, which make this family of schemes less competitive in practice with other lattice-based encryption methods, such as the ones based on the Learning With Errors (LWE) problem (REGEV, 2005) and the NTRU cryptosystem (HOFFSTEIN; PIPHER; SILVERMAN, 1998). The technique proposed by Smart and Vercauteren (SMART; VERCAUTEREN, 2010) and improved by Gentry and Halevi (GENTRY; HALEVI, 2011), aimed at homomorphic encryption, can be modified to tackle these specific issues, being applicable not only in homomorphic schemes, but also in traditional public-key cryptography.

Very recently, the British Government Communications Headquarters (GCHQ) announced a quantum attack on their homemade Soliloquy scheme (CAMPBELL; GROVES;

SHEPHERD, 2014). This revelation has caused great discussion, as it allows for the recovery of the private key in polynomially-bounded time complexity. The issue is still surrounded by controversy and far from settled, but it is nevertheless a very important topic and must be taken into account when designing improvements to these methods.

1.2 Goals

The broad goal of this work is to layout efficient methods for lattice-based cryptography. Our focus is on improving the performance of constructions based on ideal lattices, such as the aforementioned GGH-style cryptosystems. Because efficiency in general is still one of the major hindrances to the development of these families of schemes, our goal is to allow not only for better processing times, but also for lower bandwidth occupation.

We target at generalizing existing methods, expanding their range of applications for contexts other than the ones they were originally designed for. In order to investigate the applicability of such methods in both traditional asymmetric cryptography and homomorphic encryption, we study the specific cases of the GGH-YK-M and LMSV constructions.

Also, it is crucial that these improvements do not affect negatively the security of said schemes. Moreover, increasing their security is also devised as a complimentary goal, taking into account the newly developed quantum key recovery attacks.

1.3 Methodology and Metrics

To measure the success of our method, our primary metrics are key sizes and processing times of all basic operations in asymmetric cryptosystems, namely, key generation, encryption and decryption.

The performance assessment of the proposed method is both theoretical and practical. Firstly, we estimate the complexity and amount of memory used by our algorithm, and then, we implement it. Our implementation is applied in two different cryptosystems, specifically, GGH-YK-M, representing a traditional public-key encryption scheme, and LMSV, which is thus far the only known instance of IND-CCA1 secure somewhat homomorphic encryption. Our results are compared to previous results available in the literature. When evaluating our results, we perform conversions whenever necessary to compensate for hardware differences.

1.4 Contributions

In this work, we extend the technique proposed by Gentry and Halevi, making it more comprehensive, and apply it to an traditional public-key encryption scheme and a homomorphic cryptosystem, specifically, the GGH-YK-M and LMSV schemes. We reduce public key bandwidth occupation of GGH-YK-M by an order of complexity, specifically, from $O(n^2 \lg n)$ down to $O(n \lg n)$ bits, where n is a public parameter of the scheme. The new technique also attains faster processing in all operations involved in an asymmetric cryptosystem, that is, key generation, encryption, and decryption. The most significant improvement in the performance of GGH-YK-M is in key generation, which becomes more than 3 orders of magnitude faster than previous results, while encryption becomes about 2 orders of magnitude faster. For decryption, our implementation is ten times faster than the literature.

In LMSV, our main focus is on increasing its security against the quantum attack devised by GCHQ. We suggest a different choice of polynomial ring, namely, an irreducible instance that yields nearly circulant matrices. This particular choice is able to successfully thwart the new attack and does not have an appreciable impact on the performance of encryption and decryption, if compared to usual ring choices. Key generation, however, is greatly affected, as there is no known dedicated technique and

it is necessary to resort to more generic algorithms. How to improve efficiency of key generation in this particular case remains as an open research problem, but our results show that it is possible to increase security of the scheme without great loss of performance in encryption and decryption.

In addition, we note that part of the work presented in Chapter 3 and some preliminary results have been published (BARGUIL; LINO; BARRETO, 2014).

While exploring applications of somewhat homomorphic encryption that might benefit from our new techniques, we briefly considered some e-cash protocols, in particular Sarkar's (SARKAR, 2013) because of its apparent reliance on ring operations. While that protocol turned out not to be a significant example of a scenario where somewhat homomorphic encryption would be useful *per se*, as a side contribution of our analysis we were able to entirely break it, undermining all of its security goals. These results have also been published (BARGUIL; BARRETO, 2015), and are summarized in Appendix A.

1.5 Outline

This thesis is organized as follows. Chapter 2 provides the theoretical background that supports lattice-based encryption, as well as a discussion of the state of the art of research in this field and the definition of the GGH-YK-M and LMSV cryptosystems. Chapter 3 describes our original contributions, namely, the improved method for generating cryptographic keys for GGH-style schemes. The security of the improved method is discussed in Chapter 4, including an analysis of how the new quantum attack can be prevented. Obtained results are shown in Chapter 5. A final analysis and considerations about future work possibilities are presented in Chapter 6. As an additional contribution, we describe Sarkar's e-cash protocol and its cryptanalysis in Appendix A.

2 LITERATURE REVIEW

This chapter contains an overview of related work used as a base for the achieved results, as well as an introduction to essential theory.

Section 2.1 introduces some fundamental notions of cryptography, defining cryptosystems and discussing some possibilities for their utilization. Section 2.2 provides the theoretical background for lattices, defining what lattices are and presenting some well-known hard problems. In Section 2.3, a review of related lattice-based encryption is presented, examining some of the many uses of lattices in cryptography. Section 2.4 defines the notation used in the next chapters and introduces further theoretical concepts. Finally, Sections 2.5 and 2.6 outline the GGH-YK-M and LMSV cryptosystems, on which we later apply our contribution.

2.1 Cryptography fundamentals

Before the introduction of the theoretical background that supports lattice-based cryptography, it is important to present some basic terminology and definitions, addressing the different kinds of existing cryptosystems and a few of their possible applications. Readers who are already familiar with basic cryptography concepts are invited to skip directly to Section 2.2.

2.1.1 Basic concepts

Take the message *"Dobro jutro!"* as an initial example. It is a piece of data, represented by a sequence of letters, and, except for people who speak a few south slavic[1] languages, it is absolutely meaningless. But for southern slavs, it conveys a very definite message: it is a morning greeting. In different contexts, the same greeting could be represented by different words. For instance, in a Finnish sauna[2] the best way to transmit the same message would probably be *"Hyvää huomenta!"*. The morning greeting, therefore, is information. The letters and symbols are data, and can assume different forms depending on the context.

If we apply some sort of transformation, altering our data to be *"ma4+FOIaxi7ki"*, it becomes an apparently meaningless message. It does not correspond to anything in any natural language, even though it is still somehow related to our original message. If we are able to reverse the process, we can recover the first message (data), and from that, the original information. If our attackers are not able to revert the transformation, data can be exchanged through an *insecure channel*, that is, a channel where it can be intercepted by others (like the Internet).

Data from which information can be directly extracted is called *plaintext*, whereas the result of some special sort of cryptographic transformation is named *ciphertext*. The information contained in a ciphertext is protected, and can only be recovered by reversing the transformation and reconstructing the corresponding plaintext. The process of transforming plaintext in ciphertext is called *encryption*, and the reverse process is known as *decryption*. A diagram depicting these operations can be seen in Figure 1.

Modern cryptography is used for not only providing confidentiality, but also for guaranteeing integrity, authentication and non-repudiation. In Figure 2, Alice is send-

[1] Also known as Yugoslavian. The former country of Yugoslavia, or *"Jugoslavija"*, was named after the south slavic peoples, as *"jugo"* means *"south"* (RADOŠ, 2003).

[2] The word *"sauna"* comes from the Finnish term *"sauna"*, which means, not surprisingly, sauna (HÄKKINEN, 2004).

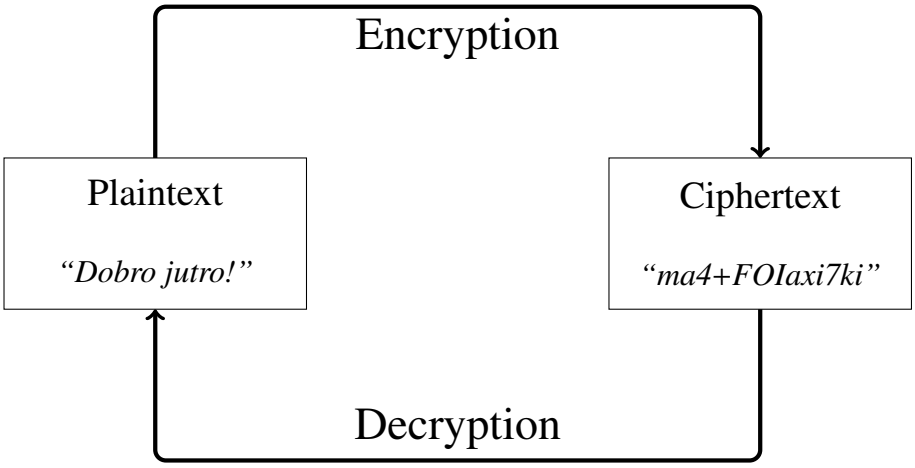

Figure 1: Encryption and decryption.

ing a message m to Bob through an insecure channel. Eve is eavesdropping on their communication, being able to not only passively read data, but also to actively modify it (which means she can be either a *passive* or an *active* attacker). In this scenario, the following aspects can be defined:

- **Confidentiality**: Only Bob is able to extract information from any message m, even though Eve has access to all exchanged data.
- **Integrity**: Bob is able to detect when any of Alice's messages has been modified, either because of an aggression by Eve or due to some network error.
- **Authentication**: Upon reception of a message that was supposedly sent by Alice, Bob can obtain proof to ensure that the other party is, indeed, who they claim to be.
- **Non-repudiation**: Alice is not able to repudiate the authenticity of any message sent in the past, that is, she cannot deny being the author of any of the messages she actually sent.

Modern cryptography is based on special functions, called *one-way trapdoor functions*. If we define a function $f : A \rightarrow B$ and its inverse f^{-1}, f is said to be a trapdoor

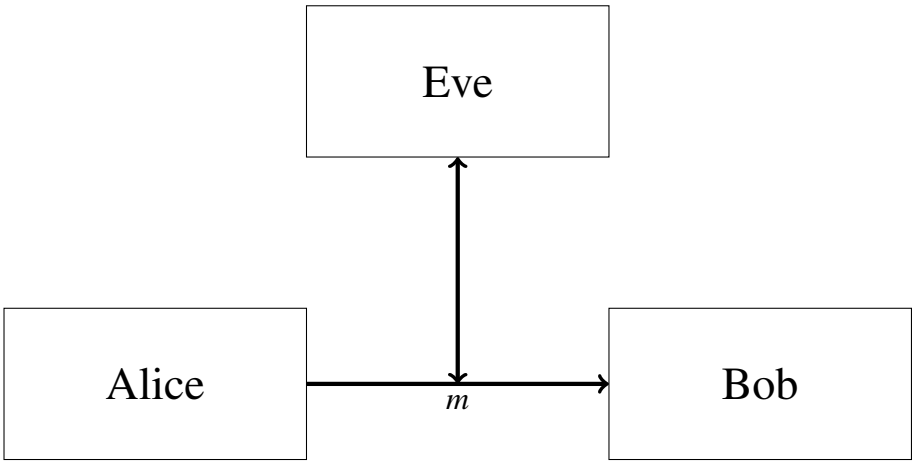

Figure 2: Communication through an insecure channel.

function if $f(a)$ is easy to compute for any $a \in A$ and $f^{-1}(b)$ is infeasible to calculate for most $b \in B$, but $f^{-1}(b)$, $b \in B$, is easy to determine given certain additional information (SHANNON, 1949; VAN TILBORG, 2000). This additional information is kept secret, being dubbed the *key*. Cryptographic keys can be of many different formats (e.g., numbers, vectors, codes), depending on what kind of trapdoor function is used.

To illustrate this concept, take the discrete logarithm problem (ADLEMAN, 1979). This problem consists on solving the equation $a^x = b$ on x, where a and b are elements of a finite group. Let $\mathbb{Z}_p$ be the integer multiplicative group modulo p, and take $\mathbb{Z}_{11}$ as an example (i.e., the integers from 0 through 10). Calculating $b = 2^8 \bmod 11 = 256 \bmod 11 = 3$ is only a matter of multiplication and modulo operations. But the inverse, that is, calculating x in $2^x = 3 \bmod 11$ is not trivial. In a small group like this, one can test all possibilities, but if we choose a large enough group, brute-forcing becomes infeasible.

Table 1: Powers of 2 mod 11

n	0	1	2	3	4	5	6	7	8	9	10
2^n	1	2	4	8	16	32	64	128	256	512	1024
$2^n \bmod 11$	1	2	4	8	5	10	9	7	3	6	1

Modular exponentiation is not the only known instance of the discrete logarithm problem. It can be generalized to any multiplicative group, such as elliptic

curves (SMART, 1999). There are numerous families of cryptosystems based on different kinds of problems, such as integer factorization (RIVEST; SHAMIR; ADLEMAN, 1978), coding theory (MCELIECE, 1978), lattices (which will be discussed later in this chapter), and many others.

2.1.2 Symmetric and asymmetric cryptosystems

A valid cryptosystem is a set of definitions, describing:

- A set of possible symbols contained in a plaintext, called the alphabet $\mathcal{A}$. A plaintext m is the concatenation of an arbitrary number of symbols in $\mathcal{A}$, that is, $m \in \mathcal{A}^*$.
- A second alphabet $\mathcal{B}$, so that $c \in \mathcal{B}^*$ for any ciphertext c. Frequently, $\mathcal{A} = \mathcal{B}$.
- A keyspace $\mathcal{K}$, which contains all possible keys.
- An encryption function $E_k(x)$ which receives as input a key $k \in \mathcal{K}$ and $x \in \mathcal{A}^*$, outputting $y \in \mathcal{B}^*$.
- A decryption function $D_k(y)$ which receives as input a key $k \in \mathcal{K}$ and $y \in \mathcal{B}^*$, outputting $x \in \mathcal{A}^*$.

For any cryptosystem, the property $D_{k_2}(E_{k_1}(x)) = x$ must hold for every valid combination of (k_1, k_2) and for any $x \in \mathcal{A}^*$ (VAN TILBORG, 2000).

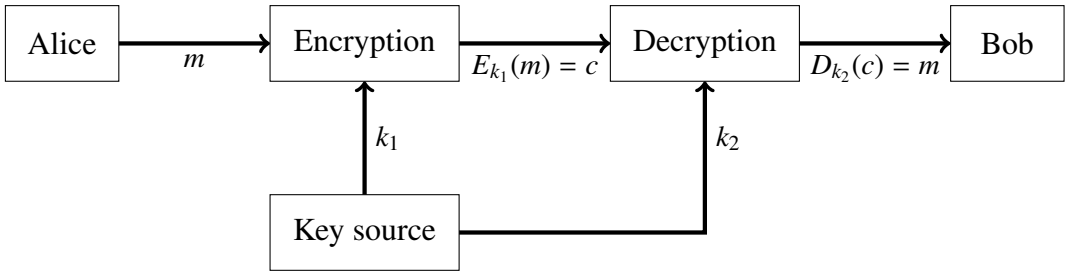

Figure 3: A cryptosystem. Modified from (VAN TILBORG, 2000).

A cryptosystem, as defined by Shannon (SHANNON, 1949), can be seen in Figure 3. In this scenario, Alice encrypts a message m using key k_1, resulting in a ciphertext $c = E_{k_1}(m)$. Only then it is sent forward through an insecure communication channel. After receiving the ciphertext, Bob decrypts it to recover the original message $m = D_{k_2}(c)$.

Cases where $k_1 = k_2$ are named *symmetric*, and cases where $k_1 \neq k_2$ are called *asymmetric*. In asymmetric cryptography, both keys are generated at the same time, forming a pair (k_1, k_2). In most cases one of them is openly published (the *public* key), while the other is kept secret (the *private* or *secret* key). For this reason, another frequent denomination for this kind of scheme is *public-key cryptosystem*.

In general, symmetric schemes are faster than asymmetric ones, but it is infeasible to use only symmetric cryptography for communication. Each participant in a network would have to settle a shared key with every other node, resulting in an infrastructure for key distribution that would be increasingly hard to maintain. Fortunately, key-agreement primitives based on asymmetric schemes can be used to distribute keys in a flexible way. An example is the Diffie-Hellman method (DIFFIE; HELLMAN, 1976), which can be used to establish a shared symmetric key between two parties. This shared key is then used with some symmetric cipher to protect exchanged messages, keeping computational overhead to a minimum. Symmetric cryptography can also be used when there is no communication involved, for instance, for protecting data in permanent storage media, such as a hard-drive.

Provided that the secret key is kept safe from attackers, public-key cryptography can be used for many purposes. The Secure Shell (SSH) (YLONEN; LONVICK, 2006) protocol uses this kind of cryptosystem for authentication and key exchange. Further than that, digital signature schemes can also provide integrity, authenticity and non-repudiation. Asymmetric methods can be used for digitally signing messages and documents, and also for creating digital certificates. The Transport Layer Security (TLS) (DIERKS; RESCORLA, 2008) protocol relies on certificates for verifying authen-

ticity of servers. These must be signed by trusted Certificate Authorities (CAs), which are responsible for validating the identity of the websites for which they provide a digital certificate (DURUMERIC et al., 2013). When a modern web browser is used to access *www.alice.com* and a certificate claiming to be from Alice is received, its signature is verified. If it is signed by a trusted CA, the web browser accepts it. Otherwise, the web browser issues a warning to the user.

Two examples of well-known symmetric schemes are the Data Encryption Standard (DES) (STANDARDS, 1977), developed in the 1970s and later broken (BIHAM; SHAMIR, 1991), and its successor, the Advanced Encryption Standard (AES) (DAEMEN; RIJMEN, 1998). The famous RSA cryptosystem (RIVEST; SHAMIR; ADLEMAN, 1978), named after its creators, can be highlighted as an instance of asymmetric scheme. Two signature schemes are the Digital Signature Algorithm (DSA) (STANDARDS, 1994) and the one proposed by ElGamal (ELGAMAL, 1985).

2.2 Lattices

A lattice in $\mathbb{R}^n$ is a discrete subgroup of the n-dimensional vector space V defined in $\mathbb{R}^n$, generated from any basis for $\mathbb{R}^n$, taking all linear combinations with integer coefficients of this same basis. Therefore, a lattice $\mathcal{L}$ in $\mathbb{R}^n$ has the form

$$\mathcal{L} = \{v \in V;\ v = a_1 v_1 + \cdots + a_n v_n,\ a_i \in \mathbb{Z},\ 1 \leq i \leq n\}$$

where $\{v_1, \ldots, v_n\}$ is a basis for $V = \mathbb{R}^n$. Figure 4 shows an example of a two-dimensional lattice defined in $\mathbb{R}^2$. A simple example of a lattice in $\mathbb{R}^n$ is $\mathbb{Z}^n$ itself. For any given lattice, there is an infinite number of bases that can generate it. An orthogonal basis composed only by vectors of magnitude 1 is named *orthonormal*.

Definition 1. *(Euclidean norm). The Euclidean norm of a vector is defined as* $\| v \| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$ *where* $v = v_1 w_1 + v_2 w_2 + \cdots + v_n w_n$ *and* $\{w_1, \ldots, w_n\}$ *is an orthonormal basis for* $\mathbb{R}^n$.

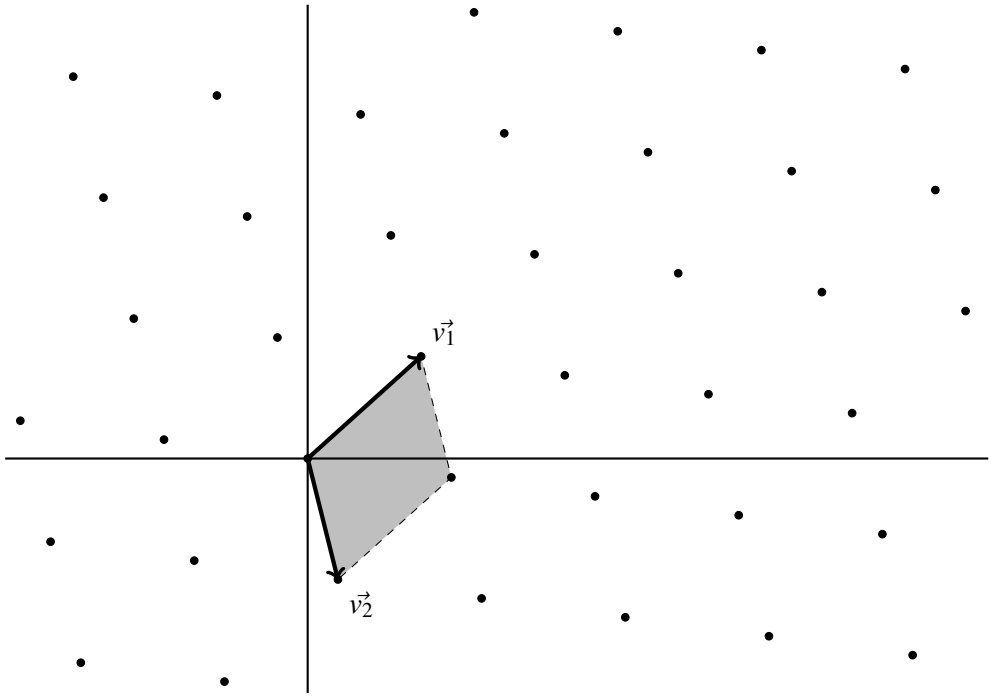

Figure 4: A two-dimensional lattice in $\mathbb{R}^2$ with a possible basis.

Based on the definition of Euclidean norm, it is possible to specify notions of "short" and "long" vectors. The *lattice reduction problem* can be defined as follows.

Definition 2. *(Lattice Reduction Problem). Given any basis B for a lattice $\mathcal{L}$, find an equivalent basis for $\mathcal{L}$ with short, approximately orthogonal vectors.*

Let B be a basis for lattice $\mathcal{L}$ in $\mathbb{R}^n$. It is then possible to define

$$\delta(B) = \frac{\prod_{i=1}^{n} \|b_i\|}{|det(B)|}$$

called *orthogonality defect*. As a result of Hadamard's inequality (GARLING, 2007), we have that for any B, $\delta(B) \geqslant 1$, and $\delta(B) = 1$ if and only if B is an orthogonal basis for $\mathbb{R}^n$. A basis with a small orthogonality defect is considered to be a *good* basis, whereas a basis with a high value for its orthogonality defect is considered *bad*. The smaller the value of $\delta(B)$, the closer B is to being orthogonal.

Given a good basis, it is relatively easy to find a bad one, but the opposite does not necessarily happen. The problem of finding the basis with the smallest defect possible

allows for the construction of trapdoor functions suitable enough for the definition of cryptosystems.

In truth, there are polynomial time solutions to this problem when some restrictions are applied. Algorithms such as the one proposed by Babai (BABAI, 1986), LLL (LENSTRA; LENSTRA; LOVÁSZ, 1982) and BKZ (SCHNORR, 1987) are able to solve in polynomial time the problem of finding a basis B such that $\delta(B) \leq \gamma(n)$, to an arbitrary factor $\gamma(n)$ where n is the lattice dimension. But these algorithms depend heavily on the orthogonality defect of the input basis and on the value of n. They may not yield suitable solutions in larger dimensions, specially when receiving bad bases as input. Because of this, the lattice reduction problem is deemed hard in arbitrarily large dimensions.

Two other hard problems on lattices are the *Shortest Vector Problem* (SVP) and *Closest Vector Problem* (CVP), which can be defined as follows.

Definition 3. *(Shortest Vector Problem (SVP)). Given a basis for a lattice $\mathcal{L}$, find the shortest vector $v \neq 0$ in $\mathcal{L}$.*

Definition 4. *(Closest Vector Problem (CVP)). Given a basis for a lattice $\mathcal{L}$ and a vector u not necessarily in $\mathcal{L}$, find the vector $v \in \mathcal{L}$ closest to u.*

Both of these problems are deemed hard to solve for the Euclidean norm and suitably chosen $\gamma(n)$, as CVP can be reduced to SVP, and SVP, by its turn, can be reduced to the lattice reduction problem. Known algorithms for solving the CVP work well with short lattice bases, but not with long bases.

There are numerous other problems on lattices, such as the Shortest Independent Vectors Problem (SIVP), but most of them have the same behavior as SVP. Roughly explained, they are also based on finding a short approximation of a vector, for some definition of *"short"*.

While the subject of problems on lattices is interesting, it is a large topic and a

deeper analysis falls outside of the scope of this work. The book by Micciancio and Goldwasser (MICCIANCIO; GOLDWASSER, 2002) is indicated for further reading, where discussions about the complexity of these and other lattice problems can be found.

2.3 Lattice-based cryptography

In Cryptology, lattices have been used in different applications. Following the development of algorithms for approximations of the lattice reduction problem, they were used to conceive polynomial time solutions to many classical problems in computer science, such as factoring polynomials over the rationals, solving integer programs in a fixed number of variables, and other cryptanalysis problems (MICCIANCIO; GOLDWASSER, 2002). These uses resulted in the effective destruction of many known cryptosystems. After the discovery of a connection between the worst-case and average-case hardness of certain lattice problems (AJTAI, 1996), lattices have also been employed in *constructive* applications, supporting the creation of many new cryptosystems (NGUYEN; STERN, 2001).

Traditional schemes based on the assumed hardness of integer factorization or discrete logarithm computation can be successfully attacked with the help of quantum computers (SHOR, 1995). As a consequence, many of the currently most used algorithms, such as RSA (RIVEST; SHAMIR; ADLEMAN, 1978) and ECDSA (JOHNSON; MENEZES; VANSTONE, 2001), would be as good as broken once quantum computers become technologically feasible. On account of this concern, the scientific community started the pursuit for new schemes based on different computational problems, leading to the development of purely classical, but believed to be quantum-resistant constructions known as *post-quantum* cryptosystems (BERNSTEIN; BUCHMANN; DAHMEN, 2008). One of the most promising current trends in post-quantum cryptography is the family of schemes based on lattices. Lattices not only seem to yield quantum-resistant methods, but are also remarkably flexible. They have been employed in many

different contexts, as we will now expose.

2.3.1 Asymmetric schemes

Three families of schemes can be highlighted as examples of lattice-based public-key cryptosystems: the pioneering technique proposed by Goldreich, Goldwasser and Halevi (GGH) (GOLDREICH; GOLDWASSER; HALEVI, 1997), the group of schemes based on the Learning With Errors (LWE) problem (REGEV, 2005), and the NTRU cryptosystem (HOFFSTEIN; PIPHER; SILVERMAN, 1998). Both LWE and NTRU resort to certain rings of structured matrices, namely, circulant and negacyclic matrices (both concepts are further explained in Section 3.1), while GGH is based on classic lattices.

Considering that our contribution is aimed at GGH-style lattices, a brief description of this cryptosystem is due. In this scheme, which can be seen as a generalization of the McEliece scheme (MCELIECE, 1978), a good lattice basis B is used as the private key, represented in its matrix form. By submitting B to the process of calculating its corresponding Hermite normal form (HNF) (COHEN, 1993, section 2.4.2), the result is an equivalent basis H for the same lattice, but with rather large vectors (i.e., a *bad* basis). The public key is defined as H. For encryption, a message m is translated to a vector in the lattice, and then a small error $\vec{e}$ is added. Therefore, the cryptogram is $\vec{c} = H \cdot \vec{m} + \vec{e}$. For decryption, the message is recovered by solving the CVP, thanks to the good lattice basis B. This yields the lattice vector $H \cdot \vec{m}$, hence, the original message m. This process can be seen in Figure 5.

Two years after the publication of GGH, Nguyen proved that the original method had inherent structural flaws (NGUYEN, 1999) and was able to break typical, realistic GGH instances by using lattice reduction algorithms. The scheme was considered irreversibly broken for a long time, to the extent that other kinds of lattices stemming from the LWE problem have essentially dominated the research in the area. This situation began to change when Yoshino and Kunihiro (YOSHINO; KUNIHIRO, 2012) described a

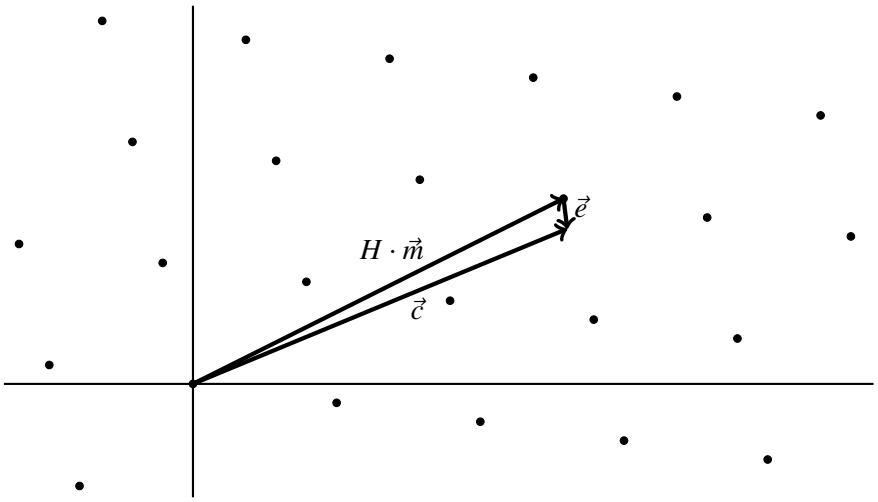

Figure 5: The GGH scheme.

variant of GGH that thwarts all known attacks. Aptly called GGH-YK, their scheme was, however, incomplete in the sense that, by blindly following their prescriptions, no proper parameter set can be feasibly constructed. This means it is not implementable in a secure and efficient way, and, therefore, it is not practical.

Recently, Barros and Schechter (BARROS; SCHECHTER, 2014) revisited the GGH-YK construction, and proposed a surprising modification of that scheme that adequately provides a suitable parametrization. This new scheme is named GGH-YK-M due to the fact that it makes essential use of M-matrices (BERMAN; PLEMMONS, 1994). The result is greatly promising, as it brings the simplicity of GGH and GGH-YK back to life. This last variant is discussed in detail in Section 2.5.

2.3.2 Homomorphic encryption

Homomorphism as a research topic was originally proposed by Rivest, Adleman and Dertouzos (RIVEST; ADLEMAN; DERTOUZOS, 1978). In their work, they conjectured whether it was possible to construct a cryptosystem that allowed for arbitrary computation over encrypted data, while still yielding valid results after decryption. They proposed private data banks as an application of such a system. Users can store

their encrypted data in an untrusted server and, later, send queries to it. The server is able to calculate the requested job and return an encrypted result that, upon decryption, outputs a correct plaintext, the same as if the operation had been done without involving encryption at all. Since then, many other applications for fully homomorphic schemes have been devised, such as systems which must deal with sensitive data, like medical and financial applications (NAEHRIG; LAUTER; VAIKUNTANATHAN, 2011). However, for a long time, no fully homomorphic scheme was known.

A cryptosystem is said to be homomorphic if, given the ciphertexts $\psi_1, \ldots, \psi_t$ corresponding to the plaintexts $m_1, \ldots, m_t$, anyone (not just the key-holder) is capable to perform computations over encrypted data, yielding results that are equivalent to a valid encryption of $f(m_1, \ldots, m_t)$, for a given function f, without leaking any information about $m_1, \ldots, m_t$ or $f(m_1, \ldots, m_t)$ or intermediate values (GENTRY, 2009a). Cryptosystems that support any function f, as long as f is computable, are said to be *fully* homomorphic, whereas schemes that support only a finite set of functions are called *somewhat* or *leveled* homomorphic schemes.

Until recently, only instances of leveled homomorphic schemes had been defined. One example is RSA (RIVEST; SHAMIR; ADLEMAN, 1978). In this scheme, encryption is defined as $E(m) = m^e \bmod n$, for public parameters e and n. Therefore, $E(m_1) \cdot E(m_2) = m_1^e m_2^e \bmod n = (m_1 m_2)^e \bmod n = E(m_1 \cdot m_2)$, which means that RSA is homomorphic for multiplication. But operations on ciphertexts and plaintexts do not have to be the same for a system to be considered homomorphic. An example of such scenario is the cryptosystem proposed by Paillier (PAILLIER, 1999), in which, for the encryption and decryption functions E and D, we have that $D(E(m_1) \cdot E(m_2)) = m_1 + m_2$ and $D(E(m_1)^{m_2}) = m_1 \cdot m_2$.

In 2009, a major breakthrough was achieved after the first fully homomorphic scheme was proposed by Gentry (GENTRY, 2009a). Based on GGH-style lattices, the scheme outputs a ciphertext with a small noise parameter with maximum value η. De-

cryption works as long as the ciphertext noise is less than $N \gg \eta$. Computations over ciphertexts can be performed at the cost of increasing the noise parameters. To roughly illustrate, we could consider that the result of the multiplication of two ciphertexts with noise $O(\eta)$ would produce noise of order $O(\eta^2)$. The computation of arbitrarily large circuits, then, would result in equally large noises of order $O(N)$, which would render correct decryption impossible. To tackle this issue, a Recrypt algorithm is defined. It takes a ciphertext $E(a)$ as input and outputs another ciphertext that also encrypts a, but with maximum noise of order $\sqrt{N}$. The scheme is then able to support the computation of arbitrary circuits by recrypting the ciphertext whenever necessary, and thus, keeping its noise within the necessary boundaries for correct decryption. Furthermore, this strategy can also be used to, given some restrictions, transform a somewhat homomorphic scheme into a fully homomorphic one.

Gentry's work proved that fully homomorphic encryption is theoretically possible. However, it yields prohibitively large ciphertexts and keys. It demands high processing times for operations as well, and for these reasons, the scheme is not practical. Other proposals have been made, trying to evade this problem (STEHLÉ; STEINFELD, 2010; LOFTUS et al., 2012). In particular, the technique devised by Smart and Vercauteren (SMART; VERCAUTEREN, 2010) and later improved by Gentry and Halevi (GENTRY; HALEVI, 2011), aimed at homomorphic encryption, seems to be also applicable in other lattice-based cryptosystems. This is discussed in Chapter 3.

In truth, there are other homomorphic schemes that do not stem from Gentry's original scheme, based on other kinds of lattices. For instance, methods based on the LWE problem, such as (NAEHRIG; LAUTER; VAIKUNTANATHAN, 2011; BRAKERSKI; GENTRY; VAIKUNTANATHAN, 2012; BRAKERSKI, 2012; GENTRY; SAHAI; WATERS, 2013; BRAKERSKI; VAIKUNTANATHAN, 2014). There also exist methods based on NTRU, such as the aforementioned scheme by Stehle and Steinfeld and also (LÓPEZ-ALT; TROMER; VAIKUNTANATHAN, 2012; BOS et al., 2013).

However, almost all of these schemes have succumbed to cryptanalysis. Loftus *et al.* (LOFTUS et al., 2012), Szydlo (SZYDLO, 2003), Chenal and Tang (CHENAL; TANG, 2014; CHENAL; TANG, 2015) and Dahab, Galbraith and Morais (DAHAB; GALBRAITH; MORAIS, 2015) described key recovery attacks against many of them. The only scheme, thus far, that seems to resist such attacks is the somewhat homomorphic scheme of Loftus *et al.* (LOFTUS et al., 2012), detailed in Section 2.6. A more thorough discussion of the security of existing homomorphic schemes is presented in Section 4.2.

2.3.3 Other applications of lattices in cryptography

Some signature schemes based on lattices exist, such as the one proposed by Lyubashevsky (LYUBASHEVSKY, 2012) and improved by Güneysu *et al.* (GÜNEYSU; LYUBASHEVSKY; PÖPPELMANN, 2012). The already mentioned NTRU cryptosystem can be modified to construct a signature scheme (HOFFSTEIN; PIPHER; SILVERMAN, 2001; HOFFSTEIN et al., 2003). However, it is not secure, as key recovery attacks have been described (SZYDLO, 2003; NGUYEN; REGEV, 2006). Rückert has developed a blind signature scheme (RÜCKERT, 2010) based on lattices, that is, a signature where the signer does not have access to the document that is being signed. This scheme is quite inefficient, and its security requires that the underlying problems remain intractable for rather coarse approximation factors (RICARDINI, 2014).

Lattices have also been used in many other applications, such as identity-based encryption (GENTRY; PEIKERT; VAIKUNTANATHAN, 2008), oblivious transfer (PEIKERT; VAIKUNTANATHAN; WATERS, 2008), and zero-knowledge proofs (MICCIANCIO; VADHAN, 2003).

2.4 Further concepts and notation

In this section we further introduce basic concepts and define notation used throughout this document.

Vector and matrix indices are numbered starting from 0. We denote by $M_{(i)}$ the i-th row of a matrix M, and by M_j the j-th element on its first row, i.e., $M_j := M_{(0),j}$. We also denote by $x \xleftarrow{\$} U$ the uniformly random sampling of variable x from set U. A vector $v \in \mathbb{R}^n$ with n zeros is denoted by 0^n.

Definition 5. *(BOLDRINI et al., 1980, Definition 1.2.1)* **(Square matrix).** *A* square matrix *is a matrix with the same number of rows and columns.*

Examples:

$$\begin{bmatrix} 5 & 8 & 1 \\ 1 & 0 & -4 \\ 6 & 5 & 2 \end{bmatrix} \text{ and } \begin{bmatrix} 3 & 7 \\ -1 & 9 \end{bmatrix}.$$

Definition 6. *(BOLDRINI et al., 1980, Definition 1.2.5)* **(Diagonal matrix).** *A* diagonal matrix *is a square matrix where all elements not in the diagonal are zero, that is, $a_{ij} = 0$, for $i \neq j$.*

Examples:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & -7 \end{bmatrix} \text{ and } \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 8 \end{bmatrix}.$$

Definition 7. *(BOLDRINI et al., 1980, Definition 1.2.6)* **(Identity matrix).** *An* identity

matrix *is a diagonal matrix where all elements in the diagonal are* 1*, that is,* $a_{ii} = 1$ *and* $a_{ij} = 0$*, for* $i \neq j$*. The identity matrix of dimension n is denoted by* I_n*.*

Examples:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

For any square matrix M, we have that $MI = M$.

Definition 8. *(BOLDRINI et al., 1980, Definition 1.2.7)* **(Upper triangular matrix).** *An* upper triangular matrix *is a square matrix where all elements below the diagonal are zero, that is,* $a_{ij} = 0$*, for* $i > j$*.*

Examples:

$$\begin{bmatrix} 2 & -1 & 0 \\ 0 & 3 & \sqrt{5} \\ 0 & 0 & 7 \end{bmatrix} \text{ and } \begin{bmatrix} a & b \\ 0 & c \end{bmatrix}$$

Definition 9. *(HORN; JOHNSON, 2012, Section 0.9.6)* **(Circulant matrix).** *A* circulant matrix *is an* $n \times n$ *matrix of the form*

$$M = \begin{bmatrix} a_1 & a_2 & a_3 & \dots & a_{n-1} & a_n \\ a_n & a_1 & a_2 & \dots & a_{n-2} & a_{n-1} \\ a_{n-1} & a_n & a_1 & \dots & a_{n-3} & a_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_2 & a_3 & a_4 & \dots & a_n & a_1 \end{bmatrix},$$

that is, $M_{i,j} = M_0[i + j \bmod n]$*, where* M_0 *is the first row of the matrix and* $M_0[j]$ *is the j-th element of* M_0*.*

Another way of understanding this concept is to think of a square matrix where each row is the previous row cycled forward one step.

Definition 10. (Negacyclic matrix). *A* negacyclic matrix *is of the form*

$$M = \begin{bmatrix} a_1 & a_2 & a_3 & \dots & a_{n-1} & a_n \\ -a_n & a_1 & a_2 & \dots & a_{n-2} & a_{n-1} \\ -a_{n-1} & -a_n & a_1 & \dots & a_{n-3} & a_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -a_2 & -a_3 & -a_4 & \dots & -a_n & a_1 \end{bmatrix},$$

that is, $M_{i,j} = M_0[i + j \bmod n] \cdot (-1)^{i+j \; div \; n}$*, where* M_0 *is the first row of the matrix,* $M_0[j]$ *is the j-th element of* M_0 *and a div b is the result of the integer division between a and b.*

Negacyclic matrices are very similar to circulant ones, except that elements returning to the first position are multiplied by -1.

Definition 11. *(BOLDRINI et al., 1980, Definition 6.1.2)* **(Eigenvalue and eigenvector).** *Define a vector space* V *and a linear operator* $T : V \to V$*. If there exist a vector* $v \in V, v \neq 0$ *and* $\mu \in \mathbb{R}$ *such that* $Tv = \mu v$*,* μ *is an* eigenvalue *of* T *and* v *is an* eigenvector *of* T *associated to* μ*.*

Definition 12. (Spectral radius). *Define* $P \in \mathbb{C}^{n \times n}$*. The* spectral radius *of P is defined as* $\rho(P) := \max\{|\lambda| : \lambda \text{ is an eigenvalue of } P\}$.

Definition 13. *(BERMAN; PLEMMONS, 1994, Definition 1.2)* **(*M*-Matrix).** *Define* $P \in \mathbb{Z}^{n \times n}$ *such that* $P_{ij} \leqslant 0$ *for all* $0 \leqslant i, j < n$*. A (nonsingular)* M-matrix *is a matrix of form* $A = \gamma I + P$ *for some* $\gamma > \rho(P)$*, where I is the Identity matrix.*

Definition 14. *(HORN; JOHNSON, 2012, Section 0.9.11)* **(Vandermonde Matrix).** *A* Vandermonde matrix *is a matrix built from a vector* $\{x_0, \ldots, x_{n-1}\} \in \mathbb{R}^n$ *with the form*

$$V = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ x_0 & x_1 & x_2 & \ldots & x_{n-1} \\ x_0^2 & x_1^2 & x_2^2 & \ldots & x_{n-1}^2 \\ x_0^3 & x_1^3 & x_2^3 & \ldots & x_{n-1}^3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_0^{n-1} & x_1^{n-1} & x_2^{n-1} & \ldots & x_{n-1}^{n-1} \end{bmatrix},$$

that is, where $V_{ij} := x_j^i$.

Definition 15. *(COHEN, 1993, Section 2.4.2)* **(Hermite normal form (HNF)).** *A matrix* $H \in \mathbb{Z}^{n \times n}$ *is said to be in* Hermite normal form *(HNF) if it is upper triangular, all its elements are non-negative and the entries on the diagonal are positive and are the largest entries in their respective columns.*

Examples:

$$\begin{bmatrix} 10 & 2 & 0 & 1 \\ 0 & 5 & 0 & 6 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 7 \end{bmatrix} \text{ and } \begin{bmatrix} 3 & 1 & 0 \\ 0 & 2 & 5 \\ 0 & 0 & 8 \end{bmatrix}.$$

Definition 16. (Minimal Hermite normal form). *A matrix* $H \in \mathbb{Z}^{n \times n}$ *in HNF is said*

to be minimal *if it has the form*

$$H = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 & v_0 \\ 0 & 1 & 0 & \dots & 0 & 0 & v_1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 & v_{n-3} \\ 0 & 0 & 0 & \dots & 0 & 1 & v_{n-2} \\ 0 & 0 & 0 & \dots & 0 & 0 & d \end{bmatrix},$$

which can also be written as

$$H = \left[\begin{array}{c|c} I_{n-1} & v^T \\ \hline 0^{n-1} & d \end{array}\right],$$

where I_{n-1} is the Identity matrix with dimension $n-1$, $v \in \mathbb{Z}^{n-1}$ and $d \in \mathbb{Z}$.

One can check by direct inspection that the inverse (over $\mathbb{Q}$) of a matrix H in minimal HNF is

$$H^{-1} = \left[\begin{array}{c|c} I_{n-1} & -(1/d)v^T \\ \hline 0^{n-1} & 1/d \end{array}\right].$$

Thus a matrix H in minimal HNF can be conveniently represented by $(v, d) \in \mathbb{Z}^n$ alone. Also, it is clear that $\det(H) = d$.

Definition 17. *(APOSTOL, 1970)* **(Resultant of two polynomials).** *Given two polynomials A and B, say*

$$A(x) = \sum_{k=0}^{n} a_k \cdot x^k \text{ and } B(x) = \sum_{k=0}^{m} b_k \cdot x^k,$$

and defined a square matrix M of dimensions $n + m$ as

$$M = \begin{bmatrix} a_{n-1} & a_{n-2} & a_{n-3} & \dots & a_1 & a_0 & & & \\ & a_{n-1} & a_{n-2} & \dots & a_2 & a_1 & a_0 & & \\ & & & \dots & & & & & \\ & & & & a_{n-1} & a_{n-2} & \dots & a_1 & a_0 \\ b_{m-1} & b_{m-2} & b_{m-3} & \dots & b_1 & b_0 & & & \\ & b_{m-1} & b_{m-2} & \dots & b_2 & b_1 & b_0 & & \\ & & & \dots & & & & & \\ & & & & b_{m-1} & b_{m-2} & \dots & a_1 & a_0 \end{bmatrix},$$

the remaining entries being equal to zero, the resultant of A and B is defined as $\mathsf{Resultant}(A, B) = det(M)$*, where $det(M)$ is the determinant of M.*

Calculating the determinant, however, is too inefficient for our purposes. If the coefficients of the polynomials are in a field, it is possible to resort to the following algorithm:

```
1:  function Resultant(F, G)
2:      res ← 1
3:      while Degree(G) > 0 do
4:          T ← F mod G
5:          res ← res · (−1)^(Degree(F)·Degree(G)) · LeadingCoefficient(G)^(Degree(F)−Degree(T))
6:          F ← G
7:          G ← T
8:      end while
9:      if F ≠ 0 then
10:         return G · res
11:     else
12:         return 0
13:     end if
14: end function
```

2.5 The GGH-YK-M scheme

In this section, we summarize the GGH variant recently proposed by Barros and Schechter (BARROS; SCHECHTER, 2014). As previously discussed in Subsection 2.3.1, the GGH cryptosystem (GOLDREICH; GOLDWASSER; HALEVI, 1997) had been considered broken beyond repairs after the publication of the cryptanalysis by Nguyen (NGUYEN, 1999). More recently, Yoshino and Kunihiro (YOSHINO; KUNIHIRO, 2012) developed a new method, dubbed GGH-YK, that effectively avoided all known attacks, but did not support proper parametrization. The proposition by Barros an Schechter was called GGH-YK-M by virtue of resorting to M-matrices (BERMAN; PLEMMONS, 1994) to complete its specification.

2.5.1 Cryptosystem definition

For simplicity and efficiency, in our description of GGH-YK-M we explicitly require that the private lattice basis A be such that its HNF is minimal.

Let n be an integer (usually, but not necessarily, a power of 2), let γ be a multiple of n by some small factor (i.e., $\gamma = \alpha n$ for some small integer α), let σ be an even integer, and let h and k be integers such that $h + k < \gamma < 2h$. The GGH-YK-M encryption scheme (BARROS; SCHECHTER, 2014) consists of the following three algorithms:

1. Keygen: Sample $P \xleftarrow{\$} \{-1, 0\}^{n \times n}$, compute $A \leftarrow \gamma I + P$ and its HNF $H := \mathrm{HNF}(A)$ until $\rho(P) < \gamma$, $1/\gamma < \left|(A^{-1})_{ii}\right| \leqslant 2/\gamma$ for $0 \leqslant i < n$, $\left|(A^{-1})_{ij}\right| < 2/\gamma^2$ for $i \neq j$, and H is in minimal form. Empirically, taking α in the definition $\gamma = \alpha n$ to be as small as 2 is usually enough to ensure that these conditions hold with high probability. The private key is A, and the public key is H, which can be represented by its last column $(v, d) \in \mathbb{Z}^n$ (see Definition 16). Since $v_i < d$ from the definition of the HNF (Definition 15), and $d = O(\gamma^n)$ by virtue of the Hadamard bound on the size of the determinant of

a matrix (GARLING, 2007), it follows that the public key has size $O(n^2 \lg \gamma)$ or simply $O(n^2 \lg n)$ bits, while the private key, which is essentially P, has size n^2 bits.

2. Encrypt: Let $m \in \{0,1\}^{n-k}$ be the plaintext. Select a random subset $S \subset \{1 \ldots n\}$ with k elements. The encoding of m is a vector $r \in \mathbb{Z}^n$ such that $r_i = h$ for $i \in S$, otherwise $r_i \xleftarrow{\$} \{1 \ldots \sigma/2\}$ if $m_j = 0$, and $r_i \xleftarrow{\$} \{\sigma/2 + 1 \ldots \sigma\}$ if $m_j = 1$, where i corresponds to the j-th index not in S. Compute $r - \lfloor rH^{-1} \rfloor H$, which, because of the particularly simple structure of the minimal HNF (see Definition 16), has the form $(0, \ldots, 0, c)$. The cryptogram is $c \in \mathbb{Z}$, the only nonzero coefficient thereof.

3. Decrypt: Let $c \in \mathbb{Z}$ be the ciphertext. Compute $c' \leftarrow (0, \ldots, 0, c)A^{-1} \in \mathbb{Q}^n$, which means simply $c' \leftarrow cA^{-1}_{(n-1)}$, and let $r' \leftarrow (c' - \lfloor c' \rceil)A$. Compute the error vector $e \in \{0,1\}^n$ by letting $e_i \leftarrow 1$ whenever $r'_i < 0$, otherwise $e_i \leftarrow 0$, for all $0 \leqslant i < n$. Compute the recovered message encoding as $r \leftarrow r' + eA$. Let $S := \{i \mid r_i = h\}$ (this is the same set S chosen during encryption). For all $0 \leqslant i < n$ such that $i \notin S$, extract $m_j \leftarrow 0$ if $0 < r_i \leqslant \sigma/2$, and $m_j \leftarrow 1$ if $\sigma/2 < r_i \leqslant \sigma$, where i corresponds to the j-th index not in S.

Notice that, strictly speaking, this is only a trapdoor one way function, not a full semantically secure encryption scheme. To attain semantic security, a suitable transform like Fujisaki-Okamoto (FUJISAKI; OKAMOTO, 1999) should be used.

2.5.2 Discussion of GGH-YK-M

The variant proposed by Barros and Schechter (BARROS; SCHECHTER, 2014) prevents all known attacks to the GGH family to date. The modifications ensure that the

underlying problem is not CVP, as it is in the original scheme. This, indeed, brings the scheme back to life, but it is still rather inefficient.

The remaining aspect to address, therefore, is to circumvent the inherent high bandwidth occupation and computational cost incurred by all traditional variants of GGH, which make this family of schemes less competitive in practice with other lattice-based encryption methods like Lindner-Peikert (LINDNER; PEIKERT, 2011). Even for rather small values of n, key sizes are of the order of hundreds of kB. Key generation is too slow for practical scenarios, taking several minutes (or even hours for larger dimensions). The obvious way to obtain shorter keys in other lattice-based settings like LWE (REGEV, 2005) or NTRU (HOFFSTEIN; PIPHER; SILVERMAN, 1998), namely, resorting to certain rings of structured (e.g., circulant or negacyclic) matrices, fails for GGH because mapping the private key to a public key, that is, computing the HNF, ends up destroying the underlying structure that would enable the size reduction, and thus does not help in attaining that goal. These alternatives will be discussed in depth in Chapter 3, which also presents a viable solution for this matter.

2.6 The LMSV scheme

In this section, we explain the second scheme on which we test our method. This cryptosystem is a *somewhat homomorphic* scheme, and it was named after its creators Loftus, May, Smart and Vercauteren (LOFTUS et al., 2012). As explained in 2.3.2, homomorphic cryptosystems allow for computations over encrypted data. The resulting ciphertext, when decrypted, corresponds to the same plaintext as if the target function had been computed on plaintexts the entire time. When any function is supported, there is fully homomorphism, whereas when only a limited set of functions is supported, the scheme is said to be somewhat homomorphic. LMSV fits into this last case.

The first fully homomorphic scheme was proposed by Gentry (GENTRY, 2009a),

a construction based on GGH-style lattices that successfully proves that fully homomorphism is, indeed, possible. Nonetheless, the prohibitively large size of its keys and ciphertexts, between GB and TB, remained an aspect to be improved. To tackle this issue, Smart and Vercauteren (SMART; VERCAUTEREN, 2010) proposed a modification that attains better results, but it is still far from practical. This method was additionally revised by Gentry and Halevi (GENTRY; HALEVI, 2011), and a few relevant aspects of these two contributions are detailed in Chapter 3. The proposition of Loftus, May, Smart and Vercauteren, based on these methods, represents the "fourth generation" of descendants of Gentry's original scheme. Moreover, even though there exist other families of homomorphic schemes, LMSV is the only thus far that has not succumbed to cryptanalysis (CHENAL; TANG, 2014), making it an extremely relevant study case.

Considering the great potential of homomorphic encryption, and also the fact that both LMSV and our method are based on Gentry and Halevi's technique, it was a natural path to investigate whether our method can be applied to improve it. This topic is further discussed in later chapters.

2.6.1 Cryptosystem definition

Let n be a power of 2, t be an integer greater than $2^{\sqrt{n}}$ and μ a small integer (usually one). Pick an irreducible polynomial $F \in \mathbb{Z}[x]$ of degree n. The LMSV scheme consists of the following three algorithms:

1. Keygen: Sample a polynomial $G \in \mathbb{Z}[x]$ of degree at most $n-1$ and coefficients bounded by t. Calculate $d \leftarrow \mathsf{Resultant}(F, G)$. G is chosen such that $G(x) \equiv 1 \bmod 2$, and $G(x)$ has a single unique root in common with $F(x)$ modulo d, denoted by u. Calculate $Z(x) \leftarrow d/G(x) \bmod F(x)$. The public key is the pair (u, d), and the private key is the pair $(Z(x), d)$.

2. Encrypt: The plaintext is $M(x) \in \mathbb{F}_2/F(x)$, i.e., messages are given by binary polynomials of degree less than n. Sample $R(x) \in \mathbb{Z}[x]$ such that its norm is no greater than μ. Calculate $C(x) \leftarrow M(x) + 2 \cdot R(x)$. The ciphertext is $c \leftarrow C(u) \bmod d$.

3. Decrypt: Calculate $C(x) \leftarrow c - \lfloor c \cdot Z(x)/d \rceil$. The plaintext is $M(x) \leftarrow C(x) \bmod 2$.

An analysis of the security of LMSV and its homomorphism is available in Section 4.2. Key and ciphertext sizes, as well as computing times, are shown in Section 5.2, which compares results in the literature with the ones obtained in our tests.

2.7 Synopsis

This chapter has presented an overview of related work used as a base for the results achieved, as well as an introduction to essential concepts.

We have discussed some fundamental notions of cryptography, introducing terminology and defining cryptosystems and their utilization. The theoretical background for lattices has been presented, including some relevant lattice problems. We have reviewed many existing cryptographic schemes based on lattices, in particular the asymmetric GGH cryptosystem. We have defined the notation used in this work and provided some definitions for concepts used throughout this work. Finally, we introduced the GGH-YK-M and LMSV cryptosystems, on which we test our method, as explained in the following chapters.

3 IMPROVEMENTS ON EFFICIENCY

In this chapter we describe an efficient key generation technique for GGH-style cryptosystems. When applied to the GGH-YK-M cryptosystem (BARROS; SCHECHTER, 2014), it greatly reduces public key bandwidth occupation, while also achieving much higher performance. It can also be applied to the LMSV scheme (LOFTUS et al., 2012) for improving its security against newly developed quantum key recovery attacks.

The technique proposed by Smart and Vercauteren (SMART; VERCAUTEREN, 2010) and perfected by Gentry and Halevi (GENTRY; HALEVI, 2011), targeted at homomorphic encryption, can be used to address the performance problem of the GGH family of schemes. However, both have limitations to practical use. The former depends on the lattice determinant to be prime, while the latter relies heavily on the special form of the ring $\mathbb{Z}[x]/(x^n + 1)$ where n is a power of 2. Besides, it requires the computation of resultants and the explicit extraction of the roots of polynomials modulo the lattice determinant, which is done through a quite complex modification of the extended Euclidean algorithm.

Our work extends their technique to any value of n and also for the circulant ring $\mathbb{Z}[x]/(x^n - 1)$, for which we also provide a structural security analysis. In particular, and surprisingly, prime values of n are observed to lead to faster key generation, despite the unavailability of fast Fourier transform techniques to speed up the computations. Our technique only requires a straightforward application of the usual extended Euclidean algorithm, coupled with the Chinese remainder theorem and the fast Fourier transform.

Furthermore, we avoid the need to resort to a full-fledged HNF algorithm, in the same way as the Smart-Vercauteren and Gentry-Halevi methods, for the reason that the best known algorithm for calculating the HNF is of $O(n^2)$ space complexity and $O(n^5)$ running time (MICCIANCIO; WARINSCHI, 2001). This is accounted as too inefficient for our purposes.

3.1 Improvements

The usual technique adopted to reduce space requirements and bandwidth occupation in lattice-based cryptosystems is to resort to certain structured matrices that correspond to ideals in polynomial rings (MICCIANCIO, 2002; LYUBASHEVSKY; MICCIANCIO, 2006; MICCIANCIO, 2007). These matrices are associated to specific kinds of ideal lattices.

The greatest advantage of this technique is that cyclic ideal lattices can be defined by a single vector. It is necessary to store only the first row of the matrix representation of the lattice basis, because deriving the rest of the matrix is a trivial process. In this way, there is no need to store all n^2 elements, reducing key size from $O(n^2)$ to $O(n)$ elements.

The most popular choices are circulant matrices, associated to the polynomial ring $\mathbb{Z}[x]/(x^n - 1)$, and negacyclic matrices, which correspond to the polynomial ring $\mathbb{Z}[x]/(x^n + 1)$. Circulant matrices are matrices that each row is the previous row cycled one step to the right (see Definition 9). Figure 6 shows an example of a circulant matrix. Negacyclic matrices are almost the same, except that the element returning to the first position is also multiplied by -1 (see Definition 10). An example of a negacyclic matrix can be seen at Figure 7.

More generally, one could consider the $n \times n$ matrices whose i-th row contains the coefficients of $a(x)x^i \bmod p(x)$ for some $a(x)$ and a fixed but arbitrary monic polyno-

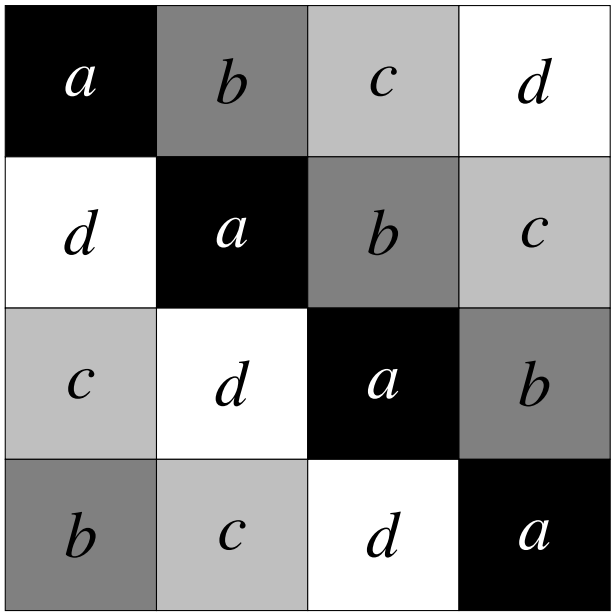

Figure 6: A circulant matrix.

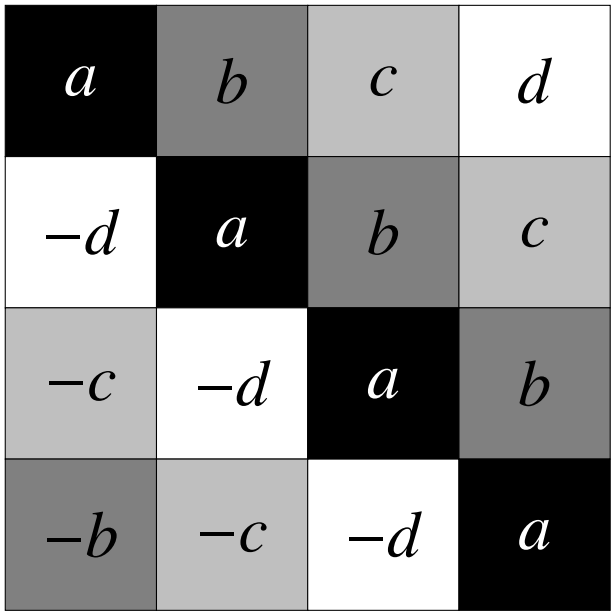

Figure 7: A negacyclic matrix.

mial $p(x)$ of degree n without multiple roots (and preferably small coefficients). Such matrices correspond to the ideals of a polynomial ring $\mathbb{Z}[x]/p(x)$.

Unfortunately, this technique does not seem to improve the space requirements of GGH, nor, for that matter, of any other cryptosystem that relies on the HNF as public key. This is because the HNF is usually *not* in the same (structured) ring as the original matrix. Thus, for instance, HNF(A) in general is *not* circulant or negacyclic even though A displays such symmetries (except if A is a scalar matrix). Therefore, by resorting to circulant or similarly structured matrices one would apparently be able at most to reduce the size of private keys from n^2 down to n elements, but public keys would remain the same.

Contrary to this intuitive observation, one can still benefit from an underlying structure in the private key to reduce the size of the public key in a nontrivial way. This was first indicated by Smart and Vercauteren (SMART; VERCAUTEREN, 2010), but it seems to require computing the HNF of the lattice basis. Gentry and Halevi (GENTRY; HALEVI, 2011, Lemma 1) offer a proof of this property that avoids computing the HNF for the case $p(x) = x^n + 1$ (where n is a power of 2). We show that, in fact, it holds for any ideal matrix, regardless of the choice of $p(x)$, even though some choices may be more efficient (and possibly more secure) than others.

3.1.1 A method to calculate the HNF

We now present a method to calculate the minimum Hermite normal form, if it exists, of a matrix without making use of a generic (and more expensive) algorithm.

If matrix P in the Keygen algorithm of GGH-YK-M is associated to a polynomial ring $\mathbb{Z}[x]/p(x)$, then matrix A is associated to a polynomial in the same ring, and although $H :=$ HNF(A) does not display the ring symmetry (i.e., H is not circulant, etc), its rows still correspond to elements of that ring, as it is an equivalent basis for the same lattice. Thus, if $a(x)$ is the polynomial associated to any row of H, then $xa(x) \bmod p(x)$

and $x^{-1}a(x) \bmod p(x)$ are two other (independent) vectors on the same lattice.

Given that $H_{(n-2)} = (0, \ldots, 0, 1, u)$ for some $u \in \mathbb{Z}$ (because H is assumed to be minimal), the polynomial associated to it is $ux^{n-1} + x^{n-2} = (ux + 1)x^{n-2}$, and hence $(ux + 1)x^i = (x^{-1})^{i-(n-2)}(ux + 1)$ stands for yet another vector on that lattice for every $0 \leqslant i < n - 1$. Collecting all of these vectors together with $H_{(n-1)}$, one gets

$$H' = \begin{bmatrix} 1 & u & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & u & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & u & 0 \\ 0 & 0 & 0 & \ldots & 0 & 1 & u \\ 0 & 0 & 0 & \ldots & 0 & 0 & d \end{bmatrix}, \tag{3.1}$$

which is an alternative basis for the same lattice, since all of its rows are linearly independent vectors from that lattice, and H' shares the same determinant d as H (and A). But because the HNF is unique, it also follows that $\mathrm{HNF}(H') = H$, and by applying a straightforward Gaussian elimination on H', namely by changing $H'_{(n-1-j)} \leftarrow H'_{(n-1-j)} - uH'_{(n-j)}$ successively for $2 \leqslant j < n$ and then reducing modulo d, one gets

$$H = \begin{bmatrix} 1 & 0 & \ldots & 0 & 0 & -(-u)^{n-1} \bmod d \\ 0 & 1 & \ldots & 0 & 0 & -(-u)^{n-2} \bmod d \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 & -u^2 \bmod d \\ 0 & 0 & \ldots & 0 & 1 & u \\ 0 & 0 & \ldots & 0 & 0 & d \end{bmatrix}, \tag{3.2}$$

and by comparing the result with the definition of minimal H (see Definition 16) yields $v_i = -(-u)^{n-1-i} \bmod d$ for $0 \leqslant i < n - 1$.

Therefore, H (and its inverse) can be efficiently represented simply by $(u, d) \in \mathbb{Z}^2$. Because $0 < u < d$ and d satisfies the Hadamard bound for A, which is $d \leqslant |a|^n$, where

$|a|$ is the norm of the first vector of A, it follows that, for GGH-YK-M, $d \leqslant \gamma^n$. Thus, H can be represented with only $2n \lg \gamma$ and hence $O(n \lg n)$ bits. This represents a vast improvement over the naive $O(n^2 \lg \gamma)$ or $O(n^2 \lg n)$ size of the whole (v, d) for practical values of n (typically in the hundreds).

The remaining tasks are computing d and u from A. We now address these tasks individually. Our approach avoids both the computation of resultants and complex modifications of the extended Euclidean algorithm.

3.1.1.1 Computing the determinant *d*

Computing the determinant d is accomplished by diagonalizing the projections of A onto a number of finite fields $\mathbb{F}_{q_0}, \ldots, \mathbb{F}_{q_{t-1}}$ such that $d < \prod_k q_k$, since this enables computing d mod q_k for each q_k, and then recovering d by means of the Chinese remainder theorem. This is possible as long as the polynomial $p(x)$ splits completely into n distinct linear factors over each of those fields. If that is the case, let $V \in \mathbb{F}_{q_k}^{n \times n}$ be the Vandermonde matrix (see Definition 14) built from the n distinct roots of $p(x)$ over $\mathbb{F}_{q_k}$, i.e., $V_{ij} := z_j^i$ with $p(z_j) = 0$ and $z_j \in \mathbb{F}_{q_k}$. Then V is invertible, and the diagonal form of A is $V^{-1}AV$ (the eigenvalues themselves are just the sequence of components of $A_{(0)}V$).

The obstacle to this approach is finding the fields $\mathbb{F}_{q_k}$ such that $p(x)$ splits in the required form over all of them. Exhaustive search via the factorization of an arbitrary $p(x)$ over candidate fields is far too expensive, even for fairly small n. One could reverse the reasoning and choose the roots of $p(x)$ first, but this only enables the computation of a single field $\mathbb{F}_q$ over which $p(x)$ splits, and because the coefficients of such a $p(x)$ are expected to be rather large, any private basis is usually very large as well, yielding an even larger determinant d which is likely to exceed q by a factor exponentially large in n, and hence precluding the recovery of d from its value mod q alone.

However, the circulant and negacyclic cases offer a much better prospect, since all that is required for $p(x)$ to split over $\mathbb{F}_{q_k}$ is that $n \mid q-1$ in the former case, and $2n \mid q-1$ in the latter. When n is a power of 2, the computation of the diagonal form of A amounts to a fast Fourier transform (more precisely, a fast number theoretic transform), which takes time $O(n \lg n)$ products by certain fixed roots of unity in $\mathbb{F}_{q_k}$. However, computation of the eigenvalues is fairly efficient even for general n, and as we shall see this extra flexibility in the choice of n tends, a bit surprisingly, to offer better key generation performance.

3.1.1.2 Computing u

Assuming that the fields $\mathbb{F}_{q_k}$ are available and that the determinant d has been computed, the value of u, if it exists, can be computed as follows. The first row of H' is expected to have the form $(1, u, 0, \ldots, 0)$, associated to the polynomial $ux + 1$ in the underlying polynomial ring. The rows of the matrix H^* corresponding to this polynomial spell the coefficients of $(ux+1)x^i \bmod p(x)$:

$$H^* = \begin{bmatrix} 1 & u & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & u & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & u & 0 \\ 0 & 0 & 0 & \ldots & 0 & 1 & u \\ u & 0 & 0 & \ldots & 0 & 0 & 1 \end{bmatrix}. \tag{3.3}$$

Thus H^* differs from H' only in its last row, and it defines a sub-lattice of the lattice defined by H' or, equivalently, by A.

Therefore, there must exist a matrix $M \in \mathbb{Z}^{n \times n}$ (actually in the same ring as A and H^*) such that $MA = H^*$. Let $A^\dagger$ be the classical adjoint (or adjugate) of A, i.e., $AA^\dagger = dI$. Then $dM = H^*A^\dagger$, and the peculiar structure of H^* reduces this to the Diophantine equation $dM_j - A^\dagger_{j-1}u = A^\dagger_j$ for all j. Thus, if any solution to this equation

exists, it is $u = -A_j^\dagger / A_{j-1 \pmod n}^\dagger \bmod d$ for any j, which requires all $A_j^\dagger$ to be invertible mod d. However, this in turn actually requires only that $A_0^\dagger$ and $A_1^\dagger$ be invertible mod d, since then $A_j^\dagger = A_0^\dagger (A_1^\dagger / A_0^\dagger)^j = A_0^\dagger (-u)^j \bmod d$ as one can check by induction.

This provides a simple algorithm to determine at once whether HNF(A) is minimal, and if so, what the value of u in Equation 3.1 is. Indeed, $A^\dagger$ can be computed via the Chinese remainder theorem from $A^\dagger = dA^{-1} \bmod q_k$, and the extended Euclidean algorithm then yields $u \leftarrow -A_1^\dagger / A_0^\dagger \bmod d$ or proves that no such u exists.

The efficient key pair generation this process enables, without a full HNF algorithm, arguably outweighs the practical restriction for $p(x) = x^n \pm 1$. This method works for any choice of n. Processing times are much smaller for this compact representation than they are for unstructured matrices. Due to security concerns, we also address a different case where $p(x) = x^n - x - 1$ in Chapter 4. Experimental results are reported in Chapter 5.

3.2 Synopsis

In this chapter, we have presented the theoretical aspects of our contributions. We justified why the methods for reducing key sizes in NTRU and LWE cryptosystems do not solve the present issues for GGH-style schemes like GGH-YK-M and LMSV. We then presented a new way to calculate the Hermite normal form, resorting to the Chinese remainder theorem. This method allows for much faster processing times, as we will see in Chapter 5, and also smaller sizes for public and private keys. It is a rather straightforward algorithm, which may also benefit implementability. Its security is discussed in Chapter 4, where we also present another alternative to prevent recently published quantum attacks to the lattices used by LMSV.

4 SECURITY CONSIDERATIONS

In this chapter we reflect about the security of our proposed method. Firstly, we introduce the concept of semantic security and its variants, namely, IND-CPA, IND-CCA1 and IND-CCA2. Then, we outline existing quantitative security analyses of the Smart-Vercauteren construction (and LMSV for that matter). After that, we describe an apparent key recovery attack based on the peculiar structure of circulant lattices, showing that it leaks only a small amount of information on the private key, specifically $O(\lg n)$ bits. We then discuss some very recent key recovery attacks that have been reported, specially the quantum attack on the Soliloquy scheme by the British Government Communications Headquarters (GCHQ) (CAMPBELL; GROVES; SHEPHERD, 2014). This matter is surrounded by controversy and still demands closer investigations by the scientific community, but is nonetheless relevant to our case. Finally, we ponder over how our original method can be modified to prevent these new attacks, improving its overall security.

The original authors of GGH-YK-M do not state explicitly that the security of their construction is based on the assumption that the attacker is not able to generate an equivalent lattice basis of size $O(n)$, leaving this conclusion to be tacitly understood by readers. This is, however, a much stronger assumption compared to LMSV, which requires an approximation of size $O(\sqrt{n})$. For this reason, even though LMSV was designed as a somewhat homomorphic scheme and GGH-YK-M as a traditional public-key cryptosystem, we recommend the adoption of the former in scenarios where the latter would be used, as LMSV is inherently more secure, even when instantiated with

parameter choices that do not support homomorphism.

4.1 Security fundamentals

The notion of *semantic security* was originally proposed by Turing Award laureates Shafi Goldwasser and Silvio Micali (GOLDWASSER; MICALI, 1982), and later extended by the same authors (GOLDWASSER; MICALI, 1984). This concept determines that the ciphertext leaks no interesting bit of information about the plaintext, as "semantic security means that whatever can be efficiently computed from the ciphertext can be efficiently computed when given only the length of the plaintext. Note that this formulation does not rule out the possibility that the length of the plaintext can be inferred from the ciphertext. Indeed, some information about the length of the plaintext must be revealed by the ciphertext. We stress that other than information about the length of the plaintext, the ciphertext is required to yield nothing about the plaintext." (GOLDREICH, 2004). This notion can also be described as a game between a challenger and an adversary, known as the *semantic security game*. The game is depicted in Figure 8 and runs as follows (VAUDENAY, 2005):

1. First of all, the challenger and the adversary are given the cryptosystem.
2. The challenger generates a matching pair of public and secret keys and discloses the public one.
3. The adversary selects two plaintexts x_0 and x_1 and sends them to the challenger.
4. The challenger picks uniformly at random a bit b. He encrypts x_b and sends the ciphertext c to the adversary.
5. The adversary tries to guess b. They win if $b = b'$.

As a consequence, if the adversary randomly guesses the value of b, they will win the game with probability $Pr[b = b'] = 1/2$. We can define the advantage of an

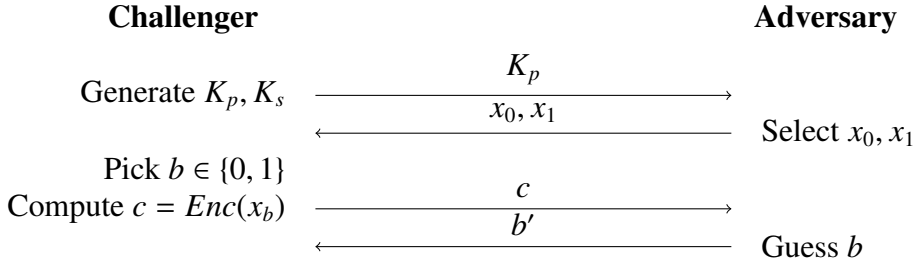

Figure 8: Semantic security game. Source: (VAUDENAY, 2005).

adversary $\mathbb{A}$ attacking a cryptosystem $\mathbb{E}$ as

$$Adv_{\mathbb{A}}(\mathbb{E}) = \left| Pr[b = b'] - \frac{1}{2} \right|.$$

If $\mathbb{A}$ is not able to guess the correct value of b with a significant advantage, the messages are said to be *indistinguishable* and the cryptosystem is *indistinguishable under chosen plaintext attack* (IND-CPA).

The notion of semantic security can be extended to model adversaries with more resources. Assume that the adversary has access to a decryption oracle during step 3. Prior to the selection of x_0 and x_1, they can query this oracle as many times as they wish. After this phase, the adversary has no longer access to this oracle, selecting the two plaintexts and playing like in the previous game. This is called a *chosen ciphertext attack* (CCA1) – or *lunchtime attack*, due to limited access to the oracle – and a secure cryptosystem in this case is considered *indistinguishable under chosen ciphertext attack* (IND-CCA1).

A third scenario is to allow access to the decryption oracle even after receiving the ciphertext c, with the only restriction being that the challenger is not allowed to decrypt c itself. If the cryptosystem is still secure in this case, then it is *indistinguishable under adaptive chosen ciphertext attack* (IND-CCA2). A more formal definition is as follows.

Definition 18. *(GENTRY, 2009a, Definition 2.2.1)* **(Semantic Security against (CPA, CCA1, CCA2) attacks).** *We say* $\mathbb{E}$ *is* semantically secure *against (CPA, CCA1, CCA2)*

attacks if no polynomial time (CPA, CCA1, CCA2)-adversary $\mathbb{A}$ breaks $\mathbb{E}$ with non-negligible advantage.

4.2 Security of homomorphic schemes

Due to the extreme malleability of their ciphertexts, IND-CCA2 security is unattainable for homomorphic schemes (GENTRY, 2009a). The reason for that is simple: in this scenario, the adversary is allowed to manipulate the challenged ciphertext and submit it to the decryption oracle. Relaxations of the notion of CCA2 security have been proposed with the goal of defining "homomorphic-CCA security" (PRABHAKARAN; ROSULEK, 2008), but these do not extend to fully homomorphic constructions.

As previously explained in Section 2.3, the first fully homomorphic cryptosystem was described by Gentry (GENTRY, 2009a). This original proposition is IND-CPA, and the author left the definition of CCA1-secure homomorphism as an open problem. Later papers followed Gentry's line of work (SMART; VERCAUTEREN, 2010; GENTRY; HALEVI, 2011; LOFTUS et al., 2012), and instances of homomorphic encryption based on other kinds of lattices have been described, such as (NAEHRIG; LAUTER; VAIKUNTANATHAN, 2011; BRAKERSKI; GENTRY; VAIKUNTANATHAN, 2012; BRAKERSKI, 2012; GENTRY; SAHAI; WATERS, 2013; BRAKERSKI; VAIKUNTANATHAN, 2014), based on the LWE problem (REGEV, 2005), and also (STEHLÉ; STEINFELD, 2010; LÓPEZ-ALT; TROMER; VAIKUNTANATHAN, 2012; BOS et al., 2013), based on NTRU (HOFFSTEIN; PIPHER; SILVERMAN, 1998).

Unfortunately, it has been proven that most existing homomorphic encryption schemes are not IND-CCA1 secure (LOFTUS et al., 2012; SZYDLO, 2003; CHENAL; TANG, 2014; DAHAB; GALBRAITH; MORAIS, 2015; CHENAL; TANG, 2015). In fact, these schemes "suffer from key recovery attacks (stronger than a typical IND-CCA1 attacks),

which allow an adversary to recover the private keys through a number of decryption oracle queries" (CHENAL; TANG, 2014). The only scheme thus far deemed to be IND-CCA1 secure is the one by Loftus *et al.* (LOFTUS et al., 2012). However, some newly developed classic and quantum key recovery attacks seem to apply to this construction. These particular issues are throughly discussed in Section 4.4.

Most studies about security levels of current state-of-the-art homomorphic schemes have been limited to qualitative analyses, classifying schemes and parameter choices in ranges such as "toy", "low" and "high". There has been little preoccupation about more precise estimations from a quantitative point of view, due to the prohibitively large sizes of ciphertexts and keys (and consequently, their proportionally slow processing times).

Table 2: Security and homomorphism in the Smart-Vercauteren scheme

n	Security level	Multiplicative depth
2^8	2^{25}	0.3
2^9	2^{31}	0.8
2^{10}	2^{41}	1.2
2^{11}	2^{54}	1.7
2^{12}	2^{73}	2.1
2^{13}	2^{100}	2.5

By virtue of its similarity to our proposition from Chapter 3, the Smart-Vercauteren cryptosystem is of particular interest. Its predicted security is summarized in Table 2. The Gentry-Halevi and LMSV constructions are correlated enough to this first scheme to be safe to say that approximately the same security levels are obtained for the same parameter choices. From Table 2, it is plain to see that estimated security levels are still far below the current standard of 2^{128}, even for the largest shown dimensions. Nonetheless, this is not the greatest issue. The multiplicative depth, that is, the maximum number of multiplications that can be successfully computed homomorphically while still guaranteeing correct output upon decryption, even for the largest value of n presented, is merely two. This is too little if we consider that the purpose of these schemes is to support the homomorphic computation of arbitrarily complex functions.

As a result, it is not viable to attain fully homomorphic encryption for practical values of n. In truth, it is possible to have fully homomorphism for $n \geqslant 2^{27}$ (SMART; VERCAUTEREN, 2010), but this would produce extremely large key sizes.

This justifies why security has not been a central concern for homomorphic encryption proposals. There is no point in further scrutinizing the security of these schemes while they still demand preposterously large keys to support full homomorphism. This fact reinforces the importance of the study and development of efficient methods for lattice-based cryptography. Performance of LMSV is further discussed in Section 5.2.

4.3 Attacking circulant lattices

In Chapter 3, we presented a new method for generating keys, displaying the circulant case as an example. However, adopting a structured matrix as the private key must be made carefully to avoid introducing weaknesses. The particular case $p(x) = x^n + 1$ where n is a power of 2 has received a considerable amount of attention in the literature. We now analyze how circulant lattices, corresponding to $p(x) = x^n - 1$, have the drawback of leaking a small amount of information on the private key, specifically $O(\lg n)$ bits thereof. As always, our analysis does not require n to be a power of 2. Admittedly, the security level attainable when generalizing n is less clear, though it seems unlikely that this would introduce any weakness that is not already present in the more extensively analyzed NTRU scenario, where prime n is the usual choice.

We begin by noticing that the sum $\lambda := \sum_j A_j$ is bound between $\gamma - n$ (when P is the all-one ring element) and γ (when P is zero). Let $\Lambda := \sum_j A_j^\dagger$. The following property holds:

Lemma 1. $d = \lambda\Lambda$.

Proof. By definition of adjugate matrix, $AA^\dagger = dI$. Then $A_{(j)}A^\dagger = dI_{(j)}$ and hence

$\sum_j A_{(j)}A^\dagger = \sum_j dI_{(j)}$, which yields $(\lambda, \ldots, \lambda)A^\dagger = (d, \ldots, d)$, since the elements on each column of A are the same except for a circular permutation, and thus all columns of $\sum_j A_{(j)}$ take the value $\sum_j A_j = \lambda$. Now $(\lambda, \ldots, \lambda)A^\dagger = \lambda(1, \ldots, 1)A^\dagger$, which is simply $(\lambda\Lambda, \ldots, \lambda\Lambda)$ because $(1, \ldots, 1)A^\dagger = (\sum_j A_j^\dagger, \ldots, \sum_j A_j^\dagger) = (\Lambda, \ldots, \Lambda)$. Therefore $(\lambda\Lambda, \ldots, \lambda\Lambda) = (d, \ldots, d)$ which repeats the claim n times, i.e., $d = \lambda\Lambda$. □

Lemma 2. $(-u)^n - 1 \equiv 0 \bmod d$.

Proof. We show that $(0, \ldots, -(-u)^n + 1)$ is a lattice vector in the subspace generated by $H_{(n-1)} = (0, \ldots, d)$. Consider the lattice generated by

$$C^{(0)} = \begin{bmatrix} 1 & u & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & u & \ldots & 0 & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & u & 0 & 0 \\ 0 & 0 & 0 & \ldots & 1 & u & 0 \\ 0 & 0 & 0 & \ldots & 0 & 1 & u \\ u & 0 & 0 & \ldots & 0 & 0 & 1 \end{bmatrix},$$

where the superscript denotes a stage in the Gaussian elimination process described below, with (0) indicating the original matrix. This is a sublattice of the original lattice, since it only involves rotations of the first row of H' defined by Equation 3.1. Applying Gaussian elimination to the last row as $C_{(n-1)}^{(j+1)} \leftarrow C_{(n-1)}^{(j)} + (-u)^{j+1} C_{(j)}$ for $j = 0, \ldots, n-1$,

we get

$$C^{(n)} = \begin{bmatrix} 1 & u & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & u & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & u & 0 & 0 \\ 0 & 0 & 0 & \dots & 1 & u & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & u \\ 0 & 0 & 0 & \dots & 0 & 0 & -(-u)^n + 1 \end{bmatrix}.$$

Thus $C^{(n)}_{(n-1)}$ is in the subspace spanned by $H'_{(n-1)}$, i.e., $C^{(n)}_{(n-1)} = \kappa H'_{(n-1)}$ for some κ. Thus $-(-u)^n + 1 = \kappa d$, i.e., $(-u)^n - 1 \equiv 0 \bmod d$ as claimed. □

Let $Z := \sum_j (-u)^j \bmod d$. Given that $A^\dagger_j = A^\dagger_0(-u)^j \bmod d$, it follows that $\sum_j A^\dagger_j = A^\dagger_0 \sum_j (-u)^j \bmod d$ and thus one can deduce that

$$\Lambda = A^\dagger_0 Z \bmod d. \tag{4.1}$$

At first glance this equation might seem to provide a means to recover the full $A^\dagger_0$ by inverting $Z \bmod d$. That this cannot actually happen is established by the following property:

Lemma 3. $\Lambda \mid \gcd(Z, d)$*, and hence* Z *is not invertible* mod d*.*

Proof. From Equation 4.1 and Lemma 2 it follows that $\lambda A^\dagger_0 Z = \lambda\Lambda = 0 \bmod d$ and since, by the key generation requirement of Section 3.1, $A^\dagger_0$ itself is invertible mod d, then $\lambda Z = 0 \bmod d$, i.e., $\lambda Z = Z'd = Z'\lambda\Lambda$ for some integer Z', meaning that $Z = Z'\Lambda$, i.e., Z itself is a multiple of Λ, and hence cannot be invertible mod d by virtue of having the common factor Λ with d. □

However, equation $\Lambda = A^\dagger_0 Z \bmod d$ does reveal a small piece of information on $A^\dagger_0$.

Indeed, $\Lambda = A_0^\dagger Z + \kappa d = A_0^\dagger Z'\Lambda + \kappa\lambda\Lambda$ for some κ, and hence $1 = A_0^\dagger Z' + \kappa\lambda$ by removing the common factor Λ, or simply $1 = A_0^\dagger Z' \bmod \lambda$. This reveals $A_0^\dagger \bmod \lambda = Z'^{-1} \bmod \lambda$ as long as Z' is invertible mod λ. However, this amounts to revealing only $O(\lg n)$ bits of the private value $A_0^\dagger$.

These considerations notwithstanding, it is important to note that in cases when $\Lambda = 1$, one can fully recover $A_0^\dagger$, as it is simply the inverse of $Z \bmod d = Z \bmod \lambda$, because in this case, $d = \Lambda\lambda = \lambda$. Nevertheless, we know from Section 2.5 that $\gamma = \alpha n$ for some small integer α, and since $\gamma - n \leqslant \lambda \leqslant \gamma$, we have that $\lambda = O(n)$. We also know that $d = O(n^n)$, therefore the probability of having $d = O(n)$ decreases exponentially as n increases. Additionally, this case can be completely avoided by requiring that $d \neq \lambda$ (or that $d > \gamma$, for that matter).

Correspondingly, in cases when $\lambda = 1$, we have that $d = \Lambda$, and therefore Equation 4.1 becomes $0 = A_0^\dagger Z \bmod d$, which implies that, necessarily, $Z = 0 \bmod d$ for every possible value of $A_0^\dagger$. For the same reason as before, taking $\alpha = 2$ is enough to prevent this case from happening.

On the constructive side, $(u+1)\Lambda = A_0^\dagger(u+1)\sum_j (-u)^j \bmod d = -A_0^\dagger((-u)^n - 1) \bmod d = 0$, $(u+1)\Lambda = \xi\lambda\Lambda$ for some ξ, and hence $\lambda \mid u+1$. Thus λ is a common factor between d and $u+1$, and can be factored out by publishing the public key as the triple $(d/\lambda, (u+1)/\lambda, \lambda)$ instead of the pair (u, d), saving $O(\lg n)$ bits.

This also shows that the attack cannot be extended to recover the whole matrix $A \bmod \lambda$ (from which A could be extracted immediately) from $A^\dagger \bmod \lambda$. Because $u + 1 = 0 \bmod \lambda$ and hence $-u = 1 \bmod \lambda$, it follows that $A_j^\dagger = A_0^\dagger(-u)^j \bmod \lambda$ (this equality holds because $\lambda \mid d$) and hence $A_j^\dagger = A_0^\dagger \bmod \lambda$ for all j, so that $A^\dagger = A_0^\dagger U \bmod \lambda$ where U is the (singular) all-one matrix. Therefore the adjugate mapping mod λ cannot be inverted to recover A from $A^\dagger \bmod \lambda$.

Interestingly, this attack does not apply to negacyclic lattices (or, for that matter,

most or perhaps all other ideal lattices), because Lemma 1 does not hold, i.e., the determinant, in general, is not the product of a linear combination of the components of A and a linear combination of the components of $A^{\dagger}$.

4.4 Attacking cyclotomic lattices

Very recently, new key recovery attacks to cyclotomic rings have been described. There exist classical approaches, but a new quantum attack in particular has provoked heated discussions, as it represents big news related to what had been always deemed as *"post-quantum cryptography"* up until this point. This issue is still somewhat controversial, in particular whether the attack is polynomial-time or not. There is still very limited literature on the subject, and the matter is far from settled. Nonetheless, it is very important to take note of it, as it seems to apply to some of the most commonly used lattices.

In this section, we show that all circulant lattices, as well as negacyclic with power-of-two dimensions, are cyclotomic. After that, we discuss the freshly developed key recovery attacks, estimating in what specific scenarios they can be applied.

4.4.1 Cyclotomic rings

The notion of cyclotomic rings relies on the following definitions:

Definition 19. *(RADEMACHER, 1964, Chapter 8)* **(Primitive roots of unity).** *A solution to the algebraic equation*

$$x^n - 1 = 0 \tag{4.2}$$

is called an nth root of the unity *or a* root of unity of order n. *Those roots of Equation 4.2 which are not also roots of* $x^k - 1 = 0$ *with* $k < n$ *are called* primitive roots of unity.

Definition 20. *(APOSTOL, 1970)* **(Cyclotomic polynomial).** *The cyclotomic polynomial $\Phi_n(x)$ of order $n \geqslant 1$ is the polynomial whose roots are the primitive n-th roots of the unity,*

$$\Phi_n(x) = \prod_{\substack{1 \leq k \leq n \\ \gcd(k,n)=1}} \left(x - e^{2i\pi\frac{k}{n}}\right),$$

where the index k runs through integers relatively prime to n and i is the imaginary unit $i = \sqrt{-1}$.

Some examples of cyclotomic polynomials:

$$\begin{aligned}
\Phi_1(x) &= x - 1 & \Phi_6(x) &= x^2 - x + 1 \\
\Phi_2(x) &= x + 1 & \Phi_7(x) &= x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 \\
\Phi_3(x) &= x^2 + x + 1 & \Phi_8(x) &= x^4 + 1 \\
\Phi_4(x) &= x^2 + 1 & \Phi_9(x) &= x^6 + x^3 + 1 \\
\Phi_5(x) &= x^4 + x^3 + x^2 + x + 1 & \Phi_{10}(x) &= x^4 - x^3 + x^2 - x + 1
\end{aligned}$$

The first relevant property of cyclotomic polynomials is that $\Phi_n(x)$ of order n is a monic polynomial with integer coefficients of degree $\varphi(n)$, where φ is Euler's totient function (that is, the number of positive integers less than or equal to n that are relatively prime to n). When n is a prime number, we have that

$$\Phi_n(x) = 1 + x + x^2 + \cdots + x^{n-1} = \sum_{i=0}^{n-1} x^i. \tag{4.3}$$

The polynomial $x^n - 1$ can be factored as the multiplication of all cyclotomic polynomials Φ_d where d is a divisor of n, that is,

$$x^n - 1 = \prod_{d|n} \Phi_d(x). \tag{4.4}$$

Another relevant property is related to some polynomials with power-of-two degrees:

$$n = 2^\beta, \beta \in \mathbb{N} \Rightarrow \Phi_{2n}(x) = x^n + 1. \tag{4.5}$$

For a more detailed study of cyclotomic polynomials, we refer the interested reader to (RADEMACHER, 1964) and (LIDL; NIEDERREITER, 1997).

A field $\mathbb{Q}[x]/P(x)$ is cyclotomic if $P(x)$ can be factored into a product of cyclotomic polynomials. If this is the case, then the ring $\mathbb{Z}[x]/P(x)$ also is cyclotomic. From Equations 4.4 and 4.5, we can infer that all circulant lattices are cyclotomic rings, because they correspond to the ring $\mathbb{Z}[x]/(x^n - 1)$, and negacyclic lattices, corresponding to $\mathbb{Z}[x]/(x^n + 1)$, are cyclotomic when the dimension n is a power of two. This last case is particularly relevant because it is adopted, for instance, in the Smart-Vercauteren (SMART; VERCAUTEREN, 2010) scheme and its descendants.

4.4.2 Key recovery attacks

One way to study the security of lattice-based cryptography is to consider that "cryptosystems are not based on "lattices" *per se*, but rather on certain computational problems on lattices. There are many kinds of lattice problems, not all of which appear to be equally hard – therefore, not all lattice-based cryptosystems offer the same qualitative level of security. For ideal lattices, the choice of problem matters at least as much as the choice of ring" (PEIKERT, 2015).

Recently, both classical and quantum attacks to lattices based on cyclotomic ideals have been developed. The notion behind them is equivalent to the following problem: given an ideal that is guaranteed to have a *short* generator g (for some definition of "short"), find a sufficiently short generator, not necessarily g itself. In other words, it is not necessary to recover the private key *itself* – it is sufficient to find an equivalent lattice basis shorter than some arbitrary upper bound $\gamma(n)$. Based on that, recently developed attacks are composed of two fundamental steps. First, find an arbitrary generator of the ideal, not necessarily short. Then, transform the generator found into a short one, thus finding the secret key (or a functional equivalent) (CRAMER et al., 2015). How exactly these generators are recovered depends on the lattice problem that

supports the cryptosystem in question.

In 2014, Bernstein reported a classical approach that may yield slightly subexponential complexity in cyclotomic rings (BERNSTEIN, 2014). By the end of the same year, the Communications-Electronics Security Group (CESG) of the Government Communications Headquarters (GCHQ) – the cryptology department of the British intelligence and security organization – published, at once, a new cryptosystem dubbed "Soliloquy" and a quantum key recovery attack against it (CAMPBELL; GROVES; SHEPHERD, 2014). Developed earlier by CESG itself and kept secret, Soliloquy is based on GGH-style lattices, very similar to the Smart-Vercauteren (SMART; VERCAUTEREN, 2010) method. The authors claim that their quantum algorithm destroys Soliloquy, allowing for the production of a short generator to the lattice in polynomial time. In particular, they write that in cyclotomic rings having power-of-two index the second step of the attack is *easy*. For this reason, the development of Soliloquy has been discontinued and partial results made public (CAMPBELL; GROVES; SHEPHERD, 2014). However, their paper did not provide enough proof to substantiate their assertions, sparking debate among the scientific community.

Based on the Campbell-Groves-Shepherd quantum attack, and also on another independently developed albeit similar work by Eisenträger *et al.* (EISENTRÄGER et al., 2014), Cramer *et al.* (CRAMER et al., 2015) attempt to provide a rigorous theoretical and practical confirmation of GCHQ's affirmation, further generalizing it to all cyclotomics of prime-power index. They declare that by combining their results with some other algorithms, in particular, the ones by Biasse (BIASSE, 2014), Biasse-Fieker (BIASSE; FIEKER, 2014), Campbell-Groves-Shepherd (CAMPBELL; GROVES; SHEPHERD, 2014), and Biasse-Song (see below), "one obtains quantum polynomial-time, or classical $2^{n^{2/3+\epsilon}}$-time, key recovery algorithms" (CRAMER et al., 2015, p. 3). Specifically mentioned as vulnerable are the cryptographic constructions of Smart-Vercauteren (SMART; VERCAUTEREN, 2010), Garg-Gentry-Halevi (GARG; GENTRY;

HALEVI, 2013), Langlois-Stehlé-Steinfeld (LANGLOIS; STEHLÉ; STEINFELD, 2014), and Soliloquy (CAMPBELL; GROVES; SHEPHERD, 2014).

Nonetheless, not all agree on the complexity of the resulting attack. Biasse, author and co-author of three of the four papers on which the paper by Cramer *et al.* relies, has disagreed with the claim that the quantum attack is polynomial, declaring (BIASSE, 2015)[1]

> This statement is not true, and it should be amended.
> It is based on two references to justify the existence of a quantum polynomial time algorithm to solve the Principal Ideal Problem (PIP).
> First, an online draft from Campbel (sic), Grove and Shepherd [CGS14]. This work in progress has been publicly released as it was when the corresponding research program at CESG was interrupted. According to its authors, the polynomial run time of the attack is an overstatement that cannot be supported. [...] Then, the authors of 2015/313 refer to ongoing research by Fang Song and myself. This work in progress [...] has never been shared with the authors and has never been publicly released. [...] Fang Song and myself do not refer to it as an actual result.

Despite all debate about the new attacks and, more specifically, about their complexity, there seems to be a consensus that they apply to cyclotomics of prime-power index (or at least of powers of two). Unfortunately, as we have already shown, this includes the negacyclic ideals used in Smart-Vercauteren (SMART; VERCAUTEREN, 2010) and LMSV (LOFTUS et al., 2012). On the bright side, according to Cramer *et al.*, the attack does not seem to apply to schemes based on Ring-LWE (LYUBASHEVSKY; PEIKERT; REGEV, 2013), which means that the attack does not apply to many other currently known lattice-based cryptosystems. Moreover, there is no known extension, to date, that generalizes the attack to other non-cyclotomic rings. In other words, this means that not *all* of lattice-based cryptography is broken, only a subset of existing schemes. However, all homomorphic encryption based on these other methods are, in practice, broken, for other reasons discussed in Section 4.2.

[1]The referred *"authors of 2015/313"* are (CRAMER et al., 2015).

Considering that the choice of ring greatly influences security (PEIKERT, 2015), our method from Chapter 3 can be improved by making simple modifications to our original proposition. In what follows, we will show an alternative approach to thwart these attacks, including the new quantum algorithm.

4.5 Improving security by choosing a different lattice

The method presented in Chapter 3 can be modified to improve security against the attacks discussed in Section 4.4, without great impact on performance, as shown in Chapter 5. Due to security concerns with the notion of working on a ring (where not all nonzero elements have inverses), Bernstein (BERNSTEIN, 2014) suggests adopting a number field instead, specifically a field of form $\mathbb{Z}[x]/(x^n - x - 1)$ because of the very simple form of the irreducible polynomial $x^n - x - 1$, which yields nearly circulant matrices and fairly efficient arithmetic. Matrices in this field are of the form

$$M = \begin{bmatrix} a_0 & a_1 & a_2 & \dots & a_{n-2} & a_{n-1} \\ a_{n-1} & a_0 + a_{n-1} & a_1 & \dots & a_{n-3} & a_{n-2} \\ a_{n-2} & a_{n-1} + a_{n-2} & a_0 + a_{n-1} & \dots & a_{n-4} & a_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_2 & a_3 + a_2 & a_4 + a_3 & \dots & a_0 + a_{n-1} & a_1 \\ a_1 & a_2 + a_1 & a_3 + a_2 & \dots & a_{n-1} + a_{n-2} & a_0 + a_{n-1} \end{bmatrix}.$$

Unfortunately, in this case the method for calculating d presented in Chapter 3 is not applicable, as there is no known method for easily constructing fields $\mathbb{F}_{q_k}$ with appropriate q_k for the Chinese remainder theorem such that the splitting fields of $x^n - x - 1$ have small degree. This does not allow for efficient diagonalization, i.e., with a cost of $O(n \lg n)$ *light* operations (if the splitting fields have large degree, the operations are too expensive, and the whole process becomes worse than using the generic method). For this reason, we calculate d in the same way as Gentry-Halevi, that is, by calculat-

ing the resultant of $p(x)$ and the polynomial of the secret key. However, to increase efficiency, we suggest calculating it in enough fields $\mathbb{F}_{q_k}$ and recover its value via the CRT.

The remaining problem, then, is to calculate the value of u. In this case, the same method can still be applied, that is, finding a matrix M such that $dM = H^* A^\dagger$, except that, in this case,

$$H^* = \begin{bmatrix} 1 & u & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & u & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & u & 0 & 0 \\ 0 & 0 & 0 & \dots & 1 & u & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & u \\ u & u & 0 & \dots & 0 & 0 & 1 \end{bmatrix}.$$

Thus, if any solution to this equation exists, it is still $u = -A_j^\dagger / A_{j-1 \pmod n}^\dagger \bmod d$ for $j \neq 1$. For $j = 1$, we have that $u = -A_1^\dagger / (A_0^\dagger + A_{n-1}^\dagger) \bmod d$. The adjoint can be computed via the CRT from $A^\dagger = dA^{-1} \bmod q_k$, and the extended Euclidean algorithm then yields

$$u \leftarrow -A_0^\dagger / A_{n-1}^\dagger \bmod d. \tag{4.6}$$

For $k \in \{2, \dots, n-1\}$, elements of the adjoint can be written as

$$A_{n-k}^\dagger = A_{n-1}^\dagger (-u)^{-k+1} \bmod d. \tag{4.7}$$

These equations are enough to calculate all necessary parameters for working in $\mathbb{Z}[x]/(x^n - x - 1)$. If compared to the circulant rings suggested in previous chapters, this ring allows for competitive performance, as shown in Chapter 5. An interesting property is as follows:

Lemma 4. $(-u)^{-1} \bmod d$ *is a root of* $x^n - x - 1$, *for every odd* n.

Proof. We know that $A_1^\dagger = -(A_0^\dagger + A_{n-1}^\dagger)u \bmod d$. If we substitute it in Equation 4.7, we have that $A_{n-1}^\dagger(-u)^{-n+1} = (A_0^\dagger + A_{n-1}^\dagger) \bmod d$, and as a direct consequence, $A_0^\dagger = A_{n-1}^\dagger(-1 + (-u)^{-n+1}) \bmod d$. Substituting that in Equation 4.6 gives

$$\begin{aligned} 0 &= A_0^\dagger + A_0^\dagger(-u)^{-1}(-1 + (-u)^{-n+1}) \bmod d \\ &= 1 - u^{-1}(-1 + (-u)^{-n+1}) \bmod d \\ &= 1 + u^{-1} - (-u)^{-n} \bmod d \\ &= (-u)^{-n+1} - u - 1 \bmod d \end{aligned}$$

which, for odd values of n, is the same as $u^{-n+1} - u - 1 = 0 \bmod d$, and multiplied by u^{-1} is $(u^{-1})^n - u^{-1} - 1 = 0 \bmod d$. □

4.6 Synopsis

In this chapter, we have discussed the security of our proposed method. Basic concepts have been introduced and known IND-CCA1 key recovery attacks against homomorphic cryptosystems have been enumerated. We have shown how the security level of the Smart-Vercauteren construction is still below current security standards for practical values of n (which also yield low levels of homomorphism). We have analyzed an apparent key recovery attack against the circulant ring $\mathbb{Z}[x]/(x^n - 1)$, demonstrating that it leaks at most $O(\lg n)$ bits of information on the private key. Finally, we have presented the controversy surrounding the recently published quantum attacks against the Soliloquy and Smart-Vercauteren schemes, indicating how our method can be improved to effectively curb this quantum threat.

5 IMPLEMENTATION RESULTS

We have implemented the methods discussed in previous chapters, and in this chapter we report the results obtained. For GGH-YK-M, our alternative reduces public key bandwidth occupation by an order of complexity, specifically, from $O(n^2 \lg n)$ down to $O(n \lg n)$, where n is a public parameter of the scheme. The new technique also attains faster processing in all operations involved in a public-key cryptosystem, that is, key generation, encryption, and decryption. By far the most pronounced improvement in performance is in key generation, which becomes more than 3 orders of magnitude faster than published results (BARROS; SCHECHTER, 2014), while encryption becomes about 2 orders of magnitude faster. For decryption, our implementation is ten times faster than the literature. For LMSV (LOFTUS et al., 2012), we compare the cyclotomic ring $\mathbb{Z}[x]/(x^n - 1)$ and the more secure irreducible ring $\mathbb{Z}[x]/(x^n - x - 1)$. Performance of encryption remains virtually the same, and decryption becomes slightly worse. Key generation, however, is much slower, due to the fact that it is necessary to use a more generic and expensive method.

The times needed to gather suitable primes for the Chinese remainder theorem are not included since they are precomputed only once and stored. When n is a power of two, the Fast Fourier transform (FFT) is available for both the circulant and negacyclic cases. Considering that their performance is very similar, we opt to only present the circulant case, However, as previously discussed, cyclotomic lattices are not secure. For this reason, we also present tests with $\mathbb{Z}[x]/(x^n - x - 1)$ and prime n, as suggested by Bernstein (BERNSTEIN, 2014). Unfortunately, in this case there is no known technique

for diagonalizing matrices and a slower generic method must be used. This negatively impacts performance of the key generation algorithm.

5.1 Tests with GGH-YK-M

In tests with the GGH-YK-M encryption scheme, we disregard lattice dimensions smaller than 350, since they are susceptible to attacks (BARROS; SCHECHTER, 2014), and we set n to be either a prime or a power of 2. We provide data for dimensions around 512 as well, going somewhat beyond the dimensions found in that reference.

By design, we only consider private keys whose HNF is minimal. To this end, we adopted a rejection sampling strategy, generating uniformly random private keys and discarding those that do not satisfy the desired property, until finding one that does.

Table 3: Timings for GGH-YK-M (in seconds)

source	(n, σ, h, k)	keygen (s)	encrypt (ms)	decrypt (ms)
previous[1]	$(350, 256, 526, 64)$	368.16	13.3	37.6
Java	$(353, 256, 526, 64)$	0.34	2.7	55.2
C	$(353, 256, 526, 64)$	0.10	0.1	4.8
previous[1]	$(400, 256, 601, 64)$	692.48	15.5	59.8
Java	$(401, 256, 601, 64)$	0.46	2.0	67.9
C	$(401, 256, 601, 64)$	0.12	0.1	5.3
Java	$(509, 256, 769, 80)$	1.04	4.0	166.3
C	$(509, 256, 769, 80)$	0.22	0.2	9.4
Java	$(512, 256, 769, 80)$	3.01	2.4	124.5
C	$(512, 256, 769, 80)$	0.11	0.2	9.8

[1] (BARROS; SCHECHTER, 2014).

We have implemented the improved encryption scheme in two ways: Java and C, both running on an Intel i5-2450M 2.5 GHz platform under 64-bit Ubuntu 14.10. To facilitate comparison with the literature (BARROS; SCHECHTER, 2014), where timings, obtained from an implementation in C/C++, are only available on an AMD E-350 1.6 GHz platform, their results are scaled by a factor $3414/756$ on Table 3, corresponding to the benchmark difference between the two processors[1]. Performance turned out to

[1] Obtained from *http://www.cpubenchmark.net* on 05/10/2015.

be highly competitive with the prior state of the art.

Table 4: Public key sizes for GGH-YK-M (in bits)

source	(n, σ, h, k)	$\lvert pk \rvert$
previous[1]	$(350, 256, 526, 64)$	1157800
ours	$(353, 256, 526, 64)$	6682
previous[1]	$(400, 256, 601, 64)$	1543200
ours	$(401, 256, 601, 64)$	7738
previous[1] †	$(512, 256, 769, 80)$	2621440
ours	$(512, 256, 769, 80)$	10240

[1] (BARROS; SCHECHTER, 2014).

† Inferred.

Interestingly, prime values of n tend to yield lattices with minimal HNF far more often than composite n. Empirically, the probability that a random circulant matrix A has a minimal HNF is heavily affected by the choice of lattice parameters, particularly its dimension n, being roughly $O(1/D)$ where D is the number of irreducible factors of $x^n - 1$. Tourloupis (TOURLOUPIS, 2013) addresses this issue (for a generic matrix A, not necessarily circulant) by sieving the randomly sampled A to have prime or near-prime determinant, thus ensuring that it has a 99% probability of sporting a minimal HNF. However, choosing n itself to be prime increases that probability to the same level (since the number of irreducible factors of $x^n - 1$ coincide with the number of factors of n), without having to resort to primality testing during key generation. This behavior is only counterbalanced for composite n when the FFT is available, in which case processing is fast enough to roughly compensate for the rejection sampling overhead. This can be noticed when comparing Java results with $n = 509$ and $n = 512$.

The same effect is not observed in C. This can be explained if we take into consideration some implementation details. In our C implementation, multiplicative inverses are calculated using the extended Euclidean algorithm implementation provided by RELIC Toolkit (ARANHA; GOUVÊA, 2013). In Java, the same calculation is done using the *modInverse* method of the *BigInteger* class. This procedure is optimized in native code for the Java Virtual Machine, whereas the same operation is not available in C, at least not in an optimized implementation. As a result, in cases when n is a power

of 2 (and the fast Fourier transform is available), the efficiency gain by avoiding the calculation of multiplicative inverses is much more dramatic in C than it is in Java.

Key sizes are essentially the same in our proposal for a given dimension n regardless of the choice of $p(x)$. Sample public key sizes are listed on Table 4.

5.2 Tests with LMSV

Tests with LMSV aimed at evaluating the impact on performance caused by adopting the more secure $\mathbb{Z}[x]/(x^n - x - 1)$ ring. The scheme was implemented in Java on an Intel i5-2450M 2.5 GHz platform under 64-bit Ubuntu 14.10, and the results are shown in Table 5. For encryption and decryption, both cases yield similar efficiency. For key generation, however, the circulant case is much better. As explained in Section 4.5, there is no known method for efficiently diagonalizing matrices in the ring $\mathbb{Z}[x]/(x^n - x - 1)$, which needs two expensive computations: the resultant of polynomials and the Euclidean algorithm. This is much more expensive than the technique applied in the circulant case, based on Vandermonde matrices (see Section 3.1).

Table 5: Timings for LMSV

$P(x)$	n	keygen (s)	encrypt (ms)	decrypt (ms)
$x^n - 1$	353	0.3	1.5	41.1
$x^n - x - 1$	353	15.3	1.8	45.0
$x^n - 1$	401	0.4	2.5	60.1
$x^n - x - 1$	401	20.3	2.4	64.8
$x^n - 1$	509	1.4	7.2	129.9
$x^n - x - 1$	509	42.8	7.3	134.0
$x^n - 1$	2039	100.3	390.0	10.4×10^3
$x^n - 1$	8191	7.3×10^3	18.3×10^3	879×10^3

Despite being a widely used and throughly available platform, Java provides only limited support for cryptographic applications. Firstly, there is no guarantee that implementations are isochronous on all available Java Virtual Machines (JVMs), a potential vulnerability that could be exploited in side-channel attacks. Java does not provide a method for efficient multiplication of big and small integers, demanding conversions

to the more expensive *BigInteger* representation. Also, there is no method to calculate, at once, the quotient and modular remainder of an integer division (the only similar method yields a signed remainder that behaves like the "%" operator, not like the *mod()* method). As a result, it is necessary to compute essentially the same operation twice. But maybe the biggest hindrance is due to the fact that *BigInteger* objects are immutable. This means that new instances are created for each and every arithmetic operation. When a large amount of such operations is executed, a significant impact on performance is observed, caused not by the arithmetic calculations themselves, but by the expensive memory management associated to repeated object instantiation. Analyzing the results presented in Table 5, it is clear that for $n > 509$ these limitations distort the obtained results, as timings increase disproportionately fast.

Implementing the algorithm in C or C++, for instance, could help achieve better performance. It would not, however, address the main issue, that is, the complexity of key generation in $\mathbb{Z}[x]/(x^n - x - 1)$. In particular, the Karatsuba algorithm (KARATSUBA; OFMAN, 1963) can be used for faster polynomial multiplication, as it is $O(n^{\lg 3})$, against $O(n^2)$ of the naive method. Very good implementations of the Euclidean algorithm for calculating the GCD of polynomials can also attain slightly lower complexity. However, all of these methods introduce other overheads, such as memory management of recursive function calls. It is then an open question whether the lower *asymptotic* complexity generates an appreciable improvement for the rather small values of n that are used in practice. Furthermore, at best, these improvements would have a limited impact in the execution time of the algorithm. Hence, the issue of poor performance in $\mathbb{Z}[x]/(x^n - x - 1)$ is not simply a matter of fine-tuning the implementation, but still an algebraic problem.

5.3 Synopsis

We have presented the obtained results for both cryptosystems, considering the circulant ring $\mathbb{Z}[x]/(x^n - 1)$ for many values of n and the more secure polynomial ring $\mathbb{Z}[x]/(x^n - x - 1)$. It is important to note, however, that even though the circulant ring has been tested, we strongly recommend *against* using it in real application scenarios, as IND-CCA-1 key recovery attacks have been described for all cyclotomic rings. Those results are presented purely as a performance assessment. Finally, we have also shown that key generation in the irreducible ring $\mathbb{Z}[x]/(x^n - x - 1)$ can still be greatly improved.

6 CONCLUSION

We have shown a way to improve the efficiency and security of cryptosystems based on GGH-style lattices. Our contributions stem from the technique first put forward by Smart and Vercauteren, which we optimize in a simpler and more efficient way than the Gentry-Halevi method. Our new method has been tested with two different schemes. The first one is the GGH-YK-M scheme by Barros and Schechter, and we were able to reduce its public key size by an order of complexity from $O(n^2 \lg n)$ down to $O(n \lg n)$ bits. As a result, key generation times decrease as compared to the Barros-Schechter variant by more than 3 orders of magnitude. Besides the key generation speedup, encryption becomes almost 2 orders of magnitude faster; decryption is about ten times faster though the reason for the improvement in this particular operation could be simply related to different implementation details.

Another concern is related to the security of the Smart-Vercauteren family of homomorphic cryptosystems. The British GCHQ recently published a quantum attack that is able to recover the private key of these constructions, and we have shown that our method can be adapted to prevent this attack. In our tests with the LMSV scheme, we adopt an irreducible polynomial ring that yields nearly circulant matrices and compare it with the circulant case. As a result, encryption remains virtually the same, and decryption becomes only slightly worse. Key generation, though, is much slower, due to the fact that there is no dedicated algorithm for diagonalizing matrices, forcing the use of a more generic (and more expensive) method. Our benchmarks were obtained using Java, and suffered with some of the limitations of this platform when working

with greater values of n. A C/C++ implementation is likely to improve performance, but it certainly would not be enough to solve the issue with key generation, as this is not a matter of fine-tuning the implementation, but still an open algebraic problem. The research of dedicated methods for improving the efficiency of key generation in $\mathbb{Z}[x]/(x^n - x - 1)$ is devised as a possibility of continuation of this work. In particular, a method for constructing fields $\mathbb{F}_{q_k}$ with appropriate q_k for the Chinese remainder theorem such that the splitting fields of $p(x)$ have small degree, allowing for efficient diagonalization with a cost of $O(n \lg n)$ *light* operations.

REFERENCES

ADLEMAN, L. A subexponential algorithm for the discrete logarithm problem with applications to cryptography. In: *XX Annual Symposium on Foundations of Computer Science*. San Juan, Puerto Rico: IEEE Computer Society, 1979. (SFCS '79), p. 55–60. ISSN 0272-5428.

AJTAI, M. Generating hard instances of lattice problems. In: *XXVIII Annual ACM Symposium on Theory of Computing*. New York, NY, USA: ACM, 1996. p. 99–108.

APOSTOL, T. M. Resultants of cyclotomic polynomials. *American Mathematical Society*, v. 24, n. 3, p. 457–462, 1970.

ARANHA, D. F.; GOUVÊA, C. P. L. *RELIC is an Efficient LIbrary for Cryptography*. 2013. Available at: <http://github.com/relic-toolkit/relic>. Accessed: Monday 5th October, 2015.

BABAI, L. On Lovász'lattice reduction and the nearest lattice point problem. *Combinatorica*, Springer, v. 6, n. 1, p. 1–13, 1986.

BARGUIL, J. M. M.; BARRETO, P. S. L. M. Security issues in Sarkar's e-cash protocol. *Information Processing Letters*, Elsevier, v. 115, n. 11, p. 801–803, 2015. ISSN 0020-0190.

BARGUIL, J. M. M.; LINO, R. Y.; BARRETO, P. S. L. M. Efficient variants of the GGH-YK-M cryptosystem. In: *Brazilian Symposium on Computer System and Information Security – SBSeg 2014*. Belo Horizonte, Brazil: SBC, 2014.

BARROS, C. F. de; SCHECHTER, L. M. GGH may not be dead after all. In: *XXXV Congresso Nacional de Matemática Aplicada e Computacional – CNMAC 2014*. Natal, Brazil: Sociedade Brasileira de Matemática Aplicada e Computacional – SBMAC, 2014.

BERMAN, A.; PLEMMONS, R. J. Nonnegative matrices in the mathematical sciences. *Classics in Applied Mathematics*, Society for Industrial and Applied Mathematics (SIAM), v. 9, 1994.

BERNSTEIN, D. J. *A subfield-logarithm attack against ideal lattices*. February 2014. Blog entry. Available at: <http://blog.cr.yp.to/20140213-ideal.html>. Accessed: Monday 5th October, 2015.

BERNSTEIN, D. J.; BUCHMANN, J.; DAHMEN, E. *Post-Quantum Cryptography*. Heidelberg, Deutschland: Springer, 2008.

BIASSE, J.-F. Subexponential time relations in the class group of large degree number fields. *Advances in Mathematics of Communications*, v. 8, n. 4, 2014.

BIASSE, J.-F. *Quantum attacks against the short PIP problem are not polynomial time*. 2015. IACR Cryptology ePrint Archive, Discussion Forum. Available at: <http://eprint.iacr.org/forum/read.php?16,945>. Accessed: Monday 5th October, 2015.

BIASSE, J.-F.; FIEKER, C. Subexponential class group and unit group computation in large degree number fields. *LMS Journal of Computation and Mathematics*, Cambridge University Press, v. 17, n. A, p. 385–403, 2014.

BIHAM, E.; SHAMIR, A. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*, Springer, v. 4, n. 1, p. 3–72, 1991.

BOLDRINI, J.; COSTA, S.; RIBEIRO, V.; WETZLER, H. *Álgebra Linear*. São Paulo, Brazil: Harper & Row do Brasil, 1980.

BOS, J. W.; LAUTER, K.; LOFTUS, J.; NAEHRIG, M. Improved security for a ring-based fully homomorphic encryption scheme. In: *XIV IMA International Conference on Cryptography and Coding*. Oxford, England: Springer Berlin Heidelberg, 2013. p. 45–64.

BRAKERSKI, Z. Fully homomorphic encryption without modulus switching from classical GapSVP. In: *Advances in Cryptology – CRYPTO 2012*. Santa Barbara, USA: Springer, 2012. p. 868–886.

BRAKERSKI, Z.; GENTRY, C.; VAIKUNTANATHAN, V. (Leveled) fully homomorphic encryption without bootstrapping. In: *III Innovations in Theoretical Computer Science Conference – ITCS 2012*. Cambridge, MA, USA: ACM, 2012. (ITCS '12), p. 309–325.

BRAKERSKI, Z.; VAIKUNTANATHAN, V. Efficient fully homomorphic encryption from (standard) LWE. *SIAM Journal on Computing*, SIAM, v. 43, n. 2, p. 831–871, 2014.

CAMPBELL, P.; GROVES, M.; SHEPHERD, D. Soliloquy: A cautionary tale. In: *ETSI 2nd Quantum-Safe Crypto Workshop*. Ottawa, Canada: ETSI, 2014.

CHAUM, D. Blind signatures for untraceable payments. In: *Advances in Cryptology – CRYPTO '82*. Santa Barbara, USA: Plenum, 1983. p. 199–203.

CHENAL, M.; TANG, Q. On key recovery attacks against existing somewhat homomorphic encryption schemes. In: *International Conference on Cryptology and Information Security in Latin America – Latincrypt 2014*. Florianópolis, Brazil: Springer, 2014. (Lecture Notes in Computer Science).

CHENAL, M.; TANG, Q. Key recovery attack against an NTRU-type somewhat homomorphic encryption scheme. *IACR Cryptology ePrint Archive*, v. 2015, p. 83, 2015. Available at: <http://eprint.iacr.org/2015/083>. Accessed: Monday 5th October, 2015.

COHEN, H. *A course in computational algebraic number theory*. Berlin, Germany: Springer, 1993.

CRAMER, R.; DUCAS, L.; PEIKERT, C.; REGEV, O. *Recovering Short Generators of Principal Ideals in Cyclotomic Rings*. 2015. IACR Cryptology ePrint Archive, Report 2015/313. Available at: <http://eprint.iacr.org/2015/313>. Accessed: Monday 5th October, 2015.

DAEMEN, J.; RIJMEN, V. AES proposal: Rijndael. 1998.

D'AGAPEYEFF, A. *Codes and Ciphers - A History Of Cryptography*. Oxford, UK: Oxford University Press, 1939.

DAHAB, R.; GALBRAITH, S.; MORAIS, E. Adaptive key recovery attacks on NTRU-based somewhat homomorphic encryption schemes. In: *Information Theoretic Security*. Lugano, Switzerland: Springer, 2015. p. 283–296.

DIERKS, T.; RESCORLA, E. *The Transport Layer Security (TLS) protocol version 1.2*. 2008. Available at: <http://www.ietf.org/rfc/rfc5246.txt>. Accessed: Monday 5th October, 2015.

DIFFIE, W.; HELLMAN, M. E. New directions in cryptography. *Information Theory, IEEE Transactions on*, IEEE, v. 22, n. 6, p. 644–654, 1976.

DURUMERIC, Z.; KASTEN, J.; BAILEY, M.; HALDERMAN, J. A. Analysis of the HTTPS certificate ecosystem. In: *Internet Measurement Conference 2013*. Barcelona, Spain: ACM, 2013. p. 291–304.

EISENTRÄGER, K.; HALLGREN, S.; KITAEV, A.; SONG, F. A quantum algorithm for computing the unit group of an arbitrary degree number field. In: *XLVI Annual ACM Symposium on Theory of Computing – STOC 2014*. New York, NY, USA: ACM, 2014. p. 293–302.

ELGAMAL, T. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, v. 31, n. 4, p. 469, 1985.

FUJISAKI, E.; OKAMOTO, T. Secure integration of asymmetric and symmetric encryption schemes. In: *Advances in Cryptology – CRYPTO '99*. Santa Barbara, USA: Springer, 1999. (Lecture Notes in Computer Science, v. 1666), p. 537–554.

GARG, S.; GENTRY, C.; HALEVI, S. Candidate multilinear maps from ideal lattices. In: *Advances in Cryptology – EUROCRYPT 2013*. Athens, Greece: Springer, 2013. v. 7881, p. 1–17.

GARLING, D. J. H. *Inequalities: A Journey into Linear Analysis*. Cambridge, UK: Cambridge University Press, 2007.

GENTRY, C. *A fully homomorphic encryption scheme*. Thesis (PhD) — Stanford University, 2009.

GENTRY, C. Fully homomorphic encryption using ideal lattices. In: *XLI Annual ACM Symposium on Theory of Computing – STOC 2009*. Bethesda, MD, USA: ACM, 2009. v. 9, p. 169–178.

GENTRY, C.; HALEVI, S. Implementing Gentry's fully-homomorphic encryption scheme. In: *Advances in Cryptology – EUROCRYPT 2011*. Tallinn, Estonia: Springer, 2011. (Lecture Notes in Computer Science, v. 6632), p. 129–148.

GENTRY, C.; PEIKERT, C.; VAIKUNTANATHAN, V. Trapdoors for hard lattices and new cryptographic constructions. In: *XL Annual ACM Symposium on Theory of Computing – STOC '08*. Victoria, Canada: ACM, 2008. (Lecture Notes in Computer Science), p. 197–206.

GENTRY, C.; SAHAI, A.; WATERS, B. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: *Advances in Cryptology – CRYPTO 2013*. Santa Barbara, USA: Springer, 2013. p. 75–92.

GOLDREICH, O. *Foundations of cryptography: volume 2, basic applications*. Cambridge, UK: Cambridge University Press, 2004.

GOLDREICH, O.; GOLDWASSER, S.; HALEVI, S. Public-key cryptosystems from lattice reduction problems. In: *Advances in Cryptology – CRYPTO '97*. Santa Barbara, USA: Springer, 1997. p. 112–131.

GOLDWASSER, S.; MICALI, S. Probabilistic encryption & how to play mental poker keeping secret all partial information. In: *XIV Annual ACM Symposium on Theory of Computing – STOC '82*. San Francisco, CA, USA: ACM, 1982. p. 365–377.

GOLDWASSER, S.; MICALI, S. Probabilistic encryption. *Journal of computer and system sciences*, Elsevier, v. 28, n. 2, p. 270–299, 1984.

GÜNEYSU, T.; LYUBASHEVSKY, V.; PÖPPELMANN, T. Practical lattice-based cryptography: A signature scheme for embedded systems. In: *Cryptographic Hardware and Embedded Systems – CHES 2012*. Leuven, Belgium: Springer Berlin Heidelberg, 2012, (Lecture Notes in Computer Science, v. 7428). p. 530–547. ISBN 978-3-642-33026-1.

HÄKKINEN, K. *Nykysuomen etymologinen sanakirja*. Juva: WSOY, 2004.

HOFFSTEIN, J.; HOWGRAVE-GRAHAM, N.; PIPHER, J.; SILVERMAN, J.; WHYTE, W. NTRUSign: Digital signatures using the NTRU lattice. In: *Topics in Cryptology – CT-RSA 2003*. California, USA: Springer Berlin Heidelberg, 2003, (Lecture Notes in Computer Science, v. 2612). p. 122–140.

HOFFSTEIN, J.; PIPHER, J.; SILVERMAN, J. NTRU: A ring-based public key cryptosystem. In: *Algorithmic Number Theory*. Oregon, USA: Springer Berlin Heidelberg, 1998, (Lecture Notes in Computer Science, v. 1423). p. 267–288.

HOFFSTEIN, J.; PIPHER, J.; SILVERMAN, J. NSS: An NTRU lattice-based signature scheme. In: *Advances in Cryptology – EUROCRYPT 2001*. Innsbruck, Austria: Springer, 2001, (Lecture Notes in Computer Science, v. 2045). p. 211–228. ISBN 978-3-540-42070-5.

HORN, R. A.; JOHNSON, C. R. *Matrix analysis*. Cambridge, UK: Cambridge University Press, 2012.

IEEE P1363 Working Group. *Standard Specifications for Public-Key Cryptography – IEEE Std 1363-2000*. 2000.

JOHNSON, D.; MENEZES, A.; VANSTONE, S. The elliptic curve digital signature algorithm (ECDSA). *International Journal of Information Security*, Springer, v. 1, n. 1, p. 36–63, 2001.

JOUX, A.; STERN, J. Lattice reduction: A toolbox for the cryptanalyst. *Journal of Cryptology*, Springer, v. 11, n. 3, p. 161–185, 1998.

KARATSUBA, A.; OFMAN, Y. Multiplication of multidigit numbers on automata. v. 7, p. 595, 1963.

LANGLOIS, A.; STEHLÉ, D.; STEINFELD, R. Gghlite: More efficient multilinear maps from ideal lattices. In: *Advances in Cryptology – EUROCRYPT 2014*. Copenhagen, Denmark: Springer, 2014. p. 239–256.

LEE, C. *Litecoin*. 2011. Available at: <http://litecoin.info/Litecoin>. Accessed: Monday 5th October, 2015.

LENSTRA, A. K.; LENSTRA, H. W.; LOVÁSZ, L. Factoring polynomials with rational coefficients. *Mathematische Annalen*, Springer, v. 261, n. 4, p. 515–534, 1982.

LIDL, R.; NIEDERREITER, H. *Finite fields*. Cambridge, UK: Cambridge University Press, 1997.

LINDNER, R.; PEIKERT, C. Better key sizes (and attacks) for LWE-based encryption. In: *Topics in Cryptology – CT-RSA 2011*. San Francisco, CA, USA: Springer, 2011. (Lecture Notes in Computer Science, v. 6558), p. 319–339.

LOFTUS, J.; MAY, A.; SMART, N. P.; VERCAUTEREN, F. On CCA-secure somewhat homomorphic encryption. In: *International Conference on Selected Areas in Cryptography – SAC 2011*. Toronto, Canada: Springer, 2012. (Lecture Notes in Computer Science, v. 7118), p. 55–72.

LÓPEZ-ALT, A.; TROMER, E.; VAIKUNTANATHAN, V. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In: *XLIV Annual ACM Symposium on Theory of Computing – STOC 2012*. New York, USA: ACM, 2012. p. 1219–1234.

LYUBASHEVSKY, V. Lattice signatures without trapdoors. In: *Advances in Cryptology – EUROCRYPT 2012*. Cambridge, UK: Springer, 2012. (Lecture Notes in Computer Science, v. 7237), p. 738–755.

LYUBASHEVSKY, V.; MICCIANCIO, D. Generalized compact knapsacks are collision resistant. In: *XXXIII International Colloquium on Automata, Languages and Programming – ICALP 2006*. Venice, Italy: Springer, 2006. (Lecture Notes in Computer Science, v. 4052), p. 144–155.

LYUBASHEVSKY, V.; PEIKERT, C.; REGEV, O. On ideal lattices and learning with errors over rings. *Journal of the ACM (JACM)*, ACM, v. 60, n. 6, p. 43, 2013.

MAO, W. Lightweight micro-cash for the internet. In: *IV European Symposium on Research in Computer Security – ESORICS '96*. Rome, Italy: Springer, 1996. (Lecture Notes in Computer Science, v. 1146), p. 15–32.

MARKUS, W. *Dogecoin*. 2013. Available at: <http://dogecoin.com/>. Accessed: Monday 5th October, 2015.

MCELIECE, R. J. A public-key cryptosystem based on algebraic coding theory. *DSN progress report*, v. 42, n. 44, p. 114–116, 1978.

MICCIANCIO, D. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. In: *XLIII Annual IEEE Symposium on Foundations of Computer Science – 2002*. Vancouver, BC, Canada: IEEE, 2002. p. 356–365.

MICCIANCIO, D. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, Springer, v. 16, n. 4, p. 365–411, 2007.

MICCIANCIO, D.; GOLDWASSER, S. *Complexity of Lattice Problems: a cryptographic perspective*. Boston, MA, USA: Kluwer Academic Publishers, 2002. (The Kluwer International Series in Engineering and Computer Science, v. 671).

MICCIANCIO, D.; VADHAN, S. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In: *Advances in Cryptology – CRYPTO 2003*. Santa Barbara, USA: Springer Berlin Heidelberg, 2003, (Lecture Notes in Computer Science, v. 2729). p. 282–298. ISBN 978-3-540-40674-7.

MICCIANCIO, D.; WARINSCHI, B. A linear space algorithm for computing the Hermite normal form. In: *International Symposium on Symbolic and Algebraic Computation – ISSAC 2001*. London, Canada: ACM, 2001. p. 231–236.

MIERS, I.; GARMAN, C.; GREEN, M.; RUBIN, A. D. Zerocoin: Anonymous distributed e-cash from bitcoin. In: *XXXIV IEEE Symposium on Security and Privacy – S&P 2013*. San Francisco, CA, USA: IEEE, 2013. p. 397–411.

NAEHRIG, M.; LAUTER, K.; VAIKUNTANATHAN, V. Can homomorphic encryption be practical? In: *III ACM Workshop on Cloud Computing Security Workshop*. Chicago, IL, USA: ACM, 2011. p. 113–124.

NAKAMOTO, S. *Bitcoin: A Peer-to-Peer Electronic Cash System*. 2008. Available at: <http://bitcoin.org/bitcoin.pdf>. Accessed: Monday 5th October, 2015.

NGUYEN, P. Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from CRYPTO '97. In: *Advances in Cryptology – CRYPTO '99*. Santa Barbara, USA: Springer, 1999. p. 288–304.

NGUYEN, P.; REGEV, O. Learning a parallelepiped: Cryptanalysis of ggh and ntru signatures. In: *Advances in Cryptology - EUROCRYPT 2006*. Saint Petersburg, Russia: Springer, 2006, (Lecture Notes in Computer Science, v. 4004). p. 271–288. ISBN 978-3-540-34546-6.

NGUYEN, P. Q.; STERN, J. The two faces of lattices in cryptology. In: *Cryptography and lattices*. Providence, RI, USA: Springer, 2001. p. 146–180.

ODLYZKO, A. M. The rise and fall of knapsack cryptosystems. *Cryptology and computational number theory*, v. 42, p. 75–88, 1990.

PAILLIER, P. Public-key cryptosystems based on composite degree residuosity classes. In: *Advances in cryptology – EUROCRYPT 1999*. Prague, Czech Republic: Springer, 1999, (Lecture Notes in Computer Science, v. 1592). p. 223–238. ISBN 978-3-540-65889-4.

PEIKERT, C. *What does GCHQ's "cautionary tale" mean for lattice cryptography?* February 2015. Blog entry. Available at: <http://www.cc.gatech.edu/ cpeikert/soliloquy.html>. Accessed: Monday 5th October, 2015.

PEIKERT, C.; VAIKUNTANATHAN, V.; WATERS, B. A framework for efficient and composable oblivious transfer. In: *Advances in Cryptology – CRYPTO 2008*. Santa Barbara, USA: Springer Berlin Heidelberg, 2008, (Lecture Notes in Computer Science, v. 5157). p. 554–571. ISBN 978-3-540-85173-8.

PRABHAKARAN, M.; ROSULEK, M. Homomorphic encryption with cca security. In: *XXXV International Colloquium on Automata, Languages and Programming – ICALP 2008*. Reykjavik, Iceland: Springer, 2008. p. 667–678.

RADEMACHER, H. *Lectures on elementary number theory*. New York, NY, USA: Blaisdell Publishing Company, 1964.

RADOŠ, M. *Dicionário de Sérvio e Croata-Português e Português-Sérvio e Croata*. Porto, Portugal: Porto Editora, 2003.

REGEV, O. On lattices, learning with errors, random linear codes, and cryptography. In: *XXXVII Annual ACM Symposium on Theory of Computing – STOC 2005*. Baltimore, MD, USA: ACM, 2005. p. 84–93.

RICARDINI, J. E. *Emparelhamentos e reticulados: Estado-da-arte em algoritmos e parâmetros para as famílias mais flexíveis de sistemas criptográficos*. Thesis (MSc) — University of São Paulo, 2014.

RIVEST, R. L.; ADLEMAN, L.; DERTOUZOS, M. L. On data banks and privacy homomorphisms. *Foundations of secure computation*, v. 4, n. 11, p. 169–180, 1978.

RIVEST, R. L.; SHAMIR, A.; ADLEMAN, L. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, ACM, New York, NY, USA, v. 21, n. 2, p. 120–126, 1978.

RÜCKERT, M. Lattice-based blind signatures. In: *Advances in Cryptology – ASIACRYPT 2010*. Singapore, Singapore: Springer, 2010. (Lecture Notes in Computer Science, v. 6477), p. 413–430.

SARKAR, P. Multiple-use transferable e-cash. *International Journal of Computer Applications*, v. 77, n. 6, p. 35–38, 2013. ISSN 0975 8887.

SCHNORR, C.-P. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical computer science*, Elsevier, v. 53, n. 2, p. 201–224, 1987.

SHANNON, C. E. Communication theory of secrecy systems*. *Bell system technical journal*, Wiley Online Library, v. 28, n. 4, p. 656–715, 1949.

SHOR, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, v. 26, p. 1484–1509, 1995.

SHOUP, V. *Sequences of games: a tool for taming complexity in security proofs*. 2004. Manuscript. Revised, May 2005; Jan. 2006.

SMART, N. P. The discrete logarithm problem on elliptic curves of trace one. *Journal of cryptology*, Springer, v. 12, n. 3, p. 193–196, 1999.

SMART, N. P.; VERCAUTEREN, F. Fully homomorphic encryption with relatively small key and ciphertext sizes. In: *XIII International Conference on Practice and Theory in Public Key Cryptography – PKC 2010*. Paris, France: Springer, 2010. (Lecture Notes in Computer Science, v. 6056), p. 420–443.

STANDARDS, N. B. of. *Data Encryption Standard (DES)*. 1977. FIPS publication.

STANDARDS, N. B. of. Digital signature standard (DSS). *National Institute of Standards and Technology (NIST)*, 1994.

STEHLÉ, D.; STEINFELD, R. Faster fully homomorphic encryption. In: *Advances in Cryptology – ASIACRYPT 2010*. Singapore, Singapore: Springer, 2010. p. 377–394.

SZYDLO, M. Hypercubic lattice reduction and analysis of GGH and NTRU signatures. In: *Advances in Cryptology – EUROCRYPT 2003*. Warsaw, Poland: Springer, 2003, (Lecture Notes in Computer Science, v. 2656). p. 433–448.

TOURLOUPIS, V. E. *Hermite normal forms and its cryptographic applications*. Thesis (MSc) — University of Wollongong, 2013.

VAN TILBORG, H. C. Fundamentals of cryptology. *KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE*, KLUWER ACADEMIC PUBLISHERS GROUP, 2000.

VAUDENAY, S. *A Classical Introduction to Cryptography: Applications for Communications Security*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

YLONEN, T.; LONVICK, C. The secure shell (SSH) authentication protocol. 2006.

YOSHINO, M.; KUNIHIRO, N. Improving GGH cryptosystem for large error vector. In: *XI International Symposium on Information Theory and its Applications – ISITA 2012*. Honolulu, HI, USA: IEEE, 2012. p. 416–420.

APPENDIX A – SECURITY ISSUES IN SARKAR'S E-CASH PROTOCOL

In the course of investigating applications of somewhat homomorphic encryption that might benefit from the techniques presented in the main body of this thesis, we briefly assessed some e-cash protocols, in particular Sarkar's (SARKAR, 2013) because of its apparent reliance on ring operations. While that protocol turned out not to be a significant example of a scenario where somewhat homomorphic encryption would be useful *per se* (in the sense of functional rather than security aspects), as a by-product of our analysis we were able to entirely break it, subverting all of its security goals. The results were published as (BARGUIL; BARRETO, 2015), and are herein reproduced.

The notion of digital currency (e-cash) is far from new (CHAUM, 1983), but its practical importance has increased over the past few years, fueled by public awareness of surprisingly successful proposals like Bitcoin (NAKAMOTO, 2008). Regardless of being centralized or peer-to-peer, an ideal e-cash protocol should feature some essential security properties:

1. ***(Unforgeability/integrity).*** No user should be able to forge a coin or to modify the value of an existing coin.

2. ***(Untraceability/privacy).*** No user should be able to extract transaction details or user details from any given coin.

3. ***(Double-spending prevention).*** No user should be able to spend a coin more than once.

Additionally, an e-cash protocol may satisfy some other desirable properties like currency divisibility and low bandwidth occupation. The difficulty to obtain all these requirements often yields protocols that satisfy some but not all of them, thus opening many venues of improvement as new protocols offer more features without sacrificing the strictly required ones. Indeed, the shortcomings of Bitcoin sparked the design of new protocols aiming at enhancing efficiency (LEE, 2011; MARKUS, 2013) and, more importantly, security (MIERS et al., 2013).

A recent protocol designed by P. Sarkar (SARKAR, 2013) has been presented as an alternative to Bitcoin and claims strong security properties, but offers no formal proof of security (e.g., in the well-established form of a sequence of games (SHOUP, 2004)). This situation is risky even in basic cryptographic schemes like plain digital signatures; in a scenario as multifaceted as a full-fledged e-cash protocol, it is critical, and can lead to devastating consequences, as we will argue it does.

In this chapter we perform a complete cryptanalysis of Sarkar's protocol, showing that none of the claimed security properties is effectively satisfied.

Besides constituting further evidence that intricate cryptographic protocols need to be accompanied by formal security proofs, the relative simplicity of Sarkar's protocol and its cryptanalysis could play the role of a clear and realistic counterexample to secure e-cash design, and help preventing similar flaws from finding their way into future protocols.

The remainder of this chapter is organized as follows. In Section A.1 we recapitulate the basics of Sarkar's protocol. We describe our attacks against all security aspects of the protocol in Section A.2. We conclude in Section A.3.

A.1 Sarkar's e-cash protocol

Sarkar's protocol is structurally closer to Mao (MAO, 1996) than to Bitcoin and other distributed cryptocurrencies, since it relies upon a central authority, referred to as the "bank," which is assumed to have a (presumably certified) pair of RSA (RIVEST; SHAMIR; ADLEMAN, 1978) keys (d_u, n_u), (e_u, n_u), which are used respectively to sign and verify each issued coin. For conciseness, we denote by $\mathsf{sign}(x)$ the result of signing some value x into $x^{d_u} \mod n_u$.

When creating an account at the bank, the i-th user obtains a unique identifier I_i. The bank generates a pair of RSA keys (d_i, n_i), (e_i, n_i), revealing only (e_i, n_i) to the user while keeping (d_i, n_i) in its possession for user identification. The protocol requires the user to encrypt her identity within coins under the key (e_i, n_i) at coin transfer transactions. For conciseness, we denote by $\mathsf{encr}_i(x)$ the result of encrypting some value x into $x^{e_i} \mod n_i$.

A coin has the form

$$[A_k, \mathsf{sign}(A_k), F_k, \mathsf{sign}(A_k \oplus F_k), (e_u, n_u), \{\text{User\_History}\}]$$

where A_k is a unique coin identifier and F_k is its value. User_History stores the sequence of transaction records involving the coin, each in the form

$$[S_{ij}, \mathsf{encr}_i(S_{ij} \oplus I_i), (S_{ij} \oplus n_i) \cdot b_{ij}, (n_i \oplus b_{ij})] \tag{A.1}$$

where S_{ij} is called a *receiving-end number* and b_{ij} is called a *spending-end number*.

When withdrawing or receiving a coin in payment, the i-th user randomly generates S_{ij} and encrypts it into User_History using her identity I_i and public key (e_i, n_i). When spending or depositing the coin (to the j-th user or to the bank, respectively), after the j-th recipient already holds the whole coin except the last two components of the transaction record and has checked the bank's signature on the initial coin components,

the i-th user randomly generates b_{ij} and computes the remaining two components of the transaction record, appending them to the User_History.

After m transactions, the coin has been through m receptions and $m-1$ spendings. When the m-th user deposits the coin to the bank, the bank verifies if any other coin with the same identifier A_k has been previously deposited to check for occurrences of double spending.

We point out at this point that Sarkar does not explicitly define the algebraic nature of the dot product involving b_{ij} above. However, Equation 6 of (SARKAR, 2013, Section 10) makes it clear that it must be *distributive* over the exclusive-or to enable the detection of double spending, namely, given two transaction records $[S_{ij}, \mathsf{encr}_i(S_{ij} \oplus I_i), (S_{ij} \oplus n_i) \cdot b_{ij}, (n_i \oplus b_{ij})]$ and $[S_{ij}, \mathsf{encr}_i(S_{ij} \oplus I_i), (S_{ij} \oplus n_i) \cdot b'_{ij}, (n_i \oplus b'_{ij})]$ corresponding to two spendings of the same coin, the modulus n_i can be extracted from them as:

$$\begin{aligned} & ((S_{ij} \oplus n_i) \cdot b_{ij} \oplus (S_{ij} \oplus n_i) \cdot b'_{ij}) \,/\, ((b_{ij} \oplus n_i) \oplus (b'_{ij} \oplus n_i)) \\ = \;& ((S_{ij} \oplus n_i) \cdot (b_{ij} \oplus b'_{ij})) \,/\, (b_{ij} \oplus b'_{ij}) \\ = \;& S_{ij} \oplus n_i \end{aligned}$$

(note the implicit property $\alpha \cdot (b_{ij} \oplus b'_{ij}) = \alpha \cdot b_{ij} \oplus \alpha \cdot b'_{ij}$ with $\alpha := S_{ij} \oplus n_i$). The extracted modulus can then be looked up in a database of private keys (d_i, n_i) to allow recovering I_i from the value $\mathsf{encr}_i(S_{ij} \oplus I_i)$ stored in the transaction records above, thus revealing the identity of the double spender.

Therefore, rather than a plain integer or modular product (which would not be distributive over the exclusive-or operation), the dot must denote the product in the binary polynomial ring $\mathbb{F}_2[x]$ or, more likely, some binary finite field $\mathbb{F}_{2^m}$, so that the multiplicative inverse is defined for all nonzero values.

Sarkar claims (without providing a formal proof) that his proposal satisfies se-

curity requirements 1 through 3, as well as certain functional requirements, such as supporting offline transactions and multiple transferability.

A.2 Attacking the security claims of Sarkar's protocol

We now show that Sarkar's protocol fails to achieve the goals its author set forth. In order, we show that the protocol, in its present form:

1. does not prevent modifying a coin's value,

2. does not protect the users' anonymity, and

3. does not thwart double spending.

A.2.1 Changing the coin value

Suppose an attacker collects two coins

$$\begin{aligned} M_1 &:= [A_1, \mathsf{sign}(A_1), F_1, \mathsf{sign}(A_1 \oplus F_1), \text{User\_History}_1], \\ M_2 &:= [A_2, \mathsf{sign}(A_2), F_2, \mathsf{sign}(A_2 \oplus F_2), \text{User\_History}_2]. \end{aligned}$$

Let $\delta := A_1 \oplus A_2$, and define $F_1' := F_2 \oplus \delta$. With this information alone, the attacker can already violate security requirement 1 (unforgeability/integrity) by existentially forging a coin $M_1' := [A_1, \mathsf{sign}(A_1), F_1', \mathsf{sign}(A_1 \oplus F_1'), \text{User\_History}_1]$, which has the modified value F_1'. This is possible because $\mathsf{sign}(A_1 \oplus F_1')$ is simply $\mathsf{sign}(A_1 \oplus (F_2 \oplus \delta)) = \mathsf{sign}((A_1 \oplus \delta) \oplus F_2) = \mathsf{sign}(A_2 \oplus F_2)$, i.e. a copy of the signature from coin M_2.

Moreover, this attack can easily have practical consequences. As the protocol imposes no restrictions on the value of A, it is plausible that in practice it will take the form of a serial (sequential) number. If so, for coins withdrawn within a reasonably short time interval, δ will be small enough for F_1' to be an admissible and reasonable coin value.

A.2.2 Subverting the traceability of multiple spending

Requirement 3 (double-spending prevention) succumbs by virtue of the following observation: there is no way to verify that the user is really using her identifier I_i, not even at the very moment the coin is spent. The only exception would be if the coin were checked online – but this would trivially violate the user's anonymity by making the transaction traceable (security requirement 2). Besides, it would also subvert one of the functional features of Sarkar's protocol, specifically, its support for offline operation.

Because of this, the j-th coin recipient (which can be any user except the withdrawer herself) could simply omit the last two components of the last User_History entry received from the i-th user, and transfer the truncated coin to any number of other recipients. After those users check the bank's signature, the malicious j-th user delivers the missing two components he received from the i-th user, thereby personifying her since the bank will afterward mistakenly attribute the double spending to the i-th user. Alternatively, the attacker could merely intercept a coin and replace the whole User_History by random values, since the receivers have no way to distinguish them from well-formed ones. The difference is that, in this case, the bank will not be able to trace a double spending to any user whatsoever.

A.2.3 Defeating anonymity altogether

Sarkar's protocol appears to assume tacitly that user account creation is somehow anonymized in the sense that the bank cannot map the identifier I_i to the private key (d_i, n_i). Otherwise, a dishonest bank would be able to bypass untraceability/anonymity by abusing the double-spending detection procedure, that is, by systematically testing all keys against the first two components $[S_{ij}, v_{ij}]$ within each User_History entry of every deposited coin until $(v_{ij}^{d_i} \bmod n_i) \oplus S_{ij} = I_i$, since by definition $v_{ij} := \mathsf{encr}_i(S_{ij} \oplus I_i)$.

As we pointed out, although the protocol does not explicitly define the algebraic

nature of the dot product involving the spending-end number, it is clearly distributive over exclusive-or. Unfortunately, the distributivity destroys anonymity: as the coin contains the values $r_{ij} := b_{ij} \oplus n_i$ and $t_{ij} := (S_{ij} \oplus n_i) \cdot b_{ij}$, any user with access to the coin can substitute $b_{ij} = r_{ij} \oplus n_i$ in the second relation, obtaining a simple quadratic equation in n_i, namely,

$$n_i^2 \oplus (r_{ij} \oplus S_{ij}) \cdot n_i \oplus (r_{ij} \cdot S_{ij} \oplus t_{ij}) = 0.$$

Solving this equation is now a trivial matter (see e.g., (IEEE P1363 Working Group, 2000, Section A.4.7) for the case when the dot product denotes multiplication in a binary field). We point out *en passant* that the same reasoning would still apply if the dot product actually denoted integer multiplication, although it would require replacing the exclusive-ors by additions or subtractions as appropriate.

From there, the bank can trace *the entire path traversed by the coin*, in the exact order it was passed from user to user, since this path is implicit in the sequence of the n_i values within a coin (randomizing this sequence would merely hide the order, but all users involved in the exchanges would still be revealed).

Even more critically, this process does not require any user to double-spend a coin, but simply to deposit it. In this sense, it harms genuine, honest users.

A.3 Conclusion

As we have seen, Sarkar's e-cash protocol fails to attain any of its purported security goals, being easily susceptible to a number of simple attacks. Fixing all the problems is far from being a trivial task, and would seem to require a complete redesign of the protocol, with the side effect that the result would likely become far more complex than it presently is.

Crucially, the protocol entirely lacks a formal security proof, and this should be

the very first aspect to addressed.

To conclude on a positive note, one might argue that both Sarkar's protocol and its cryptanalysis are simple enough (and yet, realistic enough) that they can serve the additional purpose of providing a clear reference counterexample to secure e-cash design, and a source of insight for future robust proposals.