Contributions to design and analysis of Fully Homomorphic Encryption schemes

Francisco Vial Prado

université PARIS-SACLAY

NNT : 2017SACLV107

Thèse de doctorat

UNIVERSITÉ DE VERSAILLES ST-QUENTIN-EN-YVELINES

université PARIS-SACLAY

Contributions to design and analysis of Fully Homomorphic Encryption schemes

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'Université de Versailles Saint-Quentin-en-Yvelines

École doctorale n°580
Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat: Informatique

Thèse présentée et soutenue à Versailles, le 12 juin 2017, par

M. Francisco José Vial Prado

Composition du Jury :

M. Carlos Aguilar-Melchor Maître de Conférences, INP Toulouse	Examinateur
M. Daniel Augot Directeur de recherche, Inria, Centre Inria-Saclay	Président
M. Louis Goubin Professeur à l'Université de Versailles Saint-Quentin-en-Yvelines	Directeur
Mme. Aline Gouget Ingénieur de recherches, Gemalto	Examinatrice
M. Duong Hieu Phan Professeur à l'Université de Limoges	Rapporteur
Mme. Malika Izabachene CEA, LIST	Examinatrice
M. Fabien Laguillaumie Professeur à l'Ecole Normale Supèrieure de Lyon	Rapporteur
M. Pascal Paillier CEO et Seniot Security Expert à CryptoExperts	Examinateur

al Doctor Salvador Vial Urrejola, mi amigo,

Remerciements

Lors de son cours donné au Master d'Algèbre Appliquée à la Cryptographie de l'Université de Versailles, Louis Goubin m'a parlé pendant une pause café des schémas de chiffrement complètement homomorphe. J'avais déjà un haut degré d'admiration pour lui, son cours et sa personnalité, et quand il m'a proposé de travailler sous sa tutelle et sur ce sujet fantastique, ma réponse n'a pas tardé. Sa remarquable politesse, gentillesse, et finesse d'esprit sont couplés à une acuité et à une intuition exceptionnelle. Dès la première rencontre, on ne peut passer à coté de ces qualités, grâce à sa disponibilité et à son honnêteté. J'ai eu l'énorme chance de travailler avec un grand homme, et mon humble gratitude l'accompagnera pendant le reste de ma vie.

Les rapporteurs de ce manuscrit sont Duong Hieu Phan et Fabien Laguillaumie, professeurs à l'Université de Limoges et Lyon respectivement. Je vous remercie pour la lecture de ce manuscrit, et pour vos commentaires encourageants.

Monsieur Jean-Luc Vayssière, ancien Directeur de l'Université, m'a accordé une bourse de thèse exceptionnelle, grâce à laquelle ces travaux ont pû être effectués. Je lui suis reconnaissant d'avoir pris cette décision.

Je remercie l'accueil chaleureux de mes collègues du LMV (ex PRISM): Christina, Illaria, Luca, Nicolas, Michaël, Jacques, Axel, Benoît, Ninon, qui malgré ma présence intermittente ont toujours été très aimables et m'ont aidé dans mes démarches administratives. Mes amis du Master, Marie, Cécile, Mailody, Jean-Yves, et Benoît, merci pour votre convivialité. Tout particulièrement je pense à mon collègue et ami Cyril, avec qui j'ai partagé le bureau 309-A pendant toutes ces années. Merci de ta compagnie et confiance, rarement j'ai connu quelqu'un d'un esprit aussi pur.

Mon arrivée à Versailles fût aussi motivée par la présence de Philippe Cuper. Cinq ans qu'on se connaît, mais je pourrais continuer à apprendre de vous pour des années encore à venir. Merci de m'avoir accepté dans votre classe malgré mon début tardif et mon profil scientifique, et de m'avoir enseigné de façon desintéressée et passionnée. J'ai connu l'essence devant l'esthétique, la beauté de la simplicité, la recherche de la vérité. Vous m'avez tant parlé de Musique, d'Art et de l'Excellence, mais vous avez apporté dans ma vie et dans cette thèse bien plus que vous n'y croyez.

Je vous remercie d'avoir exigé de moi d'aller au-delà de mes limites et de m'avoir encouragé dans les moments difficiles. Encore un grand homme pour me montrer la bonne voie, qu'elle est grande ma fortune !

Ainsi, je profite pour remercier mes amis du Conservatoire : Oleksandr, Yumi, Sayo, Christatai, Sooyeon, Pauline, Vincent, Silvestre, Wendy et Miro, Cécile B. et Cécile D., Nozomi, Nami, Marilou, Sara, Rama, Léa, Hyon-Song, Pierre, Maura, Claire, Yu-Chen, Garance, Ariane, Ingrid, Mathilde, Orane, Camille, Adrien et Armelle, Diana, Liz, Sylvain, Alberto, Yebin, Luis, Beatriz, Jenny, Jinoh, Miso, Mei-Hung, Teresa, Elvis, Jan, Antonio, Pablo, Ludwig, Emilio, Anne, Emile, Charline, Estefanía, Paul, Jin Joo, Geumbit, Hyewon, Tom, Hye Jin, et enfin la formidable Kanako. Merci aussi le Directeur Bernard Soulès, Françoise de Maubus, Caroline Esposito, Victor, Franck, Alex, Alexandre, Daniel, Amanda, Jacqueline, Martine Huve, Sabine Bienaimé, Jackie et mon cher ami Jean-François Gonzales. Je remercie particulièrement Maestro Alexandre Ouzounoff pour nos conversations, cruciales à mes yeux, Jaime Rivera et Maestro Jezdimir Vujičić pour les parties d'échecs.

Parlant d'échecs, mes soirées à Paris ont bien souvent eu lieu au People's Drug-store. Dans un mélange de bonnes bières, cavaliers, dames, pions et tours, j'ai appris dans cet endroit que l'amitié n'a pas d'âge. J'ai pu rencontrer d'excellent joueurs d'échecs et amis, je pense à Colin, Adam, Ariane, Cagatay, Antonin, tous les Antoine, mon cher George, et tant d'autres. Le jour où le bar a dû fermer par ordre de la police a marqué une véritable fin de chapitre dans ma vie. Guillaume, Nadjib, François, merci d'avoir donné vie à ce lieu magique.

Mikhail, Micha, Miguel, cette page de remerciements serait cruellement incomplète sans te mentionner. Je t'ai vu évoluer et je suis fier, regarde comme tu es devenu un Artiste. Je raconte autour de moi ton histoire plus souvent que je ne le devrais, mais tu me pardonneras ; elle est tellement incroyable. Tu vis un roman, et j'ai eu la chance d'y participer. Tu vis un concerto, et ses mouvements commencent à peine. J'espère pouvoir un jour vraiment estimer la taille de ton talent.

Grandissime Simon, je te remercie d'être là et de m'avoir toujours encouragé. Tu as été un ami et collègue précieux; j'espère qu'on continuera à s'appeler pour ne rien se dire, et à partager notre philosophie dépourvue de tout sens qui génére des situations vraisemblablement surréalistes à chaque fois que l'on se voit.

Dans ce long voyage derrière la Cordillera des Andes, j'aurais été sans doute perdu sans la générosité de vous tous. Lors de mon arrivée en France, deux vrais Bretons: Antoine et Angeline. Merci de votre patience, amitié, et tous les moments passées ensemble sur un campus désert ! Merci Martin pour toutes les soirées de musique, Alexandra pour les folies, les conversations, les livres. Mes compagnons de voyage Flavio, Rafael, Juan Raphael, Lucho, Vinicius, Cassio, Igor, Celio, Ciro, Pedro, Mario, Flavia, merci d'avoir partagé votre authentique joie brésilienne, pour

moi c'est un trésor. Mes amis russes Anastasia, Nathalie, Andrey, Alexei, Gerard, Anna, Sergey m'ont accueilli à deux reprises dans un Moscou gelé à l'aide de café, vodka, concerts et escape games. A Milan, j'ai connu de vrais amici: Luigi, Marta, Michelle, Ted, Davide, Federico, Sara, Umberto, Matteo, Lorenzo, Pietro, Alessandro, Paola, Francesca, Alessandra, Consuelo, à qui je dit grazie de tout mon coeur. Je pense aussi à mes compadres chiliens, Constantino et la Turquie, Juan Ignacio et l'Italie, c'est des voyages que je n'oublierai jamais.

A toute l'équipe d'étude et recherches X Alora P/W je dois une reconnaissance spéciale. Juan Pablo, Marco, Rafael, merci pour les cafés et les colloques organisés autour des sujets d'actualité.

Je tiens à féliciter Marco & Roberta et Matias & Despina pour leur mariage à venir, ainsi qu'à Deaquél Cruz & Trinidad, Gabriel & Élodie, Carlos & Alba, Maria & Alistair de m'avoir invité à leur mariage. Caro Marco, merci pour Monza, pour Bormio, pour les rapidito, pour les sessions virtua tennis, pour tout. Tu es comme un grand frère pour moi. Deaquél, merci pour ton humour et ton amitié profonde, je suis tellement désolé d'avoir dû m'absenter de votre mariage.

Mes colocataires, je ne vous oublierai jamais. Laura avec sa danse, sa joie de vivre et son intensité, Anthony avec son Ricard, pétanque et tennis, Clothilde avec sa véritable bonté et ses plats délicieux. Merci pour les soirées, les repas, les moments partagés avec vous, et excusez-moi pour les clés oubliées et les plaques allumées.

Juan et Pablo sont les derniers arrivés sur ma route. J'ai eu cependant le temps d'admirer et d'apprendre beaucoup d'eux. De l'un, la joie de vivre, la simplicité, la persévérance, l'émerveillement devant les petites choses. De l'autre, la poésie, l'imagination, les bonnes bières, et quelques ouvertures folles d'échecs.

Chers Sophie, Mélanie, Perrine, Oliver, Gustav, Greta, Étienne, Joëlle, Josette, Marion, Nina, Benjamin, Raphaël, Joaquim, vous êtes devenus pour moi une deuxième famille, merci pour votre accueil et soutien.

No habría llegado aquí de no ser por Edmundo Núñez, quien me compartió su visión de las Matemáticas, y Rafael Benguria, quien me recomendó venir a Francia. Pienso sensiblemente en Álvaro Martinez, quién me dió las primeras nociones de Música en ese piano inolvidable.

Por sobre todas las personas que me han formado, pienso en particular en mi Maestro Francisco Gouët y su familia Lenka, Anne-Marie y Camilo. Francisco, su confianza y consejos son invaluables para mí. Cada conversación junto a Usted es para mí una lección de vida. Muchas gracias por esas tardes de Música y mucho más.

Juan Sebastián, Francisca, Manuel, José Ignacio, Magdalena y todos de mis amigos chilenos, gracias por apoyarme y creer en mí. Hay tanto para ponerse al día!

Mi familia es demasiado grande para nombrarlos a todos. Mis Padres, Inés, Clemente, Rosa, Lorenzo, Exequiel, Nicolás, Luisa, Daniel, Sofía, mi sobrina Emilia, mi abuela Inés, Ubriaco, sigamos unidos en nuestras diferencias, espero que mi camino y el suyo permitan reunirnos nuevamente.

Clémence, je pense enfin à toi. Merci de ces merveilleuses années, de ton support, tes conseils, tes livres. Je n'arrive toujours pas à croire que j'ai eu le courage de te parler. Tu sais me comprendre et m'encourager pendant les moments obscurs, et tu crois en moi plus que moi-même. Le hasard est beau car nous voilà réunis !

Abstract

Fully Homomorphic Encryption schemes allow public processing of encrypted data. Since the groundbreaking discovery of the first FHE scheme in 2009 by Craig Gentry, an impressive amount of research has been conducted to improve efficiency, achieve new levels of security, describe applications and detect connections to other areas of cryptography. In this Dissertation, we first give a detailed account on research these past years. Our contributions include a key-recovery attack against the ideal lattices FHE scheme and a new security conception inside hierarchic structures, avoiding betrayal between users to some extent, while maintaining the homomorphic encryption flexibility and also addressing an homomorphic re-encryption paradigm we detected. We also describe a working implementation of a recent proposal. This research was done in the Laboratoire de Mathématiques de Versailles, under supervision of Prof. Louis Goubin.

Résumé

Les schémas de Chiffrement Complètement Homomorphe permettent de manipuler des données chiffrées avec une grande flexibilité : ils rendent possible l'évaluation de fonctions à travers les couches de chiffrement. Depuis la découverte du premier schéma FHE en 2009 par Craig Gentry, maintes recherches ont été effectuées pour améliorer l'efficacité, atteindre des nouveaux niveaux de sécurité, trouver des applications et détecter des liens avec d'autres domaines de la cryptographie. Dans cette thèse, nous avons étudié en détail ce type de schémas. Nos contributions font état d'une nouvelle attaque de récuperation des clés contre le premier schéma FHE, et d'une nouvelle notion de sécurité en structures hiérarchiques, évitant une forme de trahison entre les usagers tout en gardant la flexibilité FHE. Enfin, on décrit des implémentations informatiques. Cette recherche a été effectuée au sein du Laboratoire de Mathématiques de Versailles avec le Prof. Louis Goubin.

Contents

List of Algorithms **viii**

1 Fully Homomorphic Encryption: A review **1**
1.1 Introduction . . . 3
1.2 Overview of this Dissertation . . . 6
1.3 Reported practical applications . . . 6
1.4 Connections: FHE and 8
1.4.1 ... Functional Encryption . . . 8
1.4.2 ... Cryptographic Obfuscation . . . 9
1.4.3 ... Secure Multi-Party Computation . . . 9
1.4.4 ... Identity-/Attribute-Based Encryption . . . 9
1.4.5 ... Proxy Re-encryption . . . 10
1.5 The five families . . . 10
1.5.1 Ideal Lattices . . . 10
1.5.2 Integers . . . 11
1.5.3 (Ring-)Learning with errors . . . 11
1.5.4 NTRU-based . . . 12
1.5.5 Approximate eigenvectors . . . 12
1.6 Definition of FHE . . . 13
1.6.1 Circuits versus functions . . . 13
1.6.2 Definitions . . . 13
1.6.3 Security definitions . . . 15
1.7 The bootstrapping procedure and other techniques . . . 16
1.7.1 Circular security . . . 17
1.8 Hard problems used in FHE . . . 17
1.9 Conclusion . . . 20

2 Gentry's ideal lattices scheme **23**
2.1 Introduction . . . 25
2.2 Preliminaries: Ideal lattices . . . 26
2.2.1 Lattices . . . 26

2.2.2 Notation . . . 28
2.2.3 Ideal lattices . . . 29
2.2.4 Polynomial principal ideal lattices . . . 29
2.2.5 Ring operations in $\mathbb{Z}^n$ . . . 31
2.3 The basic homomorphic scheme . . . 31
2.3.1 Does it work? Why does it work? . . . 33
2.3.2 Key generation and correctness of the algorithms . . . 34
2.3.3 Choice of the polynomial ring . . . 34
2.3.4 Lattice I: choice of the plaintext space . . . 35
2.3.5 Lattice J: ciphertext space . . . 36
2.3.6 Generating the bases . . . 37
2.3.7 Role of the representative sampling function . . . 38
2.3.8 Correctness of encryption and decryption . . . 39
2.3.9 Correctness of homomorphic operations . . . 40
2.3.10 Maximum depth of allowed circuits . . . 41
2.4 Security . . . 41
2.4.1 The Ideal Coset Problem . . . 41
2.5 Natural density of some classes of lattices . . . 42
2.5.1 Natural density of diagonal elements of HNF's . . . 43
2.5.2 Natural density of coprime lattices . . . 44
2.5.3 Natural density of odd ideal lattices . . . 45
2.5.4 Remark on natural density of circulant matrices with odd determinant . . . 46
2.6 Proof-of-concept implementation . . . 47
2.6.1 Basic functions . . . 47
2.6.2 Scheme algorithms . . . 48
2.6.3 Performances . . . 50
2.7 Conclusion . . . 51

3 A Key-Recovery Attack Against Gentry's Ideal Lattice Scheme via ad-HPP 53
3.1 Introduction . . . 55
3.2 Preliminaries . . . 57
3.2.1 Notation . . . 57
3.2.2 Lattices . . . 57
3.2.3 Polynomial ideal lattices . . . 58
3.2.4 Key-Recovery Attacks against SHE schemes . . . 59
3.3 Gentry's ideal lattice basic scheme . . . 59
3.4 The attack . . . 61
3.4.1 Reduction to ad-HPP . . . 61

3.4.2 Feeding Nguyen-Regev's algorithm and Random Scattering . 62
3.4.3 Falling towards vertices 63
3.5 Length of binary searches in the neighbor algorithm 69
3.6 Conclusion 70

4 A betrayal problem - The Excalibur Property 73
4.1 Introduction 74
4.2 Preliminaries 78
4.2.1 Notation 78
4.2.2 The quotient ring R_q 78
4.2.3 Bounded discrete Gaussian samplings on $\mathbb{Z}[x]/(x^n + 1)$ 79
4.3 Modified NTRU encryption 79
4.3.1 The multikey property 80
4.4 Hardness assumptions 81
4.4.1 Small Polynomial Ratio Problem, from [LATV12] 81
4.4.2 Small factorizations in the quotient ring 82
4.5 Two-party multiplication protocols in R_q 84
4.5.1 Secret inputs setting 84
4.5.2 Shared inputs setting 85
4.6 Excalibur key generation 86
4.7 Security 87
4.7.1 Honest-but-curious model 89
4.7.2 Security against one malicious party 90
4.8 Extensions 93
4.8.1 Chains of keys 93
4.8.2 Plugging in LATV-FHE 94
4.9 Conclusion 94

5 The promising BGV scheme: an implementation 96
5.1 Introduction 98
5.2 The scheme 98
5.2.1 Parameters 98
5.2.2 Key Generation – sampling from $\mathbb{Z}_p[x]/(x^N + 1)$ 99
5.2.3 Encryption and decryption 100
5.2.4 Relinearization and Modulus switching 100
5.2.5 BGV Homomorphic operations 102
5.3 Implementation 104
5.3.1 Math Layers 104
5.3.2 Cryptography layers 106
5.4 Performance 109
5.5 Conclusion 109

A Seeing the whole picture: An article reading guide **124**

B Homomorphic operations on sets **128**

C Coefficient correlation in the Small Factors Problem **132**
C.1 $\phi(x) = x^n - 1$ (NTRU ring) 132
C.2 $\phi(x) = x^n + 1$ (rLWE ring) 134

List of Figures

1.1 An easy maze 13
1.2 The re-encrypt procedure 17

2.1 A lattice of $\mathbb{R}^2$ and two bases. 26
2.2 A basis modular reduction in $\mathbb{R}^2$ 28
2.3 Fundamental parallelepipeds in Gentry's scheme 32

4.1 M. and G. perform Excalibur $\mathsf{Keygen}_{\mathsf{sk}}$ 89

5.1 BGV implementation: `C++` Class dependencies 104
5.2 First five Irwin-Hall probability density functions. 106

List of Algorithms

1 The Re-Encrypt algorithm 17
2 Gentry's homomorphic ideal lattice scheme 33
3 Key Generation for Gentry's IL homomorphic scheme 38
4 Unimodular Matrix Generator 48
5 Hermite Normal Form 49
6 Gentry's homomorphic ideal lattice scheme 60
7 Finding a point near ∂P 64
8 Orthogonal bounces towards a vertex 65
9 Fall onto a vertex 65
10 Finding a neighbor vertex 66
11 Excalibur - TMP 84
12 Excalibur - $\mathsf{SharedTMP}$ 86
13 Excalibur - $\mathsf{Keygen}_{\mathsf{pk}}$ 87
14 Excalibur - $\mathsf{Keygen}_{\mathsf{sk}}$ 87
15 Excalibur - Validation function (performed by Alice) 88
16 Excalibur - Validation protocol (performed by Alice and Bob) 88
17 BGV - Key Generation 99
18 BGV - Encryption and Decryption 100
19 BGV - Key-Switching Procedure 101
20 BGV - Homomorphic Operations 102
21 BGV - Scaling ciphertexts 103
22 Recursive FFT 108
23 Homomorphic Intersection 130
24 Homomorphic Union 130

List of Tables

2.1 Timings of Gentry's key-generation procedure in toy settings 50
2.2 Reported secret key Euclidean size in Gentry's *tweaked* scheme. . . . 51

3.1 Direction searching in the dichotomy of algorithm 10. 70

5.7 BGV algorithms performance in user seconds. 110

A.1 The five families 124
A.2 FHE publications map 125

Chapter 1

Fully Homomorphic Encryption: A review

Contents

1.1 Introduction 5
1.2 Overview of this Dissertation 8
1.3 Reported practical applications 8
1.4 Relations: FHE and 10
1.4.1 ... functional encryption 10
1.4.2 ... cryptographic obfuscation 11
1.4.3 ... secure multi-party computation 11
1.4.4 ... identity-/attribute-based encryption 11
1.4.5 ... proxy re-encryption 12
1.5 The five families 12
1.5.1 Ideal Lattices 12
1.5.2 Integers 13
1.5.3 (Ring-)Learning with errors 13
1.5.4 NTRU-based 14
1.5.5 Approximate eigenvectors 14
1.6 Definition of FHE 15
1.6.1 Circuits versus functions 15
1.6.2 Definitions 15
1.6.3 Security definitions 17
1.7 The bootstrapping procedure and other techniques . . . 18
1.7.1 Circular security 19
1.8 Hard problems used in FHE 19
1.9 Conclusion 22

1.1 Introduction

The word *cryptography* comes from the Greek *kryptos* and *graphia* – it means "hidden writing". Cryptography is the science of protecting sensitive information, as it allows to transform readable text into gibberish writings called *ciphertexts*. These random-looking messages can in turn be reversed into readable text, but only by those whose the message was intended for. The first transformation is called *encryption*, and the reverse transformation is called *decryption* (*un-hiding*). Disguising secrets is inherent to human nature: the first reported usage of secret writing dates back thousands of years. Since the famous Enigma machine used in World War II and its rotor solver[1], increasingly refined ways of hiding information and attacking ciphertexts are available, whose complexities escape by far the human scope. Today, cryptography is a machine's game, played billions of times a day in order to protect our data and communications. Recently, a new kind of encryption made its way into our cryptography toolkit, and exciting research has been conducted to defend and improve this new instrument: Fully Homomorphic Encryption (henceforth FHE).

A beautiful peculiarity of cryptography as a science is that the basic, invariant and common element in all its branches is information itself. It is fair to say that information is to cryptography as energy is to physics. A bit more far-fetched would be to say that Einstein's relativity is to physics as FHE is to cryptography, but, on the very least, it has been referred to as *the holy grail of Cryptography*.

Let us have a glance at this with the following question. Alice sends Bob a ciphertext c, duly encrypting a message m. If $f(m)$ is some function of the message, can Bob deduce an encryption of $f(m)$ with the sole knowledge of c? Much counter-intuitively, the answer is yes for a special family of encryption schemes.

Oblivious cookie baking. Alice encrypts butter, sugar, eggs, flour and chocolate, she gives these cipher-ingredients to Bob and asks him to bake a chocolate cookie. Notice that Bob is not able to decrypt since he does not possess Alice's secret, thus he faces the task without awareness of the ingredients – and eventually does not know what a chocolate cookie is. Nonetheless, he does the baking and in the end, Alice decrypts the chocolate cookie and eats it with warm milk. Bob does not understand what has just happened, and there is no cookie for him since the result of his baking is only decryptable by Alice.

In 1978, two years after the revolutionary works of Diffie and Hellman, cryptographers Ronald Rivest, Len Adleman and Michael Dertouzos raised in [RAD78] the question of performing arbitrary operations on ciphertexts, without access to the underlying data:

[1] For excellent surveys on history of cryptography, see [Kah96, Sin99].

> *"Encryption is a well-known technique for preserving the privacy of sensitive information. One of the basic, apparently inherent, limitations of this technique is that an information system working with encrypted data can at most store or retrieve the data for the user; any more complicated operations seem to require that the data be decrypted before being operated on. This limitation follows from the choice of encryption functions used, however, and although there are some truly inherent limitations on what can be accomplished, we shall see that it appears likely that there exist encryption functions which permit encrypted data to be operated on without preliminary decryption of the operands, for many sets of interesting operations. These special encryption functions we call "privacy homomorphisms"; they form an interesting subset of arbitrary encryption schemes (called "privacy transformations")".*

These encryption functions were at first called *Privacy Homomorphisms*, in the sense that given an encryption $E(D)$ of some data D and a functionality f, it is possible to obtain an encryption of $f(D)$ without knowing D nor any decryption secret, hence providing an homomorphism between message and ciphertext spaces. In fact, in some encryption schemes, plaintext and ciphertext messages lie in algebraic structures, like a ring or a group, and partial algebraic homomorphisms between the two can sometimes be constructed. In their seminal paper, they give four examples of simple (insecure) privacy homomorphisms, based on discrete-logarithms, RSA-like encryption, Chinese Remainder Theorem and Radix-n respectively (these last two are *non-deterministic*). Even before the existence of a secure privacy homomorphism was proven – or believed, for that matter – multiple applications arose specially in the context of outsourced processing. As a matter of fact, some years later there were several public encryption schemes already allowing this for some functionalities. For (a by now folklore) instance, basic RSA encryption happens to be multiplicative. If $c_1 = m_1^e \mod n$ and $c_2 = m_2^e \mod n$, then $c_3 = c_1 \times c_2 \mod n$ is an encryption of $m_1 \times m_2$. In classic cryptography, this is to be avoided at all costs: an attacker can construct a valid ciphertext by simply multiplying two or more encryptions – basic RSA is said to be *malleable*. In light of this, any privacy homomorphism is malleable.

Notice, however, that it is not clear how to construct a valid RSA encryption of $m_1 + m_2$ from c_1, c_2. Hence basic RSA is not known to be an almighty privacy homomorphism: in order to evaluate an arbitrary polynomial-time computation f defined as a circuit C with $+$ and $\times$ gates, a full privacy homomorphism must at least describe two operations corresponding to these gates (or any other complete set of operations). Other partial privacy homomorphisms followed: Paillier, Okamoto–Uchiyama, Naccache–Stern, Cohen–Fischer, all of them achieving *homomorphic* addition or multiplication of messages, but not both. At this point, community was

already on the search for almighty privacy homomorphisms, who were baptized as *fully homomorphic encryption* schemes.

The first example of an additively and multiplicatively homomorphic scheme is due to Boneh, Goh and Nissim. Their scheme, based on bilinear couplings in elliptic curves, was capable of evaluating any number of $+$ gates, but only one $\times$ gate. This was a partial victory, since their scheme allows to evaluate all affine, bilinear and quadratic functionalities. Another effort was a scheme proposed by Fellows and Koblitz, based on the hardness of a membership problem in the ring $\mathbb{F}_q[X_1, \cdots, X_n]$, however it suffered some attacks and is not considered secure.

All of these efforts and the apparent slippery of true FHE were coupled with some degree of skepticism. In spite of this, the quest continued until 2009, when Craig Gentry proposed the first fully homomorphic encryption scheme in his Ph.D. dissertation at Stanford University. In theory, his scheme is capable of homomorphically evaluating any polynomial-time functionality f over encrypted data. His merit does not only consist in having the insight to find the mathematical object that allowed the construction, but also proposing generic transformations between a family of weak homomorphic schemes – called *somewhat homomorphic encryption (SHE) schemes* – and a fully homomorphic encryption scheme. The schemes in this family must own a self-reference property called "bootstrapping" that allows to "clean" the output of a large circuit.

In [Gen09a, Gen09b, Gen09c, Gen10a], Gentry describes this revolutionary idea and gives an explicit construction of a family of SHE schemes relying on *ideal lattices*. This family is parameterized by the maximum depth of allowed circuits, and can be transformed into an FHE scheme. Consequently, Gentry had constructed the first blindfolded computing machine. Since his breakthrough, other schemes have been proposed, all of them following this blueprint.

In 2009, the theoretical playground in cryptographic security was fairly ready to the arrival of FHE, as malleability of ciphertexts was widely discussed. Homomorphic encryption allowed to relativise previous notions, in that it gives honest use of ciphertext modification and introduces new levels of security that are, and rightly so, considered to be grand contributions.

Two flat tires of this thrilling discovery are the ones of efficiency and underlying assumptions. Indeed, Gentry's ideal lattices scheme is complex and suffers from enormous ciphertext expansion and public-key size, and its celebrated *bootstrapping* procedure had to wait for two years to be actually implemented. On the other hand its security relied in, though reasonable and arguable, new assumptions. The fact that this scheme was not even remotely comparable to other encryption schemes provoked a second wave of skepticism, yet, thanks to enthusiastic scientific research, multiple optimizations and new candidates have been proposed to meet applications. There are now five independent families of homomorphic schemes, each member ad-

dressing a different challenge from the community (efficiency, optimal depth of decryption, optimization for static functions, identity-based, attribute-based, support for multiple users, fast bootstrapping). We describe these families in this Introduction in 1.5.

Today, we can encrypt, decrypt and add packages of binary messages in a matter of milliseconds, and multiply, bootstrap and relinearize in some seconds. Manifestly, this is sufficient for some growing set of exciting applications, see 1.3.

1.2 Overview of this Dissertation

In the rest of this Chapter, we describe some practical applications, report some connections with other cryptography fields, explain how proposals are today ordered into five branches, give a formal definition of FHE and list underlying computationally-hard problems.

In Chapter 2, we revisit Gentry's first FHE proposal based on ideal lattices. We also propose a proof-of-concept implementation and a digression about natural densities of some special sets of lattices.

Our main contributions are given in Chapters 3 and 4. In Chapter 3, we describe a key-recovery attack against Gentry's first scheme that, in spite of the enormous parameters selected for security and feasibility, recovers a secret-key of the scheme in sub-quadratic time with a reasonable number of decryption queries. This is the subject of a paper submitted to an international conference on Cryptography. In Chapter 4, we describe a technique of *one-way gluing* keys from different users of an FHE scheme, addressing for the first time a natural betrayal problem that arises in hierarchic scenarios. With this, we deal with a homomorphic proxy re-encryption paradigm we detected. To that end, we define the Excalibur property and show that the celebrated NTRU-based FHE of Lopez-Alt et al. displays it. We reproduce an article included in the proceedings of the Indocrypt 2016 conference [GVP16].

Finally, in Chapter 5, we report a `C++` implementation of the FHE scheme of Brakerski, Gentry and Vaikuntanathan, which is believed to be one of the most promising candidates in terms of efficiency.

In addition, appendix A "Seeing the whole picture: An article reading guide" may be of help when approaching the disconcerting and multidimensional literature of fully homomorphic encryption.

1.3 Reported practical applications

In principle, all outsourced statistical analysis of sensible data can be performed homomorphically. This is now being done in some domains. We collect here some of the recent – public – applications.

Processing Genomic Data

Immense progress has been done in study of genomes, and large databases of human genomic data are essential to researchers. However, as pointed out in the report [ADCHT13], given a short DNA sequence of an individual, it is possible to search by pattern matching in available genomic databases, and eventually retrieve his whole sequence. With this, the individual privacy is highly invaded: the attacker may compute illness risks, physical characteristics and eventually identity (for instance having access to hospital or police records). For that reason, there is legal and ethic need to protect database volunteers. Researchers have recently propose to encrypt the human DNA sequence with a fully homomorphic encryption scheme, in order to conduct meaningful statistic experiences over the encrypted data. In [LLAN15] and [KL15], such homomorphic evaluations are performed, aiming to process encrypted genomic data. Authors present computational experiments and report correct and efficient evaluation of statistic tests such as Pearson's Goodness-of-Fit test, D' and r^2 measures for linkage disequilibrium, Estimation Maximization for haplotyping, and Cochran-Armitage tests. In the follow-up article [BLN13], authors explain how to perform predictive analysis of encrypted genomic data. In addition, they provide a working implementation of an on-line privacy-respecting cardiovascular disease (such as heart failures) probability predictor.

Electronic voting

Recently in [CGGI16b], authors presented a new post-quantum e-voting protocol using lattice-based fully homomorphic encryption. Their scheme proves to be very transparent, since no additional verifications need to be taken care of when counting votes or revealing the final result of an election. Indeed, former protocols already proposed to count votes homomorphically, but relied on non-interactive zero-knowledge proofs carried out by several trustees for ballot validations, which looks like a tail-biting drawback. In [CGGI16b], only two trustees are required to give independent proofs of correct decryptions.

Targeted Advertising

Gender, age, location and known interests of an individual are currently been used to provide personalized advertising. Yet if a mobile phone sends the owner's location to an advertising provider who answers tailored ads, this provider can keep record of his client's activities. Ensuring privacy is thus paramount, and it can be done performing the advertising algorithms homomorphically in a certain scenario, see [LNV11]. Also in [JPH13], authors use secure multi-party computation protocols and somewhat homomorphic encryption to secure privacy when the user is asking recommendations for a given product to friendly users, preserving everyone's

anonymity.

Outsourced forensic image recognition

Detection of illegal images in a confiscated hard-drive is done by comparing the contents with a confidential database of hashes of known illegal images. From [BPHJ14] we quote:

> *"The Dutch police, for example, owns a database consisting of hash values of so-called PIPs (Picture that Interests the Police), such as images showing glorification or overexposure of violence, indignity or pornographic content, like zoophilia and pedophilia. When the police confiscates equipment with data storage, the hash of each picture found in the storage is computed and looked up in the PIP database. If there are many matches, the police knows that the confiscated equipment contains PIPs and the investigation is continued manually to crosscheck."*

To stop the distribution of such images inside a company's in-line traffic, for instance, it would be helpful to have remote access to this confidential databases, however law enforcement agencies refuse outsourcing at all costs to prevent ill use thereof. In this article, authors propose SOFIR, an on-line algorithm which encrypts databases and allows recognition of illegal images in a company's traffic, sending reports in real time while respecting privacy of other files.

Biometric authentication

Biometric authentication consists in identification of an individual using binary encodings of some biological attribute, such as fingerprints or retina. A scan allowing some small error can be performed in order to authenticate, followed by pattern matching against a database. In [YSK+13] authors provide a privacy-preserving secure pattern-matching, optimized for biometric clearance.

1.4 Connections: FHE and ...

As expected, fully homomorphic encryption did not land in a desert. It has proven to be closely connected to numerous problems and research subjects.

1.4.1 ... Functional Encryption

A functional encryption scheme is a public-key encryption scheme allowing generation of special secret-keys such that, when used in decryption, the output is a specific function of the plaintext. Namely, for a valid function f there is a secret key sk_f such that for a ciphertext c encrypting m, the decryption $\mathsf{Dec}(\mathsf{sk}, c_f)$

outputs $f(m)$ and nothing more. This notion controls malleability, in as much as secrets must be possessed in order to learn more functions of the ciphertext. In opposition, FHE gives total freedom of computation, but the output is encrypted. Authors in [ABF+13] provide a detailed study of this relationship and show that under certain assumptions, a functional encryption scheme supporting operations on two ciphertexts actually implies the construction of an FHE scheme.

1.4.2 ... Cryptographic Obfuscation

Obfuscating a program is hiding the information embedded in the code, without hurting functionality. The obfuscated program behaves like a *black-box* implementation of the original one, in the sense that no information other than the input and output can be obtained from the program. With FHE it is possible to obfuscate some functionalities, but the result is by definition, encrypted, thus conditioned decryption may be necessary to perform the last step of the program, i.e. producing meaningful output. A natural idea is to hide this conditioned decryption inside the program. Heavily using FHE and cryptographic multilinear maps as building bricks, authors in [GGH+13] proposed the first candidate for cryptographic obfuscation.

1.4.3 ... Secure Multi-Party Computation

In a secure multi-party computation (MPC) protocol, two or more mutually distrusting parties wish to compute a function of their joint data. For instance, Alice holds some value x and Bob holds y, they want to perform some protocol such that they both learn $f(x, y)$ and nothing more that can be deduced from their input and output. Using a variant of FHE called *Multikey*-FHE, authors in [LATV12] showed that the general MPC problem can be reduced to a specific MPC instance – the one of a joint decryption of a ciphertext in their scheme.

1.4.4 ... Identity-/Attribute-Based Encryption

IBE is a scenario where the public-key of a user can be deduced from a tag such as a string (e.g. "Name.Surname", address, e–mail, phone number, subscription number). This was proposed in [Sha85], and it relies on an authority that distributes secret-keys to users corresponding to their public-key after proper clearance. Two constructions were proposed in 2001 using quadratic residues and Weil pairings of elliptic curves (see [Coc01] and [BF01] respectively). In an attribute-based encryption scheme, decryption of a ciphertext is possible whenever the decryptor meets some predefined conditions. In other words, secret-keys depend on user's attributes, e.g. company, name, kind of subscription. The first ABE scheme was proposed in [SW05] (see [LCH13] for a nice survey). Hybrid schemes have been proposed in [GSW13]:

Attribute-Based HE, Identity-Based FHE (IBFHE) and Multi-identity IBFHE. Interestingly, these constructions provide compilers that enhance any LWE-based FHE scheme with the desired property.

1.4.5 ... Proxy Re-encryption

A proxy re-encryption procedure allows the public transformation of ciphertext c decryptable by Alice into a ciphertext c' decryptable by Bob. It is said to be multi-hop if this new ciphertext can be in turn re-encrypted for another party, and bidirectional if there are reverse transformations. This is useful in mail redirecting applications. It has been claimed (for instance in the seminal work [Gen09b]) that following the bootstrapping argument, FHE is the first candidate for multi-hop bidirectional proxy re-encryption. However, we point out in [GVP16] that this relies on a non-standard assumption. We propose an "automatic re-encryption" key-generation procedure that addresses this paradigm and, moreover, solves an interesting betrayal issue in hierarchic structures. This is detailed on Chapter 4.

1.5 The five families

To the date, there are five recognizable families of homomorphic schemes, addressing different security challenges, exhibiting fancy properties and competing for efficiency.

1.5.1 Ideal Lattices

The first proposal of a fully homomorphic scheme uses the notion of ideal lattices. These are regular lattices laying in the image of a ring homomorphism between $R = \mathbb{Z}[x]/(f(x))$ and $\mathbb{Z}^n$ for some monic polynomial f of degree n, when the homomorphism is fed with a principal ideal of R. By structure transport, this gives a nice representation of $\mathbb{Z}^n$, supporting addition, multiplication and *basis reduction*. With these three operations and careful sampling of two ideal lattices I, J, Gentry constructed the first fully homomorphic scheme in [Gen09b].

Simplified variants and new proposals follow with works of Smart and Vercauteuren with improvements by Stehlé, Steinfeld, Loftus and May. See [SS10,LMSV12].

As an homage to his dissertation, we devote Chapter 2 to the study of this scheme in sufficient level of detail. We consider this study to be rewarding and, contradicting the pro-FHE enthusiasm exposed in this Introduction, we propose a new key-recovery attack in Chapter 3 that reveals completely the secret key. We consider the relatively new and rarely used in cryptography problem of guessing a $\mathbb{Z}^n$-parallelepiped given a collection of points. We mention that authors in [LMSV12] presented the only homomorphic scheme known to be CCA-1 secure so far.

1.5.2 Integers

Only a year after the first proposal, van Dijk, Gentry, Halevi and Vaikuntanathan proposed a fully homomorphic encryption scheme that used integers as plaintexts and ciphertexts, in [vDGHV10]. This allowed a much better understanding of the efficiency gap between homomorphic and classic schemes, and conceptually suggests that FHE is not *that* artificial, as algorithms are simple to understand. This scheme has been revisited recently to support batching in [CCK$^+$13], allowing to pack ciphertexts and perform parallel homomorphic operations. However, to the date this family is not considered to be the most promising.

1.5.3 (Ring-)Learning with errors

Brakerski and Vaikuntanathan proposed a scheme based on the learning with errors problem, which has been deeply studied since its introduction by Regev in 2005 (see [Reg05, Reg10]). This problem is conjectured to be hard to solve, and faces a polynomially large linear system in which every equation is almost correct – approximated up to some small error. Let us reproduce Regev's folklore example here; an instance of the LWE problem is to recover an integer vector $s = (s_1, s_2, s_3, s_4) \in \mathbb{Z}_{17}^4$ satisfying

$$\begin{cases} 14s_1 + 15s_2 + 5s_3 + 2s_4 & \approx & 8 & \mod 17, \\ 13s_1 + 14s_2 + 14s_3 + 6s_4 & \approx & 16 & \mod 17, \\ 6s_1 + 10s_2 + 13s_3 + 1s_4 & \approx & 3 & \mod 17, \\ 10s_1 + 4s_2 + 12s_3 + 16s_4 & \approx & 12 & \mod 17, \\ 9s_1 + 5s_2 + 9s_3 + 6s_4 & \approx & 9 & \mod 17, \\ 3s_1 + 6s_2 + 4s_3 + 5s_4 & \approx & 16 & \mod 17, \end{cases}$$

where "$\approx$" means that the equation is correct up to an error of ± 1. As the abstract of [BGV12] reads,

> *"We present a radically new approach to fully homomorphic encryption (FHE) that dramatically improves performance and bases security on weaker assumptions. A central conceptual contribution in our work is a new way of constructing leveled fully homomorphic encryption schemes (capable of evaluating arbitrary polynomial-size circuits), without Gentry's bootstrapping procedure"*

They give constructions based in LWE and its ring variant, and introduced novel noise-management techniques called relinearization and modulus switching, which replace the bootstrapping procedure. In this scenario, homomorphic evaluation and decryption climb a ladder of decreasing moduli and scale the ciphertexts properly to ensure correctness. Also, in [Bra12] Brakerski proposed a scheme in which the same

modulus is used throughout evaluation; these schemes are called *scale invariant*. Some remarks and tweaks in this family bring a great deal of efficiency improvements, allowing for instance to bootstrap a ciphertext in less than 1s [DM15] and subsequently in less than 0.1s [CGGI16a].

1.5.4 NTRU-based

The "N-th degree TRUncated" scheme is an efficient lattice based encryption scheme proposed in [HPS98]. For some time, due to absence of formal proofs, its security level was not clear. In 2011, Stehlé and Steinfeld [SS11b] provided modifications that allowed to complete the proofs (more precisely, the use of cyclotomic rings and a power of 2 degree). In the groundbreaking article [LATV12], authors constructed an homomorphic scheme based on this encryption flavor. Their contribution is large: they achieve homomorphic support for multiple users proposing *Multikey*-FHE, conceive the notion of *on-the-fly* MPC, and propose a leading candidate for practical FHE. Based on this work, Goubin and V. propose in [GVP16] a method to blend NTRU keys together, addressing a betrayal problem in hierarchic encryption scenarios. For more on this see Chapter 4.

1.5.5 Approximate eigenvectors

In all previous BGV-like encryption schemes, ciphertext multiplication are seemingly complex, artificial procedures. Motivated by this, Gentry, Sahai and Waters in [GSW13] proposed a new generation of fully homomorphic schemes. Ciphertext and key representation is new, while security is still based on variants of the LWE problem. In this scheme, a secret-key is a vector v in $\mathbb{Z}_q^n$ with at least one large coefficient. A ciphertext C corresponding to this key is a $n \times n$ binary matrix over $\mathbb{Z}_q$ for a large prime q, such that v is close to an eigenvector of C, the corresponding eigenvalue being the underlying plaintext $m \in \{0, 1\}$. In other words, m, C and v verify

$$C \cdot v = m \cdot v + e \mod q$$

for small $e \in \mathbb{Z}_q^n$. Homomorphic evaluation is simply given by matrix arithmetic, producing asymptotic improvements in efficiency: their construction is asymptotically faster than any previous FHE scheme (alas with a large constant keeping it out of the practical scope). To present their analysis, authors consider $\{\texttt{NAND}\}$ as the complete set of operations to evaluate circuits. In addition, matrix multiplication is not commutative, and therefore some ways to multiply ciphertexts are better than others, regarding noise growth.

1.6 Definition of FHE

We give in this section a theoretical definition of homomorphic encryption. Let us first remark why we use circuits instead of general functions.

1.6.1 Circuits versus functions

Homomorphically solving a maze runs in worst-case complexity. Suppose that P is a 2-D labyrinth solver algorithm. It receives the plot of a maze M and a point x inside the maze, and the algorithm outputs a path from x to the exterior of the maze, or the symbol $\perp$ if there is no exit. Typically, such algorithm must test conditions such as "if the point $y \in M$ has already been visited", or "if this is a dead end". If for any reason the maze walls, its inside points and the position of the player are probabilistically encrypted, such comparisons become comparison or identity circuits with an encrypted output. The whole algorithm can be carried out homomorphically and it actually produces the correct (encrypted) output, but because of this lack of decision it runs in its worst-case complexity, outputting the path as a complicated polynomial expression depending on all points of the maze. The same applies when running set or graph algorithms homomorphically. This is why we only consider being able to evaluate circuits of fixed depth, instead of general functions or procedures. To avoid this paradigm for the moment, all research in homomorphic encryption embraces this consideration.

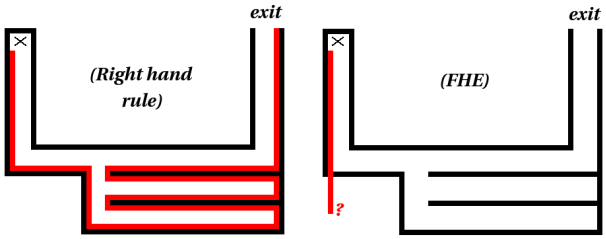

Figure 1.1: An easy maze

1.6.2 Definitions

Let us now give some definitions.

Definition 1.6.1 (HE)**.** *An homomorphic encryption scheme is a public-key encryption scheme $\mathcal{E}$ with four PPT algorithms with polynomial complexity in the security parameter λ:*

– Keygen*:* $\lambda \mapsto (\mathsf{sk}, \mathsf{pk})$*, generates a key pair.*

- Enc*:* $(\mathsf{pk}, \pi) \mapsto \psi$*, encrypts a message* π *under public-key* pk*.*
- Dec*:* $(\mathsf{sk}, \psi) \mapsto \pi$*, decrypts a ciphertext* ψ *under secret-key* sk*.*
- Eval*:* $(\mathrm{pk}, C, \psi_1, \ldots, \psi_t) \mapsto \psi$*, where* C *is a circuit that belongs to a set of allowed circuits* $\mathcal{C}$*, evaluates the circuit* C *gate by gate on inputs* $\psi_1, \ldots, \psi_t$*.*

The evaluation algorithm should describe how to perform addition and multiplication gates (or any other complete set of operations). If the resulting ciphertext decrypts to the desired message, we say that the scheme is correct. Let $\mathcal{C}$ be a collection of circuits:

Definition 1.6.2 (Correct HE)**.** *An HE scheme is correct for all circuits in* $\mathcal{C}$ *if for all* $C \in \mathcal{C}, (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Keygen}(1^\lambda)$ *and for all plaintexts* $\pi_1, \ldots, \pi_t$*, if* $\psi_i \leftarrow \mathtt{Enc}(\mathsf{pk}, \pi_i)$ *for* $i = 1, \ldots, t$*, then*

$$\psi \leftarrow \mathsf{Eval}(\mathsf{pk}, C, \psi_1, \ldots, \psi_t) \quad \textit{implies} \quad \mathsf{Dec}(\mathsf{sk}, \psi) = C(\pi_1, \ldots, \pi_t).$$

The decryption algorithm can be itself written as a circuit D; if the scheme correctly evaluates D it will be later possible to introduce the bootstrapping procedure. Hence the following definition:

Definition 1.6.3 (Compact HE)**.** *An homomorphic encryption scheme is compact if there is a polynomial* f *such that for any value of* λ*, the algorithm* $\mathsf{Dec}_{\mathcal{E}}$ *may be computed by a circuit of depth* $\leq f(\lambda)$*.*

I.e. an homomorphic encryption scheme is compact if its decryption algorithm is of reasonable size.

Definition 1.6.4 (Compact evaluation)**.** *An homomorphic scheme compactly evaluates a set of circuits* $\mathcal{C}$ *if it is compact and correct for all circuits in* $\mathcal{C}$*.*

Definition 1.6.5 (FHE)**.** *A Fully Homomorphic Encryption Scheme is an homomorphic scheme that compactly evaluates all circuits.*

Definition 1.6.6 (Levelled FHE)**.** *A family of HE schemes* $\{\mathcal{E}^{(d)}, d \in \mathbb{N}\}$ *is levelled fully homomorphic if*

(i) they all have the same decryption circuit D

(ii) for each $d \in \mathbb{N}$*,* $\mathcal{E}^{(d)}$ *compactly evaluates every circuit of depth* $\leq d$ *and all algorithms have polynomial complexity.*

1.6.3 Security definitions

Consider the following game between parties A and B.

Game 1.6.1. Let $\mathcal{E}$ be a public key encryption scheme of plaintext space $\mathcal{P}$ and algorithms $\mathcal{E}.\mathsf{Keygen}, \mathcal{E}.\mathsf{Enc}, \mathcal{E}.\mathsf{Dec}$. Let A, B be two parties.

1. A generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{E}.\mathsf{Keygen}(\lambda)$ and publishes pk.
2. For i from 1 to t, B asks A to decrypt a ciphertext ψ_i, and A answers $\mathcal{E}.\mathsf{Dec}(\mathsf{sk}, \psi_i)$. These exchanges cannot be repeated after the challenge ciphertext.
3. (Challenge) B generates two plaintexts $\pi_0, \pi_1 \in \mathcal{P}$ and sends them to A, who chooses $b \in_R \{0, 1\}$, and sets $\psi^* \leftarrow \mathcal{E}.\mathsf{Enc}(\mathsf{pk}, \pi_b)$. She sends ψ^* to B.
4. B wins if he guesses the value of b.

Definition 1.6.7. *Let $b' \in \{0, 1\}$ be B's final guess in the game. The advantage of B is defined by*

$$\mathrm{Adv}(B, \mathcal{E}, \lambda) = |\Pr(b = b') - 1/2|.$$

Definition 1.6.8 (Semantic security)**.** *We say that $\mathcal{E}$ is semantically secure against CPA,CCA1,CCA2 if it is not possible for a CPA,CCA1,CCA2 (respectively) adversary to win the game with a non negligible advantage and polynomial time in λ.*

For a homomorphic scheme, consider the following game.

Game 1.6.2. Let $\mathcal{H}$ be a homomorphic encryption scheme of plaintext space $\mathcal{P}$ and algorithms $\mathcal{H}.\mathsf{Keygen}, \mathcal{H}.\mathsf{Enc}, \mathcal{H}.\mathsf{Dec}, \mathcal{H}.\mathsf{Eval}$. Let A, B be two parties.

1. A generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{H}$ and publishes pk.
2. For i from 1 to t, B asks A to decrypt a ciphertext ψ_i, and A answers $\mathcal{H}.\mathsf{Dec}(\mathsf{sk}, \psi_i)$. This step can be repeated after the challenge, with new ciphertexts not matching the challenge ciphertext.
3. (Challenge) B generate two sets of plaintexts $(\pi_{0i})_{i=1}^k, (\pi_{1i})_{i=1}^k \in \mathcal{P}^k$ and an allowed circuit C and sens them to A, who chooses $b \in_R \{0, 1\}$, computes

   $$\psi^* \xleftarrow{R} \mathcal{H}.\mathsf{Eval}(\mathsf{pk}, C, (\psi_{bi})_{i=1}^k)$$

   where $\psi_{bi} = \mathsf{Enc}(\mathsf{pk}, \pi_{bi})$ and sends ψ^* to B.
4. B makes a guess $b' \in \{0, 1\}$ and wins if $b' = b$.

It is not difficult to show with an hybrid argument that an algorithm breaking the (homomorphic) semantic security with advantage ε can be used to break the (classical) semantic security with advantage ε/k. The hybrid argument "travels" through the gates of the chosen circuit and estimates the advantage. Therefore, to prove the semantic security of an homomorphic scheme, it suffices to prove the semantic security in the classical sense.

1.7 The bootstrapping procedure and other techniques

In all known homomorphic schemes, the encryption procedure adds random noise to plaintexts. If the norm of this noise is beyond a certain threshold, the resulting ciphertext does not decrypt to the expected underlying plaintext. In addition, every homomorphic evaluation results in a growth of the corresponding noise. This limits the maximum depth achieved for correct evaluation of circuits at parameter setting time, and hence something must be done to achieve true FHE. Gentry achieves this with his most groundbreaking idea: If an homomorphic scheme $\mathcal{E}$ can homomorphically evaluate its own decryption circuit $D_{\mathcal{E}}$, one can use $\mathcal{E}$ to construct other schemes capable of homomorphically evaluate any circuit of fixed depth, i.e. to construct a family of levelled FHE schemes. Thereafter, this achieves a theoretical construction of an FHE scheme.

Definition 1.7.1 (Bootstrappable scheme)**.** *Let $\mathcal{E}$ be an homomorphic encryption scheme with compact evaluation of circuits in $\mathcal{C}$, and let D be its decryption circuit. We say that $\mathcal{E}$ is bootstrappable if $D \in C_{\mathcal{E}}$.*

This self-reference property permits to implement an algorithm that can "update" or "clean" noisy ciphertexts, the price being a change of keys. In effect, let π be a plaintext and suppose that a ciphertext ψ encrypting π is the output of an homomorphic circuit of high depth allowed by the scheme. The ciphertext can be used in homomorphic computations, but not for long, since adding too much gates would trespass the maximum allowed depth. Is it possible to obtain another ciphertext ψ' that also encrypts π and such that it is useful for more homomorphic operations than ψ? In general, it is straightforward to construct other ciphertexts encrypting the same thing because of malleability – just "add" encryptions of 0 or "multiply" encryptions of 1. Gentry's idea is far more deep: encrypt ψ with another public key (obtaining two layers of encryption) and homomorphically run the decryption circuit over this message and encryptions of the first key. The result is a new encryption of π under a new key. Figure 1.2 show this idea.

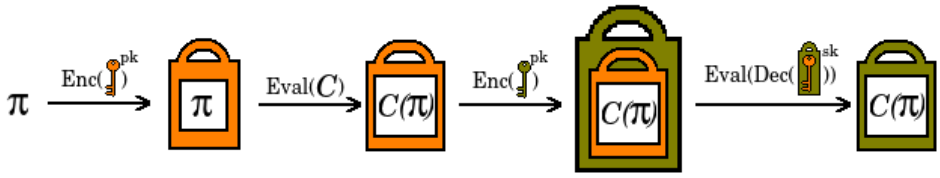

Figure 1.2: The re-encrypt procedure

Algorithm 1 The Re-Encrypt algorithm

Require: A ciphertext ψ encrypting a message π under public-key pk, another public-key pk', an encryption of the old secret key $\tau = \mathsf{Enc}(\mathsf{pk}', \mathsf{sk})$, and the decryption circuit $D_{\mathcal{E}}$.

Ensure: A ciphertext ψ' encrypting π under pk'.

1: Set $\psi_2 \leftarrow \mathsf{Enc}(\mathsf{pk}', \psi)$

2: Set $\psi' \leftarrow \mathsf{Eval}(\mathsf{pk}', D_{\mathcal{E}}, \tau, \psi_2)$.

3: Output ψ'.

1.7.1 Circular security

The bootstrapping procedure uses public encryptions of secret-keys. This calls into question the security of such encryptions: is an attacker more successful if he acquires ciphertexts of the form $\mathsf{Enc}(\mathsf{pk}, [\mathsf{sk}]_i)$, where $[\mathsf{sk}]_i$ are the secret-key bits? Or if she can distinguish such encryptions from regular ciphertexts? To the day, this question remains open. Most known FHE schemes rely on the hypothesis that they are in fact circular secure. This seems reasonable, because on one hand, it is not clear how to use encryptions of secret keys (or the ability to distinguish them from regular ciphertexts, in message interception scenarios) to win advantage in any security game or to break schemes. On the other hand, there are counterexamples of public encryption schemes that are IND-CPA secure in normal use, but security fails completely when participants are encrypting secret keys. This is becoming alarming, since some counterexamples are constructed with LWE instances (see [ABBC10, BHW15, KW16]).

Addressing this, in [BV11b] Brakerski and Vaikuntanathan proposed the first homomorphic encryption scheme proven to be Key-Dependent Message secure, and in particular, circular secure.

1.8 Hard problems used in FHE

As any other encryption scheme, FHE schemes need supposedly (or proven) hard underlying problems. They enjoy many of them, mostly coming from lattice-based cryptography. We list and define these problems here. For the sake of completeness, let us define a lattice. More details are given in 3.2.2 of Chapter 2.

A *lattice* L of $\mathbb{R}^n$ is a discrete subgroup of $(\mathbb{R}^n, +)$ which spans $\mathbb{R}^m$ as a $\mathbb{R}$-vector space. The integer m is called the *dimension* of L. A lattice may be described in terms of a linearly independent set $\mathbf{B} \subset L \subset \mathbb{R}^n$, which $\mathbb{Z}$-spans L and $\mathbb{R}$-spans $\mathbb{R}^n$. The set $\mathbf{B}$ is called a *basis* of L. All bases of L have cardinality m, and if $m = n$ we say that L is a *full-rank* lattice.

Let $\mathcal{B}(r)$ denote the closed $\mathbb{R}^n$-ball centered at 0 of radius r with respect to the Euclidean norm $||x||_2 := \sqrt{\sum_{i=1}^{n} x_i^2}$. For a lattice L of dimension m, define the sequence of successive minima as $\lambda_1(L) = \min_{v \in L \setminus \{0\}} ||v||_2$ and for $i = 2, \ldots, m$,

$$\lambda_i(L) = \min_{v \in L \setminus \mathcal{B}(\lambda_{i-1}(L))} ||v||_2.$$

All of the following problems are lattice-based.

Problem 1.8.1 (SVP,γ-SVP)**.** *Given a lattice L, the Shortest Vector Problem is to find a non-zero vector $v \in L$ of smallest possible norm $||v||_2 = \lambda_1(L)$. The γ-SVP variant is to find a non-zero vector of norm at most $\gamma \cdot \lambda(L)$.*

Problem 1.8.2 (GapSVP$_\beta$)**.** *Given a lattice L and $\beta \in \mathbb{R}^+$, the Gap Shortest Vector Problem is to decide whether $\lambda_1(L) \leq 1$ or $\lambda_1(L) \geq \beta$*

Problem 1.8.3 (CVP)**.** *Given a lattice L and a vector $v \in \mathbb{R}^n$ (not necessarily in L), the Closest Vector Problem is to find the vector in L closest to v. The CVP$_\gamma$ variant consists in finding a vector in L with at distance at most γ from v.*

Problem 1.8.4 (BDDP)**.** *The Bounded Distance Decoding Problem consists in solving the closest vector problem with the promise that* $\min_{l \in L} ||v - l||_2 \leq \lambda_1(L)/2$.

Problem 1.8.5 (SIVP)**.** *Given a lattice L of dimension n, the Smallest Independent Vectors Problem is to find n linearly independent lattice vectors $v_1 \ldots, v_n$ such that* $\max_i ||v_i||_2 \leq \lambda_n(L)$*. The γ-SIVP variant is to find n l.i. vectors such that* $\max_i ||v_i||_2 \leq \gamma \lambda_n(L)$

Problem 1.8.6 (SBP)**.** *Given a basis $\mathbf{B}$ of a lattice L, the Shortest Basis Problem consists in outputting a basis $\mathbf{B}'$ of the same lattice such that the length of vectors in $\mathbf{B}'$ is as short as possible.*

In 2005, Regev introduced the celebrated *Learning With Errors Problem* as a generalization of the parity learning problem. He proved that LWE is hard if GapSVP$_\gamma$ and SIVP are quantum-hard.

Problem 1.8.7 (LWE)**.** *Let q be an integer and χ be an error distribution on $\mathbb{Z}$. Let $s \leftarrow \mathbb{Z}_q^n$ be a secret vector, and define $A_{q,\chi}$ as the distribution that first sets $a \xleftarrow{R} \mathbb{Z}_q^n, e \leftarrow \chi$ and then outputs $(a, a \cdot s + e)$. The Learning With Errors Problem is to deduce s given polynomially many samples of $A_{q,\chi}$. Its decision variant (called Decision-LWE) consists in distinguishing m samples of $A_{q,\chi}$ from uniformly random samples of $\mathbb{Z}_q^n \times \mathbb{Z}_q$.*

Problem 1.8.8 (RLWE). *The Ring-LWE Problem is the analog of LWE but replacing $\mathbb{Z}_q^n$ with the polynomial ring $R_q = \mathbb{Z}_q[x]/(f(x))$ where f is a monic n-th degree polynomial. Typically, $f(x) = x^n + 1$ and n is a power of 2 (therefore, f is the $2n$-th cyclotomic polynomial)*

The following are two integer-based problems. The first was used in the construction of FHE over the integers, while the second is a combinatorial problem needed to "squash" the decryption circuit of Gentry's ideal lattice scheme, in order to make it bootstrappable.

Problem 1.8.9 (AGCD). *The Approximate Greatest Common Divisor Problem is to find a secret number p given a list of m integers of the form $x_i = q_i p + r_i$ for integers q_i, r_i sampled from possibly different distributions over $\mathbb{Z}$.*

Problem 1.8.10 (SSSP). *The Sparse Subset Sum Problem consists in deciding whether a set of integers contains a small subset such that the sum of its elements equals 0 in $\mathbb{Z}_q$. More precisely, given a set A of t integers, the SSSP asks to decide if there is a subset $A' \subset A$ of cardinality at most $s \ll t$ such that $\sum_{a \in A'} a \equiv 0 \mod q$.*

Finally, we describe polynomial ring problems. For a distribution ξ over a ring R, let $\xi^\times$ be the distribution that samples repeatedly from ξ until the output is invertible.

Problem 1.8.11 ($\mathrm{DSPR}_{\phi,q,\xi}$). *Let ϕ be a monic polynomial of degree n, $q \in \mathbb{Z}$ a prime and ξ a distribution over the ring $R_q = \mathbb{Z}_q[x]/(\phi(x))$. The Decisional Small Polynomial Ratio problem is to distinguish between (i) a polynomial h of the form $h = g/f$ where $g \leftarrow \xi$ and $f \leftarrow \xi^\times$, and (ii) a polynomial h' sampled uniformly at random from R_q.*

We propose in [GVP16] two new factorization problems in cyclotomic rings, that support the security in our solution to the betrayal problem. We call these problems the *Special Factors Problem* and the *$\xi^\times$-Common Divisor Problem*.

Problem 1.8.12 (SFP). *Let ξ be a distribution on some ring R. Let $a, b \in \xi^\times$ and let $c = a \cdot b \in R$. The Special Factors Problem consists in finding a, b given c and a test function $T : R \to \{0, 1\}$ that outputs 1 if the input is in $\{a, b\}$ and 0 otherwise.*

Problem 1.8.13 ($\xi^\times$-CDP). *Let ξ be a distribution on some ring R. Let $a, b \leftarrow \xi^\times$ and $y \xleftarrow{R} R$. Let $u = a \cdot y \in R$ and $v = b \cdot y \in R$. The $\xi^\times$-Common Divisor Problem is to guess (a, b, y) given (u, v) and a test function $T : R \to \{0, 1\}$ that outputs 1 if the input is in $\{a, b, y\}$ and 0 otherwise.*

Remark. In Chapter 4 and in [GVP16], these two problems are called the "Small Factors Problem" and the "invertible Gaussian GCD Problem". This is because the *specialness* of polynomials a, b in our cyclotomic ring scenario is that they have small l^2 norm.

1.9 Conclusion

In this Chapter, we approached the area of fully homomorphic encryption. We highlighted some practical applications, relations to other cryptographic properties, and described how different generations of proposals are ordered today in five branches. Also, we defined fully homomorphic encryption from the theoretical point of view and discussed some computational problems that lay in the foundations of these constructions. We believe that this review is of help to FHE-neophyte readers; to more informed readers wishing to follow the vast FHE literature in a certain angle (efficiency, applications, security, etc), please refer to appendix A, where we cite important references in a list arrangeable by different criteria.

Chapter 2

Gentry's ideal lattices scheme

Contents

2.1 Introduction 27

2.2 Preliminaries: Ideal lattices 28

2.2.1 Lattices 28

2.2.2 Notation 31

2.2.3 Ideal lattices 31

2.2.4 Polynomial principal ideal lattices 31

2.2.5 Ring operations in $\mathbb{Z}^n$ 33

2.3 The basic homomorphic scheme 33

2.3.1 Does it work? Why does it work? 34

2.3.2 Key generation and correctness of the algorithms 37

2.3.3 Choice of the polynomial ring 37

2.3.4 Lattice I: choice of the plaintext space 38

2.3.5 Lattice J: ciphertext space 38

2.3.6 Generating the bases 40

2.3.7 Role of the representative sampling function 41

2.3.8 Correctness of encryption and decryption 41

2.3.9 Correctness of homomorphic operations 42

2.3.10 Maximum depth of allowed circuits 43

2.4 Security 44

2.4.1 The Ideal Coset Problem 44

2.5 Natural density of some classes of lattices 45

2.5.1 Natural density of diagonal elements of HNF's 46

2.5.2 Natural density of coprime lattices 47

2.5.3 Natural density of odd ideal lattices 48

2.5.4 Remark on natural density of circulant matrices with odd determinant 48
2.6 Proof-of-concept implementation 49
2.6.1 Basic functions 50
2.6.2 Scheme algorithms 50
2.6.3 Performances 52
2.7 Conclusion 54

2.1 Introduction

In this Chapter, we revisit Gentry's ideal lattice homomorphic scheme. It has been widely mentioned that this scheme, while achieving fully homomorphic encryption for the first time, is not even remotely competitive with contemporary encryption schemes regarding efficiency. The main bottleneck is ciphertext and public-key size, which grows exponentially with the level of security. Nonetheless, understanding the scheme's construction is very rewarding and we firmly believe that it deserves study, at least with similar deepness than later homomorphic schemes. New cryptographic ideas and mathematical birds live here, and it would be a pity to pass them by because the scheme does not meet our prompt needs.

Our objective is to present the scheme in a sufficient level of detail, as a tribute to Gentry's original Ph.D. dissertation. The first idea leading to the solution is remarkably simple: it reduces the so-called *Holy Grail of Cryptography* quest to the problem of finding an algebraic ring with some class of ideals that support hard problems for security and specific operations. Of course, it would be terribly unfair to state that Gentry's contribution relied on finding a proper ring for a generic construction. But still, his solution can be understood with the following image.

Ideal + lattices = Ideal Lattices. A cryptographer C with infinite amounts of intelligence and ignorance is asked to solve the fully homomorphic encryption problem. To solve this, he would most likely invent algebra, take a ring R and define the following encryption scheme:

> Let I be some ideal of R, let $m \in R/I$ an encoding of a message, and define ciphertext $\mathsf{Enc}(m) = m + i$ where $i \in I$.

This is indeed homomorphic, since if $\oplus, \otimes$ denote both ring operations,

$$\mathsf{Enc}(m) \oplus \mathsf{Enc}(m') \text{ encrypts } m \oplus m',$$
$$\mathsf{Enc}(m) \otimes \mathsf{Enc}(m') \text{ encrypts } m \otimes m'.$$

He would then search for a ring R that enjoys

- random efficient sampling of $\alpha + I$ for $\alpha \in R/I$ to encrypt[1],
- an "ideal annihilation" procedure $m + xI \mapsto m$ to decrypt,
- connection to hard problems.

If in addition C met another cryptographer C' with unbounded knowledge of recent research, he would be inclined to believe that the answer may be found in lattice-based cryptography. This was, in fact, the solution to the FHE problem: use

[1] C would understand on-the-spot that an homomorphic encryption scheme cannot be deterministic, since it would be possible to solve equations on ciphertexts.

polynomial rings and specialize the latter basic construction using two sets of ideal lattices.

A study on some natural densities. Sampling random coprime lattices has some subtleties. We collect some results and quantify the structure of random bases in Hermite Normal Forms, which represent special bases for lattices.

Overview of this Chapter. In section 2.2, we define ideal lattices. In 2.3 we reconstruct Gentry's basic *somewhat homomorphic* scheme, which supports a bounded amount of homomorphic operations. A discussion of the underlying security follows in section 2.4, and a proof-of-concept implementation is presented in 2.6. We finish with the study of natural densities of some lattice bases in 2.5.

2.2 Preliminaries: Ideal lattices

2.2.1 Lattices

Definition 2.2.1 (Lattice)**.** *A lattice in $\mathbb{R}^n$ is a discrete subgroup of $(\mathbb{R}^n, +)$ which spans $\mathbb{R}^m$ as a $\mathbb{R}$-vector space.*

Every lattice L has a minimal subset of m elements $B \subset L$ such that $L = \bigoplus_{b \in B} \mathbb{Z}b$. When $n \geq 2$ there is an infinite number of such sets and we call them *bases* of L. The integer m does not depend on the chosen basis and is called the *rank* of L, therefore, to describe a lattice, only one basis is required.

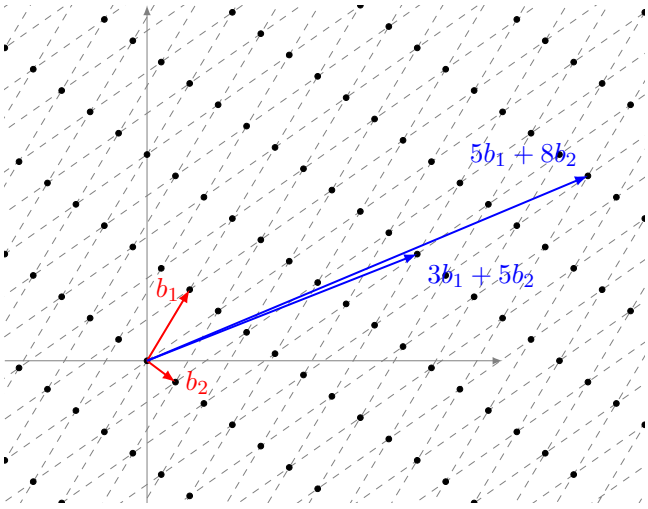

Figure 2.1: A lattice of $\mathbb{R}^2$ and two bases.

Given a basis of rank m in matrix form $\mathbf{B} = (\mathbf{b}_1|\mathbf{b}_2|\cdots|\mathbf{b}_m)$, the quantity $|\det(\mathbf{B})| := \sqrt{\det({}^t\mathbf{B}\mathbf{B})}$ is an invariant of L, which we note by $\det L$. Let the

fundamental parallelepiped of the basis $\mathbf{B}$ be the translated convex hull delimited by the column vectors in $\mathbf{B}$. We note this parallelepiped $P(\mathbf{B})$:

$$P(\mathbf{B}) = \left\{ \sum_{1 \leq i \leq m} x_i \mathbf{b}_i \in \mathbb{R}^n \ ; \ x_i \in [-1/2, 1/2) \right\}$$

By definition, the volume of $P(\mathbf{B})$ equals the determinant of the lattice:

$$\det(L) = \mathbb{R}^n\text{-measure}(P(\mathbf{B})) \text{ for all basis } \mathbf{B}.$$

Let L be a full rank lattice. If $x \in \mathbb{R}^n$, we note by $x \bmod L$ the equivalence class of x in the quotient group $\mathbb{R}^n/L$. We use the fundamental parallelepiped $P(\mathbf{B})$ as the set of representatives of the equivalence classes when describing L with the basis $\mathbf{B}$. Therefore, it is natural to note by $x \bmod \mathbf{B}$ the single element in the intersection $(x \bmod L) \cap P(\mathbf{B})$. To reduce a vector $x \in \mathbb{R}^n$ modulo $\mathbf{B}$, use $\mathbb{Z}$-linear combinations of vectors in $\mathbf{B}$ to find the equivalent vector inside $P(\mathbf{B})$. In other words,

Lemma 2.2.1. *Let $x \in \mathbb{R}^n$ and $\mathbf{B}$ a basis of a full-rank lattice L. Then $x \bmod \mathbf{B} = x - \mathbf{B}\lfloor \mathbf{B}^{-1} x \rceil$, where $\lfloor v \rceil := (\lfloor v_1 \rceil, \lfloor v_2 \rceil, \cdots, \lfloor v_n \rceil)$ is the closest $\mathbb{Z}^n$ vector to v.*

Proof : The invertibility of $\mathbf{B}$ comes from the fact that $\det L \neq 0$. We have

$$x - \mathbf{B}\lfloor \mathbf{B}^{-1} x \rceil = \mathbf{B}(\mathbf{B}^{-1} x - \lfloor \mathbf{B}^{-1} x \rceil) = \mathbf{B} y,$$

where $y = \mathbf{B}^{-1} x - \lfloor \mathbf{B}^{-1} x \rceil$ is a $\mathbb{R}^n$-vector with coordinates in $[-1/2, 1/2)$. ■

In the scheme, two *coprime* lattices I, J will define the plaintext and ciphertext space respectively.

Definition 2.2.2 (Relatively prime lattices)**.** *Two lattices L, L' of $\mathbb{Z}^n$ are relatively prime (or coprime) if their sum is $\mathbb{Z}^n$. We note this $(L, L') = 1$.*

Keys in Gentry's scheme are lattice bases, which we identify with square matrices. The private-key is a "good" basis in the orthogonality defect sense and the public key is an eccentric, large basis of the same lattice. More precisely, this large basis is the *Hermite Normal Form* of the lattice.

Definition 2.2.3 (Hermite Normal Form)**.** *A square matrix $\mathbf{H} \in \mathcal{M}_{n \times n}(\mathbb{Z})$ of entries h_{ij} is in Hermite Normal Form if*

1. *$\mathbf{H}$ is lower triangular, semi-positive definite: $h_{kk} \geq 0$ and $h_{ij} = 0$ for $i < j$.*
2. *The pivot (first nonzero entry) of a column is strictly below the pivots of precedent columns.*
3. *Entries to the right of pivots are zero and elements to the left of pivots are nonnegative, strictly smaller than the pivot.*

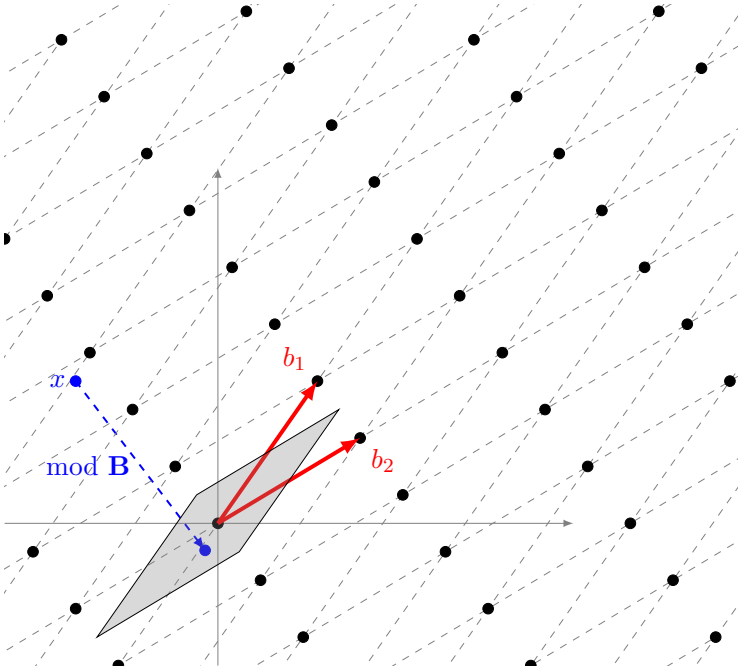

Figure 2.2: A basis modular reduction in $\mathbb{R}^2$

For instance, the following matrices are in HNF.

$$\begin{pmatrix} 17 & 0 & 0 & 0 \\ 2 & 3 & 0 & 0 \\ 50 & 0 & 51 & 0 \\ 6 & 6 & 6 & 7 \end{pmatrix}, \begin{pmatrix} 12 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 10 & 13 & 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 0 & 0 & 0 \\ 78 & -26 & 10 & 0 & 0 & 0 \\ 13 & 13 & 1 & 14 & 0 & 0 \end{pmatrix}.$$

A lattice is uniquely defined by a matrix in HNF, and according to this we will define the HNF of a lattice as its unique basis in HNF. This allows us in section 2.3.5 to provide a characterization of coprime lattices.

2.2.2 Notation

In this Chapter, all lattices are of full rank ($m = n$) unless stated otherwise. In addition, if $\mathbf{B} = \{\mathbf{b}_1, \cdots, \mathbf{b}_n\} \subset L$ is a basis of the lattice L, we will not distinguish between $\mathbf{B}$ and the matrix $B = (\mathbf{b}_1|\mathbf{b}_2|\cdots|\mathbf{b}_n)$. We note by I_n the $n \times n$ identity matrix and $\mathbf{e}_k$ the k-th column of I_n.

2.2.3 Ideal lattices

Let G be an additive group and $\alpha : G \to (\mathbb{R}^n, +)$ a group homomorphism, then $\alpha(G)$ is a lattice. This happens because α transports the additive structure of G to points in $\mathbb{R}^n$ in order to generate linear integer combinations of them. Of course, this can also be performed with rings, considering the additive operation, and it proves to be fruitful as it allows to define $\mathbb{R}^n$-products (if α is suitable) and ideal lattices.

Definition 2.2.4 (Ideal lattice)**.** *Let A be a ring and $\alpha : A \to \mathbb{R}^n$ and additive ring homomorphism. An ideal lattice of $\mathbb{R}^n$ is a lattice of the form $\alpha(I)$ for a principal ideal $I \subset A$.*

Remark. A definition with non-principal ideals seems worthy of attention. However we will not address this, since in polynomial quotient rings the non-principal case can be regarded as a linear algebra problem considering reduction of union of basis coming precisely from principal ideals.

2.2.4 Polynomial principal ideal lattices

Consider the ring $R = \mathbb{Z}[x]/(f(x))$ where f is a monic polynomial of degree n, and the natural homomorphism $\alpha : R \to \mathbb{Z}^n$ given by

$$\alpha : \quad a(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} \quad \mapsto \quad a = (a_0, \ldots, a_{n-1}).$$

We can now define a precise ideal lattice: if $I = (v)$ is a principal ideal of R, then $\alpha(I)$ is a lattice of $\mathbb{R}^n$ with a remarkable basis:

Lemma 2.2.2. *Let $v \in R$, $I = (v)$ and $v_i = v \times X^i \mod f(X)$ for $i = 1, \ldots, n-1$. Then a basis of the ideal lattice $\alpha(I)$ is given by a linearly independent subset of*

$$\mathrm{rot}(v) := \{v, v_1, v_2, \cdots, v_{n-1}\}.$$

Proof : If $w \in (v)$, then $w = v \times a = v \times \left(\sum_{i=0}^{n-1} a_i X^i\right)$ for some $a \in A$. Hence $\alpha(w) = \sum_{i=0}^{n-1} a_i v_i \in \mathrm{rot}(v)$. Conversely, if (the vector) $w \in \mathrm{rot}(v)$, then $w = \sum_{i=0}^{n-1} a_i \alpha(v_i) = \alpha(P)$ for $P = \sum_{i=0}^{n-1} a_i v_i$. □

Remark. The ideal lattice $\alpha(I)$ may not be of full rank (hence the "given by a linearly independent set of" statement). For instance, if $R = \mathbb{Z}[x]/(x^n - 1)$ then the ideal lattice $\alpha((1+x+x^2+\cdots+x^{n-1}))$ is of rank 1. This happens because the polynomial $1 + x + x^2 + \cdots + x^{n-1}$ is not invertible in R. We do not worry much about this, and we will deal only with full-rank lattices, thanks to the choice of the polynomial ring and the ideal generator.

From now on, we will not distinguish between the polynomial $w \in R$ and the corresponding vector $\alpha(w) \in \mathbb{Z}^n$. Sometimes, we will not distinguish between a polynomial ideal (v) and the corresponding ideal lattice $\alpha((v))$.

Definition 2.2.5 (Rotation basis)**.** *The set* $\{v_i = v \times x^i \mod f(x); i = 0, \ldots, n-1\}$ *is called the rotation basis of* (v)*. The matrix* $\mathrm{rot}(v) = (v|v_1|\ldots|v_{n-1})$ *is called the rotation matrix of* v*.*

Proposition 2.2.3 (Inversing rotation matrices)**.** *Let* $v \in R^\times$*. Then* $\mathrm{rot}(v) \times \mathrm{rot}(v^{-1}) = I_n$*.*

Proof: It follows directly from the fact that for every $a, b \in \mathbb{Z}^n$, $a \times b = \mathrm{rot}(a)b = \mathrm{rot}(b)a$. $\square$.

Example 2.2.1. If $n = 4$ and $R = \mathbb{Z}[x]/(x^4 + 2)$, the ideal lattice $\alpha((v))$ where $v(x) = 1 + x^2 = (1, 0, 1, 0)$ is generated by

$$\begin{aligned} v(x) &\mod f(X) = 1 + x^2 = (1, 0, 1, 0), \\ xv(x) &\mod f(X) = x + x^3 = (0, 1, 0, 1), \\ x^2v(x) &\mod f(X) = -2 + x^2 = (-2, 0, 1, 0), \\ x^3v(x) &\mod f(X) = -2x + x^3 = (0, -2, 0, 1), \end{aligned}$$

i.e. by the columns of the matrix

$$\mathrm{rot}(1 + x^2) = \begin{pmatrix} 1 & 0 & -2 & 0 \\ 0 & 1 & 0 & -2 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

This matrix is invertible, and therefore the corresponding ideal lattice is of full-rank, with $\det(\mathrm{rot}(1 + x^2)) = 9$.

Let us give another example which we will find very often in fully homomorphic encryption schemes, in various forms.

Example 2.2.2. If $f(x) = x^n - 1$, the rotation of a polynome $a(x) \mapsto xa(x) \mod f$ corresponds to a coefficient-wise rotation of $\alpha(a)$:

$$(a_0, \ldots, a_{n-2}, a_{n-1}) \mapsto (a_{n-1}, a_0, \ldots, a_{n-2}),$$

therefore the ideal lattice (a) is generated by the columns of the *circulant* matrix

$$\mathrm{rot}(a_0, \ldots, a_{n-1}) = \begin{pmatrix} a_0 & a_{n-1} & a_{n-2} & \cdots & a_1 \\ a_1 & a_0 & a_{n-1} & & a_2 \\ a_2 & a_1 & a_0 & & a_3 \\ \vdots & & & \ddots & \vdots \\ a_{n-1} & a_{n-2} & a_{n-3} & \cdots & a_0 \end{pmatrix}.$$

Lattices generated by circulant bases (or equivalently, ideal lattices of the ring $\mathbb{Z}[x]/(x^n - 1)$) are called *cyclic lattices*. They verify the following property: if $(a_1, a_2, \ldots, a_n) \in L$ then $(a_2, a_3, \ldots, a_n, a_1) \in L$. Notice also that all of these lattices contain vectors in the lines spanned by $(1, 1, 1, \ldots, 1)$ and $(1, -1, 1, \ldots, \pm 1)$.

2.2.5 Ring operations in $\mathbb{Z}^n$

Given that we identified $\mathbb{Z}^n \simeq R = \mathbb{Z}[x]/(f(x))$, ring operations are inherited by structure transport. This way, for all $v, w \in \mathbb{Z}^n$ we define

$$\begin{aligned} &\text{addition } v + w := \alpha(v(x) + w(x)), \\ &\text{product } v \times w := \alpha(v(x) \times w(x)) \end{aligned}$$

Of course, the first operation is usual addition in $\mathbb{Z}^n$ and the second operation is a convolution product in $\mathbb{Z}^n$. A third operation was already discussed, that allows to reduce a vector in $\mathbb{R}^n$ modulo a basis $\mathbf{B}$ of a lattice L. We are now ready to define the scheme.

2.3 The basic homomorphic scheme

The scenario of this scheme is as follows

- We fix a dimension $n \geq 2$ and a monic polynomial f of degree n,
- let $R = \mathbb{Z}[x]/(f(x))$,
- let I be an ideal lattice with a base $\mathbf{B}_I$,
- let J be an ideal lattice relatively prime to I (i.e. $I + J = \mathbb{Z}^n$), with two bases $\mathbf{B}_J^{\mathsf{sk}}, \mathbf{B}_J^{\mathsf{pk}}$, and finally
- choose an algorithm $\mathsf{Samp} : \mathbb{Z}^n/(I) \to \mathbb{Z}^n$ that given a vector x, samples randomly from $x + (I)$.

The plaintext space in the scheme is $\mathbb{Z}^n \cap P(\mathbf{B}_I)$ (a small, quite orthogonal parallelepiped) and ciphertext space is $\mathbb{Z}^n \cap P(\mathbf{B}_J^{\mathsf{pk}})$ (a huge, very eccentric parallelepiped). The elements $n, f, I, J, \mathbf{B}_I$ are all public, and the keys are

$$\mathsf{pk} = \{\mathbf{B}_J^{\mathsf{pk}}\}, \qquad \mathsf{sk} = \{\mathbf{B}_J^{\mathsf{sk}}\}.$$

The successive choices of $f, I, \mathbf{B}_J^{\mathsf{sk}}, \mathbf{B}^{\mathsf{pk}}$ are essential and non-trivial to understand and eventually implement the scheme. Let us give a second look at these elements in secure scenarios:

- n is large to prevent lattice reduction attacks ($n \geq 2^9$ avoids LLL and BKZ reduction, for instance),
- f is of the form $X^n + h(X)$ where $\deg(h)$ is small (for instance $h(X) = -1$), in order to control product norms,

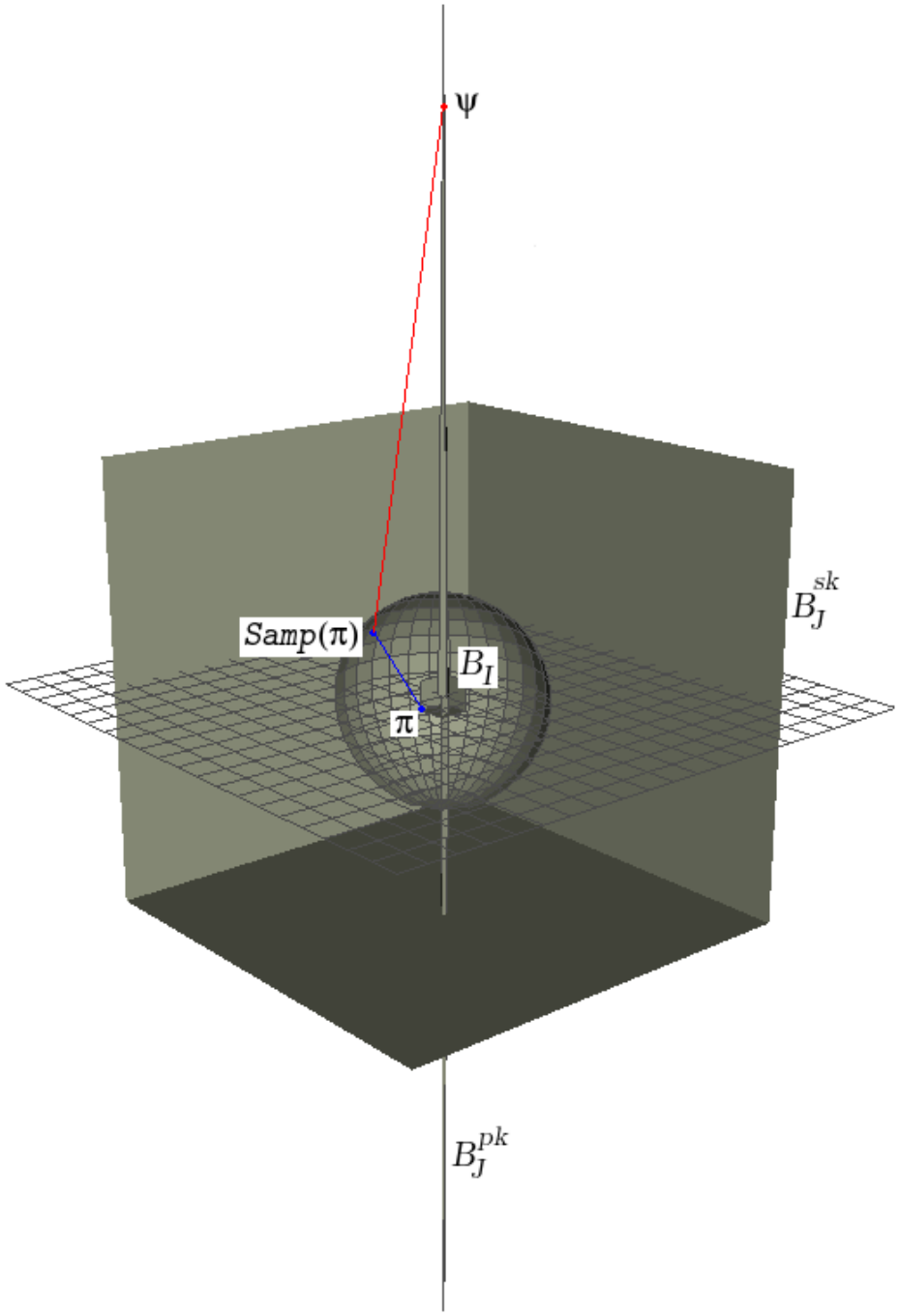

Figure 2.3: Fundamental parallelepipeds in Gentry's scheme

– the parallelepiped $\mathbf{B}_I$ consists of small vectors (the binary case is $I = (2)$ and $\mathbf{B}_I = 2I_n$),

– $\mathbf{B}_J^{\mathsf{sk}}$ is quite orthogonal, with large volume and $P(\mathbf{B}_J^{\mathsf{sk}})$ contains $P(\mathbf{B}_I)$,

– $\mathbf{B}_J^{\mathsf{pk}}$ is on the contrary, very eccentric (it is defined by large and almost parallel vectors),

– the algorithm $\mathsf{Samp} : \mathcal{P} \to \mathbb{Z}^n$ randomly chooses a representative of $x + I$ inside a ball of fixed radius l_{Samp} contained in $P(\mathbf{B}_J^{\mathsf{sk}})$.

The scheme is described in 6.

Algorithm 2 Gentry's homomorphic ideal lattice scheme

function $\mathsf{Keygen}(\lambda)$
  Generate $\mathbf{B}_J^{\mathsf{sk}}$ and $\mathbf{B}_J^{\mathsf{pk}}$ randomly from a security parameter.
end function

---

function $\mathsf{Enc}(\mathsf{pk}, \pi \in P(\mathbf{B}_I))$
  To encrypt message π, compute

$$\psi \leftarrow \mathsf{Samp}(\pi) \mod \mathbf{B}_J^{\mathsf{pk}}$$

  and output ψ. It is an integer vector of the form $\pi + i + j$ where $i \in I, j \in J$, and it is inside $P(\mathbf{B}_J^{\mathsf{pk}})$.
end function

---

function $\mathsf{Dec}(\mathsf{sk}, \psi)$
  To decrypt ψ, compute

$$\mu = (\psi \bmod \mathbf{B}_J^{\mathsf{sk}}) \bmod \mathbf{B}_I = (\psi - \mathbf{B}_J^{\mathsf{sk}} \lfloor (\mathbf{B}_J^{\mathsf{sk}})^{-1} \psi \rceil) \bmod \mathbf{B}_I$$

  and output μ.
end function

---

function $\mathsf{Eval}((\mathsf{pk}, \mathrm{C}, \psi_1, \cdots, \psi_t)$
  Perform the circuit C on inputs $\psi_1, \cdots, \psi_t$ replacing each "+" and "×" gate of C by addition and multiplication modulo $\mathbf{B}_J^{\mathsf{pk}}$ respectively. Output the result.
end function

2.3.1 Does it work? Why does it work?

Before looking at correctness of the algorithms, we will have a glance at what happens when a plaintext is encrypted, then decrypted. Suppose $I = \alpha((2))$. Let π be a plaintext (π is a binary vector). The first step of encryption is to define $\pi' \leftarrow \mathsf{Samp}(\pi)$. Thus, $\pi' = \pi + i$ for $i \in I$ (a binary vector plus multiples of 2

component wise), in clear, $\pi' = \pi \mod I$. The next step is to reduce $\pi' \mod \mathbf{B}_J^{\mathsf{pk}}$. This will map π' to a point $\psi = \pi' + j$ for $j \in J$. As the generating vectors in $\mathbf{B}_J^{\mathsf{pk}}$ are long and almost parallel, it is more than likely that this point ψ will be very far along the direction $\mathbf{e}_n$ and outside $P(\mathbf{B}_J^{\mathsf{sk}})$: this is the ciphertext. To decrypt, the first step is to recover π' from ψ. As $\psi = \pi' \mod J$, the reduction $\psi \mod \mathbf{B}_J^{\mathsf{sk}}$ will output π'

$$\textit{if and only if } \pi' \in P(\mathbf{B}_J^{\mathsf{sk}}).$$

This is ensured by

$$\mathsf{Samp}(P(\mathbf{B}_I)) \subset P(\mathbf{B}_J)^{\mathsf{sk}}.$$

Then, recovering π is easy. This is why the sampling procedure outputs must be l_2-bounded by a constant l_{Samp}. What about homomorphic operations of ciphertexts? Will they decrypt correctly? Let $\psi_1 = \mathsf{Enc}(\mathsf{pk}, \pi_1), \psi_2 = \mathsf{Enc}(\mathsf{pk}, \pi_2)$. Then

$$\psi_1 + \psi_2 = (\pi_1 + \pi_2) + i' + j' \text{ for some } i' \in I, j' \in J, \tag{2.1}$$

$$\psi_1 \times \psi_2 = (\pi_1 \times \pi_2) + i'' + j'' \text{ for some } i'' \in I, j'' \in J, \tag{2.2}$$

i.e. they will decrypt to $\pi_1 + \pi_2$ and $\pi_1 \times \pi_2$ respectively

$$\textit{if and only if } (\pi_1 + \pi_2) + i' \in P(\mathbf{B}_J)^{\mathsf{sk}} \textit{and } (\pi_1 \times \pi_2) + i'' \in P(\mathbf{B}_J)^{\mathsf{sk}}.$$

As $P(\mathbf{B}_J)^{\mathsf{sk}}$ is bounded in $\mathbb{R}^n$, it is not possible to carry out an arbitrary number of correct homomorphic operations, since at some point the latter vectors will get out of $P(\mathbf{B}_J)^{\mathsf{sk}}$, ruining decryption. This also means that the growth of the $\mathbb{Z}^n$ operations will directly affect the number of homomorphic operations that can be correctly performed. In the next section, we will describe kind choices to this end, that look out to security as well.

2.3.2 Key generation and correctness of the algorithms

We will explain how to choose parameters and generate keys to ensure correctness of the scheme and its homomorphic operations.

2.3.3 Choice of the polynomial ring

The choice $R = \mathbb{Z}[x]/(f(x))$ has consequences in the overhead of operations in $\mathbb{Z}^n$. For instance, if f has lots of monomials, the product of two polynomials $a, b \subset R$ is likely to have larger l_2 norm than when using a sparse polynomial f, since the reduction modulo f involves more euclidean divisions. For monic polynomials of degree n, let us define the product overhead

$$\lambda(f) := \max_{u,v \in \mathbb{Z}[x]/(f(x))} \frac{||u \times v||_2}{||u||_2 ||v||_2}.$$

A good choice of ring involves a polynomial with small product overhead. The reader may consult [Gen09b], § 7.4 for more details. We highlight the fact that if f is of the form $f(x) = x^n - h(x)$ for a polynomial of degree $\leq n/2$, then $\lambda(f) \leq c\sqrt{n}$ for some constant c depending on f. For instance,

$$\lambda(x^n \pm 1) = \sqrt{n}.$$

For completeness, let us prove this.

Lemma 2.3.1. *Let $R = \mathbb{Z}[x]/(x^n - 1)$. If $a \times b = z = (z_0, \cdots, z_{n-1}) \in R$, then*

$$z_k = \sum_{j=0}^{n-1} a_{k-j \bmod n} b_j \quad \text{and} \quad ||z||_2 \leq \sqrt{n}\, ||a|| \cdot ||b||.$$

Proof : Given that $X^{n+k} \equiv X^k \mod (X^n - 1)$ (the following are equations in the ring),

$$\begin{aligned}
\sum_{i,j=0}^{n-1} a_i b_j X^{j+i} &= \left(\sum_{j=0}^{n-1} \sum_{i=0}^{n-j-1} + \sum_{j=0}^{n-1} \sum_{i=n-j}^{n-1} \right) a_j b_{i-j} X^{i+j} \\
&= \sum_{i=0}^{n-1} \sum_{j=0}^{i} a_j b_{i-j} X^i + \sum_{i=0}^{n-1} \sum_{j=i+1}^{n} a_j b_{n+i-j} X^i \\
&= \sum_{i=0}^{n-1} X^i \left(\sum_{j=0}^{n-1} a_j b_{i-j \bmod n} \right).
\end{aligned}$$

In consequence, z_i, the coefficient of X^i in z, is the scalar product between a and rotations of b. The result is given as an application of Cauchy-Schwartz inequality. □

As an easy corollary, we have that if $f(X) = X^n - m$ for $m \in \mathbb{Z}$, then

$$z_k = \sum_{j=0}^{k} a_{k-j} b_j + m \sum_{j=k+1} a_{n+k-j} b_j \quad \text{and} \quad ||\mathbf{z}|| \leq C(m)\, ||\mathbf{a}|| \cdot ||\mathbf{b}||$$

for some constant $C(m)$ such that $\sqrt{n} \leq C(m) \leq \sqrt{n|m|}$. As noted in section 3.2.3, for $R = \mathbb{Z}[x]/(x^n - 1)$ the ideal lattices are *cyclic* (or *circulant*): if $(a_1, \cdots, a_n) \in L$, then $(a_n, a_1, \cdots, a_{n-1}) \in L$. This choice is elegant and allows to apply the very developed theory of circulant matrices, see for instance [Gra06]. In this chapter, we will fix $f(x) = x^n - 1$.

2.3.4 Lattice I: choice of the plaintext space

In Gentry's scheme, the message space is (a subset of)

$$\mathcal{P} = \mathbb{Z}^n \mod \mathbf{B}_I = \mathbb{Z}^n \cap P(\mathbf{B}_I),$$

where $\mathbf{B}_I$ is a matrix whose columns $\mathbb{Z}-$span I. Here are four examples in $\mathbb{Z}^3$.

$$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{pmatrix}, \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ 0 & t_{22} & t_{23} \\ 0 & 0 & t_{33} \end{pmatrix}, \begin{pmatrix} r_1 & r_3 & r_2 \\ r_2 & r_1 & r_3 \\ r_3 & r_2 & r_1 \end{pmatrix}.$$

In the first case, plaintext space $\mathcal{P}$ is $\{0,1\}^3$. In the second and third case, a message is a vector (x_1, x_2, x_3) whose components verify $|x_i| < |d_i|$ for $i = 1, 2, 3$ and $\{|x_3| < |t_{33}|, |x_2| < |t_{22}| + |t_{23}|, |x_1| < |t_{11}| + |t_{12}| + |t_{13}|\}$, respectively. The binary case $I = (2)$, $\mathbf{B}_I = 2I_n$ and $\mathcal{P} = \{0,1\}^n$ suffices for applications and is the simplest to analyze. We will of course adopt this choice, however, it raises the nontrivial question of studying the distribution of lattices that are relatively prime to $\alpha((2))$. See next section.

2.3.5 Lattice J: ciphertext space

The lattice J must be chosen relatively prime to I and have two bases – a small, quite orthogonal $\mathbf{B}_J^{\mathsf{sk}}$ and a large, eccentric $\mathbf{B}_J^{\mathsf{pk}}$. Let us begin by giving following characterization of relatively prime lattices:

Proposition 2.3.2. *Let L, L' be two lattices of $\mathbb{Z}^n$ and $\mathbf{B}, \mathbf{B}'$ their respective bases. Recall that $\mathbf{e}_i$ is the i-th column of the identity I_n. The following statements are equivalent:*

(i) $L + L' = \mathbb{Z}^n$.

(ii) (Bézout's identity for lattices) There are two matrices M, M' with integer entries such that $\mathbf{B}M + \mathbf{B}'M' = I_n$.

(iii) Let $\mathbf{H}$ and $\mathbf{H}'$ be the Hermite normal forms of $\mathbf{B}$ and $\mathbf{B}'$ respectively. The respective diagonal elements of $\mathbf{H}, \mathbf{H}'$ are relatively prime.

Proof : The equivalence (i)⇔(ii) comes from the fact that if L, L' are relatively prime, then one can write each column of the identity matrix as a linear integer combination of elements in $\mathbf{B} \cup \mathbf{B}'$, thus generating $\mathbb{Z}^n$. The equivalence (i)⇔(iii) is simply the fact that $\mathbf{H}, \mathbf{H}'$ are bases of L, L' in triangular form. Indeed, suppose that (i) holds, then for $1 \leq k \leq n$, $\mathbf{e}_k$ is a linear integer combination of columns of $(\mathbf{H}|\mathbf{H}')$, and because of the triangular structure this combination is of the form

$$\mathbf{e}_k = \sum_{i=1}^{k} \alpha_i h_i + \beta_i h'_i.$$

In particular, $1 = \alpha_k \mathbf{H}_{kk} + \beta_k \mathbf{H}'_{kk}$, proving (iii). Conversely, if (iii) holds, one can perform column operations to the rectangular matrix $(\mathbf{H}|\mathbf{H}')$ following Bézout's identity between corresponding columns, resulting in a rectangular matrix with 1's in the diagonal, consequently, its columns generate $\mathbb{Z}^n$. ■

Corollary. *Let L, L' be two lattices in $\mathbb{Z}^n$. If $(\det(L), \det(L')) = 1$ then $(L, L') = 1$.*

Remark. The converse is not true. For instance, the lattices of $\mathbb{Z}^2$ generated by $\{2\mathbf{e}_1, 3\mathbf{e}_2\}$ and $\{3\mathbf{e}_1, 2\mathbf{e}_2\}$ are relatively prime but they have equal determinants.

Let us come back at our choice of a binary plaintext $I = (2)$. Is it easy to find an ideal lattice $J = \alpha((v))$ relatively prime to I, i.e. such that $\det J$ is odd? A probability result states that the distribution of determinants of uniformly random matrices is not uniformly distributed in $\mathbb{Z}$ (this distribution is an interesting problem, see [CM00]). In fact, the Hermite normal form of a uniformly random matrix follows a structured distribution. Asymptotically (in a bound on entry-wise uniform samplings), for a matrix M there are around 40% chances that the first $n-1$ diagonal elements of the $\mathrm{HNF}(M)$ are 1 and the last equals $|\det(M)|$. We quantify this distribution in section 2.5. We also compute the natural density of matrices in $\mathcal{M}_{n\times n}(\mathbb{Z})$ with odd determinant, which answer the former question. Let δ_n denote this density, then we show in 2.5 that

$$\delta_n = \frac{|SL_n(\mathbb{F}_2)|}{2^{n^2}} = \prod_{i=1}^{n}\left(1 - \frac{1}{2^i}\right),$$

i.e. this density is between $\lim_{n\to\infty}\delta_n \approx 0.28$ and $\delta_2 \approx 0.38$.

2.3.6 Generating the bases

We address now the "shape" of bases $\mathbf{B}_J^{\mathsf{sk}}, \mathbf{B}_J^{\mathsf{pk}}$ and the method to generate them. Because of how decryption works, $P(\mathbf{B}_J^{\mathsf{sk}})$ must contain a large $\mathbb{R}^n$-ball, which in turn must contain the set $\mathsf{Samp}(\mathcal{P})$. Indeed, if a message $x \in \mathcal{P} = \{0,1\}^n$ is such that $\mathsf{Samp}(x) \notin P(\mathbf{B}_J^{\mathsf{sk}})$, then $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, x)) \neq x$. On the contrary, the "public" base $\mathbf{B}_J^{\mathsf{pk}}$ must be large and eccentric, imitating the shape of a very long segment in $\mathbb{R}^n$, to ensure that the encryption $\mathsf{Samp}(x) \mod \mathbf{B}_J^{\mathsf{pk}}$ is very far away (and in particular, different modulo $\mathbf{B}_I$) from $\mathsf{Samp}(x)$.

We already assumed that n is large so no lattice reduction is available (in order to generate the quite orthogonal $\mathbf{B}_J^{\mathsf{sk}}$). Instead, one can simply generate this matrix using the rotation basis of a suitable polynomial. In point of fact, choose a secret vector v^{sk} of large l_2 norm and next to the direction $\mathbf{e}_1 = (1, 0, \cdots, 0)$ (or any other axis) and set $J = \alpha((v^{\mathsf{sk}}))$, $\mathbf{B}_J^{\mathsf{sk}} = \mathrm{rot}(v^{\mathsf{sk}})$. To generate the public key, just compute the Hermite normal form of $\mathbf{B}_J^{\mathsf{sk}}$.

As a minor detail, note that if $v^{\mathsf{sk}} = (v_1, \ldots, v_n) \in \mathbb{Z}^n$, then both $s = v_1 + \cdots + v_n$ and $d = -v_1 + v_2 - v_3 - \ldots (-1)^n v_n$ divide $\det(\mathrm{rot}(v^{\mathsf{sk}}))$ (s and d are eigenvalues of $\mathrm{rot}(v^{\mathsf{sk}})$ with corresponding eigenvectors $\sum_k \mathbf{e}_k$ and $\sum_k (-1)^k \mathbf{e}_k$ respectively). Therefore, if the sampled secret vector corresponds to an even value of s or d, it can be ruled out since the determinant of $\mathrm{rot}(v^{\mathsf{sk}})$ is even.

Let us give a toy example.

Algorithm 3 Key Generation for Gentry's IL homomorphic scheme

Require: Security parameter λ, parameters n, l_{Samp}
Ensure: Key-pair $\mathbf{B}_J^{\mathsf{sk}}, \mathbf{B}_J^{\mathsf{pk}}$

function $\mathsf{Keygen}(\lambda)$
  $\mathbf{B}_J^{\mathsf{sk}} \leftarrow 0$
  while $(\det(\mathbf{B}_J^{\mathsf{sk}}) = 0 \mod 2)$ **do**
    Sample $v^{\mathsf{sk}} \leftarrow \mathbb{Z}^n$ such that v^{sk} is close to a $\mathbb{R}^n$ axis and $||v^{\mathsf{sk}}||_2 \gg l_{\mathsf{Samp}}$
    Set $\mathbf{B}_J^{\mathsf{sk}} = \mathrm{rot}(v^{\mathsf{sk}})$
  end while
  $\mathbf{B}_J^{\mathsf{pk}} = \mathrm{HNF}(\mathbf{B}_J^{\mathsf{sk}})$
  Output $(\mathbf{B}_J^{\mathsf{sk}}, \mathbf{B}_J^{\mathsf{pk}})$
end function

Example 2.3.1. Let $n = 7$ and sample $v^{\mathsf{sk}} = (18, -3, 0, 0, 1, 0, -1)$ (or the polynomial $18 - 3X + X^4 - X^6$) which is close to the direction $\mathbf{e}_1$ (notice also that $s = 15$ and $d = 21$ are odd). The key generation algorithm outputs

$$\mathbf{B}_J^{\mathsf{sk}} = \mathrm{rot}(v^{\mathsf{sk}}) = \begin{pmatrix} 18 & -1 & 0 & 1 & 0 & 0 & -3 \\ -3 & 18 & -1 & 0 & 1 & 0 & 0 \\ 0 & -3 & 18 & -1 & 0 & 1 & 0 \\ 0 & 0 & -3 & 18 & -1 & 0 & 1 \\ 1 & 0 & 0 & -3 & 18 & -1 & 0 \\ 0 & 1 & 0 & 0 & -3 & 18 & -1 \\ -1 & 0 & 1 & 0 & 0 & -3 & 18 \end{pmatrix},$$

$$\mathbf{B}_J^{\mathsf{pk}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 516180659 & 338284094 & 98416679 & 286516904 & 328581644 & 458971154 & 573665415 \end{pmatrix}.$$

Now, $\det J = 573665415$ is odd, therefore $(I, J) = 1$ (we also have the minor details $(18 - 3 + 1 - 1) | 573665415$ and $(-18 - 3 - 1 + 1) | 573665415$). Notice how the entries of the last column of $\mathbf{B}_J^{\mathsf{pk}}$ are large in comparison to the secret vector.

2.3.7 Role of the representative sampling function

To encrypt a message $\pi \in \{0,1\}^n$, one computes $\mathsf{Enc}(\pi) = \mathsf{Samp}(\pi) \mod \mathbf{B}_J^{\mathsf{pk}}$, where Samp is a randomized procedure that chooses a representative of the class $\pi + I$. In the absence of this sampling, the scheme would be deterministic. A huge comeback, since it would allow to solve equations on ciphertexts. Indeed, suppose that the encrypting equation was simply $\mathsf{Enc}(\pi) = \pi \mod \mathbf{B}_J^{\mathsf{pk}}$, then we have $\mathsf{Enc}(\mathsf{pk}, x) = x$ for every $x \in \{0,1\}^n \cap P(\mathbf{B}_J^{\mathsf{pk}})$ (for instance $x \in \{0, \mathbf{e}_n\}$). Therefore, the role of this function is to randomize encryption and separate the message from the eccentric parallelepiped $P(\mathbf{B}_J^{\mathsf{pk}})$ while keeping the parity of the

coordinates, in order that mod $\mathbf{B}_J^{\mathsf{pk}}$ reduction outputs a very far away point of $\mathbb{Z}^n$ and, above all, not equivalent to π modulo 2 (i.e. modulo $\mathbf{B}_I$).

In fact, we have $\{0,1\}^n \cap P(\mathbf{B}_J^{\mathsf{pk}}) = \{0, \mathbf{e}_n\}$, it hence suffices to take Samp as a distribution that outputs randomly a representative of $\pi + (I)$ in a ball of radius l_{Samp}. Gentry proposes $l_{\mathsf{Samp}} = n$.

2.3.8 Correctness of encryption and decryption

The size of the private key $||v^{\mathsf{sk}}||_2$ and radius of the sampling l_{Samp} must be chosen to ensure correctness of decryptions.

Proposition 2.3.3. *Let* $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Keygen}(\lambda)$. *If* $P(\mathbf{B}_J^{\mathsf{sk}}) \supset \mathsf{Samp}(\{0,1\}^n)$, *then* $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, x)) = x$ *for all* $x \in \{0,1\}^n$.

Proof: Let $x \in \{0,1\}^n$ and suppose that the representative $x + i = \mathsf{Samp}(x)$ lays in the interior of $P(\mathbf{B}_J^{\mathsf{sk}})$. To encrypt, one performs

$$x \xrightarrow{\mathsf{Samp}} x + i \xrightarrow{\text{mod } \mathbf{B}_J^{\mathsf{pk}}} x + i + j = y \in P(\mathbf{B}_J^{\mathsf{pk}}),$$

where $i \in^R I, j \in^R J$. Decryption gives

$$y \xrightarrow{\text{mod } \mathbf{B}_J^{\mathsf{sk}}} y - j' \xrightarrow{\text{mod } \mathbf{B}_I} y - j' - i',$$

where $i' \in I, j' \in J$. The reduction $y \mod \mathbf{B}_J^{\mathsf{sk}}$ outputs the only representative of y in $P(\mathbf{B}_J^{\mathsf{sk}})$, and by hypothesis $x + i$ is inside $P(\mathbf{B}_J^{\mathsf{sk}})$, thus $y - j' = x + i$ and $j = j'$. Also $y - j' \bmod \mathbf{B}_I = x + i \bmod \mathbf{B}_I = x$. □

On the contrary, if $x + i \notin P(\mathbf{B}_J^{\mathsf{sk}})$, the reduction $y \mod \mathbf{B}_J^{\mathsf{sk}}$ outputs a point $z \in \mathbb{Z}^n$ that does not match $x + i$ since $z \in P(\mathbf{B}_J^{\mathsf{sk}})$, ruining decryption.

Example 2.3.2. In the example 2.3.1 above, the sampling radius must not exceed $||v^{\mathsf{sk}}||/2 \approx 9.15$. As a matter of fact, our choice of v^{sk} ensures correctness of decryptions since $9.15 > 7 = n$, i.e. $P(\mathbf{B}_J^{\mathsf{sk}}) \supset \mathsf{Samp}(\{0,1\}^n)$ (however, as we will see in the next section, this choice does not allow homomorphic operations). Let us encrypt the message $\pi = (1,1,0,1,0,0,1)$ under the public key of example 2.3.1. We first sample a representative of $\pi + I$ contained within a ball of radius 7 centered at 0: $\mathsf{Samp}(\pi) = (1,-1,4,1,4,-2,1)$ (notice that parity is respected and the point is not close to the stable direction $\mathbf{e}_7$). We then compute the representative of this vector within $P(\mathbf{B}_J^{\mathsf{sk}})$ and output

$$\mathsf{Enc}(\mathsf{pk}, (1,1,0,1,0,0,1)) = (0,0,0,0,0,0,-107133622) = \psi.$$

To decrypt this message, first compute

$$(0,0,0,0,0,0,-107133622) \mod \mathbf{B}_J^{\mathsf{sk}} = (1,-1,4,1,4,-2,1)$$

and output this point modulo 2. We find the message $x = (1,1,0,1,0,0,1)$.

2.3.9 Correctness of homomorphic operations

As we discuss here, in order to securely evaluate circuits of high depth, the size of public keys and ciphertexts must be very large. This is a major drawback of the scheme.

Let $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Keygen}$, x, x' be two messages in $\{0,1\}^n$ and let $i, i' \in I, j, j' \in J$ such that $\mathsf{Enc}(\mathsf{pk}, x) = x + i + j, \mathsf{Enc}(\mathsf{pk}, x') = x' + i' + j'$. To verify correctness of homomorphic addition, the vector $s \in \mathbb{Z}^n$ given by $s = \mathsf{Enc}(\mathsf{pk}, x) + \mathsf{Enc}(\mathsf{pk}, x')$ must be a valid ciphertext and correctly decryptable in the sense of section 2.3.8. In clear,

$$s = (x + x') + (i + i') + (j + j'),$$

and with the same reasoning as before, the image of s under $\mathsf{Dec}(\mathsf{sk}, \cdot)$ will not output $x + x' \mod 2$ if $(x + x') + (i + i')$ lies outside $P(\mathbf{B}_J^{\mathsf{sk}})$. Given that

$$||(x + x') + (i + i')|| \leq 2l_{\mathsf{Samp}},$$

homomorphic addition is correct if $l_{\mathsf{Samp}} < ||v^{\mathsf{sk}}||/4$, where v^{sk} is an edge of $\mathbf{B}_J^{\mathsf{sk}}$. In general, to ensure correctness of k fan-in homomorphic addition

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, x_1) + \cdots + \mathsf{Enc}(\mathsf{pk}, x_k)) = x_1 + \cdots + x_k \mod 2$$

for all tuple of vectors $x_1, \cdots, x_k$ in $\{0,1\}^n$, the condition is $l_{\mathsf{Samp}} < \dfrac{||v^{\mathsf{sk}}||}{2k}$. In our example 2.3.1, not even an homomorphic addition can be supported since $n = 7 \not< \frac{1}{4}||v^{\mathsf{sk}}|| \approx 4.58$. More precisely, for the secret key $v^{\mathsf{sk}} = (18, -3, 0, 0, 1, 0, -1)$, one homomorphic addition can be carried out if Samp outputs vectors within a ball of radius 4.

Of course, to support homomorphic multiplication of two ciphertexts, the inequality between $||v^{\mathsf{sk}}||$ and l_{Samp} is far more restrictive. Consider the same scenario as before and $p = \mathsf{Enc}(\mathsf{pk}, x) \times \mathsf{Enc}(\mathsf{pk}, x')$, i.e. ,

$$p = (x + i) \times (x' + i') + j(x' + i') + j'(x + i) \in \mathbb{Z}^n.$$

For correct decryption, the point $(x + i) \times (x' + i')$ must lie within the secret parallelepiped. Applying the product overhead inequality $||\mathbf{a} \times \mathbf{b}|| \leq \sqrt{n}||\mathbf{a}|| \times ||\mathbf{b}||$ for all $\mathbf{a}, \mathbf{b}$ in $\mathbb{Z}^n$, this is verified if $2\sqrt{n}(l_{\mathsf{Samp}})^2 < ||v^{\mathsf{sk}}||$. To perform a k-fan-in multiplication gate, one needs

$$2n^{\frac{k-1}{2}}(l_{\mathsf{Samp}})^k < ||v^{\mathsf{sk}}||.$$

In our 7-th dimensional example with $l_{\mathsf{Samp}} = n$, this is $||v^{\mathsf{sk}}|| > 260$, meaning that to support a single multiplication, the entries in the last row of a public key base are 56-bit integers. For $n = 334$ (preventing current lattice reductions in the average case), supporting a single multiplication needs a public base of vectors with 7335-bit integers.

2.3.10 Maximum depth of allowed circuits

With the previous reasoning we quantify the maximum depth for correct evaluation. As we will see, an increase in the maximum depth needs a drastic growth of key sizes.

Theorem 2.3.4. *Let v^{sk} be a secret key vector (i.e. $\mathbf{B}_J^{\mathsf{sk}} = \mathrm{rot}(v^{\mathsf{sk}})$). The maximum depth that can be correctly homomorphically evaluated is*

$$d_{\max} = \log_2 \log_2(||v^{\mathsf{sk}}||) - \log_2 \log_2(\sqrt{n} l_{\mathsf{Samp}}),$$

with at most $\sqrt{n}$ addition fan-in and 2 product fan-in.

Proof: Let $r_i \geq 1$ be a superior bound to the l_2 norm of outputs on level i of the circuit C. Suppose that C is of depth $d_{\max}$ and set $r_0 = l_{\mathsf{Samp}}$. At level $i-1$ for $i \geq 1$,

- $||v_1 + \cdots + v_{\lfloor\sqrt{n}\rfloor}|| \leq \sqrt{n} \cdot r_i$,
- $||u \times v|| \leq \sqrt{n}\,||u|| \cdot ||v|| \leq \sqrt{n} \cdot r_i^2$.

Therefore the output of gates at level i can be bounded by $\sqrt{n} \cdot r_i^2$. Therefore $r_{i-1} \leq \sqrt{n} \cdot r_i^2$ and

$$r_0 \leq \sqrt{n} r_1^2 \leq \sqrt{n}^{1+2} r_2^2 \leq \cdots \leq \sqrt{n}^{1+2+\cdots+2^{k-1}} r_k^{2^k} = \sqrt{n}^{2^k-1} r_k^{2^k} \leq (\sqrt{n} r_k)^{2^k}.$$

For the evaluation to be correct al level $d_{\max}$ we must have $r_{d_{\max}} \leq ||v^{\mathsf{sk}}||_2$ i.e.

$$l_{\mathsf{Samp}} \leq (\sqrt{n}||v^{\mathsf{sk}}||_2)^{2^{d_{\max}}}. \quad \blacksquare$$

Said otherwise, the secret key size $||v^{\mathsf{sk}}||_2$ is super-exponential in the maximum depth.

2.4 Security

2.4.1 The Ideal Coset Problem

Gentry's scheme semantic security is based on the Ideal Coset Problem (ICP). It is the problem of distinguishing between a valid encryption and a random vector in the eccentric parallelepiped $P(\mathbf{B}_J^{\mathsf{pk}})$.

Definition 2.4.1 (ICP)**.** *Let $R = \mathbb{Z}[x]/(f(x))$ for a monic polynomial f of degree n, and let I, J be relatively prime ideal lattices. Let* Samp *be an algorithm that, on input x, samples a representative of the class $x+I$. Let $b \xleftarrow{R} \{0,1\}$ and $(\mathbf{B}_J^{\mathsf{sk}}, \mathbf{B}_J^{\mathsf{pk}}) \leftarrow \mathsf{Keygen}(\lambda)$.*

- *If $b = 0$, let $x \xleftarrow{R}$ and set $\mathbf{v}' \leftarrow \mathsf{Samp}(x) \mod \mathbf{B}_J^{\mathsf{pk}}$.*

– *If* $b = 1$, *let* $\mathbf{v} \xleftarrow{R} R \mod \mathbf{B}_J^{\mathsf{pk}}$.

The Ideal Coset Problem is to guess b *with the knowledge of* $(\mathbf{v}, \mathbf{B}_J^{\mathsf{pk}})$.

If the image of Samp samples from a ball of too small radius, solving ICP is easy. Gentry reduces the semantic security of the homomorphic scheme to the Ideal Coset Problem.

Theorem 2.4.1. *If* $\mathcal{A}$ *is an algorithm that breaks the semantic security of the ideal lattice homomorphic scheme with advantage* ε*, then there is an algorithm* $\mathcal{B}$ *that solves ICP with advantage* $\varepsilon/2$.

Proof: Suppose an attacker Bob wants to solve ICP. He uses the algorithm $\mathcal{A}$, which asks B to send the encryption of one of two messages π_0 or π_1. Bob defines a bit β and sends $\mathsf{Enc}(\pi_\beta)$ to algorithm $\mathcal{A}$. The algorithm outputs a guess β' to the challenge, which will be correct with an advantage ε. Bob answers the ICP challenge with the bit $\beta \oplus \beta'$.

Indeed, if $b = 0$, Bob's simulation is perfect: the message given to Bob from the ICP instance is of the form $\psi \leftarrow \pi_\beta + \mathbf{v} \times s \mod \mathbf{B}_J^{\mathsf{pk}}$ and therefore of the form $\mathsf{Enc}(\pi_\beta)$. The algorithm $\mathcal{A}$ will output the correct $\beta' = \beta$ with an advantage of ε, thus giving an advantage of ε pour B. If $b = 1$, the advantage of Bob is 0 since ψ is a uniformly random element of the parallelepiped $P(\mathbf{B}_J^{\mathsf{pk}})$. Hence Bob total advantage is $\varepsilon/2$. □

The ratio between the size of $\lambda_1(J)$ (the norm of the smallest vector of J) and the sampling radius l_{Samp} is crucial to conserve the difficulty of this problem. If fact, if $\lambda_1(J)/l_{\mathsf{Samp}} \geq 2^n$, one can find the nearest lattice element $\mathbf{j} \in J$ to the challenge $\mathbf{v}$ (using variants of LLL and Babai's nearest plane algorithm). If $\mathrm{dist}(\mathbf{j}, \mathbf{t}) > l_{\mathsf{Samp}}$, then $\mathbf{v}$ was generated uniformly, and if $\mathrm{dist}(\mathbf{j}, \mathbf{t}) < l_{\mathsf{Samp}}$ then more than likely $\mathbf{v}$ is an actual encryption.

There are no known attacks for $l_{\mathsf{Samp}} = \mathrm{poly}(n), \lambda_1(J) = 2^{O(\sqrt{n})}$, and controlling the value $\lambda_1(J)$ is possible via a spectral analysis of rotation matrices (see [OYKU10]).

2.5 Natural density of some classes of lattices

We are motivated by the choice of I, J, which are meant to be relatively prime lattices. Practical questions arise, for instance, how "dense" is the set of relatively prime lattices in the set of pairs of random lattices? As stated before, each lattice of $\mathbb{Z}^n$ is uniquely defined by a matrix in Hermite Normal Form. This allows to imagine a random lattice sampler:

Definition 2.5.1. *Let* $\mathcal{H}_n(t)$ *be the distribution that samples* n *positive integers* $d_1, \ldots, d_n$ *uniformly in* $\{1, \ldots, t\}$ *and* $(n^2 - n)/2$ *integers* a_{ij} *for* $i = 1, \ldots, n$ *and*

$j = i+1, \ldots, n$ uniformly in $\{-t, \ldots, t\}$ for a large integer B, and then outputs the Hermite Normal Form of the following matrix

$$\begin{pmatrix} d_1 & a_{1,2} & a_{1,3} & \cdots & a_{1,n-1} & a_{1,n} \\ 0 & d_2 & a_{2,3} & \cdots & a_{2,n-1} & a_{2,n} \\ 0 & 0 & d_3 & \cdots & a_{3,n-1} & a_{3,n} \\ \vdots & & \ddots & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & d_{n-1} & a_{n-1,n} \\ 0 & 0 & 0 & \cdots & 0 & d_n \end{pmatrix}.$$

Let $||\cdot||_{\max}$ be the component-wise max norm defined by $||M||_{\max} := \max_{i,j} |M_{ij}|$. Let also $\mathcal{B}_t$ be the ball of matrices centered at 0 with radius t with respect to the max norm, then image of this sampler is the set of all random matrices in HNF lying inside the ball $\mathcal{B}_t$, or equivalently, all lattices with $\det(L) \leq t^n$. In this section we aim to quantify some probability distributions on random lattices, i.e. random Hermite Normal Forms. These results aim to give intuition on the more complex problem of studying the distribution of relatively prime random ideal lattices. Let us first define the natural density of a set of matrices.

Definition 2.5.2 (Matrix natural density)**.** *Let $\mathcal{S} \subset \mathcal{M}_n(\mathbb{Z})$ be a subset of integer matrices. Let $\sigma_{\max} : \mathbb{N} \to \mathbb{N}$ be the counting function $\sigma_{\max}(t) := \#(S \cap \mathcal{B}_t)$. The natural density of $\mathcal{S}$ is*

$$\delta(\mathcal{S}) := \lim_{t\to\infty} \frac{\sigma_{\max}(t)}{t^{n^2}},$$

when the limit exists.

2.5.1 Natural density of diagonal elements of HNF's

Here, we show that *uniformly random HNF matrix* in the sense of the sampler $\mathcal{H}_n$ follows a very different structure as the *HNF of a uniformly random matrix*. In [Maz11], authors prove the following:

Theorem 2.5.1. *(i) The natural density of matrices having at least one zero in the diagonal of their HNF is zero.*

(ii) Let $(d_1, \cdots, d_{n-1}) \in (\mathbb{N}^*)^n$. The natural density of matrices having a HNF of the form

$$\mathrm{HNF}(A) = \begin{pmatrix} d_1 & * & * & \cdots & * & * \\ 0 & d_2 & * & \cdots & * & * \\ 0 & 0 & d_3 & \cdots & * & * \\ \vdots & & \ddots & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & d_{n-1} & * \\ 0 & 0 & 0 & \cdots & 0 & \beta \end{pmatrix}, \text{ where } \beta = \frac{|\det(A)|}{\prod_{i=1}^{n-1} d_i},$$

is equal to

$$\delta(d_1, \cdots, d_{n-1}) = (\zeta(n) \cdot \zeta(n-1) \cdot \ ... \ \cdot \zeta(2) \cdot d_1^n \cdot d_2^{n-1} \cdot \ ... \ \cdot d_{n-2}^2 \cdot d_{n-1})^{-1}.$$

The proof of this theorem relies on two fundamental observations: (a) The natural density of k-tuples $(r_1, \ldots, r_k)$ with $\text{pgcd}(r_1, \ldots, r_k) = d$ is $(\zeta(k) \cdot d^k)^{-1}$ where ζ is the Riemann Zeta function and (b) if E is a set of elements of $\mathbb{Z}^n$, then $UE := \{Ue, e \in E\}$ has the same natural density that E, where U is any unimodular matrix. The proof is interesting, see [Maz11].

In this fashion, if by *uniformly random lattice* we mean the lattice generated by the columns of a uniformly random matrix, then the natural density of lattices L whose HNF diagonal elements are $d_1 = d_2 = \cdots = d_{n-1} = 1$ and $d_n = \det(L)$ is $(\prod_{k=2}^{\infty} \zeta(k))^{-1} = 0.436....$ Hence, it is likely that public bases of Gentry scheme have "1" entries almost everywhere in the diagonal, and large divisors of the determinant near the tail. In our implementation, we observed precisely this behavior.

2.5.2 Natural density of coprime lattices

Now, let I be a fixed lattice generated by the columns of A, and let ϕ denote Euler's totient function. We prove the following:

Proposition 2.5.2. *Let $B \leftarrow \mathcal{H}_n$. The probability that the lattice J generated by B is coprime to I verifies*

$$\frac{1}{\det I} \leq \mathbb{P}((I, J) = Id) \leq \frac{\varphi(\det I)}{\det I},$$

with equality at left if $\det I$ is a power of 2 and equality at right if the diagonal elements of HNF(I) are pairwise coprime.

Proof: From proposition 2.3.2 we know that $(I, J) = 1$ if and only if $(A_{kk}, B_{kk}) = 1$ for all $0 \leq k \leq n$. Hence

$$\begin{aligned}
\mathbb{P}((I, J) = \text{Id}) &= \mathbb{P}(\forall\, 0 \leq k \leq n : (A_{kk}, B_{kk}) = 1) \\
&= \prod_{k=1}^{n} \mathbb{P}((A_{kk}, B_{kk}) = 1) = \prod_{k=1}^{n} \frac{\varphi(A_{kk})}{A_{kk}} \\
&= \frac{1}{\det(I)} \prod_{k=1}^{n} \varphi(A_{kk}).
\end{aligned}$$

where $\varphi(x)$ is Euler's totient function (the amount of integers relatively prime to and smaller than x). Note that $\varphi(ab) \geq \varphi(a)\varphi(b)$ (with equality if $(a, b) = 1$), and the result follows. □

Remark. A plaintext space $\mathbf{B}_I$ for Gentry's scheme which gives a good probability involves a choice of I such that $\det(I)$ is a product of n (not necessarily distinct) prime

numbers $p_1, \ldots, p_n$, each one greater than an integer αn, and such that $\mathrm{HNF}(I) = \mathbf{H}$ verifies $\mathbf{H}_{ii} = p_i$. The probability then gives

$$\mathbb{P}((I, J) = \mathrm{Id}) = \prod_{i=1}^{n} \left(1 - \frac{1}{p_i}\right) > \left(1 - \frac{1}{\alpha n}\right)^n > \exp(-1/\alpha).$$

For instance, if $\alpha = 1$, then the probability of sampling an ideal lattice J relatively prime to I is greater than $1/e \approx 36.79\%$.

We consider now the natural density of two coprime HNF's:

Proposition 2.5.3. *Let $H \leftarrow \mathcal{H}_n, H' \leftarrow \mathcal{H}_n$. The natural density of coprime matrices H and H' is*

$$\delta((H, H') = Id) = \left(\frac{6}{\pi^2}\right)^n.$$

Proof: If follows directly from the fact that the natural density of relatively prime integers is $6/\pi^2$. □

2.5.3 Natural density of odd ideal lattices

Let us say that an ideal lattice is *odd* if it is relatively prime to the ideal lattice (2). Therefore, L is odd if all diagonal entries of $\mathrm{HNF}(L)$ are odd, or equivalently, if $\det L$ is odd. If lattices are sampled via a uniformly random matrix, it suffices to compute the natural density of matrices with odd determinant.

Lemma 2.5.4. *The natural density of matrices in $\mathcal{M}_{n\times n}(\mathbb{Z})$ with odd determinant is*

$$\delta_n = \frac{|SL_n(\mathbb{F}_2)|}{2^{n^2}} = \prod_{i=1}^{n} \left(1 - \frac{1}{2^i}\right).$$

Proof: We wish to compute

$$\delta_n(t) = \frac{1}{t^{n^2}} \#\{M \in \mathcal{M}_{n\times n}(\mathbb{Z}), |m_{ij}| \leq t \, \forall i, j, \det(M) \equiv 1 \bmod 2\}$$

and $\delta_n := \lim_{t\to\infty} \delta_n(t)$. We first perform a reduction modulo 2 and work in $\mathbb{F}_2$, this eliminates the dependency on the radius t:

$$\delta_n = \frac{1}{2^{n^2}} \#\{M \in \mathcal{M}_{n\times n}(\mathbb{F}_2), \det(M) = 1 (\bmod\ 2)\}.$$

We hence see that the natural density δ_n is exactly the proportion of invertible elements of $\mathcal{M}_{n\times n}(\mathbb{F}_2)$. One can actually count this elements: let $(x_1, \cdots, x_n), x_i \in \{0,1\}^n$ be the columns of a matrix in $\mathcal{M}_{n\times n}(\mathbb{F}_2)$. Then if this matrix is invertible, the family $(x_1, \cdots, x_n)$ is linearly independent. Thus, for the first column there are $2^n - 1$ choices since it can be zero. The second column must not verify an equation of the form $\mathbf{e}_1 x_1 = x_2$ by independence, hence there are $2^n - 2$ choices.

The third column must not verify an equation of the form $\mathbf{e}_1 x_1 + \mathbf{e}_2 x_2 = x_3$, and thus there are $2^n - 4$ choices. Continuing the argument, the column x_k must be linearly independent of the precedent $k-1$ columns, i.e. a choice between $2^n - 2^{k-1}$ elements. This way,

$$\#\{M \in \mathcal{M}_{n\times n}(\mathbb{F}_2),\, \det(M) = 1\} = \prod_{i=0}^{n-1}(2^n - 2^i),$$

and the result follows. □

Note that, for all $n \geq 2$, δ_n is a strictly decreasing sequence, from where we conclude

$$0.28 \approx \lim_{k\to\infty} \delta_k < \delta_n < \delta_2 \approx 0.38.$$

2.5.4 Remark on natural density of circulant matrices with odd determinant

Let L be an ideal lattice and let R_L its rotation basis. The set of bases of L is exactly

$$\mathcal{B}(L) = R_L \cdot GL_n(\mathbb{Z}),$$

i.e. , the set of all matrices of the form $R_L U$ where U is a unimodular matrix. If the set of circulant matrices is not uniformly distributed in $\mathcal{M}_{n\times n}(\mathbb{Z})$, the latter lemma may not be relevant: the parity of the determinant may have a very different distribution when constrained to cyclic matrices.

In a more direct approach, the determinant of circulant matrices can be written in terms of the generating polynomial. Indeed, if $M = \mathrm{rot}(x_0, x_1, \cdots, x_{n-1})$, we have $M = x_0\,\mathrm{Id} + x_1 T + x_2 T^2 + x_{n-1}T^{n-1}$ where

$$T = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & & 0 & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}.$$

The eigenvalues of T are n-th roots of unity $\mu_k = e^{(2\pi i/n)k}, k = 0, \ldots, n-1$, and therefore the eigenvalues of M are

$$\lambda_k = x_0 + x_1\mu_k + x_2\mu_k^2 + \cdots + x_{n-1}\mu_k^{n-1}.$$

As the determinant is the product of eigenvalues, parity may be established with a deeper analysis of vanishing sums of roots of unity. In other words, the proportion

of circulant integer matrices with odd determinant amongst all circulant integer matrices is given by

$$\frac{1}{2^n} \times \#\{(x_0, \ldots, x_{n-1}) \in \{0,1\}^n : \prod_{k=0}^{n-1} \sum_{i=0}^{n-1} x_i \mu_k^i = 0 \mod 2\}$$

For more on vanishing sums of roots of unity, the reader may refer to [CJ76, Len78, LL00, Ste08, Eve99]. Finally, we state that if the set of circulant matrices up to multiplication by a unimodular matrix is a topological cover of the set of all matrices, then the answer relates to lemma 2.5.4. Unfortunately, this exits the scope of this Dissertation.

2.6 Proof-of-concept implementation

We coded a working instance of Gentry's ideal lattice homomorphic scheme in `C` with the `GMP` large numbers library. In the first place, we needed to build linear algebra and lattice layers. Note that, in order to compute basis reductions, we needed to calculate $\mathbb{Q}$-inverses of matrices, therefore with the type `mpf_t` of `GMP`. For completeness, let us enumerate some of the functions we implemented.

2.6.1 Basic functions

A large integer number (having up to millions of bits) is defined in `GMP` by the prefix `mpz_t`, and a floating point number (with a user-predefined precision) by the prefix `mpf_t`. Some of the functions we implemented follow.

`FFT(mpf_t &v[N])` ($v \leftarrow \mathrm{FFT}(v)$)
`FastMatrixVector(mpz_t A[N][N],mpz_t x[N],mpz_t b[N])` ($b \leftarrow Ax$)
`MultiplyCtxt(mpz_t x[N],mpz_t y[N],mpz_t prod[N])` (prod $\leftarrow x \times y \mod \mathbf{B}_J^{\mathsf{pk}}$)
`MultiplyZn(mpz_t x[N],mpz_t y[N],mpz_t prod[N])` (prod $\leftarrow x \times y \in \mathbb{Z}^n$)
`RandomBinaryVector(mpz_t x[N])` ($x \xleftarrow{R} \{0,1\}^n$)
`RandomVector(mpz_t x[N],int lambda)` ($x \xleftarrow{R} \mathbb{Z}^n$ bounded by λ)
`RintVector(mpz_t X[N],mpf_t x[N])` (nearest integer vector)
`Det(mpz_t A[N][N],mpz_t d)` ($d \leftarrow \det(A)$)
`HNF(mpz_t &A[N][N])` ($A \leftarrow \mathrm{HNF}(A)$)
`Inverse(mpz_t A[N][N],mpf_t Ainv[N][N])` (`Ainv` $\leftarrow A^{-1}$)
`ModBasis(mpz_t v,mpz_t A[N][N],mpf_t Ainv[N][N])` ($v \leftarrow v - A\lfloor A^{-1}v \rceil$).
`MatrixProduct(mpz_t A[N][N],mpz_t B[N][N],mpz_t P[N][N])` ($P \leftarrow AB$)
`PALU(mpz_t P[N][N],mpz_t A[N][N],mpf_t L[N][N],mpf_t U[N][N])` ($PA = LU$ decomposition)
`RandomUnimodular(mpz_t A[N][N])` ($A \in^R GL(n, \mathbb{Z})$)
`RotationMatrix(mpz_t v,mpz_t A[N][N])` ($A \leftarrow \mathrm{rot}(v)$)

`Solve(mpz_t A[N][N],mpf_t x[N],mpz_t b[N])` (Solve $Ax = b$)
`Transpose(mpz_t A[N][N])` ($A \leftarrow {}^tA$)

At key-generation, one must generate a random vector v close enough to $\mathbb{Z}\mathbf{e}_1$ and define the secret and public bases as the rotation matrix of v and its Hermite normal form. Also, to test is two lattices are prime, the determinant can be computed. Finally, to test a variant of the key-generation procedure where one randomizes the secret key (instead of computing the HNF) we wrote a function that generates unimodular matrices. See algorithms 4 and 5.

This algorithm runs in cubic complexity, which is enough for our implementation since HNF's are only computed in key-generation time. We mention that better algorithms are reviewed in [PS10].

Algorithm 4 Unimodular Matrix Generator

procedure UnimodularMatrixGen
Require: A bound $B \in \mathbb{R}^+$
Ensure: A unimodular matrix U such that $\max_{i,j} |u_{i,j}| \geq B$
  for $0 \leq i \leq N-1$ and $0 \leq j \leq N-1$ **do**
    if $(i = j)$ **then** $u_{ii} \xleftarrow{R} \{-1, 1\}$
    else if $(i > j)$ **then** $u_{ij} \in^R [\![-\lfloor \sqrt[3]{B} \rfloor, \lfloor \sqrt[3]{B} \rfloor]\!]$
    else $u_{ij} = 0$
    end if
  end for
  while $(\max_{i,j} |u_{i,j}| < B)$ **do**
    $i \in^R [\![0, N-1]\!],\ i' \in^R [\![0, N-1]\!] \setminus \{i\}$
    $\lambda \in^R [\![-\lfloor \sqrt[3]{B} \rfloor, \lfloor \sqrt[3]{B} \rfloor]\!]$
    $\mathrm{Row}(i) \leftarrow \mathrm{Row}(i) - \lambda \times \mathrm{Row}(i')$
    $j \in^R [\![0, N-1]\!]$
    $j' \in^R [\![0, N-1]\!] \setminus \{j\}$
    $\mu \in^R [\![-\lfloor \sqrt[3]{B} \rfloor, \lfloor \sqrt[3]{B} \rfloor]\!]$
    $\mathrm{Column}(j) \leftarrow \mathrm{Column}(j') - \mu \times \mathrm{Column}(j)$
    $\mathrm{SwapRows}(i, i')$
    $\mathrm{SwapColumns}(j, j')$
  end while
end procedure

2.6.2 Scheme algorithms

With these functions, the scheme can be described neatly:

```
   KeyGen(int lambda, mpz_t BJsk[N][N],mpz_t BJpk[N][N])
{
  mpz_t v[N];
  InitVector(v);
```

```
  RandomVector(v,lambda);
  while("v is far from every axis")
  {
    RandomVector(v,lambda);
  }
  RotationMatrix(v,BJsk);
  RotationMatrix(v,BJpk);
  HNF(BJpk);
}
   Encrypt(mpz_t pi[N], mpz_t Bp[N][N], mpf_t Bpinv[N][N])
{
  Samp(pi,Bp);
  ModBasis(pi,Bp,Bpinv);
}

   Decrypt(mpz_t psi[N], mpz_t Bs[N][N], mpf_t Bsinv[N][N])
{
  ModBasis(psi,Bs,Bsinv);
  Mod2(psi);
}
```

Algorithm 5 Hermite Normal Form

Require: M is a non-singular $N \times N$ matrix

function HNF(M)
  for k from 0 to $N-1$ **do**
    while $\exists i > k, m_{ik} \neq 0$ **do**
      $j' \leftarrow$ such that $m_{j'k} = \min_{j \geq k}(m_{jk})$
      Swap rows j' and k
      for l from $\max(k-1, 0)$ to $N-1$ **do**
        Row(l) $\leftarrow$ Row(l) $- \lambda$Row(k) where $\lambda = \lfloor m_{lk}/m_{kk} \rfloor$
      end for
    end while
    if $m_{kk} < 0$ **then** Row(k) $\leftarrow (-1) \times$Row(k)
    end if
  end for ▷ M is now upper triangular
  for k from $N-1$ to 0 (decreasing) **do**
    for l from $\max(k-1, 0)$ to 0 (decreasing) **do**
      Row(l) $\leftarrow$ Row(l) $- \lambda$Row(k) where $\lambda = \lfloor m_{lk}/m_{kk} \rfloor$
    end for
  end for
  Output M
end function

2.6.3 Performances

Because of the size of public keys, our implementation operates up to $n = 200$, which is clearly too small to avoid lattice reduction attacks (i.e. recover $\mathbf{B}_J^{\mathsf{sk}}$ from $\mathbf{B}_J^{\mathsf{pk}}$). For a choice of parameters allowing depth-2 circuits, we have the performances in seconds given in figure 2.6.3.

Function	$n = 32$	$n = 64$	$n = 128$	$n = 200$
`HNF`[s]	<0.1	<0.1	8	993
`Inverse`[s]	<0.02	2	42	2507
`KeyGen`[s]	<0.02	2	50	3516

Table 2.1: Timings of Gentry's key-generation procedure in toy settings

Encryption, addition, multiplication, and decryption took < 0.1 seconds in this setting, but we point out that for larger values of n this performances must be taken into account. For instance, authors in [GH11b] report 3-minute encryptions and key-generation for large dimensions in the somewhat homomorphic scheme. We highlight also that when converted into FHE, key-generation and re-crypt procedures took 2.2 hours and 31 minutes, respectively.

Multiplication of ciphertexts consists exactly in one polynomial product (where inputs are very sparse) and one basis reduction. The efficiency bottleneck manifests for larger values of n and other choices of the polynomial ring (where fresh ciphertexts are not sparse). Notice also that key generation essentially computes $(\mathbf{B}_J^{\mathsf{sk}})^{-1}$ and $\mathrm{HNF}(\mathbf{B}_J^{\mathsf{sk}})$ (the inversion of a matrix in Hermite normal form being easy). Above $n = 256$, the size of matrices and, over all, the floating point precision required, makes our key generation algorithm no longer capable of outputting keys that correctly evaluate homomorphic circuits. A series of *tweaks* to improve efficiency must be applied. However we only intended to provide a proof-of-concept implementation, since other homomorphic schemes are far more efficient and deserve better analysis. See Chapter 5 for our implementation of the Brakerski-Gentry-Vaikunthanatan scheme, for instance.

We also tested circuit evaluation with artificial keys, for the sake of completeness. To this end we considered the following variant of the scheme (Gentry's first *tweak*), where no matrix inversion is needed. Note by v^{sk} the first column of $\mathbf{B}_J^{\mathsf{sk}}$. As $\mathrm{rot}(v^{\mathsf{sk}}) = \mathbf{B}_J^{\mathsf{sk}}$, only v^{sk} needs to be stocked in memory. It is easy to see that for every $a, b \in \mathbb{Z}^n$, $a \times b = \mathrm{rot}(a)b = \mathrm{rot}(b)a$. Now suppose that $w \in \mathbb{Q}^n$ is such that $w \times v^{\mathsf{sk}} = 1 \in R$. For $0 \leq k \leq n-1$, we have $w \times (X^k v^{\mathsf{sk}}) = X^k$, and therefore $w \times \mathrm{rot}(v^{\mathsf{sk}}) = \mathrm{Id}$, hence $\mathrm{rot}(w) = (\mathbf{B}_J^{\mathsf{sk}})^{-1}$. With this notations, the decryption equation becomes

$$\mathsf{Dec}(\psi) = (\psi - v^{\mathsf{sk}} \times \lfloor w \times \psi \rceil) \mod 2.$$

When $R = \mathbb{Z}[x]/(f(x))$, in order to compute $w = (v^{\mathsf{sk}})^{-1}$, just apply the Extended Euclidean algorithm on inputs v^{sk} and $f(x)$. This variant reduces considerably the size of keys, however, the floating point precision is still of order about $\log \det J$ to ensure correct decryptions. With this tweak, we evaluated circuits to give in table 2.2 the required size of $||v^{\mathsf{sk}}||$ to perform 1.000.000 correct successive evaluations when $l_{\mathsf{Samp}} = n$.

$l_{\mathsf{Samp}} = n$	Depth 2	Depth 3	Depth 5
$n = 10$ (toy)	2^{34}	2^{46}	2^{70}
$n = 32$	2^{40}	2^{61}	2^{103}
$n = 64$	2^{56}	2^{74}	2^{158}
$n = 128$ (very small)	2^{64}	2^{88}	2^{257}

Table 2.2: Reported secret key Euclidean size in Gentry's *tweaked* scheme.

In an application of depth 10 with $n = 1024$, we estimate that the secret key in our implementation is of Euclidean norm $O(2^{1.1 \cdot 10^8})$, which corresponds to about 14MB of data. This agrees to [GH11b] where authors report a public-key size of 70MB for $n = 2048$. In contrast, they implement the full scheme (bootstrapping included) and they use highly optimized algorithms to compute resultants and arithmetic in $\mathbb{Z}[x]/(f(x))$.

2.7 Conclusion

In this Chapter, we described in detail the first FHE scheme ever constructed, as an homage to Craig Gentry's dissertation. We verified with a `C+gmp` implementation its huge handicap in efficiency, and agreed to the fact that it is not remotely competitive with other FHE schemes. Finally, we gave a digression about the probabilistic structure of some classes of random lattices.

Chapter 3

A Key-Recovery Attack Against Gentry's Ideal Lattice Scheme via ad-HPP

The contents of this Chapter include an article submitted to an international conference. This is joint work with Louis Goubin and Cyril Hougounenq.

Contents

3.1 Introduction 58
3.2 Preliminaries 60
3.2.1 Notation 60
3.2.2 Lattices 60
3.2.3 Polynomial ideal lattices 61
3.2.4 Key-Recovery Attacks against SHE schemes 62
3.3 Gentry's ideal lattice basic scheme 62
3.4 The attack 64
3.4.1 Reduction to ad-HPP 64
3.4.2 Feeding Nguyen-Regev's algorithm and Random Scattering 65
3.4.3 Falling towards vertices 66
3.4.4 Proofs of algorithms 68
3.5 Length of binary searches in the neighbor algorithm . . 71
3.6 Conclusion 72

3.1 Introduction

A parallelepiped of $\mathbb{R}^n$, or parallelotope, is the polytope obtained when translating a point of $\mathbb{R}^n$ successively in n independent directions. It is the set delimited by $2n$ hyperplanes of dimension $n-1$ such that there are exactly n different pairs of parallel hyperplanes. Consider the following.

Adaptive Hidden Parallelepiped Problem (ad-HPP): Let P be a secret parallelepiped in $\mathbb{R}^n$ centered at 0 with vertices in $\mathbb{Z}^n$. Let χ be the indicator function of P. Given adaptive access to χ and a superior bound D on the diameter of P, recover P.

Gentry's celebrated first fully homomorphic scheme relies on ideal lattices of $\mathbb{R}^n$. In particular, the keys $(\mathsf{sk}, \mathsf{pk})$ are two parallelepipeds of the same volume, such that sk is quite orthogonal and small, and pk is eccentric and large (it may look like a long segment of $\mathbb{R}^n$). The latter one is public and it serves to encrypt messages, implying that any attacker may use it to try and recover information about sk or the underlying message. In this article we show that an attacker having access to adaptive decryption queries may construct a function that matches the indicator function of sk in all points of $\mathbb{R}^n$ with overwhelming probability. With sufficient amount of queries they can use any algorithm solving ad-HPP to find the vertices of sk, or approximate its shape with some precision, which can be enough to recover the parallelepiped.

Fully Homomorphic Encryption. In a groundbreaking sequence of articles [Gen09b,Gen09c], Gentry constructed the first fully homomorphic encryption (FHE) scheme. An FHE scheme is a regular public key encryption scheme that allows public processing of ciphertexts, without access to decryption secrets. More precisely, it allows to publicly transform a set of ciphertexts into a new one that encrypts any desired polynomial function on the underlying plaintexts. FHE finds several modern applications, and exciting research has been performed since 2009. In order to fulfill the needs of these applications and to compete with contemporary encryption schemes, optimizations, modifications and new proposals have been appearing in order to improve efficiency and security. Today, there are five families of homomorphic schemes, based on ideal lattices, integers, NTRUEncrypt, (Ring) Learning With Errors and Approximate Eigenvalues, respectively. Some constructions are better in security, others in efficiency, and others exhibit fancy properties (let us highlight the Multikey FHE scheme of Lopez-Alt et al. [LATV12], the provable circular-secure of Brakerski et al. [BV11b], and the attribute-based of Gentry et al. [GSW13]). Recent proposals greatly outperform the original ideal lattices scheme as they are better in efficiency, more suited for applications and less complex in structure. We review the first scheme in section 3.3, in order to present our key-recovery attacks in section 3.4.

FHE and chosen ciphertext attacks. It is known that FHE schemes cannot resist CC-2 attacks, i.e. when an adversary aims to decrypt a challenge ciphertext with the help of an oracle that decrypts ciphertexts under the corresponding key. The reason for this is that despite the attacker is not allowed to submit the challenge, she can transform it into a different ciphertext using identity homomorphic operations (such as $m \mapsto m \oplus 0$) and submit the result. Of course, this does not mean that the secret of the oracle can be retrieved by the adversary in a CC attack. Precisely, the aim of this paper is to show that the original scheme proposed by Gentry allows an attacker to recover the secret key in polynomial time.

Our Contributions. We first show how to use an ad-HPP solver to recover a secret key of Gentry's scheme. We propose two algorithms that solve ad-HPP. Our first proposal consists in guessing sufficiently many points near the boundary and then approximating an edge of the parallelepiped (which recovers the full secret by rotations). It uses $O(n^2 \log D)$ adaptive calls to χ. Our second and more promising proposal imitates the effect of gravity, bouncing repeatedly against ∂P and converging to a vertex. To this end, after n bounces the attacker computes the $(n-1)$-surface generated by the n bouncing points, and repeats the process starting from the center of the hyperrectangle formed by these points. We prove that $\lceil \log(D) \rceil$ repetitions are enough to output a vertex, and that two neighbor vertices can be found adaptively by submitting $O(n \log^3(D))$ points to χ. With this, retrieving a secret-key of Gentry's scheme can be done with $O(n \log^3 n)$ adaptive calls to the oracle. We give a complete analysis and proof of our attack.

Related Work. The Hidden Parallelepiped Problem has already been used in cryptanalysis in its non-adaptive form, i.e. , guess the parallelepiped given a list of random uniformly distributed points in its interior, without access to the indicator function or the integer vertices hypothesis. In [NR06], Nguyen and Regev introduced a nice algorithm solving the Hidden Parallelepiped Problem based on an optimization problem on a n-hypercube after proper transformations. We refer to it as the Nguyen-Regev algorithm. They applied it to cryptanalyze NTRU and GGH signature schemes, reporting full recovery of NTRU secret keys using only 400 message-signature pairs for advised parameters. In the adaptive version of HPP we introduce, the attacker may generate a list of interior points using the indicator function and feed the NR algorithm to find the parallelepiped. However, a large number of points is required to proceed. Authors overcome this using known symmetries on NTRU lattices, which allow to generate a number of signatures from a given message-signature list. Unfortunately, these symmetries are not present in Gentry's ideal lattices. Some other key-recovery attacks against homomorphic encryption schemes have been published, we mention [DGM15] and [LMSV12]. The latter includes an attack to a variant of Gentry's scheme that turns out to be isomorphic to the $I = (2)$ case. In contrast, to the best of our knowledge, our proposal

is the first direct key-recovery attack to the scheme.

Overview of this chapter. In sections 3.2 and 3.3 we review some concepts and revisit Gentry's scheme, for the sake of completeness and independence of the Chapters. In section 3.4 we describe our key recovery attack, and present some numerical evidence in section 3.5.

3.2 Preliminaries

3.2.1 Notation

Vectors in $\mathbb{R}^n$ are noted with an arrow: $\vec{a}$, canonical vectors in $\mathbb{R}^n$ are noted by bold letters: $\mathbf{e}_1 = (1, 0, 0, \ldots, 0)$. If S if a set of vectors of $\mathbb{R}^n$, $\langle S \rangle$ denotes the vector space generated by elements in S, and $\langle S \rangle^\perp$ is the orthogonal complement of $\langle S \rangle$. If E is a set, $e \xleftarrow{\$} E$ means that e was sampled randomly from E using the uniform distribution. Unless stated otherwise, the norm in $\mathbb{R}^n$ is the Euclidean norm $||\vec{x}||_2 = \sqrt{\sum_i x_i^2}$. If P is a polytope, ∂P denotes its boundary. The $(n-1)$-th dimensional hyperplanes delimiting the interior of a parallelepiped are called facets, and the lines connecting two consecutive vertices are called edges. Two vertices are said to be neighbors if there is an edge that connects them. Overloading notation, if $p(x)$ is a polynomial of degree $n-1$, its coefficient vector $\alpha(p)$ will be also noted p. If (g) is an ideal in R, the ideal lattice $\alpha((g))$ will be also noted (g). The rotation basis of (g) is denoted by $\mathrm{rot}(g)$. Bases of ideal lattices will be noted with bold letters. All lattices in this article are of full rank, we identify their bases with square invertible matrices. Let I_n be the $n \times n$ identity matrix.

3.2.2 Lattices

Definition 3.2.1 (Lattice)**.** *A lattice in $\mathbb{R}^n$ is a discrete subgroup of $(\mathbb{R}^n, +)$ which spans $\mathbb{R}^m$ as a $\mathbb{R}$-vector space.*

Every lattice L has a minimal subset of m elements $B \subset L$ such that $L = \bigoplus_{b \in B} \mathbb{Z}b$. When $n \geq 2$ there is an infinite number of such sets and we call them *bases* of L. The integer m does not depend on the chosen basis and is called the *rank* of L, therefore, to describe a lattice, only one basis is required. Given a basis of rank m in matrix form $\mathbf{B} = (\mathbf{b}_1 | \mathbf{b}_2 | \cdots | \mathbf{b}_m)$, the quantity $|\det(\mathbf{B})| := \sqrt{\det({}^t\mathbf{B}\mathbf{B})}$ is an invariant of L, which we note by $\det L$. Let the *fundamental parallelepiped* of the basis $\mathbf{B}$ be the translated convex hull delimited by the column vectors in $\mathbf{B}$. We note this parallelepiped by $P(\mathbf{B}) = \left\{ \sum_{1 \leq i \leq m} x_i \mathbf{b}_i \in \mathbb{R}^n \ ; \ x_i \in [-1/2, 1/2[\right\}$. By definition, the volume of $P(\mathbf{B})$ equals the determinant of the lattice: $\det(L) = \mathrm{meas}_n(P(\mathbf{B}))$ for any basis $\mathbf{B}$. Let L be a full rank lattice. If $x \in \mathbb{R}^n$, we note by $x \bmod L$ the equivalence class of x in the quotient group $\mathbb{R}^n / L$. We use the

fundamental parallelepiped $P(\mathbf{B})$ as the set of representatives of the equivalence classes when describing L with the basis $\mathbf{B}$. Therefore, it is natural to note by $x \bmod \mathbf{B}$ the singleton in $(x \bmod L) \cap P(\mathbf{B})$. To reduce a vector $x \in \mathbb{R}^n$ modulo $\mathbf{B}$, use $\mathbb{Z}$-linear combinations of vectors in $\mathbf{B}$ to find the equivalent vector inside $P(\mathbf{B})$. In other words,

Lemma 3.2.1. *Let $x \in \mathbb{R}^n$ and $\mathbf{B}$ a basis of a full-rank lattice L. Then $x \bmod \mathbf{B} = x - \mathbf{B}\lfloor \mathbf{B}^{-1}x \rceil$, where $\lfloor v \rceil := (\lfloor v_1 \rceil, \lfloor v_2 \rceil, \cdots, \lfloor v_n \rceil)$ is the closest $\mathbb{Z}^n$ vector to v.*

In the scheme, two *coprime* lattices I, J will define the plaintext and ciphertext space respectively.

Definition 3.2.2 (Relatively prime lattices)**.** *Two lattices L, L' of $\mathbb{Z}^n$ are relatively prime (or coprime) if their sum is $\mathbb{Z}^n$. We note this $(L, L') = 1$.*

Keys in Gentry's scheme are lattice bases. The private-key is a "good" basis in the orthogonality defect sense and the public key is an eccentric, large basis of the same lattice. More precisely, this large basis is the *Hermite Normal Form* of the lattice.

3.2.3 Polynomial ideal lattices

Let $(R, +, \times)$ be a commutative ring and $\alpha : R \to \mathbb{R}^n$ an additive group homomorphism, then $\alpha(R)$ is a lattice. This yields because α transports the additive structure of R to points in $\mathbb{R}^n$ in order to generate linear integer combinations of them.

Definition 3.2.3 (Ideal lattice)**.** *Let A be a commutative ring and $\alpha : A \to \mathbb{R}^n$ an additive ring homomorphism. An ideal lattice of $\mathbb{R}^n$ is a lattice of the form $\alpha(I)$ for a principal ideal $I \subset A$.*

Consider the ring $R = \mathbb{Z}[x]/(f(x))$ where f is a monic polynomial of degree n, and the natural homomorphism $\alpha : R \to \mathbb{Z}^n$ given by $\alpha : a(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} \mapsto (a_0, \ldots, a_{n-1})$. We can now define a precise ideal lattice: if $I = (v)$ is a principal ideal of R, then $\alpha(I)$ is a lattice of $\mathbb{R}^n$ with a remarkable basis:

Lemma 3.2.2. *Let $v \in R$, $I = (v)$ and $v_i = v \times X^i \bmod f(X)$ for $i = 1, \ldots, n-1$. Then a basis of the ideal lattice $\alpha(I)$ is given by a linearly independent subset of $\mathrm{rot}(v) := \{v, v_1, v_2, \cdots, v_{n-1}\}$.*

Definition 3.2.4 (Rotation basis)**.** *The set $\{v_i = v \times x^i \bmod f(x); i = 0, \ldots, n-1\}$ is called the rotation basis of (v). The matrix $\mathrm{rot}(v) = (v|v_1| \ldots |v_{n-1})$ is called the rotation matrix of v.*

Proposition 3.2.3 (Inversing rotation matrices)**.** *Let $v \in R^\times$. Then $\mathrm{rot}(v) \times \mathrm{rot}(v^{-1}) = I_n$.*

3.2.4 Key-Recovery Attacks against SHE schemes

In the chosen ciphertext (CC) model, an attacker faces an oracle possessing a secret who publishes a challenge – the encryption of a random message r – and the goal of the attacker is to guess the message. To this end they engage in exchanges, in which the attacker submits a set of ciphertexts of his choice (different from the challenge ciphertext) and the oracle responds with the decryption of each ciphertext. The publication of the challenge can occur after or before the exchanges (which is referred to as CCA-1 or CCA-2). The attack is successful whenever the adversary is able to guess r with better probability than a third party who guesses the plaintext without access to any decryption oracle. If in the course of these exchanges, the attacker reconstructs the oracle's secret, the attack is said to be a Key-Recovery (KR) attack. This is evidently stronger than a CC attack. To the date, a number of KR attacks against SHE schemes have been published, including one against the Gentry-Halevi variant of Gentry's ideal lattice scheme [LMSV12]. As this variant is essentially homomorphic to the $I = (2)$ Gentry's scheme, their attack applies in this case, as mentioned in the abstract of [LMSV12]. The attack we propose is, to the best of our knowledge, the first direct KR attack to [Gen09b, Gen09c].

Let us specify the KR game addressed by our attack. Let $\mathcal{E} = (\mathsf{Keygen}, \mathsf{Enc}, \mathsf{Dec})$ be a public encryption scheme and let $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Keygen}(1^\lambda)$. Let $\mathcal{O}$ be an oracle that possesses sk and assume there is an adversary $\mathcal{A}$ with knowledge of every public element of the scheme.

Key-Recovery Game.

1. $\mathcal{A}$ generates a ciphertext c and sends it to $\mathcal{O}$.
2. $\mathcal{O}$ computes the underlying plaintext m and sends it to $\mathcal{A}$.
3. Repeat the two last steps ℓ times.
4. $\mathcal{A}$ makes a guess sk', and wins if $\mathsf{sk}' = \mathsf{sk}$.

We say that the attack is successful if $\mathcal{A}$ wins the game with better probability than a random guesser.

3.3 Gentry's ideal lattice basic scheme

With the homomorphism $\mathbb{Z}^n \simeq R = \mathbb{Z}[x]/(f(x))$, ring operations are inherited by structure transport. This way, for all $v, w \in \mathbb{Z}^n$ and a basis $\mathbf{B}$ we define addition $v + w := \alpha(v(x)+w(x))$, product $v \times w := \alpha(v(x)\times w(x))$, and basis reduction $v \bmod \mathbf{B} := v - \mathbf{B}\lfloor \mathbf{B}^{-1} v \rceil$. We are now ready to define the scheme. The scenario of this scheme is as follows

– Fix a dimension $n \geq 2$ and a monic polynomial f of degree n. To prevent lattice reduction attacks, n must be sufficiently large.

– Let $R = \mathbb{Z}[x]/(f(x))$. In order to control the norm of products, f is of the form $X^n + h(X)$ where $\deg(h)$ is small (for instance $h(X) = -1$).

– Let I be an ideal lattice with a basis $\mathbf{B}_I$. The binary plaintext setting is set with $I = (2)$, although we do not limit ourselves to this case.

– Let J be an ideal lattice relatively prime to I, with two bases $\mathbf{B}_J^{\mathsf{sk}}, \mathbf{B}_J^{\mathsf{pk}}$. The secret $\mathbf{B}_J^{\mathsf{sk}}$ is quite orthogonal, with large volume such that $P(\mathbf{B}_J^{\mathsf{sk}})$ contains $P(\mathbf{B}_I)$, and $\mathbf{B}_J^{\mathsf{pk}}$ is very eccentric (defined by large and almost parallel vectors).

– Choose an algorithm $\mathsf{Samp} : \mathbb{Z}^n/(I) \to \mathbb{Z}^n$ that given a vector x, samples randomly from $x + (I)$. The samples should lie inside a ball of fixed radius l_{Samp} contained in $P(\mathbf{B}_J^{\mathsf{sk}})$.

The plaintext space in the scheme is $\mathcal{P} := \mathbb{Z}^n \cap P(\mathbf{B}_I)$ (a small, quite orthogonal parallelepiped) and ciphertext space is $\mathbb{Z}^n \cap P(\mathbf{B}_J^{\mathsf{pk}})$ (a huge, very eccentric parallelepiped). The elements $n, f, I, J, \mathbf{B}_I$ are all public, and the keys are $\mathsf{pk} = \{\mathbf{B}_J^{\mathsf{pk}}\}, \mathsf{sk} = \{\mathbf{B}_J^{\mathsf{sk}}\}$. The scheme is described in 6.

Algorithm 6 Gentry's homomorphic ideal lattice scheme

function $\mathsf{Keygen}(\lambda)$
    Generate $\mathbf{B}_J^{\mathsf{sk}}$ and $\mathbf{B}_J^{\mathsf{pk}}$ randomly from a security parameter.
end function

---

function $\mathsf{Enc}(\mathsf{pk}, \pi \in P(\mathbf{B}_I))$
    To encrypt message π, compute $\psi \leftarrow \mathsf{Samp}(\pi) \bmod \mathbf{B}_J^{\mathsf{pk}}$ and output ψ. It is an integer vector of the form $\pi + i + j$ where $i \in I, j \in J$, and it is inside $P(\mathbf{B}_J^{\mathsf{pk}})$.
end function

---

function $\mathsf{Dec}(\mathsf{sk}, \psi)$
    To decrypt ψ, compute $\mu = (\psi \bmod \mathbf{B}_J^{\mathsf{sk}}) \bmod \mathbf{B}_I = (\psi - \mathbf{B}_J^{\mathsf{sk}} \lfloor (\mathbf{B}_J^{\mathsf{sk}})^{-1} \psi \rceil) \bmod \mathbf{B}_I$ and output μ.
end function

---

function $\mathsf{Eval}(\mathsf{pk}, \mathrm{C}, \psi_1, \cdots, \psi_t)$
    Perform the circuit C on inputs $\psi_1, \cdots, \psi_t$ replacing each "+" and "×" gate of C by addition and multiplication modulo $\mathbf{B}_J^{\mathsf{pk}}$ respectively. Output the result.
end function

3.4 The attack

We assume that the secret basis is the rotation matrix of a secret polynomial, i.e. $\mathbf{B}_J^{\mathsf{sk}} := \mathrm{rot}(v^{\mathsf{sk}})$ (this is the setting proposed in [Gen09b] and maintained in later revisions). In other words, the secret basis is generated by any of its elements. This implies that the full parallelepiped may be retrieved when a vector along any edge is known. We first show that any algorithm solving ad-HPP may be used to recover a secret vector in Gentry's scheme: to this end we define a function $\mathbb{Z}^n \to \{0,1\}$ that matches the indicator function of $P(\mathbf{B}_J^{\mathsf{sk}})$ over $\mathbb{Z}^n$ with overwhelming probability, and estimate the diameter bound D using public values.

3.4.1 Reduction to ad-HPP

In order to recover the secret key $\mathbf{B}_J^{\mathsf{sk}}$ of the scheme, first remark that only an edge of $P(\mathbf{B}_J^{\mathsf{sk}})$ is sufficient, since it allows to find a polynomial that generates the basis by rotations. Recall that the ciphertext space is $P(\mathbf{B}_J^{\mathsf{pk}})$. The decryption equation of a ciphertext $\psi \in P(\mathbf{B}_J^{\mathsf{pk}})$ with secret-key $\mathbf{B}_J^{\mathsf{sk}}$ is

$$\mathsf{Dec}(\mathbf{B}_J^{\mathsf{sk}}, \psi) = (\psi \mod \mathbf{B}_J^{\mathsf{sk}}) \mod \mathbf{B}_I.$$

Now, consider the function $g : \mathbb{Z}^n \to P(\mathbf{B}_I)$:

$$g : x \mapsto (x \mod \mathbf{B}_I) - \mathsf{Dec}(\mathbf{B}_J^{\mathsf{sk}}, x).$$

However, note that x may not lie in the ciphertext space, and the decryption query may be refused depending on the setting. Addressing this, define $\theta : \mathbb{Z}^n \to P(\mathbf{B}_I)$ as

$$\theta = g \circ \mathrm{mod}\ \mathbf{B}_J^{\mathsf{pk}}.$$

It follows that $\theta(x) = 0$ for all $x \in P(\mathbf{B}_J^{\mathsf{sk}})$ and, with overwhelming probability, $\theta(y) \neq 0$ if $y \notin P(\mathbf{B}_J^{\mathsf{sk}})$. Finally, define $\chi : \mathbb{Z}^n \to \{0,1\}$ as follows

$$x \mapsto \chi(x) = \begin{cases} 1 \text{ if } \theta(x) = 0, \\ 0 \text{ if } \theta(x) \neq 0. \end{cases}$$

This function matches the indicator function of $P(\mathbf{B}_J^{\mathsf{sk}})$ in $\mathbb{R}^n$ with overwhelming probability. Remark also that the attacker can estimate the diameter as $D \approx 2\sqrt{n} \cdot ||v^{\mathsf{sk}}||_2$, or using only public values,

$$D \approx \sqrt{n} \cdot \mathrm{Vol}(P(\mathbf{B}_J^{\mathsf{sk}}))^{1/n} = \sqrt{n} \cdot |\det P(\mathbf{B}_J^{\mathsf{sk}})|^{1/n} = \sqrt{n} \cdot |\det P(\mathbf{B}_J^{\mathsf{pk}})|^{1/n}.$$

For correct evaluation of k successive multiplications, $||v^{\mathsf{sk}}||_2 = O(n^k)$, hence $\log D = O(\log n)$. The attacker guesses the secret key feeding (χ, D) to an algorithm solving ad-HPP.

3.4.2 Feeding Nguyen-Regev's algorithm and Random Scattering

We first propose a method that solves ad-HPP using the well-known Nguyen-Regev algorithm, and a second proposal that collects information about the facets of P. We point out that these two methods require a large number of decryption queries.

Authors in [NR06] solve non adaptive HPP given a list of uniformly random points in the interior of the secret parallelepiped. They first perform a transformation to obtain a list of points inside a hypercube of dimension n and then they recover the shape with a multivariate optimization problem, solving by gradient descent. This gives a first approach to solve ad-HPP in a non adaptive manner: just ask the decryption of a uniform set of points inside a ball of radius $D/2$. In the case of Nguyen and Regev, the knowledge of a point inside a NTRU parallelepiped implies the knowledge of a number of different points. This is because of symmetries in NTRU lattices: as stated in [NR06], from a list of 400 points inside the parallelepiped, it is possible to generate 100,400. In Gentry's ideal lattices, however, no such symmetries are present. According to their experimental analysis in GGH signatures, over 200,000 uniformly distributed points are needed to disclose a secret parallelepiped of diameter $\approx 4n$ in dimension $n = 400$. Gentry's parallelepiped has diameter $O(n^k)$ where k is the maximum allowed depth of evaluation circuits, drastically increasing the number of required samples. Nevertheless, we point out that new samples can be generated using homomorphic evaluations, although disturbing the uniformity assumption of Nguyen-Regev algorithm. We leave the analysis of this possibility as an open problem.

In the second method we propose, the attacker finds $K = O(n)$ points near a neighborhood of ∂P and fit an hyperplane between them in order to approximate a facet. They repeat the process in other $O(n)$ directions, and then solve linear systems to find approximation of vertices.

To find a single point near ∂P, the attacker starts from 0, chooses a random direction $\vec{d} \in \mathbb{R}^n$ and tests integer points near $\langle \vec{d} \rangle$ (i.e. inside the cylinder $\mathcal{B}_n(0, \sqrt{n}) \times \langle \vec{d} \rangle$). They can find a point lying within distance of at most $\sqrt{n}$ of ∂P in $O(\log D)$ calls to the membership function χ, since $\text{dist}(O, \partial P) \leq D/2$. To search the remaining points near the facet, they can repeat this procedure using small perturbations of $\vec{d}$ of the form $\vec{d_i} = \vec{d} + \vec{r_i}$ for $i = 1, \ldots, K-1$ with $||\vec{r_i}||_2 \ll ||\vec{d}||_2$ (in order to avoid other facets). Another approach to find points near a facet is to perform the dichotomy search inside a cylinder $\mathcal{B}_n(0, r) \times \langle \vec{d} \rangle$ for $r > \sqrt{n}$. Once K points near each facet are found, the attacker fits an hyperplane (for instance using linear least squares). The procedure finds an approximation of a vertex in $O(n^2 \log D)$ calls to the indicator function χ.

There are many possible descriptions and optimizations of this algorithm, which cost more calls to χ but gives more precision in the approximation of vertices. We state that this can be carried out in $O(n^2 \log D)$ calls to χ and the result are points

at distance at most $O(n)$ from vertices. As this method requires too many queries and the algorithm analysis is straightforward, we focus on our next proposal.

3.4.3 Falling towards vertices

In this approach, the attacker travels onto vertices imitating the effect of gravity and bouncing against ∂P. This is achieved with a first stage of successive falls to find a vertex V, and a second stage to find a neighbor vertex V' of V. Let us first expose the algorithms and then describe our attack.

- Touch_Border: Starting from a point inside P and considering a direction $\vec{u}$, it performs a binary search to find the closest integer point to the boundary ∂P in that direction.
- Bounces: Using Touch_Border, it finds n integer points close to ∂P that form an hyperrectangle that roughly points to the vertex.
- Find_Vertex: Uses Bounces repeatedly to find a vertex of P.
- Find_Neighbor_Vertex: Given a vertex V, it outputs another vertex V' connected to V by an edge.

Algorithm 7 Finding a point near ∂P

Require: An indicator function χ of a parallelepiped P, the diameter D of P, a point $x_0 \in P$, and a vector $\vec{u}$.

Ensure: Two points $x \in P$ and $y \notin P$ such that $\mathrm{d}(x, y) < \sqrt{n}$ and (x_0, x, y) are collinear up to translation of $\sqrt{n}$.

1: **function** TOUCH_BORDER($x_0, \vec{u}$)
2: $\quad \vec{u} \leftarrow \vec{u}/||\vec{u}||$
3: $\quad$ Let $x_1 \leftarrow x_0 + D\vec{u}$ ▷ Notice that $x_1 \notin P$
4: $\quad$ **return** TOUCH_BORDER_REC($x_0, x_1, \vec{u}$)
5: **end function**

6: **function** TOUCH_BORDER_REC($x_{\text{in}}, x_{\text{out}}, \vec{u}$)
7: $\quad$ **if** $\mathrm{d}(x_{\text{in}}, x_{\text{out}}) < \sqrt{n}$ **then**
8: $\quad\quad$ **Return** $(x_{\text{in}}, x_{\text{out}})$
9: $\quad$ **else**
10: $\quad\quad$ Let $y \leftarrow \lfloor x_{\text{in}} + \frac{||x_{\text{out}} - x_{\text{in}}||}{2}\vec{u} \rceil$
11: $\quad\quad$ **if** $(\chi(y) = 1)$ **then**
12: $\quad\quad\quad$ **Return** TOUCH_BORDER_REC($y, x_{\text{out}}, \vec{u}$)
13: $\quad\quad$ **else**
14: $\quad\quad\quad$ **Return** TOUCH_BORDER_REC($x_{\text{in}}, y, \vec{u}$)
15: $\quad\quad$ **end if**
16: $\quad$ **end if**
17: **end function**

Algorithm 8 Orthogonal bounces towards a vertex

Require: An indicator function χ of a parallelepiped P, an interior point $x \in P$, a set of orthogonal vectors $E = \{\vec{v_1}, \cdots \vec{v_{n-1}}\} \subset \mathbb{R}^n$ and a vector $\vec{u} \in \mathbb{R}^n$ independent from E.

Ensure: A set of n points $S \subset \mathbb{Z}^n$ such that $\mathrm{d}(y, \partial P) < \sqrt{n}$ for all $y \in S$.

function NEXT_POINT($x \in \mathbb{Z}^n, \vec{u}$)
    Outputs y, the integer point closest to the segment $\{x + \lambda\vec{u}/||\vec{u}|| : 1 \leq \lambda \leq 2\sqrt{n}\}$.
end function
function BOUNCES($x, \vec{u}, \vec{v_1} \ldots, \vec{v_{n-1}}$)
    $y_0 \leftarrow x$
    $\vec{v_0} \leftarrow \vec{u}$
    for $i = 1$ to n **do**
        if $\chi(\text{NEXT\_POINT}(y_{i-1}, \vec{v}_{i-1})) = 1$ **then** $(y_i, \cdot) \leftarrow$TOUCH_BORDER$(y_{i-1}, \vec{v}_{i-1})$
        else if $\chi(\text{NEXT\_POINT}(y_{i-1}, -\vec{v}_{i-1})) = 1$ **then** $(y_i, \cdot) \leftarrow$TOUCH_BORDER$(y_{i-1}, -\vec{v}_{i-1})$
        else $y_i = y_{i-1}$
        end if
    end for
    return $\{y_1, \ldots, y_n\}$
end function

Algorithm 9 Fall onto a vertex

Require: A parallelepiped P centered at O, the diameter D of P, and a set of orthogonal vectors $\{\vec{u}, \vec{v_1} \ldots, \vec{v_{n-1}}\}$.

Ensure: A point in $\mathbb{Z}^n$

function FIND_VERTEX($\vec{u}, \vec{v_1} \ldots, \vec{v_{n-1}}$)
    $\{y_1, \ldots, y_n\} \leftarrow$BOUNCES$(O, \vec{u}, \vec{v_1} \ldots, \vec{v_{n-1}})$
    $x \leftarrow (y_1 + y_n)/2$
    $\vec{u'} \leftarrow \overrightarrow{Ox}$ ▷ Note that $\vec{u'}$ is not necessarily orthogonal to $\{\vec{v_1}, \ldots, \vec{v_{n-1}}\}$
    return FIND_VERTEX_REC$(x, \vec{u'}, \vec{v_1}, \ldots, \vec{v_{n-1}})$
end function

---

function FIND_VERTEX_REC($x, \vec{u'}, \vec{v_1}, \ldots, \vec{v_{n-1}}$)
    $\{y_1, \ldots, y_n\} \leftarrow$BOUNCES$(x, \vec{u'}, \vec{v_1}, \ldots, \vec{v_{n-1}})$
    if $(y_1 = y_n)$ **then**
        return y_n
    else
        $x \leftarrow (y_1 + y_n)/2$
        $\vec{u'} \leftarrow \overrightarrow{Ox}$
        return FIND_VERTEX_REC$(x, \vec{u'}, \vec{v_1}, \ldots, \vec{v_{n-1}})$
    end if
end function

Algorithm 10 Finding a neighbor vertex

Require: The center O and a vertex V of a parallelepiped P. A superior bound L.
Ensure: A vertex V' such that $||V' - V|| < L$.

function FIND_NEIGHBOR_VERTEX(V, L)
  $\vec{u} \leftarrow V - O$.
  $\vec{v} \xleftarrow{\$} \langle \vec{u} \rangle^{\perp}$
  return FIND_NEIGHBOR_VERTEX_REC($O, V, \vec{u}, \vec{v}, L, 0, 1$)
end function
—— ▷ Look by dichotomy a direction that leads to a neighbor vertex
function FIND_NEIGHBOR_VERTEX_REC($O, V, \vec{u}, \vec{v}, L, a, b$)
  $t \leftarrow \frac{a+b}{2}$
  $\vec{w} \leftarrow (1-t)u + tv$
  Compute W', an orthogonal base of $\langle w \rangle^{\perp}$
  $V' \leftarrow$FIND_VERTEX($O, \vec{w}, W'$)
  if $V = V'$ **then return** FIND_NEIGHBOR_VERTEX_REC($V, \vec{u}, \vec{v}, L, \frac{a+b}{2}, b$)
  else if $||V' - V|| < L$ **then return** V'.
  else return FIND_NEIGHBOR_VERTEX_REC($V, \vec{u}, \vec{v}, L, a, \frac{a+b}{2}$)
  end if
end function

Falling towards V

We now detail the algorithms, prove their correctness and state their complexity. The aim is to find a vertex of P. First choose a set of orthogonal vectors $\vec{u}, \vec{v}_1, \ldots, \vec{v}_{n-1}$ and start from the center of the parallelepiped, denoted O. Using algorithm 7, find a point $y_1 \in P$ along $O + \langle \vec{u} \rangle$ such that $\text{dist}(y_1, \partial P) < \sqrt{n}$. From y_1, repeat the procedure advancing in the inward pointing direction in $\{-\vec{v}_1, \vec{v}_1\}$: again use algorithm 7 along $y_1 + \langle \vec{v}_1 \rangle$ to find a point $y_2 \in P$ such that $\text{dist}(y_2, \partial P) < \sqrt{n}$ (if there are no inward pointing directions, simply set $y_2 = y_1$). Continuing recursively, at step k we have a point y_k near the boundary, from which we choose an inward pointing direction in $\{\vec{v}_k, -\vec{v}_k\}$. If no such directions are available, set $y_{k+1} = y_k$, else travel with algorithm 7 along $y_k + \langle \vec{v}_k \rangle$ to find a point y_{k+1} near ∂P. This procedure performs n steps and outputs n points near the boundary. The fall is detailed in algorithm 8. The next step is to compute $c = (y_1 + y_n)/2$ and repeat the fall starting from c in the direction $\vec{u}' = \overrightarrow{Oc}$ while maintaining the previous directions $\vec{v}_1, \ldots, \vec{v}_{n-1}$. Continue recursively and stop when there are no inward pointing, norm growing directions. The whole procedure is detailed in algorithm 9. We show in theorem 3.4.3 that this procedure stops, and outputs a vertex in $O(n \log^2(D))$ calls to χ.

Proposition 3.4.1. *The* TOUCH_BORDER *algorithm has a complexity of* $O(\log D)$

calls to χ.

Proof. Let k be the number of recursive calls to TOUCH_BORDER. At each step the function computes a candidate to a point near the border, and reduces by 2 that distance, which is initially less than D. □

Let us discuss the correctness of the FIND_VERTEX algorithm. Without loss of generality take the canonical vectors $\vec{u} = \mathbf{e}_1, \vec{v}_1 = \mathbf{e}_2, \ldots, \vec{v}_{n-1} = \mathbf{e}_n$. Let O be the center of P, and define $S_0 \leftarrow$BOUNCES$(O, \mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n)$. Recall that S_0 is a set of n points very close to ∂P, generated by bouncing against ∂P successively along the directions $\mathbf{e}_1, \ldots, \mathbf{e}_n$. Parse $S_0 = \{x_1^{(0)}, \ldots, x_n^{(0)}\}$, and note that the sequence $||x_1^{(0)}||, \ldots, ||x_n^{(0)}||$ is strictly increasing with overwhelming probability, unless we have reached a vertex. Now, set $c_0 \leftarrow (x_1^{(0)} + x_n^{(0)})/2, u_0 = \overrightarrow{Oc_0}$ and define recursively for $i \geq 1$:

$$\begin{array}{rcl} S_i & = & \{x_1^{(i)}, \ldots, x_n^{(i)}\} \leftarrow \text{BOUNCES}(c_{i-1}, \vec{u}_{i-1}, \mathbf{e}_2, \ldots, \mathbf{e}_n), \\ c_i & \leftarrow & (x_1^{(i)} + x_n^{(i)})/2 \\ u_i & \leftarrow & \overrightarrow{Oc_i} \end{array}$$

Proposition 3.4.2. *The program* FIND_VERTEX *outputs a point in $\mathbb{Z}^n$ in a finite number of steps.*

Proof. The algorithm outputs a point when there are no inward-pointing and norm-growing directions available (by convexity of the parallelepiped this is a vertex). Before this happens, the integer sequence of bouncing points norms

$$||x_1^{(0)}||, \ldots, ||x_n^{(0)}||, ||x_1^{(1)}||, \ldots, ||x_n^{(1)}||, ||x_1^{(2)}||, \ldots ||x_n^{(2)}||, \ldots$$

is strictly increasing and bounded by $D/2$. □

We quantify in the following theorem the ratio of diameters of two successive hyperrectangles containing the bounces in the process of finding a vertex.

Theorem 3.4.3. *For all $k \geq 0$, let $d_k := \text{dist}(x_1^{(k)}, x_n^{(k)})$. Then $d_{k+1} \leq d_k/2$.*

Proof. At any iteration of the algorithm, we have an initial point c and suppose without loss of generality that the given set of vectors to perform a fall from c is $\{\vec{u}, \mathbf{e}_2, \ldots, \mathbf{e}_n\}$, where $\vec{u}$ is independent from $\{\mathbf{e}_2, \ldots, \mathbf{e}_n\}$. Let $x_1, \ldots, x_n$ be the bouncing points obtained after n bounces. Recall that by construction, we have

$$\left\{ \begin{array}{rcl} x_1 & = & c + \lambda_1 \vec{u}, \\ x_2 & = & x_1 + \lambda_2 \mathbf{e}_2, \\ x_3 & = & x_2 + \lambda_3 \mathbf{e}_3, \\ & \vdots & \\ x_n & = & x_{n-1} + \lambda_n \mathbf{e}_n, \end{array} \right.$$

for $\lambda_i \in \mathbb{Q}$, and let $c' = (x_1 + x_n)/2$, $\vec{u}' = \overrightarrow{Oc'}$. The next step is to perform a new fall starting from c', with initial direction $\vec{u}'$. After n new bounces against the wall (in the same directions than the previous step) we will have points

$$\begin{cases} y_1 &= c' + \beta_1 \vec{u}', \\ y_2 &= y_1 + \beta_2 \mathbf{e}_2, \\ y_3 &= y_2 + \beta_3 \mathbf{e}_3, \\ &\vdots \\ y_n &= y_{n-1} + \beta_n \mathbf{e}_n, \end{cases}$$

First perform the translation $y'_1 = y_1 - \beta_1 \vec{u}', y'_2 = y_2 - \beta_1 \vec{u}', \ldots, y'_n = y_n - \beta_1 \vec{u}'$. We therefore have that $\text{dist}(y'_1, y'_n) = \text{dist}(y_1, y_n)$ and $y'_1 = c'$. Consider H, the n-hyperrectangle formed by $x_1, \ldots, x_n$. With the given translation y'_1 lies at its center. Notice also that every point $y'_2, \ldots, y'_n$ lies inside of this hyperrectangle, because if there is a point $y'_i \notin H$, the straight line connecting O and y_i has a segment outside the parallelepiped, contradicting the convexity of P. In particular, $y'_n \in H$, therefore its distance to the center is less than half the diameter of H. In other words,

$$\text{dist}(y_1, y_n) = \text{dist}(y'_1, y'_n) \leq \text{diameter}(H)/2 = \text{dist}(x_1, x_n)/2,$$

completing the proof. □

Corollary. *The* Find_Vertex *algorithm outputs a vertex of P after at most $\lceil \log(D) \rceil$ recursive calls.*

Proof. After k iterations, $d_k \leq \frac{d_0}{2^k} < \frac{D}{2^k}$, and $d_k < 1$ is therefore reached after at most $\lceil \log D \rceil$ iterations. The fact that this sequence converges to a vertex of P follows immediately by convexity of P. □

Proposition 3.4.4. *The* Find_Vertex *algorithm has a complexity of $O(n \log^2 D)$ calls to χ.*

Proof. At the first iteration of Bounces we have the bound $d_1 \leq D$. Since at each step we divide d_i by 2, then the number of steps is less than $\log D$. The most costly operation of Find_Vertex is Bounces, who needs n calls to Touch_Border. Therefore, Bounces has a complexity of $O(n \log(D))$ calls to χ. □

Falling towards V'

When a vertex V is found, one can repeat algorithm 9 to fall into a neighbor vertex, given a proper initial direction to fall into. Let us explain how to establish this direction and justify the correctness of our method.

Proposition 3.4.5. *Let P be a parallelepiped centered in O with a known vertex V. Let $\vec{v} = \overrightarrow{OV}$, and $\vec{u} \xleftarrow{\$} \langle\vec{v}\rangle^{\perp}$. Then, with overwhelming probability, there is a vector $\vec{w} \in \{(1-t)v + tu : t \in [0,1]\}$ such that a fall inside P in the direction $\vec{w}$ converges to a neighbor of V.*

Proof. Consider a linear transformation $\psi : [-1,1]^n \to P$ such that $\psi((1,1,\ldots,1)) = V$. Let $\vec{a} = \psi^{-1}(\vec{v}), \vec{b} = \psi^{-1}(\vec{u})$. Now, let $f : [0,1] \to \mathbb{R}^n$ such that $f(t) := t\vec{a} + (1-t)\vec{b}$. With overwhelming probability, $\vec{b}$ is not normal to a cube's facet, and in this case as $f(1) = \vec{a}$ and $f(0) = \vec{b}$, by continuity of f it follows that there is a value t_0 such that $f(t_0)$ has exactly one negative and $n-1$ positive non-zero coordinates (i.e., $f(t_0)$ points inside a neighbor hyperoctant). Now, note that a fall inside the cube towards a direction $\vec{d} = (d_1, d_2, \ldots, d_n)$ converges with overwhelming probability to the vertex $(\mathrm{sign}(d_1), \mathrm{sign}(d_2), \ldots, \mathrm{sign}(x_n))$ where $\mathrm{sign}(x) = x/|x|$. With this, a fall towards $f(t_0)$ converges to a neighbor C of $(1,1,\ldots,1)$, thus a fall inside the parallelepiped towards $\vec{w} = \psi(f(t_0))$ converges to $V' = \psi(C)$, a neighbor of V. □

This vector $\vec{w}$ can be found by dichotomy, which is what we propose in algorithm 10. Moreover, our numerical analysis of section 3.5 suggests that in dimension $n = 512$ there are about 64% chances that this dichotomy ends in less than 7 steps, and about 77% in less than 11 (compared to the theoretical $\log(D) \approx k\log(512)$ steps where k is the maximum allowed depth of the scheme). In dimension $n = 1024$, we report 42% and 57%, respectively. With this method, only one additional complete fall is necessary to find a neighbor vertex, the falls converging elsewhere can be discarded after one bounce against ∂P because of too small or excessive distances from V.

Summing up all procedures, the attacker may find two adjacent vertices V_1, V_2 in $O(n\log^3 D)$ calls to χ as follows:

1. $V_1 \leftarrow$FIND_VERTEX$(O, \mathbf{e}_1, \ldots, \mathbf{e}_n)$.
2. $V_2 \leftarrow$FIND_NEIGHBOR_VERTEX$(V, \alpha D/\sqrt{n})$ with a constant $\alpha \gtrapprox 1$

With two neighbor vertices, the attacker finally establishes $v^{\mathsf{sk}} = V_1 - V_2$.

3.5 Length of binary searches in the neighbor algorithm

Our algorithm 10 uses binary searches to find a direction leading to a neighbor vertex. In theory, these searches finds such a direction in $\log D$ steps. However, we conducted experiments and found that even in high dimensions, the search has good chances of succeeding in very few steps. Consider the hypercube $[-1,1]^n$ and let $v = (1,1,\ldots,1)$ be the already found vertex. It is clear that the neighbors of v are the n elements of $\{-1,1\}^n$ that possess only one negative coordinate, and

that a fall towards direction $(u_1, \ldots, u_n)$ inside the hypercube converges towards $(\text{sign}(u_1), \ldots, \text{sign}(u_n))$. Therefore, take a random vector $\vec{v} \in \langle \vec{u} \rangle^{\perp}$, the goal is to find an element $f(t_0)$ in the set $\{f(t) = (1-t)u + tv, t \in [0,1]\}$ with exactly one negative coordinate. We implemented the dichotomy in `C++`, and obtained apparent bounds for the parameter t_0 depending on the dimension, as well as the expected number of dichotomic tries. In the table below, k is the number of dichotomic steps performed. For each setting, we performed 10,000,000 searches. A success is when a search outputs a vector with only one negative coordinate before k recursions.

n	$[\min t_0, \max t_0]$	Mean(t_0)	$k \leq 7$ success rate	$k \leq 10$ success rate
256	[0.600,0.688]	0.630	79.8%	87.8%
512	[0.602,0.669]	0.631	64.2%	76.6%
1024	[0.610,0.659]	0.632	42.6%	57.3%
2048	[0.614,0.650]	0.634	19.9%	30.3%

Table 3.1: Direction searching in the dichotomy of algorithm 10.

Notice how the interval size containing the desired parameter diminishes as the dimension grows. We point out that if the linear transformation between the parallelepiped and the hypercube is almost homothetic up to rotations (for instance in Gentry's setting), a similar behavior should be observed when performing the binary search inside the parallelepiped.

3.6 Conclusion

We proposed a new key recovery attack on Gentry's ideal lattice fully homomorphic scheme, based on the adaptive Hidden Parallelepiped Problem. We proposed two solutions, one of them exposing the secret in $O(n \log^3 D)$ adaptive calls to the decryption oracle. It would be interesting to see an implementation, and to analyze the possibility of determining points inside the parallelepiped with help of homomorphic operations. We point out that the gap between theoretical and practical analysis is large. For instance, evidence in appendix B suggests that finding a new direction costs less than 10 parameter searches, with good probability, thus relaxing a $\log(D)$ factor. It is also safe to assume that the complexity of Bounces is a great deal lower than $O(n \log D)$, since each search starts from points progressively away from the origin. With this, we estimate that for an application of the scheme in $n = 334$ (safe against lattice reduction attacks) of depth d, the number of decryption queries is about $2500d^2$ to recover a secret key with good probability. It would be of interest to apply this attack to other lattice based encryption schemes with smaller parameters.

Chapter 4

A betrayal problem - The Excalibur Property

The contents of this Chapter include an article published in the Proceedings of Indocrypt 2016. This is joint work with Louis Goubin.

Contents

4.1 Introduction 77
4.2 Preliminaries 81
4.2.1 Notation 81
4.2.2 The quotient ring R_q 81
4.2.3 Bounded discrete Gaussian samplings on $\mathbb{Z}[x]/(x^n+1)$ . . . 82
4.3 Modified NTRU encryption 82
4.3.1 The multikey property 83
4.4 Hardness assumptions 84
4.4.1 Small Polynomial Ratio Problem, from [LATV12] 84
4.4.2 Small factorizations in the quotient ring 85
4.5 Two-party multiplication protocols in R_q 86
4.5.1 Secret inputs setting 87
4.5.2 Shared inputs setting 88
4.6 Excalibur key generation 89
4.7 Security 90
4.7.1 Honest-but-curious model 92
4.7.2 Security against one malicious party 93
4.8 Extensions 96
4.8.1 Chains of keys 96
4.8.2 Plugging in LATV-FHE 97
4.9 Conclusion 97

4.1 Introduction

Is it possible to avoid betrayal in a hierarchic scenario? Imagine a chain of users equipped with a public-key encryption scheme, where high level users can decrypt ciphertexts intended to all lower level users in the chain. This is trivial to construct using any public-key cryptosystem $\mathcal{E}$: just transfer low-level secret-keys to upper levels following the hierarchy. The evident drawback is that high-level users can betray their children and distribute their secrets to other parties. Using a proxy re-encryption procedure or multiple trapdoors is hence preferred, because parents do not have direct knowledge of their children's secrets. A proxy re-encryption scheme is a cryptosystem that allows a public transformation of ciphertexts such that they become decryptable to an authorized party. This is a particular case of a cryptosystem allowing delegation of decryption, which finds applications in mail redirection, for instance. In this Chapter, we give a solution to the betrayal issue in another perspective, relying on a new property we found in the well-known modified NTRU encryption scheme, and which we refer to as "Excalibur". Basically, this feature allows to generate a secret-key that decrypts encryptions under multiple public-keys and behaves like a regular key of the cryptosystem.

The Excalibur Property. A public-key encryption scheme $\mathcal{E} = (\mathsf{Keygen}, \mathsf{Enc}, \mathsf{Dec})$ with plaintext space $\mathcal{M}$ has the Excalibur property if there is an algorithm that allows two users Alice and Bob with key-pairs $(\mathsf{sk}_A^{old}, \mathsf{pk}_A^{old})$ and $(\mathsf{sk}_B, \mathsf{pk}_B)$ respectively to forge a new key-pair for Alice $(\mathsf{sk}_A, \mathsf{pk}_A)$ such that

- Alice's key sk_A can decrypt ciphertexts in $\mathsf{Enc}(\mathsf{pk}_A, \mathcal{M}) \cup \mathsf{Enc}(\mathsf{pk}_B, \mathcal{M})$.
- Bob cannot decrypt ciphertexts in $\mathsf{Enc}(\mathsf{pk}_A, \mathcal{M})$.
- Alice cannot generate a secret-key sk'_B that is able to decrypt ciphertexts in $\mathsf{Enc}(\mathsf{pk}_B, \mathcal{M})$ but is not able to decrypt ciphertexts in $\mathsf{Enc}(\mathsf{pk}_A, \mathcal{M})$ (i.e. she cannot give away access to Bob's secret without leaking her own).

The intuition is that sk_A is a one-way expression of $(\mathsf{sk}_A^{old}, \mathsf{sk}_B)$. As Alice owns decryption rights over Bob's ciphertexts, this can be seen as *automatic* proxy re-encryption, in the sense that the re-encryption procedure is the identity. The idea is to "glue" Alice and Bob secret-keys together, resulting on a master key given to Alice. This Excalibur master key can be separated into factors only by Bob, hence the name of the feature: Bob plays the role of young Arthur, who is the only man in the kingdom able to separate Excalibur from the stone. Moreover, Alice can glue her key to an upper user's key, who inherits decryption over Bob's ciphertexts, and so forth, and if we suppose that no user is willing to give away own secrets, this achieves automatic N–hop re-encryption and sets a hierarchic chain.

We therefore have a scheme in which a single private-key can decrypt messages under multiple public-keys, and we will see that if a group of low-level users cheated

in the joint key generation of this private-key (in order to sabotage or harden decryption), the secret-key holder may be able to trace it back to the wrongdoers, by simply testing decryptions and looking at the private-key's coefficients. In a sense, this is the inverse setting of a public-key traitor tracing scheme, where there are multiple secret-keys associated with a single public-key, and such that if a group of users collude in creating a new private-key achieving decryption with the public-key, it is possible to trace it to its creators, see for instance [BF99].

Three main advantages of this property over the trivial transfer of keys, over re-encryption schemes and over multiple trapdoor schemes are (i) there are no extra space or time costs: as soon as the keys are blended, the resulting key-pair acts as a fresh one and no ciphertext modification is necessary, (ii) our key generation procedure can be plugged directly into the (multikey) NTRU-based fully homomorphic encryption scheme, supporting homomorphic operations and automatic N-hop re-encryption and (iii) a user with a powerful key does not need to handle a "key ring" of secret-keys of her children; her key-pair $(\mathsf{sk}, \mathsf{pk})$ acts as a regular NTRU key. In contrast, the classical proxy re-encryption scenario is more flexible; a user can agree a decryption delegation at any moment to any user, whereas in our proposal once the keys are blended, modifications in hierarchy involve new key generations. This is why our proposal is more suitable to a rigid pre-defined hierarchic scenario.

Modified NTRU. The NTRUEncrypt cryptosystem is a public-key encryption scheme whose security is based on short vector problems on lattices. Keys and ciphertexts are elements of the polynomial ring $\mathbb{Z}[X]/\langle\phi(x)\rangle$ where $\phi(x) = x^n - 1$, and coefficients are considered modulo a large prime q. This scheme was defined in 1996 by Hoffstein, Pipher, Silverman and gained much attention since its proposal because of its efficiency and hardness reductions. In [SS11b], Stehlé and Steinfeld provided modifications to the scheme in order to give formal statistic proofs, which ultimately led to support homomorphic operations with an additional assumption in [LATV12]. Among these modifications, we highlight the change of ring and parameters restrictions: $R = \mathbb{Z}[x]/\langle\phi(x)\rangle$ where now $\phi(x) = x^n + 1$, n is a power of 2 (hence ϕ is the $2n$-th cyclotomic polynomial), and the large prime modulus is such that $x^n + 1$ splits into n different factors over $\mathbb{F}_q$ (namely, $q = 1 \mod 2n$). We will consider the modified NTRU scheme, but we believe that, possibly via a stretching of parameters, the original NTRU may also exhibit the Excalibur property.

Excalibur key generation. The way to glue two secret-keys is very simple: just multiply them together! Indeed, the modified NTRU scheme offers a fruitful property: *If one replaces a secret-key with a small polynomial multiple of it, decryption still works.* If this polynomial multiple is itself a secret-key, then by symmetry decryption with the resulting key will be correct in the union of ciphersets decryptable by one key or another. However, addressing the main point of this Chapter, parties must multiply the involved polynomials using multiparty protocols, since they do

not want to trust individual secrets to each other. To achieve this joint key generation, we rely on multiparty protocols in the polynomial ring $R_q = \mathbb{F}_q[x]/(x^n + 1)$ in both the secret and shared setting. To this end, we describe two multiplication protocols between mutually distrusting Alice and Bob:

1. *Secret inputs setting* : Alice and Bob hold $f, g \in R_q$ respectively. They exchange random polynomials and at the end Alice learns $fg + r \in R_q$ where r is a random polynomial known by Bob, and Bob learns nothing.

2. *Additively shared inputs setting* : Alice and Bob hold $f_A, g_A \in R_q$ and $f_B, g_B \in R_q$ respectively such that $f = f_A + f_B$ and $g = g_A + g_B$. They exchange some random polynomials, and at the end Alice and Bob learn π_A, π_B respectively such that $\pi_A + \pi_B = fg \in R_q$. Revealing π_A or π_B to each other does not leak information about the input shares.

Let us illustrate how to use these protocols in Alice's key generation. Suppose that Bob keys were previously generated. Generating Alice's secret-key is fairly easy: Informally, if $\beta \in R_q$ is Bob's secret-key, let Alice and Bob sample random $\alpha_A, \alpha_B \in R_q$ respectively, with small coefficients. They perform the first protocol on inputs $f = \alpha_A$ and $g = \beta$, and Bob chooses $r = \alpha_B\beta$. At the end, Alice learns $\gamma = \alpha_A\beta + \alpha_B\beta = \alpha\beta \in R_q$, and Bob learns nothing. One may stop here and let Alice compute her public-key $\mathsf{pk}_A = 2h\gamma^{-1} \in R_q$ for suitable $h \in R_q$, but she may cheat and generate other NTRU fresh keys $(\mathsf{sk}_A, \mathsf{pk}'_A)$ and then distribute freely Bob's secret γ. This is why the public-key is also generated jointly, and moreover, the public-key will be generated before the secret-key, this way Alice must first commit to a public-key pk_A.

Fully Homomorphic Encryption. Fully Homomorphic Encryption schemes allow public processing of encrypted data. Since Gentry's breakthrough in [Gen09a, Gen09b, Gen09c], there has been considerable effort to propose FHE schemes that are efficient [HS14, BGH13b, GHS12c, GHS12b, GHS12a, GHPS12, GH11b, ASP14, ASP13], secure [GH11a, BV11b, BV11a, Bra12, ASP14], and having other properties [BV11b, GH11a, BGH13b, GSW13]. We highlight the existence of Multikey FHE schemes, in which some ciphertexts can only be decrypted with the collaboration of multiple key-holders. This was first constructed in [LATV12], and it reduces the general multiparty computation problem to a particular instance. We encourage the reader to see the latest version of this article.

All of the above schemes have a PPT encryption algorithm that adds random "noise" to the ciphertext, and propose methods to add and multiply two ciphertexts. With these methods they give an (homomorphic) evaluation algorithm of circuits. The noise in ciphertexts grows with homomorphic operations (especially with multiplication gates) and after it reaches a threshold, the ciphertext can no longer be decrypted. Thus, only circuits of bounded multiplicative degree can be evaluated:

these schemes are referred to as leveled FHE schemes. Gentry proposed a technique called "bootstrapping" that transform a ciphertext into one of smaller noise that encrypts the same message, therefore allowing more homomorphic computations. This (algorithmically expensive) technique remains the only known way to achieve pure FHE scheme from a leveled FHE scheme. In order to do this, the decryption circuit of the leveled scheme must be of permitted depth and the new scheme relies on non-standard assumptions.

Nevertheless, leveled FHE schemes with good *a priori* bounds on the multiplicative depth do satisfy most applications requirements, see [YSK+13]. We suggest that the use of our protocols in the LATV scheme use the leveled version, but as pointed out in [LATV12], the scheme can be transformed into a fully homomorphic scheme by boostrapping and modulus reduction techniques, both adaptable to the use of Excalibur keys.

FHE and bidirectional multi-hop re-encryption paradigm. It has been widely mentioned (for instance in the seminal work [Gen09b]) that a fully homomorphic encryption scheme allows bidirectional multi-hop proxy re-encryption. The argument is similar to the celebrated bootstrapping procedure: let c be an encryption of m using Bob's secret-key s_B. First publish τ, an encryption of s_B under Alice's public-key, then homomorphically run the decryption circuit on c and τ, the result is an encryption of m decryptable by Alice's secret-key. However, we point out that this is *pure* re-encryption only if Alice never gets access to τ, since she can decrypt and learn s_B directly. This restriction tackles the pure re-encryption definition, and in light of this the NTRU-based FHE scheme with the Excalibur property may be a starting point to clear out this paradigm (as it satisfies the pure definition, but fails to be bidirectional).

Our contributions. In this chapter, we propose a key generation protocol that allows to glue NTRU secret-keys together in order to equip a hierarchic chain of users, such that a given user has the ability to decrypt all ciphertexts intended to all lower users in the chain, and she cannot give away secrets without exposing her own secret-key. This procedure can be plugged directly into the (multikey) FHE-scheme by Lopez-Alt et al. , it is compatible with homomorphic operations and has no space costs or ciphertext transformations, and important users do not have to handle key rings. To achieve this, we describe two-party computations protocols in cyclotomic polynomial rings that may be of independent interest. We base the semantic security on the hardness of RLWE and DSPR problems, and the semi-honest and malicious security in a new hardness assumption which we call "Small Factors Assumption". In this assumption we define the "Small GCD Problem" and we show that any algorithm solving this problem can be used to break the semantic security of the modified NTRU scheme.

4.2 Preliminaries

4.2.1 Notation

Let q be a large prime. We let the set $\{-\lfloor q/2 \rfloor, \ldots, \lfloor q/2 \rfloor\}$ represent the equivalence classes of $\mathbb{Z}/q\mathbb{Z}$, and both notations $[x]_q$ or $x \mod q$ represent modular reduction of x into this set. For a ring A, $A^\times$ stands for the group of units (or invertible elements) of A, $\langle a \rangle$ or (a) is the ideal generated by $a \in A$. Also, we denote by $\mathbb{F}_k$ the finite field of k elements, for $k = q^l \in \mathbb{Z}$. The notation $e \leftarrow \xi$ indicates that the element e is sampled according to the distribution ξ, and $e \xleftarrow{R} S$ means that e was sampled from the set S using the uniform distribution. Similarly, $A \overset{R}{\subset} S$ means that each $a \in A$ was sampled uniformly at random on S. Finally, let $R \overset{def}{=} \mathbb{Z}[x]/(x^n + 1)$, we identify an element of R with its coefficient vector in $\mathbb{Z}^n$, and for $v(x) = v_0 + v_1 x + \cdots + v_{n-1} x^{n-1}$ in R, we denote by $||v||_\infty, ||v||_2$ its l_∞, l_2 norm respectively.

4.2.2 The quotient ring R_q

Operations in the modified NTRU scheme are between elements of $R_q \overset{def}{=} \mathbb{F}_q[x]/(x^n + 1)$, the ring of polynomials modulo $\Phi_{2n}(x) = x^n + 1$ (i.e. Φ_{2n} is the $2n$–th cyclotomic polynomial) and coefficients in $\mathbb{F}_q$, where n is a power of 2 and q is a large prime. Addition and multiplication of polynomials are performed modulo $\Phi_{2n}(x)$ and modulo q. The ring R_q is not a unique factorization domain, in fact, small units of this ring serve as NTRU secret-keys. The Chinese remainder theorem shows that the group of units is large, and thus $y = ru \in R_q$ where $r \in R_q$ is a random element and u is a unit is a good masking of u: it is unfeasible to recover u from y for large n. Let us collect some lemmas related to the set of invertible elements of R_q.

Lemma 4.2.1. *Let $q \geq 3$ be a prime number and $\Phi_n(x) \in \mathbb{Z}[x]$ be the n–th cyclotomic polynomial. Then $\Phi_n(x)$ is irreducible over $\mathbb{F}_q$ if and only if q is a generator of the group $(\mathbb{Z}/n\mathbb{Z})^\times$.*

Lemma 4.2.2. *If $n > 2$ is a power of 2, then $(\mathbb{Z}/2n\mathbb{Z})^\times$ is not cyclic and therefore $\Phi_{2n}(x) = x^n + 1$ is not irreducible over $\mathbb{F}_q$. In addition, $x^n + 1$ decomposes into l distinct irreducible factors over $\mathbb{F}_q$ for prime $q \geq 3$: Let $(\phi_i)_{i=1}^l \subset \mathbb{F}_q[x]$ respectively such that $x^n + 1 = \prod_{i=1}^l \phi_i(x)$ over $\mathbb{F}_q$. Then we have a ring isomorphism*

$$\pi : \frac{\mathbb{F}_q[x]}{(x^n + 1)} \to \prod_{i=1}^{l} \frac{\mathbb{F}_q[x]}{(\phi_i(x))} \text{ where } \frac{\mathbb{F}_q[x]}{(\phi_i(x))} \simeq \mathbb{F}_{q^{\deg \phi_i}}.$$

Corollary. $\mathrm{Card}(R_q^\times) = \prod_{i=1}^l \left(q^{\deg \phi_i} - 1\right)$.

The proofs are straightforward. In the original modifications in [SS11b], $q = 1 \mod 2n$ and hence $x^n + 1$ splits into n distinct linear factors, yielding $\mathrm{Card}(R_q)^\times = (q-1)^n$.

4.2.3 Bounded discrete Gaussian samplings on $\mathbb{Z}[x]/(x^n+1)$

Let n be a power of 2 and q a prime number, $R = R_0 \stackrel{def}{=} \frac{\mathbb{Z}[x]}{(x^n+1)}$ and as before $R_q \stackrel{def}{=} \frac{\mathbb{F}_q[x]}{(x^n+1)}$. The modified NTRU scheme uses a particular distribution in R_q, which we refer to as K-bounded by rejection discrete Gaussian, serving to sample both message noises and secret-keys. Definitions follow.

Definition 4.2.1. *Let $\mathcal{G}_r$ be the discrete Gaussian distribution over R, centered about 0 and of standard deviation r.*

Sampling from $\mathcal{G}_r$ can be done in polynomial time, for instance approximating with Irwin-Hall distributions and outputting the nearest integer vector. Consider the following definitions from [LATV12]:

Definition 4.2.2. *A polynomial $e \in R$ is called K-bounded if $||e||_\infty < K$.*

Definition 4.2.3. *A distribution is called K-bounded over R if it outputs a K-bounded polynomial.*

Definition 4.2.4. *(K-bounded by rejection discrete Gaussian) Let $\bar{\mathcal{G}}_K$ be the discrete distribution $\mathcal{G}_{K/\sqrt{n}}$ that repeats sampling if the output is not K-bounded.*

Lemma 4.2.3 (Expansion factors for $\phi(x) = x^n + 1$, from [LATV12])**.** *For any polynomials $s, t \in R$,*

$$\begin{aligned} ||s \cdot t \mod \phi(x)||_2 &\leq \sqrt{n} \cdot ||s||_2 \cdot ||t||_2, \\ ||s \cdot t \mod \phi(x)||_\infty &\leq n \cdot ||s||_\infty \cdot ||t||_\infty. \end{aligned}$$

Corollary. *Let χ be a K-bounded distribution over R and let $s_1, \ldots, s_l \leftarrow \chi$. Then $\prod_{i=1}^{l} s_i$ is $(n^{l-1}K^l)$-bounded.*

4.3 Modified NTRU encryption

We review the modified NTRU encryption scheme as presented in [LATV12], and we insist on the multi-key property. The message space is $\{0,1\}$ and the ciphertext space is $R_q = \frac{\mathbb{F}_q[x]}{(x^n+1)}$. Let q be a large prime, $0 < K \ll q$, n be a power of 2 and $\bar{\mathcal{G}}_K$ be the K-bounded by rejection discrete Gaussian. A key-pair $(\mathsf{sk}, \mathsf{pk})$ is a tuple of polynomials in R_q, the secret-key being K-bounded.

$\underline{\mathsf{Keygen}(1^\kappa)}$:

Step 1. Sample a polynomial $f \leftarrow \bar{\mathcal{G}}_K$. Set $\mathsf{sk} = 2f + 1$, if sk is not invertible in R_q start again.
Step 2. Sample a polynomial $g \leftarrow \bar{\mathcal{G}}_K$ and set $\mathsf{pk} = 2g \cdot \mathsf{sk}^{-1} \in R_q$.
Step 3. Output $(\mathsf{sk}, \mathsf{pk})$.

$\underline{\mathsf{Enc}(\mathsf{pk}, m)}$: Sample polynomials $s, e \leftarrow \bar{\mathcal{G}}_K$. For message $m \in \{0,1\}$, output $c = m + 2e + s \cdot \mathsf{pk} \mod q$.

$\underline{\mathsf{Dec}(\mathsf{sk}, c)}$: For a ciphertext $c \in R_q$, compute $\mu = c \cdot \mathsf{sk} \in R_q$ and output $m = \mu \mod 2$.

4.3.1 The multikey property

We describe a decryption property that states that one can decrypt a ciphertext with the secret-key required for decryption, or a small polynomial multiple of it.

Lemma 4.3.1. *Let $(f,h) \leftarrow \mathsf{Keygen}(1^\kappa)$, $m \in \{0,1\}$ and let $c \leftarrow \mathsf{Enc}(h,m)$. Let $\theta \in R$ be a M-bounded polynomial satisfying $\theta \mod 2 = 1$. If $M < (1/72)(q/n^2K^2)$, then*

$$\mathsf{Dec}(f,c) = \mathsf{Dec}(\theta \cdot f, c) = m.$$

Proof: There exist K-bounded polynomials s, e such that $c = m + hs + 2e$. Decryption works since

$$[fc]_q = [fm + fhs + 2fe]_q = [fm + 2gs + 2fe]_q$$

and supposing there is no wrap-around modulo q in the latter expression, we have $[fc]_q \mod 2 = fc \mod 2 = m$. If we replace f by $\theta \cdot f$ and try to decrypt, we have $\theta fc = \theta fm + 2\theta gs + 2\theta fe$, and then again, if there is no wrap-around modulo q (i.e. if M is small enough), $\theta fc \bmod 2 = m$ is verified. To ensure that there is no wrap-around modulo q, one has to give an *a priori* relation between K, n and M. In fact, using corollary 4.2.3, we have $||gs||_\infty < nK^2$ and $||fe||_\infty < n(2K+1)K$, and thus

$$||fc||_\infty < 2nK^2 + 2n(2K+1)K + K.$$

Decryption using f is correct if $2nK^2 + 2n(2K+1)K + K < q/2$, and decryption using θf is correct if $nM(2nK^2 + 2n(2K+1)K + K) < q/2$. Therefore, decryption using f is ensured by $36nK^2 < q/2$, decryption using θf is ensured by $36n^2MK^2 < q/2$. □

Corollary (The multikey property)**.** *Let (f_1, h_1) and (f_2, h_2) be valid keys, $m_1, m_2 \in \{0,1\}$ and let $c_1 \leftarrow \mathsf{Enc}(h_1, m_1)$, $c_2 \leftarrow \mathsf{Enc}(h_2, m_2)$. Let $\tilde{f} \leftarrow f_1 \cdot f_2 \in R_q$. Then*

$$\mathsf{Dec}(\tilde{f}, c_1) = m_1,\ \mathsf{Dec}(\tilde{f}, c_2) = m_2$$

provided that K is small enough.

Proof: Apply 4.3.1 with $f = f_1$ and $\theta = f_2$ for the first equation and $f = f_2, \theta = f_1$ for the second. □

We can of course extend this facts to show that a highly composite key of the form $\tilde{f} = \prod_{i=1}^{l} f_i \in R_q$ can decrypt all messages decryptable by any of f_i: Just apply lemma 4.3.1 with $f = f_i$ and $\theta = \tilde{f}/f_i$, provided good *a priori* bounds: In fact $||\tilde{f}||_\infty \leq n^{l-1}K^l$, therefore decryption with this key is ensured by $n^{l-1}K^l \ll q$.

4.4 Hardness assumptions

The modified NTRU-FHE scheme semantic security is based on the celebrated *Ring Learning With Errors problem* (RLWE) and the new *Small Polynomial Ratio problem* (SPR). For the original modified NTRU parameters, the decisional SPR problem reduces to RLWE, but not a single homomorphic operation can be assured. A stretch of parameters is needed to overcome this, though it severely harms the statistic proofs of Stehlé and Steinfeld. The *DSPR assumption* states that the decisional SPR problem with stretched parameters is computationally hard. We adopt this same assumption, and in addition, we base the security of the honest-but-curious model on two problems that involve decomposing a polynomial into bounded factors. In the first, one wants to factorize a polynomial in R_q into two K-bounded polynomials, given the information that this is possible. In the second, one wants to extract a common factor of two polynomials such that the remaining factors are K-bounded. We first describe the DSPR assumption and then our *"Small Factors"* assumption.

4.4.1 Small Polynomial Ratio Problem, from [LATV12]

In [SS11b] Stehlé and Steinfeld based the security of the modified NTRU encryption scheme on the Ring Learning With Errors (RLWE) problem [LPR10]. They showed that the public-key $\mathsf{pk} = 2g \cdot \mathsf{sk}^{-1} \in R_q$ is *statistically close to uniform* over R_q, given that g and $f' = (\mathsf{sk}-1)/2$ were sampled using discrete Gaussians. Their results holds if (a) n is a power of 2, (b) $x^n + 1$ splits over n distinct factors over R_q (i.e. $q = 1 \mod 2n$) and (c) the Gaussian error distribution has standard deviation of at least $\text{poly}(n)\sqrt{q}$. However, these distributions seem too wide to support homomorphic operations in the NTRU-FHE scheme. To overcome this, authors in [LATV12] defined an additional assumption which states that if the Gaussian is contracted, it is still hard to distinguish between a public-key and a random element of R_q (even if the statistic-closeness result does not hold).

Definition 4.4.1 (DSPR Assumption)**.** *Let $q \in \mathbb{Z}$ be a prime integer and $\bar{\mathcal{G}}_K$ denote the K-bounded discrete Gaussian distribution over $R_0 = \mathbb{Z}[X]/(x^n + 1)$ as defined in 4.2.4. The decisional small polynomial ratio assumption says that it is hard to distinguish the following two distributions on R_q: (1) A polynomial $h = [2gf^{-1}]_q \in R_q$ where f', g were sampled with $\bar{\mathcal{G}}_K$ and $f = 2f' + 1$ is invertible over R_q, and (2) a polynomial $u \xleftarrow{R} R_q$ sampled uniformly at random.*

Finally, in a work by Bos et.al. [BLLN13], authors achieved to base the security on RLWE alone, alas achieving multikey FHE for a constant number of keys, a property inherent to any FHE scheme (as proved in the latest version of [LATV12]).

4.4.2 Small factorizations in the quotient ring

In addition to the RLWE and DSPR assumptions, we rely the semi-honest security on the hardness of the following problems. Let us define the distribution $\bar{\mathcal{G}}_K^\times$ which samples repeatedly from $\bar{\mathcal{G}}_K$ until the output is invertible over R_q.

Small Factors Problem: *Let $a, b \leftarrow \bar{\mathcal{G}}_K^\times$ and let $c(x) = a(x) \cdot b(x) \in R_q$. Find $a(x)$ and $b(x)$, given $c(x)$ and a test routine $T : R_q \to \{0,1\}$ that outputs 1 if the input is in $\{a, b\}$ and 0 otherwise.*

$\bar{\mathcal{G}}_K^\times$–GCD Problem: *Let $a, b \leftarrow \bar{\mathcal{G}}_K^\times$, and $y \xleftarrow{R} R_q$. Let $u(x) = a(x) \cdot y(x) \in R_q$ and $v(x) = b(x) \cdot y(x) \in R_q$. Find $a(x), b(x)$ and $y(x)$, given $u(x), v(x)$ and a test routine $T : R_q \to \{0,1\}$ that outputs 1 if the input is in $\{a, b, y\}$ and 0 otherwise.*

Proposition 4.4.1. *An algorithm solving the $\bar{\mathcal{G}}_K^\times$–GCD problem can be used to break the semantic security of the NTRU scheme.*

Proof: Given only a public-key of the form $\mathsf{pk} = [2ab^{-1}]_q$ where a is the secret-key, sample $p \xleftarrow{R} R_q$ and define $(u', v') = (ab^{-1}p, p)$. Define also $T : R_q \to \{0,1\}$ that for input $\alpha \in R_q$, samples random $r \xleftarrow{R} \{0,1\}$, checks if $\mathsf{Dec}(\alpha, \mathsf{Enc}(\mathsf{pk}, r)) \stackrel{?}{=} r$ and outputs 1 if α pass several such tests. Note that $u' = ay'$ and $v' = by'$ for $y = b'^{-1}p$, therefore seeding u', v', T to such algorithm outputs a, b, y'. □

Small Factors Assumption: *For the modified NTRU parameters, it is unfeasible to solve the small factors problem.*

In the absence of a formal proof, let us motivate the hardness of the small factors problem. The SF problem is equivalent to solve a quadratic system of equations over $\mathbb{F}_q$ with additional restrictions on the unknowns. Indeed, each coefficient of $c(x)$ is a quadratic form on coefficients of $a(x), b(x)$:

$$c_k = \sum_{i=0}^{n-1} a_i b_{k-i \bmod n} \cdot \sigma_k(i) \mod q,$$

where $\sigma_k(i) = +1$ if $i \leq k$ and -1 otherwise, the unknowns a_i, b_j follow a Gaussian distribution about 0 and are bounded in magnitude by K. As $K \ll q$, one can consider the equations over the integers. This results in a Diophantine quadratic system of n equations in $2n$ variables. Quadratic systems of m equations with n unknowns can be the Achilles heel for strong cryptographic primitives, as they can be attacked in the very overdetermined ($m \geq n(n-1)/2$) or very under-determined ($n \geq m(m+1)$) cases in fields with even characteristic. In [TW12], authors adapt an algorithm of Kipnis-Patarin-Goubin [KPG99] to odd characteristic fields and show a gradual change between the determined case ($m = n$, $\exp(m)$ runtime) and the massively under-determined case ($n \geq m(m+1)$, $\mathrm{poly}(n)$ runtime). According to

their analysis, our system ($n = 2m$) escapes the polynomial-time scope. Let us write the system in clear:

$$\forall i \in \{0, \dots, n-1\}, ||a_i||_\infty < K \text{ and } ||b_i||_\infty < K,$$

$$\begin{cases} c_0 = a_0b_0 - a_1b_{n-1} - a_2b_{n-2} - \dots - a_{n-1}b_1, \\ c_1 = a_0b_1 + a_1b_0 - a_2b_{n-1} - \dots - a_{n-1}b_2, \\ c_2 = a_0b_2 + a_1b_1 + a_2b_0 - \dots - a_{n-1}b_3, \\ \vdots \\ c_{n-1} = a_0b_{n-1} + a_1b_{n-2} + a_2b_{n-3} + \dots + a_{n-1}b_0. \end{cases}$$

As this is an under-determined system, the linearization $Z_{i,j} = a_ib_j$ results in a linear system with too many degrees of freedom to select the correct solution; this is not better than guessing in the initial quadratic system. On the other hand, this system presents cyclic anti-symmetry, which one could exploit to find a solution. However, it is not clear how to use the additional symmetry to make progress in finding solutions (this is also the case when trying to solve lattice problems in the particular case of ideal lattices).

From another point of view, if nothing was required from a, b, we are given an element c and a test routine $T : R \to \{0, 1\}$ that outputs 1 if the input is b and 0 otherwise (in our scenario, the test routine is to simply try out the extracted key $\beta = b$ via decryptions). An algorithmic issue arises again: There is a degree of freedom of one ring unit in the small factors problem, and an algorithm must exclude trivial factorizations of c: for instance, if nothing was required for a and b, the size of the candidates list for (a, b) is at least the number of units of R_q, since it contains all pairs $(a, b) = (cu, cu^{-1})$ for invertible $u \in R_q$. Using the K-boundedness of a, b, the list is to be reduced rejecting all incorrect pairs. To optimize up the rejection, we suggest a study of the distribution $\chi \stackrel{def}{=} (\bar{\mathcal{G}}_K)^{-1}$, which samples e according to $\bar{\mathcal{G}}_K$ and then outputs $e^{-1} \in R_q$ if e is invertible.

If a, b were sampled using another distribution, distinguishing $c = a \cdot b \in R_q$ from random may be achievable. For instance, if a, b coefficients were sampled with Bernoulli trials, we detected some linear correlation between the coefficients of c in this case. See appendix C.

4.5 Two-party multiplication protocols in R_q

In this section we introduce two protocols to jointly achieve multiplication in the quotient ring between two mutually distrusting parties. We distinguish two settings, the "secret inputs" (which is the classical MPC scenario) and the "shared inputs" which supposes that both parties have additive shares of some elements. The latter setting, however, can be regarded as a classical MPC computing a quadratic expression of the inputs.

4.5.1 Secret inputs setting

Alice and Bob hold $f \in R_q$ and $g \in R_q$ respectively. The following protocol allows them to multiply these elements: Alice will learn $fg+r \in R_q$ where r is a polynomial chosen by Bob. The reason of this is that if Alice learns fg, she can compute $g = fg/f$. The utility of this protocol may seem questionable, in the sense that it transfers Alice's obliviousness from g to r, nevertheless we will see that careful selection of r will allow the two parties to generate Alice's NTRU keys. This protocol is inspired on [AD01], where authors propose a protocol to compute scalar products as a building block to perform much more complex functionalities. It is detailed in algorithm 11.

Algorithm 11 Excalibur - TMP

Require: Alice holds $f \in R_q$, Bob holds $g \in R_q$. Let p, m be public integers.
Ensure: Alice learns $fg + r \in R_q$ where Bob knows $r \in R_q$

1: Alice generates m random polynomials $\{f_1, \ldots, f_m\} \overset{R}{\subset} R_q$ such that $\sum_{i=1}^{m} f_i = f$.
2: Bob generates m random polynomials $\{r_1, \ldots, r_m\} \overset{R}{\subset} R_q$ and $r \overset{def}{=} \sum_{i=1}^{m} r_i$.
3: **for** $i = 1, \ldots, m$ **do**
4: Alice generates a secret random number $k, 1 \leq k \leq p$.
5: Alice generates random polynomials $v_1, \cdots, v_p$, sets $v_k = f_i$, and send all these polynomials to Bob.
6: Bob computes the products and masks them: For all $j = 1, \ldots, p$ $z_{i,j} = v_j g + r_i$.
7: Alice extracts $z_{i,k} = f_i g + r_i$ from Bob with a 1–out–of–p OT protocol.
8: **end for**
9: Alice computes $\sum_{i=1}^{m} z_{i,k} = fg + r$.

Note that throughout the protocol, Bob always computed products of random polynomials, and to guess the value of f he has to perform $\approx p^m$ additions.

Lemma 4.5.1. *If it is not feasible to compute $O(p^m)$ additions in R_q, and if the RLWE assumption holds for $q, \phi(x) = x^n + 1$ and uniform χ over R_q,* TMP *securely outputs $fg + r$ to Alice and r to Bob in the presence of semi-honest parties.*

Proof: In this model, both parties follow exactly the protocol but try to learn as much information as possible from their transcript of the protocol. Let view_A, view_B be the collection of learned elements by Alice and Bob respectively. We have that view_B contains only polynomials $v_j^{(i)}$ indistinguishable from uniform (since they were sampled by semi-honest Alice), and these elements are independent from Bob's input, samplings, and computations. Therefore, to learn f, he needs to perform $\approx p^m$ additions. On the other hand Alice wants to learn g or r and she only has m pairs of the form $(f_i, f_i g + r_i)$ (and the output which is the component-wise sum of these), which by the RLWE assumption are indistinguishable from (f_i, u_i) for uniform u_i. In other words, the view of each adversary contains her input, her output, and a list

of polynomials indistinguishable from random by construction. We can construct simulators $\mathcal{S}_A, \mathcal{S}_B$ of protocol TMP for both parties, and it follows immediately that the views of Alice and Bob are indistinguishable from the simulators. □

Remark: If both parties are malicious but they do not want to leak their own inputs, at the end of the protocol they learn nothing about the other party's input.

This holds because Alice may deviate from the samplings, but she sends pm random elements computationally hiding f, Bob will process these pm elements (deviating as much as he wants from the actual required computation) and send m elements computationally hiding g and r to Alice via the OT protocol, thus Bob learns nothing. In this case, deviations from the protocol may cause the output to be incorrect. We do not worry much about this as soon as Bob's input is safe, since we will see that it will result in invalid keys for Alice and the honest party will know that the other is malicious.

4.5.2 Shared inputs setting

In this setting, two parties share two elements of R_q additively, and they want to compute shares of the product of these elements. Let Alice and Bob hold x_A, y_A and x_B, y_B respectively such that

$$x = x_A + x_B \quad \text{and} \quad y = y_A + y_B.$$

We propose a protocol $\mathsf{SharedTMP}$, at the end of which Alice and Bob will learn additive shares π_A, π_B respectively of the product:

$$\pi_A + \pi_B = xy \in R_q.$$

Algorithm 12 Excalibur - $\mathsf{SharedTMP}$

Require: Alice holds $(x_A, y_A) \in R_q^2$, Bob holds $(x_B, y_B) \in R_q^2$ such that $x = x_A + x_B$, $y = y_A + y_B$

Ensure: Alice learns $\pi_A \in R_q$, Bob learns $\pi_B \in R_q$ such that $\pi_A + \pi_B = xy$

1: Alice samples $r_A \xleftarrow{R} R_q$, Bob samples $r_B \xleftarrow{R} R_q$
2: Alice and Bob perform $\mathsf{TMP}(x_A, y_B)$ using Bob's randomness r_B, thus Alice learns $u_A = x_A y_B + r_B$ and Bob learns nothing.
3: Bob and Alice perform $\mathsf{TMP}(x_B, y_A)$ using Alice's randomness r_A, thus Bob learns $u_B = x_B y_A + r_A$ and Alice learns nothing.
4: Alice computes the share $\pi_A = x_A y_A + u_A - r_A \in R_q$
5: Bob computes the share $\pi_B = x_B y_B + u_B - r_B \in R_q$

Note that $\pi_A + \pi_B = (x_A + x_B)(y_A + y_B) = xy$. Since they only communicate in steps 2 and 3, security is reduced to two independent instances of the TMP protocol. We also have the following observation:

Lemma 4.5.2. *Let Alice and Bob perform* SharedTMP *on some non-trivial inputs, learning at the end π_A and π_B respectively. Even if Alice reveals π_A to Bob, he cannot deduce Alice's inputs.*

Proof: This follows directly from the randomness of $u_A - r_A$. □

4.6 Excalibur key generation

We present our main contribution, three protocols $\mathsf{Keygen}_{\mathsf{pk}}, \mathsf{Keygen}_{\mathsf{sk}}$ and a validation protocol, to be performed by Alice and Bob that will generate the public and the (blended) private-key of Alice, in that order. Let us first give an informal outline of the protocol. Bob has already generated his key-pair $(\beta, 2h\beta^{-1}) \in R_q \times R_q$. They want to compute a new key-pair $(\mathsf{sk}_A, \mathsf{pk}_A) = (\alpha\beta, 2g(\alpha\beta)^{-1}) \in R_q \times R_q$ for Alice, which correctly decrypts encryptions under pk_B since it contains the factor β.

- Excalibur generation of pk_A
  1. They share polynomials α, g, r of R_q additively, such that $\alpha = 1 \mod 2$.
  2. They perform SharedTMP to obtain shares of $\alpha r, gr$. Alice reveals her shares to Bob.
  3. Bob computes $2(gr) \cdot (\alpha r)^{-1} \cdot \beta^{-1} = 2g(\alpha\beta)^{-1}$ in R_q and broadcasts the result.
- Excalibur generation of sk_A (to be performed after publication of pk_A)
  1. Let $\alpha_A + \alpha_B = \alpha$ denote the same additive sharing of α than in the previous steps, where Alice holds α_A and Bob holds α_B. Alice and Bob perform TMP on entries α_A, β respectively, and Bob chooses $r = \alpha_B\beta$ as the randomness in the protocol.
  2. At the end of the protocol, Alice learns $\alpha_A\beta + r = \alpha\beta = \mathsf{sk}_A \in R_q$, and Bob learns nothing.
- Validation protocol : Alice and Bob run tests to be convinced that the keys are well formed and behave as claimed.

The protocols are described formally in algorithms 13, 14, 15 and 16.

If protocol 13 was carried out properly, a ciphertext encrypted with pk_A is correctly decrypted by any secret-key having the factor $\alpha\beta$ and reasonable coefficient size. Remark that in step 2, Bob received the element $z = \alpha \cdot r$: this does not allow to deduce a functional equivalent of the secret-key $\alpha\beta$, since r has large coefficients. Also, chances are overwhelmingly high that this element is in fact invertible in view of section 4.2.2.

Algorithm 13 Excalibur - $\mathsf{Keygen}_{\mathsf{pk}}$

Require: Bob already has his own key-pair $(\mathsf{sk}_B, \mathsf{pk}_B) = (\beta, 2h\beta^{-1}) \in R_q \times R_q$.

Ensure: A public-key for Alice pk_A

1: Alice and Bob sample random shares of elements in R_q:

   – Alice samples $s_A \leftarrow \bar{\mathcal{G}}_K, r_A \xleftarrow{R} R_q, g_A \leftarrow \bar{\mathcal{G}}_K$

   – Bob samples $s_B \leftarrow \bar{\mathcal{G}}_K, r_B \xleftarrow{R} R_q, g_B \leftarrow \bar{\mathcal{G}}_K$

   Let $\alpha = 2(s_A + s_B) + 1, r = r_A + r_B, g = g_A + g_B$ denote the shared elements.

2: Alice and Bob perform $\mathsf{SharedTMP}$ twice to obtain shares of $z = \alpha \cdot r$ and $w = g \cdot r$. Alice reveals her shares, thus Bob learns z, w.

3: Bob checks: If z is not invertible in R_q, restart the protocol.

4: Bob computes $2w(z\beta)^{-1} = 2g(\alpha\beta)^{-1}$ and publishes it as pk_A, along with a NIZK proof showing that z, w come from step 2 and that pk_A is well-formed.

5: Alice verifies Bob's proof. If it is not correct, abort the protocol.

Algorithm 14 Excalibur - $\mathsf{Keygen}_{\mathsf{sk}}$

Require: Bob's secret-key β and the same sharing of $\alpha = 2(s_A + s_B) + 1$ than in protocol 13.

Ensure: A secret-key for Alice $\mathsf{sk}_A = \alpha\beta$

1: Bob computes $r := (2s_B + 1)\beta \in R_q$

2: Alice and Bob perform the protocol $\mathsf{TMP}(2s_A, \beta)$, and Bob uses r as the random polynomial. At the end Alice knows $2s_A\beta + r = \alpha\beta \in R_q$.

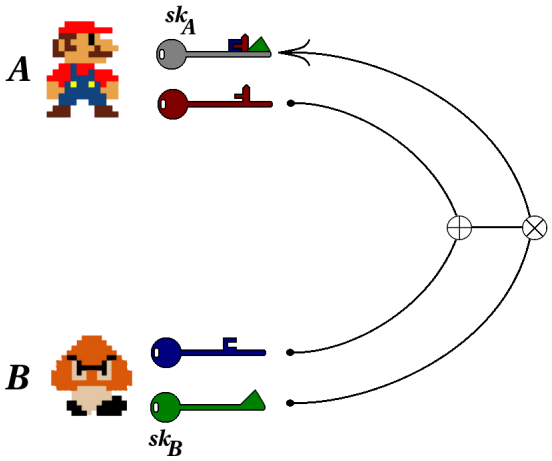

Figure 4.1: M. and G. perform Excalibur $\mathsf{Keygen}_{\mathsf{sk}}$

Once the keys are generated, they must pass a series of decryption and a well-formedness test. This is described in algorithms 15 and 16. First, Alice checks if her new secret-key works as expected, and then she convinces Bob, via a game of decryptions that she is indeed capable of decrypting ciphertexts encrypted under

pk_A and under pk_B. As we will see, this validation protocol avoids malicious activity.

Algorithm 15 Excalibur - Validation function (performed by Alice)

function VALIDATE($\mathsf{sk}_A, \mathsf{pk}_A, \mathsf{pk}_B$)
  for i from 1 to k **do**
    $\mu \xleftarrow{R} \{0,1\}$
    $\mu_1 \leftarrow \mathsf{Dec}(\mathsf{sk}_A, \mathsf{Enc}(\mathsf{pk}_A, \mu))$
    $\mu_2 \leftarrow \mathsf{Dec}(\mathsf{sk}_A, \mathsf{Enc}(\mathsf{pk}_B, \mu))$
    if $\mu_1 \neq \mu$ or $\mu_2 \neq \mu$ **then** output `reject`
    end if
  end for
  if $||\mathsf{sk}_A||_\infty > n(2K+1)^2$ or $||\mathsf{sk}_A \cdot \mathsf{pk}_B||_\infty > 2(2K+1)$ **then** output `size warning`
  end if
  output `accept`
end function

Algorithm 16 Excalibur - Validation protocol (performed by Alice and Bob)

Require: Alice holds $(\mathsf{sk}_A, \mathsf{pk}_A)$ and Bob holds $(\mathsf{sk}_B, \mathsf{pk}_B)$

1. Alice runs VALIDATE($\mathsf{sk}_A, \mathsf{pk}_A, \mathsf{pk}_B$). If the output is `reject`, abort.
2. Bob picks $2k$ random messages $(m_1^{(A)}, \ldots, m_k^{(A)})$ and $(m_1^{(B)}, \ldots, m_k^{(B)})$, and for each $i = 1, \ldots, k$ he computes ciphertexts $c_i^{(A)} = \mathsf{Enc}(\mathsf{pk}_A, m_i^{(A)}), c_i^{(B)} = \mathsf{Enc}(\mathsf{pk}_B, m_i^{(B)})$. He send all ciphertexts to Alice.
3. For each $i = 1, \ldots, k$, Alice compute $\mu_i^{(A)} = \mathsf{Dec}(\mathsf{sk}_A, c_i^{(A)}), \mu_i^{(B)} = \mathsf{Dec}(\mathsf{sk}_A, c_i^{(B)})$. She sends all plaintexts to Bob.
4. For each $i = 1, \ldots, k$, Bob checks if $\mu_i^{(A)} = m_i^{(A)}$ and $\mu_i^{(B)} = m_i^{(B)}$.

4.7 Security

We first discuss the honest-but-curious model, where the protocol is strictly followed but parties try to learn secrets. Then we look at the malicious model, where one party does not follow the protocol properly, in order to steal secrets or to sabotage the key generation.

4.7.1 Honest-but-curious model

In this model, we suppose that Alice and Bob follow exactly the instructions in algorithms 13, 14, 15 and 16 but they try to learn about each other's secret with all collected information.

Proposition: *If Alice is able to extract Bob's key from the protocol, she can solve the Small Factors Problem or the $\bar{\mathcal{G}}_K^\times$-GCD Problem. If Bob is able to extract Alice's key, he can solve the $\bar{\mathcal{G}}_K^\times$-GCD Problem.*

Proof: Let us focus first in Bob's chances on learning α (or a functional equivalent of the form $\theta \cdot \alpha$ for small $\theta \in R_q$). Recall that

$$\begin{cases} \mathsf{pk}_B = 2h\beta^{-1}, \\ \mathsf{pk}_A = 2g(\alpha\beta)^{-1}, \\ z = \alpha \cdot r, \\ w = g \cdot r, \\ \alpha = 2(s_A + s_B) + 1. \end{cases}$$

Let us focus on *Bob's view of the protocol:*

$$V = \{(s_B, r_B, g_B, z_B, w_B, z_A, w_A, \beta, \mathsf{pk}_B), \mathsf{pk}_A, \alpha \cdot r, g \cdot r\} \subset R_q.$$

What Bob is curious about: Any element of the set

$$U = \{\alpha, g, r, s_A, g_A, r_A\} \subset R_q.$$

The parentheses in V indicate that he sampled or received the elements contained, and the rest are results of joint computation. The knowledge of any element in U allows Bob to deduce Alice's secret-key α, and only the last three elements $\mathsf{pk}_A, \alpha \cdot r, g \cdot r$ of V depend on elements in U. Thus, extracting α is equivalent to solve the following system of equations in the unknowns $(X, Y, Z) = (\alpha, r, g)$:

$$\begin{cases} b_1 &= XY, \\ b_2 &= ZY, \\ b_3 &= ZX^{-1}, \end{cases}$$

where $b_1 = \alpha \cdot r, b_2 = g \cdot r, b_3 = \beta \mathsf{pk}_A/2$. We can eliminate the third equation noting that $b_1 b_3 = b_2$, and thus Bob faces the Small GCD Problem of section 4.4.2.

Let us now focus in Alice chances of learning Bob's secret.

Alice's view of the protocol: $W = \{(s_A, r_A, g_A, z_A, w_A), \mathsf{pk}_B, \mathsf{pk}_A, \alpha\beta\} \subset R_q$.
What Alice is curious about: Any element of the set
$Q = \{\alpha, \beta, h, s_B, \{w_B, z_B\}, \{w, z\}\}$.

First, extracting α or β directly from $\alpha \cdot \beta$ is exactly the small factors Problem. Using the only three sensitive elements of W, she faces the following system of equations in $(X, Y, Z) = (h, \beta, \alpha)$

$$\begin{cases} a_1 &= XY^{-1}, \\ a_2 &= (ZY)^{-1}, \\ a_3 &= ZY, \end{cases}$$

where $a_1 = \mathsf{pk}_B/2, a_2 = \mathsf{pk}_A/2g, a_3 = \alpha\beta$. After elimination of the third equation since $a_3 = a_2^{-1}$, Alice also faces the small GCD problem (actually, mapping $Y \mapsto Y^{-1}, Z \mapsto Z^{-1}$ yields to the same gcd problem faced by Bob). $\square$

4.7.2 Security against one malicious party

We consider the presence of one malicious adversary, a party that deviates as much as she wants from the protocol, but has a list of paramount objectives which she is not willing to sacrifice. We suppose that one of the two parties strictly follows the protocol and the other one is malicious, given the objectives below. We consider the presence of only one somewhat malicious adversary, given that both parties have concurrent objectives (for instance, Bob is trying to protect his key, and Alice to extract it from the protocols). In other words,

What curious Alice wants:

(A1) A functional secret-key sk_A associated with pk_A,

(A2) such that sk_A decrypts encryptions under pk_B,

(A3) protecting elements of $U = \{g_A, r_A, s_A\}$ from Bob and

(A4) to learn β.

What curious Bob wants:

(B1) To give Alice a functional secret-key sk_A associated with pk_A with decryption rights on $\mathsf{Enc}(\mathsf{pk}_B, \mathcal{M})$.

(B2) to protect elements of $Q = \{\beta, h, s_B, \{w_B, z_B\}\}$ from Alice,

(B3) (if malicious) overloading Alice's secret-key sk_A to have large coefficients, and

(B4) to learn α.

We will show that either the keys will be correctly generated or one party will not fulfill all of her objectives.

Malicious Alice, semi-honest Bob

Suppose that Bob is strictly following the protocol and Alice may deviate from the protocol but wants to fulfill (A1) to (A4). Let us summarize Alice's participation in the key generation:

1. *Samples* $s_A \leftarrow \bar{\mathcal{G}}_K, r_A \xleftarrow{R} R_q, g_A \leftarrow \bar{\mathcal{G}}_K$.

   Trivial samplings of these elements may ultimately leak α to Bob. For instance, if $s_A = 0$, $\alpha = 2s_B + 1$, if $r_B = 0$, $z/r_B = \alpha$, if $g_A = 0$ $g = w/g_B$. Also, if s_A or g_A have large coefficients, there is risk of $\bmod q$ wrap-around in the decryption procedure with sk_A. As she is sampling only shares of elements, she cannot force algebraic relations with them: regardless of her samples, α, g, r will remain indistinguishable from random.

2. *Participates in* $\mathsf{SharedTMP}((2s_A, r_A), (2s_B+1, r_B))$ *and learns* z_A*, participates in* $\mathsf{SharedTMP}((g_A, r_A), (g_B, r_B))$ *and learns* w_A*, then sends* z_A, w_A *to Bob.*

   As discussed in section 4.5, TMP and SharedTMP are secure if Bob is honest, in the sense that either Alice learns the correct output, or either she learns indistinguishable from random elements, but she learns nothing about Bob's input. She is limited to alter the inputs of both instances of SharedTMPand then giving wrong z_A or w_A to Bob. Nevertheless if she inputs different r_A's in both protocols or if she changes the values of z_A, w_A before sending them to Bob, from the linearity of shares and the randomness of Bob's entries it follows that this sabotages the relation $wz^{-1} = g\alpha^{-1}$, needed for correctness of decryption. In other words, in order to ensure (A1) and (A2), she is forced to maintain the input r_A for both instances of SharedTMPand send the correct output to Bob.

3. *Participates in* $\mathsf{TMP}(\{2s_A\}, \{2s_B + 1\})$ *and learns* $\alpha\beta$*.*

   If she uses the correct value of $2s_A$ (i.e. the same as in step 2), she learns the correct output $\alpha\beta$. If she inputs another value $x \neq 2s_A$, she does learn a functional equivalent of Bob secret (namely, $(x + 2s_B + 1)\beta$), but she is not able to decrypt encryptions under the already published pk_A, failing the verification procedure.

Malicious Bob, semi-honest Alice

Now suppose the inverse case, where Alice follows the protocol strictly and Bob is protecting β and guessing α, deviating as much as he wants from the protocol but fulfilling (B1) to (B4). We begin by saying that (B3) is unavoidable (unless the presence of a zero-knowledge proof that Bob's polynomials are of the right size), but Alice can tell if Bob overloaded the secret-key $\alpha\beta$ simply looking at the coefficients. Let us now summarize Bob's participation in the key generation:

1. *Samples* $s_B \leftarrow \bar{\mathcal{G}}_K, r_B \xleftarrow{R} R_q, g_B \leftarrow \bar{\mathcal{G}}_K$*.*

   Trivial sampling may compromise sensible elements as before. He must ensure the randomness of α, r if he wants to protect these elements, and on the other hand Alice will know if he deviates from a K-bounded sampling (just looking at the coefficients in $\alpha \cdot \beta$. Therefore, he gains nothing in deviating from a K-bounded sampling.

2. *Participates in* $\mathsf{SharedTMP}((2s_A, r_A), (2s_B+1, r_B))$ *and learns* z_B*, participates in* $\mathsf{SharedTMP}((g_A, r_A), (g_B, r_B))$ *and learns* w_B*.*

   As noted in section 4.5, because of Alice's randomness in SharedTMP, either Bob obeys the protocol and receives the correct outputs, either he deviates

and receives random outputs, from which he cannot deduce secret values and which sabotage key generation. Also, if he uses different r_B's in both instances, Alice will not be able to decrypt since the decryption relation $wz^{-1} = g\alpha^{-1}$ is not fulfilled (and he remains oblivious of r_A, not being able to force this relation). Hence, he is forced to follow $\mathsf{SharedTMP}$ and use the same r_B in both instances if he wants to fulfill (B1).

3. *Receives z_A, w_A, learning z, w. Checks if z is invertible and publishes $\mathsf{pk}_A = 2w(z\beta)^{-1}$. He then participates in $\mathsf{TMP}(\{2s_A\}, \{2s_B+1\})$, chooses $R = (2s_B+1)\beta$ and learns nothing.*

   Suppose that he published pk'_A as Alice's public-key and participated in the TMP instance with generic values, indicated by an apostrophe. At the end, Alice knows pk'_A and sk'_A. She will run the validate function of algorithm 15 to check (i) if sk'_A has the expected coefficient size, (ii) if $\mathsf{sk}'_A \cdot \mathsf{pk}'_A = 2g'$ for a vector $g' \in \chi$ and (iii) if she is able to decrypt encryptions of messages under pk'_A and pk_B. If she is indeed able to decrypt encryptions under pk_B, then sk'_A contains the factor β, thus by randomness of s_A, $\beta' = \theta\beta$ and $R' = \omega\beta$ for small θ and ω. Also, as long as the polynomials sk'_A and $\mathsf{sk}'_A \cdot \mathsf{pk}'_A$ are of the right form, she does not care about how Bob computed pk'_A, as decryption of encryptions under pk'_A work as claimed. If on the contrary a single decryption fails or if sk'_A or $\mathsf{sk}'_A \cdot \mathsf{pk}'_A$ have large coefficients, she can claim one of the following Bob's wrongdoings: Either he did not include β, either he included $\theta\beta$ for too large θ, either he sabotaged entirely the key generation in a change of input or inside a multiparty multiplication protocol. This allows to conclude that if Bob fails to give what is expected, the output keys will be rejected by Alice, who discovers Bob's maliciousness after the validation protocol 16.

We should point out another strategy that Bob could maliciously try. When generating Alice's secret-key, he could simply ignore Alice's input share $2s_A$, and thus the protocol gives Alice the key $\mathsf{sk}'_A = \alpha'\beta$, for an $\alpha' \in R_q^\times$ of Bob's choice. Bob, who received no output from the protocol, can reconstruct this key, thus gaining Alice's secret. However, this key will be rejected by Alice since it cannot be associated with the previously generated $\mathsf{pk}_A = 2g(\alpha\beta)^{-1}$. To avoid this rejection, Bob should have published $\mathsf{pk}'_A = 2g'(\alpha'\beta)^{-1}$ instead, but it is easy to see that this publication would contradict the NIZK proof of step 4 of algorithm 13: Because of the way $\mathsf{SharedTMP}$ works, Bob has no way of choosing α' of his choice in the expression $z = \alpha r$. In view of this, passing the validation protocol with such a key is overwhelmingly unlikely.

4.8 Extensions

4.8.1 Chains of keys

Suppose Alice and Bob perform the latter protocols, such that Alice has now a private-key of the form $\mathsf{sk}_A = \alpha\beta$ where β is Bob's secret-key. Alice can repeat the protocol with a third user Charlie (with slight coefficients size modifications at the validation protocol), who at the end receives a pair of keys of the form $(\mathsf{sk}_C, \mathsf{pk}_C) = (\alpha\beta\gamma, 2g_C(\alpha\beta\gamma)^{-1})$. As his secret-key contains the factors β and $\alpha\beta$, he can decrypt both Bob's and Alice's ciphertexts. This shows that easy modifications to the protocol allows to generate a chain of users, each one inheriting the previous user decryption rights. From corollary 4.3.1, it is easy to see that the length of such a chain is at most $\approx \log(q/nK)$ to ensure decryptions (this matches the maximum number of keys on the multikey LATV FHE scheme for the same parameters). We point out that intersecting chains are also possible, meaning that a user can glue her secret-keys to two or more upper-level users and even if they collude they are not able to extract his key. This comes from an easy generalization of our $\bar{\mathcal{G}}_K^{\times}$–GCD problem.

4.8.2 Plugging in LATV-FHE

Because of the form of an Excalibur key, i.e. $(\mathsf{sk}, \mathsf{pk}) = (\prod_{i=1}^{r} \alpha_i, 2g \prod_{i=1}^{r} \alpha_i^{-1})$, the inclusion of our protocols into the Multikey FHE scheme from [LATV12] is immediate. The only missing element are the evaluation keys, which can be generated easily by the secret-key holder after the (Excalibur) key generation: they are "pseudo-encryptions" of the secret-key sk under the public-key pk. This achieves a somewhat homomorphic encryption scheme in the chain of users, where in addition they can combine ciphertexts generated by any public-key.

4.9 Conclusion

In this Chapter, we proposed a new protocol to generate NTRU keys with additional decryption rights, allowing to form a hierarchic chain of users. We motivated such a procedure because it avoids betrayal naturally, and since it applies to the FHE-NTRU scheme, it may contribute to clear the bootstrapping-like re-encryption paradigm, since it is to our knowledge the first FHE scenario featuring (the pure definition of) proxy re-encryption. In this light, it concurs with other proxy re-encryption schemes, as, while being rigid, ciphertext transformation is no necessary at all, since decryption rights are defined in key-generation time. We used two-party computation protocols as building blocks, and relied the semantic security on the well-known RLWE and DSPR assumptions, and security in presence of semi-honest parties on a hardness assumption in cyclotomic polynomial rings.

Chapter 5

The promising BGV scheme: an implementation

Contents

5.1 Introduction 101
5.2 The scheme 101
5.2.1 Parameters 101
5.2.2 Key Generation – sampling from $\mathbb{Z}_p[x]/(x^N + 1)$ 102
5.2.3 Encryption and decryption 103
5.2.4 Relinearization and Modulus switching 103
5.2.5 BGV Homomorphic operations 105
5.3 Implementation 107
5.3.1 Math Layers 107
5.3.2 Cryptography layers 109
5.4 Performance 112
5.5 Conclusion 112

5.1 Introduction

The Brakerski-Gentry-Vaikuntanathan scheme has been praised as the most auspicious in terms of efficiency amongst other homomorphic schemes. Its structure allows to use the corresponding Galois group in order to pack ciphertexts and perform parallel operations in the style of SIMD techniques.

In this Chapter we provide a description and a working implementation of the Brakerski-Gentry-Vaikuntanathan homomorphic scheme, using `C++` with the multiple precision library `GMP` and additional support to complex numbers. In order to handle ciphertext noise, we use the scale and modulus switching techniques. This chapter is motivated by the article [CS16], where authors compare ring-based homomorphic schemes. In an appendix, they list security parameters; we follow these suggestions in our implementation.

The well-known HElib "Homomorphic Encryption library" (`https://github.com/shaih/HElib`) also implements BGV homomorphic scheme. Of course, this implementation with highly optimized routines outperforms ours.

5.2 The scheme

We present here a full description of the BGV homomorphic scheme. First let us precise the ring framework.

5.2.1 Parameters

Consider the following parameters and plaintext/ciphertext spaces.

N	–	A power of 2.
R	–	The cyclotomic ring $\mathbb{Z}[x]/(x^N+1)$.
L	–	The desired homomorphic depth.
$p \in \mathbb{N}$	–	The plaintext modulus.
R_p	–	The plaintext space $\mathbb{Z}_p[x]/(x^N+1)$
$p_0, \ldots, p_{L-1}$	–	L primes s.t. $p_i \equiv 1 \mod 2N$
$q_0, \ldots, q_{L-1}$	–	The "chain of moduli". For k from 0 to $L-1$, $q_k = \prod_{j=0}^{k} p_j$.
R_q	–	The polynomial ring $\mathbb{Z}_q[x]/(x^N+1)$, for any q. This is the chain of ciphertext spaces.

For our implementations, $p \in \{2, 101, 2^{32}\}$ (binary, small alphabet and 32-bit plaintext space). In the scheme, a message must be encoded into one or several plaintexts $m \in R_p \simeq (-p/2, p/2]^N$. Before defining decryption, let us slightly jump ahead. Fresh ciphertexts are elements of R_{q_L}, and upon multiplication, they jump to smaller ciphertext spaces. This way, the scheme operates in $L-1$ levels of

"decreasing" polynomial cyclotomic rings as follows :

$$\textit{'Fresh' ciphertext space} = R_{q_{L-1}} \longrightarrow R_{q_{L-2}} \longrightarrow \cdots \longrightarrow R_{q_1} \longrightarrow R_{q_0}.$$

Ciphertexts in the last ring cannot be used to multiply, however *they can still be used in homomorphic additions*. This chain of moduli allows therefore to perform $L-1$ multiplications.

More precisely, ciphertexts consist in a pair of polynomials in these spaces, along with two tags, one indicating the level of this ciphertext and another one representing an approximation of the noise. See 5.2.3 below.

An important remark is that the primes $p_0, \ldots, p_{L-1}$ are chosen to satisfy $p_i \equiv 1 \mod 2N$, implying the existence of a primitive $2N$-th root of unity in $\mathbb{Z}_{p_i}$ (this will prove helpful in order to compute number theoretic transform of polynomials). As suggested in [CS16], we require in addition that $p_i \equiv 1 \mod p$, in order to simplify the scaling operation (and slightly control noise growth).

5.2.2 Key Generation – sampling from $\mathbb{Z}_p[x]/(x^N+1)$

Toward defining key generation, four random samplings of elements in R_q are defined:

$\mathcal{U}_q$	–	The uniform distribution on $R_q \simeq \{-q/2, \ldots, q/2-1\}^N$.
$\mathcal{HWT}(h)$	–	Samples a $\mathbb{Z}^N$ vector with h non-zero entries in $\{-1, +1\}$.
$\mathcal{ZO}(\rho)$	–	Samples a $\mathbb{Z}^N$ vector, where each coordinate is 1 or -1 with probability $\rho/2$ each and is 0 with probability $1-\rho$.
$\mathcal{DG}_q(\sigma^2)$	–	Let $\mathcal{N}(0,\sigma^2)$ denote the Gaussian distribution on real numbers, with zero-mean and variance σ^2. $\mathcal{DG}_q(\sigma^2)$ draws a real vector from $\mathcal{N}(0,\sigma^2)^N$, rounds it to the nearest integer vector, and outputs the result modulo q.

With this, the key generation procedure outputs a sparse polynomial as the secret-key and a pair of uniform-looking elements of $R_{q_{L-1}}$ as the public-key. See algorithm 17.

Algorithm 17 BGV - Key Generation

1: **function** KeyGen(1^λ)
2: Define sk, e, a, $b \in R$
3: $\mathsf{sk} \leftarrow \mathcal{HWT}(h)$ ▷ $h = 64$ is suggested in [CS16].
4: $a \leftarrow \mathcal{U}_{q_{L-1}}$.
5: $e \leftarrow \mathcal{DG}(\sigma^2)$ ▷ $\sigma^2 = 1.3$ is suggested in [CS16].
6: Set $b \leftarrow [a \cdot \mathsf{sk} + pe]_{q_{L-1}}$
7: Output secret key sk and public key $\mathsf{pk} = (a, b)$
8: **end function**

Keys are therefore of the form $(s, (a, b)) \in R_p \times R_{q_{L-1}}^2$ such that $b - sa$ is a polynomial with small coefficients and equivalent to 0 modulo p.

5.2.3 Encryption and decryption

Ciphertexts are of the form $((c_0, c_1), \nu, t) \in R^2 \times \mathbb{R} \times [0, L]$: (c_0, c_1) is the ring gibberish, ν is a bound for the noise of the ciphertext and t indicates the multiplication level. Therefore, $(c_0, c_1) \in R_{q_t}$. To encrypt a message $m \in R_p$, do as described in algorithm 18.

Algorithm 18 BGV - Encryption and Decryption

function ENC($\mathsf{pk}, m \in R_p$)
  Define $v, e_0, e_1, c_0, c_1 \in R$
  $v \leftarrow \mathcal{ZO}(\rho)$ ▷ ρ is not important, 0.5 is suggested
  $e_0 \leftarrow \mathcal{DG}_{q_{L-1}}(\sigma^2)$
  $e_1 \leftarrow \mathcal{DG}_{q_{L-1}}(\sigma^2)$
  $c_0 \leftarrow [b \cdot v + p \cdot e_0 + m]_{q_{L-1}}$
  $c_1 \leftarrow [a \cdot v + p \cdot e_1]_{q_{L-1}}$
  $\mathbf{c} = (c_0, c_1, L-1, B_{\text{clean}})$ ▷ B_{clean} is a noise norm upper bound for correct decryption
  Output $\mathbf{c}$
end function

---

function DEC($\mathsf{sk}, \mathbf{c} \in R^2 \times \mathbb{R} \times \mathbb{N}$)
  Parse $\mathbf{c} = (c_0, c_1, t, \nu)$
  Set $m' \leftarrow [c_0 - c_1 \cdot \mathsf{sk}]_{q_t}$ ▷ A "verify_noise" procedure may be performed here
  Output $m' \mod p$
end function

5.2.4 Relinearization and Modulus switching

It is possible to define multiplication of ciphertexts now, but for correct decryption this would need powers of the secret key. Let us explain this, consider two fresh ciphertexts encrypting m, m' respectively under the same public key:

$$\begin{aligned} \mathbf{c} &= (c_0, c_1, \nu, t) \text{ such that } [c_0 - c_1 \cdot \mathsf{sk}]_{q_t} = m \mod p, \\ \mathbf{c}' &= (c_0', c_1', \nu, t) \text{ such that } [c_0' - c_1' \cdot \mathsf{sk}]_{q_t} = m' \mod p. \end{aligned}$$

Then it is clear that the "extended ciphertext" D defined as follows

$$D = (d_0, d_1, d_2, \nu_D, t_D) \text{ such that } \begin{cases} d_0 = [c_0 \cdot c_0']_{q_t} \\ d_1 = [c_0 c_1' + c_1 c_0']_{q_t} \\ d_2 = [c_1 \cdot c_1]_{q_t} \end{cases}$$

verifies the "extended decryption equation"

$$[d_0 + d_1\mathsf{sk} + d_2\mathsf{sk}^2]_{q_t} = m \cdot m' \mod p.$$

This indeed defines ciphertext multiplication, but fails to meet the required FHE compactness as ciphertexts (and decryption circuits) grow after each product.

To overcome this, authors in [BGV12] propose a relinearization procedure. This needs the publishing of circular ciphertexts, somehow encrypting sk^2. This public information looks a list of public keys, but for whom the attribute b is not exactly a valid ciphertext, but "almost" encryptions of the new key "$\mathsf{sk} \cdot \mathsf{sk}$" in base-$T$ decomposition for some T. To see how to generate these lists, see algorithm 19.

Algorithm 19 BGV - Key-Switching Procedure

procedure SwitchKeyGen(sk,sk')
  for $i = 0$ to $\theta - 1$ **do** ▷ Let $\theta = \lceil \log_T(q_{L-1}) \rceil$
    $a_i \leftarrow \mathcal{U}_{q_{L-1}}$
    $e \leftarrow \mathcal{DG}_{q_{L-1}}(\sigma^2)$
    $b_i \leftarrow [a_i \cdot \mathsf{sk} + p \cdot e_i + T^i \cdot \mathsf{sk}']_{q_{L-1}}$
  end for
  $swk \leftarrow \{a_i, b_i\}_{i=0}^{\theta-1}$
  Output swk
end procedure

The following is the first variant proposed in [CS16]. It is the last step in ciphertext multiplication. The polynomials d_0, d_1, d_2 are three ring elements containing the coefficients of $1, \mathsf{sk}, \mathsf{sk}^2$ in the product $(c_0 - \mathsf{sk}c_1)(c'_0 - \mathsf{sk}c'_1)$ for ciphertexts (c_0, c_1, etc) and (c'_0, c'_1, etc), while t, ν are level and noise tags.

---

procedure SwitchKey($swk, d_0 \in R, d_1 \in R, d_2 \in R, t, \nu$)
  Write d_2 component-wise in base T: $d_2 = d_{2,0}, d_{2,1}, \ldots, d_{2,\theta-1}$
  $c_0 \leftarrow d_0 + \sum d_{2,i} \cdot b_i$
  $c_1 \leftarrow d_1 + \sum d_{2,i} \cdot a_i$
  $\nu' = \nu + B(t)$
  Output ciphertext $\mathbf{c} = (c_0, c_1, t, \nu')$ ▷ SwitchKey does not modify the level t.
end procedure

The additive noise depends on the level:

$$B(t) = \frac{8}{\sqrt{3}} p\sigma NT\lceil \log_T q_t \rceil$$

There is a common confusion between this operation, called *Key Switching* and Modulus Switching. They are independent: modulus switching is a noise management technique, based on the fact that two ciphertexts can be just multiplied, or

they can be translated into another ciphertext space via scaling and then be multiplied, with smaller error growth (see the latter equation). In our implementation, we adopt the scaling variant proposed in [CS16], in which an auxiliary modulus Q is introduced. This operation aims to reduce noise/modulus ratio, and as a positive side effect it reduces ciphertext sizes. This implies that, somehow counter-intuitively, *multiplications near the end of the circuit take considerably less time.*

5.2.5 BGV Homomorphic operations

With these procedures we are ready to describe how to perform homomorphic operations. See algorithms 20 and 21.

Algorithm 20 BGV - Homomorphic Operations

procedure AddCipher(`cipher` **a**,`cipher` **b**)
  $t \leftarrow \min(\mathbf{a}.t, \mathbf{b}.t)$
  a ←`ReduceLevel`$(\mathbf{a}, t)$
  b ←`ReduceLevel`$(\mathbf{b}, t)$
  Define **c**
  $\mathbf{c}.c_0 \leftarrow \mathbf{a}.c_0 + \mathbf{b}.c_0 \mod q_t$
  $\mathbf{c}.c_1 \leftarrow \mathbf{a}.c_1 + \mathbf{b}.c_1 \mod q_t$
  $\mathbf{c}.t \leftarrow t$
  $\mathbf{c}.\nu \leftarrow \mathbf{a}.\nu + \mathbf{b}.\nu$ ▷ Noise grows linearly
  Output **c**
end procedure

---

procedure MulCipher(`cipher` **a**,`cipher` **b**, `switchingkey` swk)
  $t \leftarrow \min(\mathbf{a}.t, \mathbf{b}.t)$
  a ←`ReduceLevel`$(\mathbf{a}, t)$
  b ←`ReduceLevel`$(\mathbf{b}, t)$
  Define $\mathbf{c}, d_0, d_1, d_2$
  $d_0 \leftarrow \mathbf{a}.c_0 \cdot \mathbf{b}.c_0$
  $d_1 \leftarrow \mathbf{a}.c_0 \cdot \mathbf{b}.c_1 + \mathbf{a}.c_1 \cdot \mathbf{b}.c_0$
  $\nu \leftarrow \mathbf{a}.\nu \cdot \mathbf{b}.\nu$ ▷ Noise grows exponentially
  c ←`SwitchKey`$(swk, d_0, d_1, d_2, t, \nu)$
  c ←`ReduceLevel`$(\mathbf{c}, t-1)$ ▷ Reducing modulus (next product will be faster)
  Output **c**
end procedure

Algorithm 21 BGV - Scaling ciphertexts

procedure ScaleVariant(cipher $\mathbf{c}, Q \in \mathbb{N}$) ▷ This is the variant proposed in [CS16]
Require: $Q | q_{\mathbf{c}.t}$, $P \equiv 1 \mod p$.
    Parse $\mathbf{c} = (c_0, c_1, t, \nu)$
    Let $P = Q/q_t$
    For $i = 1, 2$ compute small δ_i such that $\begin{cases} \delta_i = -c_i \mod P \\ \delta_i = 0 \mod p \end{cases}$
    Define $c'_0 \leftarrow (c_0 + \delta_0)/P$, $c'_1 \leftarrow (c_1 + \delta_1)/P$
    and $\nu' \leftarrow \nu/P + B_{\text{scale}}$ ▷ B_{scale} is an additive global constant.
    Output ciphertext $\mathbf{c}' \leftarrow (c'_0, c'_1, t, \nu')$
end procedure

---

procedure Scale(cipher $\mathbf{c}, t_{out} \in \mathbb{N}$)
    Define $\mathbf{c}' \leftarrow \mathbf{c}$
    $\mathbf{c}'.t \leftarrow t_{out}$ ▷ $\mathbf{c}'$ is a clone of $\mathbf{c}$, but assigned to the new level
    Output `ScaleVariant(`$\mathbf{c}', q_{\mathbf{c}.t}$`)`
end procedure ▷ (Remark that $q_{\mathbf{c}.t}$ satisfies both requirements of `ScaleVariant`)

---

procedure ReduceLevel(cipher $\mathbf{c}, t' \in \mathbb{N}$)
    if $\mathbf{c}.t \leq t'$ **then** return $\mathbf{c}$
    end if
    if $\mathbf{c}.\nu > B$ **then** $\mathbf{c} \leftarrow$`Scale(`$\mathbf{c}, t'$`)`
    else
        $c'_0 \leftarrow c_0 \mod q_{t'}$
        $c'_1 \leftarrow c_1 \mod q_{t'}$
        $\mathbf{c} \leftarrow (c'_0, c'_1, t', \nu)$
        Output $\mathbf{c}'$
    end if
end procedure

5.3 Implementation

We programmed the above procedures in `C++` with the `gmp` large numbers library. As figure 5.3 display, we implemented various classes. We distinguish the cryptography layer (two top floors) and the ring arithmetic layer (first and second floors).

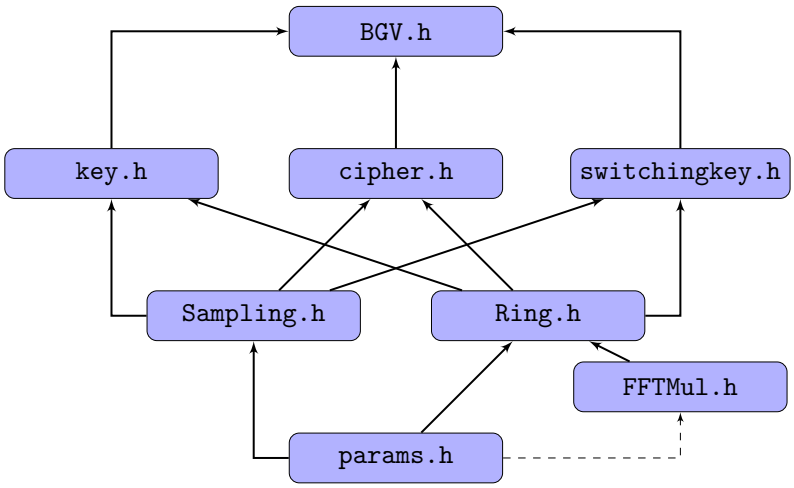

Figure 5.1: BGV implementation: `C++` Class dependencies

5.3.1 Math Layers

The basic bricks aim to support necessary operations in the polynomial ring R_q. We define N as a (global, constant) power of 2 and $M = 2N$, but use the modulus q as an argument to our functions, since it decreases from one level to another. We propose four bricks `params.h, Ring.h, Sampling.h` and `FFT.h`.

Overall parameters

The most basic layer contains all parameters for the implementation. Even if bricks are portable, we regroup all parameters in `params.h` for testing different sets thereof. This way, to instance the first set of parameters provided in [CS16], we set

```
//in params.h
#define N 1024
#define h 64
#define L 2
#define p 2                          //plaintext space iz R_p
#define bitsp 14                     //number of bits of the
                                     first prime
```

```
#define pdiff 16                        //measures the distance
                                        in bits between the
                                        p[i]'s
extern mpz_class primes[L],chain[L];    // Chain of moduli
#define sigma 1.3                       //variance of the noise
                                        distribution
#define bitsT 26
#define rho 0.5                         //See "ZO" distribution
extern double Bclean,Bscale,B;          //These are BGV
                                        constants
extern mpz_class T;                     //Base for decomposition
extern mpf_class root;                  //Root of unity for FFT
#define bitsCoeff bitsp*L
#define PREC bitsCoeff*5
```

The `gmp` parameters cannot be set globally, this is why we include a function `SetParameters()` in the `BGV.h` file that sets all remaining parameters.

Ring of polynomials

The file `Ring.h` provides all necessary functions on polynomials: addition, component-wise reduction modulo q, component-wise decomposition in base T (for key switching), prime numbers generator of a given number of bits and congruent to 1 modulo $2N$, and fast multiplication in the ring (via the number theoretic transform).

Sampling

The file `Sampling.h` includes four different samplings over R_q:

```
//in Sampling.h
void Uniform(mpz_class vec[N],mpz_class q)  //We inherit the
                                            uniform distribution
                                            from gmp random
                                            generators with
                                            appropriate seeds.
void HWT(mpz_class v[N])                    //See 5.2.2
void ZO(mpz_class v[N])                     //See 5.2.2.
void DG(mpz_class v[N])                     //Approximated
                                            with Irwin-Hall
                                            distributions.
```

To approximate the normal distribution, we use the Irwin-Hall distribution, de-

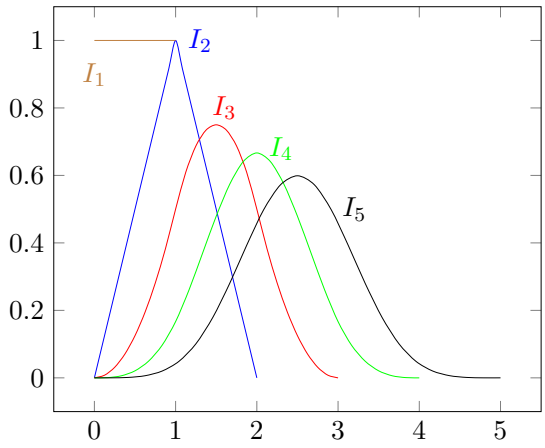

Figure 5.2: First five Irwin-Hall probability density functions.

fined as the sum of k independent, uniformly distributed random variables in $(0, 1)$. The mean and variance of an Irwin-Hall distribution are $k/2$ and $k/12$ respectively, and the probability density function approaches the normal p.d.f. as k grows. We used a value of k according to the desired variance σ^2, i.e. $k = 12 \cdot \sigma^2$, with proper scaling and translating.

Fast polynomial multiplication

In `FFTMul.h`, we implement Cooley-Tukey's recursive algorithm to compute the FFT's of polynomials, in time $O(N \log N)$. The complex numbers are defined with the `mpc.h` library, and we fix a precision in bits in `params.h`, which grows with the input size of coefficients. See algorithm 22.

The *Free* statements in the algorithm are advised in order to avoid memory allocation problems. To soften the butterfly, we omitted the product $\omega \cdot \omega_n$, instead we precomputed all $2N-$th roots of unity and choose ω using the relation $\omega_n^k = \omega_{2N}^{2Nk/n}$.

5.3.2 Cryptography layers

Key object

The class `key.h` has three attributes and three member functions:

```
//in key.h
mpz_class sk[N]              //The secret key
mpz_class a[N],b[N]          //The public key pair (a,b)
void key::Print();
void key::KeyGen();          //Generates a key-pair
```

```
void key::Verify():          //This tests the relation (b-a.sk mod
                             p)==0, for correct decryption
```

Cipher

The class `cipher.h` links to `Ring.h` and `params.h`, and it has four attributes:

```
//in cipher.h
mpz_class c0[N],c1[N]        //Contains ring gibberish
mpz_class nu                 //An upper bound on the noise (set to
                             Bclean in a fresh encryption)
int t                        //The level of a ciphertext, set to
                             L-1 initially and decreasing with
                             multiplication.  If t=0, ciphertext
                             is no longer usable in products.
```

Switching-key object

In order to relinearize an already evaluated ciphertext, for which decryption would normally need powers of the secret key, some information about the Key Switching procedure must be published (see 5.2.4). We decided to create an independent class `switchingkey.h` to contain this information, rather than including this into the `key.h` class. The reason for this is that this special key is the largest object of the scheme and its handling can be certainly optimized. It contains two attributes

```
//in switchingkey.h
mpz_class a[theta][N];
mpz_class b[theta][N];                  //We introduced
                                        the parameter
                                        theta=log_T(q_L-1) for
                                        memory allocation.
void switchingkey::Print();
void switchingkey::Verify();            //Verify the correctness
                                        of semi-encryptions, see
                                        5.2.4.
```

The BGV scheme

Linking to all previous classes, the `BGV.h` is at user-end and contains the cryptographic functions, as key-generation, encryption, decryption, homomorphic evaluation and some test routines;

Algorithm 22 Recursive FFT

1: **procedure** RecFFT$(n, (a[0], \cdots a[n-1]), (y[0], \cdots y[n-1]))$ ▷ Puts the FFT of $\vec{a}$ in $\vec{y}$.

Require: Primitive n-th root of unity ω_n

2: **if** $n = 1$ **then**
3: $y[0] \leftarrow a[0]$
4: **return**
5: **end if**
6: $\vec{a}_0 \leftarrow (a[0], a[2], \cdots, a[n-2])$ ▷ Even part of a
7: $\vec{a}_1 \leftarrow (a[1], a[3], \cdots, a[n-1])$ ▷ Odd part of a
8: RecFFT$(n/2, a_0, y_0)$
9: RecFFT$(n/2, a_1, y_1)$
10: (Free a_0, a_1)
11: $\omega \leftarrow 1$
12: **for** $k = 0$ to $n/2 - 1$ **do** ▷ Butterfly
13: $y[k] \leftarrow y_0[k] + \omega y_1[k]$
14: $y[k + n/2] \leftarrow y_0[k] - \omega y_1[k]$
15: $\omega \leftarrow \omega \cdot \omega_n$
16: **end for**
17: (Free y_0, y_1, ω)
18: **end procedure**

```
//in BGV.h
void PrepareParams();                          //Parameter computations
void KillParams();

cipher Encrypt(int message,key k);             //Encryption
mpz_class Decrypt(cipher c,key k);             //Decryption

//Bricks for homomorphic multiplication:

cipher ScaleVariant(cipher c,mpz_class Q);     //See 5.2.4
cipher Scale(cipher c,int t);
cipher ReduceLevel(cipher c, int t);
switchingkey SwitchKeyGen(key k1, key k2);
cipher SwitchKey (switchingkey sw,mpz_class
d0[N],mpz_class d1[N],mpz_class d2[N],int
t,mpz_class nu);

//Homomorphic operations:
```

```
cipher AddCipher(cipher c1,cipher c2);
cipher MulCipher(cipher ca,cipher
cb,switchingkey sw);

//Timings:

void ChronoBGV();                          //Displays performances in
                                           machine and user seconds
                                           of all functions and
                                           evaluation of a depth-L
                                           circuit.
```

5.4 Performance

We selected sets of parameters from [CS16] and performed an evaluation of different depth-L circuits. This tests were run on a desktop computer running at 2.70 GHz. A consideration about the choice of the degree N:

For each suggested value of N, we chose the smallest power of 2 greater than the given value. Since our polynomial product runs in $O(N \log N)$, allowing smaller values of N can fasten the algorithms in a factor of at most 2. In view of this, we have included the factor $\alpha =$ (value of N)/(suggested value of N).

5.5 Conclusion

In this Chapter we described our implementation of the BGV fully homomorphic scheme. We correctly evaluated circuits up to depth 30, and we acknowledge that other implementations (such as HElib) outperform ours by far. In contrast, we did not analyze the mean timings with respect to packs of ciphertexts or depth level: We only provided timings of the worst case (i.e. fresh ciphertext multiplication with no packing). However, with this analysis and considering optimization of our functions, our timings should be divided by a factor at least the number of packed ciphertexts times the ratio of $l = 1$ and $l = L$ multiplication timings, giving an acceleration factor of about 10^3 for homomorphic product.

Depth	Plaintext space	FFT Mult	KeyGen	Switching KeyGen	Encryption	Decryption	Homomorphic product (fresh, non-packed)	$\alpha = N/(\text{suggested } N)$
$L = 2$	$\{0, 1\}$	0.047	0.045	0.22	0.097	0.044	0.55	1.29
	$\mathbb{Z}_{101}$	0.05	0.046	0.23	0.096	0.046	0.58	1.05
	$\mathbb{Z}_{2^{32}}$	0.15	0.14	0.54	0.3	0.14	1.5	1.03
$L = 5$	$\{0, 1\}$	0.53	0.47	12.75	1.04	0.49	28.39	1.08
	$\mathbb{Z}_{101}$	0.72	0.65	15.09	1.38	0.62	33.21	1.01
	$\mathbb{Z}_{2^{32}}$	1.91	1.79	35.71	3.82	1.73	79.99	1.17
$L = 10$	$\{0, 1\}$	2.21	2.013	101.1	4.449	2.005	226.4	1.05
	$\mathbb{Z}_{101}$	3.11	2.91	145.6	6.23	3.00	318.2	1.22
	$\mathbb{Z}_{2^{32}}$	6.33	5.88	315	12.66	6.12	649.9	1.18
$L = 20$	$\{0, 1\}$	2.558	2.295	355.1	5.142	2.295	804.3	1.05
	$\mathbb{Z}_{101}$	3.69	3.39	509.22	6.55	3.41	1163.4	1.21
	$\mathbb{Z}_{2^{32}}$	8.03	7.64	1113.5	16.49	7.89	2531.8	1.19
$L = 30$	$\{0, 1\}$	8.44	8.09	1170.39	46.81	27.13	2984.1	1.48
	$\mathbb{Z}_{101}$	9.17	8.88	1271.1	50.858	29.47	3242.0	1.82
	$\mathbb{Z}_{2^{32}}$	19.1	18.3	2418.9	103.3	59.4	6749.9	1.78

Table 5.7: BGV algorithms performance in user seconds.

Bibliography

[ABBC10] T. Acar, M. Belenkiy, M. Bellare, and D. Cash. *Cryptographic Agility and Its Relation to Circular Encryption*, pages 403–422. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[ABC+16] F. Armknecht, C. Boyd, C. Carr, K. Gjosteen, A. Jaschke, C. A. Reuter, and M. Strand. *A Guide to Fully Homomorphic Encryption*. Cryptology ePrint Archive, Report 2015/1192, 2016.

[ABD16] M. Albrecht, S. Bai, and L. Ducas. *A Subfield Lattice Attack on Overstretched NTRU Assumptions*, pages 153–178. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.

[ABF+13] J. Alwen, M. Barbosa, P. Farshim, R. Gennaro, S. D. Gordon, S. Tessaro, and D. A. Wilson. *On the Relationship between Functional Encryption, Obfuscation, and Fully Homomorphic Encryption*, pages 65–84. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[AD01] M. J. Atallah and W. Du. *Secure multi-party computational geometry*. In *International Workshop on Algorithms and Data Structures*, pages 165–179. Springer-Verlag, 2001.

[ADCHT13] E. Ayday, E. De Cristofaro, J.-P. Hubaux, and G. Tsudik. *The Chills and Thrills of Whole Genome Sequencing*. CoRR, abs/1306.1264, 2013.

[ASP13] J. Alperin-Sheriff and C. Peikert. *Practical Bootstrapping in Quasilinear Time*, pages 1–20. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[ASP14] J. Alperin-Sheriff and C. Peikert. *Faster Bootstrapping with Polynomial Error*, pages 297–314. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[BF99] D. Boneh and M. Franklin. *An Efficient Public Key Traitor Tracing Scheme*, pages 338–353. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[BF01] D. Boneh and M. Franklin. *Identity-Based Encryption from the Weil Pairing*, pages 213–229. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[BGH+13a] D. Boneh, C. Gentry, S. Halevi, F. Wang, and D. J. Wu. *Private Database Queries Using Somewhat Homomorphic Encryption*. In *Proceedings of the 11th International Conference on Applied Cryptography and Network Security*, ACNS'13, pages 102–118, Berlin, Heidelberg, 2013. Springer-Verlag.

[BGH13b] Z. Brakerski, C. Gentry, and S. Halevi. *Packed Ciphertexts in LWE-Based Homomorphic Encryption*. In *Public-Key Cryptography – PKC 2013*, volume 7778 of *Lecture Notes in Computer Science*, pages 1–13. Springer Berlin Heidelberg, 2013.

[BGV12] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. *Fully Homomorphic Encryption Without Bootstrapping*. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 309–325, New York, NY, USA, 2012. ACM.

[BHW15] A. Bishop, S. Hohenberger, and B. Waters. *New Circular Security Counterexamples from Decision Linear and Learning with Errors*, pages 776–800. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[BLLN13] J. W. Bos, K. Lauter, J. Loftus, and M. Naehrig. *Improved Security for a Ring-Based Fully Homomorphic Encryption Scheme*, pages 45–64. Springer Berlin Heidelberg, 2013.

[BLN13] J. Bos, K. Lauter, and M. Naehrig. *Private Predictive Analysis on Encrypted Medical Data*. Technical report, 2013.

[BLP+13] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. *Classical Hardness of Learning with Errors*. In *Proceedings of STOC*, pages 575–584, United States, 2013.

[BPHJ14] C. Bösch, A. Peter, P. Hartel, and W. Jonker. *SOFIR: Securely outsourced Forensic image recognition*. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2694–2698, May 2014.

[Bra12] Z. Brakerski. *Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP*, pages 868–886. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[BSW12] D. Boneh, G. Segev, and B. Waters. *Targeted Malleability: Homomorphic Encryption for Restricted Computations*. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 350–366, New York, NY, USA, 2012. ACM.

[BV11a] Z. Brakerski and V. Vaikuntanathan. *Efficient Fully Homomorphic Encryption from (Standard) LWE*. In *Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 97–106, Washington, DC, USA, 2011. IEEE Computer Society.

[BV11b] Z. Brakerski and V. Vaikuntanathan. *Fully Homomorphic Encryption from ring-LWE and Security for Key-Dependent Messages*. In *Proceedings of the 31st Annual Conference on Advances in Cryptology*, CRYPTO'11, pages 505–524, Berlin, Heidelberg, 2011. Springer-Verlag.

[BV14] Z. Brakerski and V. Vaikuntanathan. *Lattice-based FHE As Secure As PKE*. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, ITCS '14, pages 1–12, New York, NY, USA, 2014. ACM.

[CCF+15] A. Canteaut, S. Carpov, C. FONTAINE, T. Lepoint, M. Naya-Plasencia, P. Paillier, and R. Sirdey. *How to Compress Homomorphic Ciphertexts*. Research Report 2015/113, IACR Cryptology ePrint Archive, February 2015.

[CCK+13] J. H. Cheon, J.-S. Coron, J. Kim, M. S. Lee, T. Lepoint, M. Tibouchi, and A. Yun. *Batch Fully Homomorphic Encryption over the Integers*, pages 315–335. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[CDN01] R. Cramer, I. Damgård, and J. B. Nielsen. *Multiparty Computation from Threshold Homomorphic Encryption*, pages 280–300. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[CDPR16] R. Cramer, L. Ducas, C. Peikert, and O. Regev. *Recovering Short Generators of Principal Ideals in Cyclotomic Rings*, pages 559–585. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.

[CGGI16a] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. *Faster Fully Homomorphic Encryption: Bootstrapping in less than 0.1 Seconds*. IACR Cryptology ePrint Archive, 2016, 2016.

[CGGI16b] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. *A Homomorphic LWE Based E-voting Scheme*, pages 245–265. Springer International Publishing, Cham, 2016.

[CJ76] J. Conway and A. Jones. *Trigonometric diophantine equations (On vanishing sums of roots of unity).* Acta Arithmetica, 30(3):229–240, 1976.

[CLT14] J.-S. Coron, T. Lepoint, and M. Tibouchi. *Scale-Invariant Fully Homomorphic Encryption over the Integers*, pages 311–328. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[CM00] G. M. Cicuta and M. L. Mehta. *Probability density of determinants of random matrices.* Journal of Physics A: Mathematical and General, 33(45):8029, 2000.

[CM15] M. Clear and C. McGoldrick. *Multi-identity and Multi-key Leveled FHE from Learning with Errors*, pages 630–656. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[CMNT11] J.-S. Coron, A. Mandal, D. Naccache, and M. Tibouchi. *Fully Homomorphic Encryption over the Integers with Shorter Public Keys*, pages 487–504. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[CMO+14] X. Cao, C. Moore, M. O'Neill, N. Hanley, and E. O'Sullivan. *High-Speed Fully Homomorphic Encryption Over the Integers*, pages 169–180. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[CN12] Y. Chen and P. Q. Nguyen. *Faster Algorithms for Approximate Common Divisors: Breaking Fully-homomorphic-encryption Challenges over the Integers.* In *Proceedings of the 31st Annual International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'12, pages 502–519, Berlin, Heidelberg, 2012. Springer-Verlag.

[CNT12] J.-S. Coron, D. Naccache, and M. Tibouchi. *Public Key Compression and Modulus Switching for Fully Homomorphic Encryption over the Integers*, pages 446–464. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[Coc01] C. Cocks. *An Identity Based Encryption Scheme Based on Quadratic Residues*, pages 360–363. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[CS16] A. Costache and N. P. Smart. *Which Ring Based Somewhat Homomorphic Encryption Scheme is Best?*, pages 325–340. Springer International Publishing, Cham, 2016.

[DGM15] R. Dahab, S. Galbraith, and E. Morais. *Adaptive Key Recovery Attacks on NTRU-Based Somewhat Homomorphic Encryption Schemes*, pages 283–296. Springer International Publishing, Cham, 2015.

[DM15] L. Ducas and D. Micciancio. *FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second*, pages 617–640. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[DS16] L. Ducas and D. Stehlé. *Sanitization of FHE Ciphertexts*. In *Proceedings of the 35th Annual International Conference on Advances in Cryptology — EUROCRYPT 2016 - Volume 9665*, pages 294–310, New York, NY, USA, 2016. Springer-Verlag New York, Inc.

[Eve99] J. Evertse. *The number of solutions of linear equations in roots of unity*. Acta Arithmetica, 89(1):45–51, 1999.

[FGP14] D. Fiore, R. Gennaro, and V. Pastro. *Efficiently Verifiable Computation on Encrypted Data*. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 844–855, New York, NY, USA, 2014. ACM.

[FSF+13] S. Fau, R. Sirdey, C. Fontaine, C. Aguilar-Melchor, and G. Gogniat. *Towards Practical Program Execution over Fully Homomorphic Encryption Schemes*. In *Proceedings of the 2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, 3PGCIC '13, pages 284–290, Washington, DC, USA, 2013. IEEE Computer Society.

[FV12] J. Fan and F. Vercauteren. *Somewhat Practical Fully Homomorphic Encryption*. Cryptology ePrint Archive, Report 2012/144, 2012.

[Gen09a] C. Gentry. *Computing on Encrypted Data*. In *Proceedings of the 8th International Conference on Cryptology and Network Security*, CANS '09, pages 477–477, Berlin, Heidelberg, 2009. Springer-Verlag.

[Gen09b] C. Gentry. *A Fully Homomorphic Encryption Scheme*. PhD thesis, Stanford University, Stanford, CA, USA, 2009. AAI3382729.

[Gen09c] C. Gentry. *Fully Homomorphic Encryption Using Ideal Lattices*. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pages 169–178, New York, NY, USA, 2009. ACM.

[Gen10a] C. Gentry. *Computing Arbitrary Functions of Encrypted Data*. Commun. ACM, 53(3):97–105, March 2010.

[Gen10b] C. Gentry. *Toward Basing Fully Homomorphic Encryption on Worst-case Hardness*. In *Proceedings of the 30th Annual Conference on Advances in Cryptology*, CRYPTO'10, pages 116–137, Berlin, Heidelberg, 2010. Springer-Verlag.

[GGH+13] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. *Candidate Indistinguishability Obfuscation and Functional Encryption for All Circuits*. In *Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, FOCS '13, pages 40–49, Washington, DC, USA, 2013. IEEE Computer Society.

[GGP10] R. Gennaro, C. Gentry, and B. Parno. *Non-interactive Verifiable Computing: Outsourcing Computation to Untrusted Workers*, pages 465–482. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[GH09] C. Gentry and S. Halevi. *Hierarchical Identity Based Encryption with Polynomially Many Levels*, pages 437–456. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[GH11a] C. Gentry and S. Halevi. *Fully Homomorphic Encryption Without Squashing Using Depth-3 Arithmetic Circuits*. In *Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 107–109, Washington, DC, USA, 2011. IEEE Computer Society.

[GH11b] C. Gentry and S. Halevi. *Implementing Gentry's Fully-homomorphic Encryption Scheme*. In *Proceedings of the 30th Annual International Conference on Theory and Applications of Cryptographic Techniques: Advances in Cryptology*, EUROCRYPT'11, pages 129–148, Berlin, Heidelberg, 2011. Springer-Verlag.

[GHPS12] C. Gentry, S. Halevi, C. Peikert, and N. P. Smart. *Ring Switching in BGV-Style Homomorphic Encryption*. In *Proceedings of the 8th International Conference on Security and Cryptography for Networks*, SCN'12, pages 19–37, Berlin, Heidelberg, 2012. Springer-Verlag.

[GHPS13] C. Gentry, S. Halevi, C. Peikert, and N. P. Smart. *Field Switching in BGV-style Homomorphic Encryption*. J. Comput. Secur., 21(5):663–684, September 2013.

[GHS12a] C. Gentry, S. Halevi, and N. P. Smart. *Better Bootstrapping in Fully Homomorphic Encryption*. In *Proceedings of the 15th International Conference on Practice and Theory in Public Key Cryptography*, PKC'12, pages 1–16, Berlin, Heidelberg, 2012. Springer-Verlag.

[GHS12b] C. Gentry, S. Halevi, and N. P. Smart. *Fully Homomorphic Encryption with Polylog Overhead*. In *Proceedings of the 31st Annual International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'12, pages 465–482, Berlin, Heidelberg, 2012. Springer-Verlag.

[GHS12c] C. Gentry, S. Halevi, and N. P. Smart. *Homomorphic Evaluation of the AES Circuit*, pages 850–867. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[GHV10] C. Gentry, S. Halevi, and V. Vaikuntanathan. *A Simple BGN-type Cryptosystem from LWE*. In *Proceedings of Eurocrypt 2010, LNCS 6110*, pages 506–522, 2010.

[GN08] N. Gama and P. Q. Nguyen. *Predicting Lattice Reduction*, pages 31–51. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[Gra06] R. Gray. *Toeplitz and Circulant Matrices: A Review*. Foundations and Trends® in Communications and Information Theory, 2(3):155–239, 2006.

[GSW13] C. Gentry, A. Sahai, and B. Waters. *Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based*. In *Advances in Cryptology – CRYPTO 2013*, volume 8042 of *Lecture Notes in Computer Science*, pages 75–92. Springer Berlin Heidelberg, 2013.

[GVP16] L. Goubin and F. J. Vial Prado. *Blending FHE-NTRU Keys - The Excalibur Property*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016. To appear.

[Hai08] I. Haitner. *Semi-honest to Malicious Oblivious Transfer: The Black-box Way*. In *Proceedings of the 5th Conference on Theory of Cryptography*, TCC'08, pages 412–426, Berlin, Heidelberg, 2008. Springer-Verlag.

[HPS98] J. Hoffstein, J. Pipher, and J. H. Silverman. *NTRU: A Ring-Based Public Key Cryptosystem*. In *Lecture Notes in Computer Science*, pages 267–288. Springer-Verlag, 1998.

[HRSV11] S. Hohenberger, G. N. Rothblum, A. Shelat, and V. Vaikuntanathan. *Securely Obfuscating Re-Encryption*. Journal of Cryptology, 24(4):694–719, 2011.

[HS14] S. Halevi and V. Shoup. *Algorithms in HElib*. In *Advances in Cryptology – CRYPTO 2014*, volume 8616 of *Lecture Notes in Computer Science*, pages 554–571. Springer Berlin Heidelberg, 2014.

[HS15] S. Halevi and V. Shoup. *Bootstrapping for HElib*, pages 641–670. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[JPH13] A. Jeckmans, A. Peter, and P. Hartel. *Efficient Privacy-Enhanced Familiarity-Based Recommender System*, pages 400–417. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[Kah96] D. Kahn. *The codebreakers : the story of secret writing*. Scribner, New York, 1996.

[KL15] M. Kim and K. Lauter. *Private genome analysis through homomorphic encryption*. BMC Medical Informatics and Decision Making, 15(5):S3, 2015.

[KMR11] S. Kamara, P. Mohassel, and M. Raykova. *Outsourcing Multi-Party Computation*. Cryptology ePrint Archive, Report 2011/272, 2011.

[KPG99] A. Kipnis, J. Patarin, and L. Goubin. *Unbalanced Oil and Vinegar Signature Schemes*, pages 206–222. Springer Berlin Heidelberg, 1999.

[KRW15] V. Koppula, K. Ramchen, and B. Waters. *Separations in Circular Security for Arbitrary Length Key Cycles*, pages 378–400. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[KW16] V. Koppula and B. Waters. *Circular Security Counterexamples for Arbitrary Length Cycles from LWE*. IACR Cryptology ePrint Archive, 2016:117, 2016.

[LATV12] A. López-Alt, E. Tromer, and V. Vaikuntanathan. *On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption*. In *In STOC*, pages 1219–1234, 2012.

[LCH13] C.-C. Lee, P.-S. Chung, and M.-S. Hwang. *A Survey on Attribute-based Encryption Schemes of Access Control in Cloud Environments.* I. J. Network Security, 15(4):231–240, 2013.

[Len78] H. W. Lenstra. *Vanishing Sums of Roots of Unity*, volume 101, pages 249 – 268. Math. Centre Tracts 101, Vrije Univ, Amsterdam, 1978.

[LL00] T. Lam and K. Leung. *On Vanishing Sums of Roots of Unity*. Journal of Algebra, 224(1):91 – 109, 2000.

[LLAN15] K. Lauter, A. López-Alt, and M. Naehrig. *Private Computation on Encrypted Genomic Data*, pages 3–27. Springer International Publishing, Cham, 2015.

[LMSV12] J. Loftus, A. May, N. P. Smart, and F. Vercauteren. *On CCA-Secure Somewhat Homomorphic Encryption*, pages 55–72. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[LN14] T. Lepoint and M. Naehrig. *A Comparison of the Homomorphic Encryption Schemes FV and YASHE*. In *AFRICACRYPT 2014*, volume 8469 of *Lecture Notes in Computer Science*, pages 318–335, Marrakesh, Morocco, May 2014. Springer.

[LNV11] K. Lauter, M. Naehrig, and V. Vaikuntanathan. *Can Homomorphic Encryption Be Practical?* In *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop*, CCSW '11, pages 113–124, New York, NY, USA, 2011. ACM.

[LPR10] V. Lyubashevsky, C. Peikert, and O. Regev. *On ideal lattices and learning with errors over rings*. In *In Proc. of EUROCRYPT, volume 6110 of LNCS*, pages 1–23. Springer, 2010.

[LPR13] V. Lyubashevsky, C. Peikert, and O. Regev. *A Toolkit for Ring-LWE Cryptography*, pages 35–54. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[Maz11] G. Maze. *Natural density distribution of Hermite normal forms of integer matrices*. Journal of Number Theory, 131(12):2398 – 2408, 2011.

[Mic01] D. Micciancio. *Improving Lattice Based Cryptosystems Using the Hermite Normal Form*, pages 126–145. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[Mic11] D. Micciancio. *The Geometry of Lattice Cryptography*, pages 185–210. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[MP12] D. Micciancio and C. Peikert. *Trapdoors for lattices: Simpler, tighter, faster, smaller*. In *In EUROCRYPT*, pages 700–718, 2012.

[NR06] P. Q. Nguyen and O. Regev. *Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures*, pages 271–288. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[OYKU10] N. Ogura, G. Yamamoto, T. Kobayashi, and S. Uchiyama. *An Improvement of Key Generation Algorithm for Gentry's Homomorphic Encryption Scheme*, pages 70–83. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[Pei16] C. Peikert. *A Decade of Lattice Cryptography*. Found. Trends Theor. Comput. Sci., 10(4):283–424, March 2016.

[PS10] C. Pernet and W. Stein. *Fast computation of Hermite normal forms of random integer matrices*. Journal of Number Theory, 130(7):1675 – 1683, 2010.

[PTK13] A. Peter, E. Tews, and S. Katzenbeisser. *Efficiently Outsourcing Multiparty Computation Under Multiple Keys*. Trans. Info. For. Sec., 8(12):2046–2058, December 2013.

[RAD78] R. L. Rivest, L. Adleman, and M. L. Dertouzos. *On Data Banks and Privacy Homomorphisms*. Foundations of secure computation, pages 169–177, 1978.

[Reg05] O. Regev. *On Lattices, Learning with Errors, Random Linear Codes, and Cryptography*. In *In STOC*, pages 84–93. ACM Press, 2005.

[Reg10] O. Regev. *The learning with errors problem*, pages 191–204. 2010.

[Sha85] A. Shamir. *Identity-Based Cryptosystems and Signature Schemes*, pages 47–53. Springer Berlin Heidelberg, Berlin, Heidelberg, 1985.

[Sil13] A. Silverberg. *Fully Homomorphic Encryption for Mathematicians*. Cryptology ePrint Archive, Report 2013/250, 2013.

[Sin99] S. Singh. *The Code Book: The Secret History of Codes and Codebreaking*. Harper Collins Australia, 1999.

[SS10] D. Stehlé and R. Steinfeld. *Faster Fully Homomorphic Encryption*. In *Advances in Cryptology - ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 377–394. Springer Berlin Heidelberg, 2010.

[SS11a] P. Scholl and N. P. Smart. *Improved Key Generation for Gentry's Fully Homomorphic Encryption Scheme*, pages 10–22. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[SS11b] D. Stehlé and R. Steinfeld. *Making NTRU as Secure as Worst-Case Problems over Ideal Lattices*, pages 27–47. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[Ste08] J. P. Steinberger. *Minimal vanishing sums of roots of unity with large coefficients*. Proceedings of the London Mathematical Society, 97(3):689–717, 2008.

[SV10] N. P. Smart and F. Vercauteren. *Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes*, pages 420–443. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[SV14] N. P. Smart and F. Vercauteren. *Fully homomorphic SIMD operations*. Designs, Codes and Cryptography, 71(1):57–81, 2014.

[SW05] A. Sahai and B. Waters. *Fuzzy Identity-Based Encryption*, pages 457–473. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[TW12] E. Thomae and C. Wolf. *Solving Underdetermined Systems of Multivariate Quadratic Equations Revisited*, pages 156–171. Springer Berlin Heidelberg, 2012.

[Vai12] V. Vaikuntanathan. *How to Compute on Encrypted Data*, pages 1–15. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[vDGHV10] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. *Fully Homomorphic Encryption over the Integers*. In *Proceedings of the 29th Annual International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'10, pages 24–43, Berlin, Heidelberg, 2010. Springer-Verlag.

[vdPS13] J. van de Pol and N. P. Smart. *Estimating Key Sizes for High Dimensional Lattice-Based Systems*, pages 290–303. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[VYY15] M. Varia, S. Yakoubov, and Y. Yang. *HEtest: A Homomorphic Encryption Testing Framework*, pages 213–230. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[YSK+13] M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama, and T. Koshiba. *Secure Pattern Matching Using Somewhat Homomorphic Encryption*. In *Proceedings of the 2013 ACM Workshop on Cloud Computing Security Workshop*, CCSW '13, pages 65–76, New York, NY, USA, 2013. ACM.

Appendix A

Seeing the whole picture: An article reading guide

Approaching fully homomorphic encryption literature can be very disconcerting. Several teams of expert cryptography research have published hundreds of articles since 2009, and partly because of the speed of advancements, understanding the different lines of work and their relationships is not an easy task. We will list here some of the principal articles, ordered by chronology and battlefronts: efficiency, security, applications, implementations and displaying other properties. We begin by stating that we highly recommend the lecture of *On data banks and Privacy Homomorphisms*, by Adleman, Dertouzos and Rivest [RAD78].

Tag	Ref.	Title
1	[Gen09c]	*FHE using ideal lattices*
2	[vDGHV10]	*FHE over the Integers*
3	[BV11a]	*Efficient FHE from (Standard) LWE*
4	[LATV12]	*On-the-Fly MPC on the Cloud via Multikey FHE*
5	[GSW13]	*HE from LWE: Conceptually-Simpler [...]*

Table A.1: The five families

In table A.1, seminal works at the origin of the five FHE families are highlighted. In A.2, relevant publications are listed, ordered by year, alphabetically, and with a tag indicating the type of contributions. The objective of table A.2 is to facilitate the task of filtering publications, for instance, the tags “i,3” and “i,4” regroup principal articles providing an implementation of LWE or NTRU based schemes, and the tag “2” lists articles related to FHE over the integers.

e	=	Studies/Improves Efficiency
s	=	Studies/Improves Security
i	=	An implementation is discussed or provided
a	=	Describes practical applications
p	=	Displays special properties
{1,2,3,4,5}	=	Belongs to family (see table A.1)

Disclaimer: the following list is not intended to be exhaustive. It contains all principal articles to the date of this writing, in the opinion of the author. According to CiteSeerX and ACM digital libraries, all principal FHE focused articles with at least 2 non-self citations and citing Gentry's seminal work are in table A.2.

Table A.2: FHE publications map

Year	**Reference**	**Tags**	**Title**
2010	[Gen10b]	1,s	Toward Basing FHE on Worst-Case Hardness
	[GHV10]	p	A Simple BGN-type Cryptosystem from LWE
	[SV10]	1,e	FHE with Relatively Small Key and Ciphertext sizes
	[SS10]	1,e,s	Faster FHE
2011	[BV11b]	3,s,p	FHE from Ring-LWE and Security for KDM
	[CMNT11]	2,s,e	FHE over the Integers with Shorter Public Keys
	[GH11a]	p,e	FHE without squashing [...]
	[GH11b]	1,i	Implementing Gentry's FHE scheme
	[LNV11]	3,a,i	Can homomorphic encryption be practical?
	[SS11a]	1,e,i	Improved Key Generation for Gentry's FHE scheme
2012	[BGV12]	3,p,e	FHE without bootstrapping
	[Bra12]	e	FHE without modulus switching from classical GapSVP
	[BSW12]	p,s	Targeted Malleability: HE for Restricted Computations
	[FV12]	3,e,	Somewhat practical FHE
	[GHS12b]	3,e	FHE with Polylog overhead
	[GHPS12]	3,e	Ring Switching in BGV-Style HE
	[LMSV12]	1,s	On CCA-Secure SHE
	[CNT12]	2,e,i,s	Public Key Compression and Modulus Switching [...]
2013	[ASP13]	1,e	Practical Bootstrapping in Quasilinear Time
	[BGH$^+$13a]	3,a	Private Database Queries Using SHE
	[BLLN13]	3,s	Improved Security for a Ring-Based FHE scheme
	[BLN13]	1,a,i	Private Predictive Analysis on Encrypted Medical Data
	[BLP$^+$13]	3,s	Classical Hardness of LWE
	[BGH13b]	3,p,e	Packed Ciphertexts in LWE-Based HE
	[CCK$^+$13]	3,e,p	Batch FHE over the Integers
	[FSF$^+$13]	3,i	Towards practical program execution over FHE schemes
	[GHPS13]	3,e,s	Field Switching in BGV-style Homomorphic Encryption
	[JPH13]	3,a	Efficient privacy-enhanced [...] recommender system
	[vdPS13]	3,e,i	Estimating Key Sizes For High Dimensional [...]
2014	[ASP14]	5,s,e	Faster Bootstrapping with Polynomial Error
	[BV14]	5,s,	Lattice Based FHE as secure as PKE
	[CMO$^+$14]	2,i	High-Speed FHE Over the Integers

(Continued on next page)

Table A.2 – continued from previous page

Year	Reference	Tags	Title
	[CLT14]	2,p,i	Scale-Invariant FHE over the Integers
	[LN14]	3,e,i	A Comparison of [...] FV and YASHE
	[HS14]	3,i	Algorithms in HElib
	[FGP14]	p	Efficiently Verifiable Computation on Encrypted Data
	[SV14]	1,a,e	Fully homomorphic SIMD operations
2015	[CCF+15]	e	How to Compress Homomorphic Ciphertexts
	[CM15]	3,p	Multi-identity and Multi-key Leveled FHE from LWE
	[DGM15]	3,4,s	Adaptive Key Recovery Attacks on NTRU [...]
	[DM15]	3,e,i	Bootstrapping in less than a second
	[HS15]	3,e	Bootstrapping for HElib
	[KL15]	3,i,a	Private genome analysis through HE
	[LLAN15]	3,4,a,i	Private Computation on Encrypted Genomic Data
	[VYY15]	i	HEtest: A HE Testing Framework
2016	[ABD16]	4,s	A subfield lattice attack on overstretched [...]
	[CDPR16]	4,s	Recovering Short Generators of Principal Ideals [...]
	[CGGI16a]	5,e,i	Bootstrapping in less than 0.1 second
	[CGGI16b]	3,a	A Homomorphic LWE Based E-voting Scheme
	[CS16]	e	Which Ring Based SHE Scheme is Best

We also recommend the lecture of the following articles, which are related to or have implications on FHE: [Gen10a, Mic01, GN08, LPR13, Vai12, Sil13, MP12, HRSV11, KMR11, Mic11, Pei16].

Appendix B

Homomorphic operations on sets

In this appendix, we describe how to handle homomorphic operations over sets of data, where cardinality and order of the elements are kept private. We give circuits that perform equality and membership tests, cardinality, union, intersection and Jaccard similarity.

Context and encoding conventions

We suppose that there are multiple users holding sets of similar data (i.e. all sets are made of lists of names, or account numbers, etc). Users want to store their data in the cloud, in order to perform homomorphic evaluations on two different sets as wholes. They do not want to leak the cardinality of their sets, nor any ciphertext identification. We suppose that

- Each element of a set can be encoded using κ bits with non-zero Hamming weight.
- There is a null element encoded with κ zero bits.
- Encryption of a set is performed element-wise, padding with encryptions of `null` until a number n of elements is reached, and then permuting randomly the ciphertexts.
- Decryption is performed element-wise.

For instance, if $\kappa = 4$ and $n = 8$, the set $S = \{1001, 0110\}$ is encrypted by a set $C = \{c_1, c_2, \ldots, c_8\}$ such that two elements in C are encryptions of "1001" and "0110" and the rest are encryptions of `null`="0000".

Equality circuits

Let $a, b \in \{0,1\}^\kappa$ be two plaintexts. The equality circuit is the bitwise computation of NXOR, followed by the AND of all outputs:

$$\mathrm{Eq}(a,b) = 1 \oplus \prod_{i=1}^{\kappa}(a_i \oplus b_i \oplus 1) = \begin{cases} 0 & \text{if} \quad a = b, \\ 1 & \text{if} \quad a \neq b. \end{cases}$$

The multiplicative depth of this circuit is $\lceil \log \kappa \rceil$. Let $\mathrm{Eq}_0 = \mathrm{Eq}(a, 0)$ the function that tests if a is the null element. It can be performed with the following circuit

$$\mathrm{Eq}_0(a) = \mathrm{NOT}\left(\mathrm{AND}_{i=1}^{\kappa}\,\mathrm{NOT}(a_i)\right) = 1 \oplus \prod_{i=1}^{\kappa}(a_i \oplus 1) = \begin{cases} 0 & \text{if} \quad a = 0, \\ 1 & \text{if} \quad a \neq 0. \end{cases}$$

Another possible equality circuit involves computing the Hamming distance of a and b:

$$\mathrm{EqH}(a,b) = \mathrm{Ham}(a \,\mathrm{XOR}\, b) = \begin{cases} 0 & \text{if} \quad a = b, \\ h > 0 & \text{if} \quad a \neq b. \end{cases}$$

$$\mathrm{EqH}_0(a,b) = \mathrm{Ham}(a) = \begin{cases} 0 & \text{if} \quad a = 0, \\ h > 0 & \text{if} \quad a \neq 0. \end{cases}$$

Cardinality

Let $E = \{c_1, \ldots, c_n\}$ a cipherset. To homomorphically compute the cardinality of the underlying set, one can simply perform

$$\#(E) = \mathrm{Eq}_0(c_1) + \mathrm{Eq}_0(c_2) + \cdots + \mathrm{Eq}_0(c_n),$$

which will return an encryption of the number of non zero integers in the underlying set counting repetitions.

Membership

Let $E = \{c_1, \ldots, c_n\}$ a cipherset and z a ciphertext. Then

$$\mathrm{in}(z,E) = \mathrm{Eq}(z,c_1) + \mathrm{Eq}(z,c_2) + \cdots + \mathrm{Eq}(z,c_n) = \begin{cases} \text{encryption of } 0 & \text{if} \quad z \notin E, \\ \text{encryption of } 1 & \text{if} \quad z \in E. \end{cases}$$

Intersection and union

Let $A = \{a_1, \ldots, a_n\}, B = \{b_1, \ldots, b_n\}$ be two ciphersets. One can compute the intersection and union with the following algorithms:

The intersection algorithm outputs a set of n ciphertexts, which are encryptions of 0 or encryptions of elements in the underlying intersection.

The union algorithm outputs a set of $2n$ ciphertexts, which are encryptions of 0 or encryptions of elements in the underlying union, using n^2 calls to the equality circuit.

Algorithm 23 Homomorphic Intersection

$I \leftarrow \emptyset$
for i from 1 to n **do**
    $z \leftarrow \text{in}(a_i, B) \cdot a_i$
    $I \leftarrow I \cup \{z\}$
end for
Output I.

Algorithm 24 Homomorphic Union

$U \leftarrow A$
for i from 1 to n **do**
    $z \leftarrow \text{NOT in}(b_i, A) \cdot b_i$
    $U \leftarrow U \cup \{z\}$
end for
Apply a random permutation to the set U
Output U.

Jaccard Similarity

With this one can compute the homomorphic Jaccard index of two sets, which is defined as

$$J(A, B) = \frac{\#(A \cap B)}{\#(A \cup B)}.$$

This can be done with $2n^2 + 3n$ calls to the equality circuit as written. Noticing that $\#(A \cup B) = \#(A) + \#(B) + \#(A \cap B)$, this can be reduced to $n^2 + 3n$ calls.

Appendix C

Coefficient correlation in the Small Factors Problem

In this Appendix, we quantify the linear correlation between any two coefficients of the known polynomial in the Small Factors Problem, when instantiated with Bernoulli sampling of the factors. We shall see that this correlation is inversely proportional to the sparseness of the secret factors, and for inattentive selection of parameters this could help to attack SFP.

Preliminaries

A discrete random variable θ is a Bernoulli trial with parameter p if it produces only two outcomes, one with probability p and the other with probability $1-p$. If θ is a Bernoulli trial,

$$\begin{cases} \theta \in \{0,1\}, \\ \mathbb{P}(\theta = 1) = p, \mathbb{P}(\theta = 0) = 1-p, \\ \mathbb{E}(\theta) = p, \\ \mathrm{Var}(\theta) = \mathbb{E}((\theta - p)^2) = p(1-p). \end{cases}$$

For $i = 1, \ldots, n$, let θ_i be independent Bernoulli trials with the same parameter p, and let $\chi = \sum_{i=1}^{n} \theta_i$. Then χ follows the Binomial distribution with parameters n, p.

$$\begin{cases} \chi \in \{0, \ldots, n\}, \\ \mathbb{P}(\chi = k) = \binom{n}{k} p^k (1-p)^{n-k}, \\ \mathbb{E}(\chi) = np, \\ \mathrm{Var}(\chi) = np(1-p). \end{cases}$$

C.1 $\phi(x) = x^n - 1$ (NTRU ring)

Proposition C.1.1. *Let $\alpha(x), \beta(x)$ two polynomials of degree $n-1$ and with random coefficients in $\{0,1\}$. Each coefficient was sampled using independent coin tosses*

with probability p of being 1. Let $\gamma(x) = \alpha(x)\beta(x) \bmod (x^n - 1)$. Then each coefficient of γ follows a binomial distribution of parameters n, p^2. The linear correlation coefficient between two distinct coefficients is $\frac{2p}{p+1}$.

Proof: For $0 \leq i, j \leq n-1$, let α_i, β_j be the independent Bernoulli trials of parameter p used to generate the polynomials $\alpha(x) = \sum_{i=0}^{n-1} \alpha_i x^i$ and $\beta(x) = \sum_{j=0}^{n-1} \beta_j x^j$. For $k = 0, \ldots, n-1$ let $\gamma_k \in \mathbb{Z}$ be the random variable defined by the equation

$$\gamma(x) = \sum_{k=0}^{n-1} \gamma_k x^k = \alpha(x) \cdot \beta(x) \mod (x^n - 1).$$

Expanding the right-hand product and performing modular reduction gives

$$\gamma_k = \sum_{i=0}^{n-1} \alpha_i \beta_{k-i \bmod n}.$$

Note that the random discrete variables $Z_{i,k} := \alpha_i \beta_{k-i}$ are Bernoulli trials of parameter p^2 (as they are a product of two independent B.t.). This allows to compute

$$\begin{cases} \gamma_k \in \{0, \ldots, n\}, \\ \mathbb{P}(\gamma_k = l) = \binom{n}{l} p^{2l} (1-p^2)^{n-l}, \\ \mathbb{E}(\gamma_k) = np^2, \\ \mathrm{Var}(\gamma_k) = np^2(1-p^2). \end{cases}$$

From this is is clear that every pair γ_a, γ_b for $a \neq b$ consists of dependent random variables. We compute the covariance of this pair: let ξ_c be the characteristic function of the condition c ($\xi_c = 1$ if c is true). Then

$$\begin{aligned} \mathrm{Cov}(\gamma_a, \gamma_b) &= \mathbb{E}((\gamma_a - np^2)(\gamma_b - np^2)) \\ &= \mathbb{E}(\gamma_a \gamma_b) - np^2(\mathbb{E}(\gamma_a) + \mathbb{E}(\gamma_b)) + n^2 p^4 \\ &= \mathbb{E}(\gamma_a \gamma_b) - n^2 p^4. \\ \mathbb{E}(\gamma_a \gamma_b) &= \mathbb{E}\left(\left(\sum_{i=0}^{n} \alpha_i \beta_{a-i \bmod n} \right) \left(\sum_{j=0}^{n} \alpha_j \beta_{b-j \bmod n} \right) \right) \\ &= \sum_{0 \leq i,j < n} \mathbb{E}(\alpha_i \alpha_j) \mathbb{E}(\beta_{a-i[n]} \beta_{b-j[n]}), \\ &= \sum_{0 \leq i,j < n} (\xi_{i \neq j, a-i \neq b-j[n]} + \xi_{i=j} + \xi_{a-i=b-j[n]}) \mathbb{E}(\alpha_i \alpha_j) \mathbb{E}(\beta_{a-i[n]} \beta_{b-j[n]}) \\ &= (n^2 - 2n)p^4 + np^3 + np^3. \end{aligned}$$

Therefore,

$$\begin{aligned} \mathrm{Cov}(\gamma_a, \gamma_b) &= 2n(p^3 - p^4), \\ \mathrm{Corr}(\gamma_a, \gamma_b) &= \frac{\mathrm{Cov}(\gamma_a, \gamma_b)}{\sigma_a \sigma_b} = \frac{2n(p^3 - p^4)}{np^2(1-p^2)} = \frac{2p}{1+p}. \end{aligned}$$ ■

C.2 $\phi(x) = x^n + 1$ (rLWE ring)

Proposition C.2.1. *Let $\alpha(x), \beta(x)$ two polynomials of degree $n-1$ and with random coefficients in $\{0,1\}$. Each coefficient was sampled using independent coin tosses with probability p of being 1. Let $\gamma(x) = \alpha(x)\beta(x) \bmod (x^n+1)$. Then each coefficient of γ is sampled as the substraction of two binomial distributions. Moreover, if γ_k denotes the k-th coefficient of $\gamma(x)$, we have*

$$\begin{cases} \mathbb{E}(\gamma_k) = p^2(2k+2-n), \\ \mathrm{Var}(\gamma_k) = np^2(1-p^2), \\ \mathrm{Corr}(\gamma_a, \gamma_b) = \left(1 - 2\frac{b-a}{n}\right)\frac{2p}{1+p} \text{ if } a < b. \end{cases}$$

Proof: The setting is the same, but now the coefficients of $\gamma(x)$ are

$$\gamma_k = \sum_{i=0}^{n-1} \alpha_i \beta_{k-i[n]} \cdot \pi_k(i) \quad \text{where} \quad \pi_k(i) = \begin{cases} 1 \text{ if } i \le k, \\ -1 \text{ if } i > k. \end{cases}$$

By linearity and independence, we have

$$\begin{aligned} \mathbb{E}(\gamma_k) &= p^2 \sum_{i=0}^{n-1} \pi_k(i) = p^2(2k+2-n), \\ \mathrm{Var}(\gamma_k) &= \mathbb{E}(\gamma_k^2) - p^4(2k+2-n)^2, \\ \mathbb{E}(\gamma_k^2) &= \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} \mathbb{E}(\alpha_i\alpha_j)\mathbb{E}(\beta_{k-i[n]}\beta_{k-j[n]})\pi_k(i)\pi_k(j) \\ &= p^4 \sum_{i,j<n; i\neq j} \pi_k(i)\pi_k(j) + p^2 \sum_{i=0}^{n} \pi_k(i)^2; \\ &= p^4((2k+2-n)^2 - n) + p^2 n, \end{aligned}$$

yielding the claimed expression for $\mathrm{Var}(\gamma_k)$. Now, let $a < b$, then

$$\begin{aligned} \mathbb{E}(\gamma_a\gamma_b) &= \sum_{i,j<n} \mathbb{E}(\alpha_i\alpha_j\beta_{a-i[n]}\beta_{b-j[n]})\pi_a(i)\pi_b(j) \\ &= p^4 \sum_{i\neq j; a-i\neq b-j[n]} \pi_a(i)\pi_b(j) + p^3 \sum_{l=0}^{n-1} \pi_a(l)\pi_b(l) + p^3 \sum_{l=0}^{n-1} \pi_a(l)\pi_b(b-a+l[n]) \\ &= p^4 S_1 + p^3(S_2 + S_3). \\ S_2 &= \sum_{l=0}^{n-1} \pi_a(l)\pi_b(l) = n - 2(b-a). \end{aligned}$$

It holds that

$$\pi_b(b-a+l[n]) = \begin{cases} 1 & \text{if} \quad b-a+l[n] \le b \\ -1 & \text{if} \quad b-a+l[n] > b \end{cases} = \begin{cases} \pi_a(l) & \text{if} \quad l < n-(b-a) \\ 1 & \text{if} \quad l \ge n-(b-a) \end{cases},$$

therefore

$$\begin{aligned} S_3 &= \sum_{l=0}^{n-1} \pi_a(l)\pi_b(b-a+l[n]) \\ &= \sum_{l=0}^{n-(b-a)-1} \pi_a(l)^2 + \sum_{l=n-(b-a)}^{n-1} \pi_a(l) \\ &= n-(b-a)-(n-(n-(b-a))) \\ &= n-2(b-a) \end{aligned}$$

and

$$\begin{aligned} S_1 &= \sum_{i\neq j; a-i\neq b-j[n]} \pi_a(i)\pi_b(j) \\ &= \left(\sum_{i=0}^{n-1} \pi_a(i)\right)\left(\sum_{j=0}^{n-1} \pi_b(j)\right) - S_2 - S_3 \\ &= (2(a+1)-n)(2(b+1)-n) - 2n + 4(b-a). \end{aligned}$$

Replacing these expressions we obtain

$$\mathbb{E}(\gamma_a\gamma_b) = p^4((2(a+1)-n)(2(b+1)-n) - 2n + 4(b-a)) + 2p^3(n-2(b-a)),$$

allowing to compute

$$\begin{aligned} \operatorname{Cov}(\gamma_a,\gamma_b) &= \mathbb{E}(\gamma_a\gamma_b) - \mathbb{E}(\gamma_a)\mathbb{E}(\gamma_b) \\ &= 2(n-2(b-a))(p^3-p^4), \end{aligned}$$

and finally

$$\begin{aligned} \operatorname{Corr}(\gamma_a,\gamma_b) &= \frac{2(n-2(b-a))(p^3-p^4)}{np^2(1-p^2)} \\ &= \left(1-2\frac{b-a}{n}\right)\frac{2p}{1+p}. \end{aligned}$$

Note that $|\operatorname{Corr}(\gamma_a,\gamma_b)| \leq (1-1/n)2p/(p+1)$, the extreme cases achieved when $b=a+1$ and $b=n-1, a=0$. ■

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Contributions à la conception et analyse des schémas de chiffrement complètement homomorphe

Mots clés : cryptographie, chiffrement complètement homomorphe, cryptographie à clé publique

Résumé : Les schémas de Chiffrement Complètement Homomorphe permettent de manipuler des données chiffrées avec une grande flexibilité : ils rendent possible l'évaluation de fonctions à travers les couches de chiffrement. Depuis la découverte du premier schéma FHE en 2009 par Craig Gentry, maintes recherches ont été effectuées pour améliorer l'efficacité, atteindre des nouveaux niveaux de sécurité, trouver des applications et détecter des liens avec d'autres domaines de la cryptographie. Dans cette thèse, nous avons étudié en détail ce type de schémas. Nos contributions font état d'une nouvelle attaque de récupération des clés contre le premier schéma FHE, et d'une nouvelle notion de sécurité en structures hiérarchiques, évitant une forme de trahison entre les usagers tout en gardant la flexibilité FHE. Enfin, on décrit des implémentations informatiques. Cette recherche a été effectuée au sein du Laboratoire de Mathématiques de Versailles avec le Prof. Louis Goubin.

Title : Contributions to Design and Analysis of Fully Homomorphic Encryption Schemes

Keywords : cryptography, fully homomorphic encryption, public-key cryptography

Abstract : Fully Homomorphic Encryption schemes allow public processing of encrypted data. Since the groundbreaking discovery of the first FHE scheme in 2009 by Craig Gentry, an impressive amount of research has been conducted to improve efficiency, achieve new levels of security, describe applications and detect connections to other areas of cryptography. In this Dissertation, we first give a detailed account on research these past years. Our contributions include a key-recovery attack against the ideal lattices FHE scheme and a new security conception inside hierarchic structures, avoiding betrayal between users to some extent, while maintaining the homomorphic encryption flexibility and also addressing an homomorphic re-encryption paradigm we detected. We also describe a working implementation of a recent proposal. This research was done in the Laboratoire de Mathématiques de Versailles, under supervision of Prof. Louis Goubin.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France