University of Wollongong 

FULLY HOMOMORPHIC ENCRYPTION

A Thesis Submitted in Partial Fulfilment of
the Requirements for the Award of the Degree of

Master of Computer Science - Research

from

UNIVERSITY OF WOLLONGONG

by

Zhunzhun CHEN
BEng, Mphil

School of Computer Science and Software Engineering
Faculty of Informatics

Master of Philosophy

CERTIFICATION

I, Zhunzhun CHEN, declare that this thesis, submitted in partial fulfilment of the requirements for the award of Master of Computer Science - Research, in the School of Computer Science and Software Engineering, Faculty of Informatics, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications at any other academic institution.

(Signature Required)

Zhunzhun CHEN
Master of Philosophy

Table of Contents

Fully Homomorphic Encryption

Zhunzhun CHEN

A Thesis for

School of Computer Science and Software Engineering
University of Wollongong

ABSTRACT

Fully Homomorphic encryption can compute arbitrary functions on encrypted data which proposed in 1978. After Gentry propsed the first Fully homomorphic encryption scheme in 2009, FHE has made a great progress. Lattice-based cryptography is considered to be secure against quantum computers. Lattice-based cryptographic schemes has simple computations and their hardness are as hard as approximating several lattice problem in the worst case. FHE still facing some problems, so it is important to construct better FHE scheme. The notion of a fully homomorphic encryption scheme over integers with public key compression has been proposed by Coron. The main attractive feature of this scheme is the reduction of the public key size, which is obtained by encrypting the plaintext with a quadratic form in the public key elements instead of in a linear form. In this work, we adopt this technique and apply it to the hidden ideal lattice scheme to acquire a more efficient scheme based on the hidden ideal lattice. The security of our scheme is based on the bounded distance decoding over the hidden ideal lattice. Additionally, we also describe a variant of the scheme with higher degrees. The scheme shows a better level of efficiency in comparison to the original scheme.

KEYWORDS: RSA, Fully Homomorphic Encryption, Compress

Acknowledgements

I would have not finished this project without the support of my family who has always been there for me whenever I need them, the encouragement they give to keep me going and their love to empower me that never fails all the time. Thank you.

I would like to thank my supervisors Willy SUSILO and Thomas PLANTARD who have given me a chance to prove that I can do things on my own. They gave me a lot of positive perspective in life. They who taught me things far more of my understanding. I thank them for challenging me to do this project. Thank you.

Chapter 1

Introduction

1.1 Introduction

Information security is the most important area in computer science today. Cryptosystems as the fundamental primitives ensures data transfer more efficient and accurate. Encryption scheme is provided by different protocols and systems. With the development of computer technology, although quantum computer is still in its infancy, quantum cryptography has developed rapidly. Post-quantum cryptography aims to construct cryptographic algorithms which are secure against an attack by a quantum computer. The security of the algorithms relies on hard mathematical problems, there are three classical hard problems: the integer factorization problem, the discrete logarithm problem or the elliptic-curve discrete logarithm problem[57][36]. Under the current classical computer, these algorithms are hard to solve, however, they can be solved by quantum computers easily by running Shor's algorithm [61]. Symmetric cryptographic algorithms and hash functions in the public-key system are considered to be relatively secure against attacks by quantum computers [2].

Post-quantum cryptography has six categories of algorithms based on different hard mathematical problems.

- Latticed-based cryptography: the cryptographic system includes Learning with Errors, Ring-Learning with Errors (Ring-LWE)[55], the Ring Learning with Errors Key Exchange and the Ring Learning with Errors Signature, the older NTRU or GGH encryption schemes, and the newer NTRU signature and BLISS signatures [50].

- Multivariate cryptography: the cryptographic system includes the Rainbow scheme [21]. Rainbow also could provide a quantum secure multivariate signature scheme called the Rainbow Signature Scheme [21].

- Hash-based cryptography: the cryptographic system includes Lamport signatures and the Merkle signature scheme. RSA and DSA are the most familiar hash based digital signatures.

- Code-based cryptography: the cryptographic system is based on error-correcting codes, the McEliece and Niederreiter encryption algorithms are two classic algorithms [48].

- Supersingular elliptic curve isogeny cryptography: the cryptographic system relies on the properties of supersingular elliptic curves. Diffie-Hellman like key exchange has better performance to resist quantum computing than the Diffie-Hellman and elliptic curve DiffieHellman key exchange methods [23].

- Symmetric Key based cryptography: Grover's algorithm is the best quantum attack against generic symmetric-key systems. This approach is more effective in small key size for post-quantum cryptography [34].

To prove the security of a cryptographic algorithm is equivalent to proving the mathematical problem is hard. The procedure of this proof is called "security reductions". The security of given cryptographic algorithms above are reduced to the security of different known hard problems.

- Ring-LWE Signature: the security reduction of RLWE is the shortest-vector problem (SVP) in a lattice as a lower bound on the security which is a NP-hard problem [41].

- NTRU, BLISS: the security reduction of these two algorithms are the closest-vector problem (CVP) in a lattice as a lower bound on the security which is also a NP-hard problem [22].

- Rainbow: multivariate quadratic equation cryptosystems called "Unbalanced Oil and Vinegar Cryptosystems" is a NP-hard problem. The Rainbow Multivariate Equation Signature Scheme is a class of multivariate quadratic equation cryptosystems [8]. The Rainbow Multivariate Equation Signature Scheme is a hard problem .

- Merkle signature scheme: one-way hash functions is a well known hard problem. The security reduction of Merkle Hash Tree signatures has proved to rely on the one-way hash function [24].

- McEliece: the Syndrome Decoding Problem (SDP) is also known to be NP-hard problem. The security reduction of McEliece Encryption System is SDP [60].

- Supersingular elliptic curve isogeny cryptography: Unlike other cryptosystems, this system has no security reduction to a known NP-hard problem. Delfs and Galbraith indicates the difficulty of the problem is as hard as the inventors of the key exchange which relies on constructing an isogeny between two supersingular curves with the same amount of points [18].

1.1.1 Lattice-Based Cryptography

A classical algorithm such as the RSA and Diffie-Hellman cryptosystem are easily attacked by a quantum computer. Recently, to resist the attack by both classical

and quantum computers, the lattice-based cryptosystem has been widely introduced. Lattice-based cryptography is the asymmetric cryptographic according to the properties of lattices. In the n-dimensional Euclidean space $\mathbb{R}^n$, we define a lattice $\mathcal{L}$ as a set of points which has a strong periodicity property in real analysis. A basis is the basic component of $\mathcal{L}$. Basis is a set of vectors which is represented by the linear combination of any element with unique integer coefficients. Due to the property of the cryptosystems, the ciphertext, public key, and private key must be taken from a finite space, therefore, the lattices used for cryptography is over the finite field only. The most two famous mathematical problems based on the lattices properties. One is the Shortest Vector Problem (SVP) and the other is Closest Vector Problem (CVP) [1]. Both are hard to be solved without a good basis, so the security of algorithm relies on the hard problem to find the good basis. An effective method to find the good basis (nearly orthogonal vectors) is using lattice basis reduction. If an attacker can compute such a lattice basis within polynomial time, the CVP and SVP problems are not hard to solve any more, the corresponding algorithms are not secure as well. The LLL algorithm is a quite effective to compute good basis, and so many alternative algorithms based on LLL algorithm to run faster or more efficienctly [54].

1.1.2 Fully Homomorphic Encryption

The encryption is an effective strategy to secure the data information when its transforming through the channel. The secret key helps the key owner to decrypt the ciphertext but it is useless for other people. Can we find the algorithm which can perform arbitrary operations on the ciphertext without secret key. A scheme is called fully homomorphic if it can operate on the ciphertext without the knowledge of the secret key. The notion of a fully homomorphic encryption scheme has been known to be very useful in the cloud computing environment, ciphertext retrieval and secure

multi-party computation [13]. For instance, the clouding computing, the customer encrypts the plaintext, keeps the data storage in the server of cloud. When customer request the operation, the server of cloud does not need to decrypt the ciphertext, but can perform the operation on the ciphertext. The decryption of the ciphertext will have the same result as the operation on the plaintext. The homomorphism gives the same outcome as the operation on the plaintext and plaintext.

In 1978, Rivest, Adleman and Dertouzos [56] introduced the original concept of privacy homomorphism which allows computation on encrypted data without decryption. They posed the construction of privacy homomorphism (and hence, fully homomorphic encryption) as an open research problem. For any valid function f and plaintext m, the operation on ciphertexts is equivalent to the same operation on plaintext. In such a definition, given a function f and a ciphertext c which encrypt a plaintext m, it is possible to transfer c into a new ciphertext c' which encrypts $f(m)$ [56] . There have been many attempts to achieve this goal. Some of them can satisfy additive homomorphism only or multiplicative homomorphism only, and meanwhile some other schemes have been successful enabling both operations with limited level of operations. The 'Polly Cracker' scheme can evaluate arbitrary level operation in any circuit. Nevertheless, the size of ciphertext will increase exponentially with the depth of the circuit [65]. We note that none of these schemes is a fully homomorphic scheme. The first breakthrough has been provided by Gentry in his construction of the first fully homomorphic encryption in 2009 [25]. Due to the characteristic of the addition and multiplication over $\mathbb{Z}_2$, which forms a complete set of those operations, the scheme can evaluate the operation on encrypted data in polynomial time.

Gentrys approach to fully homomorphic encryption is achieved by incorporating the bootstrapping technique, which seems to be the inherent efficiency bottleneck [28]. That is why fully homomorphic encryption schemes cannot be adopted in practice yet.

The natural fully homomorphic encryption scheme has not been found so far, the majority schemes proposed use Gentry's first idea: construct a somewhat homomorphic encryption scheme first, then applied the squash on the decryption algorithm, finally use the bootstrapping technique to achieve the fully homomorphic encryption scheme [26].

To construct a somewhat homomorphic encryption scheme means construct a scheme with a limited number of homomorphic operations. The somewhat homomorphic of Gentry's framework is GGH cryptosystem which based on the ideal lattice. There are two kinds of basis, one is 'good' basis which can be used as the secret key, another basis is 'bad' for the public key [26]. The hard lattice problem is a bounded distance decoding problem over an ideal lattice. The encryption is mapping a plaintext to a vector which is close to the lattice by using public key, the process is to encrypt plaintext by using the bad basis. The decryption is reducing the vector to the message by using the good basis which is the secret key.

During the evaluations, since the noise of the ciphertext is expanded over the bound especially in the multiplicative operation, it occurs the failure in the decryption. Gentry used 'homomorphic decryption' to control the noise increasing. Encrypt the ciphertext and the corresponding public key by evaluate key, and input the result into the decryption circuit, output a new ciphertext. If the error of the ciphertext is able to evaluate one more time especially in multiplication after each operation, then the ciphertext can perform unlimited times operation [28], [30]. Since the somewhat homomorphic encryption scheme can only perform a limited number of operations with low-degree polynomials, the next step is to squash the decryption procedure. The purpose is to express the decryption function as a low-degree polynomial which can be supported by the scheme. Finally, the last step is bootstrapping, it is the major procedure to transfer the somewhat homomorphic encryption scheme into a

fully homomorphic scheme [26].

There are three categories of fully homomorphic encryption scheme: ideal lattice based scheme, integer based scheme and learning with error based scheme. Smart and Vercauteren [63] used the principle of ideal lattice to construct the fully homomorphic scheme. They selected two integers to represent the lattice and maintained a smaller key size. The integer based scheme proposed by van Dijk [20], where its security is based on the approximate greatest common factor. Plantard, Susilo and Zhang [51] proposed the notion of hidden ideal lattice for the construction of fully homomorphic encryption schemes. The hidden ideal lattice scheme unifies the ideal lattice scheme and integer scheme. Instead of publishing the lattice, they used vectors close to a lattice which is called the hidden ideal lattice. The security of the hidden ideal lattice scheme relies on a bounded distance decoding problem over hidden ideal lattice rather than the subset sum problem.

The implementation of the fully homomorphic encryption scheme by van Dijk et al. [20] shows that the public key size is too big for any practical system [29]. Reducing the size of the public key is the main point to make the scheme more practical, which is achieved by shortening the length or decreasing the number of public keys [58]. Coron proposed a technique to reduce the number of public key, then shrunk the size of the public key based on the scheme over the integers [15]. The technique can improve the efficiency of implementing of the fully homomorphic encryption by the small size of public key.

1.2 Research Objective

This thesis aims for the following main research objective according to the above discussing:

- Constructing new fully homomorphic encryption scheme. The original fully ho-

momorphic scheme constructed by Gentry[26] uses bootstrapping which has low efficiency. To construct new fully homomorphic without bootstrapping will be our main research.

- The DGHV[20] has large public key size which affects the implementing performance. To apply the public key compression to our scheme can improve the performance of the implementation and save more storage space.

1.3 Research Outcomes

Since Coron's scheme is over integers, his work can be reduced to the AGCD problem, the attacker can recover the noise or public key by lattice reduction. We choose Plantard, Susilo and Zhang's hidden ideal lattice scheme to combine the bounded distance decoding problem (BDD) with approximate greatest common divisor problem. Therefore, the scheme based on the ideal lattice gives a stronger security by the hardness of problems. Coron's technique can be applied on the the Plantard, Susilo and Zhang's hidden ideal lattice scheme, which improves the efficiency exponentially by smaller size of public key. The less public key we publish, the less information of the public key or noise will be leaked. In this work, we are to construct a somewhat homomorphic scheme with public key compression based on hidden ideal lattice. Our approach is summarized as follows. We first generate a random polynomial vector as the ring element, then divide these vectors into two groups. Then, we choose a vector from each group and the product of two polynomial. Therefore, the original public key will be replaced by the new quadratic key. The scheme can reduce the number of public keys from τ keys to $2\sqrt{\tau}$ keys. We also extend the technique into higher degrees to reduce the public key size further.

According to the developing of fully homomorphic encryption, we find the fully

homomorphic encryption is an attraction topic with highly application in the future. There are two main areas need to be concerned, one is security and another one is efficiency. Our aim is to construct a new scheme can improve the efficiency of the fully homomorphic encryption based on the current hard problem which can guarantee the security of the scheme.

Chapter 2

Background

In this section, we will provide the backgroud of research on three disciplines: Lattice, Public-Key Cryptosystems, and Fully Homomorphic Encryption. This section we will emphasize on introducing lattice and Public-Key Cryptosystems. In the next section, we will explain the fully homomorphic encryption.

2.1 Notation

The parameters that are used in the scheme are as follows:

- λ: security parameter.
- ρ: the norm of random noise vector.
- η: the bit length of the norm of generating polynomial (secret vector).
- γ: the bit length of the norm of the random multiplier vector.
- τ: the number of vectors in the public key in encryption algorithm.
- β: the number of vectors in the public key.
- ζ: the norm of noise used in encryption.

- n: the dimension of the hidden lattice.
- θ: the constant factor depending on the polynomial.

For integers z and d, denote $[z]_d$ as the $z \bmod d$ with in $(-d/2, d/2]$ and $\lceil z \rfloor$ as the closest integer to d. Recall the definition of the integer residue ring $\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$, the element in the residue ring is generated by modular operation which in the set $\{0, 1, 2, \ldots, n-1\}$. For $x \operatorname{Mod} n \equiv y$, y is defined as $y \equiv x \bmod n$ with the interval $(-\frac{n}{2}, \frac{n}{2}]$. $\mathcal{D}$ is a distribution by parameter γ and ρ, $\mathcal{D}_{\gamma,\rho}(p) := \{\text{choose } q \leftarrow \mathbb{Z} \cap [0, q_0), e \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho) : \text{output } x = pq + e\}$.

Denote the vector v to represent the coefficient of polynomial $f(x)$. Let the polynomial $f(x)$ in the form of $Vec(f(x)) = \sum_{i=0}^{n-1} v_i x^i$, denote vector $v =< v_1, \ldots, v_n >$, where v_i represents the coefficient of element of x^i. For two vectors v_1 and v_2, denote $v_1 \times v_2$ be the polynomial multiplication over the ring, $v_1 \times v_2 = Vec(v_1(x) \times v_2(x) \bmod f(x)$.

2.2 Lattice

Based on introduction of lattices by Micciancio, Nguyen and Lenstra, this section will give the definition, proofs and properties of lattice.

2.2.1 Definition

Let $\mathbb{R}^n$ be the n-dimensional Euclidean vector. $\mathbf{x}$ and $\mathbf{y}$ are denoted as column vectors like $\mathbf{x}= (x_1, \ldots, x_n)^T$ and $\mathbf{y}= (y_1, \ldots, y_n)^T$ in $\mathbb{R}^n$. The Euclidean inner product is denoted by $< \mathbf{x}, \mathbf{y} > = \sum_{i=1}^{n} x_i y_i$, and the corresponding norm is $\|\mathbf{x}\| = \sqrt{x_1^2 + \cdots + x_n^2}$. The distance between two vectors is $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$. The distance between a vector $\mathbf{x} \in \mathbb{R}^n$ and a subset $E \subset \mathbb{R}^n$ is defined as $dist(\mathbf{x},E) = \min_{\mathbf{y}\in\mathbf{E}}\{d(\mathbf{x},\mathbf{y})\}$.

Definition 2.2.1. (Lattice)[54] Lattice is the set of integer combinations of n linearly independent vectors $v_1, \ldots . v_n$ in $\mathbb{R}^n$. Denote the set of vectors $v_1, \ldots, v_n$ as the basis of the lattice.

$$\mathcal{L}(v_1, \ldots, v_n) = \{\sum_{i=1}^{n} v_i b_i : v_i \in \mathbb{Z} \text{ for } 1 \leq i \leq n\},$$

In the matrix notation, $\mathbf{B} = [\mathbf{v_1}, \ldots, \mathbf{v_n}] \in \mathbb{R}^{\mathbf{n}\times\mathbf{n}}$ denotes as the basis for lattice $\mathcal{L}(\mathbf{B}) = \{\mathbf{Bx} : \mathbf{x} \in \mathbb{Z}^{\mathbf{n}}\}$. The determinant of a lattice is $det(\mathcal{L}) = \sqrt{\mathbf{B} \times \mathbf{B^T}}$.

Definition 2.2.2. (Norm)[54] Let $v =< v_1, \ldots, v_n > \in \mathbb{R}^n$ be the vector of lattice, the Euclidean norm is defined as $\|v\| = \sqrt{\sum_{i=1}^{n} v_i^2}$. For two vectors v_1, $v_2 \in \mathbb{R}^n$, we have $\|v_1 + v_2\| \leq \|v_1\| + \|v_2\|$, and $\|v_1 \times v_2\| \leq \theta \cdot \|v_1\| \cdot \|v_2\|$, where $\theta = \sqrt{n}$.

2.2.2 Ideal Lattice

Definition 2.2.3. (Ideal Lattice)[43] Denote an ideal lattice $\mathcal{L}(Rot(v, f))$ over a polynomial ring $\mathbb{Z}[X]/f$. $f \in \mathbb{Z}[X]$ is a monic irreducible polynomial of degree n. Denote $Rot(v, f)$ is the rotation matrix. The $i-$th row of this matrix equals to the coefficients of $v \times x^{i-1} \mod f$.

Plantard, Susilo and Zhang constructed a fully homomorphic encryption scheme by using hidden ideal lattice [51]. The hidden ideal lattice scheme is unified by two schemes, one is ideal lattice based schemes and the other one is integer based schemes. The security of this scheme does not rely on the sparse subset sum problem (SSSP), but rather, it relies on the bounded distance decoding problem (BDD) of ideal lattices.

Definition 2.2.4. (Hidden Ideal Lattice)[51]

Let $v_i \in \mathbb{Z}^n$ be τ integer vectors and $\alpha \in \mathbb{R}^+$ be a positive real. There exists a unique (ideal) lattice $\mathcal{L}$ and some unique vectors $v_i \in \mathcal{L}$ respecting $\forall 1 \leq i \leq \tau$, $dist(v_i, w_i) \leq \alpha$. Then $\mathcal{L}$ is called an $\alpha-$hidden ideal lattice hidden under $\{v_i\}$.

Denoted $dist(v, L) = min(\|v - u\|)$, $\forall u \in \mathcal{L}$ as the distance between the a vector $v \in \mathbb{R}^n$ and a lattice $\mathcal{L}$.

Definition 2.2.5. (BDD over Hidden Ideal Lattice) [51]

Let $\gamma \in \mathbb{R}^+$ be a positive real and L be an n dimensional ideal lattice. $v \in \mathbb{Z}$ is a random vector, there exists a unique vector $u \in \mathcal{L}$ satisfying $dist(v, u) \leq \gamma$. The hard problem $\gamma - BDDHI_n$ is called γ-Bounded Distance Decoding problem over ideal lattice. For given a basis of $\mathcal{L}$ and v, find u.

2.3 Hard Problems

To prove the security of algorithms, we introduce several classic computational problems which are based on lattices computation in this section. Then we review the lattice application on cryptosystem with LWE and R-LWE.

Definition 2.3.1. (Classic Lattice Problems) [42], [52]

- **Decisional Shortest Vector Problem** (GapSVP_γ): $\mathcal{L}$ is a full-rank n-dimensional lattice, for a given basis $\mathbf{B}$, decide if $\lambda_1(\mathcal{L}) \leq 1$ or $\lambda_1(\mathcal{L}) > \gamma(n)$.
- **Shortest Independent Vectors Problem** (SIVP_γ): $\mathcal{L}$ is a full-rank n-dimensional lattice, for a given basis $\mathbf{B}$, find a set of linearly independent vectors $\mathbf{S} = \{\mathbf{s_1}, \ldots, \mathbf{s_n}\}$, for $\mathbf{s_i} \in \mathcal{L}(\mathbf{B})$, minimizing the quantity $\|\mathbf{S}\| = \max_{\mathbf{i} \in [\mathbf{n}]} \|\mathbf{s_i}\|$.

Definition 2.3.2. (Modern Lattice Problems) [52], [54]

- **Small Integer Solutions** (SIS_β): Given a prime q, a random matrix $\mathbf{A} \in \mathbb{Z}_\mathbf{q}^{\mathbf{n} \times \mathbf{m}}$ and a real number β. Find a non-zero vector $\mathbf{d} \in \mathbb{Z}^\mathbf{m}$ such that $\mathbf{Ad} = \mathbf{0} \mod \mathbf{q}$ and $\|d\| \leq \beta$. Note, finding a solution of SIS_β can be seen as finding a short lattice point in lattice $\Lambda_q^\perp(\mathbf{A})$.

Lemma 2.3.1. *(Average-case to Worst-case) [45], [53] Let $n, p \geq 1$ be some integers and χ be some disribution on $\mathbb{Z}_p$. Assume that we have access to a distinguisher W that distinguishes $A_{s,\chi}$ from U for a non-negligible fraction of all possible s, then there exists an efficient algorithm W' that for all s accepts with probability exponentially close to 1 on inputs from U.*

Lemma 2.3.2. *(Decision to Search) [53] Let $n \geq 1$ be some integers, $2 \leq p \leq poly(n)$ be a prime, and χ be some distribution on $\mathbb{Z}_p$. Assume that we have access to procedure W that for all s accepts with probability exponentially close to 1 on inputs from $A_{s,\chi}$ and rejects with probability exponentially close to 1 on inputs from U. Then, there exists an efficient algorithm W' that, given samples from $A_{s,\chi}$ for some s, outputs s with probability exponentially close to 1.*

Lemma 2.3.3. *(Discrete to Continuous) [53] Let $n, p \geq 1$ be some integers, let ϕ be some probability density function on $\mathbb{T}$, and let $\bar{\phi}$ be its discretisation to $\mathbb{Z}_p$. Assume that we have access to an algorithm W that solves $\boldsymbol{LWE}_{p,\bar{\phi}}$. Then, there exists an efficient algorithm W' that solves $\boldsymbol{LWE}_{p,\phi}$.*

2.3.1 Learning With Error (LWE)

Based on worst-case hardness assumption, the "learning with error" problem is a classical problem which is to distinguish random linear equations with small amount of noise from uniform ones. The main theory is to recover a secret key $s \in \mathbb{Z}_q^n$ given a sequence of "approximate" random linear equations on s. If without error, the secret key s can be found by gaussian elimination in polynomial time. Introducing the small error perturbs the linear combinations into nonlinear combinations, then the gaussian elimination algorithm seems impossible to solve the problem directly

Definition 2.3.3. (GLWE) [5] For security parameter λ, let $n = n(\lambda)$ is the dimension, and the polynomial $f(x) = x^d + 1$ with d is power of 2, fix $q = q(\lambda) \leq 2$ as a prime

integer, let $R = \mathbb{Z}[x]/(f(x))$ and $R_q = R/qR$, and let $\chi = \chi(\lambda)$ be a distribution over R. The $GLWE_{n,f,q,\chi}$ problem is to distinguish the following two distributions: In the first distribution, one samples (a_i, b_i) uniformly from R_q^{n+1}. In the second distribution, one first draws $s \leftarrow R_q^n$ uniformly and then samples $(a_i, b_i) \in R_q^{n+1}$ by sampling $a_i \leftarrow R_q^n$ uniformly, $e_i \leftarrow \chi$, and setting $b_i =< a_i, s > +e_i$. The $GLWE_{n,f,q,\chi}$ assumption is that the $GLWE_{n,f,q,\chi}$ problem is infeasible.

LWE is simply GLWE instantiated with $d = 1$ and RLWE is GLWE instantiated with $n = 1$. The brief description of the LWE problem is, given a size parameter $n \geq 1$ and a modulus $q \geq 2$, also given an error probability distribution χ on $\mathbb{Z}_q$. Choose a random vector $\alpha \in \mathbb{Z}_q^n$ uniformly and $e \in \mathbb{Z}_q$ according to χ, then output $(\alpha, < \alpha, s > +e)$ in $\mathbb{Z}_q$. The $\mathbb{A}_{s,\chi}$ consists of independent and uniform random α. The LWE problem can be considered as decoding from random linear codes. On the lattice view, the LWE problem is decoding the code by a random bounded distance [39]. The maximum likelihood algorithm is a way to solve the LWE problem with the running time $2^{\mathcal{O}(n \log n)}$. The best known algorithm for the LWE problem is Blum et al. algorithm, and the running time is $2^{\mathcal{O}(n)}$ [9], [32], [47], [59]. We believe the LWE problem is hard for several reasons. The well known algorithm for solving the LWE problem is runs in exponential time. Next, the LWE problem is based on certain assumptions regarding the worst-case hardness of standard lattice problems such as GapSVP and SIVP on lattices. The 'dual' problem of LWE problem is the SIS problem. The hardness of the SIS problem is also based on the worst-case lattice problem such as SIVP and GapSVP [44], [45], [49].

Theorem 2.3.4. *(Informal) [53] Let n, p be integers and $\alpha \in (0, 1)$ be such that $\alpha p > 2\sqrt{n}$. If there exists an efficient algorithm that solves $\mathbf{LWE}_{p,\tilde{\Psi}_\alpha}$ then there exists an efficient quantum algorithm that approximates the decision version of the shortest vector problem (GapSVP) and the shortest independent vectors problem (SIVP) to*

within $\tilde{\mathcal{O}}(n/\alpha)$ in the worst case.

2.3.2 Ring-Learning With Error (R-LWE)

Normal LWE problem is defined over integers. It turns out the LWE problem over some special rings is also hard. Due to the large size of key over the integers, the ring structure allows LWE-based cryptosystems to have a shorter public key size, which is reduced to almost linear size from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ [41]. The structure in the Ring-LWE of the NTRU cryptosystem is as follows. Choose a random vector $\alpha_1 = (a_1, a_2, \ldots, a_n)$ uniformly on the ring $\mathbb{Z}_q[x]/ < x^n + 1 >$, the rest of $n-1$ vectors $(a_2, \ldots, a_n)$ in the form of $a_i = (a_i, a_{i+1}, \ldots, a_n, -a_1, \ldots, -a_{i-1})$ with n is the power of 2. Some schemes do not require the $x^n + 1$ has to be irreducible [41]. Fix the $s \in \mathbb{R}_q$ as the secret key, $\alpha \in \mathbb{R}_q$ chosen uniformly, and e is an error chosen from the distribution over $\mathbb{R}_q$. The sample forms as $(\alpha, b = \alpha \cdot s + e) \in \mathbb{R}_q \times \mathbb{R}_q$, where each α is uniformly random and each inner product $\alpha \cdot s$ is perturbed by a term draw independently from the error distribution over $\mathbb{R}$ [41].

The hardness of the Ring-LWE problem is based on the worst-case lattice problem. The goal is to recover the secret s from these samples. The main theory [49] is

Theorem 2.3.5. *Suppose that it is hard for polynomial-time quantum algorithms to approximate (the search version of) the shortest vector problem (SVP) in the worst case on ideal lattices in $\mathbb{R}$ to within a fixed poly(n) factor. Then any poly(n) number of samples drawn from the R-LWE distribution are pseudorandom to any polynomial-time attacker.*

2.3.3 Ideal Lattice Hard Problem

The Hidden Ideal Lattice Problem is defined as follows: given some vectors close to a (ideal) lattice, find such a lattice.

Definition 2.3.4. (Dec BDD over Hidden Ideal Lattice) [51] Let $\gamma \in \mathbb{R}^+$ be a positive real. Let L be an n dimensional ideal lattice, and $v \in \mathbb{Z}$. The decisional γ-Bounded Distance Decoding problem over ideal lattice, denoted by Dec $\gamma - BDDHI_{n,\tau}$, is to decide if there exists a unique vector $u \in \mathbb{L}$ satisfying $dist(v, u) \leq \gamma$. or not, given a basis of $\mathcal{L}$ and v.

Definition 2.3.5. (Subset Sum Problem) [51] Let $\{c_1, c_2, \ldots, c_n\}$ be a set of positive integers. Let $c = \sum_{i=1}^{n} s_i c_i$, where $s_i \in \{0, 1\}$. Let $d \leftarrow \sum_{i=1}^{n} s_i$. The subset sum problem, denoted by d, n-SSP, is to find $\{s_i\}$, given $\{c_i\}$ and c.

Gentry choose to use ideal lattice as mathematics tool to construct Fully Homomorphic Encryption scheme [26]. Because Ideal lattice has simple operation in decryption algorithm, such as vector scalar product and vector inner product of matrix, which has lower complexity in decryption circuit. Another reason to choose ideal lattice is, ideal lattice satisfies both additional and multiplication homomorphism.

2.4 Public Key Cryptography

Public key cryptography is asymmetric cryptography. Any cryptographic system creates public key and private key in pairs. Private keys are known to the owner only but public keys can be public widely. Everyone can encrypt plaintext by using public key, and the ciphertext can be decrypted by owner's private key. To strength on a public key cryptography system should be increased the difficulty of generating a private key from its corresponding public key. Since the public key may be published without compromising security, keeping the private key private is the key point of the security of public key cryptography systems [62]. So far, public key cryptographic algorithms are based on mathematical problems, such as integer factorization, discrete logarithm, and elliptic curve relationships.

Diffie and Hellman [19] introduced a new approach for distributing the key information over public insecure channels. Public key cryptosystem also known as asymmetric key cryptosystem. Each user has two keys, one is *a public encryption key* another is *a private decryption key.* The public key is generated by the secret key with different mathematical techniques, while the private key cannot be generated by the public key. In other words, everybody knows the public key but nobody except owner knows the secret key. Messages are encrypted with the any public key, the ciphertext can only be decrypted with the corresponding private key relative to public key [57]. The first implementation of the idea is published by Rivest, Shamir and Adleman in 1977. It is also known as the *RSA algorithm* .

The LWE problem has been widely used in public key cryptosystem [55]. There is an example of the application of LWE with a public key cryptosystem, but the efficiency needs to be improved.

- **Private Key**: Choose a random vector s uniformly from $\mathbb{Z}_q^n$.
- **Public Key**: Choose m samples $(\alpha_i, b_i)_{i=1}^m$ from the LWE distribution. The error distribution is generated by a function with secret s, modulus q, and error parameter α.
- **Encryption**: To encrypt each single binary bit of the message. Choose a random set S uniformly among all 2^m subsets of $[m]$. To encrypt bit 0, use $(\sum_{i\in S}\alpha_i, \sum_{i\in S} b_i)$, and to encrypt bit 1, use $(\sum_{i\in S}\alpha_i, \lfloor\frac{q}{2}\rfloor + \sum_{i\in S} b_i)$. Generate the ciphertext pair (α, b).
- **Decryption**: The decryption method uses approximately analysis. For each cipertext pair (α, b) with secret key, if $b- < \alpha, s >$ is closer to 0 than $\lfloor\frac{q}{2}\rfloor$ modulo q, the decryption is 0, and if $b- < \alpha, s >$ is closer to $\lfloor\frac{q}{2}\rfloor$ modulo q than 0, the decryption is 1.

The correctness is obvious. It requires the error of s smaller than $q/4$, since each error's standard deviation is αq, the standard deviation of the sum is at most $\sqrt{m}\alpha q < q/\log n$. The distribution over s of a random subset sum $(\sum_{i\in S}\alpha_i, \sum_{i\in S} b_i)$ is approximately to be considered uniform in statistical distance, since the choice of $(\alpha_i, b_i)_{i=1}^m$ [32] has high probability. The encryptions of 0 or 1 is essentially identically distributed, the algorithm assuming decision LWE is hard [32], [49]

The R-LWE is more widely and efficiently used in the real world. There is an example of public key cryptosystem satisfying the semantically secure by Lyubashevsky, Peikert and Regev [41]. Fix the ring $\mathbb{R} = \mathbb{Z}[x]/ < x^n + 1 >$ with the n is a power of 2.

- **Secret Key**: Choose a random vector s uniformly from $\mathbb{R}$.
- **Public Key**: Choose a uniformly random ring element $\alpha \in \mathbb{R}_q$, and another small element $e \in \mathbb{R}$ from the error distribution. Output a pair $(a, b = a \cdot s + e)$ as the public key.
- **Encryption**: To encrypt a n-bits binary message, choose three random elements $r, e_1, e_2 \in \mathbb{R}$ from the Gaussian distribution. Output the ciphertext in pair $(u, v) \in \mathbb{R}_q^2$, where

$$u = a \cdot r + e_1 \text{ and } v = b \cdot r + e_2 + \lfloor q/2 \rceil \cdot z \mod q$$

- **Decryption**: The decryption computes

$$v - u \cdot s = (r \cdot e - sc \cdot e_1 + e_2) + \lfloor q/2 \rceil \cdot z \mod q.$$

This application requires the coefficient of the error item $r \cdot e - s \cdot e_1 + e_2 \in \mathbb{R}$ must be less than $q/4$ [41].

Chapter 3

Fully Homomorphic Encryption

3.1 The Definition of Fully Homomorphic Encryption

Fully Homomorphic Encryption is a way to delegate processing of your data, without giving away access to it. There are four algorithms: **KeyGen**, **Encrypt**, **Decrypt** and **Evaluate**. The first three are the basic algorithms of a public key encryption system and the evaluation algorithm is the core algorithm of fully homomorphic encryption which performing the operation of ciphertext. The algorithm is inputting a group ciphertext $\mathbf{c} =< c_1, c_2, \ldots, c_t >$ which has been encrypted for plaintext into a circuit $\mathcal{C}$, each circuit $\mathcal{C}$ represents a function. The group of new output ciphertext can be decrypted to the corresponding plaintext.

Definition 3.1.1. (Correct Homomorphic Decryption) [25]

The **KeyGen** algorithm generates many key-pair (pk, sk). After performing t-input circuit, for any plaintext $m_1, \ldots, m_t$ and any ciphertext $c_1, \ldots, c_t$, which are

generated by algorithm $c_i \leftarrow \mathbf{Encrypt}(m_i)$. If

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

exists, then scheme $\mathcal{E}$ =(**KeyGen**, **Encrypt**, **Decrypt**, **Evaluate**) is *correct*.

Definition 3.1.2. (Compact Homomorphic Decryption) [26] For any security parameter λ, if there exists a polynomial f, the output length of the evaluate algorithm is at most $f(\lambda)$. The scheme $\mathcal{E}$ =(**KeyGen**, **Encrypt**, **Decrypt**, **Evaluate**) is *compactness*.

Definition 3.1.3. (FHE - Fully Homomorphic Decryption) [26] A scheme $\mathcal{E}$ is fully homomorphic if it is compact and homomorphic for the arithmetic circuit.

Definition 3.1.4. (SHE - Somewhat Homomorphic Decryption) [26] The encryption scheme can handle circuits of depth roughly $\log\log N - \log\log n$, which means the minimum depth of the permit circuits is greater than twice of the depth of decryption circuit.

Definition 3.1.5. (Leveled Fully Homomorphic Decryption) [26] For any $d \in \mathbb{Z}^+$, the scheme $\mathcal{E}^{(d)}$ with the same decryption circuit is compactness and the depth of circuit is d. The complexity of the scheme $\mathcal{E}^{(d)}$ is polynomial with parameter λ, d. A family of schemes $\mathcal{E}^{(d)}$ is leveled homomorphic.

3.2 The Construction of FHE

The crucial point to construct a fully homomorphic encryption scheme is able to evaluate polynomials of higher degree, in other words, consider the decryption procedure as a polynomial with lower degree. When the degree of the decryption polynomial

is less than the degree of the polynomial which is used for the evaluation of scheme, therefore, the scheme is a fully homomorphic scheme [26].

There is no natural fully homomorphic encryption scheme so far, the majority of schemes are constructed by Gentry's idea. Firstly, construct a somewhat homomorphic encryption scheme which is a linear code C on the ring $\mathcal{R}$. Linear code satisfies the additive homomorphism and error correcting code means the code with an error. Since C is an ideal of the ring, it satisfies the multiplicative homomorphism. The code C has two kinds of basis, one is 'good' basis which can be used as a secret key, another basis is 'bad' for public key [26].

Since the error of the ciphertext will be expanded over the bound especially in multiplication, a failure in the decryption occurs. Gentry used 'homomorphic decryption' to control the noise increasing. Encrypt the ciphertext and the corresponding public key with the evaluate key, and input the result into the decryption circuit, output a new ciphertext. If the error of the ciphertext is able to evaluate one more time especially in multiplication after each operation, then the ciphertext can perform the operation unlimited times[28], [30]. Since the somewhat homomorphic encryption scheme can only perform limited times operations with low-degree polynomials , the second step is to squash the decryption algorithm to support the scheme, which means convert into the low-degree polynomial. Finally the application of a bootstrapping can transform the somewhat homomorphic encryption scheme to a fully homomorphic scheme [26].

3.3 The Security of FHE

The SHE and FHE is secure against chosen plaintext attacks. But no SHE and FHE scheme can be IND-CCA2 secure, based on the fact that the adversary is allowed to manipulate the challenged ciphertext and submit it to the decryption oracle in an

IND-CCA2 attack. The IND-CCA1 has been proved to be not secure for FHE and SHE scheme [40]. Zhang et. al provided a way to recover the secret key by using the decryption oracle over the DGHV scheme [66] [67]. Chenal gave more algorithms to allow an adversary to recover the pubic keys through decryption oracle queries [11].

3.4 Technique of Fully Homomorphic Encryption

The key point to construct fully homomorphic encryption is how to control the increase of the noise, there are some techniques like bootstrapping, key switching and modulus switching.

3.4.1 Bootstrapping

The fully homomorphic decryption requires the depth of decryption circuit less than the depth of the decryption circuit of the evaluate algorithm. In fact, the depth of decryption of circuit is greater than the depth of the decryption circuit of evaluate algorithm. Using 'sub set sum phrase' is the way to squash the depth of decryption circuit [26].

3.4.2 Homomorphic Decryption

Homomorphic decryption can generate new ciphertext and reduce the error of ciphertext with conditions. Let $\mathbf{Encrypt}(pk_1, m) \rightarrow c_1$ and $\mathbf{Encrypt}(pk_2, sk_{1j}) \rightarrow \bar{sk_1}$, the algorithm of homomorphic decryption is:

$$\mathbf{Recrypt}(pk_2, D, \bar{sk_1}, c_1)$$

$$\mathbf{Encrypt}(pk_2, c_{1j}) \rightarrow \bar{c_1}$$

$$\mathbf{Evaluate}(pk_2, D, \bar{sk_1}, \bar{c_1}) \rightarrow c_2$$

D is the circuit of the decryption algorithm [26]. It decrypts the ciphertext after first encryption which eliminates the error, then encrypted by new public key to get new ciphertext with new error. If the new error allows one more multiplication, in other words, the error is still within the bound after multiplication, then the goal of homomorphic decryption is achieved [26].

3.4.3 Key Switching

Key switching technique is based on the LWE of R-LWE, it can generate a new ciphertext corresponding to the different secret keys and reduce the dimension of ciphertext [5]. The new ciphertext c_2 is formed by a matrix M multiplying the fresh ciphertext c_1. The row of matrix M is the dimension of c_1 and the column of M is the dimension of c_2. The technique transform c_1 with dimension n_1 to c_2 with dimension n_2 with the same modulus, the error of c_2 increases $< \mathbf{BitDecomp}(c_1), e_2 >$ than the error of c_1. The algorithm is:

$$\mathbf{SwitchKeyGen}(s_1 \in R_q^{n_1}, s_2 \in R_q^{n_2}) : \mathbf{A} \leftarrow \mathbf{E.PublicKeyGen}(s_2, N)$$

$$\mathbf{B} \leftarrow \mathbf{A} + \mathbf{Powersof2}(s_1),\ where\ N = n_1 \cdot \lceil \log q \rceil$$

$$output\ \tau_{s_1 \to s_2} = \mathbf{B}$$

$$\mathbf{SwitchKey}(\tau_{s_1 \to s_2}, c_1) : \qquad output\ c_2 = \mathbf{BitDecomp}(c_1)^T \cdot \mathbf{B} \in r_q^{n_2}$$

3.4.4 Modulus Switching

Let the modulus be $q = x^k$, and each ciphertext with error x, the new error is approximately x^2 after multiplication. The error will reach the bounds of the decryption circuit after $\log k$ levels multiplicative. If the error times $1/x$ after each operation, the error will be reduced to the original value, meanwhile, the modulus decrease to q/x [5].

The Iteration can perform k levels without bootstrapping before reaching the bound of error. The algorithm is :

$$\textbf{Scale}(c, q, p, r) : \ \textit{input } s,\ q \textit{ and } p \textit{ with } (q > p > m)$$
$$\textit{output } (p/q) \cdot c \textit{ and } c' = c \mod r$$

3.4.5 Chinese Reminder Theorem

$p_1, \ldots, p_2$ are pairwise co-prime integers and $CRT_{(p_1,\ldots,p_2)}(m_1, \ldots, m_k)$ is a number in $\mathbb{Z} \cap (-\frac{\pi}{2}, \frac{\pi}{2}]$, where $\pi = \prod_{i=1}^{k} p_i$. $CRT_{(p_1,\ldots,p_2)}(m_1, \ldots, m_k)$ is equivalent to $m_i \bmod p_i$ for all $i \in \{1, \ldots, k\}$. So we have

$$CRT_{(p_1,\ldots,p_2)}(m_1, \ldots, m_k) = \sum_{i=1}^{k} m_i M_i (M_i^{-1} \bmod p_i) \bmod \pi$$

where $M_i = \frac{\pi}{p_i}$.

The distributions of single bit with single private key is

$$\mathcal{D}_{\gamma,\rho}(p) := \{\text{choose } q \leftarrow \mathbb{Z} \cap [0, \frac{2^\gamma}{p}),\ e \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho) : \text{ output } x = pq + e\}$$

The distributions of ℓ_Q-bits with multi-private keys is

$$\mathcal{D}_\rho(p_1, \ldots, p_k; Q_1, \ldots, Q_k; q_0) := \{\text{choose } e_0 \leftarrow \mathbb{Z} \cap [0, q_0), e_i \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho)$$
$$\text{output } x = CRT_{(q_0,p_1,\ldots,p_k)}(e_0, e_1 Q_1, \ldots, e_k Q_k), \text{for } \forall i \in \{1, \ldots, k\}\}$$

Consider the value of x when $k = 1$, since $\mathcal{D} := \{\text{choose } q \leftarrow \mathbb{Z} \cap [0, q_0), e \leftarrow$

$\mathbb{Z} \cap (-2^\rho, 2^\rho)$: output $x = p_1 q + e \bmod p_1 q_0\}$, there is $x \leftarrow \mathcal{D}_\rho(p_1; q_0)$.

$$\begin{aligned} x &= CRT_{(q_0;p_1)}(e_0, e_1) \\ &= e_0 p_1 (p_1^{-1} \bmod q_0) + e_1 q_0 (q_0^{-1} \bmod p_1) \bmod q_0 p_1 \\ &= e_0 p_1 \alpha + e_1 (p_1 \beta + 1) \bmod q_0 p_1 \\ &= (e_0 \alpha + e_1 \beta) p_1 + e_1 \bmod q_0 p_1 \end{aligned}$$

for some α and β. Since $e_0 \not\equiv \bmod q_0$ and $\gcd(\alpha, q_0) = 1$, $(e_0\alpha + e_1\beta) \bmod q_0$ is uniformly in $\mathbb{Z} \cap [0, q_0)$.

3.4.6 Public Key Compression

The implementation of the DGHV fully homomorphic encryption scheme has a large size of public key in $\widetilde{\mathcal{O}}(\lambda^{10})$. To resist lattice attack, each public key needs at least 2^{23} bits, the size of the public key will be 2^{46} bits, it is too large for a practical system. Coron [16] presented an efficient way to compress the public key of the DGHV scheme, by using quadratic form instead of linear form when computing a ciphertext:

- KeyGen(λ):

  Pick a random prime $p \in [2^{\eta-1}, 2^\eta)$. Let $x_0 = q_0 \cdot p$ where q_0 is a random square free $2^\lambda-$rough integer in $[0, 2^\gamma/p)$. Generate integers $x_{i,b} \leftarrow p \cdot q_{i,b} + r_{i,b}$, where $1 \le i \le \beta$, $b \in \{0, 1\}$, $q_{i,b}$ is a random integer in $[0, q_0)$ and $r_{i,b}$ is a random integer in $(-2^\rho, 2^\rho)$. Output $sk = p$ and $pk =< x_0, x_{1,0}, x_{1,1}, \ldots, x_{\beta,0}, x_{\beta,1} >$.

- Encrypt(pk,m):

  Input a random vector $b = (b_{i,j}) \in [0, 2^\alpha)$, of size $\tau = \beta^2$. Generate a random integer $r \in (-2^\rho, 2^\rho)$. Output a ciphertext $c \leftarrow m + 2r + 2\sum_{1 \le i,j \le \beta} b_{i,j} \cdot x_{1,0} \cdot x_{j,1}$

- Decrypt and Evaluate:

  It is the same as the original scheme but with modulus x_0 after addition and multiplication.

The scheme can extend into higher degrees [17]. Use the same way to generate elements and encrypt the plaintext as follow:

$$c^* = m + 2r + 2 \sum_{1 \le i,j \le \beta} b_{ij} \dots x_{i,0} \dots x_{j,1} \mod x_0$$

The authors proved the scheme is semantically secure under the error-free approximate GCD assumption. They applied the leftover hash lemma on hash function family $h' : [0, 2^\alpha)^{\beta^2} \to \mathbb{Z}_{q_0}$, where $h'(b) = \sum_{1 \le i,j \le \beta} b_{i,j} \cdot q_{i,0} \cdot q_{j,1} \mod q_0$.

Definition 3.4.1. (Hash Function) [16] A family $\mathcal{H}$ of hash function $h : X \to Y$ is $\varepsilon-$pairwise independent if

$$\sum_{x \ne x'} (Pr_{h \leftarrow \mathcal{H}}[h(x)) = h'(x)] - \frac{1}{Y}) \le |X|^2 \cdot \frac{\varepsilon}{|Y|}$$

Lemma 3.4.1. *(Leftover Hash Lemma) [16] Let $\mathcal{H}$ be a family of $\varepsilon-$pairwise independent hash functions. Choose random $h \leftarrow \mathcal{H}$ and $x \leftarrow X$ uniformly and independently. Then $(h, h(X))$ is $(\frac{1}{2}\sqrt{|Y|/|X| + \varepsilon})-$uniformly over $\mathcal{H} \times Y$*

The key element $x_{i,b}$ have been proved that a certain family of quadratic hash function is close enough to be pairwise independent, so this can apply the leftover hash lemma. The significance of this method is reducing the size of public key from $\tau = \widetilde{\mathcal{O}}(\lambda^3)$ down to $2\beta = \widetilde{\mathcal{O}}(\lambda^{1.5})$.

The semantic security of the scheme based on approximate-GCD assumption with error-free x_0. The adversary can find the exact multiple p by solving the AGCD problem. The known attack had been presented on van Dijk's paper [20]

Definition 3.4.2. (AGCD Problem) [51] Let $c_i \in \mathbb{Z}$, there exists τ unique integers $r_i \in \mathbb{Z}$ and a unique integer $p \in \mathbb{N}$. For $\forall i$, there is $(c_i - r_i)|p$ and $|r_i| \leq \gamma \leq p/2$. The Approximate Greatest Common Divisor problem denoted as, for given c_i, find p.

3.5 Existing Fully Homomorphic Encryption Schemes

Many SHE and FHE schemes have been proposed after Gentry's work. These schemes can be classified based on different hardness assumptions as in figure 3.1 [11].

- The first category is based on hard problems on lattices that starts with Gentry's work[Gen09a,Gen09b] [25], [26], [63], [27], [64], [27].
- The second category relies on the approximate greatest common divisor (AGCD) problem and some variants. The typical scheme is [vDGHV10] [15], [20].
- The third category is based on the learning with errors (LWE) and on the ring-learning with errors (RLWE) problems like schemes [NLV11,BGV12,GHS12b,Bra12,GSW13] [4], [5], [7], [6], [31].

3.5.1 Gentry's First Fully Homomorphic Encryption Scheme

Gentry used ideal lattices to construct the fully homomorphic encryption scheme of PKE [26]. A ciphertext ψ is in form of $v+x$ when v is the ideal lattices and x is the error distribution. The coefficient of ciphertext vectors is the elements in a polynomial ring $\mathbb{Z}[x]/f(x)$. The addition and multiplication of ciphertext satisfy the ring operation. The security of the scheme is based on finding the closest vector problem for ideal lattices in a ring.

Gentry's initial somewhat homomorphic encryption scheme is based on lattice [25].

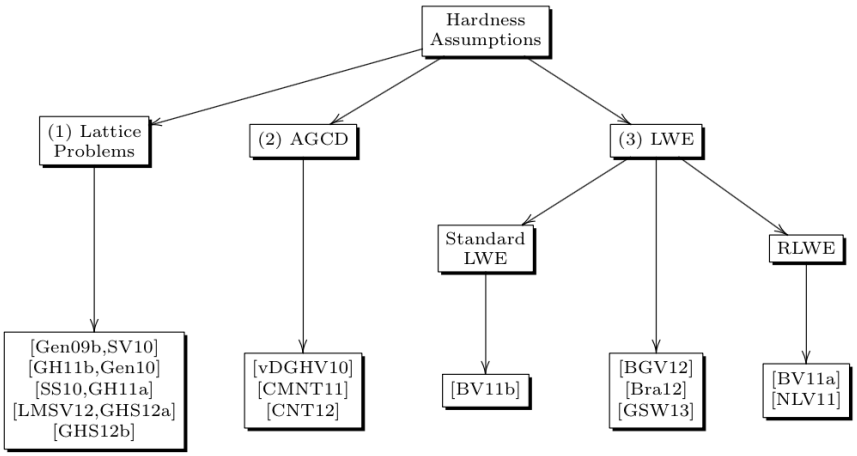

Figure 3.1: Hardness assumptions and schemes

- $\mathbf{KeyGen}_{\mathcal{E}}(R, \mathbf{B_I})$:

  Input a ring R and a basis $\mathbf{B_I}$ of lattice $\mathbf{I}$. It sets $(\mathbf{B_J^{sk}}, \mathbf{B_J^{sk}}) \leftarrow \mathbf{IdealGen}(\mathbf{R}, \mathbf{B_I})$.

  The public key has $R, \mathbf{B_I}, \mathbf{B_J^{pk}}$ and **Samp**, where **Samp** is an algorithm which sampling basis from the coset of lattice.

  The secret key is $\mathbf{B_J^{sk}}$.

- $\mathbf{Encrypt}_{\mathcal{E}}(pk, m)$:

  Input a public key pk and a plaintext $m \in \mathcal{P}$. It sets $\psi' \leftarrow \mathbf{Samp}(\mathbf{m}, \mathbf{B}_I, R, \mathbf{B}_J^{pk})$

  Output a ciphertext $\psi' \leftarrow \psi \mod \mathbf{B}_J^{pk}$.

- $\mathbf{Decrypt}_{\mathcal{E}}(sk, \psi)$:

  Input the secret key and a ciphertext ψ.

  Output $m \leftarrow (\psi \mod \mathbf{B}_J^{sk}) \mod \mathbf{B}_I$

- $\mathbf{Evaluate}_{\mathcal{E}}(pk, C, \Psi)$:

  Input the public key pk and a circuit C consist of $\mathbf{Add}_{\mathbf{B}_I}$ and $\mathbf{Mult}_{\mathbf{B}_I}$, and a

set of ciphertext ψ .

Output new ciphertext ψ.

Add(pk, ψ_1, ψ_2). Output $\psi_1 + \psi_2 \mod \mathbf{B}_J^{pk}$.

Mult(pk, ψ_1, ψ_2). Output $\psi_1 \times \psi_2 \mod \mathbf{B}_J^{pk}$.

This somewhat homomorphic scheme without bootstrapping is not a fully homomorphic encryption function. The bootstrapping is the way to squash the decryption procedure, since reducing the degree of the decryption polynomial, decryption can be performed as many times as required. Gentry hid the public key in the form of a sparse subset-sum problem (SSSP). The public key becomes a big set of vectors [27]. A ciphertext of the scheme can be decrypted with a low-degree polynomial. The scheme with bootstrapping is the fully homomorphic scheme $\mathcal{E}^{(d)}$ with security parameter λ which can handle all circuits of depth d is given:

- **KeyGen**$_{\mathcal{E}^{(d)}}(\lambda, d)$:

  $(sk_i, pk_i) \leftarrow$ **KeyGen**$_{\mathcal{E}}(\lambda)$ for $i \in [0, d]$

  $\bar{sk}_{ij} \leftarrow$ **Encypt**$_{\mathcal{E}}(pk_{i-1}, sk_{ij})$ for $i \in [1, d], j \in [1, \ell]$

  $sk^{(d)} \leftarrow sk_0, \qquad pk^{(d)} \leftarrow (< pk_i >, < \bar{sk}_{ij} >)$

- **Encrypt**$_{\mathcal{E}^{(d)}}(pk^d, m)$:

  Input a public key $pk^{(d)}$ and a plaintext $m \in \mathcal{P}$,

  Output a ciphertext $\psi \leftarrow$ **Encrypt**$_{\mathcal{E}}(pk_d, m)$.

- **Decrypt**$_{\mathcal{E}^{(d)}}(sk^{(d)}, \psi)$:

  Input a secret key $sk^{(d)}$ and a ciphertext ψ.

  Output **Decrypt**$_{\mathcal{E}}(sk_0, \psi)$.

- **Evaluate**$_{\mathcal{E}^{(\delta)}}(pk^{(\delta)}, C_{(\delta)}, \Psi_{(\delta)})$:

  Input the public key $pk^{(\delta)}$, a circuit $C_{(\delta)}$ of depth at most δ, and a tuple of ciphertext $\Psi_{(\delta)}$.

Output a new tuple of ciphertext $\Psi_{(\delta-1)}$ until $\delta = 0$ and terminates.

Set $(C^{\dagger}_{\delta-1}, \Psi^{\dagger}_{\delta-1}) \leftarrow \mathbf{Augment}_{\mathcal{E}^{(\delta)}}(pk^{(\delta)}, C_{(\delta)}, \Psi_{(\delta)})$.

Set $(C_{\delta-1}, \Psi_{\delta-1}) \leftarrow \mathbf{Reduce}_{\mathcal{E}^{(\delta-1)}}(pk^{(\delta-1)}, C^{\dagger}_{(\delta-1)}, \Psi^{\dagger}_{(\delta-1)})$.

Runs $\mathbf{Evaluate}_{\mathcal{E}^{(\delta-1)}}(pk^{(\delta-1)}, C_{(\delta-1)}, \Psi_{(\delta-1)})$.

Unfortunately, Gentry's scheme has inherent efficiency limitations. In the decryption circuit, the original secret-key has been encrypted into a large ciphertext. The complexity of the scheme is extremly large, which will be defined as the bit-length of the individual ciphertexts that are used to encrypt the bits of the secret key times the complexity of the decryption[28]. The bottleneck in practice is the time of per-gate evaluation.

3.5.2 Dijk, Gentry, Halevi and Vaikuntanathan's Scheme Over The Integers (DGHV)

Dijk, Gentry, Halevi and Vaikuntanathan published a fully homomorphic encryption scheme over the integer rather than on ideal lattice [20].The construction of the somewhat fully homomorphic scheme consist of:

- KeyGen(λ): Secret Key: choose random odd η- bits integer p: $p \leftarrow (2\mathbb{Z}+1) \cap [2\eta\cap, 2\eta)$.

  Public Key: sample uniformly $x_i \leftarrow \mathcal{D}_{\gamma}, \rho(p)$, for $i = 1, 2, \ldots, \tau$. The odd integer x_0 has to be the largest number and the remainder of $x_0 \mod p$ is even.

  Output $sk = p$ and $pk =< x_1, x_2, \ldots, x_{\tau} >$.

- Encrypt(pk,m): Input a random subset $S \subset 1, \ldots, \tau$, a random integer r in $(-2^{\rho'}, 2^{\rho})$ and a plaintext $m \in 0, 1$

  Output a ciphertext $c \leftarrow [m + 2r + 2\sum_{i \in S} x_i]_{x_0}$.

- Decrypt(sk,c): Input a secret key sk and a ciphertext c.

  Output $m \leftarrow (c \mod p) \mod 2$.

- Evaluate($pk, C, c_1, \ldots, c_\tau$): Input t ciphertext c_i as t inputs to the binary circuit $C_\mathcal{E}$, apply addition and multiplication gates of $C_\mathcal{E}$ on ciphertext.

  Output the integer of operation result.

The noise expands quickly especially under multiplication. Assuming the bound of noise in the fresh ciphertext x_0 is B, let the degree of decryption polynomial f is d. The scheme can decrypt the ciphertext correctly when $|f| < p/2$. Due to the condition, the bound of noise has $t^d \cdot B^d < p/2$ after d times multiplication, in other words $d < (\log p)/(\log tB)$. The depth of decryption circuit depends on the operation levels on $c \cdot p^{-1}$ which is at least $2(\log p)^{2.71}$ levels. It is obviously bigger than the polynomial degree d. To get fully homomorphic encryption, the bootstrapping with squashing the decryption circuit is still essential [30].

The security of the DGHV scheme is based on two problems, one is the hardness of approximate-gcd problem in somewhat homomorphic encryption and another is SSSP in bootstrapping. To denoted against the known attack on the approximate-gcd problem like brute-forcing the remainders, continued fraction and Howgrave-Graham's approximate gcd algorithm, the security parameter of the scheme needs at least 2λ [10], [35].

3.5.3 Brakerski and Vaikuntanathan's Scheme Based on RLWE (BV11b)

Brakerski and Vaikuntanathan presented a scheme based on RLWE [6]. They used two techniques: re-linearization and dimension-modulus reduction to construct the scheme. Re-linearization can reduce the size of the ciphertext back down to $n+1$. Let

s be the original secret key and t is the new secret key. Each ciphertext is $(\alpha_{i,j}, b_{i,j})$ where

$$b_{i,j} =< \alpha_{i,j}, t > +2e_{i,j} + s[i] \cdots [j] \approx< \alpha_{i,j}, t > +s[i] \cdot [j].$$

Consider the multiplication of two polynomials,

$$\begin{aligned} f_{\alpha,b}(x) \cdot f_{\alpha',b}(x) &= (b - \sum \alpha[i]x[i]) \cdot (b' - \sum \alpha'[i]x[i]) \\ &= h_0 + \sum h_i \cdot x[i] + \sum h_{i,j} \cdot x[i]x[j] \\ &= h_0 + \sum h_i(b_i - < \alpha_i, t >) + \sum_{i,j} h_{i,j} \cdot (b_{i,j} - < \alpha_{i,j}, t >) \end{aligned}$$

the result is the linear polynomial with $n + 1$ coefficients and can be decrypt by the new secret key t. It is a good way to multiply two ciphertext without expanding the size and can be decrypted under the new secret key. The somewhat fully homomorphic encryption scheme is given as:

- KeyGen(λ):

  Choose random$L + 1$ vectors $s_0, s_1, \ldots, s_L \leftarrow \mathbb{Z}_q^n$

  Choose random matrix $A \leftarrow \mathbb{Z}_q^{m \times n}$

  Choose a random vector $e \leftarrow \chi^m$

  Compute $b = As_0 + 2e$

  Output $sk = s_L$ and $pk = (A, b)$.

- Encrypt(pk,m):

  Choose a random vector $r \leftarrow \{0, 1\}^m$

  Let$v = A^T r \in \mathbb{Z}_q^n$

  Let $w = b^T r + m \in \mathbb{Z}_q$

  Output a ciphertext $c \leftarrow ((v, w), \ell)$.

- Decrypt(sk,c):

Input a secret key sk and a ciphertext c.

Output $m \leftarrow (w - < v, s_L > \mod p) \mod 2$.

- Evaluate($pk, f, c_1, \ldots, c_\tau$):

  Addition gates: $c_{Add} = ((v_{Add}, w_{Add}, \ell) := ((\sum_i v_i, \sum_i w_i), \ell)$

  Multiplication gates: $c_{Mult} = ((v_{Mult}, w_{Mult}), \ell)$

The security of the scheme relies on the worst-case hardness of classical problem on lattices [45].

3.5.4 Brakerski, Gentry and Vaikuntanathan' Scheme Based on LWE (BGV12)

BGV is the most efficienct scheme so far. The scheme applies key switching and modulus switching, it is a leveled fully homomorphic encryption scheme without bootstrapping [5]. It reduces the production of two ciphertext down to the original dimension by key switching and reduces the noise by modulus switching on each level[5]. The scheme can be based on the LWE and also the RLWE. The scheme on the RLWE has a better efficiency than on the LWE. Let ring $R = \mathbb{Z}[x]/(x^d + 1)$, where d is the power of 2 and $N = \lceil (2n + 1) \log q \rceil$. The schemes is:

- KeyGen(λ):

  Choose random $s' \leftarrow \chi^n$

  Let $s = (1, s)$ with $s[0] = 1$ and $s' \in R_q^n$ Choose random matrix $A \leftarrow R_q^{N \times n}$

  Choose a random vector $e \leftarrow \chi^N$

  Compute $b = A's' + 2e$

  Set A is the (n+1) column matrix $(b| - A')$

  Output $sk = (1, s'[1], \ldots, s'[n]) \in R_q^{n+1}$ and $pk = A$.

- Encrypt(pk,m):

  Let $m \leftarrow (m, 0, \ldots, 0) \in R_q^{n+1}$

  Choose a random vector $r \leftarrow R_2^N$

  Output a ciphertext $c \leftarrow m + A^T r \in R_q^{n+1}$.

- Decrypt(sk,c):

  Input a secret key sk and a ciphertext c.

  Output $m \leftarrow (< v, s_L > \mod p) \mod 2$.

- Evaluate($pk, f, c_1, \ldots, c_\tau$):

  Addition gates: $c_4 = \mathbf{Refresh}(c_3, \tau''_{s_j} \rightarrow s_{j-1}, q_j, q_{j-1})$, where $c_3 \leftarrow c_1 + c_2 \mod q_j$

  Multiplication gates: $c_4 = \mathbf{Refresh}(c_3, \tau''_{s_j} \rightarrow s_{j-1}, q_j, q_{j-1})$, where c_3 is the linear equation of $L^{long}_{c_1,c_2}(x \bigotimes x)$

Assuming the error with bound B and the corresponding modulus is q_j. The noise will increase to $2B$ by addition and approximated to be B^2 by multiplication. After the key switching, the error becomes $E^2 + e_{switch}$. Processing the modulus switching, the error decreases to $(q_{j-1}/q_j) \cdot (E^2 + e_{switch}) + e_{scale}$. To decrypt the ciphertext correctly, the error should be smaller than B on each level [3]. The scheme can operate on a circuit of depth L. It can transfer to the fully homomorphic encryption by bootstrapping. The security of the scheme relies on the SVP problem on lattices [42].

3.5.5 Brakerski's Scheme Based on LWE (Bra12)

This scheme is also based on the LWE problem and can be extend to the RLWE. The advantages of Bra12 scheme are: using the same modulus which means the scheme does not need to do modulus switching. The security can be classically reduced to the worst-case hardness of the GapSVP problem [4].

The technique they used to construct the scheme is vector decomposition and key switching. Vector decomposition is a way to operate the inner product.

- **BitDecomp**$_q(x)$: For $x \in \mathbb{Z}^n$, let $w_i \in \{0,1\}^n$, x can represent as $x = \sum_{i=0}^{\lceil \log q \rceil - 1} 2^i \cdot w_i \mod q$.
- **PowerOfTwo**$_q(y)$: For $y \in \mathbb{Z}^n$, output $[(y, 2 \cdot y, \ldots, 2^{\lceil \log q \rceil - 1} \cdot y)]_q \in \mathbb{Z}_q^{n \cdot \lceil \log q \rceil - 1}$

The somewhat homomorphic encryption scheme is:

- KeyGen(λ):

  Choose random$L+1$ vectors $s_0, s_1, \ldots, s_L \leftarrow \mathbb{Z}_q^n$

  Choose random matrix $A \leftarrow \mathbb{Z}_q^{N \times n}$

  Choose a random vector $e \leftarrow \chi^m$

  Compute $b_0 = As_0 + 2e$

  Set P_0 is the $(n+1)$ column matrix $(b_0| - A')$

  $\tilde{s}_{i-1} = \mathbf{BitDecomp}(1, s_{i-1}) \bigotimes \mathbf{PowerOfTwo}(1, s_{i-1}) \in \{0,1\}^{((n+1)\lceil \log q \rceil)^2}$

  Compute $P_{i-1} : i \leftarrow \mathbf{SwitchKeyGen}(\tilde{s}_{i-1}, s_i)$

  Output $sk = s_L$ and $pk = P_0$, and $evk = \{P_{(i-1):i}\}_{i \in [L]}$.

- Encrypt(pk,m):

  Let $m \leftarrow (m, 0, \ldots, 0) \in R_q^{n+1}$

  Choose a random vector $r \leftarrow R_2^N$

  Output a ciphertext $c \leftarrow m + A^T r \in R_q^{n+1}$

- Decrypt(sk,c):

  Input a secret key sk and a ciphertext c.

  Output $m \leftarrow < v, s > \mod p) \mod 2$.

- Evaluate($pk, f, c_1, \ldots, c_\tau$):

  Addition gates: $c_{Add} = \mathbf{SwitchKey}(P_{(i-1):i}, \tilde{c}_{Add}) \in \mathbb{Z}_q^{n+1}$, where $\tilde{c}_{Add} =$

$\mathbf{PowerOfTwo}(c_1 + c_2) \bigotimes \mathbf{PowerOfTwo}(1, 0, \ldots, 0)$

Multiplication gates: $c_{Mult} = \mathbf{SwitchKey}(P_{(i-1):i}, \tilde{c}_{Mult}) \in \mathbb{Z} :_q^{n+1}$, where $\tilde{c}_{Mult} = \lfloor \frac{2}{q} \cdot \mathbf{PowerOfTwo}(c_1) \bigotimes \mathbf{PowerOfTwo}(c_2)) \rceil$

The noise increasing in this scheme is different from previous schemes. Assuming the noise bound of the ciphertext is E and the fresh ciphertext has noise bound of $N \cdot B$. The noise increases to $2E + n^2 \log q^3$ after addition and $(n \cdot \log q) \cdot E + (n^2 \log^{q^3}) \cdot B$ after multiplication. Since multiplication is defined as $(2/q) \cdot (c_1 \bigotimes c_2)$, each item divided by q, $E2/q$ can be ignored when q is large enough for classical reduction from GapSVP [4]. To get the decryption correctly, the error needs to be less than $\lfloor q/2 \rfloor/2$, therefore $q/B > (n \cdot \log q)^L$. The depth L depends on the ratio of q/B.

3.5.6 Plantard, Susilo and Zhang' s Hidden Ideal Lattice

Plantard, Susilo and Zhang constructed a fully homomorphic encryption scheme by using hidden ideal lattice [51]. They used hidden ideal lattice to unify two schemes which are ideal lattice based schemes and integer based schemes. The security of the scheme does not rely on the sparse sub set sum problem (SSSP), but rely on the bounded distance decoding problem(BDDH)of ideal lattice and approximate greatest common divisor problem (AGCD) of an integer.

The hidden ideal lattice homomorphic encryption scheme gives an idea, that instead of giving the lattice as the public key, given some vectors close to the lattice. The lattice is only known by the secret key holder. Since the ciphertext are also vectors close to the lattice with a bounded distance, the property of the homomorphism of the ciphertext still holds [39]. The somewhat scheme is :

- $\mathsf{KeyGen}(\lambda)$:

  Choose a random irreducible polynomial of degree n, $f(x) = x^n + 1$.

  Choose a random vector v in $\{u \in \mathbb{Z}^n, 2^{\eta-1} < \|u\| < 2^{\eta}, \ \sum_{i=0}^{n-1} u_i \mod 2 = 1\}$.

Generate the random matrix $\mathcal{V} \leftarrow Rot(v, f)$:

$$Rot(v,f) = \begin{vmatrix} v_0 & v_1 & v_2 & \dots & v_{n-1} \\ -v_{n-1} & v_0 & v_1 & \dots & v_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -v_1 & -v_2 & -v_3 & \dots & v_0 \end{vmatrix}$$

Let $d \leftarrow |det(\mathcal{V})|$ is the determinant of $\mathcal{V}$.

Choose random $\tau - 1$ vectors g_i in $\{u \in \mathbb{Z}^n, 2^{\gamma-1} < \|u\| < 2^{\gamma}\}$, and another vector g_τ in $\{u \in \mathbb{Z}^n, \|u\| < 2^{\gamma}, \sum_{i=0}^{n-1} u_i \mod 2 = 1\}$.

Choose random $\tau - 1$ vectors r_i in $\{u \in \{-1, 0, 1\}^n, \|u\| \leq \rho\}$, and another vector r_τ in $\{u \in \{-1, 0, 1\}^n, \|u\| \leq \rho, \sum_{i=0}^{n-1} u_i \mod 2 = 1\}$.

Compute τ vectors $\pi_i \leftarrow g_i \times v + r_i$, for $1 \leq i \leq \tau$.

Find the integer polynomial $w(x)$, which satisfies $w(x) \times v(x) = d \mod f(x)$, denote $\mathcal{W} \leftarrow Rot(w, f)$. Output $sk = s\{d, w\}$ and $pk = \{\pi_i\}$.

- Encrypt(pk,m):

  Choose random $\tau - 1$ vector s_i in $\{\sum_{j=1}^{n} s_{i,j} \mod 2 = 0, 1 \leq \imath \leq \tau - 1\}$, a vector s_τ in $\{\sum_{j=1}^{n} s_{\tau,i} \mod 2 = m\}$, and a vector $s_{\tau+1}$ in $\{\sum_{j=1}^{n} s_{\tau+1,j} \mod 2 = 0\}$.

  Output a ciphertext $c \leftarrow \sum_{i=1}^{n} s_i \times \pi_i + s_{\tau+1}$.

- Decrypt(sk,c):

  $c' \leftarrow \lfloor c \times w/d \rceil$. Output $c \leftarrow c'(1) \mod 2$.

- Evaluate($pk, C, \dots, c_\tau$):

  Addition gates (c_1, c_2): Output $c \leftarrow c_1 + c_2$.

  Multiplication gates (c_1, c_2): Output $c \leftarrow c_1 \times c_2$.

The semantic security of the scheme has been proved:

Theorem 3.5.1. *[33] If an algorithm A breaks the semantic security with advantage ε, then there exists an algorithm B that solves the Dec α, $\beta - BDDHi_{n,\tau}$ with advantage of $\varepsilon/8$. The running time of B is polynomial in the running time of A.*

3.5.7 Nuida and Kurosawa's Batching Scheme

The majority of fully homomorphic encryption schemes and somewhat homomorphic schemes can only encrypt a single bit each time. The efficiency can be improved by using batch plaintext into a single bit [12]. The scheme can encrypt multiple bits into a single ciphertext by using the Chinese Remainder Theorem. But it is only applied in binary space. Kofi et. modified the scheme into the non-binary space [12].

In DGHV scheme, the public key size in *somewhat homomorphic encryption* s $\widetilde{\mathcal{O}}(\lambda^{10})$ and in *fully homomorphic encryption* is $\widetilde{\mathcal{O}}(\lambda^{13})$. Coron[15] describes public key compression for fully homomorphic encryption over integers. It reduces the public key size to $\widetilde{\mathcal{O}}(\lambda^5)$ of *somewhat homomorphic encryption* and $\widetilde{\mathcal{O}}(\lambda^8)$ of *fully homomorphic encryption*. Consider the batch fully homomorphic encryption scheme in non-binary space, the public key size is $\widetilde{\mathcal{O}}(\lambda^8)$ in both *somewhat homomorphic encryption* and *fully homomorphic encryption*. To achieve this goal, the CRT (Chinese Remainder Theory) is an important technique [37]. Let $p_1, \ldots, p_2$ be pairwise coprime integers and $CRT_{(p_1,\ldots,p_2)}(m_1, \ldots, m_k)$ is a number in $\mathbb{Z} \cap (-\frac{\pi}{2}, \frac{\pi}{2}]$, where $\pi = \prod_{i=1}^{k} p_i$. $CRT_{(p_1,\ldots,p_k)}(m_1, \ldots, m_k)$ is equivalent to $m_i \bmod p_i$ for all $i \in \{1, \ldots, k\}$. So we have

$$CRT_{(p_1,\ldots,p_2)}(m_1, \ldots, m_k) = \sum_{i=1}^{k} m_i M_i (M_i^{-1} \bmod p_i) \bmod \pi$$

where $M_i = \pi/p_i$. The distributions of ℓ_Q-bits with multi-private keys is

$$\mathcal{D}_\rho(p_1, \ldots, p_k; Q_1, \ldots, Q_k; q_0) := \{\text{choose } e_0 \leftarrow \mathbb{Z} \cap [0, q_0), e_i \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho)\}.$$

Output

$$x = CRT_{(q_0,p_1,\ldots,p_k)}(e_0, e_1 Q_1, \ldots, e_k Q_k), \text{ for } \forall i \in \{1, \ldots, k\}.$$

Consider the value of x when $k = 1$, since $\mathcal{D} := \{\text{choose } q \leftarrow \mathbb{Z} \cap [0, q_0), e \leftarrow$

$\mathbb{Z} \cap (-2^\rho, 2^\rho)$: output $x = p_1 q + e \bmod p_1 q_0\}$, there is $x \leftarrow \mathcal{D}_\rho(p_1; q_0)$.

$$\begin{aligned} x &= CRT_{(q_0;p_1)}(e_0, e_1) \\ &= e_0 p_1 (p_1^{-1} \bmod q_0) + e_1 q_0 (q_0^{-1} \bmod p_1) \bmod q_0 p_1 \\ &= (e_0\alpha + e_1\beta)p_1 + e_1 \bmod q_0 p_1 \end{aligned}$$

for some α and β. Since $e_0 \not\equiv \bmod q_0$ and $\gcd(\alpha, q_0) = 1$, $(e_0\alpha + e_1\beta) \bmod q_0$ is uniformly in $\mathbb{Z} \cap [0, q_0)$. Recall Nuida and Kurosawa's batch somewhat homomorphic encryption scheme. The plaintext space is $\mathcal{M} = (\mathbb{Z}_{Q_1})^{h_1} \times (\mathbb{Z}_{Q_2})^{h_2} \times \cdots \times (\mathbb{Z}_{Q_k})^{h_k}$, where $k \geqslant 1$, $h_j \geqslant 1$ and $Q_1, \ldots, Q_k$ are distinct primes. The scheme to pack ℓ plaintext bits $m_0, \ldots, m_{\ell?1}$ into a single ciphertext is the extension of the DGHV scheme which listed as following [12]:

- KeyGen (1^λ):

  Pick random prime numbers $p_{i,j}$ as secret key, $(i, j) \in \mathcal{I}$ and $\mathcal{I} := \{(i, j) | i, j \in \mathbb{Z}, 1 \leq i \leq k, 1 \leq j \leq h_i\}$. $p_{(i,j)}$ and $Q_{i'}$are different. Choose

  $$q_0 \leftarrow [1, 2^\gamma / \prod_{(i,j)\in\mathcal{I}} p_{i,j}) \cap 2^{\lambda^2},$$

  which is coprime to all $p_{i,j}$ and all $Q_{i'}$. Choose $e_{\xi;0}$ and $e_{\xi;i,j}$ for $\xi \in \{1, 2, \ldots, \tau\}$ and $(i, j) \in \mathcal{I}$:

  $$e_{\xi;0} \leftarrow [0, q_0) \cap \mathbb{Z},\ e_{\xi;i,j} \leftarrow (-2^\rho, 2^\rho) \cap \mathbb{Z}.$$

  Let x_ξ be the unique integer in $(-N/2, N/2]$ satisfying

  $$x_\xi \equiv e_{\xi;0} \pmod{q_0},\ x_\xi \equiv e_{\xi;i,j} Q_i \pmod{p_{i,j}}\ for\ (i, j) \in \mathcal{I}$$

Similarly, for (i, j), $(i', j') \in \mathcal{I}$, choose $e'_{i,j;0}$ and $e_{i,j;i',j'}$:

$$e'_{i,j;0} \leftarrow [0, q_0) \cap \mathbb{Z},\ e_{i,j;i'j'} \leftarrow (-2^\rho, 2^\rho) \cap \mathbb{Z},$$

and let $x'_{i,j}$be the unique integer in $(-N/2, N/2]$ satisfying:

$$x'_{i,j} \equiv e'_{i,j;0} \pmod{q_0}$$

$$x'_{i,j} \equiv e'_{i,j;i',j'} Q_{i'} + \delta_{(i,j),(i',j')} \pmod{p_{i',j'}},$$

where δ is the Kronecker delta. The public key pk is N, x_ξ and $x'_{i,j}$, and the secret key sk consists of all $p_{i,j}$.

- Encrypt $(pk, m \in \mathcal{M})$:

  Generate a random subset $T \subseteq \{1, 2, \ldots, \tau\}$. Output the ciphertext as

$$c := \sum_{(i,j)\mathcal{I}} m_{i,j} x'_{i,j} + \sum_{\xi \in T} \text{Mod } N,\ c \in (-N/2, N/2] \cap \mathbb{Z}.$$

- Evaluate $(pk,\ f,\ c_1, \ldots, c_n)$:

  Given a polynomial f with integer coefficients and ciphertext $c_1, \ldots, c_n$, output c^* is

$$c^* := f(c_1, \ldots, c_n) \text{ Mod } N$$

- Decrypt (sk, c):

  Output $m := ((c \text{ Mod } p_{i,j}) \bmod Q_i)_{(i,j)} \in \mathcal{I}$.

The scheme is secured under the Error- Free Approximate-GCD assumption.

Definition 3.5.1. (Error-Free Approximate GCD problem) [51] For a random η-bit prime p, choose a random integer q_0 in $[0, 2^\gamma/\rho)$. Let $y_0 = q_0 \ldots p$ and sample many elements from $\mathcal{D}_\rho(p, q_0)$, then output p.

Similarly, for batching scheme, the specific integers are q_0 and $p_{i,j}(i,j) \in \mathcal{I}$. We input a vector $m \in \mathbb{Z}^\ell$ into the oracle $\mathcal{O}_{q_0,(pi,j)}(i,j) \in \mathcal{I}$ and the output will be X,

$$X = CRT_{q_0,(pi,j)}(q_0, m_{1,1} + Q_1 \cdot r_{1,1}, \ldots, m_{k,h_i} + Q_k \cdot r_{k,h_i})$$

where $q \leftarrow [0, q_0)$ and $r_{i,j} \leftarrow (-2^\rho, 2^\rho)$ Since it is hard to distinguish between an encryption of zero and an encryption of a random message by using the public-key encryption instead of the oracle. The scheme can be proved to be semantically secure [33].

3.5.8 Analysis of Existing Schemes

We will give the analysis of existing schemes based on two aspects: performance and security in this section.

	Public Key	Private Key	Ciphertext	Generate Pub-Key
BGV-L	$2n(n+1)\log q^2$	$(L+1)(n+1)\log B$	$(n+1)\lceil \log q \rceil$	$L(n+1)^3\lceil \log q \rceil^2$
Bra-L	$2n(n+1)\log q^2$	$(L+1)(n+1)\lceil \log q \rceil$	$(n+1)\lceil \log q \rceil$	$L(n+1)^3\lceil \log q \rceil^4$
GSW-L	$2n(n+1)\log q^2$	$(n+1)\lceil \log q \rceil^2$	$(n+1)^2\lceil \log q \rceil^3$	0
BGV-RL	$2n\log q^2$	$(L+1)(n+1)\lceil \log B \rceil$	$2n\lceil \log q \rceil$	$6Ln\lceil \log q \rceil^2$
Bra-RL	$2n\log q^2$	$(L+1)(n+1)\lceil \log B \rceil$	$2n\lceil \log q \rceil$	$6Ln\lceil \log q \rceil^2$
GSW-RL	$2n\log q^2$	$(n+1)\lceil \log q \rceil^2$	$4n\lceil \log q \rceil^3$	0

The table shows the storage size of different schemes. From the table, it shows the Ring-LWE schemes has a better performance which is $n \log q$ times less than the LWE schemes, and the generate public key is n^2 times less than the LWE schemes. The size of the private key and ciphertext are almost the same for both LWE and R-LWE schemes. GSW13 has different structure that there is no process of generating public

key. The size of ciphertext is $n \log q^2$ times greater than others based on LWE schemes and $\log q^2$ times greater than others based on R-LWE schemes. The size of public key and private key has slightly difference. BGV and Bra12 have similar storage size.

The table shows different schemes has different noise expanding performance which gives the depth of the operation in evaluation stage.

Schemes	Noise Expanding
BGV-LWE	$(q_{i-1}/q_i)\cdot(E^2+2(n+1)^2 B \log q_i)+(n+1)B \leq E \leq q_{i-1}/2$, E is the noise ceiling of the decryption circuit
Bra12-LWE	$\lvert t_1^L \cdot E + L \cdot t_1^{L-1} \cdot t_2 \rvert < \lfloor \frac{q}{2} \rfloor /2$, where $t_1 = 4(n+1)\lceil \log q \rceil$, $t_2 = 2(n+1)^2 \lceil \log q \rceil^3 B$ and $E = 2nB \log q$
GSW13-LWE	$(N+1)^L E = ((n+1)\log q + 1)^L \cdot (2nB \log q) < q/8$, where $E = 2nB \log q$
BGV-RLWE	$(q_{i-1}/q_i) \cdot (E^2 + 6n^{\frac{3}{2}} B \log q_i) + n^{\frac{3}{2}}(n+1)B \leq E$, E is the noise ceiling of the decryption circuit
Bra12-RLWE	$\lvert t_1^L \cdot E + L \cdot t_1^{L-1} \cdot t_2 \rvert < \lfloor \frac{q}{2} \rfloor /2$, where $t_1 = 2n^2 B + 8n$, $t_2 = 2n^B(4+B) + 2nB \log q$ and $E = 2n^2 B + B$
GSW13-RLWE	$(N+1)^L E = (2n \log q + 1)^L \cdot (2nB^2 + B) < q/8$, where $E = 2nB^2 + B$

From the table, we can conclude the noise expanding while increasing the depth of the circuit. Bra12 and GSW13 are similar, but the Bra12's expanding is slower than GSW13's. The BGV is different, the noise dose not expand since the modular exchange for every single evaluation, the noise will keep as the initial size for all operations. Generally, the Ring-LWE schemes has better noise control than LWE schemes except Bra12 scheme.

The security is another important part to compare schemes. Here we give an spe-

cial case when security parameter $\lambda = 80$, the table will list size of public key, private key and ciphertext while the adversary's advantage is $adv = 2^{-1}$,

Scheme	Depth	Pub Key	Gen Key	Private Key	Ciphertext	Total
GSW13 -LWE	$L = 0$	21564.98	0	26.42	248604.97	2.7×10^5
	$L = 5$	3×10^7	0	5301.23	1.9×10^9	2×10^9
	$L = 10$	4.7×10^8	0	40157.49	5.6×10^{16}	5.6×10^{10}
GSW13 -RLWE	$L = 0$	2.54	0	30.56	2926.68	2959.77
	$L = 5$	93.69	0	5857.56	3×10^6	3×10^6
	$L = 10$	365.80	0	44451.41	4.3×10^8	4.3×10^7
Bra12 -LWE	$L = 0$	17355.49	0	1.03	1.03	17357.55
	$L = 5$	5.7×10^7	9×10^{15}	355.24	59.21	9.7×10^{15}
	$L = 10$	8×10^8	2×10^{18}	2436.75	221.52	2×10^{18}
Bra12 -RLWE	$L = 0$	2.99	0	0.35	3.36	6.70
	$L = 5$	137.97	621218	19.53	219.13	621595.4
	$L = 10$	529.79	9216657	70.83	839.40	9×10^6

As shown in the table, consider the size of public key, private key and ciphertext, the R-LWE schemes have smaller size than the LWE schemes, which means the R-LWE schemes have better performance in the implementing. The Bra12 R-LWE has the smallest size in total.

Since the noise expansion control is the key point of the implementing of Fully Homomorphic Encryption scheme, existing schemes manage noise expansion relies on three ways: the noise in the ciphertext error , the noise in the private key and the noise in the ciphertext. The scheme BGV, Bra12, GSW13 on both LWE and R-LWE shrink the size of public key, private key and ciphertext to improve the efficiency of the Fully Homomorphic Encryption scheme. As the smallest total size is 607Mb to encrypt one

bit, it is still too large to make the implementing inefficient. We try to provide a new way to shrink the size of public key to improve the efficiency of the scheme to achieve the goal.

Chapter 4

The Construction of Fully Homomorphic Encryption Scheme

4.1 Our Scheme with Compression Public Key

Our technique consists in working on vectors with integer coefficient $Vec(\pi_{i,j})$ of the form $Vec(\pi_{i,j}) = Vec(\pi_i) \times Vec(\pi_j)$. The number of the public key stored is 2β not τ as initial. The τ public keys in the encryption can be generated by $\tau = \beta^2$ public keys. Then our encryption is no longer choosing a linear form as the public key. We will use a quadratic form.

4.1.1 SHE Scheme

The somewhat homomorphic scheme with public key compression on hidden ideal lattice is constructed as follows. Generate random polynomial vectors as the ring element, divide into two groups. Choose a vector from each group, then multiply the two vector modular the irreducible polynomial. Therefore, the original public key will be replaced by the new quadratic key.

- KeyGen(λ):

  Choose a random irreducible polynomial of degree n, $f(x) = x^n + 1$.

  Choose a random vector v in $\{u \in \mathbb{Z}^n, 2^{\eta-1} < \|u\| < 2^{\eta},\ \sum_{i=0}^{n-1} u_i \mod 2 = 1\}$.

  Generate the random matrix $\mathcal{V} \leftarrow Rot(v, f)$:

$$Rot(v,f) = \begin{vmatrix} v_0 & v_1 & v_2 & \dots & v_{n-1} \\ -v_{n-1} & v_0 & v_1 & \dots & v_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -v_1 & -v_2 & -v_3 & \dots & v_0 \end{vmatrix}$$

  Let $d \leftarrow |det(\mathcal{V})|$ is the determinant of $\mathcal{V}$. It is almost the same as the initial scheme so far, the difference is starting from the public key vector generation

  Choose two groups of random vectors, each group has β vectors g_i or g_j for $1 \le i, j \le \beta$, the Euclidean norm of each vector is in $\{u \in \mathbb{Z}^n,\ 2^{\gamma-1} < \|u\| < 2^{\gamma}\}$. There is at least one vector of each group with the Euclidean norm in $\{u \in \mathbb{Z}^n,\ \|u\| < 2^{\gamma},\ \sum_{i=0}^{n-1} u_i \mod 2 = 1\}$. The total number of vectors is 2β which equals $2\sqrt{\tau}$.

  Choose two groups of random vectors, each group has β vectors r_i or r_j for $1 \le i, j \le \beta$, the Euclidean norm of each vector is in $\{u \in R^n,\ \|u\| \le \rho\}$. There is at least one vector of each group with the Euclidean norm in $\{u \in R^n, \|u\| \le \rho,\ \sum_{i=0}^{n-1} u_i \mod 2 = 1\}$. The total number of vectors is 2β which equals $2\sqrt{\tau}$.

  Compute β vectors $\pi_i \leftarrow g_i \times v + r_i$ and β vectors $\pi_j \leftarrow g_j \times v + r_j$, for $1 \le i, j \le \beta$.

  Find the integer polynomial $w(x)$, which satisfies $w(x) \times v(x) = d \mod f(x)$, denote $\mathcal{W} \leftarrow Rot(w, f)$.

  Output $sk = \{d, w\}$ and $pk = \{\pi_i \text{ and } \pi_j\}$.

- Encryption:

  Choose random $\tau - 1$ vector $s_{i,j}$ in $\{\sum_{t=1}^{n} s_{i,j,t} \mod 2 = 0,\ 1 \le i, j \le \beta - 1\}$, a vector s_τ in $\{\sum_{t=1}^{n} s_{\tau,t} \mod 2 = m\}$ and a vector $s_{\tau+1}$ in $\{\sum_{t=1}^{n} s_{\tau+1,t} \mod 2 =$

0}. The Euclidean norm of all s are $\|s\| \leq \zeta$.

Output a ciphertext $c \leftarrow \sum_{i,j=1}^{\beta} s_{i,j} \times \pi_i \times \pi_j + s_{\tau+1}$.

- Evaluation and Decryption:

  The decryption and evaluation are the same as the initial scheme.

4.1.2 Correctness of Somewhat Homomorphic Encryption

First, we recall the definition of r_{Enc} and r_{Dec} in Gentry's idea [26].

Definition 4.1.1. (r_{Enc} and r_{Dec}) [51] r represents the distance between a ciphertext ψ and the hidden ideal lattice $\mathcal{L}$. r_{Enc} is the maximum possible distance for the encryption algorithm, and r_{Dec} is the minimum possible distance for the decryption algorithm.

We define the r_{pk} as the maximum distance between a public key $\pi_{i,b}$ and the hidden ideal lattice. According to the KeyGen algorithm, we have $r_{pk} \geq \theta \cdot \rho^2$. The noise of a ciphertext is the production of s and $r_{i,b}$ with the quadratic form in the encryption. The distance between a ciphertext and the hidden ideal lattice is $r_{Enc} \leq \theta \cdot (\theta \cdot \rho^2) \cdot \zeta = \theta^2 \cdot \rho^2 \cdot \zeta$. Since $\theta = \sqrt{n}$ is a constant of polynomial, we have $r_{Enc} \leq n \cdot \rho^2 \cdot \zeta$. From the result in [29], it shows $r_{Dec} \sim 2^{\eta}$. To decrypt the ciphertext correctly, we assume $r_{Enc} < r_{Dec}$ which means $n \cdot \rho^2 \cdot \zeta \leq 2^{\eta}$.

We will first prove the correctness of the decryption algorithm. For any ciphertext ψ, we consider the ciphertext has two parts, $\psi = a + b$, where $a \in \mathbb{Z}^n$ and $b \in \mathcal{L}$. Since $a = \sum_{i,j=1}^{\beta} r_i \times r_j \times s_{i,j} + s_{\tau+1}$, we have$\|a\| \leq n\rho^2\zeta$. Because $b \in \mathcal{L}$, we realize the only factor impacts on the decryption is a, hence $a = \psi \mod V = \psi - \lfloor \psi \cdot V^{-1} \rceil \cdot V$. $V^{-1} = W/d$ and the norm of the lattice $\mathcal{L}$ d is odd. Since d and 2 is co-prime, so we have $a \mod 2 = \lceil \psi \cdot W/d \rfloor \mod 2 = \lceil \psi \cdot w/d \rfloor \mod 2 = \psi' \mod 2$. Therefore $\psi' \mod 2 = a \mod 2 = \sum_{i,j=1}^{\beta} r_i \times r_j \times s_{i,j} + s_{\tau+1} \mod 2$. Next consider $\psi'(1)$

$\bmod 2 = \sum_{i,j=1}^{\beta} r_i(1) \times r_j(1) \times s_{i,j}(1) + s_{\tau+1}(1) \mod 2$, since random vectors $s_{i,j}$ are in $\{\sum_{t=1}^{n} s_{i,j,t} \mod 2 = 0, 1 \leq i \leq \tau - 1\}$ and $\{\sum_{t=1}^{n} s_{\tau+1,t} \mod 2 = 0\}$, we have $\psi'(1) \mod 2 = r_{\beta,\beta}(1)s_{\beta,\beta}(1) \mod 2 = m \mod 2$. The correctness of decryption has been proved .

Next, we prove the correctness of evaluation algorithm. Suppose $\psi_1 = a_1 + b_1$, and $\psi_2 = a_2 + b_2$ where $a_1,\ a_2 \in \mathbb{Z}^n$, $b_1,\ b_2 \in \mathcal{L}$, $\|a_1\|,\ \|a_2\| \leq n\rho^2\zeta$, and $a_t(x) = \sum_{i=1}^{\tau^2} r_{t,i}(x)s_{t,i}(x) + s_{j,\beta^2+1}(x) \mod f(x)$. Consider the Add algorithm first, $\psi(x) \leftarrow \psi_1(x) + \psi_2(x) \mod f(x) = (a_1(x) + a_2(x)) + (b_1(x) + b_2(x)) \mod f(x)$. Since $(b_1(x) + b_2(x)) \in \mathcal{L}$, we have $\|Vec(a_1(x) + a_2(x))\| \leq r_{Dec}$, decryption will be $a_1(1) + a_2(1) = m_1 + m_2$. The add algorithm is correct.

Similarly for Multiplication algorithm, $\psi(x) \leftarrow \psi_1(x) \times \psi_2(x) \mod f(x) = (a_1(x)a_2(x)) + (a_1(x)b_2(x)) + (a_2(x)b_1(x)) + (b_1(x)b_2(x)) \mod f(x)$. Since $(b_1(x)b_2(x)), (a_1(x)b_2(x))$ and $(a_2(x)b_1(x)) \in \mathcal{L}$, we have $\|Vec(a_1(x) \cdot a_2(x))\| \leq r_{Dec}$, decryption will be $a_1(1) \times a_2(1) = m_1 \times m_2$. Multiplication algorithm is correct.

4.2 The Security

In this section, we will prove our new scheme is semantically secure under the adaptation of the approximate greatest common factor (AGCD) assumption. The adversary can break the semantic security by instead of finding the vector $\mathcal{V}$. The hash function family $h(b) = \sum_{i=1}^{\tau} b_i \cdot \pi_i$ in the linear form is pairwise independent. By applying the leftover hash lemma, we want to prove that hash function family $h^{'}(b) = \sum_{i,j=1}^{\beta} b_{i,j} \cdot \pi_i \cdot \pi_j$ in the quadratic form is almost pairwise independent, which is $\varepsilon-$pairwise independent.

4.2.1 Leftover Hash Lemma

For a pairwise independent of hash function family $\mathcal{H}$, the hash function $h : X \to Y$ holds $\Pr_h[h(x) = h(x^{'})] = 1/|Y|$ for all $x \neq x^{'}$. For our variant, the hash function family is $h^{'} : \mathbb{Z}^{(n\times\beta)\times(n\times\beta)} \to \mathbb{Z}^n$, where $h^{'}(b) = \sum_{i,j=1}^{\beta} b_{i,j} \cdot \pi_i \cdot \pi_j$. It is not an exactly pairwise independent, but it could be almost pairwise independent with parameter. The following definition gives the $\varepsilon-$pairwise independent:

Definition 4.2.1. [16](ε-pairwise independent) A family $\mathcal{H}$ of hash function $h : X \to Y$ is $\varepsilon-$pairwise independent if

$$\sum_{x\neq x'}(Pr_{h\leftarrow\mathcal{H}}[h(x)) = h'(x)] - \frac{1}{Y}) \leq |X|^2 \cdot \frac{\varepsilon}{|Y|}$$

The leftover hash lemma give by the prior definition.

Lemma 4.2.1. *(Leftover Hash Lemma) [16] Let $\mathcal{H}$ be a family of ε-pairwise independent hash functions. choose random $h \leftarrow \mathcal{H}$ and $x \leftarrow X$ uniformly and independently. Then $(h, h(X))$ is $(\frac{1}{2}\sqrt{|Y|/|X| + \varepsilon})-$uniformly over $\mathcal{H} \times Y$*

Lemma 4.2.2. *For an odd determinant d, the hash function family $\mathcal{H}$ is ε- pairwise independence, with*

$$\varepsilon = \frac{1}{d} + \frac{n^2\tau}{2^{n^2\tau-2n\beta}}$$

4.2.2 Semantic Security

The semantic security of the scheme has been proved:

Lemma 4.2.3. *If an algorithm A breaks the semantic security with advantage ε, then there exist an algorithm B that solves the Dec α, $\beta - BDDHi_{n,\tau}$ with advantage of $3\varepsilon/32$. The running time of B is polynomial in the running time of A.*

4.3 FHE Scheme

In this section, we follow the Gentry's idea, use the bootstrapping technique to achieve fully homomorphic encryption scheme. Our scheme is similar as the initial scheme, the slightly difference is we using pseudo-random vector generator to construct the public key set. First, we will squash the decryption algorithm into a lower degree of the decryption polynomial. Then, we describe the bootstrappable scheme, which the post-processed ciphertext can be decrypted by modified decryption polynomial more efficiently.

4.3.1 The squashed Scheme

First, introduce four more parameters κ, σ, Θ and ϕ with functions of λ. More precisely, $\kappa = \eta + \gamma + 1 + \phi$, $\sigma = \lambda$, $\Theta = \tilde{\mathcal{O}}(\lambda^3)$ and $\phi = \lceil \log_2(\sigma + 1) \rceil$. We will add a set of public key $y = \{y_1, \ldots, y_\Theta\}$ of rational numbers in $[0, 2)$ of κ bits. There is a sparse subset $S \subset \{1, \ldots, \Theta\}$ of size σ with $\sum_{i \in S} y_i \simeq w_i/d$. The ciphertext is expanded by computing with y_i. The secret key sk is replaced by the binary vector of the subset S.

Instead of storing the whole set of y_i in the public key, we will use the pseudo-random vector generator $f(se)$ with seed se to generate y_i for $2 \leq i \leq \Theta$. Then the public key consists of se and y_1. The scheme will be as follows:

- $\mathsf{KeyGen}(\lambda)$:

  Generated $sk^* = w_1, d$ and pk^* as for the SHE scheme. Set $x_i =< x_1, \ldots, x_n >$ with $x_i \leftarrow \lceil 2^\kappa \times w_i/d \rfloor$

  Choose n vectors $s_i =< s_{i,1}, \ldots, s_{i,\Theta} >$ with Θ-dimensional, each s_i has hamming weight σ. Specifically, let $s_{i,1} = 1$ and $S = \{i, j : s_{i,j} = 1\}$

  Set $u_{i,1}$ such that $\sum_{i,j \in S} u_{i,j} = x_i$. Use $f(se)$ to generate vectors of Θ-dimensional $u_i =< u_{i,1}, \ldots, u_{i,\Theta} >$ with $u_{i,j} \in [0, 2^{\kappa+1})$, for $2 \leq i \leq \Theta$.

Set $y_{i,j} = u_{i,j}/2^{\gamma}$ and $y_i = \{y_{i,1}, \ldots, y_{i,\Theta}\}$, with γ bits after binary point. $[\sum_{j=1}^{\Theta} y_{i,j}]_2 = (w_i/d) - \Delta_d$ for some $|\Delta_d| < 2^{-\kappa}$.

Output $sk = \{s_i\}$ and $pk = \{\pi_{i,j}, se, y_{i,1}\}$, for $i \in n$.

- Encryption and Evaluation: Given a ciphertext $\psi =< \psi_1, \ldots, \psi_n >$, for each coefficient with respect to ψ_i with $i \in n$, generate $z_j = \psi_j \cdot y_j$, for $j \in \{1, \ldots, \Theta\}$, and keep $\phi = \lceil \log_2(\sigma + 1) \rceil$ bits after binary point for each z_j.

  Output ψ and z_j

- Decryption:

  $\psi_i^* = [z_i \cdot s_i]$ and $\psi^* \leftarrow< \psi_1^*, \ldots, \psi_n^* >$, for $i \in n$.

  $\psi' = \sum \psi_i^*$

  $\psi \leftarrow \psi'(1) \mod 2$

4.3.1.1 Correctness

We first prove the correctness of the squashed algorithm by rounding off the noise. Our algorithm has assumption likes:

$$\begin{aligned}
\psi_i^* &= z_i \cdot s_i \\
&= \psi_i \cdot y_i \cdot s_i \\
&= \psi_i \cdot u_i \cdot s_i / 2^{\kappa} \\
&= \psi_i \cdot x_i / 2^{\kappa} \\
&= \psi_i \cdot \frac{2^{\kappa} \times w_i/d}{2^{\kappa}} \\
&= \psi_i \times w_i/d
\end{aligned}$$

Hence, the noise can be eliminate since w_i/d is a small number, the algorithm is correct.

Next, we recall the definition of the permitted polynomial. Considering turn of the noise, we will prove the scheme is correct for the set $C(\mathcal{P}_{\mathcal{E}})$ of circuit that computes permitted polynomial.

Lemma 4.3.1. *The squashed scheme is correct for the set $C(\mathcal{P}_{\mathcal{E}})$ of the circuit that computed permitted polynomial. For every ciphertext (ψ, z_i) that is generated by evaluating a permitted polynomial, it holds that $[s_i \cdot z_i]$ is within $1/2$.*

Proof. Fix a permitted polynomial $P(x_1, \ldots, x_t) \in \mathcal{P}_{\mathcal{E}}$, an arithmetic circuit $\mathcal{C}$ can compute P, and t fresh ciphertext $c_1, \ldots, c_t$ that encrypt the input into $\mathcal{C}$. Denote $\psi = Evaluate(pk, \mathcal{C}, c_1, \ldots, c_t)$. Meanwhile, fix the public key and the secret key with respect to security parameter λ. For each $i \in n$, we have $y_i =< y_{i,1}, \ldots, y_{i,\Theta} >$ as the integer vectors in the public key and $s_i =< s_{i,1}, \ldots, s_{i,\Theta} >$ as binary vectors in the secret key. From the above assumption, we need to prove $[\psi_i \cdot w_i/d] = [z_i \cdot s_i] \pmod 2$.

Recall $z_i \leftarrow [\psi_i \cdot y_i]$ with $\phi = \lceil \log_2(\sigma + 1) \rceil$ bits after binary point for each z_i. We have $[\psi_i \cdot y_i] = z_i - \Delta_i$, since Δ_i has ϕ bits after binary point, so the maximum bits of Δ_i is $\phi + 1 = \lceil \log_2(\sigma + 1) \rceil + 1$, $|\Delta_i| \leq 2^{-(\phi+1)} \leq 2^{-(\lceil \log_2(\sigma+1) \rceil + 1)} \leq 1/2(\sigma + 1)$.

$$
\begin{aligned}
[\psi_i \cdot w_i/d - z_i \cdot s_i] &= [\psi_i \cdot w_i/d - s_i \cdot (\psi_i \cdot y_i) - s_i \cdot \Delta_i] \\
&= [\psi_i \cdot w_i/d - \psi_i \cdot (s_i \cdot y_i) - s_i \cdot \Delta_i] \\
&= [\psi_i \cdot w_i/d - \psi_i \cdot (w_i/d - \Delta_d) - s_i \cdot \Delta_i] \\
&= [\psi_i \cdot w_i/d - \psi_i \cdot w_i/d + \psi_i \cdot \Delta_d - s_i \cdot \Delta_i] \\
&= [\psi_i \cdot \Delta_d - s_i \cdot \Delta_i]
\end{aligned}
$$

Recall $[s_i \times y_i] = (w_i/d) - \Delta_d$ with $\Delta_d \leq 2^{-\kappa}$. We have

$$\begin{aligned} |[\psi_i \cdot \Delta_d - s_i \cdot \Delta_i]| &\leq |[\psi_i \cdot \Delta_d| + |s_i \cdot \Delta_i]| \\ &\leq 2^{\gamma+\eta} \cdot 2^{-\kappa} + \sigma \cdot \frac{1}{2(\sigma+1)} \\ &= 2^{\gamma+\eta-\kappa} + \sigma \cdot \frac{1}{2(\sigma+1)} \\ &= 2^{-1-\phi} + \sigma \cdot \frac{1}{2(\sigma+1)} \\ &< \frac{1}{2(\sigma+1)} + \frac{\sigma}{2(\sigma+1)} \\ &= 1/2 \end{aligned}$$

Therefore, the claim follows. □

4.3.2 Bootstrapping

In this section, we will prove the squashed scheme is bootstrappable. From Gentry's idea [26], our scheme can achieve fully homomorphic for a circuit of any depth.

Theorem 4.3.2. *[26] Let $\mathcal{E}$ be the scheme above, and let $\mathcal{D}_\mathcal{E}$ be the set of augmented (squashed) decryption circuits. Then, $\mathcal{D}_\mathcal{E} \subset C(\mathcal{P}_\mathcal{E})$.*

Proof. To prove $\mathcal{E}$ is bootstrappable, we need to show the modified decryption $m \leftarrow \sum[z_i \cdot s_i](1) \mod 2$ is a permitted polynomial size circuit. Recall $s_i =< s_{i,1}, \ldots, s_{i,\Theta} >$ for each $i \in n$ are binary number vectors and each $s_{i,j}$ is a bit, similarly, $z_i =< z_{i,1}, \ldots, z_{i,\Theta} >$ for each $i \in n$ are rational number vectors and each $z_{i,j}$ is rational number in $[0, 2)$, keeping $\phi = \lceil \log(\sigma+1) \rceil$ bits of precision after the binary point. We also proved $\sum[s_i \cdot z_i]$ is within $1/2$, and the Hamming weight is σ of the bits s_i for each i. The computation algorithm of the decryption can be split into four steps:

- Step 1: For $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, \Theta\}$, set $a_{i,j} \leftarrow s_{i,j} \cdot z_{i,j}$. $a_{i,j} = z_{i,j}$ when $s_{i,j} = 1$ and $a_{i,j} = 0$ when $s_{i,j} = 0$, with $\phi = \lceil \log(\sigma+1) \rceil$ bits of precision after

the binary point in binary representation. Set the vectors $a_i =< a_{i,1}, \ldots, a_{i,\Theta} >$.

- Step 2: Set the vectors $x_j = \sum a_j$ for each $i \in n$.
- Step 3: For each $i \in n$, from the Θ rational numbers $\{x_j\}_{j=1}^{\Theta}$, generate other $\phi + 1$ rational numbers $\{y_t\}_{t=1}^{\Theta+1}$, each y_t has less than ϕ bits of precision, and satisfy $\sum_j x_j = \sum_t y_t \mod 2$
- Step 4: Output $m \leftarrow \sum_t y_t(1) \mod 2$

Step 1 and 2 require only constant depth, because when adding vectors, there is no expensive carry operations needed. For step 3, we will apply the grade-school addition to handle the carries, and the carries are constant-depth. The last step, we can just use a constant depth circuit, the circuit has polynomial fan-in add-gates and constant fan-in mult-gates. Therefore, the total degree of the squashed scheme depends on the decryption polynomial in the binary representation. □

From [29], the bound of the noise in the evaluated ciphertext is $r_{Eva} \leq (r_{Enc})^d \times \sqrt{m}$, where d is the degree of the polynomial and m is the number of monomials. For elementary symmetric polynomials with the degree of d, the number of monomials is $m = \prod_{i=0}^{\lfloor \log_2 d \rfloor} \binom{d}{2^i} \sim 2^{m'}$. To guarantee the ciphertext is inside the decryption radius of the secret key, we have

$$(r_{Enc})^d \sqrt{m} = (n\rho^2\zeta)^d \sqrt{m} \leq 2^{\eta}. \tag{4.1}$$

The degree of the permitted polynomial is $d \leq (\eta - m'/2)/(\log(n\rho^2\zeta))$. From the result in [26], we need to support the degree of the polynomial up to d, then we have $\eta \geq d \cdot \log(n\rho^2\zeta) + \log\sqrt{m}$ to evaluate the "squashed decryption circuit" for deep enough circuits.

4.3.3 Security of the Squashed Scheme

The security of the squashed scheme is based on the sub set sum problem. To recover the secret key, the attacker has to find all the coefficient of w_i. Since we make public the polynomial vectors instead of the ideal lattice, the attacker is not able to recover w_i correctly. A brute force attack on chosen cipertext can achieve the goal. To solve the n different t, the complexity is $\binom{\sigma}{t}^n$ [51].

4.4 Attacks

The SHE and FHE is secure against chosen plaintext attacks. Due to the adversary's attack like IND-CCA2 attack, like manipulating the challenged ciphertext and submit it to the decryption oracle, there is no SHE and FHE scheme can be IND-CCA2 secure. The IND-CCA1 has been proved to be not secure for FHE and SHE scheme [40]. Zhang provided a way to recover the secret key by using the decryption oracle over the DGHV scheme [66] [67]. The adversary can perform more algorithms to recover the pubic keys through decryption oracle queries [11]. Since the scheme based on the two cryptosystem: ideal lattice based system and integer based system. We consider the attacks on the approximate-gcd problem [35] and the BDD problem[39].

4.4.1 Brute Force Attack

The simplest attack is brute force attack on the noise in the public key. Since the ciphertext is protected by noise, this attack will guess the possible noise to recover the secret key. Since each public key $\pi_i = g_i \times v + r_i$, the v can be computed by $v = gcd(\pi_1 - r_1, \ldots, \pi_{2\beta} - r_{2\beta})$. The scheme needs the number of possible r_i more than 2^λ to against the attack. The Stehle-Zimmermann algorithm to compute the GCD's told us, the time complexity of the algorithm is $\tilde{\mathcal{O}}(\gamma + \eta)$ for the norm of the vectors

of $\gamma + \eta$ bits. The attack complexity is $\rho \cdot \tilde{\mathcal{O}}(\gamma + \eta)$. Therefore, the attack is thwarted when $\rho = \omega(\log \lambda)$.

4.4.2 Birthday Attack

To resist the birthday attack on our scheme, the running time of the attack needs to be greater than $2^{\log \rho/2}$ [29]. So the bit length of the noise has to be at least 2λ bits to against this attack, and the possible numbers of the noise relative to a single key is at least $2^{\lambda/2}$.

4.4.3 SDA-Simultaneous Diophantine Approximation

In this section, we start with the known attack based on AGCD problem. To solve the AGCD problem with many numbers, we can apply simultaneous Diophantine approximation (SDA) [38]. The element in our variant is polynomial vector rather than a single integer, we can not apply the SDA directly. Fortunately, the coefficient of each polynomial is integer, then we can modify the SDA to find the target vector. First, we generate a lattice $\mathcal{L}(B)$ by spanning the rows with $k + t \leq 2\beta$ public keys.

$$\mathrm{B} = \begin{vmatrix} \theta Mult\rho \cdot I_n & Rot(\pi_2) & Rot(\pi_3) & \dots & Rot(\pi_{k+t}) \\ 0 & -Rot(\pi_1) & 0 & \dots & 0 \\ 0 & 0 & -Rot(\pi_1) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & -Rot(\pi_1) \end{vmatrix}$$

For $\pi_1 = r_1 + g_1$ and $\pi_i = r_i + g_i$ where $1 \leq i \leq k + t$, we have $\pi_i g_1 - \pi_1 g_i = r_i g_1 - r_1 g_i$. Then we can apply LLL reduction algorithm to find the vector $u =< \theta \cdot \rho g_1,\ r_1 g_1 - r_1 g_1,\ r_2 g_1 - r_1 g_2,\ \dots,\ r_{k+t} g_1 - r_1 g_{k+t} >=< \theta Mult \cdot \rho \cdot g_1,\ r_1 g_1(\frac{r_2}{r_1} - \frac{g_2}{g_1}),\ r_1 g_1(\frac{r_3}{r_1} - \frac{g_3}{g_1}),\ \dots,\ r_1 g_1(\frac{r_{k+t}}{r_1} - \frac{g_{k+t}}{g_1}) >$. Once the attacker finds the vector u, they can recover all r_i.

The attack problem comes to the lattice reduction problem. Based on the lattice reduction algorithm [9], the target vector u cannot be found if $\frac{\lambda_2(\mathcal{L}(B))}{\|u\|} < c^{n(k+t)}$, where c is a constant reached to the smallest value of 1.009 [9].

From the definition of successive minimum $\lambda_2 \leq (n^{n/2}\ \det(\mathcal{L})/\lambda_1)^{\frac{1}{(n-1)}}$ [59], we have $\lambda_2 \leq \sqrt{n(k+t)}^{\frac{n(k+t)}{n(k+t)-1}}(\frac{\det(\mathcal{L})}{\|u\|})^{\frac{1}{n(k+t)-1}}$. If $\sqrt{n(k+t)}^{\frac{n(k+t)}{n(k+t)-1}}\det(B)^{\frac{1}{n(k+t)-1}} < c^{n(k+t)}\|u\|^{\frac{n(k+t)}{n(k+t)-1}}$, the target vector u cannot be computed by lattice reduction. Briefly, to guarantee $\|u\|$ is hard to be found, we need the matrix satisfy the condition of $\det(B) < c^{(n(k+t))(n(k+t)-1)}\|u\|^{n(k+t)}$. Since $\det(B) = (\theta\rho)^n\|\pi_1\|^{k+t-1}$, therefore we can get $\det(B) \leq (\theta\rho)^n(\theta \cdot \|g_1\| \cdot \|v\|)^{n(k+t-1)}$. As $\|u\| > \theta \cdot \rho \cdot \|g_1\|$, the successful attack achieves when $(\theta\rho)^n(\theta\|g_1\|\|v\|)^{n(k+t-1)} \geq c^{(n(k+t))(n(k+t)-1)}(\theta\cdot\rho\cdot\|g_1\|)^{n(k+t)} \geq c^{n^2(k+t)^2}(\theta\cdot\rho\cdot\|g_1\|)^{n(k+t)}$. We conclude an inequation as :

$$\eta(k+t-1) \geq n(k+t)^2 \log_2 c + \gamma + (k+t-1)\log_2 \rho, \tag{4.2}$$

and

$$\log_2 c \leq \frac{(\eta - \log_2 \rho)(k+t-1) - \gamma}{n(k+t)^2}. \tag{4.3}$$

To get the right hand side maximum, we need to find the maximum value of

$$\frac{\eta - \log_2 \rho}{n(k+t)} - \frac{\eta - \log_2 \rho - \gamma}{n(k+t)^2},$$

since $\log_2 \rho$ is quite small compared with η, which means $k+t \sim \mathcal{O}(\frac{\gamma}{\eta})$ gives the best attack.

In our parameter setting, $k+t \sim \mathcal{O}(\frac{\gamma}{\eta})$. To resist the modified SDA attack, our numbers of public key has to satisfy $\frac{\gamma}{\eta} > 2\beta$. The attacker can use all public keys to give themselves best advantage. For $2\beta \geq \frac{\gamma}{\eta}$, the time to get a 2^η approximation is $2^{\gamma/\eta}$. Therefore to thwart this attack, we need $\gamma/\eta = \omega(\log\lambda)$. The setting of γ is $\gamma = \omega(\eta\log\lambda)$.

4.4.4 Nguyen and Stern's Orthogonal Lattice

Using Nguyen and Stern's orthogonal lattice [46] is another way to operate the lattice attack. The attacker will be a failure if the dimension of the lattice is larger than the ratio of the bit length of the public key and the bit length of secret key $k+t > (\gamma+\eta)/\eta$, more precisely, the target vector will not be covered when $k+t > (\gamma+\eta)/(\eta - \log_2^{\rho})$. The time complexity is roughly $2^{2\gamma/\eta^2}$.

We generate the lattice spanned by the row of the following $(t+k) \times (t+k+1)$ matrix:

$$\mathrm{B} = \begin{vmatrix} Rot(\pi_1) & R_1 I_n & & & \\ Rot(\pi_2) & & R_2 I_n & & \\ \vdots & & & \ddots & \\ Rot(\pi_{k+t}) & & & & R_{k+t} I_n \end{vmatrix}$$

The row (i) represents the constraint $Rot(\pi_i) - r_i I_n = 0 \mod V$, where R_i is an upper bound on $|r_i|$. Let the vector $v =< v_0, v_1, \ldots, v_\beta >= \sum_{i=1}^{k+t} g_i B_{in}$, for $n \in 1, \ldots, n$ in the lattice above. We obtain

$$v_0 - \sum_{i=1}^{k+t} \frac{v_i}{R_i} \cdot r_i = \sum_{i=1}^{k+t} g_i (Rot(\pi_i) - r_i I_n) = 0 \mod V.$$

The vectors are orthogonal to $(1, -\frac{r_1}{R_1}, -\frac{r_2}{R_2}, \cdots - \frac{r_{k+t}}{R_{k+t}})$ by the lattice reduction algorithm, then the noise r_i can be recovered.

The determinant of the lattice is approximately to the product of the columns of B, which is $\sqrt{k+t}\|Rot(\pi_i)\| \prod_{i=1}^{k+t} R_i \approx \sqrt{k+t}\|Rot(\Pi_i)\| R_i^{k+t}$, where $\|Rot(\Pi_i)\|$ is the upper bound of $\|Rot(\pi_i)\|$. Therefore, we have $R \sqrt[k+t]{\Pi} < \sigma\rho$, where $k+t > (\gamma+\eta)/(\eta - \log_2 \rho)$. The parameter to use against the known attack will be similar as the SDA attack.

4.4.5 Coppersmith's Method

We consider the Coppersmith's technique [14] to attack on recovering the noise of public keys. Coppersmith's method does not only focus on the relations $Rot(\pi_i) - r_i I_n = 0 \mod V$, but also consider the relations like $(Rot(\pi_i) - r_i I_n)^2 = 0 \mod V$, or $(Rot(\pi_i) - r_i I_n) \times (Rot(\pi_i)' - r_i' I_n) = 0 \mod V$. The lattice will be generated as follows. We still set Π as the bound of the public key and R is the bound of noise, and let all $\pi_{i,j}$ be roughly the same size Π. The first row of the matrix has size $\tilde{\mathcal{O}}(\Pi^d)$, d is the relations of product, where $d \leq 2\beta$. The next 2β rows has size $\tilde{\mathcal{O}}(\Pi^{d-1}R)$ on the pivots position. In the general case, on the pivots position, there are $\binom{2\beta+d-1}{d}$ rows of the size $\tilde{\mathcal{O}}(\Pi^{d-i}R^i)$. Remaining rows are the size of $\tilde{\mathcal{O}}(R^d)$.

The determinant of the lattice $\det(B) \approx \Pi^2 \cdot (\Pi R)^{2\beta} \cdot (R^2)^{\binom{2\beta}{2}-1} = \Pi^{2+2\beta} R^{4\tau-2}$. The attacker will take the best advantage if $2\beta \leq (\gamma - \rho)/(\eta - \rho)$. To against the attack, we need to choose the number of public key $2\beta > (\gamma - \rho)/(\eta - \rho) \sim \mathcal{O}(\gamma/\eta)$. which is also close to the previous attack.

4.4.6 BDD-Bounded Distance Decoding

We will use the BDD problem to recover the random vectors $u \leftarrow < 1, s_1, s_1, \ldots, s_\beta >$. This is the known message attack to find the shortest vector by lattice reduction. By using the ciphertext, the matrix generate as follow:

$$\mathrm{B} = \begin{vmatrix} 1 & 0 & 0 & 0 & \ldots & 0 & \psi \\ 0 & I_n & 0 & 0 & \ldots & 0 & Rot(\pi_1 \times \pi_1) \\ 0 & 0 & I_n & 0 & \ldots & 0 & Rot(\pi_1 \times \pi_2) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & I_n & Rot(\pi_\beta \times \pi_\beta) \end{vmatrix}.$$

As previously, $\det(Rot(\pi_1)) \leq \|\pi_1\|^n \leq (\sigma\|g_1\|\|v\|)^n$, therefore, $\det(B) \leq \|\psi\| + \tau(\theta^2\|g_1\|\|v\|\|g_2\|\|v\|)^n$, and $\|u\| = 1 + (\tau + 1)\zeta$. From the Minkowski bounds, we relax

the condition to against the best lattice reduction, we have

$$2\log_2\theta + 2\gamma + 2\eta < \tau(n\tau+1)\log_2 c + \tau\log_2(1+\tau\zeta). \tag{4.4}$$

The number of public keys has to be $n \cdot \tau + 1$ to defend against the known attack.

4.5 Extension of The HIL Encryption to Higher Degree

In the section 3, we use a quadratic form to compute ciphertext instead of linear form. Due to reduce the number of public keys, the significant benefit of this scheme is reduce the size of the public key. Followed by this idea, we modified the scheme further by higher degree t of the encryption procedure, $c \leftarrow \sum_{i_1,\ldots,i_t=1}^{\beta} s_{i_1,\ldots,i_t} \times \pi_{i_1} \times \pi_{i_2} \cdots \times \pi_{i_t} + s_{\tau+1}$. The key point is to prove the hash function family $\mathcal{H}_t$ with $h : \mathbb{Z}^{(n\beta)^t} \to \mathbb{Z}^n$ is almost a pairwise independent hash function family, then it is suitable to apply a variant of the leftover hash lemma. The constraint will be $\beta^t \geq \gamma + \eta + \omega(\log\lambda)$.

To get the decryption correct, we also need $r_{Enc} = \theta^t \cdot \rho^t \cdot \zeta < 2^\eta$, and $r_{Enc} = \sqrt{m}(\theta^t \cdot \rho^t \cdot \zeta)^\ell < 2^\eta$ for bootstrapping. We consider the constant factor $m \cdot \theta$ as a small number, so r_{Enc} requires $\sqrt{m} \cdot \theta^t(\rho^t \cdot \zeta)^\ell < 2^\eta$. To defend against the known attack, we need γ to satisfy the equation

$$\gamma \geq (\eta - \log_2\rho)(t\beta - 1) - n(t\beta)^2\log_2 c, \tag{4.5}$$

and

$$t\log_2\gamma_m + t\gamma + t\eta < \tau(n\tau+1)\log_2 c + \tau\log_2(1+\tau\zeta). \tag{4.6}$$

We set a convenient parameter set as: $\rho = \lambda$, $\zeta = \lambda$, $\eta = \mathcal{O}(\lambda^2\log^k\lambda^2)$, $\gamma =$

$\mathcal{O}(\lambda^3 \log^k \lambda^3)$, $t = \log \lambda$ and $\tau = \beta^t = \mathcal{O}(\lambda^{3/t} \log^k \lambda^{3/t})$. Now, we store $\beta = \mathcal{O}(\lambda^{3/t} \log^k \lambda^{3/t})$ integers. Hence, the public key size becomes $\mathcal{O}(\lambda^4 \log^k \lambda^4)$ rather than $\mathcal{O}(\lambda^6 \log^k \lambda^6)$ in the original scheme.

Table 4.1: Comparisons between Quadratic and Higher Degree

	Quadratic	Higher Degree
Columns	2	$t = \log \lambda$
Numbers of PK in Each Column	$\beta = \mathcal{O}(\lambda^2 \log^k \lambda^2)$	$\beta = \mathcal{O}(\lambda^{\log \lambda} \log^k \lambda^{\log \lambda})$
Total Numbers of PK	$2 \cdot \beta = \mathcal{O}(\lambda^2 \log^k \lambda^2)$	$t \cdot \beta = \mathcal{O}(\log \lambda \cdot \lambda^{\log \lambda} \log^k \lambda^{\log \lambda})$
PK Size	$\mathcal{O}(\lambda^6 \log^k \lambda^6)$	$\mathcal{O}(\lambda^4 \log^k \lambda^4)$

4.6 Parameters and Constraints

We choose parameters under the following constraints:

- $\rho = \omega(\log \lambda)$ to avoid brute force attack on noise.
- $\eta \geq \log(n\rho^2\zeta) \cdot \Theta(\lambda / \log \lambda)$ to support the evaluation of squashed decryption circuits.
- $\gamma = \omega(\eta \cdot \lambda)$ to against lattice-based attacks.
- $\beta^2 \geq \log(\gamma + \eta) + \omega(\log \lambda)$ to use the leftover hash lemma in the reduction to approximate common vector.
- $\zeta = \omega(\log \lambda)$ for secondary noise parameter.
- $n = \omega(\lambda \log \lambda)$ to foils lattice-based reduction [29] and $\theta = \sqrt{n}$ [26].

We set a convenient parameter set as: $\rho = \lambda$, $\zeta = \lambda$, $\eta = \mathcal{O}(\lambda^2 \log^k \lambda^2)$, $\gamma = \mathcal{O}(\lambda^3 \log^k \lambda^3)$ and $\tau = \beta^2 = \mathcal{O}(\lambda^3 \log^k \lambda^3)$. The main difference is that instead of having $\tau = \mathcal{O}(\lambda \log^k \lambda)$ integers, we store $\beta = \mathcal{O}(\lambda^{1.5} \log^k \lambda^{1.5})$ integers. Hence, the public key size becomes $\mathcal{O}(\lambda^{4.5} \log^k \lambda^{4.5})$ rather than $\mathcal{O}(\lambda^6 \log^k \lambda^6)$ in the original scheme.

We will use $\lambda = 80$ as an example to compare the new scheme and the initial scheme. Under the assumption that the dimension of the lattice is the square root of the dimension of the normal lattice.

Table 4.2: The Relations between Degree of Decryption Polynomial and Number of Monomials

Degree of Decryption Polynomial	Number of Monomials
$3 = 2^2 - 1$	9
$7 = 2^3 - 1$	5145
$15 = 2^4 - 1$	$\sim 2^{34}$
$31 = 2^5 - 1$	$\sim 2^{75}$
$63 = 2^6 - 1$	$\sim 2^{176}$
...	...
$1023 = 2^{10} - 1$	$\sim 2^{3180}$

In this case, it requires the lattice with dimension 31 (From the Table 1) to be large enough to resist the lattice reduction. To resist birthday paradox attack, the maximum norm of each noise is $\sqrt{32}$. s has $\tau + 1$ blocks, to stop the brute force attack, we set maximum 5 blocks with nonzero entries besides the τ-th block. So we can find the number of public keys used in the encryption scheme. The total sample is at least $\binom{\tau+1}{5}(\binom{n}{2}2^2)^5 > 2^{80}$, which is $\tau = 111$. We keep the same security level of the squashed secret key by $\Theta = 6$ and $\sigma = 1$. We set maximum 11 coefficient to be 1 or -1 , for r_{Enc}, the maximum norm of the noise in each ciphertext is $32 \cdot \sqrt{11}^3 \sim 2^{10}$. Here, we use the suggestion in [29], the expansion factor for the production of two random vectors is much too small, we can consider $\|v_1 \times v_2\| \approx \|v_1\| \cdot \|v_2\|$ for our example in the bootstrapping. The worst case occurs when $r_{Enc} = 2^{10}$. To achieve the bootstrapping, η has to satisfy the equation [1]: $2^\eta \geq \sqrt{2^{75}}(2^{10})^{31}$, therefore, $\eta = 348$. According to the known attack, we choose $\gamma = 7090$ as the smallest value to satisfy equation [5], [6] which guarantees the scheme is secure.

In our SHE scheme, the ciphertext size is $(348 + 7090) \times 31 \sim 225Kb$, the public key space is $(348 + 7090) \times 31 \times 22 \sim 4.8Mb$. In the squashed scheme, the public key

size is $(348 + 7090) \times 31 \times 6 \sim 1.3Mb$, the secret key size is $225 \times 31 \times 6 \sim 40.9Mb$. The whole scheme with the public key size is $4.8 + 1.3 + 40.9 = 47Mbits$ which is much smaller than the original scheme with $173.5Mbits$.

Table 4.3: Comparisons with Original HIL Scheme

	GH Scheme	Initial Scheme	Qaudratic	Higher Degree
Key Columns	n/a	1	2	$t = \log \lambda$
Security Parameter	$\lambda = 7$	$\lambda = 80$	$\lambda = 80$	$\lambda = 80$
Lattice Dimension	$n = 2048$	$n = 31$	$n = 31$	$n = 31$
Public Key Size	552 Mb	173.5 Mb	47 Mb	40.4 Mb
Ciphertext Size	780 Kb	573 Kb	225 Kb	202.8 Kb

Compare with the existing schemes, our scheme compress the size of the public key on the hidden ideal lattice scheme. Due the technique, we can compress the public key size into the higher degree which can make the process more faster and have a smaller storage space. The running time of the system is mainly inuenced by the size of the ciphertext and the squashed decryption polynomial (both degree and number of monomials). We note that those parameters in our scheme are smaller than Gentry and Halevis system, the hidden ideal lattice scheme, therefore, it is straightforward to see that the running time of our scheme is shorter. To extend the technique, we can see the higher degree has the similar function to squash the key size and ciphertext size, the operation time will be shorten, but not significant other than two columns.

Chapter 5

Conclusion and Future Work

There are three categories of fully homomorphic encryption scheme: ideal lattice based scheme, integer based scheme and LWE based scheme. Each branch has designed and improved Fully homomorphic Encryption schemes. There are three main aspects need to be improve: the construction of schemes, the efficiency of schemes and the security of the schemes. The construction of schemes are based on some mathematics tools: integer ring, polynomial ring, lattice, and ideal lattice. There are many modification can apply on these schemes. Second, the security parameter needs to be large enough to against the attack, which means the operation is take long time or the storage space is large. So the efficiency needs to be improve to implement the Fully homomorphic encryption. Last is the security of the Fully homomorphic encryption scheme. The major difference between Fully homomorphic encryption scheme and general encryption scheme is, the Fully homomorphic encryption needs to consider the evaluation of the ciphertext, which may leak the private key information.

In this thesis, we focus on the research of Fully Homomorphic Encryption schemes with better parameters than previous schemes as well as the optimization from the theoretical point of view. On the other hand, we present the method to estimate the parameters of Fully Homomorphic Encryption schemes based on the public key

compression. By summarizing the result of the current works, we realized the efficiency of the fully homomorphic encryption scheme can be improved by reducing the key size. So we focus on how to apply the technique on reducing the key size especially the public key. We also identified that not all three categories can achieve this goal by the same technique. Therefore, research was to apply the public key compressing on different schemes and also applied on the batching schemes.

To answer the first question: *how to reduce the size of public key somewhat homomorphic encryption*? We use the same way to construct fully homomorphic encryption schemes. First, to construct somewhat homomorphic encryption and then use the bootstrapping to achieve fully homomorphic encryption schemes. That means, we have to make sure our somewhat homomorphic scheme is correct, then we can apply squashing and bootstrapping. We construct the squashed scheme which needs to do the "post process" on the ciphertext.

To answer the second question: *How to construct a more efficient scheme based on the existing scheme with new technique?* We focus on applying the public key compression on batching schemes to improve the efficiency of fully homomorphic encryption. We modified Plantard, Susilo and Zhang's scheme, since the scheme can encrypt a plaintext vector rather than single bit. We apply the public key compression on the scheme to check whether the scheme has better key size with less time consuming. The correctness and security was proved, and the known attack was analyzed as well.

In the future, we construct a fully homomorphic encryption scheme based on coding theory. More precisely, we want to construct a modified McEliece cryptosystem scheme to achieve fully homomorphic encryption.The most linear code like binary Goppa codes, addition homomorphism is obvious, but multiplication homomorphism failed. Some existing modified scheme can only performing limited times of multiplication. We want to modified the scheme to achieve performing unlimited times of multiplication

without rapid expansion on ciphertext.

References

[1] Miklos Ajtai. Generating hard instances of lattice problems. *Electronic Colloquium on Computational Complexity*, 1996.

[2] Daniel J Bernstein, Johannes Buchmann, and Erik Dahmen. Post-quantum cryptography. *European Journal of Operational Research*, pages 949–950, 2011.

[3] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit abe and compact garbled circuits. In PhongQ. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 533–556. Springer Berlin Heidelberg, 2014.

[4] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. In *Advances in Cryptology–CRYPTO 2012*, pages 868–886. Springer, 2012.

[5] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 309–325. ACM, 2012.

[6] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. In *Proceedings of the 2011 IEEE 52nd Annual Sympo-*

sium on Foundations of Computer Science, FOCS '11, pages 97–106, Washington, DC, USA, 2011. IEEE Computer Society.

[7] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 505–524. Springer Berlin Heidelberg, 2011.

[8] Stanislav Bulygin, Albrecht Petzoldt, and Johannes Buchmann. Towards provable security of the unbalanced oil and vinegar signature scheme under direct attacks. pages 17–32, 2010.

[9] Yuanmi Chen and Phong Q Nguyen. Bkz 2.0: Better lattice security estimates. In *Advances in Cryptology–ASIACRYPT 2011*, pages 1–20. Springer, 2011.

[10] Yuanmi Chen and Phong Q Nguyen. Faster algorithms for approximate common divisors: Breaking fully-homomorphic-encryption challenges over the integers. In *Advances in Cryptology–EUROCRYPT 2012*, pages 502–519. Springer, 2012.

[11] Massimo Chenal and Qiang Tang. On key recovery attacks against existing somewhat homomorphic encryption schemes. In *The third International Conference on Cryptology and Information Security in Latin America, Latincrypt 2014*, 2014.

[12] Jung Hee Cheon, Jean-Sébastien Coron, Jinsu Kim, Moon Sung Lee, Tancrède Lepoint, Mehdi Tibouchi, and Aaram Yun. Batch fully homomorphic encryption over the integers. In *EUROCRYPT*, volume 7881, pages 315–335. Springer, 2013.

[13] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, November 1998.

[14] Don Coppersmith. Small solutions to polynomial equations, and low exponent rsa vulnerabilities. *Journal of Cryptology*, 10(4):233–260, 1997.

[15] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Scale-invariant fully homomorphic encryption over the integers. In *Public-Key Cryptography–PKC 2014*, pages 311–328. Springer, 2014.

[16] Jean-Sébastien Coron, Avradip Mandal, David Naccache, and Mehdi Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In *Advances in Cryptology–CRYPTO 2011*, pages 487–504. Springer, 2011.

[17] Jean-Sebastien Coron, David Naccache, and Mehdi Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. Cryptology ePrint Archive, Report 2011/440, 2011. `http://eprint.iacr.org/`.

[18] Christina Delfs and Steven D Galbraith. Computing isogenies between supersingular elliptic curves over $\mathfrak{O}_p$fp. *Designs, Codes and Cryptography*, 78(2):425–440, 2016.

[19] Whitfield Diffie and Martin E Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644–654, 1976.

[20] Marten Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43. Springer Berlin Heidelberg, 2010.

[21] Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. *applied cryptography and network security*, pages 164–175, 2005.

[22] Leo Ducas, Alain Durmus, Tancrede Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. pages 40–56, 2013.

[23] L Feo, David Jao, and Jerome Plut. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology*, 8(3):209, 2014.

[24] Luis Carlos Coronado Garca. On the security and the efficiency of the merkle signature scheme.

[25] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009.

[26] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 169–178, New York, NY, USA, 2009. ACM.

[27] Craig Gentry. Computing arbitrary functions of encrypted data. *Communications of the ACM*, 53(3):97–105, 2010.

[28] Craig Gentry and Shai Halevi. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 107–109. IEEE, 2011.

[29] Craig Gentry and Shai Halevi. Implementing gentry?s fully-homomorphic encryption scheme. In *Advances in Cryptology–EUROCRYPT 2011*, pages 129–148. Springer, 2011.

[30] Craig Gentry, Shai Halevi, and Nigel P Smart. Homomorphic evaluation of the aes circuit. In *Advances in Cryptology–CRYPTO 2012*, pages 850–867. Springer, 2012.

[31] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-

based. Cryptology ePrint Archive, Report 2013/340, 2013. `http://eprint.iacr.org/`.

[32] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In Jr. Kaliski, BurtonS., editor, *Advances in Cryptology - CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 112–131. Springer Berlin Heidelberg, 1997.

[33] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270 – 299, 1984.

[34] Lov K Grover. A fast quantum mechanical algorithm for database search. pages 212–219, 1996.

[35] Nick Howgrave-graham. Approximate integer common divisors. In *CaLC 2001, LNCS*, pages 51–66. Springer-Verlag, 2001.

[36] Vivek Kapoor, Vivek Abraham, and R K Singh. Elliptic curve cryptography. *Ubiquity*, 2008:7, 2008.

[37] Jinsu Kim, Moon Sung Lee, Aaram Yun, and Jung Hee Cheon. Crt-based fully homomorphic encryption over the integers. *IACR Cryptology ePrint Archive*, 2013:57, 2013.

[38] Jeffrey C Lagarias. The computational complexity of simultaneous diophantine approximation problems. *SIAM Journal on Computing*, 14(1):196–209, 1985.

[39] Yi-Kai Liu, Vadim Lyubashevsky, and Daniele Micciancio. On bounded distance decoding for general lattices. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 450–461. Springer, 2006.

[40] Jake Loftus, Alexander May, Nigel P Smart, and Frederik Vercauteren. On cca-secure somewhat homomorphic encryption. In *Selected Areas in Cryptography*, pages 55–72. Springer, 2012.

[41] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. *Journal of the ACM (JACM)*, 60(6):43, 2013.

[42] Daniele Micciancio. The test vector in a lattice is hard to approximate to within some constant. *SIAM journal on Computing*, 30(6):2008–2035, 2001.

[43] Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007.

[44] Daniele Micciancio and Shafi Goldwasser. *Complexity of lattice problems: a cryptographic perspective*, volume 671. Springer, 2002.

[45] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, April 2007.

[46] Phong Q Nguyen and Jacques Stern. The two faces of lattices in cryptology. In *Cryptography and lattices*, pages 146–180. Springer, 2001.

[47] Phong Q Nguyen and Brigitte Vallée. *The LLL algorithm*. Springer, 2010.

[48] Raphael Overbeck and Nicolas Sendrier. *Code-based cryptography*, pages 95–145. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[49] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 333–342, New York, NY, USA, 2009. ACM.

[50] Chris Peikert. Lattice cryptography for the internet, 2014.

[51] Thomas Plantard, Willy Susilo, and Zhenfei Zhang. Fully homomorphic encryption using hidden ideal lattice. *Information Forensics and Security, IEEE Transactions on*, 8(12):2127–2137, 2013.

[52] Oded Regev. Quantum computation and lattice problems. *SIAM Journal on Computing*, 33(3):738–760, 2004.

[53] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, STOC '05, pages 84–93, New York, NY, USA, 2005. ACM.

[54] Oded Regev. Lattice-based cryptography. In *Advances in Cryptology-CRYPTO 2006*, pages 131–141. Springer, 2006.

[55] Oded Regev. The learning with errors problem. *Invited survey in CCC*, 2010.

[56] Ronald L Rivest, Len Adleman, and Michael L Dertouzos. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

[57] Ronald L Rivest, Adi Shamir, and Len Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[58] Ron Rothblum. Homomorphic encryption: From private-key to public-key. In *Theory of cryptography*, pages 219–234. Springer, 2011.

[59] Claus-Peter Schnorr. Block reduced lattice bases and successive minima. *Combinatorics, Probability and Computing*, 3(04):507–522, 1994.

[60] Nicolas Sendrier. *On the Security of the McEliece Public-Key Cryptosystem*, pages 141–163. Springer US, Boston, MA, 2002.

[61] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, October 1997.

[62] Nicolas Sklavos. Book review: Stallings, w. cryptography and network security: Principles and practice. *Information Security Journal: A Global Perspective*, 23:49–50, 2014.

[63] N.P. Smart and F. Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In PhongQ. Nguyen and David Pointcheval, editors, *Public Key Cryptography - PKC 2010*, volume 6056 of *Lecture Notes in Computer Science*, pages 420–443. Springer Berlin Heidelberg, 2010.

[64] Damien Stehl and Ron Steinfeld. Faster fully homomorphic encryption. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 377–394. Springer Berlin Heidelberg, 2010.

[65] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *Foundations of Computer Science, 1986., 27th Annual Symposium on*, pages 162–167. IEEE, 1986.

[66] Zhenfei Zhang, Thomas Plantard, and Willy Susilo. On the cca-1 security of somewhat homomorphic encryption over the integers. In *Information Security Practice and Experience*, pages 353–368. Springer, 2012.

[67] Zhenfei Zhang, Thomas Plantard, and Willy Susilo. Reaction attack on outsourced computing with fully homomorphic encryption schemes. In *Information Security and Cryptology-ICISC 2011*, pages 419–436. Springer, 2012.