HOMOMORPHIC ENCRYPTION: A COMPREHENSIVE STUDY OF TYPES, TECHNIQUES, AND REAL-WORLD APPLICATIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EZGİ NAZ TEKİN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CRYPTOGRAPHY

SEPTEMBER 2023

Approval of the thesis:

HOMOMORPHIC ENCRYPTION: A COMPREHENSIVE STUDY OF TYPES, TECHNIQUES, AND REAL-WORLD APPLICATIONS

submitted by **EZGİ NAZ TEKİN** in partial fulfillment of the requirements for the degree of **Master of Science in Cryptography Department, Middle East Technical University** by,

Prof. Dr. A. Sevtap Selçuk Kestel _____
Dean, Graduate School of **Applied Mathematics**

Assoc. Prof. Dr. Oğuz Yayla _____
Head of Department, **Cryptography**

Prof. Dr. Ferruh Özbudak _____
Supervisor, **Mathematics, METU**

Examining Committee Members:

Assist. Prof. Dr. Buket Özkaya
Cryptography, METU _____

Prof. Dr. Ferruh Özbudak
Mathematics, METU _____

Assist. Prof. Dr. Eda Tekin
Business Administration, Karabük University _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: EZGİ NAZ TEKİN

Signature :

ABSTRACT

HOMOMORPHIC ENCRYPTION: A COMPREHENSIVE STUDY OF TYPES, TECHNIQUES, AND REAL-WORLD APPLICATIONS

TEKİN, EZGİ NAZ
M.S., Department of Cryptography
Supervisor : Prof. Dr. Ferruh Özbudak

September 2023, 49 pages

Homomorphic encryption (HE), which enables computations on encrypted data without first decrypting it, is a ground-breaking advancement in the cryptographic area. In this study, many HE schemes such as partially, somewhat, and fully are examined and the details of the algorithms in these methods are given. Furthermore, this comprehensive examination goes beyond theoretical considerations to include real-world applications of this encryption technique. It examines both the potential and the challenges of using HE as a powerful tool for ensuring security and privacy in current information systems. This exploration takes into account not only the theoretical foundations but also the practical aspects of its implementation.

Keywords: homomorphic encryption, partially homomorphic encryption, somewhat homomorphic encryption, fully homomorphic encryption, etc.

ÖZ

HOMOMORFİK ŞİFRELEME: TÜRLERİ, TEKNİKLERİ VE GERÇEK DÜNYA UYGULAMALARI ÜZERİNE KAPSAMLI BİR ÇALIŞMA

TEKİN, EZGİ NAZ
Yüksek Lisans, Kriptografi Bölümü
Tez Yöneticisi : Prof. Dr. Ferruh Özbudak

Eylül 2023, 49 sayfa

Homomorfik şifreleme (HŞ), şifrelenmiş verinin şifresi çözülmeden üzerinde hesaplama yapılmasını sağlayan, kriptografik alanda çığır açan bir gelişmedir. Bu çalışmada, kısmen, biraz ve tamamen gibi birçok HŞ şeması incelenmiş ve bu yöntemlerdeki algoritmaların detayları verilmiştir. Ayrıca, bu kapsamlı inceleme teorik değerlendirmelerin ötesine geçerek bu şifreleme tekniğinin gerçek dünyadaki uygulamalarını da içermektedir. HŞ'nin mevcut bilgi sistemlerinde güvenlik ve gizlilik sağlamak için güçlü bir araç olarak kullanılmasının hem potansiyelini hem de zorluklarını incelemektedir. Bu inceleme sadece teorik temelleri değil aynı zamanda uygulamanın pratik yönlerini de dikkate almaktadır.

Anahtar Kelimeler: homomorfik şifreleme, kısmen homomorfik şifreleme, biraz homomorfik şifreleme, tamamen homomorfik şifreleme vd.

ACKNOWLEDGMENTS

I would like to express my gratitude to Professor Dr. Ferruh Özbudak for all the crucial advice given to me while preparing this thesis. I would like to express my appreciation to my esteemed thesis defense committee members, Assist. Prof. Dr. Buket Özkaya and Assist. Prof. Dr. Eda Tekin, for their dedicated time and effort.

I would like to thank my dear family - my beloved mother, Gül Tekin, for providing us with constant love and support; my dear father, Tarık Tekin, for sharing his valuable life experiences and wisdom with us; and my lovely sister, Zeynep Aslı Tekin, for being my best friend and confidant throughout my life.

I would also like to thank my beloved husband Eray İşbilir, for providing unwavering support throughout my entire journey at METU.

I would also like to thank my friends, for listening and supporting me during this process.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

HE	Homomorphic Encryption
PHE	Partially Homomorphic Encryption
SwHE	Somewhat Homomorphic Encryption
FHE	Fully Homomorphic Encryption
SVP	Shortest Vector Problem
CVP	Closest Vector Problem
$\mathbb{Z}$	Integers
$\mathbb{R}$	Real Numbers
$\mathbb{Z}_n$	Set of congruence classes modulo n
$\mathbb{Z}_n^*$	Multiplicative group of $\mathbb{Z}_n$
$gcd(a, b)$	Greatest common divisor of two non-zero integers a, b

CHAPTER 1

INTRODUCTION

In the digital age, where data is the lifeblood of modern society, the need to balance innovation and privacy has become a paramount concern. As the volume and sensitivity of data continue to grow, maintaining the confidentiality and security of information while enabling its analysis and use has become a formidable challenge. A breakthrough solution that has received considerable attention is Homomorphic Encryption (HE). HE is a revolutionary cryptographic technique that promises to allow computations to be performed on encrypted data without the need for decryption; in other words, it allows mathematical operations to be performed on encrypted data in such a way that the results of those operations are still encrypted while maintaining the confidentiality of the underlying data, thus bridging the gap between data privacy and data utility. It has the potential to reshape the way we handle sensitive information, enabling secure data analysis, secure computation outsourcing, and privacy-preserving machine learning, among other transformative applications. This thesis embarks on a comprehensive exploration of HE, delving into its underlying principles, types, applications, and the challenges it poses and addresses. [35, 17, 11].

1.1 Historical Process and Literature Review

The historical process and literature review of HE reveal a fascinating evolution of cryptographic techniques aimed at preserving data privacy while enabling computation on encrypted data. This review traces the key milestones and significant research contributions in the field of HE. The roots of HE can be traced back to the early de-

velopment of public-key cryptography and the concept of homomorphic properties. Researchers like Rivest, Adleman, and Dertouzos [30] made foundational contributions to public-key encryption, setting the stage for future developments. The seminal work of Craig Gentry [11] in 2009 marked a watershed moment in the field. Gentry's breakthrough introduced the concept of Fully Homomorphic Encryption (FHE), which allows arbitrary computations on encrypted data. His work ignited a surge of interest and research in the area. Over time, various types of HE emerged.

- **Partially Homomorphic Encryption (PHE)** was the earliest form, allowing either addition or multiplication on encrypted data.
- **Somewhat Homomorphic Encryption (SwHE)** struck a balance between efficiency and functionality.
- **Fully Homomorphic Encryption (FHE)** extended the capabilities further, enabling both addition and multiplication operations on encrypted data.

HE has found widespread applications in various fields, including healthcare, finance, cloud computing, and machine learning. It has been employed for secure data analysis, privacy-preserving machine learning, and secure computation outsourcing. While HE offers groundbreaking capabilities, it has presented significant challenges, including issues related to security proofs, computational efficiency, and usability. Balancing security and performance has been a central concern.

The field of HE is rapidly evolving, with ongoing research aiming to improve efficiency, reduce computational overhead, and enhance security guarantees. Standardization efforts are also underway to establish industry benchmarks and protocols for broader adoption. As the digital landscape evolves, HE holds great potential for addressing emerging privacy and security challenges. Future research may focus on post-quantum security, user-friendly interfaces, and practical implementations in real-world scenarios.

In summary, the historical process and literature review of HE reveal a journey marked by innovation, challenges, and transformative potential. This cryptographic technique has the power to reshape data privacy and computation in an increasingly interconnected and data-driven world.

1.2 Outline

This thesis examines homomorphic encryption over seven chapters.

In Chapter 2, definitions of important concepts such as group, ring, and homomorphism are given.

In chapter 3, the concept of partially homomorphic encryption is introduced, and a detailed exploration of algorithms like RSA, Goldwasser-Micali, ElGamal, Benaloh, and Paillier is provided.

In Chapter 4, the concept of somewhat homomorphic encryption is introduced, and a detailed exploration of algorithms like Boneh-Goh-Nissim is provided.

In Chapter 5, the concept of fully homomorphic encryption is introduced, and both ideal lattice-based and integer-based variants are conducted in detail.

In Chapter 6, real-life applications of homomorphic encryption are given, and the importance of this encryption method is explained.

CHAPTER 2

PRELIMINARIES

The basic mathematical concepts from number theory and abstract algebra used in the following chapters are defined in this chapter.

2.1 Group

A group is a fundamental algebraic construct comprising a collection of elements and an operation that combines any pair of elements to generate a third element.[35, 15]
Definition 2.1. A group is conventionally marked as $(G, \boxdot)$, where G symbolizes the set and $\boxdot$ represents the operation. The set and the operation must satisfy four main properties in order to be counted as a group:

- **Closure:** For any two elements α and λ in the set G, the outcome of the operation $\alpha \boxdot \lambda$ must also be in the set G.

- **Associativity:** For any three elements α, λ, and β in the set G, the outcome of the operation $(\alpha \boxdot \lambda) \boxdot \beta$ is the same as $\alpha \boxdot (\lambda \boxdot \beta)$.

- **Identity:** There exists an element e in the set G in such a way that for any element λ in the set, $\lambda \boxdot e = e \boxdot \lambda = \lambda$.

- **Inverse:** For each element λ in the set G, there exists an element λ^{-1} in such a way that $\lambda \boxdot \lambda^{-1} = \lambda^{-1} \boxdot \lambda = e$, where e is the identity element.

2.2 Ring

Definition 2.2. A ring is a set R along with two operations, addition, and multiplication, that correspond to the eight properties listed below: [35]:

R is an abelian group under addition:

- **Associativity:** For all α, λ, β in R, $(\alpha + \lambda) + \beta = \alpha + (\lambda + \beta)$.
- **Identity:** There exists an element 0 in R such that for any element α, $\alpha + 0 = 0 + \alpha = \alpha$. This means that 0 is the additive identity.
- **Inverse:** For each element α in R, there exists an element $-\alpha$ such that $\alpha + (-\alpha) = (-\alpha) + \alpha = 0$. This means that $-\alpha$ is the additive inverse of α.
- **Commutative:** For all α and λ in R, $\alpha + \lambda = \lambda + \alpha$.

R is a monoid under multiplication:

- **Associativity:** For all α, λ, β in R, $(\alpha \cdot \lambda) \cdot \beta = \alpha \cdot (\lambda \cdot \beta)$.
- **Identity:** There exists an element 1 in R such that for any element α, $\alpha \cdot 1 = 1 \cdot \alpha = \alpha$. This means that 1 is the multiplicative identity.
- **Left distributivity:** For all α, λ, β in R, $\alpha \cdot (\lambda + \beta) = (\alpha \cdot \lambda) + (\alpha \cdot \beta)$.
- **Right distributivity:** For all α, λ, β in R, $(\lambda + \beta) \cdot \alpha = (\lambda \cdot \alpha) + (\beta \cdot \alpha)$.

2.3 Group Homomorphism

A group homomorphism is a function that establishes a connection between two groups and guarantees the preservation of the essential structure of the groups. [35]
Definition 2.3. When considering two groups (G, $\boxdot$) and (H, $\circledast$), a group homomorphism from(G, $\boxdot$) to (H, $\circledast$) can be defined as a function f : G $\rightarrow$ H. This function follows the rule that for any elements β and $\beta^{'}$ in group G:

$$f(\beta \boxdot \beta^{'}) = f(\beta) \circledast f(\beta^{'})$$

To put it another way, this function preserves the relationships that are defined by the group operation as the elements are converted from one group to the other.

2.3.1 Example:

Consider two groups, G = $\{1, -1\}$ with multiplication as the operation, and H = $\{1, 2, 3\}$ with addition as the operation. Determine a function $\varkappa : G \to H$in such a way that:

$$\varkappa(1) = 1$$
$$\varkappa(-1) = 3$$

This function is a group homomorphism because it preserves the group operation:

$$\varkappa(1 * 1) = \varkappa(1) * \varkappa(1) = 1 * 1 = 1$$
$$\varkappa(1 * -1) = \varkappa(1) * \varkappa(-1) = 1 * 3 = 3$$

However, if we determined a function $\Gamma : G \to H$ such that:

$$\Gamma(1) = 2$$
$$\Gamma(-1) = 3$$

This would not be a group homomorphism because it doesn't preserve the group operation:

$$\Gamma(1 * -1) = \Gamma(-1) = 3$$

but

$$\Gamma(1) * \Gamma(-1) = 2 * 3 = 6$$

2.4 Ring Homomorphism

A ring homomorphism is a function that establishes a connection between two rings and guarantees the preservation of the essential structure of the rings. [35]

Definition 2.4. When considering two rings R and S, then a ring homomorphism is a function $\Gamma : R \to S$ such that

$$\Gamma(\alpha + \lambda) = \Gamma(\alpha) + \Gamma(\lambda)$$
$$\Gamma(\alpha \cdot \lambda) = \Gamma(\alpha) \cdot \Gamma(\lambda)$$

for all α and λ in R.

2.4.1 Example:

Consider two rings, R- The ring of integers ($\mathbb{Z}$) and S - The ring of even integers ($2\mathbb{Z}$) with regular addition and multiplication as operations in both of them. Determine a function:

$$\Gamma : \mathbb{Z} \to 2\mathbb{Z}$$
$$n \mapsto 2n$$

For any integer n, $\Gamma(n)$ is twice the value of n, i.e., $\Gamma(n) = 2n$.

- For any integers a and b, we have:

$$\begin{aligned}\Gamma(a + b) &= 2(a + b)\\ &= 2a + 2b\\ &= \Gamma(a) + \Gamma(b)\end{aligned}$$

  The function preserves addition.

- For any integers a and b, we have:

$$\begin{aligned}\Gamma(a \cdot b) &= 2(a \cdot b)\\ &= (2a) \cdot (2b)\\ &= \Gamma(a) \cdot \Gamma(b)\end{aligned}$$

  The function preserves multiplication.

- The multiplicative identity in both rings is 1. In R, $\Gamma(1) = 2$ and in S, $\Gamma(1) = 2$. Thus, the multiplicative identity is preserved.

The function Γ satisfies the conditions required for a ring homomorphism between $\mathbb{Z}$ and $2\mathbb{Z}$.

2.5 Homomorphic Encryption

HE is an approach that enables the execution of calculations on encrypted data, producing outcomes that mirror the results of performing the identical computation on data that is not encrypted.

Definition 2.5. HE scheme with encryption (E) over an operation $\boxdot$ as follows:

$$E(m_1) \boxdot E(m_2) = E(m_1 \boxdot m_2)$$

$\forall m_1, m_2 \in M$ where $M =$ the space of the messages.[35]

The HE scheme has four main algorithms:

- **Key Generation:** It provides a security parameter. An asymmetric HE system creates a pair of keys, one secret and one public, but in a symmetric HE scheme, it generates a single key.
- **Encryption:** It processes a plaintext message m from the set of possible messages in M using the encryption key to generate a ciphertext c, represented as $E(m)$ in the ciphertext space C.
- **Decryption:** It employs the ciphertext c in combination with the decryption key to extract the plaintext message m, represented as $D(c) = m$.
- **Evaluation (Homomorphic Property):** It gets input in the form of ciphertexts, denoted as c_1 and c_2 and conducts a computational operation, represented by the function f on these ciphertexts, resulting in evaluated ciphertexts $f(c_1, c_2) = E(f(m_1, m_2))$ all without gaining access to the underlying messages, i.e., it doesn't see the contents of m_1 and m_2. In other words, decryption of $f(c_1, c_2)$ is equal to $f(m_1, m_2)$.

CHAPTER 3

PARTTIALLY HOMOMORPHIC ENCRYPTION

Definition 3.1. Partially Homomorphic Encryption (PHE) is a type of HE that facilitates computations on encrypted data for only one specific type of operation, such as addition or multiplication. [17]

There are two primary categories of PHE schemes:

1. **PHE for Addition:** This type of PHE allows encrypted data to be subjected to homomorphic addition. With PHE for addition, we can perform addition operations on encrypted data without the need for decryption. However, other operations, such as multiplication, cannot be directly performed on the encrypted data using this type of encryption.

2. **PHE for Multiplication:** This type of PHE allows encrypted data to be subjected to homomorphic multiplication. Similar to PHE for addition, we can perform multiplication operations on encrypted data without decryption, but other operations like addition are not directly supported.

PHE schemes have practical applications in specific scenarios where only one type of operation is needed, and FHE might be computationally prohibitive or not yet available for certain use cases. However, they do not provide the same level of flexibility and versatility as FHE when it comes to performing arbitrary computations on encrypted data. In this chapter, we mention some PHE algorithms such as Goldwasser-Micali(GM), RSA, Elgamal, Paillie, Benaloh, etc.

3.1 Rivest-Shamir-Adleman Algorithm

In 1978, Ron Rivest, Adi Shamir, and Leonard Adleman [30] introduced the RSA algorithm, deriving its name from the initial letters of their surnames. This method involves the use of mathematical operations on significant prime numbers to create reliable encryption and digital signature systems based on Diffie Helman [6] developed public key cryptography in 1976. Their work is of immense importance in modern cryptography and serves as the fundamental framework for a wide variety of secure online communications and transactions. RSA security depends on the RSA Root Problem, which involves discovering e-th roots modulo k, where k is equal to pq, and when the factorization of k is undisclosed.

3.1.1 Key Generation

RSA utilizes both public and private keys in its encryption and decryption processes. The public key consists of a pair of integers (k, e), where k is the product of two distinct large prime numbers, p and q (meaning that $k = pq$). The value of e is selected in such a way that it does not share any common factors with $\phi(k)$, where $\phi(k)$ is calculated as $(p-1)$ multiplied by $(q-1)$, and specifically, e has a modular multiplicative inverse within $\phi(k)$(i.e., $gcd(e, \phi(k))$ = 1 where 1 < e < $\phi(k)$). On the other hand, the private key is represented as (d, k), with d being determined such that it is the modular inverse of e (meaning that $ed \equiv 1 \pmod{\phi(k)}$).

3.1.2 Encryption

After turning the message M into an integer m such that 0 < m < k through the application of a padding scheme to encrypt the message as follows:

$$\begin{aligned} E(m) &= m^e \pmod{k} \\ &= c \end{aligned}$$

where c is the ciphertext.

3.1.3 Decryption

For decrypting the ciphertext c with the private key (d, k) as follows:

$$\begin{aligned} D(c) &= c^d \pmod{k} \\ &= (m^e)^d \pmod{k} \\ &= m \end{aligned}$$

where $ed \equiv 1 \pmod{\phi(k)}$. Afterward, the message m is converted with the padding scheme to retrieve the original message M (unpadded plaintext).

3.1.4 Homomorphic Property

Within an unpadded RSA setting, if we consider the public key as (k, e), then the following function determined [17] as follows:

$$\begin{aligned} E : \mathbb{Z}_k &\to \mathbb{Z}_k \\ m &\mapsto m^e \end{aligned}$$

Consider two padded plaintexts m_a and m_b in $\mathbb{Z}_k$. Then

$$\begin{aligned} E(m_a) \times E(m_b) &= (m_a^e \pmod{k}) \times (m_b^e \pmod{k}) \\ &= m_a^e \times m_b^e \pmod{k} \\ &\equiv (m_a \times m_b)^e \pmod{k} \\ &= E(m_a \times m_b) \end{aligned}$$

Hence

$$D(E(m_a) \times E(m_b)) = D(E(m_a \times m_b))$$

As observed, the RSA multiplication homomorphic characteristic allows for the direct evaluation of $E(m_a \times m_b)$ using $E(m_a)$ and $E(m_b)$ without requiring decryption. RSA does not exhibit homomorphism in regard to addition. In the RSA cryptosystem, it is not possible to directly combine two encrypted ciphertexts through addition without prior decryption.

3.2 Goldwasser-Micali Algorithm

The Goldwasser-Micali (GM) algorithm was introduced by Shafi Goldwasser and Silvio Micali in 1982 [14]; it is the first public-key encryption algorithm that forms the basis for probabilistic encryption. This algorithm provides a secure method for encrypting data while allowing for efficient decryption, making it suitable for various applications in secure communication and data protection. The core concept of the GM algorithm lies in its use of quadratic residues, a fundamental concept in number theory. Quadratic residues are integers that have square roots modulo a prime number. The algorithm leverages the computational intractability of the Quadratic Residuosity Problem (QRP) [12] to achieve semantic security in encryption. The security of the GM scheme is contingent on the Integer Factorization Problem because the factorization of $k = pq$ unveils the private key (p, q).

3.2.1 Key Generation

GM utilizes both public and private keys in its encryption and decryption processes. The public key consists of a pair of integers (x, k), and the secret key is (p, q), where k is the product of two distinct large prime numbers, p, and q (meaning that $k = pq$). On the other hand, we choose a quadratic non-residue $x \in \mathbb{Z}_k^*$ to find the Jacobi symbol with the Legendre symbols $\left(\frac{x}{p}\right)$= $\left(\frac{x}{q}\right)$ = - 1 such that

$$\begin{aligned}\left(\frac{x}{p}\right) \cdot \left(\frac{x}{q}\right) &= (-1) \cdot (-1) = 1 \\ &= \left(\frac{x}{k}\right)\end{aligned}$$

3.2.2 Encryption

The plaintext consists of a series of binary bits $(m_1, m_2, \ldots, m_t)$. For every bit m_i for $i = 1, 2, \ldots, t$, y_i is uniformly selected from the set of non-zero residues modulo k (i.e., $\mathbb{Z}_k^*$) such that $gcd(y_i, k) = 1$. Performing encryption on a per-bit basis by

utilizing computations:

$$\begin{aligned} E(m_i) &= y_i^2 x^{m_i} \pmod{k} \\ &= c_i \end{aligned}$$

where for every bit m_i is in the set $\{0, 1\}$ for $i = 1, 2, \ldots, t$. The ciphertext generated is $(c_1, c_2, \ldots, c_t)$, such that $c_i \in \mathbb{Z}_k$ for $i = 1, 2, \ldots, t$.

3.2.3 Decryption

There is no requirement for a distinct decryption algorithm because, for an element x belonging to the set $\mathbb{Z}_k^*$, the ciphertext c_i is considered a quadratic residue modulo k exclusively when m_i equals 0. Therefore, if c_i is a quadratic residue modulo k, then m_i is 0; otherwise, m_i is 1. This process results in the plaintext m being represented as a sequence $(m_1, \ldots, m_t)$.

3.2.4 Homomorphic Property

The encryption function [17] employed in the GM is as follows:

$$\begin{aligned} E : (\{0, 1\}, \oplus) &\to (\mathbb{Z}_k^*, \times) \\ m &\mapsto y^2 \times x^m \end{aligned}$$

where $\oplus$ represents addition modulo 2. For bits m_a and m_b in $\{0, 1\}$ where $a, b \in \{1, 2, \ldots, t\}$:

$$\begin{aligned} E(m_a) \times E(m_b) &= (y_a^2 x^{m_a} \pmod{k}) \times (y_b^2 x^{m_b} \pmod{k}) \\ &= (y_a \times y_b)^2 x^{m_a \oplus m_b} \pmod{k} \\ &= E(m_a \oplus m_b) \end{aligned}$$

Hence,

$$D(E(m_a) \times E(m_b)) = D(E(m_a \oplus m_b))$$

Therefore, the GM algorithm is an additive homomorphic scheme. Nonetheless, E does not possess a multiplicative homomorphic property.

3.3 ElGamal Algorithm

The Elgamal encryption method is a public-key encryption algorithm introduced by Taher Elgamal [9] in 1985 based on the complexity of discrete logarithm problem [12]. The security of the ElGamal algorithm relies on the challenge of solving the Diffie-Hellman Problem (DHP) within the set Z_p^*. The Diffie-Hellman key exchange technique is effectively transformed into an encryption algorithm using the ElGamal.

3.3.1 Key Generation

After choosing large primes l and k such that $k|(l-1)$, choose a cyclic subgroup G_k of Z_l^*, of order k, with generator g. Then, compute

$$t = g^h \pmod{l}$$

for selected random $h \in \{1, 2, \ldots, k-1\}$. The published public key is (k, l, g, t), and the secret key is h.

3.3.2 Encryption

For encrypting a message m with the public key to $m^{'} \in G_k$, there must be chosen a random $r \in \{1, 2, \ldots, k-1\}$ to compute

$$c_1 = g^r \pmod{l}$$
$$c_2 = m^{'} \cdot t^r \pmod{l}$$

The ciphertext is the pair $(c_1, c_2) = (g^r, m^{'}t^r)$.

3.3.3 Decryption

For decrypting ciphertext with the secret key to compute

$$\begin{aligned} D((c_1, c_2)) &= c_2(c_1^h)^{-1} \pmod{l} \\ &= m^{'} \cdot g^{hr} \cdot g^{-hr} \pmod{l} \\ &= m^{'} \pmod{l} \end{aligned}$$

Therefore, $m^{'}$ is converted to the plaintext message m.

3.3.4 Homomorphic Property

The encryption process in the ElGamal [17] is determined as follows:

$$E : (G_k, \times) \to (G_k \times G_k, \times)$$
$$m \mapsto (g^r, m \times t^r)$$

where the symbol $\times$ represents the standard multiplication, and $G_k \times G_k$ signifies the direct combination of the group G_k with itself. For all m_a and m_b in G_k where $a, b \in \{1, 2, \ldots, k-1\}$:

$$\begin{aligned} E(m_a) \times E(m_b) &= (g^{r_a}, m_a \cdot t^{r_a}) \times (g^{r_b}, m_b \cdot t^{r_b}) \\ &= (g^{r_a} g^{r_b}, m_a t^{r_a} \cdot m_b t^{r_b}) \\ &= (g^{r_a + r_b}, (m_a m_b) t^{r_a + r_b}) \\ &= E(m_a \times m_b) \end{aligned}$$

Hence,

$$D(E(m_a) \times E(m_b)) = D(E(m_a \times m_b))$$

Therefore, the ElGamal is a multiplicative homomorphic scheme.

3.4 Benaloh Algorithm

Benaloh introduced an expansion of the GM scheme aimed at encrypting the entire message as a single unit. [2] Benaloh's security relies on leveraging the higher residuosity problem (x^n), which can be viewed as a broader and more generalized form of the quadratic residuosity problems (x^2).[12]

3.4.1 Key Generation

Let give a block size t, the two large primes p and q are chosen such that

$$t|(p-1)$$

$$gcd(t, (q-1)) = 1$$

$$gcd(t, (p-1)/t) = 1$$

Compute $k = pq$ and $\phi(k) = (p-1)(q-1)$.
Also, choose $y \in \mathbb{Z}_k^*$ such that $y^{\phi(k)/t} \neq 1 \pmod{k}$. Finally,
set $x = y^{\phi(k)/t} \pmod{k}$, then the secret key is (p, q), and the public key is (k, y).

3.4.2 Encryption

Let $m \in \mathbb{Z}_t$ be a message, needs to take $u \in Z_k^*$ then

$$\begin{aligned} E(m) &= y^m u^t \pmod{k} \\ &= c \end{aligned}$$

3.4.3 Decryption

Compute $c^{\phi(k)/t} \pmod{k}$ since $u \in \mathbb{Z}_k^*$ and $m \in \mathbb{Z}_t$:

$$\begin{aligned} c^{\phi(k)/t} \pmod{k} &\equiv (y^m u^t)^{\phi(k)/t} \pmod{k} \\ &\equiv (y^m)^{\phi(k)/t} (u^t)^{\phi(k)/t} \pmod{k} \\ &\equiv (y^{\phi(k)/t})^m (u)^{\phi(k)} \pmod{k} \\ &\equiv x^m \pmod{k} \end{aligned}$$

where $u^{\phi(k)} \equiv 1 \pmod{k}$. If t is small, then m is recovered by an exhaustive search, i.e. $x^i \equiv c^{\phi(k)/t} \pmod{k}$ for $i = 0, \ldots t-1$. When dealing with larger values of t, the baby-step giant-step algorithm can be used to compute the discrete logarithm to recover m.

3.4.4 Homomorphic Encryption

The encryption function of the Benaloh [17] is determined as follows:

$$\begin{aligned} E : (\mathbb{Z}_t, +) &\to (\mathbb{Z}_k^*, \times) \\ m &\mapsto y^m \times u^t \end{aligned}$$

where $+$ and $\times$ represent addition and multiplication operations, respectively. For all m_a and m_b in Z_t

$$\begin{aligned} E(m_a) \times E(m_b) &= (y^{m_a} u_a^t \pmod k) \times (y^{m_b} u_b^t \pmod k) \\ &= y^{m_a+m_b}(u_a \times u_b)^t \pmod k \\ &= E(m_a + m_b) \end{aligned}$$

Hence,

$$D(E(m_a) \times E(m_b)) = D(E(m_a + m_b))$$

Therefore, the Benaloh algorithm is an additive homomorphic scheme. However, this algorithm is multiplicatively and additively homomorphic with an integer scalar h. i.e. $E(m)^h = (y^m)^h (u^t)^h \pmod k = E(h \times m)$.

3.5 Paillier Algorithm

The Paillier encryption method is the probabilistic public-key encryption algorithm, initially introduced by Pascal Paillier [27] in 1999. The security of the Paillier algorithm relies on a composite residuosity problem, [12]where computing the nth residue classes is considered a complex computational challenge. The fundamental principle underlying the algorithm concerns the decisional composite residuosity assumption, a hypothesis related to intractability.

3.5.1 Key Generation

Let p and q be two large prime numbers such that

$$gcd(p, q-1) = 1$$
$$gcd(p-1, q) = 1$$

to compute $k = pq$. Then, compute

$$\lambda(k) = lcm(p-1, q-1)$$

where lcm refers to the least common multiple. Then select a random integer g $\in Z^*_{k^2}$ such that

$$\mu = (L(g^{\lambda(k)} \pmod{k^2}))^{-1} \pmod k$$

where function L is defined as

$$L(u) = \frac{u-1}{k}$$

$\forall u \in Z^*_{k^2}$. Therefore, the public key is (k, g), and the secret key is (λ, μ).

3.5.2 Encryption

Let $m \in Z_k$ be a message where $0 \leq m < k$. Then, select random $r \in Z^*_{k^2}$ such that $0 < r < k$. Afterwards, compute the ciphertext as

$$\begin{aligned} E(m) &= g^m \times r^k \pmod{k^2} \\ &= c \end{aligned}$$

3.5.3 Decryption

Let $c \in Z^*_{k^2}$ be a ciphertext. Then, compute the plaintext message as

$$\begin{aligned} D(c) &= \frac{L(c^{\lambda(k)} \pmod{k^2})}{L(g^{\lambda(k)} \pmod{k^2})} \pmod{k} \\ &= m \end{aligned}$$

3.5.4 Homomorphic Property

The encryption function can be determined [17] as for $k = pq$ and $g \in \mathbb{Z}^*_{k^2}$:

$$\begin{aligned} E_g : \mathbb{Z}_k \times \mathbb{Z}^*_k &\to \mathbb{Z}^*_{k^2} \\ (m, r) &\mapsto g^m \times r^k \end{aligned}$$

For all $m_a, m_b \in Z_k$ and $r_a, r_b \in Z^*_k$, then:

$$\begin{aligned} E_g(m_a, r_a) \times E_g(m_b, r_b) &= (g^{m_a} r_a^k \pmod{k^2}) \times (g^{m_b} r_b^k \pmod{k^2}) \\ &= g^{m_a + m_b}(r_a \times r_b)^k \pmod{k^2} \\ &= E(m_a + m_b, r_a \times r_b) \end{aligned}$$

Hence,

$$D(E(m_a) \times E(m_b)) = D(E(m_a + m_b))$$

Therefore, the Paillier is an additive homomorphic scheme.

3.6 Other PHE Algorithms

Various initiatives have aimed to improve the flexibility of current PHE schemes following: The Naccache-Stern (NS) algorithm [24], introduced by D. Naccache and J. Stern in 1998, can be seen as an extension of the Benaloh algorithm. This NS algorithm comes in two variants: a deterministic and a probabilistic, with the latter being derived through minor adjustments to the former. Both of these versions function as homomorphic encryption schemes. Likewise, The Damgard-Jurik algorithm [5], created by Ivan Damgard and Mads Jurik in 2001, can be considered an expansion of the Paillier algorithm. This algorithm naturally extends Paillier's scheme to the group $Z^*_{n^{s+1}}$. In addition, The Okamoto-Uchiyama algorithm, presented by Tatsuaki Okamoto and Shigenori Uchiyama in 1998 [26], relies on the security principles of the Integer Factorization Problem, specifically when n is represented as the product of two k-bit primes, $n = p^2 \times q$.

CHAPTER 4

SOMEWHAT HOMOMORPHIC ENCRYPTION

All homomorphic encryption schemes created before 2005 consisted of only one operation such as addition and multiplication. Only for a limited number of repetitions can Somewhat Homomorphic Encryption (SwHE) successfully achieve both additive and multiplicative operations. The scheme's ability to reliably decrypt ciphertexts associated with homomorphic operations defines this limitation.

4.1 BGN Algorithm

The Boneh-Goh-Nissim algorithm [3] is a public-key encryption that was introduced by Boneh, Goh, and Nissim in 2005. The BGN encryption scheme bears similarities to the Paillier and Okamoto-Uchiyama encryption methods. Specifically, the BGN scheme was the pioneering technique that enabled additions and multiplications using fixed-size ciphertext. This scheme's security is based on the Subgroup Decision Problem, [12], which determines if an element x in a cyclic group G belongs to a specific subgroup within G. The group has an order of $|G| = k$, where k equals $q_a \times q_b$ with q_a and q_b chosen as distinct large prime numbers.

4.1.1 Key Generation

Choose two large primes q_a and q_b to compute $k = q_a \times q_b$ and a larger prime p such that $p = l \times k - 1$ for some positive integer l and e is a bilinear map such that

$$e : G \times G \rightarrow G'$$

where $G' \subseteq \mathbb{Z}^*_{p^2}$. Choose two random generators g, u of G and set $h = u^{q_b}$. Then h is a random generator of the order q_a subgroup of G. The public key is (k, G, G', e, g, h), and the secret key is q_a.

4.1.2 Encryption

For encrypting a message $m \in \mathbb{Z}_{q_b}$, pick a random number $r \in \{0, 1, \ldots, k-1\}$ with g and h as follows:

$$\begin{aligned} E(m) &= g^m h^r \\ &= c \in G \end{aligned}$$

Each message is assigned a unique r-value on each occasion.

4.1.3 Decryption

For decrypting ciphertext c using secret key q_a. First of all, notice that:

$$\begin{aligned} c^{q_a} &= (g^m h^r)^{q_a} \\ &= (g^m)^{q_a} \times (h^r)^{q_a} \\ &= (g^m)^{q_a} \times ((u^{q_b})^r)^{q_a} \\ &= (g^{q_a})^m \times (u^{q_a \times q_b})^r \\ &= (g^{q_a})^m \in G \end{aligned}$$

Calculating the discrete logarithm of c^{q_a} with the base g^{q_a} using Pollard's lambda method [29]is adequate to decrypt as follows:

$$\begin{aligned} D(c) &= log_{g^{q_a}}(c^{q_a}) \\ &= log_{g^{q_a}}((g^{q_a})^m) \\ &= log_{g^{q_a}}(g^{q_a})^m \\ &= m \end{aligned}$$

4.1.4 Homomorphic Property

The BGN algorithm has both additive and multiplicative homomorphic properties.

4.1.4.1 Homomorphism over Addition

The encryption function can be determined for $k = q_a \times q_b$ and $g, u \in G^*$:

$$E_g : \mathbb{Z}_{q_b} \times \mathbb{Z}_k \to G$$
$$(m, r) \mapsto g^m \times h^r$$

where $r \in \mathbb{Z}_k$ is random and $\times$ represents the group operation of G. The BGN algorithm possesses additive homomorphism, where the corresponding r-value for the message sum of m_a and m_b is the sum of r_a and r_b.

$$\begin{aligned} E_g(m_a, r_a) \times E_g(m_b, r_b) &= (g^{m_a} \times h^{r_a}) \times (g^{m_b} \times h^{r_b}) \\ &= g^{m_a+m_b} \times h^{r_a+r_b} \\ &= E_g(m_a + m_b, r_a + r_b) \end{aligned}$$

Hence,

$$D(E(m_a) \times E(m_b)) = D(E(m_a + m_b))$$

Therefore, the BGN has homomorphic property over addition.

4.1.4.2 Homomorphism over Multiplication

The encryption function can be determined as:

$$E_{g_a} : \mathbb{Z}_{q_b} \times \mathbb{Z}_k \to G'$$
$$(m, r) \mapsto (g_a)^m \times (h_a)^r$$

where $r \in \mathbb{Z}_k$ is random and $\times$ represents the group operation of G'. The homomorphic multiplication of the BGN algorithm works with bilinear map, then $g_a = e(g, g)$ and $h_a = e(g, h)$ are chosen, where g_a with order k, $h_a$with order q_a, and let $\alpha \in \mathbb{Z}$ such that $h = g^\alpha$. Then, the encryption of multiplication of m_a and m_b by using

$c_a = g^{m_a} h^{r_a} \in G$ and $c_b = g^{m_b} h^{r_b} \in G$ is computed as follows:

$$\begin{aligned}
e(c_a, c_b) h_a^r &= e(g^{m_a} h^{r_a}, g^{m_b} h^{r_b}) h_a^r \\
&= e(g^{m_a + \alpha r_a}, g^{m_b + \alpha r_b}) h_a^r \\
&= e(g, g)^{(m_a + \alpha r_a)(m_b + \alpha r_b)} h_a^r \\
&= e(g, g)^{m_a m_b + \alpha(m_a r_b + m_b r_a + \alpha r_a r_b)} h_a^r \\
&= (g_a)^{m_a m_b} (g_a)^{\alpha(m_a r_b + m_b r_a + \alpha r_a r_b)} h_a^r \\
&= (g_a)^{m_a m_b} (h_a)^{(m_a r_b + m_b r_a + \alpha r_a r_b)} h_a^r \\
&= (g_a)^{m_a m_b} h_a^{r + m_a r_b + m_b r_a + \alpha r_a r_b} \\
&= E_{g_a}(m_a m_b, r + m_a r_b + m_b r_a + \alpha r_a r_b)
\end{aligned}$$

If the expression $r + m_a r_b + m_b r_a + \alpha r_a r_b$ is uniformly distributed like r, the resulting ciphertext c represents a uniformly distributed encryption of $m_a m_b$. However, it's important to highlight that c now resides in G' instead of G. It's worth noting that the BGN scheme retains its limitless additively homomorphic property in G'. Therefore, the BGN has homomorphic property over multiplication.

4.2 Other Algorithms

Melchor and colleagues, in their work [20], built upon established schemes with homomorphic properties to introduce an encryption scheme that facilitates the homomorphic evaluation of constant-depth circuits. Conversely, Sander, Young, and Yung (SYY) proposed a scheme [31] that allows for just one OR/NOT gate in combination with multiple AND gates. Nevertheless, the evaluation of circuit depth was restricted due to the fact that the ciphertext size grew with each evaluation of an OR/NOT gate.

CHAPTER 5

FULLY HOMOMORPHIC ENCRYPTION

In 2009, Craig Gentry [11] presented the initial FHE scheme as part of his doctoral thesis. This proposition of scheme also was served as a blueprint for constructing an FHE scheme to other schemes. As a result, scholars expanded upon Gentry's foundation, working towards the development of further effective FHE systems.

Definition 5.1. The Fully Homomorphic Encryption scheme (FHE) is capable of performing an unrestricted number of additive and multiplicative homomorphic operations on encrypted data. It allows for objective evaluation of data without compromising its confidentiality.

While Gentry's introduction of the FHE scheme was promising, its practical implementation was hindered by its high computational demands. Consequently, several modifications were made to enhance the scheme's viability for real-world usage. As the search for new FHE schemes persisted, many advancements prioritized addressing lattice problems.

FHE schemes can be classified into two primary categories according to the underlying problems they address:

- Gentry [11] initially introduced **the ideal lattice-based approach**, which prompted several subsequent researchers to enhance his FHE scheme rooted in ideal lattices, including the work of Smart and Vercauteren.

- Van Dijk and colleagues [7] proposed a scheme **based on over-integers**, founded upon the Approximate-GCD problems [10].

The following definitions are critical to understanding the ideal-based lattices:

Definition 5.2. (Lattice) In m-dimensional Euclidean space R^m, a lattice L is the set [21]

$$L = \{a_1 b_1 + a_2 b_2 + \cdots + a_n b_n | a_1, a_2, \ldots, a_n \in \mathbb{Z}\}$$

of a collection of linearly independent vectors, expressed as b_1 through b_n in R^m where $m \geqslant n$. n is called the rank and m is the dimension of the lattice. The sequence of vectors $\{b_1, \ldots, b_n\}$ is called a lattice basis. It is represented as the following matrix:

$$B = [b_1, \ldots, b_n] \in R^{m \times n}$$

where $m \geqslant n$. When n is equal to m, the lattice is referred to as being full-dimensional or having full rank. It's important to note that a lattice can have multiple distinct bases. As a result, lattices can be described and defined independently of any specific basis.

Within an infinitely regular n-dimensional grid, a lattice is defined as a collection of points where the grid lines intersect. It is important to note that the grid does not need to be perfectly orthogonal. Figure 5.1 illustrates this concept in a two-dimensional context—a lattice point results from the linear combination of the base vectors. The repeating pattern enclosed by these lattice points is known as the "Fundamental Parallelepiped," visible in black.

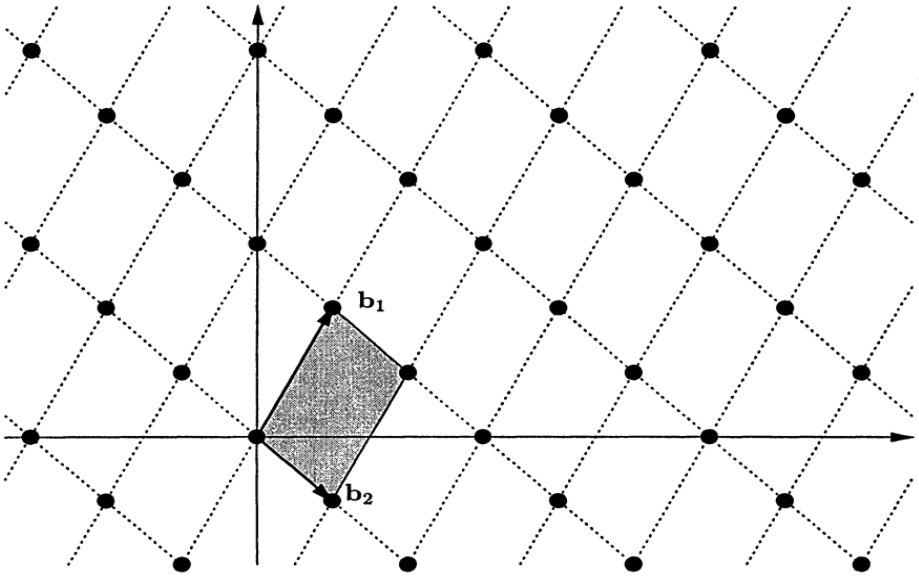

Figure 5.1: Lattice in R^2 [21]

The field of lattice theory was initially introduced by Minkowski in 1910. [22] Later,

Ajtai [1] introduced a category of random worst-case lattice problems known as the SVP and CVP in 1996. These problems have since found practical use in lattice-based cryptographic schemes. [28]

Definition 5.3. (Shortest Vector Problem, SVP) Given a basis $T \in \mathbb{Z}^{m \times n}$, find a nonzero lattice vector Tx (with $a \in \mathbb{Z}^n - \{0\}$) such that $|Ta| \leqslant |Tb|$ for any other b $\in \mathbb{Z}^n - \{0\}$, [15] i.e., using the specified basis T as a reference, SVP finds the shortest non-zero vector within the lattice.

Definition 5.4. (Closest Vector Problem, CVP) Given a basis $T \in \mathbb{Z}^{m \times n}$ and a target vector $t \in \mathbb{Z}^m$, find a lattice vector Ta closest to the target t,[15] i.e., find an integer vector $a \in \mathbb{Z}^n$ such that $|Ta - t| \leqslant |Tb - t|$ for any other $b \in \mathbb{Z}^n - \{0\}$, i.e., CVP determines the lattice point closest to a specified point within the lattice.

Definition 5.5. (Circuits) A circuit is essentially a graph with nodes and edges, specifically designed as an acyclic-directed structure. [15] The edges are referred to as **wires** while the nodes are termed **gates**. The type of gates used in the circuit, whether they represent logical operations (such as AND, OR, NOR, NAND, etc.) or arithmetic operations, depends on the input data contained within the circuit, including Boolean values, integers, and more. In figure 5.2, we observe the example circuit that symbolizes the function using AND and OR as logic gates.

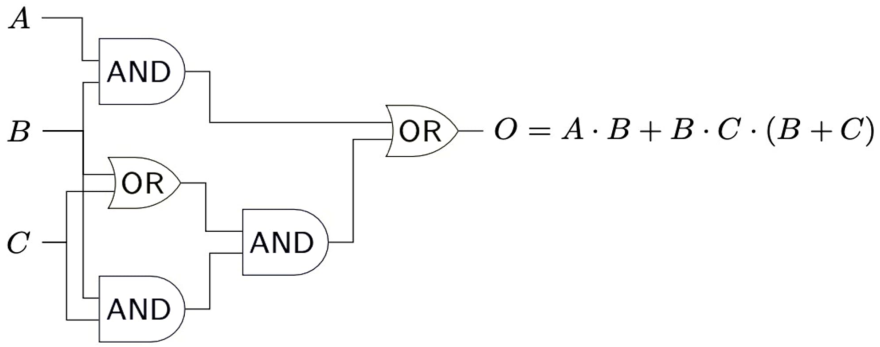

Figure 5.2: Sample Circuit [15]

5.1 Ideal-Lattice Based FHE

Craig Gentry is credited with developing the first encryption system capable of processing circuits with arbitrary depth. [11] His SwHE scheme is based on ideal lat-

tices and allows for homomorphic evaluation of ciphertexts, albeit with limitations on the number of iterations. This scheme is particularly suited for low-depth circuits. Within this SwHE scheme, a certain amount of noise is added to the plaintext during encryption. This noise increases with each addition and multiplication operation, corresponding to each gate in the circuit applied to the ciphertexts. Eventually, the noise reaches a level where it makes it impossible for the decryption algorithm to accurately recover the plaintext from the ciphertext.

To enable a ciphertext to undergo numerous homomorphic operations, Gentry employed a technique known as bootstrapping. Bootstrapping serves the purpose of reducing the noise within the ciphertext to an acceptable level. In order to execute bootstrapping successfully, the SwHE scheme must possess a decryption algorithm with low complexity, meaning it has a low-depth decryption circuit that can assess its own decryption process. In contrast to lattice-based encryption schemes, RSA and ElGamal encryption schemes have more intricate decryption algorithms, which renders them less suitable for FHE schemes.

The bootstrapping procedure efficiently revitalizes the ciphertext by reducing its noise, allowing it to be reused in additional addition and multiplication operations. Gentry's concept of "refreshing" involves several steps. Initially, a pair of encryption and decryption keys is generated. Data is then encrypted using the encryption key, and computations are performed on the encrypted data within the limitations of the SwHE scheme. Subsequently, a second pair of encryption and decryption keys is generated, and the results are encrypted under the second encryption key. To mitigate noise in the encrypted results, Gentry encrypts the first decryption key using the second encryption key. The decryption algorithm of the SwHE scheme is then applied through the evaluation algorithm, which decrypts the results initially encrypted under the first key but retains them encrypted under the second key. The computations can proceed as desired. Once the computations are finished, the final decryption key is employed to decrypt the results ciphertext, yielding the final computation result.

This process can be repeated iteratively, allowing the scheme to evaluate unlimited operations on the ciphertexts. However, it is essential to note that this scheme faces practical challenges because both the processing time and the volume of ciphertext

increase significantly. Gentry's blueprint for this procedure involves three main steps, constructing a SwHE scheme based on ideal lattices, squashing, and bootstrapping.

5.1.1 SwHE scheme based on ideal lattices

Gentry's SwHE scheme was built upon the same concepts as the GGH method proposed by Goldreich, Goldwasser, and Halevi [13] and was rooted in the domain of lattice reduction problems. [28] Lattice reduction aims to discover a reasonably short and orthogonal basis for a specific lattice, often referred to as a good basis for the lattice. Typically, solving the SVP and CVP can be achieved in polynomial time if one possesses knowledge of these good bases for a lattice. The LLL algorithm is one of the most efficient algorithms for this purpose, although it becomes exponential when dealing with unknown good bases of the lattice.

In this scheme, the public key is generated using a bad basis of the lattice, while a good basis is utilized to create the secret key. Noise is subsequently introduced to a lattice point to form the ciphertext. To decrypt the ciphertext, one must determine the closest lattice point using the secret key. In this SwHE scheme, ideals are represented using lattices. The scheme encompasses several algorithms, including the following:

5.1.1.1 Key Generation

This process begins with two inputs: a fixed ring R and a basis T_I representing a small ideal I within R. This basis is employed to embed the message into an error vector. Moreover, an algorithm called $IdealGen(R, T_I)$ is employed to generate both the public and secret keys. The public key includes a **bad basis** T^{pk}, which can be derived from the Hermite Normal Form (HNF) of the basis T^{sk} used in the secret key. T^{sk} comprises a **good basis** of the ideal lattice J, featuring short and nearly orthogonal vectors. It's important to note that the choice of the ideal lattice J is such that the sum of I and J equals R, meaning that I and J are relatively prime. In the encryption algorithm, a **Samp()** procedure is employed to select a compact vector from a coset of the ideal. This is achieved by shifting the ideal by a specific amount. Therefore, the public key is comprised of $(R, T_I, T^{pk}, Samp())$, and the secret key is

represented by T^{sk}.

5.1.1.2 Encryption

This process involves taking the message vector $\overrightarrow{m}$ and the public key T^{pk} as inputs. The plaintext space K is a subset of R modulo T_I. An additional algorithm, $Samp(\overrightarrow{m}, T_I)$ is used to select a short vector from the coset formed by adding $\overrightarrow{m}$ to I and this result is subsequently reduced modulo the public basis T^{pk}.

$$\begin{aligned} E(\overrightarrow{m}) &= \overrightarrow{m} + \overrightarrow{i} \pmod{T_J^{pk}} \\ &= \overrightarrow{c} \end{aligned}$$

In this scenario, the ciphertext is determined as a vector $\overrightarrow{c}$ and encoded based on distance from the nearest lattice point.

5.1.1.3 Decryption

For decrypting $\overrightarrow{c}$ with the secret key T^{sk} as follows:

$$\begin{aligned} D(\overrightarrow{c}) &= (\overrightarrow{c} \pmod{T_J^{sk}}) \pmod{T_I} \\ &= \overrightarrow{m} \end{aligned}$$

5.1.1.4 Homomorphic Property

The homomorphic characteristic is demonstrated in the evaluation algorithm. This algorithm receives input consisting of a circuit C from a permissible set $C_{\mathbb{E}}$, where the gates in this circuit perform operations modulo T (the public key T_{pk}), and a set of input ciphertexts $\{\overrightarrow{c_1}, \ldots, \overrightarrow{c_m}\}$. It conducts the operations Add_{T_I} and $Mult_{T_I}$ in the correct order to calculate the resulting ciphertext $\overrightarrow{c}$.

$$\begin{aligned} Add(T_J^{pk}, \overrightarrow{c_1}, \overrightarrow{c_2}) &\rightarrow \overrightarrow{c_1} + \overrightarrow{c_2} (mod T_J^{pk}) \\ Mult(T_J^{pk}, \overrightarrow{c_1}, \overrightarrow{c_2}) &\rightarrow \overrightarrow{c_1} \cdot \overrightarrow{c_2} (mod T_J^{pk}) \end{aligned}$$

During the computation of $\overrightarrow{c}$, the algorithm first uses a circuit C that operates under the modulus T_I to process the plaintexts. Then, it replaces the Add_{T_I} and $Mult_{T_I}$

operations within C with the corresponding operations, namely addition and multiplication, but now within the ring R.

5.1.2 Squashing

To streamline the decryption process without sacrificing some of its computational capabilities, Gentry made modifications that involved transferring certain decryption computations to the encryption phase. This was accomplished by including additional information about the secret key within the public key. However, it's important to note that this adjustment inherently reduced the security of the original scheme.

This transformation can be broken down into two stages:

- **Stage 1:** An initial, computationally intensive preprocessing phase carried out by the encrypter does not require the secret key.
- **Stage 2:** A computationally lightweight phase is performed by the decrypter, and it makes use of the secret key.

The revised scheme introduces three new parameters (ι, r, s) and two new algorithms, namely $SplitKey$ and $ExpandCT$, where the $SplitKey$ algorithm puts in the public key a hint about the secret key and the $ExpandCT$ algorithm prepares the ciphertext for the changed decryption algorithm.

5.1.3 Bootstrapping

The scheme capable of evaluating its own decryption algorithm circuit is referred to as $bootstrappable$ [11]. Bootstrapping essentially involves a refreshing process to reduce the noise in a ciphertext after performing homomorphic operations on it. To achieve this, the following steps are taken:

1. A pair of key sets (pk_a, sk_a) and (pk_b, sk_b) are generated.
2. The message m is encrypted under the initial public key pk_a resulting in $c = E_{pk_a}(m)$.

3. c is decrypted under the second public key pk_b specifically as $E_{pk_b}(c) = E_{pk_b}(E_{pk_a}(m))$.

4. The first secret key sk_a is encrypted under the second public key as $E_{pk_b}(sk_a)$.

5. Both $E_{pk_b}(sk_a)$ and $E_{pk_b}(c)$ are transmitted .

Because the squashed SwHE scheme can assess its own decryption circuit, the decryption circuit is applied to homomorphically decrypt the noisy ciphertext. This is achieved using the encryption of the first secret key sk_a under pk_b denoted as $E_{pk_b}(D_{sk_a}(c))$. Consequently

$$E_{pk_b}(D_{sk_a}(c)) = E_{pk_b}(m)$$

which can then be decrypted by the customer using sk_b, resulting in

$$D_{sk_b}(E_{pk_b}(m)) = m$$

5.2 Fully Homomorphic Encryption Scheme over Integers

Gentry's research significantly influenced the alternative cryptographic system proposed by Dijk and his colleagues in 2010, [7] where they replaced the ideal lattice-based approach with an integer-based one. This new scheme is simpler in concept compared to Gentry's ideal lattice-based scheme but maintains similar homomorphic operations and effectiveness characteristics. This is because it is based on the Approximate-Greatest Common Divisor (AGCD) problem, [10], which involves searching for p within the set of equations $x_i = kq_i + r_i$. The outlined symmetric and asymmetric schemes for SwHE look as follows:

5.2.1 Secret Key SwHE scheme based on Integers

5.2.1.1 Key Generation

The secret key is an odd integer chosen from some interval $k \in [2^{\xi-1}, 2^{\xi}]$.

5.2.1.2 Encryption

For encrypting a bit $m \in \{0, 1\}$, the ciphertext as

$$\begin{aligned} E(m) &= m + 2r + kq \\ &= c \end{aligned}$$

where q, and r are chosen randomly, and $2r < k/2$.

5.2.1.3 Decryption

For decrypting the ciphertext c as

$$\begin{aligned} D(c) &= (c \pmod k) \pmod 2 \\ &= (kq + 2r + m \pmod k) \pmod 2 \\ &= 2r + m \pmod 2 \\ &= m \end{aligned}$$

5.2.2 Public-Key SwHE scheme based on Integers

5.2.2.1 Key Generation

Select an odd integer k with ξ bits length as the private key. Utilize this private key to create the public key, represented by the equation:

$$x_i = kq_i + r_i$$

where $q_i \in \mathbb{Z} \cap [0, 2^\varrho/k)$ and $r_i \in \mathbb{Z} \cap (-2^\theta, 2^\theta)$ are random.

5.2.2.2 Encryption

For encrypting a message $m \in \{0, 1\}$, choose a random subset $T \subseteq \{1, 2, \ldots, t\}$ where t represents the count of integers in the public key and the output:

$$\begin{aligned} E(m) &= (m + 2r + 2\sum_{i \in T} x_i) \pmod{x_0} \\ &= c \end{aligned}$$

5.2.2.3 Decryption

For decrypting the ciphertext c with the private key k, the output:

$$\begin{aligned} D(c) &= (c \pmod{k}) \pmod{2} \\ &= (c - k \cdot [c/k]) \pmod{2} \\ &= (c \pmod{2}) \oplus ([c/k]) \pmod{2}) \\ &= m \end{aligned}$$

where $[a]$ represents the rounding to the nearest integer and $k \pmod{2}$ = 1.

5.2.3 Fully Homomorphic Property of SwHE scheme based on Integers

Given two ciphertext $c_a = kq_a + 2r_a + m_a$ and $c_b = kq_b + 2r_b + m_b$, we have

$$c_a + c_b = (q_a + q_b)k + 2(r_a + r_b) + (m_a + m_b)$$

$$c_a \cdot c_b = (kq_aq_b + 2q_ar_b + 2q_br_a + m_aq_b + m_bq_a)k + 2(2r_ar_b + m_ar_b + m_br_a) + m_am_b$$

When

$$r_a + r_b < k/2$$

$$2r_ar_b + m_ar_b + m_br_a < k/2$$

we have

$$\begin{aligned} (c_a + c_b \pmod{k}) \pmod{2} &= m_a + m_b \\ (c_a \cdot c_b \pmod{k}) \pmod{2} &= m_am_b \end{aligned}$$

Therefore, both the symmetric and asymmetric SwHE have a fully homomorphic property.

5.2.4 Squashing

There are introduced three additional parameters denoted as σ, ϱ, and ζ, all of which are functions of parameter λ. $\sigma = \frac{\gamma\xi}{\rho}$ and $\zeta = \omega(\sigma \cdot log\lambda)$ are settled. We add to the public key pk^* from the original SwHE scheme, we add to the public key a set $\{y_1, y_2, \ldots, y_\zeta\}$ of rational numbers in [0, 2) with σ bits of precision, such that there is a sparse subset $S \subset \{1, 2, \ldots, \zeta\}$ of size ϱ with

$$y_1 + y_2 + \cdots + y_\zeta \approx 1/k \pmod 2$$

for secret key sk^* and public key pk^* from the original SwHE scheme.

5.2.4.1 Key Generation

Generate $sk^* = k$ and pk^*. Set $x_k = [2^k/k]$ choose at random a ζ-bit vector
S = $(s_1, s_2, \ldots, s_\zeta)$ with Hamming weight ζ and let S = $i : s_i = 1$. Choose at random integer $u_i \in \mathbb{Z} \cap [0, 2^{\sigma+1})$, $i = \{1, 2, \ldots, \zeta\}$ subject to the condition that

$$\sum_{i \in S} u_i = x_k \pmod{2^{\sigma+1}}$$

. Set $y_i = u_i/2^\sigma$ and $y = \{y_1, y_2, \ldots, y_\zeta\}$. The chosen y_i's confirms

$$\sum_{i \in S} y_i \pmod 2 = (1/k) - \triangle_k$$

for some $|\triangle_k| < 2^{-\sigma}$. The public key is $\{pk, y\}$, and the secret key is S.

5.2.4.2 Encryption

Given a ciphertext c^*, for $i \in \{1, 2, \ldots, \zeta\}$, set

$$z_i = c^* \cdot y_i \pmod 2$$

for each z_i keeping $n = [log\zeta] + 3$ bits of precision after the binary point. The outputs are $c^*$and $z = \{z_1, z_2, \ldots, z_\zeta\}$.

5.2.4.3 Decryption

For decrypting c^* and z with private key k, output as follows:

$$m = (c^* - [\sum_{i \in S} z_i]) \pmod 2$$

5.2.5 Bootstrapping

Bootstrapping is crucial in the FHE framework. Including the bootstrapping component ensures that the decryption circuit is a subset of the permitted circuits.

Definition 5.6. (Augmented Decryption Circuit) . Let ϵ be an encryption scheme where decryption is implemented by a circuit that depends only on the security parameter.

Given a specific security parameter λ, the collection of enhanced decryption circuits comprises two distinct circuits. Each of these circuits takes as input a secret key and two ciphertexts:

- The first circuit decrypts both ciphertexts and performs an addition operation on the resulting plaintext bits modulo 2.
- The second circuit decrypts both ciphertexts and performs a multiplication operation on the resulting plaintext bits modulo 2.

Definition 5.7. (Bootstrappable Encryption) [7] Let ϵ be a HE scheme. We say that ϵ is bootstrappable if its augmented decryption circuits are permitted circuits for every value of the security parameter λ.

To reduce the size of the ciphertext during the evaluation process, van Dijk [7] and his team introduced additional elements to the public key in the form of $x_i^{'}$, which is defined as $q_i^{'}k + 2r_i$. Here, r_i is selected as usual from the range between -2^{Υ} and 2^{Υ}, while q_i values are chosen to be significantly larger than those used for other public key elements. Specifically, for values of i ranging from 0 to ϖ, the expressions are as follows:

$q_i^{'}$ belongs to the set of integers within the range $[2^{\varpi+i-1}/k, 2^{\varpi+i}/k)$, r_i is an integer within the interval $(-2^{\Upsilon}, 2^{\Upsilon})$, and $x_i^{'}$ is calculated as $2(q_i^{'} \cdot k + r_i)$.

This calculation results in $x_i^{'}$ being within the range of values from $[2^{\varpi+i}, 2^{\varpi+i+1}]$.

CHAPTER 6

REAL-WORLD APPLICATIONS OF HOMOMORPHIC ENCRYPTION

In real-life circumstances, HE has great relevance and effect. The following are some instances of its practical applications:

6.1 Secure E-voting Technology

In contemporary society, voting is an essential civic duty. Encryption methods provide a secure avenue for the implementation of computer-based voting systems, ensuring the safety of large-scale elections. The homomorphic technology-driven plan is straightforward, featuring well-defined steps that are practical for real-world application. Even in cases where the electorate's machine has been attacked by harmful programs or the voter is practically under the control of an adversary, a remote end-to-end voting scheme system [34] assures that the voter's choice remains private. It allows voters to transmit honest messages through anonymous and untraceable channels, offering effectiveness, verifiability, and anonymity. Three steps make up the proposed protocol's structure: setup, where parameters are configured, voter registration, core voting, where votes are processed, and tallying, where the results are decrypted [32]. FHE has introduced a groundbreaking solution for creating an electronic voting system that supports addition and multiplication operations. An EMH E-Voting technique has been created to alleviate the shortcomings of the current setup. In order to guarantee an effective electoral process, this solution combines the ElGamal encryption technique with decentralized decryption and makes use of a strong verification

mechanism. Grouped counting and group shuffling techniques are utilized to prevent vote manipulation and maintain count confidentiality.

6.2 Financial Privacy

A company, especially in sectors like finance where sensitive data and proprietary algorithms, such as those employed for stock value predictions, need protection, must take steps to secure this information. Naehrig [25] introduced a technique utilizing HE to securely transmit data and algorithms in an encrypted format, enabling the delegation of computational tasks to a cloud service. However, HE alone doesn't inherently safeguard the secrecy of the algorithm; this is the realm of obfuscation research. The critical feature in fully-fledged HE methods is circuit privacy, ensuring that no information about the function is revealed through the output. Nevertheless, it doesn't directly encrypt the function itself. To illustrate, consider a scenario where Company Y possesses confidential techniques for predicting stock prices, and Company X holds sensitive data like a stock portfolio. In the conventional approach, X would need to share its stock portfolio with Y, or Y would have to disclose its algorithm to X. However, X can encrypt its data using a circuit-private approach and transmit it to Y using HE. Y can then apply its proprietary method to the encrypted data, yielding results that can only be decrypted using X's secret key. This way, Y is barred from accessing X's data, and X remains unaware of the specific processes employed by Y.

6.3 Control System for Protection

A control system, often known as a cyber-physical system, is a computer-based network-connected system designed to regulate and manage signals for the operation of a physical system. It consists of a controller, a plant equipped with sensors and actuators. The controller gathers sensor data, combines it with user input to generate command instructions, and transmits them to actuators for controlling the plant accordingly. This domain encompasses a wide range of applications, including smart vehicles, drones, and critical infrastructure such as nuclear and commercial facilities.

Notably, there have been instances of control system breaches, such as a demonstration in 2015 where a hacker remotely manipulated a car's braking and acceleration systems. Similarly, in 2010, a malicious computer virus infiltrated a uranium enrichment facility's system, altering centrifuge speeds and causing damage. Preventing control system hacking is crucial but challenging. To safeguard against data theft or unauthorized control commands, it is recommended that sensors transmit encrypted data to the controller, and the controller sends encrypted control instructions to the actuators. However, this approach may not fully protect against data leakage caused by malware infiltrating the controller. Recently, some researchers have proposed using HE to secure control systems, specifically by encrypting sensor data with HE [16]. In such a scenario, the controller may not have permission to access sensor data because encryption is unnecessary for data processing. Moreover, it is likely that the actuator's sensing system can detect alterations made by intruders to the encoded information. For added security, this approach could be compared to HAE.

6.4 Healtcare Industry

Individuals' private info in computer-based archive format, which contains data pertaining to each person's health issues, are under the control of health facilities. Data centers are used to store and process these records. [23] Sensitive medical data is lost when data is encrypted using conventional methods since it must first be decrypted in order to do computations. By permitting data to be treated in an encrypted form and ensuring only encrypted data is exposed to service providers, HE arises as a solution to this problem.

6.5 Cloud Computing

In contrast to conventional encryption methods, HE offers the advantage of minimizing the need for frequent encryption and decryption between cloud services and users. This efficiency results in reduced communication and computational costs. HE is a pivotal technology for maintaining data confidentiality in cloud environments. [8]Leveraging its homomorphic capabilities, this technology can address critical se-

curity concerns within cloud services, ultimately facilitating the advancement of HE technology through the broader adoption of cloud computing applications.

CHAPTER 7

CONCLUSION

There are still unsolved security concerns surrounding FHE schemes. Enhancing the computational efficiency of FHE schemes is a significant challenge. Researchers may work on optimizing algorithms, reducing computational overhead, and developing hardware accelerators to make FHE more practical for real-world applications. Furthermore, most FHE schemes rely on Gentry's bootstrapping technique and the outstanding challenge is the development of an unlimited FHE scheme that permits unlimited operations without the need for bootstrapping. Liu [19] introduced a noise-free FHE scheme and Brakerski and Gentry [4] proposed an FHE scheme without bootstrapping. After this study was published, Yagisawa [33] also published a study about an FHE scheme without bootstrapping. However, Wang reported Liu and Yagisawa's studies as insecure. [18]

In conclusion, HE provides a transformative solution at the intersection of cryptography and data privacy. Its capability to conduct computations on encrypted data, without compromising sensitive information, has the potential to revolutionize the way we handle and analyze data, particularly in fields with high privacy concerns. Through exploration of its various types, which include PHE, SwHE, and FHE, a spectrum of capabilities ranging from basic operations to comprehensive computational power has been uncovered. The applications of HE in healthcare, finance, cloud computing, and other fields underscore its relevance and promise in modern society. However, as this technology continues to evolve, challenges related to security, efficiency, and standardization must be addressed to realize its full potential. Ongoing research and innovation suggest that HE will play a pivotal role in securing data while facilitating

advanced data analytics, machine learning, and secure communications in the digital age.

REFERENCES

[1] M. Ajtai. Generating hard instances of lattice problems. *in Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, page 99–108, 1996.

[2] J. Benaloh. Dense probabilistic encryption. *in Proceedings of the workshop on selected areas of cryptography*, page 120–128, 1994.

[3] D. Boneh, E. Goh, and K. Nissim. Evaluating 2-dnf formulas on ciphertexts. *in TCC*, 3378:325–341, 2005.

[4] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. Fully homomorphic encryption without bootstrapping. *Electron. Colloquium Comput. Complex.*, TR11, 2011.

[5] I. Damgård and M. Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. *In: Kim K. (ed) Public Key Cryptography. PKC 2001.LNCS*, page 119–136, 1992.

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

[7] M. V. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. *in Annual International Conference on the Theory and Applications of Cryptographic Techniques*, page 24–43, 2010.

[8] M. Zhao E and Y. Geng. Homomorphic encryption technology for cloud computing. *Procedia Computer Science*, 154:73–83, 2019.

[9] T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on information theory*, 31(4):469–472, 1985.

[10] S. D. Galbraith, S. W. Gebregiyorgis, and Sean Murphy. Algorithms for the approximate common divisor problem. *LMS Journal of Computation and Mathematics*, 19(A):58–72, 2016.

[11] C. Gentry. A fully homomorphic encryption scheme. *Ph.D. thesis*, 2009.

[12] K. Gjøsteen. Subgroup membership problems and public key cryptosystems. 2004.

[13] O. Goldreich, S. Goldwasser, and S. Halevi. Public-key cryptosystems from lattice reduction problems. *in Advances in Cryptology-CRYPTO'97: 17th Annual*

International Cryptology Conference, Santa Barbara, California, USA, August 1997. Proceedings, 60(5):112, 1997.

[14] S. Goldwasser and S. Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. *Symposium on the Theory of Computing*, page 365–377, 1982.

[15] S. Goluch. The development of homomorphic cryptography: from rsa to gentry's privacy homomorphism. *na*, 2011.

[16] J. Kim, C. Lee, H. Shim, J. H. Cheon, A. Kim, M. Kim, and Y. Song. Encrypting controller using fully homomorphic encryption for security of cyber-physical systems. *IFAC*, 49:175–180, 2016.

[17] Ç. K. Koç, F. Özdemir, and Z. Ödemiş Özger. *Partially Homomorphic Encryption*. Springer, 2021.

[18] J. Li and L. Wang. Noise-free symmetric fully homomorphic encryption based on noncommutative rings. *IACR Cryptol. ePrint Arch.*, page 641, 2015.

[19] D. Liu. Practical fully homomorphic encryption without noise reduction. *IACR Cryptol. ePrint Arch.*, page 468, 2015.

[20] C. A. Melchor, P. Gaborit, and J. Herranz. Additively homomorphic encryption with d-operand multiplications. *in CRYPTO*, 6223:138–154, 2010.

[21] D. Micciancio and S. Goldwasser. Complexity of lattice problems - a cryptographic perspective. *Springer Science Business Media*, 671, 2002.

[22] H. Minkowski. Geometrie der zahlen. *Ripol Classic*, 40, 1910.

[23] K. Munjal and R. Bhatia. A systematic review of homomorphic encryption and its contributions in healthcare industry. *Complex Intell Systems.*, pages 1–28, 2022.

[24] D. Naccache and J. Stern. A new public key cryptosystem based on higher residues. *In: Proceedings of the 5th ACM Conference on Computer and Communications Security*, page 59–66, 1998.

[25] M. Naehrig, K. E. Lauter, and V. Vaikuntanathan. Can homomorphic encryption be practical? *In Christian Cachin and Thomas Ristenpart, editors, Proceedings of the 3rd ACM Cloud Computing Security Workshop, CCSW*, page 113–124, 2011.

[26] T. Okamoto and S. Uchiyama. A new public-key cryptosystem as secure as factoring. *Advances in Cryptology—EUROCRYPT'98*, page 308–318, 1998.

[27] P. Paillier. Public key cryptosystems based on composite degree residue classes. *Proceedings of Advances in Cryptology, EUROCRYPT'99*, page 223–238, 1999.

[28] C. Peikert. A decade of lattice cryptography. *Foundations, and Trends in Theoretical Computer Science*, 10(4):283–424, 2016.

[29] J. M. Pollard. Monte carlo methods for index computation (mod p). *Math. Comput.*, 32(143):918–924, 1978.

[30] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21:120–126, 1978.

[31] T. Sander, A. Young, and M. Yung. Non-interactive cryptocomputing for nc1. *in Proceedings of the 40th Annual Symposium on Foundations of Computer Science, FOCS '99*, page 554–, 1999.

[32] S. S. Shinde, S. Shukla, and D. Chitre. Secure e-voting using homomorphic technology. *International Journal of Emerging Technology and Advanced Engineering*, 3:203–206, 2013.

[33] M. Yagisawa. Fully homomorphic encryption without bootstrapping, 2015.

[34] X. Yi and E. Okamoto. Practical internet voting system. *J. Netw. Comput. Appl.*, 36(1):378–387, 2013.

[35] X. Yi, R. Paulet, and E. Bertino. *Homomorphic encryption and applications*, volume 3. Springer, 2014.