
GAZİ ÜNİVERSİTESİ
Gazi
1926

SIMULATION OF A HOMOMORPHIC ENCRYPTION SYSTEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
GAZİ UNIVERSITY

BY

Hanife Çağıl BOZDUMAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL - ELECTRONIC ENGINEERING

DECEMBER 2019

The thesis study titled "THE NAME OF SIMULATION OF A HOMOMORPHIC ENCRYPTION SYSTEM" is submitted by Hanife Çağıl BOZDUMAN in partial fulfillment of the requirements for the degree of Master of Science in the Department of Electrical-Electronic Engineering, Gazi University by the following committee.

Supervisor: Prof. Dr. Erkan AFACAN

Electrical-Electronic Engineering Department, Gazi University

I certify that this thesis is a Master of Science thesis in terms of quality and content

.....

Chairman: Prof. Dr. Erdem YAZGAN

Electrical and Electronics Engineering Department, TED University

I certify that this thesis is a Master of Science thesis in terms of quality and content

.....

Member: Prof. Dr. Fırat HARDALAÇ

Electrical-Electronic Engineering Department, Gazi University

I certify that this thesis is a Master of Science thesis in terms of quality and content

.....

Date: 02/12/2019

I certify that this thesis, accepted by the committee, meets the requirements for being a Master of Science Thesis.

.....

Prof. Dr. Sena YAŞYERLİ

Dean of Graduate School of Natural and Applied Sciences

ETHICAL STATEMENT

I hereby declare that in this thesis study I prepared in accordance with thesis writing rules of Gazi University Graduate School of Natural and Applied Sciences;

- All data, information and documents presented in this thesis have been obtained within the scope of academic rules and ethical conduct,
- All information, documents, assessments and results have been presented in accordance with scientific ethical conduct and moral rules,
- All material used in this thesis that are not original to this work have been fully cited and referenced,
- No change has been made in the data used,
- The work presented in this thesis is original,

or else, I admit all loss of rights to be incurred against me.

Hanife Çağıl BOZDUMAN

02/12/2019

SIMULATION OF A HOMOMORPHIC ENCRYPTION SYSTEM

(M. Sc. Thesis)

Hanife Çağıl BOZDUMAN

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

December 2019

ABSTRACT

Cloud computing is widely used nowadays. Cloud computing is the legal transfer of information services over internet. Cloud computing is a service which allows software and hardware sources to be used from remote locations. The widespread usage of cloud computing services causes many safety questions. The popularity of research about the operations on data without decryption has increased. Here, the cryptology comes to the scene. Cryptology is the science and art of making communication incomprehensible to third parties who have no right to read and understand the data or message. Cryptology has two subparts: cryptography which analyzes methods of encrypting messages, and cryptanalysis that analyzes methods of decrypting encrypted messages. Encryption is turning a plain text data to a random and meaningless text. Decryption is the vice versa. It is the process of turning this random and meaningless text to original plaintext. In homomorphic encryption, operations are realized directly on the encrypted version of the data. The results will be the same with decrypted data even operations are done with encrypted data. In this thesis, the simulation of a homomorphic encryption system is actualized.

Science Code	:	92706
Key Words	:	Encryption; cryptology, cloud computing, homomorphic encryption, secure communication
Page Number	:	55
Supervisor	:	Prof. Dr. Erkan AFACAN

HOMOMORFİK BİR ŞİFRELEME SİSTEMİNİN SİMÜLASYONU

(Yüksek Lisans Tezi)

Hanife Çağıl BOZDUMAN

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Aralık 2019

ÖZET

Günümüzde bulut bilişim çok yaygın şekilde kullanılmaktadır. Bulut bilişim, bilişim hizmetlerinin internet üzerinden yasal olarak aktarılmasıdır. Bulut hizmetleri, uzak konumlardaki yazılım ve donanım kaynaklarının kullanılmasını sağlar. Bulut bilişimin bu kadar yaygın kullanılması, bu konudaki güvenlik açıkları ile ilgili bir sürü soruya yol açmaktadır. Veriler üzerinde deşifreleme olmaksızın işlem yapıp yapmamanın mümkün olup olmadığı hakkında araştırmalar hız kazanmıştır. Burada devreye kriptoloji girecektir. Kriptoloji, veriyi veya mesajı, üçüncü taraflarca anlaşılamayacak biçimde şifreleme bilimi ve sanatıdır. Kriptoloji, mesajları şifreleme yöntemlerini analiz eden "şifreleme" ve şifreli mesajları deşifreleme yöntemlerini analiz eden "deşifreleme" yöntemlerinden oluşur. Şifreleme, bir düz metnin verilerinin rasgele ve anlamsız görünen bir metne dönüştürülmesi işlemidir. Deşifreleme ise şifre çözme işidir. Diğer bir deyişle rasgele metni düz metne dönüştürme işlemidir. Homomorfik şifreleme, şifrelenmiş veriler üzerinde doğrudan hesaplama yapılmasına izin verir. Şifre çözülürse, sonucun, işlemin şifresiz yapılması durumunda elde edilecek olan değerle aynı olduğu görülecektir. Bu tezde bir homomorfik şifreleme sisteminin simülasyonu gerçeklenmiştir.

Bilim Kodu	:	92706
Anahtar Kelimeler	:	Şifreleme, kriptoloji, bulut bilişim, homomorfik şifreleme, güvenli iletişim
Sayfa Adedi	:	55
Danışman	:	Prof. Dr. Erkan AFACAN

ACKNOWLEDGEMENT

I would like to thank my supervisor Prof. Dr Erkan AFACAN for his trustworthy guidance, precious support for my thesis. After that, i want to say thank you to my parents Mustafa-Beyhan BOZDUMAN and my brother Çağrı BOZDUMAN for their timeless encouragement.

TABLE OF CONTENTS

Page

ABSTRACT iv

ÖZET v

ACKNOWLEDGEMENT vi

TABLE OF CONTENTS vii

LIST OF FIGURES x

SYMBOLS AND ABBREVIATIONS xii

1. INTRODUCTION 1

2. CRYPTOGRAPHY AND CRYPTANALYSIS 3

2.1. Classical Cryptography 4

2.1.1. Classical cryptographic techniques 5

2.2. Modern Cryptography 5

2.2.1. Modern cryptographic techniques 7

3. HOMOMORPHIC ENCRYTION 9

3.1. Homomorphic Encryption Description 9

3.2. Homomorphic Encryption Types 12

3.2.1. Additive homomorphic encryption 13

3.2.2. Multiplicative homomorphic encryption 13

4. GENTRY'S SCHEME 19

4.1 Encryption Process 21

4.2. Decryption 24

4.3. An Optimized Decryption Procedure 25

4.4. Basics for Homomorphic Encryption 26

4.5. Learning with Error problem (LWE) 27

4.6 From Somewhat Homomorphic Encryption to Fully Homomorphic Encryption 34

5. SIMULATION OF A HOMOMORPHIC ENCRYPTION SYSTEM 37

Page

5.1. Lattice Based Cryptography (Somewhat Homomorphic Encryption)..... 37

5.2. The Construction..... 39

5.3. Correctness..... 40

5.4. Somewhat Homomorphic Encryption Simulation Study..... 42

5.4.1. Simulation environment..... 42

5.4.2. Simulation results..... 42

5.4.3. Result 43

5.5. Another Simulation System 44

5.6. Final Say for Homomorphic Encryption Simulation Systems..... 47

6. CONCLUSION 49

REFERENCES 51

CURRICULUM VITAE..... 55

LIST OF FIGURES

Figure **Page**

Figure 2.1. Cryptology is divided in two parts 3

Figure 2.2. Types of cryptography..... 4

Figure 2.3. Symmetric cryptography 5

Figure 2.4. Asymmetric key cryptography 5

Figure 2.5. Process of ciphering in public key cryptography 6

Figure 2.6. Ciphertext-only attack 7

Figure 2.7. Known plaintext attack..... 8

Figure 2.8. Chosen plaintext attack..... 8

Figure 3.1. Homomorphic encryption..... 9

Figure 3.2. Group homomorphism..... 10

Figure 3.3. Homomorphic properties of well-known PHE schemes 13

Figure 3.4. A scheme for homomorphic encryption 15

Figure 3.4. How HE works 15

Figure 3.5. A simple circuit approach for understanding FHE..... 17

Figure 4.1. The evolution of encryption 19

Figure 4.2. Lattice points 26

Figure 4.3. Closest vector point..... 27

Figure 4.4. CVP Euclidean 28

Figure 4.5. CVP-light..... 29

Figure 4.6. CVP-hard..... 29

Figure 4.7. Improvement..... 31

Figure 4.8. Addition homomorphism 33

Figure 4.9. Addition homomorphism 33

Figure 4.10. SHE to FHE..... 36

Figure 5.1. The basic concept of encryption scheme..... 38

Figure **Page**

Figure 5.2. Plaintext length and encryption time diagram 42

Figure 5.3. Plaintext length and cipher text length diagram (bit) 43

Figure 5.4. The relationship between number of variables and largest supported degree 44

Figure 5.5. The relationship between bit length coefficient and largest supported degree 45

Figure 5.6. Cells contain the LSD 46

Figure 5.7. Relationship between c and t values 46

Figure 5.8. Steps for simulation systems which uses HE 47

SYMBOLS AND ABBREVIATIONS

Symbols and abbreviations which are used in this research can be found in following lines with their explanations.

Symbols	**Explanation**
mb	Megabyte
tb	Terabyte

Abbreviations	**Explanation**
CVP	Closest Vector Point
DH	Diffie-Helman
FHE	Fully Homomorphic Encryption
GM	Goldwasser–Micali encryption
LWE	Learning with Error
PHE	Partially Homomorphic Encryption
RSA	Rivest-Shamir-Adleman cryptosystem
SHE	Somewhat Homomorphic Encryption
SSSP	Sparse Subset Sum Problem
SVP	Shortest Vector Point

1. INTRODUCTION

The definition of cryptology is making communication meaningless to third parties who have no right to read and understand the data/message. It is the study of secret codes or ciphers; also the devices which can be serviced to create and decipher them. It is benefited in the national security, protection of electronic monetary transactions, telecommunications, military, prevention of destruction of computer data, etc.

Cryptology is separated to cryptography which analyzes methods of encrypting messages, and cryptanalysis which analyzes methods of decrypting encrypted messages. In other words, Cryptology is the application of mathematics, that build cryptography and cryptanalysis.

We live in a century where most individuals all over the world own more than one digital device with limited local storage. It is encouraged people to use cloud computing. By Cloud services ; people can use their applications etc. at distant district.

However, some personal data needs to be kept as a secret. Some cloud services allow people to securely upload private data that is encrypted, but in many cases, the cloud service can decrypt the data. So, this causes the question whether it is possible to process data without giving access to it.

Today’s encryption systems can not be used for any operations without decryption. However, homomorphic encryption make computations on encrypted data without deciphering. It means that The service provider computes on this ciphered data, without noticing what is really inside.. But, it will be seen that the result will be the same as the result which has operated with deciphered data, intended value.

Encryption is a method used for deciphering data with the aim to ensure confidentiality so that only authorized parties can access the information. The types of encryption schemes are either symmetric or asymmetric.

In the symmetric type, the key for ciphering and deciphering is the same. and in the asymmetric setting, there exists a common aka public key also a private key for each party

where this public key is for ciphering and the private key can be used to decipher. Further information could be found at a later section of this thesis.

Gentry's FHE scheme is an asymmetric encryption scheme based on ideal lattices. Essentially one generates a secret key and then a number of public keys, each containing "noise" [1].

The problem is that noise is getting grown with each additional computation. Encryption schemes with this property are called somewhat homomorphic encryption (SHE) schemes. The term somewhat stresses that the number of homomorphic operations one can perform is limited. This noise increase make impossible to decipher the encrypted data at a certain point due to the fact that noise is too big.

In this thesis, bootstrapping is also examined. It is a wat to translate somewhat homomorphic encryption to fully homomorphic encryption. Unfortunately, the bootstrapping procedure is too theoretical and therefore not very efficient.

In this research, homomorphic encryption systems are examined. The aim of this thesis is to get a deep understanding of HE schemes and the theoretical knowledge of how to build a secure HE schemes. This is succeeded by performing an extensive study and challenges when implementing HE.

The challenges include studying and translating the scheme into code, choosing parameters, designing algorithms, and running experiments on them. Gentry's scheme is studied and researched very carefully. The ambition of the thesis is that is being another step to the design more practical HE schemes in the future.

2. CRYPTOGRAPHY AND CRYPTANALYSIS

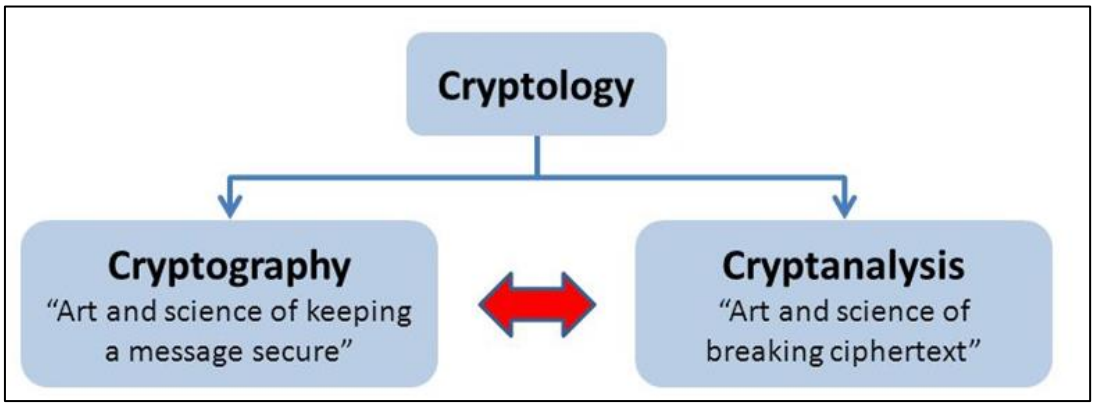

Figure 2.1. Cryptology is divided in two parts [2]

Cryptology is a science that helps us to hide information. It is divided to parts. First part is cryptography and the second part is cryptanalysis.

Cryptography is the discipline of hiding information. A cryptographic mechanism computes the data with collaboration using specified key. This key can be a phrase or a group of letters. If one has to call this encrypted data "trustworthy": the strong hidden side of the key and the mathematical algorithm is important.

Cryptography is used to provide confidentiality, authentication and integrity in the systems. This terms can be explained like below:

Confidentiality, when someone is transmitting data, the transmitted messages could not be eavesdropped by third parties. Or stored data can not be reached by people who has no authority.

Authentication, this is the sign of message comes from a specific party. The receiver demands to see a signature of that.

Integrity, this property proves that no changes have been made by third parties.

There are two types of cryptography. If the key which is used for ciphering and deciphering is the same, the system is called as Symmetric Key Cryptography or Classical Cryptography. It means to give a different key to people that is wanted to correspond.

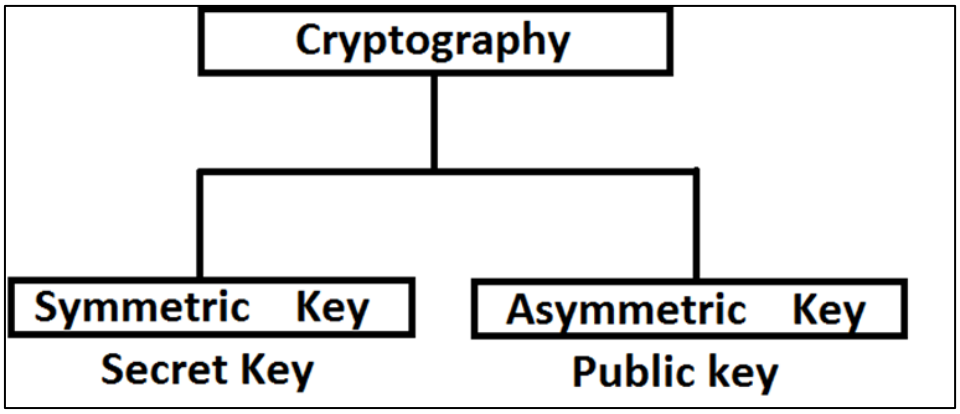

Figure 2.2. Types of cryptography [3]

However, if there are two keys for ciphering, and deciphering; it is called Asymmetric Key Cryptography or Modern Cryptography. The encryption key is can be reached by anyone, as the decryption key remains private in this type.

Asymmetric systems are further useful than symmetric ones. Due to the fact that; the sender and the receiver do not compromise on anything. However, asymmetric schemes, have a flaw: they need more mathematical operations. It means they are more slow than the symmetric ones. As an example, we can thought, RSA and ElGamal.

2.1. Classical Cryptography

Classical cryptography is also known as secret key cryptography. This system has only one key. This key is for both ciphering and deciphering. In Figure 2.3, a key is used to cipher the plaintext. After that; he/she sends this ciphered text to the intended person.

If intended person wants to decrypt the message, he/she must apply the same key to this ciphered message. Thus, the receiver uses this key to decipher the message and has the cleartext.

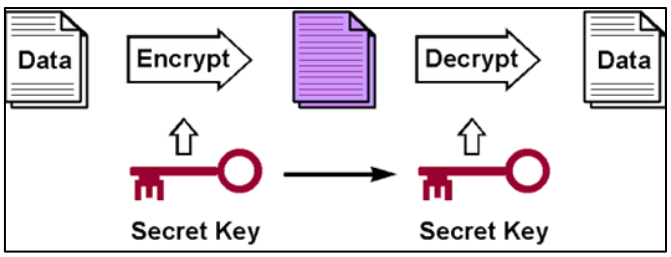

Figure 2.3. Symmetric cryptography [4]

Some of the classical cryptography systems can be found below.

2.1.1. Classical cryptographic techniques

Classical Cryptography have two basic elements of ciphers, one is substitution and the other one is transposition. In substitution ciphers; alphabetical characters are changed by other alphabetical characters. In transposition ciphers, the alphabetical characters are formed in a different order.

These ciphers could be monoalphabetic or polyalphabetic. In monoalphabetic ciphers only one substitution/ transposition is used.

However; in polyalphabetic as Vigenere cipher ; several substitutions/ transpositions are used. Multiple – Letter Encryption and play fair encryption can also be considered as another examples.

2.2. Modern Cryptography

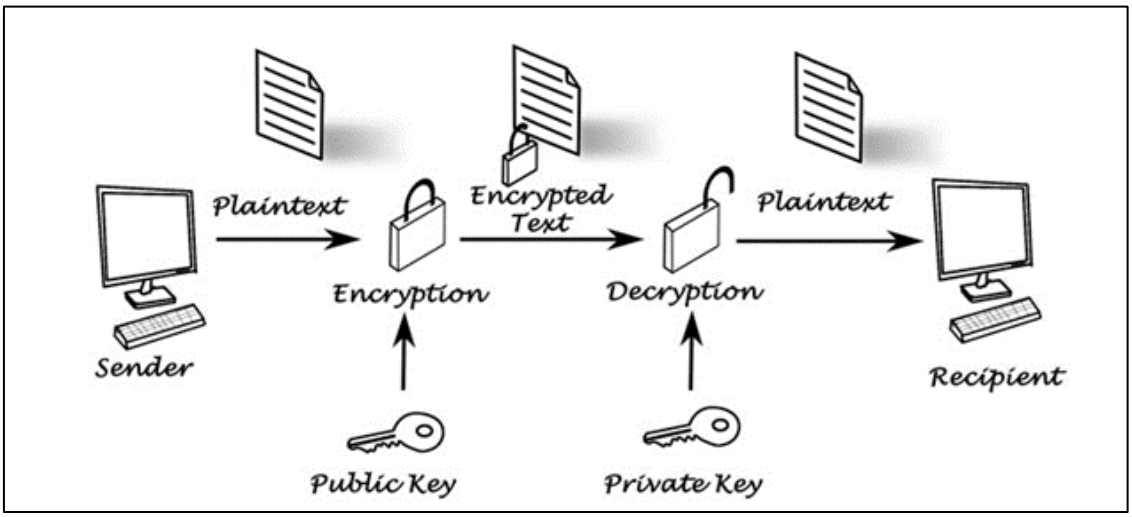

Figure 2.4. Asymmetric key cryptography [5]

Modern cryptography includes the a key pair: one is private key and the other one is public key. For ciphering and deciphering, they are needed. Modern Cryptography can also be heard as public key cryptography. The private key here can not be shared. The key owner has to protect it not to lose or ruin. Figure 2.4. describes the Public Key Cryptography [6]. Public key cryptography aims that those keys are reachable to anyone.

The encryption and decryption process can be seen in following lines.

- Let’s say there are two people, who wants to communicate each other secretly: Charles and Lilian.
- Charles encrypts the data by using his private key and Lilian’s public key. Because, anyone knows Lilian’s public key. However, Lilian will be the only person who knows her private key.
- Charles transfers the ciphered data.
- Lilian takes the data and deciphers it by using her private key and Charles’s public key.

By doing this, she will have the desired data without interruption of third parties.
Figure 2.5 can be read to understand the above process.

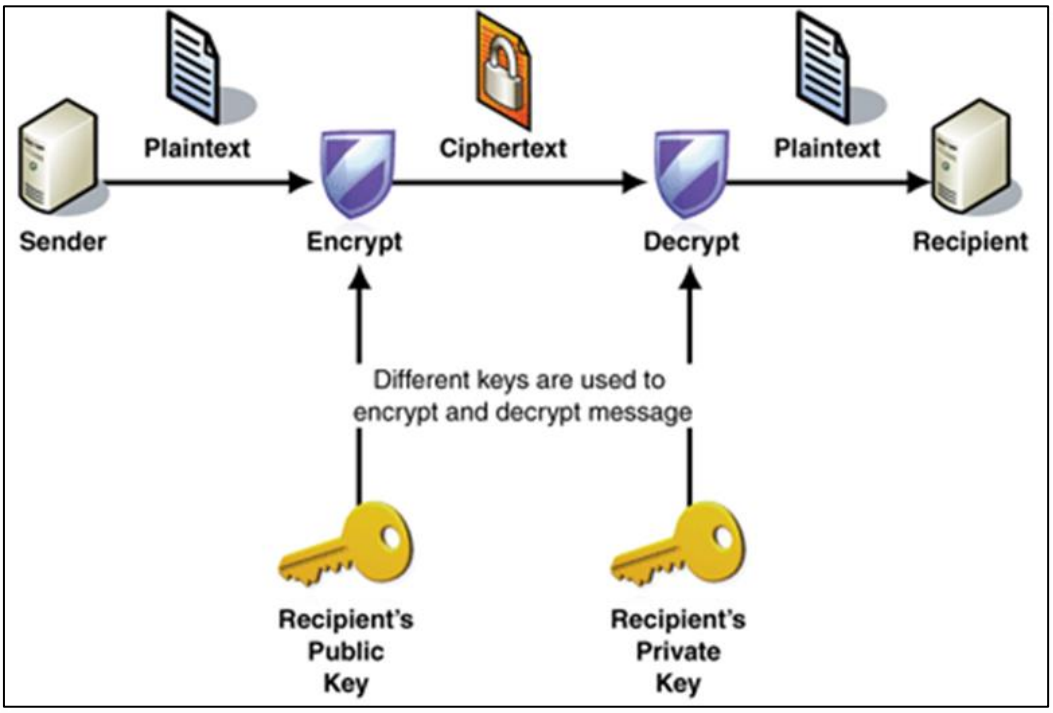

Figure 2.5. Process of ciphering in public key cryptography [7]

2.2.1. Modern cryptographic techniques

Modern cryptographic systems can be examined in by examples. RSA Ciphering Systems by Rivest, Shamir and Adleman. The logic behind it is using a key pair which are public and private. To apply this, the person uses a RSA SecurID security token. This token creates a public key that is not stabiled for any minute. This string provides a hybrid one-time use

password with collaboration of user's private key. Diffie–Hellman key exchange (DH) was created by Whitfield Diffie and Martin Hellman. This method is transferring cryptographic keys safely in a public channel.

Even the method is called as Diffie-Hellman; Ralph Merkle was the first one who designed the concept.

ElGamal Ciphering System was created by Taher Elgamal at 1985. It is a modern encryption method whose roots ground Diffie–Hellman key exchange. The difference is including one extra layer by keys previously used for symmetric message encryption.

Cryptanalysis is the art of decoding a cyphered message without using a key. In other words; Cryptanalysis is the reverse process of cryptography. The objective of cryptanalysis is to decrypt cipher text.

There are various types of attacks that a cryptanalyst can use for deciphering. Most-known ones are they include, ciphertext-only attack, known plaintext attack, chosen plaintext attack.

Ciphertext-only attack, the aim of this attack is finding the deciphering key. It is also acceptable to find an algorithm that converts encrypted message to a decrypted format.

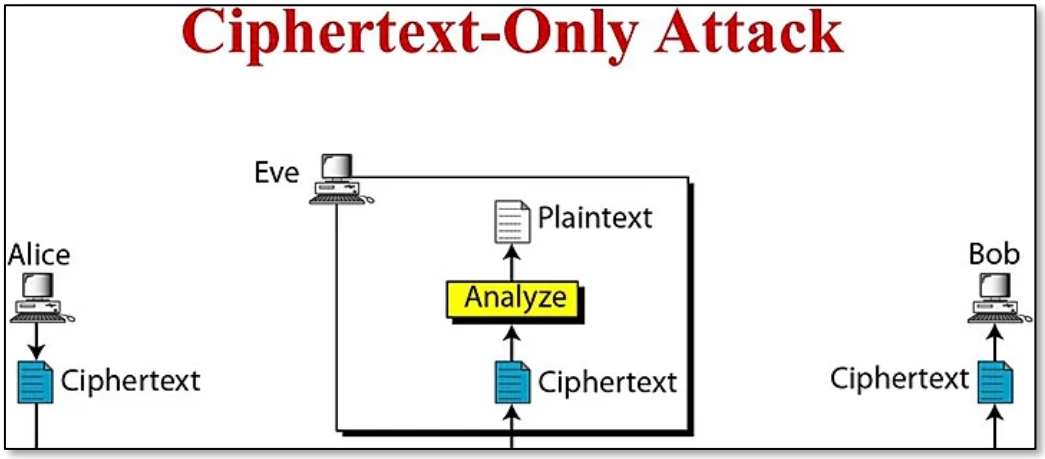

Figure 2.6. Ciphertext-only attack [8]

Known plaintext attack, if the cryptanalyst has an information of specified parts of the text and also has an access to ciphertext, it is easier to find what the hidden information. Because the know text can be used to find for remaining part.

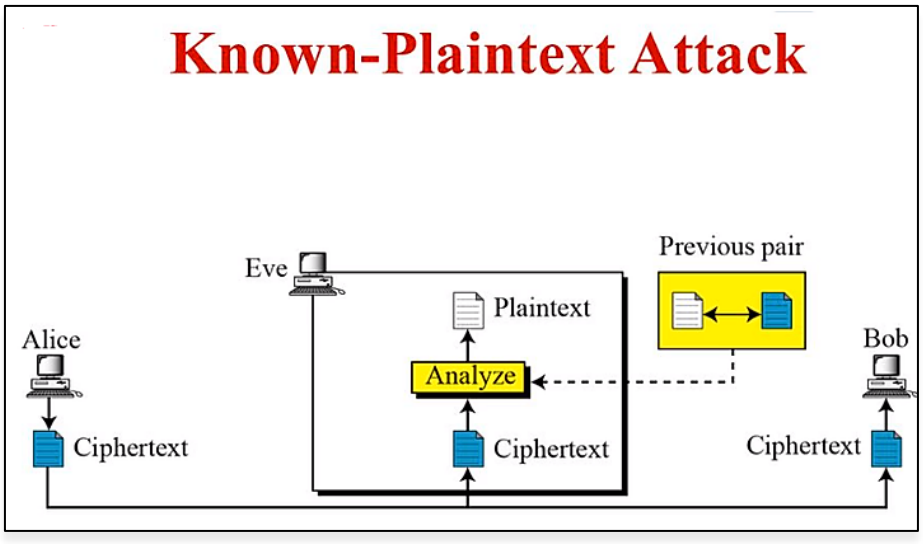

Figure 2.7. Known plaintext attack [9]

Chosen plaintext attack, the cryptanalyst can reach the specified part of plaintext. So, it is easier to solve the encrypted message.

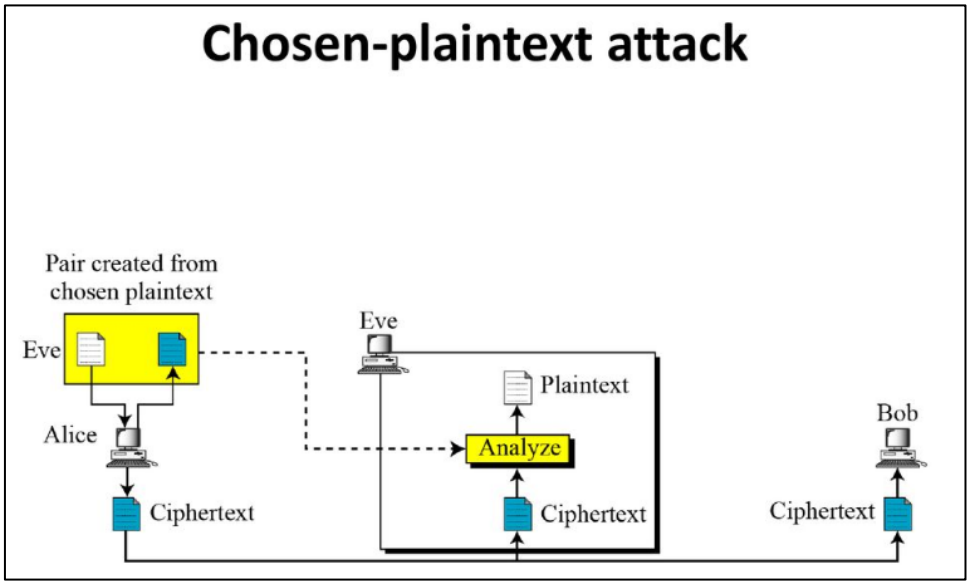

Figure 2.8. Chosen plaintext attack [10]

A cryptosystem means encryption and decryption. Key generation is one of the main steps of key generation, encryption, decryption. Encryption is the system which converts plain text to the ciphertext.

The aim of encryption systems is to provide a strong ciphering which can not be decipher without a key [11]. Decryption is the computation of translating encrypted text back to cleartext.

3. HOMOMORPHIC ENCRYTION

Homomorphic encryption is an encryption which does not need decrypted data to make computations. One can find the same results when he/she make some mathematical operations even with decrypted data or ciphered data [12].

Homomorphic encryption can be applied in sensitive data transferring areas. For example, for the tax transmission , the currency exchange rate, shipping. All in all, it provides an unencrypted data without exposing any information.

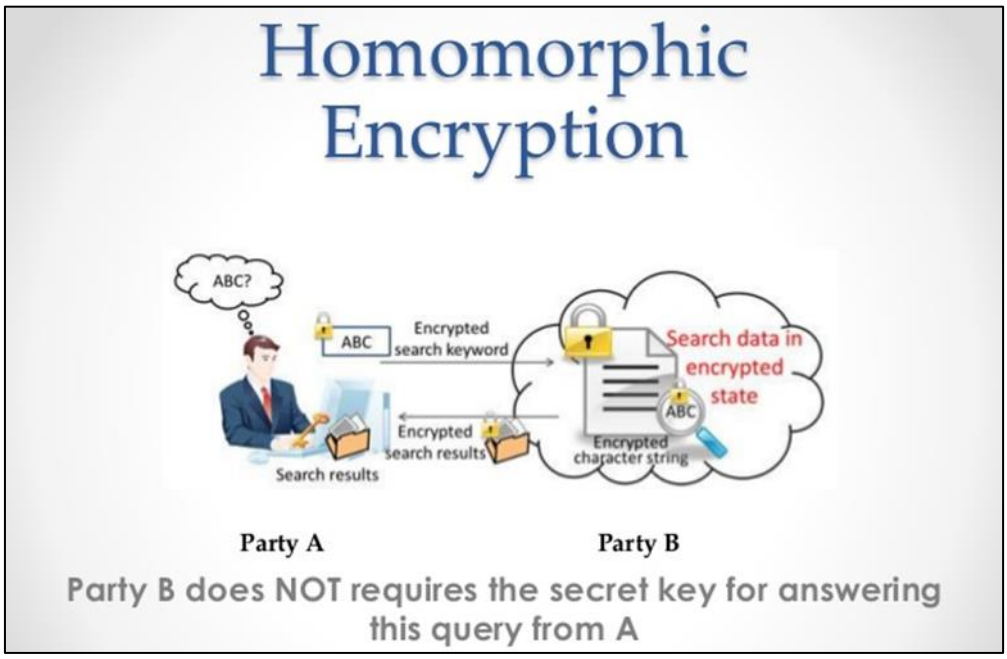

Figure 3.1. Homomorphic encryption [13]

3.1. Homomorphic Encryption Description

Homomorphism can be thought as a relationship between two algebraic groups. Set G, has with a computation that takes any two elements for processing a * b. (G; *), must satisfy four group axioms:

- Closure, The result of the operation of a * b, must be in G.
- Associativity, (a * b) * c = a * (b * c).
- Identity element If e * a = a * e = a ; the identity element is “e”
- Inverse element: a * b = b * a = e. This is the “inverse element”[14].

The result can change because of the order of the signs. Namely, the result of incorporating a and b does not need to give the similar result as b with a. It means that the equation a * b = b * a does not have to be true. But, a + b = b + a is always same due to the fact that the commutativity rule for addition; equation 3.1 can be examined.

$$f(g \Delta g') = f(g) * f(g') \tag{3.1}$$

Group homomorphism can be examined at Figure 3.1.

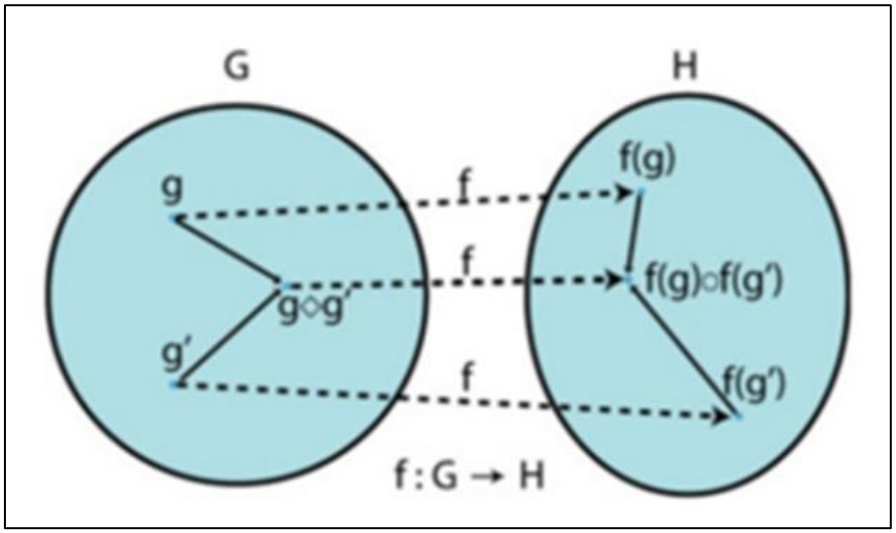

Figure 3.2. Group homomorphism [15]

To sum up, it can be said like ;

Suppose that there are 2 groups called G and H. Let's say * operation is applicable for G; ◊ operation is for H. Let's say x and y are elements of G. If it is defined a function f; goes from G to H. It can be seen what homomorphism is.

- x→f(x) x is mapped to the f(x)
- y→f(y) y is mapped to the f(y)
- z→f(z) z is mapped to the f(z)
- x, y ∈ G x * y = z (3.2)
- f: G→H
- x →f(x)
- y→ f(y)
- z→ f(z)

- $x * y = z \rightarrow f(x) \lozenge f(y) = f(z)$, (3.3)
- $f(x) \lozenge f(y) = f(x*y)$ (3.4)

Equation 3.4 is the simplest explanation of homomorphism. Another example is stated below:

- $G \rightarrow R$, under + , abelian, identity=0
- $H \rightarrow R^+$ under x , abelian, identity=1
- $f: G \rightarrow H$
- $x \rightarrow e^x$
- $f(x + y) \rightarrow f(x) \times f(y)$
- $e^{x+y} = e^x \times e^y$

This mathematical relation is also used for encryption. As it is mentioned before, a cryptosystem has a plaintext, ciphered text , encryption and decryption function.

Also; one key if it is a symmetric systems or two keys also known as private and public keys if it is an asymmetric system. The following lines could outline of this better:

- P→is the text which is not ciphered. P is under ◇operation.
- C→ is the ciphered space. C is under * operation.
- E→is the encryption function
- D→is the decryption function
- K→ is the key. It could be a public or secret key.
- $E_k : P \longrightarrow C, \ (k \varepsilon K)$
- An encryption scheme is homomorphic with respect to an operation ◇on P if the following conditions are satisfied.
- $Decrypt(Encrypt(m_1) * Encrypt(m_2)) = Decrypt(Encrypt(m_1 \diamond m_2)) = m_1 \diamond m_2$ for operation $*$ on C.

As a brief; for all a and b, if $E_k(a) \circ E_k(b) = E_k(a \Delta b)$; this system can be called as homomorphic [16].

3.2. Homomorphic Encryption Types

Homomorphic encryption can be separated as following:

- Partially homomorphic encryption (PHE)
- Fully homomorphic encryption (FHE)

A fully homomorphic encryption scheme maintains a ring-structure. This means that a Ring (R,+,*), where R are our bits on which we operate. (R,+) is an abelian group, while (R,*) is an monoid.

In fully homomorphic systems; it's possible to create NAND Gates by addition and multiplication over bits. If a person has NAND gates, it is possible to create every other Boolean gate. Furthermore; every computation on the encrypted data is possible by using NAND gates.

However; partially homomorphic encryption schemes only supports one operation because of one cannot able to create NAND gates. It means it is not possible to make every computation on the encrypted data. Generally; it is recommended to use partially homomorphic encryption between 1- 256 byte and fully homomorphic encryption between 4 MB- 73TB [17].

In partially homomorphic encryption, mathematical computations does not need encrypted values to be revealed or decrypted in the process. The fragile data remains as a secret in the database tasked while operating on it. Partially homomorphic encryption schemes can success addition or multiplication. However, one cannot apply both.

As an example for partially homomorphic encryption, The Paillier cryptosystem can be given. Two ciphered values can be the subject of a series of Paillier mathematical operations and the ciphered result can be decrypted to reveal the sum of the original plaintext values. The famous encryption standard RSA is partially homomorphic for multiplication. In this case a series of RSA mathematical operations can be conducted to reach an encrypted result that can be decrypted to declare the product of the original plaintext values [18].

As it is mentioned before, multiplication or addition cannot be performed at the same time.

So, some examples for the both parts are examined below.

	Homomorphic Operation	
Scheme	**Add**	**Mult**
RSA [Rivest et al. 1978b]		✓
GM [Goldwasser and Micali 1982]	✓	
El-Gamal [ElGamal 1985][4]		✓
Benaloh [Benaloh 1994]	✓	
NS [Naccache and Stern 1998]	✓	
OU [Okamoto and Uchiyama 1998]	✓	
Paillier [Paillier 1999]	✓	
DJ [Damgård and Jurik 2001]	✓	
KTX [Kawachi et al. 2007]	✓	
Galbraith [Galbraith 2002]	✓	

Figure 3.3. Homomorphic properties of well-known PHE schemes

3.2.1. Additive homomorphic encryption

In this type of homomorphic encryption only addition can be performed. The most well-known structures are ; The Goldwasser–Micali (GM) encryption, The Okamoto–Uchiyama cryptosystem, Paillier encryption.

- Goldwasser-Micali cryptosystem is a public-key encryption algorithm. It was created at 1982. The creators are Goldwasser and Micali. It is a public-key ciphering system which is the first probabilistic assumptions. It has proof to being secure under standard cryptographic assumptions. Unfortunately, encrypted texts are much bigger than the plaintext. This makes GM inefficient [19].
- Okamoto-Uchiyama cryptosystem was created at 1998 by Okamoto and Uchiyama.
- Paillier cryptosystem was created at 1999 by Pascal Paillier [20]. It has a probabilistic scheme.

3.2.2. Multiplicative homomorphic encryption

In this type of homomorphic encryption only multiplication can be performed. The most well-known structures are Rivest-Shamir-Adleman (RSA) cryptosystem, El-Gamal cryptosystem. RSA cryptosystem, RSA algorithm is a public key cryptography type. RSA takes it power from the difficulty of factorizing a multi digit number. The public key consists

of two number. One of these number is the multiplication of two multi digit prime. Key creation steps are following;

- Let us call those prime numbers as a, b.
- After that a*b = n has to be find. It has to be noted that the more the n, the better the security of the code. It should be minimum 512 bits.
- Later, derived number e has to be find. Derived number e must be greater than 1. Also, it should not be greater than $(a - 1)(b - 1)$.
- The important thing which should not be forgotten is this: it should not be any common factor between e and $(a - 1)(b - 1)$ except for 1. It means that, e and $(a - 1)(b - 1)$ has to be co-primes.
- The public key will be (n, e). The strength of RSA comes from the difficulty of factorizing a prime number to obtain a and b to reveal n.
- The private key can be called as d. It is also found from a, b, and e. There is an unequaled number d for those variables.
- The inverse of e modulo of the (b-1).(a-1) is d.
- $(b-1).(a-1) \bmod (e^{-1}) = d$
- $ed = 1 \bmod (a - 1)(b - 1)$

El-Gamal cryptosystem, Diffie Helman secret key change is the base of the ElGamal encryption scheme [21]. The El-Gamal ciphering scheme is valid for cyclic group G. The ElGamal ciphering system has three components:

- the key creation,
- the ciphering
- the deciphering.

Fully homomorphic cryptosystems is a system that provides random computation on ciphertexts. Fully homomorphic encryption (FHE) is more firm and strong than partial homomorphic encryption.

This type of encryption gives the opportunity to construct the systems for any wanted applicability, which can be operated on ciphered inputs to generate an encryption. Homomorphic programs does not need to deciphered.

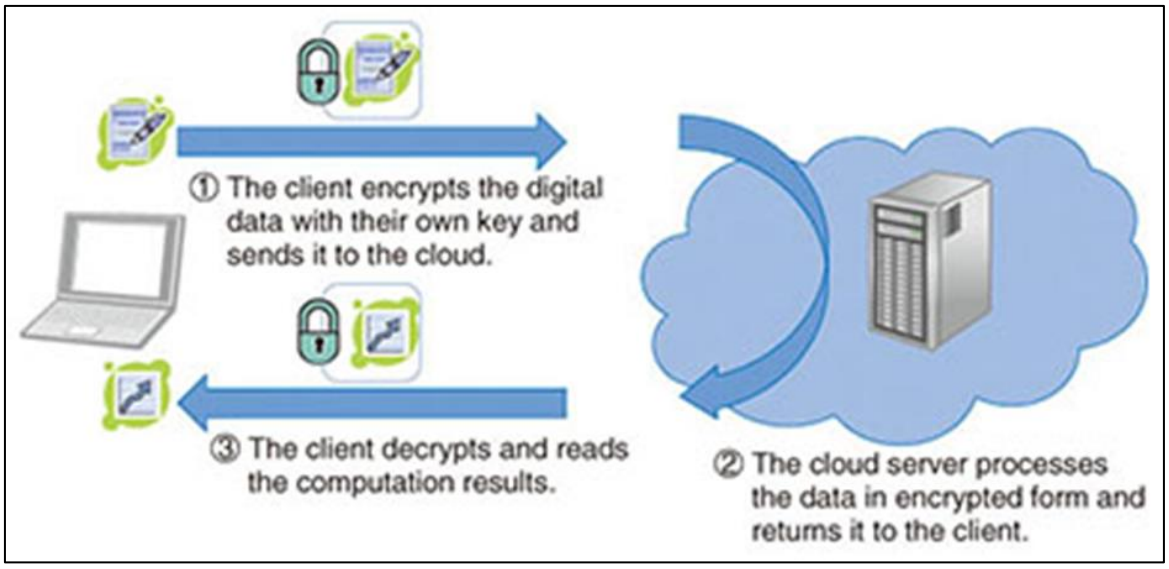

Figure 3.4. A scheme for homomorphic encryption [22]

Even homomorphic programs does not need to be deciphered, it may be operated by a third person without showing its inner info [23]. The cloud can compute mathematical operations on benefit of the user by using FHE. As it is aforementioned, FHE allows for arbitrary mathematical operations on ciphered info. Mathematical operations on ciphered input means that:

Let's say there is a person who has the mathematical operation f and some input. Let's call this inputs are $x_1, \ldots, x_n$. Assume that the $c_1, \ldots, c_n$ are the ciphered form of $x_1, \ldots, x_n$. If this person desire to have $f(x_1, \ldots, x_n)$, he or she can make operations on ciphered data of inputs.

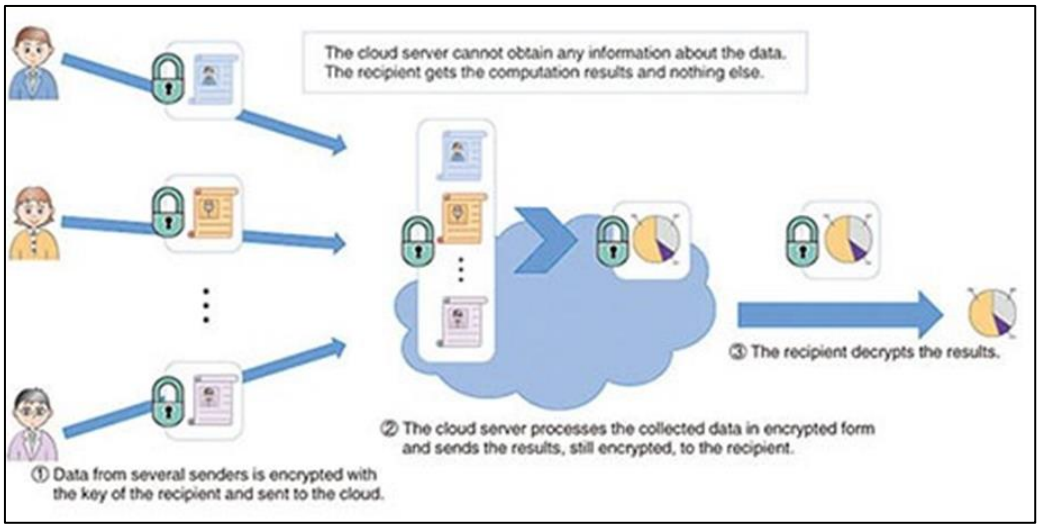

Figure 3.5. How HE works [24]

In some cryptosystem's plaintexts depend on some algebraic systems, as a group. Under this circumstances the encrypted texts depend on some relevant subsystems, that can be equivalent so can plaintexts.

If the plaintext is a group(let us say A), then the ciphertext space is the product A×A. So function f has boundaries for operations in A. The goal of FHE is increasing the boundaries of function f; for being an any function.

This goal can be succeed if and only if the system is homomorphic by looking at the functionally of total set of mathematical operations. Also, it has to be accomplishable to reiterate equations.

Encryption schemes are important in a theoretical base. Still; the security parameter, practical efficiency was not the first thought in gaining the basic fully homomorphic schemes. Unfortunately, plaintext space has a single bit. And they are only homomorphic due to multiplication or addition with respect to modulo 2 [25].

To make a better and efficient system, some variable of fully homomorphic encryption schemes set boundaries to f in different ways. Even though; a mathematical view of fully homomorphic encryption minds reaching the maximum point of f.

Taking this selection as big as it is required is important. This boundary selection can also help us to select a better scheme not only for the plaintext but also for the encrypted text if we thought binary case.

As a summary, it can be said that if a system has a homomorphic encryption property; the result of any addition or multiplication operation in the encrypted data will be the same as the result of any addition or multiplication operation of unencrypted data of the same system. In Figure 3.5. it is possible to see a very simple homomorphic system basic.

Let's say a "hello world" message is sent in a non-homomorphic system. Receiver will take this message. If the system is homomorphic, it means that the encrypted message will have no meaning for third parties. So, even the cloud server take our message and run any operations, it does not mean anything. However, the result will be the same with the non-homomorphic system.

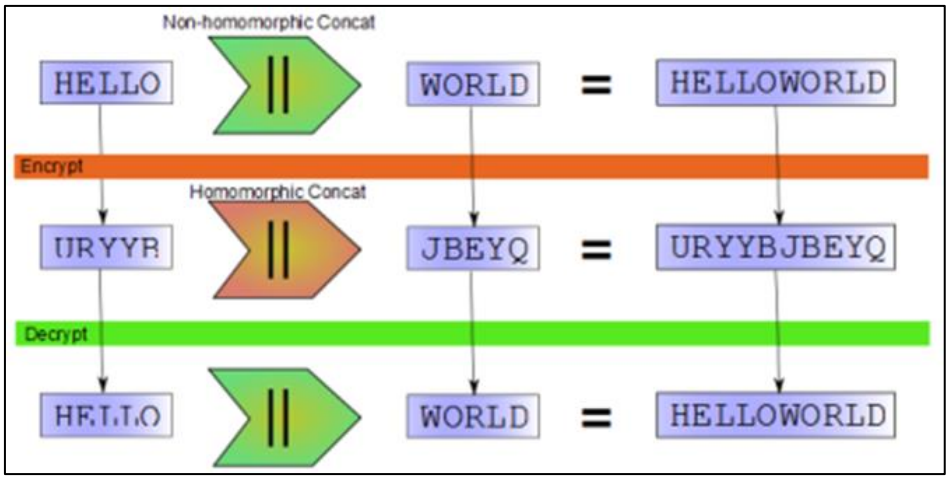

Figure 3.6. A simple circuit approach for understanding FHE [26]

Very famous example of this can help to comprehend the concept:

Maybe; processing any information without access may be thought unbelievable. Think there is a woman who has a jewelry shop (Alice). She needs her employees to collect raw valuable mines. For example gold, diamond, silver etc.

As the gold or the other raw materials are very expensive, she can be afraid of robbery. In other words, she wants her workers to make operations to this valuable mines, without any access. To protect herself from robbery, she is building transparent glove boxes. The keys of these boxes known only by her. So, she puts gold inside of the box.

An employee also controls the inside of the box by using the gloves. However, the worker cannot get to the precious materials, since the box is impenetrable.

Furthermore, this employee can put other stuff to the box. For example soldering iron to use. But, he can not take anything from the box. Because of the boxes are transparent, employee may witness what he does.

In this story, ciphering implies that workers are not able to remove anything out of the box. But it does not mean they are unable to see it. After the job is done, Alice can take the completed product by using her key. This story is not enough in the sense that the glove box may turn out to be quite be in mess, while in the FHE scheme only the result needs to be remained. Of course, this is just a story to make whole system understandable. Still, it could be very complicated to be understood.

All qualifications of homomorphic encryption systems can not be seen here. It should not be thought too literally. Otherwise, it can confuse people’s mind.

However, it is one of the way to visualize what homomorphic encryption is. Because, homomorphic encryption is not easy to understand or visualize without thinking tangible concepts [27].

4. GENTRY'S SCHEME

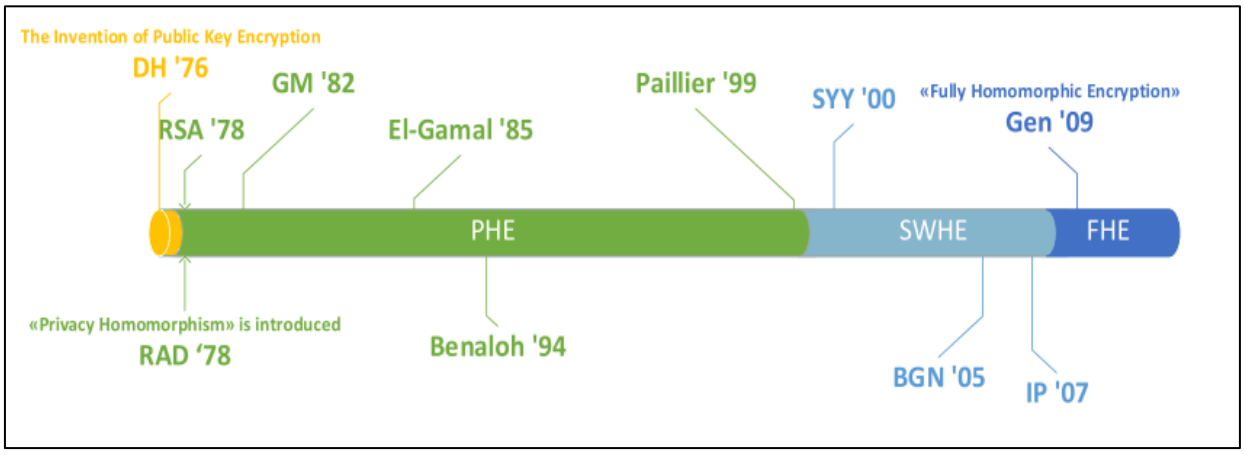

Figure 4.1. The evolution of encryption [28]

In 2009, Craig Gentry shows people a research, this research is the first fully homomorphic encryption (FHE) scheme. This research depended on a somewhat homomorphic encryption system. That is lattice based.

Somewhat homomorphic means that there is a homomorphic scheme which is only valid for restricted count of operations. After that, he showed a way to convert this lattice based; somewhat homomorphic encryption (SHE) to fully homomorphic encryption (FHE).

The name of this way is bootstrapping. However, the scheme is basically hard to use because of memory cost and computation.

After that, this research take attention from people. Many researches were done. Even fully homomorphic scheme has many flaws because of its cost of performance. The following schemes can be thought for a variety of applications practically : Yet Another Somewhat Homomorphic Encryption (YASHE), Simple Encrypted Arithmetic Library (SEAL) and Fan and Vercauteren (FV).

They are still at the center of a research due to trying new operations for decreasing the operation and storage cost. Before continue; mentioning FV and YASHE will be useful. Those are very famous schemes which are based on fully homomorphic encryption systems. They gave us a chance to solve the error growing problem. In addition, it does not require a modulus change.

However, Benabbas-Gennaro-Vahlis (BGV) scheme needs a a modulus change on contrary. In YASHE and FV, the steps are like following : error and messages are added.

After that this summation is added to the lower part vector. The reason of this is forming a ciphertext structure. Error is still buried to the cipher's lower part.

Later, q and p co-primes are selected. After that, data is multiplied by $\frac{q}{p}$. The reason of this behavior is taking data to the ciphertext's upper side. Errors and data are separated in the event of e<$\frac{q}{2p}$ on contrary to Benabbas-Gennaro-Vahlis scheme. Under these circumstances, multiplied encrypted texts must be decreased by a factor $\frac{p}{q}$ which accidentally decreases noise growth. Still it does not change the modulo.

Encrypted data supportive operations, which means homomorphic cryptosystems, have a wide usage in many applied fields. However; a lot of asymmetric system does not support both equation at the same time when the asymmetric key encryption is created. This phenomenon was explained by Rivest [29].

Namely, it can be seen that many asymmetric encryption systems give permission to addition or multiplication for ciphered info. Because, applying addition and multiplication at the same time is tougher, and all trials for creating a fully homomorphic encryption turned out to not safe.

Yet, Gentry announced the first reasonable system of a fully homomorphic encryption in 2009 [30]. The steps of his scheme are following. Bootstrapping and squash are defined.

- He generated a somewhat homomorphic scheme. This scheme gave him a chance to estimate low degree polynomials on the ciphered info,
- For expressing the deciphering step, it is needed to "squash" the decryption procedure.
- After this; to gain a fully homomorphic system, he computes "bootstrapping".

The most important part of this process is getting a system which could qualify high-enough degree polynomials. Also, this scheme must has a deciphering procedure simultaneously which could be explained as a low-enough degree polynomial.

This scheme is called bootstrappable; when the scale of polynomials could be qualified by the system which overcomes the scale of the deciphering polynomial.

It is transmitted to a fully homomorphic scheme [31]. Gentry said that this scheme in a bootstrappable scheme as a somewhat homomorphic cryptosystem. It is barely a system over lattices. After that; Gentry shows suitable key-creation steps. The safety of that system could be raised to the worst-case hardness of lattice problems. This is a system which is not bootstrappable yet.

For that reason; Gentry shows a procedure to squash the decipher process by decreasing the scale of the deciphering polynomial. A form of "sparse subset sum" (SSSP) is hint and it is the addition of secret key and public key.

At that procedure, the public key is enhanced with group of vectors. It means that there is a sparse subset of vectors that is added up to the ciphering key.

An ciphered text could be "post-performed" using this extra clue.

The post-performed ciphered text could be deciphered to a low-degree polynomial. The aim of this procedure is obtaining a bootstrappable scheme.

4.1. Encryption Process

The crucial process during ciphering is operating the degree (n−1) polynomial p at the point r. In this polynomial operation; Horner's rule is used. The procedure is starting with taking n−1 multiplications, but it is also applicable that for small coefficients it is possible to decrease the number of multiplications.

This decreasing can only be done to $O(\sqrt{n})$. Moreover, it is possible to accumulate this fast estimation algorithm. Also it is possible to estimate y such polynomials in time $O(\sqrt{yn})$. It can not be forgotten that estimating many $0,\pm1$ polynomials at the same point m can be processed as fast as a single polynomial.

Actually, if it is computed all the powers $(1,m,m^2,...,m^{n-1})$ at once; after that it can be evaluated each polynomial are the subset-summed up of these powers.

Due to the fact that multiplication is slower than addition, the decent element in the operation time would be the mathematical operations of the powers of m.

For every polynomial; it will be used just for one time. After that, the step can be estimated a single scale (n − 1) polynomial at a point m. It is operated very fast.

To do this operation; there has to be a given subprogram that estimates two degree(n/2−1) polynomials at the exact point m. What it is tried to be said is, given p(m) =$\sum_{i=0}^{n-1} p_i m^i$ it is cut it into a "bottom half".

$p^{bot}(m) = \sum_{i=0}^{\frac{n}{2}-1} p_i m^i$ and (4.1)

a "top half" $p^{top}(m) = \sum_{i=0}^{\frac{n}{2}-1} p_{i+d/2} m^i$ (4.2)

Evaluating these two smaller polynomials, it is found

$z^{bot} = p^{bot}(m)$ and $z^{top} = p^{top}(m)$, (4.3)

and then it can be computed z = p(m) by setting

$z = m^{n/2} z^{top} + z^{bot}$ (4.4)

If the subprogram computes the polynomials returns to the value of $m^{n/2}$, it is needed an extra multiplication for finding z = p(m).

Those above procedure shows that a repeated system to find the 0,±1 polynomial p of degree n − 1. Thus , it is recursively decrease the scale in half at the cost of duplicating the number of polynomials.

Trivial execution of processing all the powers of m can be used if the degree is small enough. To understand this approach, N(y, k) the number of multiplications can be showed as below. What it should not be forgotten is that taking y polynomials of degree (k − 1) for estimating. Understanding number of multiplication is very important for trivial execution. The formula can be found in following lines.

$$N(y, k) \leq \min(k - 1, N(2y,k/2) + y + 1) \quad (4.5)$$

For seeing the boundary of $N(y, k) \leq N(2y,k/2)+y+1$, remember that it is operated the upper- and lower-halves of all the y polynomials, one multiplication is needed for combining two for each polynomial.

After that; an extra multiplication for operation x^k from $x^{k/2} \cdot x^k$ has to be taken for following level of the repetition. $x^{k/2}$ was evaluated in the former level.

It is very obvious to see that taking the repeatedly call needs fewer multiplications by comparing the trivial execution any time when

$$k-1 > (k/2-1)+y+1. \quad (4.6)$$

In addition, the trivial execution is better while it is used the following formula. In this formula; the direction of the equation has to be changed.

$$k-1 < (k/2-1) + y + 1. \quad (4.7)$$

Then it is get the repeated steps

$$\begin{cases} N(y,k) = N(2y,k/2) + k + 1 & \text{when } k/2 > y + 1 \\ k - 1 & \text{otherwise.} \end{cases} \quad (4.8)$$

If that formula is solved we get $N(y, k) \leq \min(k-1,\sqrt{2}yk)$. Especially, the counts of multiplications are required for getting a single degree-$(k-1)$ polynomial is $N(1,k) \leq \sqrt{2}k$.

Of course, to apply this procedure, enough memory has to be provided. In the simulations at this research dimensions are only dimensions up to $k = 2^{15}$. However, in the future stopping recursion earlier could be done to save space.

It is also going to help to get rid of a long running time. As a solution , partial results can be stored in a disk. In the future, the existing experiments can show which approach cause a better result [32].

4.2. Decryption

The ciphertext c has the two matrices G,H. The vector

$$a = 2u + w \cdot e_1 \tag{4.9}$$

was used while a ← c mod G = [c × H/d]. After that, it shows the least significant bit of the first entry of a. This is w.

$$w := a_0 \bmod 2 \tag{4.10}$$

This deciphering process works because the rows of G are almost orthogonal to each other (and therefore also of H). So, the operator l∞-norm of H is small. It means that the biggest entry in x × H (in absolute value) for any vector x, can not be much bigger than the biggest entry in x itself. It can be said as a rule that , the above action reaches the success when every entry of a×H are smaller than d/2 (in absolute value).

It should be noted that a is the distance between c and some point in the lattice L(G). So, it can be showed as c as c = y×G + a for some integer vector y. It brings the below formula;

$$[c \times H/d]\times G = [y\times G \times H/d + a\times H/d] =^{(\star)} [a \times H/d]\times G \tag{4.11}$$

In this formula; y×G ×H/d is an integer vector where the equality (⋆) follows. The vector [a × H/d]×G is as itself, so it is needed [a × H/d]×G = a =(a×H/d)×G.

However this condition is true if and only if [a×H/d]= (a×H/d). So, a × H/d is equal to its fractional part. This means that half of each entry's absolute value must be greater than a×H/d.

4.3. An Optimized Decryption Procedure

The ciphered bit b can be regained by following steps :

It is known that $\alpha = 2u + w \cdot e_1$; and [c×H/d] = [a×H/d] = a×H/d which is mentioned previous section.

$$[c \times H]_d = [a \times H]_d = a \times H \quad (4.12)$$

So;

$$[c \times H]_d = [c \cdot \langle h0,h1,...,hn-1 \rangle]_d = \langle [ch0]_d,[ch1]_d,...,[chn-1]_d \rangle \quad (4.13)$$

Encrypted text vector can be showed that the ciphered bit b can be found by an easier approach. Remember that the encryption text vector c is decipher to the bit w when the length between c to the nearest vector in the lattice L(G) is of the form $a = 2u + we_1$, furthermore all the entities in $a \times H$ are smaller than d/2 in absolute value.

As it is mentioned, $[c \times H/d] = [a \times H/d] = a \times H/d$, which is true if and only if the equation (4.12), $[c \times H]_d = [a \times H]_d = a \times H$ is true. Remember $c = \langle c,0,...,0 \rangle$. So it can be written as following lines which was also mentioned in equation (4.13).

$$[c \times H]_d = [c \cdot \langle h0,h1,...,hn-1 \rangle]_d = \langle [ch0]_d,[ch1]_d,...,[chn-1]_d \rangle.$$

Also;

$$[c \times H]_d = a \times H = 2u \times H + we_1 \times H = 2u \times H + w \cdot \langle h_0,h_1,...,h_{n-1} \rangle \quad (4.14)$$

If we gather up this two equations, it is obtained equation (4.15)

$$\langle [ch0]_d,[ch1]_d,...,[chn-1]_d \rangle = w \cdot \langle h_0,h_1,...,h_{n-1} \rangle \pmod 2 \quad (4.15)$$

This is a general representation of any decryptable ciphertext c. To put in a different way; $[c \cdot hi]_d = w \cdot h_i \pmod 2$ for all i values. It is enough to keep only one odd w_i's and then recover $b := [c \cdot w_i]_d \bmod 2$.

4.4. Basics for Homomorphic Encryption

A group of independent vectors could be the base of a vector space. As an instance, an identity matrix is a base for the Euclidean space. Every column of it is an independent vector. As it is known that linear combination of these vectors generates the space. Space is also get by using pointing coordinates.

For instance, pointing coordinates at (2, 5,1) means that this point can be reached by adding the first basis vector twice, the second one is five times and the final one once.

To define a lattice; it can be said that it is the set of vectors that are based on basis vectors. It should be noted that that vectors should have integer coefficients.

So, it is obvious that this is a subset of the vector space. The mathematical operations addition, subtraction, multiplication by an integer are available. It is the lattice. It can be seen in Figure 4.2.

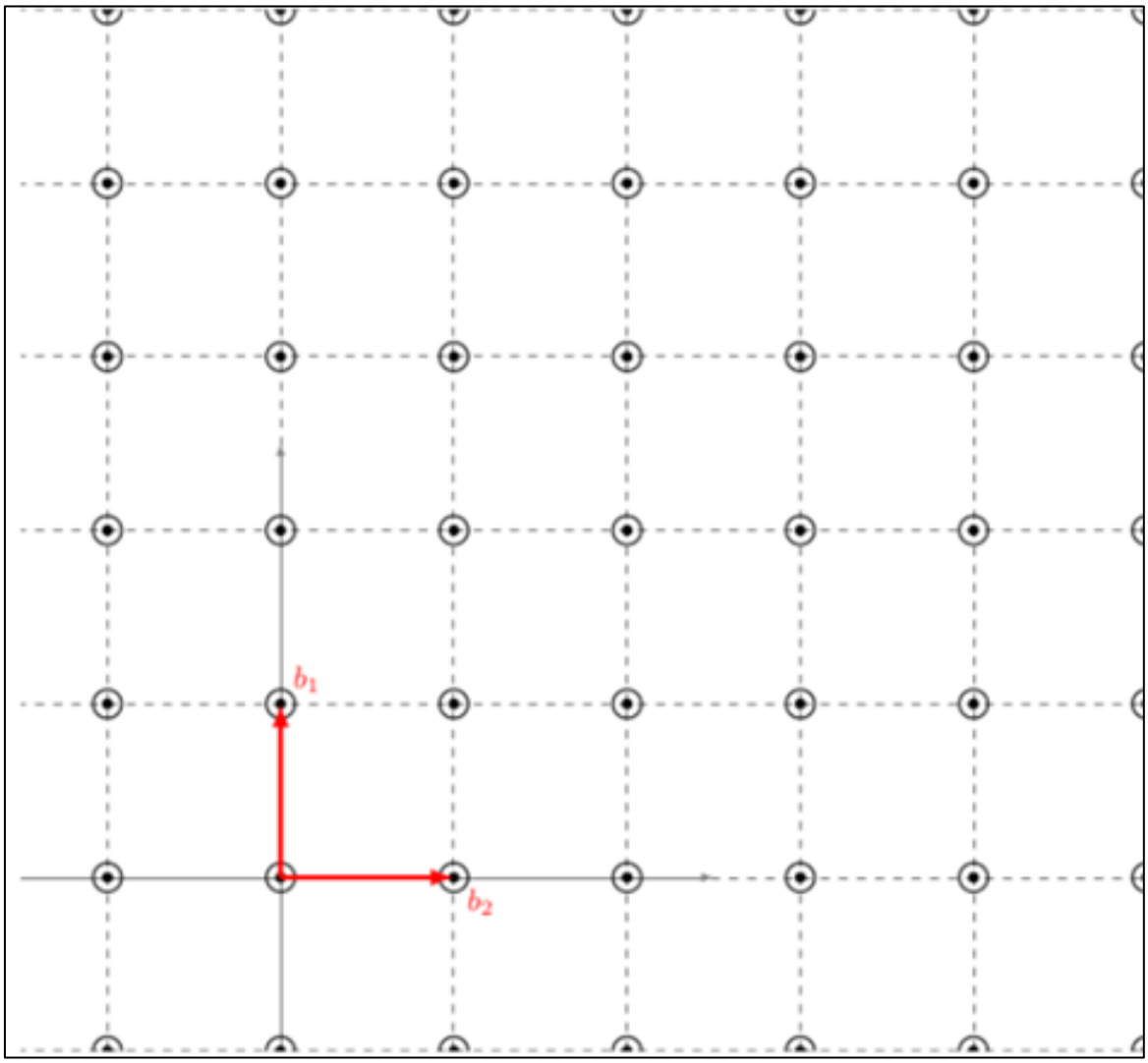

Figure 4.2. Lattice points

There is a closest lattice point for a vector which is not in the lattice. This point shows the closest lattice vector. This vector is closest to the vector which is evaluated by the point in the vector space. The result of this shows the Closest Vector Problem (CVP).

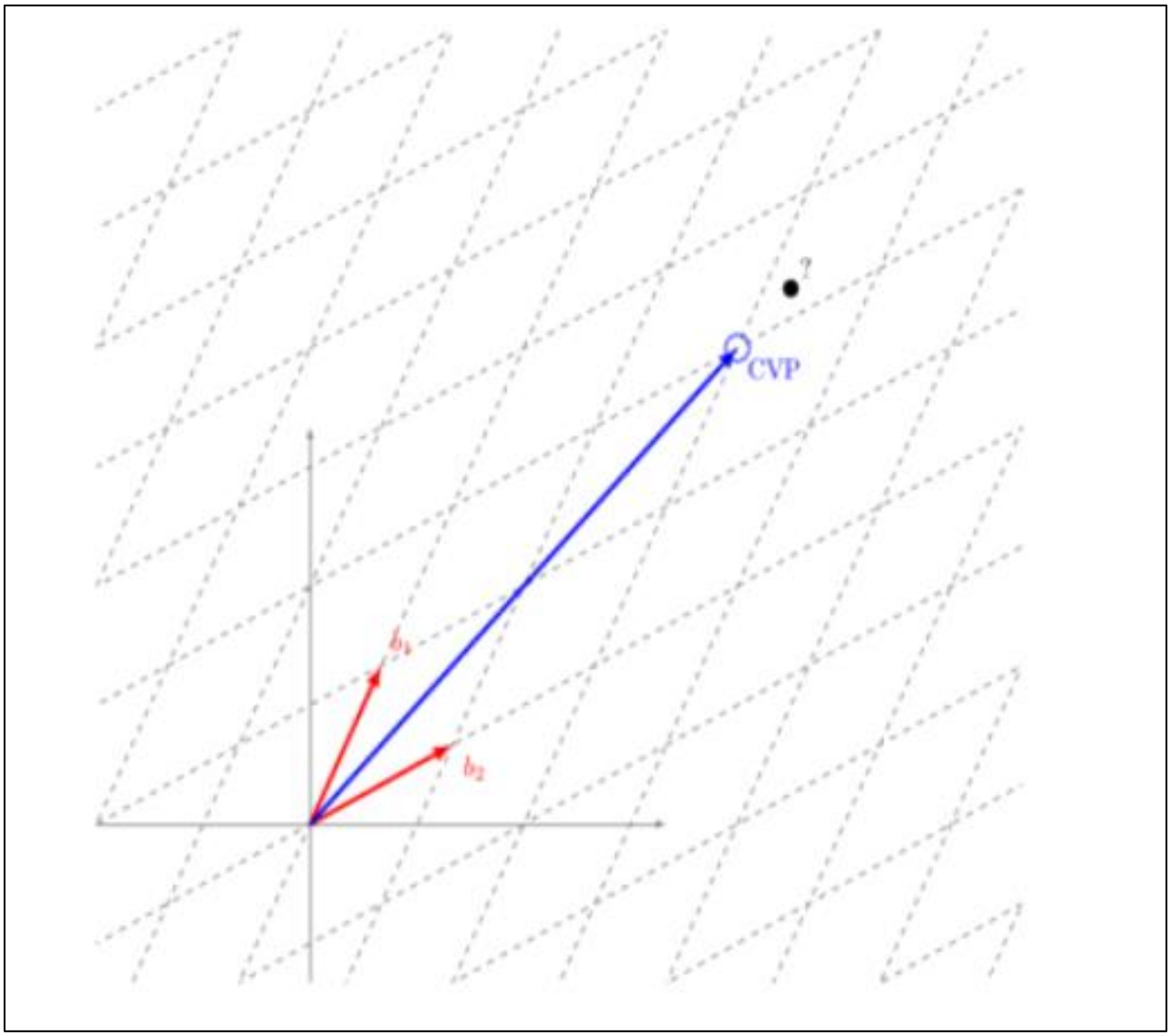

Figure 4.3. Closest vector point

4.5. Learning with Error Problem (LWE)

Learning with Error (LWE) is thought as one of the toughest issue to clarify in practical time for even post-quantum algorithms. This problem was announced by Oded Regev at 2009. The first name was "learning from parity with error" problem [33].

Regev showed that it is possible to solve Shortest Vector Problem (SVP) and LWE problems with the same algorithm. It means that if there is an algorithm that can solve LWE problem in an efficient time, this algorithm can be used for SVP problems, too.

Trying to regain the input message from a given encrypted test means solving the CVP and SVP problems.

To understand following procedure it can be thought that vectors and their small error are in a combination. The matter of recognizing the final linear combination with error by a totally excursive vector is called the Learning With Error problem. In other words, it is tried to be gain the vector of lattice which is the nearest to the vector with noise.

Let’s take modula t (random chose) of vector's coefficients. It is thought , this space as a box. Anytime a vector comes in, and goes out. It comes from other side.

For this example, it is easier to picture the algorithm which can solve the Closest Vector Point (CVP). It should be solved for every dimension, means horizontal and vertical one after the other should service our purpose.

Figure 4.4. shows the Euclidean space. If it is examined, it can be seen that this has a orthogonal basis. It is easy to solve the closest vector point problem for this system. However, the Figure 4.5 is harder. It is still easy to find a solution but not it is not easier as the previous one. The problem is because the vectors are not entirely orthogonal.

If Figure 4.6 is checked, solving the closest vector problem will be harder. This time we will need a technique to solve this problem. The reality behind this; these vectors are not totally perpendicular. This complicates our situation a little bit. However, it is possible to solve the problem.If one does not have any notion of basis reduction, it is not an easy problem to solve.

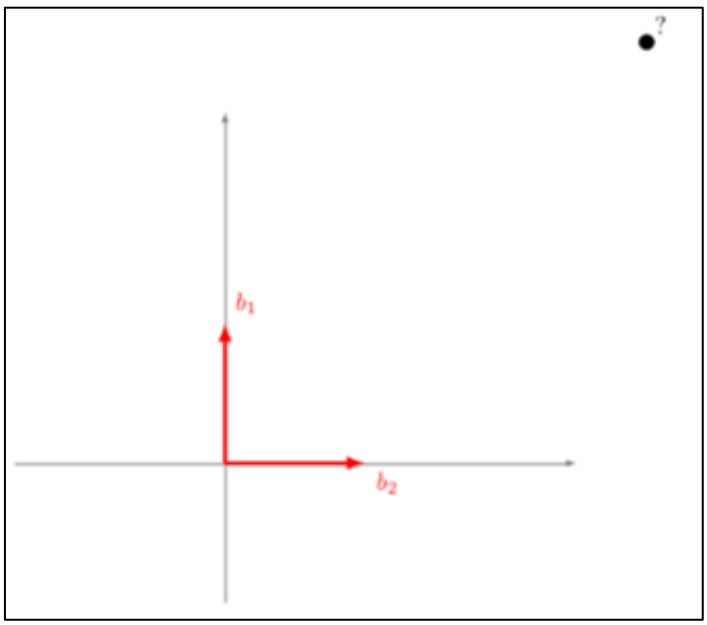

Figure 4.4. CVP Euclidean

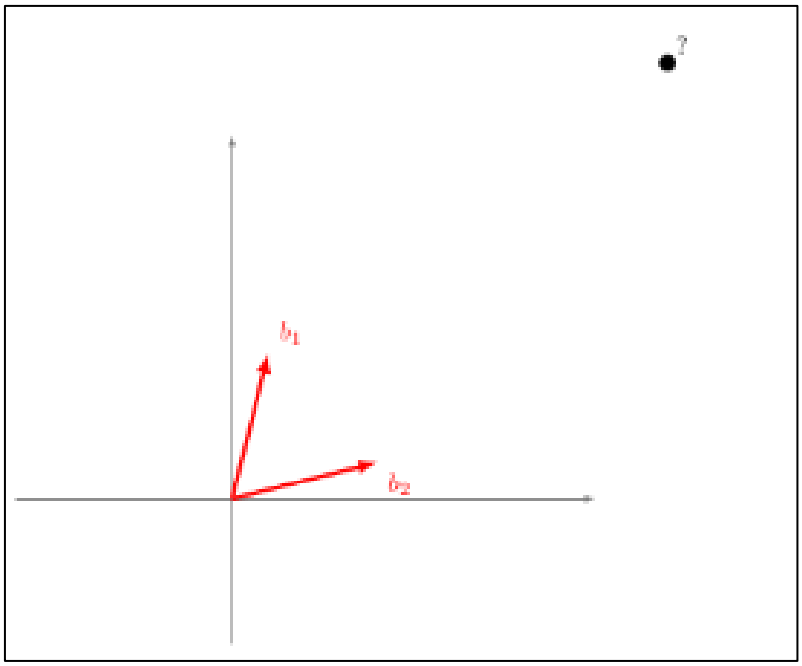

Figure 4.5. CVP-light

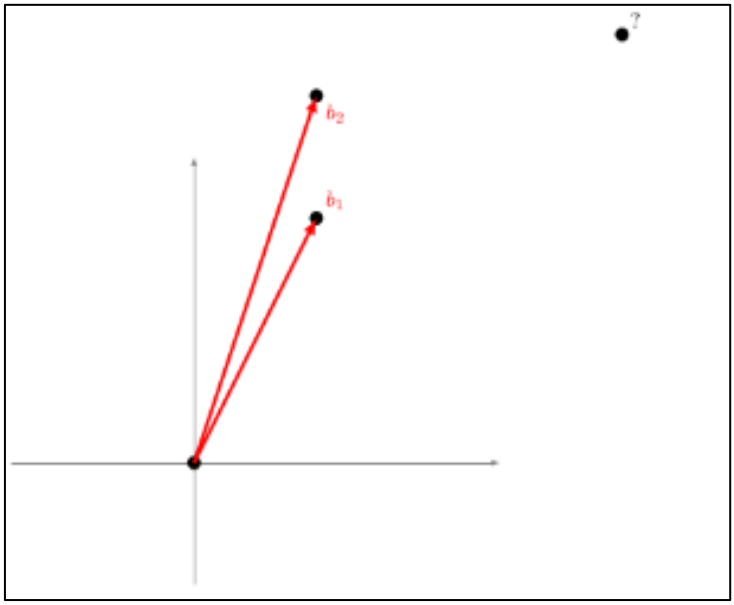

Figure 4.6. CVP-hard

We can use a methodology to solve this problem

- Let *M* has n dimensions.
- Let *A* be a unimodular-matrix whose determinant is equal to ±1.
- Then the basis $M' = M \times A$ generates the exact lattice as *M* meaning $L(M') = L(M)$. Also, it can be show like $L(MXA) = L(M)$.

It is easy to see that for this lattice, its elementary is *not* unequal. The determinant of its elementary matrix $M3$ is... -1, for instance. $M'=M1\times M3=M3$, therefore $L(M')=L(M3)=L(M1)$ when $M1$ is the identity matrix.

Here, it can be seen a simulation for LWE below:

It is showed that LWE with binary error, that is thought before in some research that is fine with this category.

LWE with binary error is cheaper to show than it is waited, changing the hybrid lattice-scale increasing and meet-in-the-middle attack by Howgrave-Graham are modified, use it to this setting. After that its difficulty is analyzed.

The research shows that the attack outperforms regular approaches such as the enumeration attack. Furthermore; the following approach can be used after some parameter sets. Figure 4.7. shows the development, by comparing the runtime of the best previously known attack with the hybrid attack. In this simulation; y = 2k samples from a LWE system with binary error, also k is the dimension of the secret vector.

As an instance, in the case of k = 256 and t = 256, the hardness of the problem decreases from 117 to 103 bits. It is obviously a crucial development.

A comparison between the hybrid attack and former systems can be seen below.

The orange line shows the previous work and the blue line shows the last work. In this simulation, Hardness of LWE length for m = 2k; modulus t = 256.

Solving LWE relates with finding a more acceptable lattice basis. This lattice must be as perpendicular as possible. This is also known as SVP (Shortest Vector Problem).

This shortest vector situation can be solved by the reduction CVP to SVP. Gentry's scheme depends on the fact on Ring Learning With Errors(RLWE).The difficulty of solving LWE is dependable to gaining a better basis for a given lattice. The most common algorithm for elementary scale decreasing is Lenstra-Lenstra-Lovasz also known as (LLL) or (3L) Algorithm. The cost of an approximation is exponential in the number of dimensions.

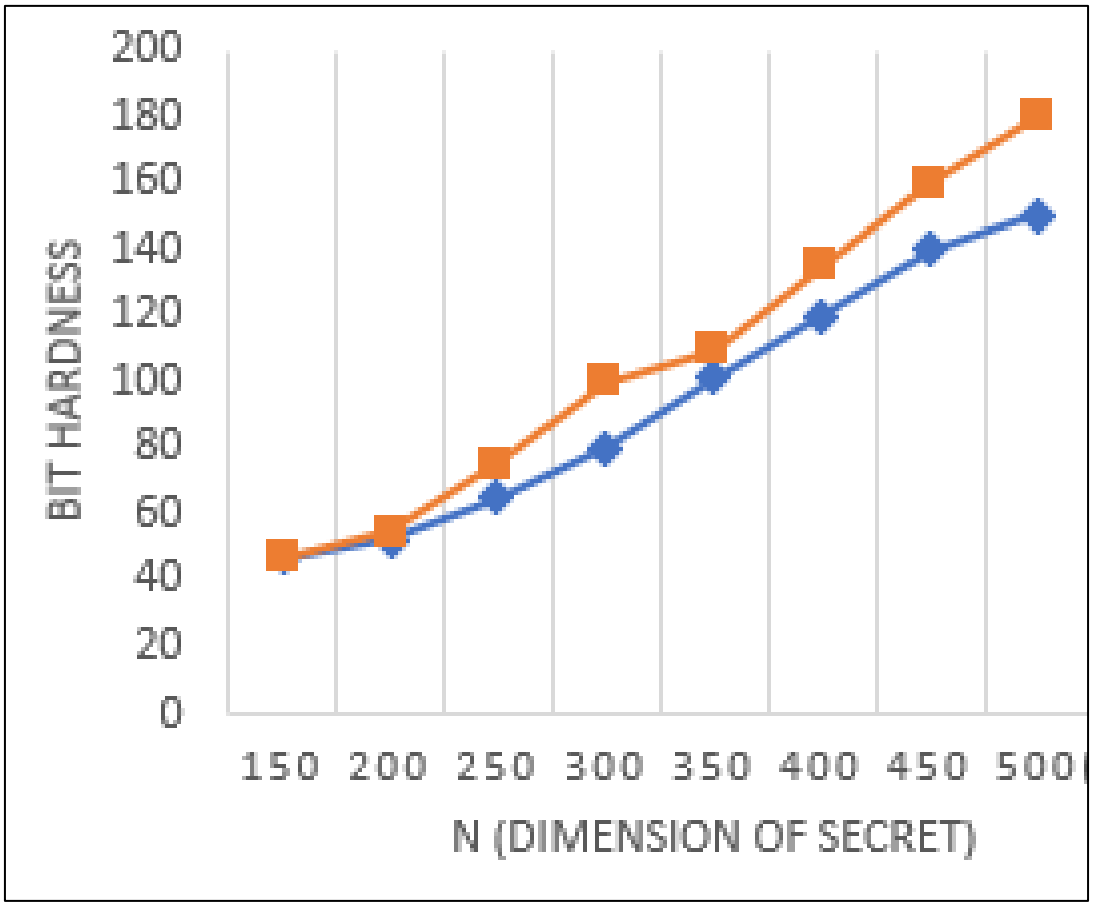

Figure 4.7. Improvement

Unfortunately; CVP creates an error whether the approximation is too dependent to our space (modulo q).

Let us check the below information

- R_a the ring of integers modulo a
- R_b the ring of integers modulo b.

It has to assumed that assume a≪b. Message space has integers in R_a. Simultaneously; ciphertext has coefficients in R_b. The vector dimension is n. with, reduction modulo b produces a result in $]-b/2, b/2]$.

The scalar product of c_1 and s in the ring of integers also known as R_b can be done by addition of two ciphertexts

$$\sum_{i=0}^{k} c1, i \cdot si \tag{4.16}$$

The scalar product of each of the matrix row vector with s can be created by the scalar product of a matrix . If the vector e and integer p are multiplied, their coefficient will be same as the integer p product vector e's production.

If it is talked somewhat homomorphic encryption, it should be mentioned these three criteria.

- a basis;
- a vector x_1 that is collaboration of the basis vectors without the error.
- a vector c_1 that is the result of a collaboration of the basis vectors plus some small error.

CVP means solving x_1 from c_1 It is not easy for out of the distance. So, the messages will be hidden in the error. This needs a solution to deviate the message from the error. By getting the data space modulo some integer a <<b and to take an error which can be divided by a.

By applying this, using a modulo a on our combination of message/error erasing the error part. However, if it is added two ciphered texts, the resulting vector will be the result of addition of sum of errors and another lattice point.

In the following figures addition homomorphism examples can be seen. Therefore, deciphering the final encrypted text and removing the error gives us m_1+m_2. Even though; it is not easier to show a graphic example, the same is applicable for multiplication.

Unfortunately, a key element of deciphering is that our addition of error/message is tiny. For getting the closest lattice point could turn out to false point actually if its norm is too high. Moreover, any computation causes the error growth.

This can causes failure on the deciphering. This results can be explained as two key notions but before that it should be recalled that, k should be big by comparing to b for making closest Vector Point (CVP) hard.

Addition homomorphism examples can be found at following page. It is possible to see c values. Two key notions are following;

- To make encryption successful, a is needed to be small by comparing to b.
- Making a as small as possible can protect our deciphering process from failing.

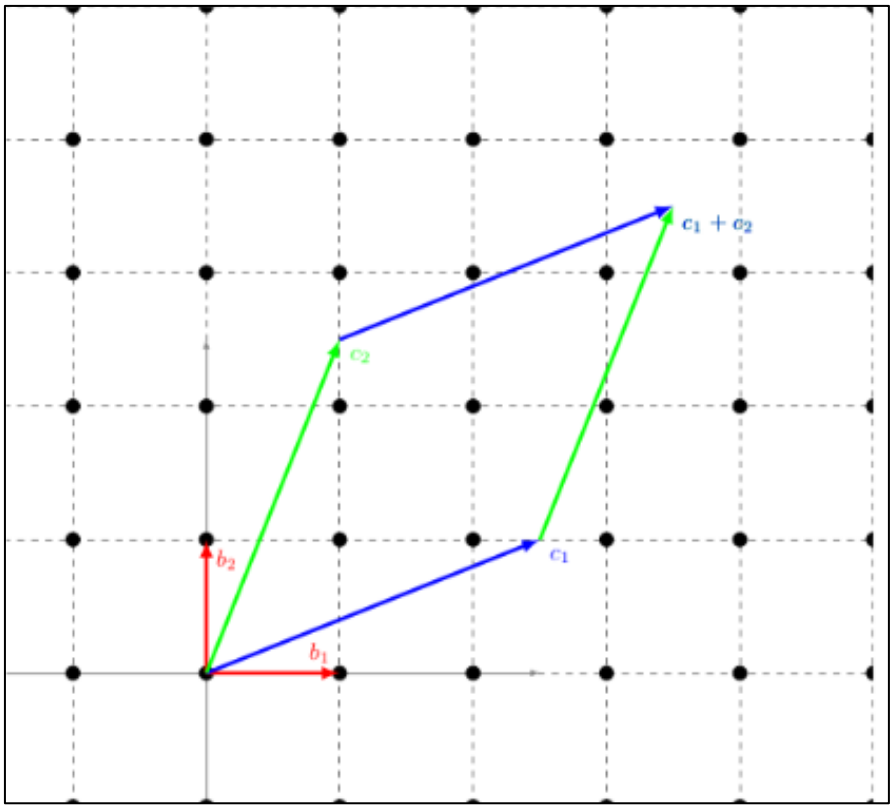

Figure 4.8. Addition homomorphism

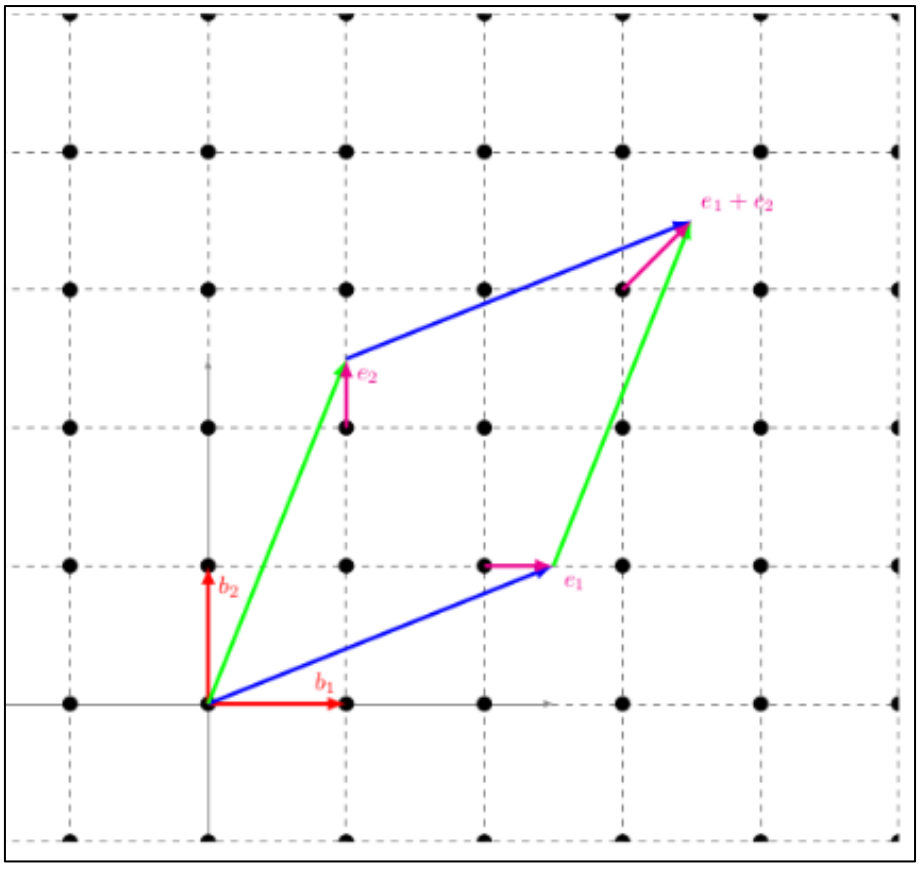

Figure 4.9. Addition homomorphism

Those notions shows that why fully homomorphic schemes causes more cost in both memory allocation and operation time. The more the b , the more the k as well.

Finally , the error growth is so crucial in multiplication as compared to addition. Due to the fact that it is said multiplicative depth instead of computation depth [34].

In that point, we can also remember the Ring learning with errors (RLWE). RLWE is an extension version of learning with errors (LWE). Algorithmic improvements and proofs for security on LWE are also accurate on RLWE.

The idea behind RLWE is assuming the vectors by looking at polynomials modulo of nth cyclotomic polynomials. n has to be power of 2. Making an operation on polynomials has better complexity than vectors.

Due to the fact that polynomials have cascading qualifications over certain rings (NTT).

4.6. From Somewhat Homomorphic Encryption to Fully Homomorphic Encryption

Somewhat homomorphic encryption scheme can operate the encrypted text homomorphically for only a finite number of operations. After that, the decryption function fails.

This means that it is not possible to regain the original message from the encrypted text in a correct way. For avoiding this problem; Gentry used two methods called squashing and bootstrapping.

Gentry's theory proposes any somewhat homomorphic encryption (SHE) scheme could be transmitted to a fully homomorphic encryption (FHE) system by using bootstrapping and squashing**.** Bootstrapping is simply reciphering procedure to obtain a fresh ciphertext from the noisy ciphertext. These two structure use same plaintext.

A scheme is can be thought as bootstrappable if it can create its own decryption algorithm circuit [35]. However; Gentry’s bootstrapping technique is applicable only for the small depth decryption algorithms. So, Gentry tried to decrease the complexity of decryption algorithms. This method is known as squashing.

The steps are explained in following lines:

- Firstly, a vector set has to be chosen. The addition of these vectors must be equal to the multiplicative inverse of the secret key.
- If the encrypted text is multiplied with this chosen set, the degree of polynomial circuit will be decreased to a level that can be calculated.
- The ciphertext can be called as bootstrappable after these steps. At this point, it is better the remember that; the secret key is based on the assumption of Sparse Subset Sum Problem (SSSP) [36].

All in all; when a homomorphic scheme is given, any function can be operated homomorphically by using bootstrapping , involving the deciphering function [37]. By doing this, it can be seen that next information:

- encrypting an old key with the new one;
- encrypting an already-encrypted-data with a new key;
- estimating the deciphering homomorphically. This causes in a ciphertext only encrypted with the second key.

$c_1' = Enc(c_1, s_2)$

$s_1' = Enc(s_1, s_2)$

$c_2' = Enc(c_{1'}, s_{\cdot})$ →it is the error from decryption.

This thought is visualized at Figure 4.11. Even we can visualized this thought very easily, it not that easy to do this in practical life. Of course, there are some problems on this approach. Those are;

- The operation of decryption has huge multiplicative depth.
- Ciphering an already ciphered message generates a quadratic overhead.

It should not be forgotten that bootstrapping method inclines the computational cost very much. Unfortunately, it becomes a big problem for the practicality of fully homomorphic

encryption schemes. Studies on FHE has two main problem:

- Increasing multiplicative depth to enable bootstrapping and decreasing the overhead.
- Discovering ways to reduce encrypted texts' errors.

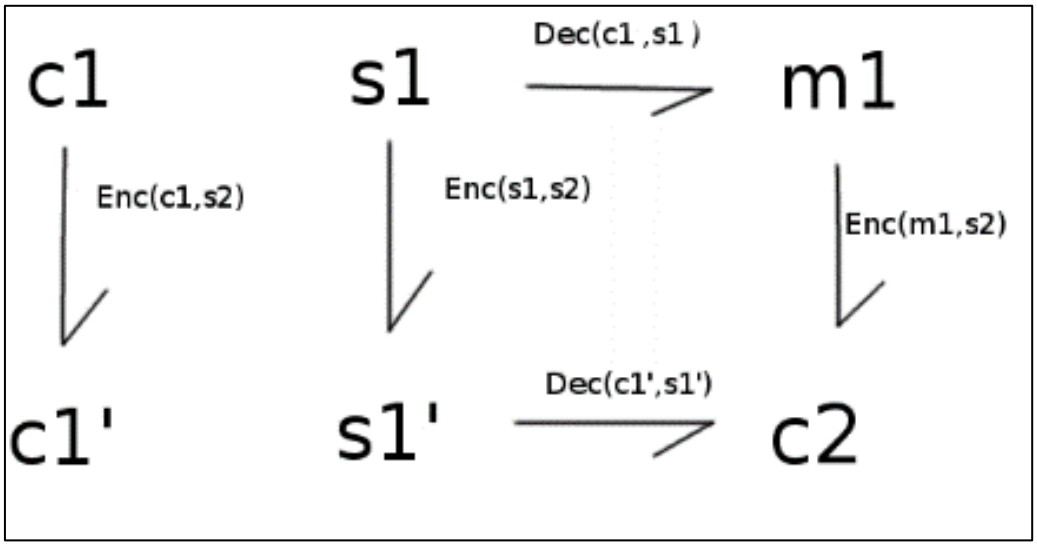

Figure 4.10. SHE to FHE [38]

5. SIMULATION OF A HOMOMORPHIC ENCRYPTION SYSTEM

In this thesis research, simulation of a homomorphic encryption system is examined. The comparisons are made.

5.1. Lattice Based Cryptography (Somewhat Homomorphic Encryption)

Homomorphic encryption systems are procedures that permits the conversion. The conversion is from encryption texts E(D) of data D, to cipher texts E(f(D)) of a computation or function of data D, without opening the data. It is mentioned in the previous chapters.

Somewhat homomorphic encryption (SHE) is also known as lattice based cryptography. The most well-known SHE method is Boneh-Goh-Nissim (BGN) method.

Any number of additions can be proceed with this method but only one multiplication can be processed on data. Two ciphertext can be added; or a ciphertext and plaintext can be multiplied.

An ciphertext scheme contains commonly three step algorithm. They are,

- Key Creation, generates two keys, the secret key s_k and the public key p_k.
- Ciphering, decipher the plaintext m with the public key p_k to ensure encrypted text c.
- Decryption, decipher the encrypted text c with the secret key s_k to regain the plaintext m [39].

Visualization can be seen at Figure 5.1. below.

Parameters, for this research, simulation of a homomorphic encryption is examined. While doing this, some parameters are used. Actually, the construction has a lot of parameters, the bit-length of the variety of integers ; monitoring the number of integers in the public key.

Even this process has many variables, but the most important four of them can be found in the following page. All these variables are used in the key generation, encryption and decryption process. Also, noise parameter should not be forgotten in the system while public and private keys are created for the system.

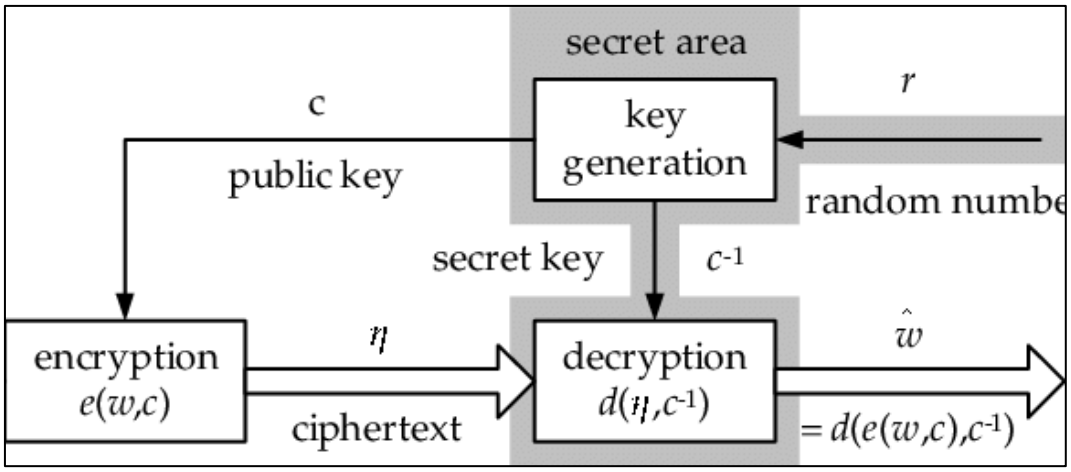

Figure 5.1. The basic concept of encryption scheme [40]

- γ is integer bit-length in the public key,
- *a* is the bit-length of the secret key (which is the hidden approximate-greatest common divisor of all the public-key integers),
- *n* is the bit-length of the noise (i.e., the distance between the public key elements and the nearest multiples of the secret key), and
- τ is the number of integers in the public key [41].

Those variables helps us to monitor the number of integers in the public key and bit-length of the divergent integers. The security parameter λ is valid for all polynomial.

While those parameters are set, following constraints must be obeyed:

- $n = \omega(\log\lambda)$, for avoiding brute-force attacks on the noise;
- $a \geq n \cdot \Theta(\lambda \log^2 \lambda)$, for supporting homomorphism for deep enough circuits;
- $\gamma = \omega(a^2 \log\lambda)$, to prevent a variety of lattice-based attacks on the underlying approximate-greatest common divisor problem;
- $\tau \geq \gamma + \omega(\log\lambda)$, for using the leftover hash lemma in the reduction to approximate greatest common divisor.

In this simulation, a secondary noise parameter will also be used. This parameter can be found at equation (5.1), this secondary noise is required because when a multiplication is done, noise will be increased. So, a new demonstration is necessary.

$$n' = n + \omega(\log\lambda) \tag{5.1}$$

A suitable parameter set to remember is

- $n = \lambda$,
- $n' = 2\lambda$,
- $a = \tilde{O}(\lambda^2)$,
- $\gamma = \tilde{O}(\lambda^5)$ and
- $\tau = \gamma + \lambda$.

This setting results in a scheme with complexity $\tilde{O}(\lambda^{10})$. The following distribution over γ-bit integers can be used for a specific (a -bit) odd positive integer n :

$$D_{\gamma,\rho}(p) = \{\text{choose } q \leftarrow Z \cap [0, 2^{\gamma}/n), r \leftarrow^{\$} Z \cap (-2^{n}, 2^{n}) : \text{output } x = n\,q + r_o \quad (5.2)$$

This distribution is obviously efficiently sampleable.

5.2. The Construction

We can divide construction parts to four steps. These steps are key generation step, encryption step, evaluation step and decryption step. The formulas and explanations for these four steps are following;

KeyGen(λ). Suppose that an asymmetric encryption system is used for the system. It means that two keys are needed. One is private key while the other one is public key. The secret key is an odd a-bit integer:

$$n \leftarrow^{\$} (2Z+1) \cap [2^{a-1}, 2a) \quad (5.3)$$

The public key is $n_k = x_0, x_1, \ldots, x_\tau$. To get that the public key, it should be sampled

$$x_i \leftarrow^{\$} D_{\gamma,n}(n) \text{ for } i = 0, \ldots, \tau. \quad (5.4)$$

To make x_0 the largest, it should be relabeled. If x_0 is not odd and $r_n(x_0)$ is not even, procedure should be restarted.

The other function that has to be used after key generation step is Encrypt function.

Encrypt (pk,m∈{ 0,1}). S ⊆{ 1,2,...,τ} should be chosen and an integer r in $(-2^{n'}, 2^{n'})$, and output c ← [m +2r +2 $\sum_{i\in s} xi]_{x0}$ also should be chosen.

Evaluate (n k,C,c_1,...,c_t). Circuit *C*ε has t inputs, and t ciphertexts c_i . Given the (binary) circuit *C*ε, it should be applied the addition and multiplication gates of *C*ε to the encrypted texts, for computing all the equations on the integers. And also to return to final integer.

Decrypt (sk,c). Output m'←(c mod *n*) mod 2. Recall that (c mod *n*) =c− *n* ·[c/ *n*]. Due to the fact that *n* is odd, following formula can be used for deciphering

$$m' \leftarrow [c-[c/\ n]]_2 = (\ c \bmod 2) \oplus ([c/\ n] \bmod 2) \tag{5.5}$$

Formerly, encryption is defined as adding m to a random subset sum of *encryptions of zero*. Actually, the system can be thought like that."

$$w_i = [2x_i]_{x0} \text{ for } i = 1,...,\tau. \tag{5.6}$$

Each w_i, and also x_0, is a ciphering of zero. The noise of them is even. Furthermore,

$$c = m+2r +\sum_{i\in s} wi - k.\ x_0 \text{ for some integer k.} \tag{5.7}$$

5.3. Correctness

For a mod-2 circuit, it can be seen its generalization for integers, for example the equal circuits with the addition or multiplication gates implemented to integers rather than to bits. Like Gentry, it is described a permitted circuit as one where for any α ≥ 1 and any set of integer inputs all less than 2α(*n* +2) in absolute value."

It shows that the general circuit's output has absolute value at most $2^{\alpha(a-4)}$. Since "new" encrypted texts output by Encrypt have noise at most $2^{n'+2}$, the encrypted text output by Evaluate operated to a permitted circuit has noise at most $2^{a-4} < n/8$.

The boundary $2^{a-2} < n/2$ would extend for true deciphering. The reality that the noise stays below n /8 will be used later to proceed the deciphering procedure using a large shallow arithmetic circuit.

The description of the set $C\varepsilon$ from above is indirectly. Specifically; this description does not mean a good picture of what $C\varepsilon$ "seems". By the triangle inequality, a *k*-fan-in Add gate clearly ascends the size of the integers by at most a factor of *k.* Although, a 2-fan-in Mult gate may square the size of the integers – i.e., double their bit-lengths.

Thus, obviously, the basic problem is multiplicative depth of the circuit, or the scale of the multivariate polynomial proceeding by the circuit.

Lemma . Let C be a Boolean circuit with t inputs, and let $C^{\dagger}$ be the associated integer circuit. It has to be thought that Boolean gates are changed places with integer operations.

Let $f(x_1,...,x_t)$ be the multivariate polynomial operation by $C^{\dagger}$; let take d as its degree. Where |f| is the l_1 norm of the coefficient vector of f, If

$$|f| \cdot (2^{n'+2})^d \leq 2^{a-4} \tag{5.8}$$

Then we can say $C \in C\varepsilon$.

In particular, ε can handle f as long as

$$d \leq \frac{a-4-\log|f|}{n'+2} \tag{5.9}$$

So it is affected to polynomials that ensures above equation as permitted polynomials and it is noted by $P\varepsilon$ the set of permitted polynomials and by $C(P\varepsilon)$ the set of circuits that operate them.

The discussion above implies that $C(P\varepsilon) \subseteq C\varepsilon$. *Remark*. For the purpose, it is seen settings where log|**f**| is small in relation to

$$a, n' = \omega(\log\lambda) \text{ and } t,\tau \leq \lambda^{\beta} \tag{5.10}$$

and it is needed to ensure polynomials of degree up to $\alpha\lambda\log^2\lambda$ (for some constants α,β). Placing these expressions , it is enough to set equation (5.11).

$$a = n \cdot \Theta(\lambda\log^2\lambda) \tag{5.11}$$

5.4. Somewhat Homomorphic Encryption Simulation Study

A somewhat homomorphic encryption simulation is done by using the above formulas. The details will be at following.

5.4.1. Simulation environment

In this research environment, safe parameter λ's length is taken as 17 bits, the order of magnitude $n = \lambda$ will be 10^5 ; order of magnitude $n' = 2\lambda$ as the 10^5, order of magnitude for $a = \tilde{O}(\lambda^2)$, as the 10^{10}; order of magnitude for $\gamma = \tilde{O}(\lambda^5)$ as the 10^{25} ; order of magnitude for $\theta = \tilde{O}(\lambda^4)$ as the 10^{20}.

5.4.2. Simulation results

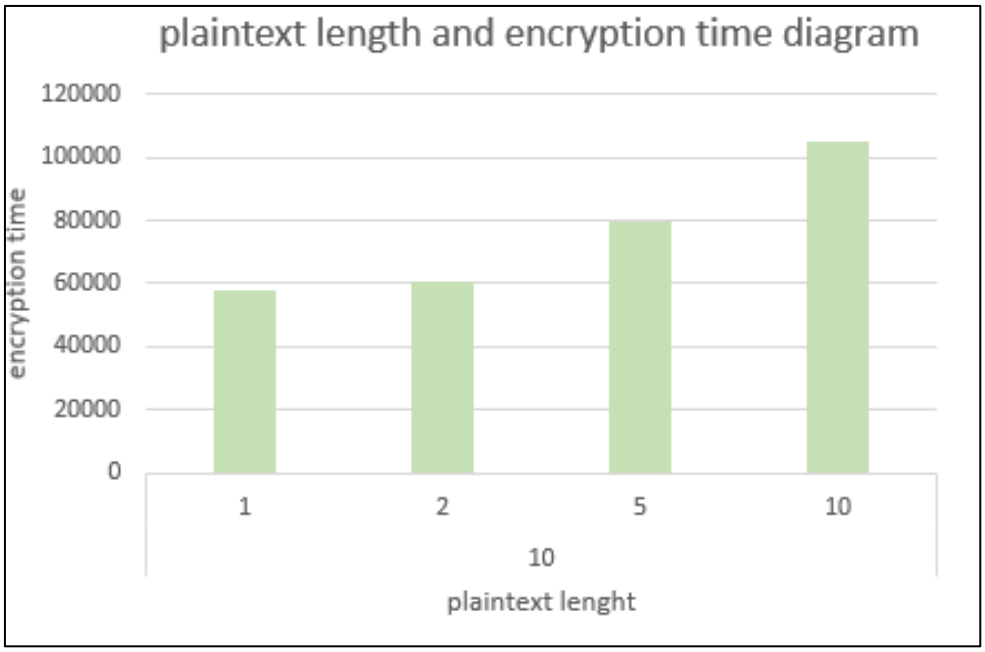

Figure 5.2. Plaintext length and encryption time diagram

For analyzing the data, it has to be recalled that magnitude and encrypted text are thought to bit size and the time in seconds. Due to the fact that the environment and algorithm restricts, the magnitude of the plaintext has only 10 bit.

The clear text size is in order of for 1 bit, 2 bit, 5 bit and 10 bit to experiment. Cleartext is the unencrypted form of an encrypted text; plain text is first unencrypted text. The encryption text size is very huge.

However, encrypted text results are calculated by bit as unit for being sure of the stability of the order of magnitude of the operations.

When the simulation is done; it can be how the bit size and encrypted text are related to each other in Figure 5.2. It is obvious to see that when the clear text size is increasing the encrypted text is also inclines. But the growth rate is not high.

So, it can be see that size of the alteration has an influence on size of the cipher text. If the plaintext sizes with time are compared to experiment costs ; it is get Figure 5.3.

Through the Figure 5.3 it can be understood that the alteration of plaintext size can alter the time which is running out for encryption is not stable. Also; increasing the time costs will be increased the size of plaintext gradually.

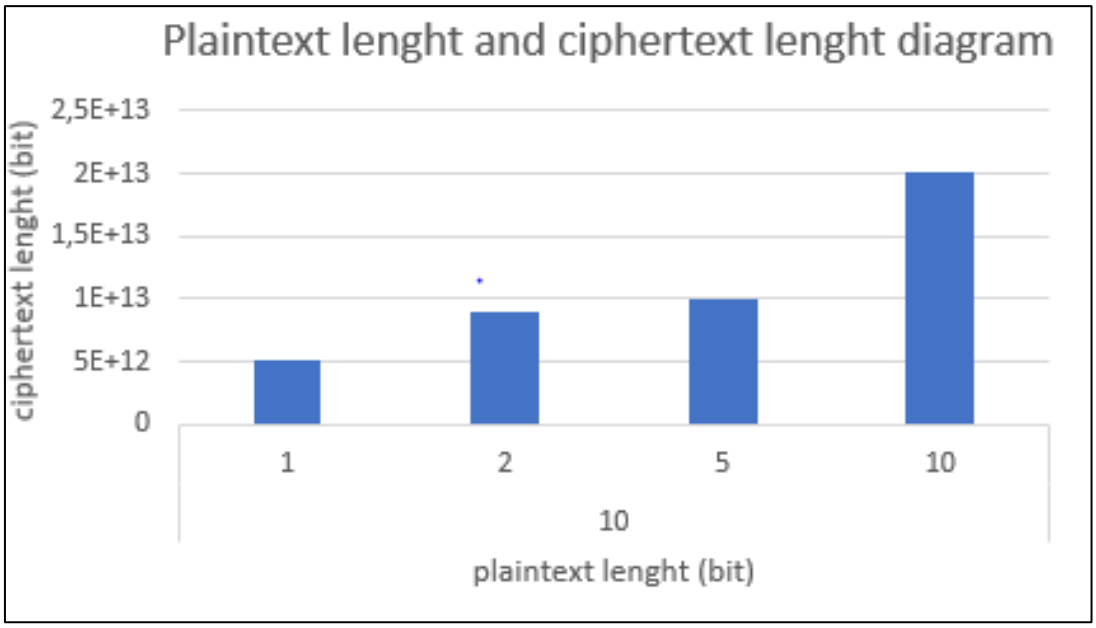

Figure 5.3. Plaintext length and cipher text length diagram (bit)

5.4.3. Result

For applying this procedure, it is assumed that we have enough memory. In the simulations which are done later on this section, dimensions are only up to $n = 2^{15}$. In the future stopping recursion earlier could be done to save space and get rid of a long running time. Alternatively, partial results can be stored in a disk. Homomorphic encryption is a relatively new technology. Because of this reason, better results will be get in the future.

New experiments will show us which approach causes a better result [42].

5.5. Another Simulation System

In the following simulation; key pairs for n (dimension) and t (bit-length) are created. After that; many bits were encrypted, estimated on the ciphertexts many elementary symmetric polynomials of various degrees and number of variables.

And finally, the results are decrypted and it is controlled whether it is get the same polynomials in the plaintext bits or not.

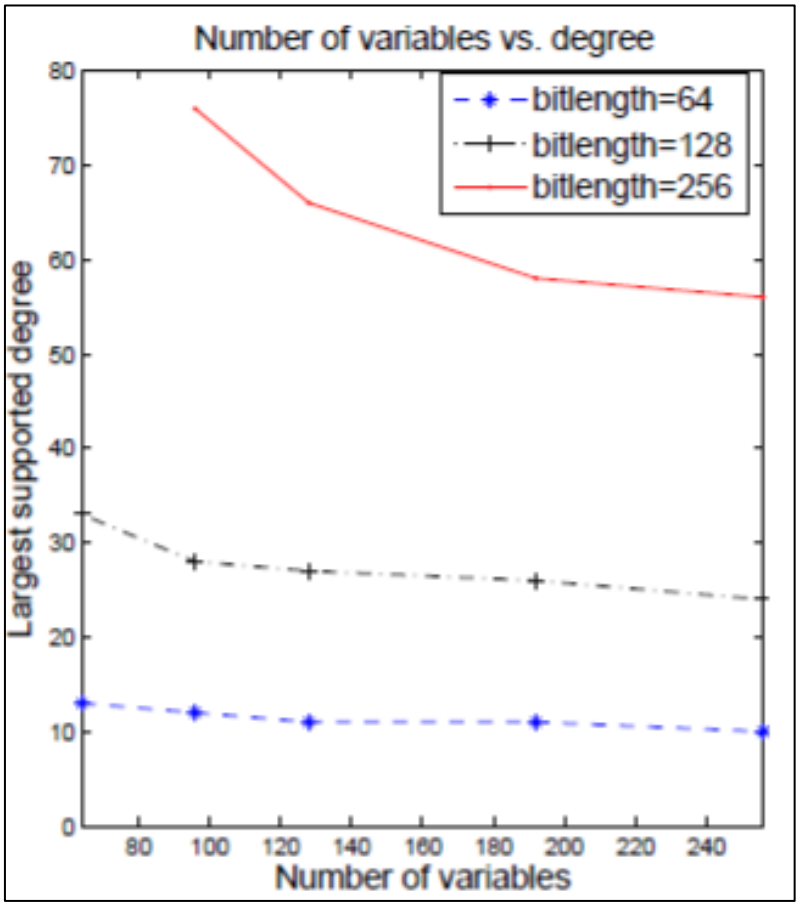

Figure 5.4. The relationship between number of variables and largest supported degree

The polynomials which are from 64 to 256 variables are tested. For every variable block, 12 tests are done. In every test; m bits are encrypted. All symmetric polynomials in these variables are evaluated. After that, those polynomials are applied to the plaintext. And a comparison was done between them and plaintext. For each m, 12 tests are recorded. Those tests cause a decryption to correct value.

It can be called as largest supported degree(LSD) for those parameters.

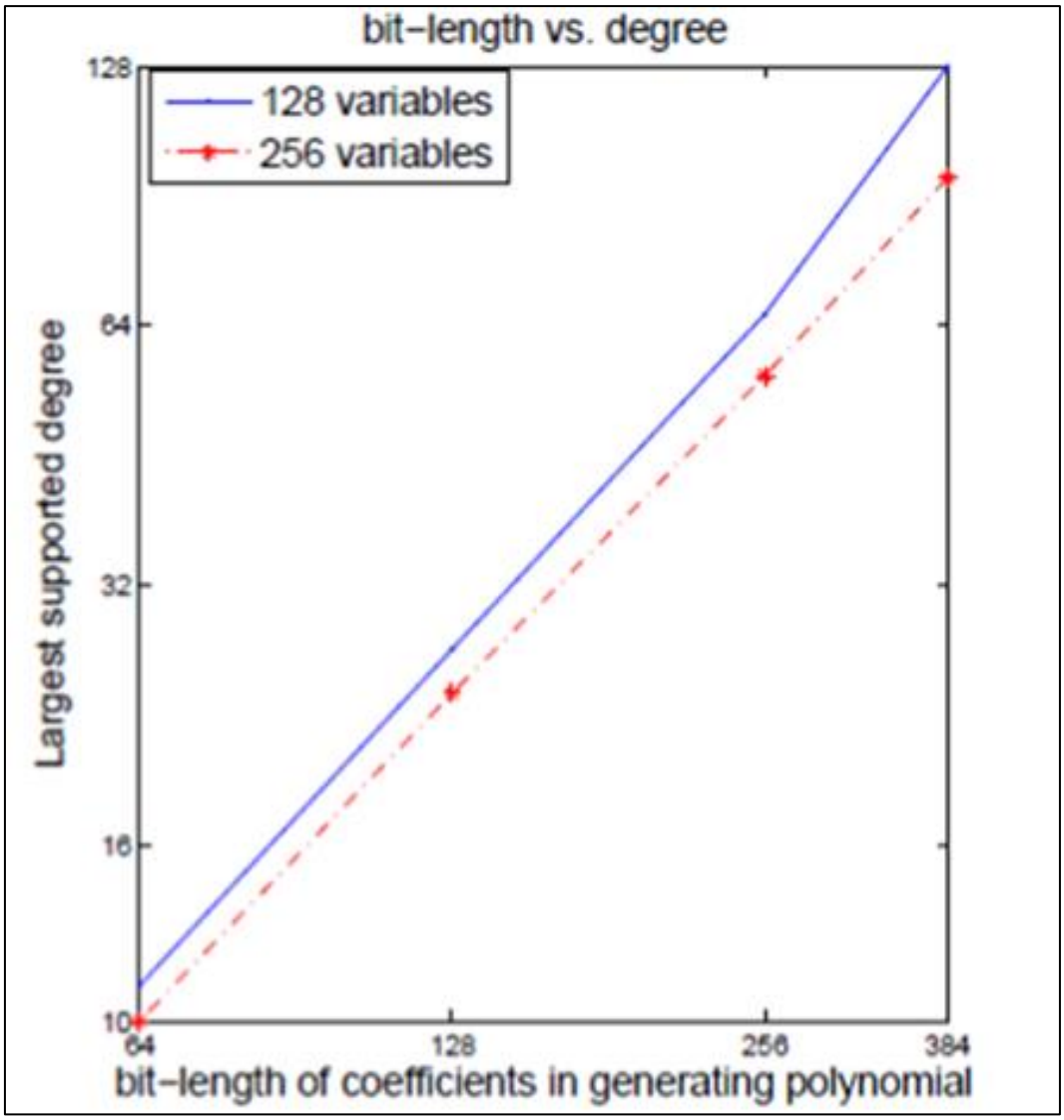

Figure 5.5. The relationship between bit length coefficient and largest supported degree

In the following table; we can see that the m and t values. Here, m values represent the number of variables while t equals bit length [43]. As it is obvious, $2 \cdot \sqrt{20} \approx 9$ will be the expected Euclidean length, unheeding of the dimension. This was selected by picking on each entry of the noise vector u as 0 with probability$(1 - 20/n)$, and as ±1 with probability 10/n each. So, the degree of polynomials are not dependable to dimension n.

To be sure, several dimensions from 128 to 2048 with a few settings of t and m are tested. It is seen that the largest supported degree was almost equal in all these dimensions. Due to the fact that, another settings are tested only in dimension n = 128. The cells which contains largest supported degree can be found in Figure 5.6.

t	m=64	m=96	m=128	m=192	m=256
t=64	13	12	11	11	10
t=128	33	28	27	26	24
t=256	64	76	66	58	56
t=384	64	96	128	100	95

Figure 5.6. Cells contain the LSD

The results can be seen in Figure 5.4. The largest supported degree increases linearly with the bit-length parameter t. Furthermore, it diminishes gradually with the number of variables.

It is also known that the more the variable the more the polynomial terms. The decryption radius of the secret key is roughly 2^t.

Also, the noise in ciphertext is approximately $c^{degree} \times \sqrt{}$number of-monomials. Here, c is not far from the Euclidean norm of fresh ciphertexts (i.e., $c \approx 9$) t has to be large enough so that $2^t \geq c^{degree} \times \sqrt{}(m\ deg)$.

Thus, the noise in the ciphertext will be within the radius of secret key.

For understanding the data from Figure 5.4 and 5.5 , we can see that c is not a constant, also it gets lesser when t gets larger. For understanding this better, Figure 5.6. can be examined. Also, the numbers can be examined.

t	c
64	[9.14, 11.33]
128	[7.36, 8.82]
256	[7.34, 7.92]
384	[6.88, 7.45]

Figure 5.7. Relationship between c and t values

As it can be seen from the Figure 5.7, the results are like below for t is 64,128,256 and 384. It is obvious that c values are decreasing while t is increasing.

- For t = 64
- c ∈ [9.14, 11.33],
- For t = 128
- c ∈ [7.36, 8.82],
- For t = 256
- c ∈ [7.34, 7.92], and
- For t = 384
- c ∈ [6.88, 7.45].

It can be said that this small deviation is related to the fact that the norm of the individual monomials is not completely c^{degree} .

Still, its distribution around that size. Moreover, the norm of the sum of all these monomials are different from somewhat from √number of-monomials times the estimated c^{degree}.

5.6. Final Say for Homomorphic Encryption Simulation Systems

Every homomorphic encryption system has to provide four steps as it is mentioned before. So, this steps has to be thought in the homomorphic encryption simulation systems, too**.**

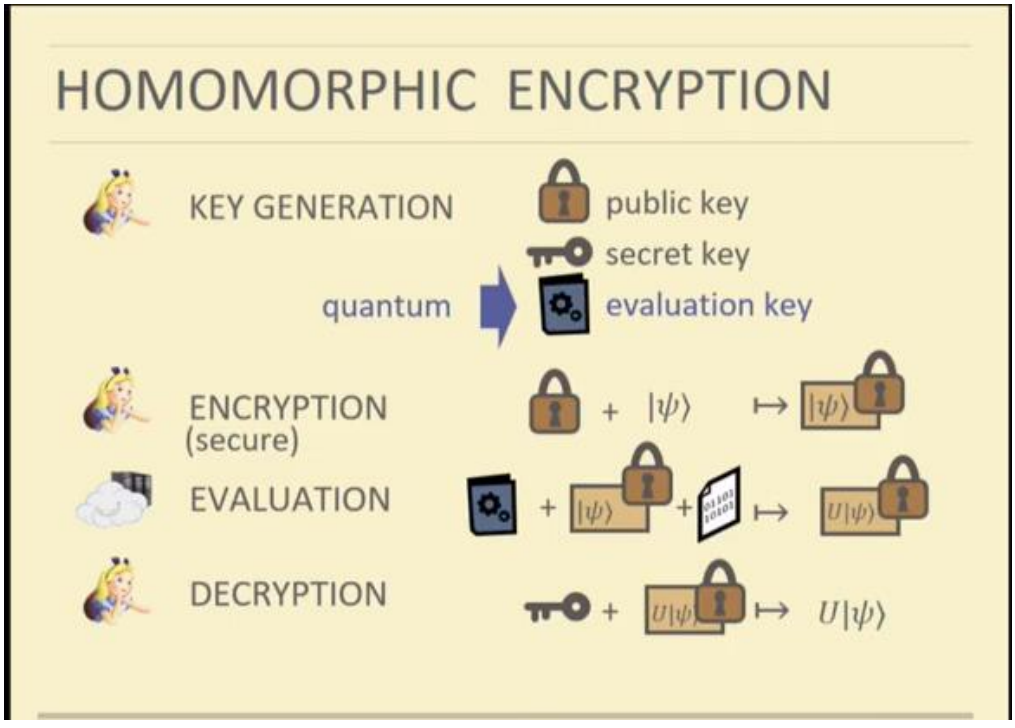

Figure 5.8. Steps for simulation systems which uses HE [44]

The relationship between bit length coefficient and largest supported degree is also very crucial for the simulation system. Because, it is the only way to see how many bits that our system will support. It should be noted that, noise is changeable [45].

6. CONCLUSION

In today's world, internet is in the center. The privacy of data has a more important role than ever before. People are using the internet almost for everything, for example online retail, e-banking, cloud servers. All those systems are critically sensitive. It is too important to protect users' assets or accounts from third parties.

Unfortunately, today's encryption systems are not protective as it has to be. These encryption systems; encrypt the data and share the keys with the service provider. Due to the fact that; the control on the privacy of the data is lost.

It means that the service providers who have the key have the same rights on the data as original users. This fact puts at risk the data. Because, even users' relationship ends with the service, the providers can continue to keep the data. This problem causes a question to protect the data. The answer can be homomorphic encryption (HE) schemes.

Homomorphic encryption (HE) is a method that encrypted data can be processed without being decrypted. A ciphered message is sent to a third-party, who can make a process on the taken data and sends back the final value. The first and real requester can decipher this final value to gain the result of their original query.

Simultaneously, the third-party who made an operation on data does not know what the query result is. Namely; HE is a new type of encryption scheme which lets any third party to make operations on the encrypted data without deciphering it in advance.

Actually, this idea came to the scene 30 years ago. Nevertheless; Craig Gentry was the first one who achieves the fully homomorphic encryption in 2009. There was seen many different schemes till today.

But, fully homomorphic encryption is still needed to be improved for being practical. In this research, starting from the basics of cryptology, the details of homomorphic encryption and a simulation of a somewhat homomorphic encryption system are presented.

In this research; simulation experiment and the simulation results are for somewhat homomorphic encryption scheme. The connection between the size of the plaintext and the

ciphertext is get. In the first simulation; the analysis is done for different plaintext size. This simulation analysis can be concluded with the following conclusions.

While the clear size is rising for the same security parameters but different input plaintext; moreover cipher text size is rising, too. But the growing is not linear and not high. For different input plaintexts when the security parameters are the same, cost will slowly incline the time needed for simulation.

It is obvious to see that; homomorphic encryption scheme can generate huge amounts of ciphertext. But, it also cost a lot of time. This problem will be worked on to improve efficiency.

For the further study; the aim is decreasing the cost of time and increasing the length of the keys. All in all, homomorphic encryption is a newly field which has a promising future. There will be more research and applications and it will get more and more practical.

REFERENCES

1. Widegren, E. (2018). *Fully Homomorphic Encryption: A Case Study*, Master's Thesis, University of Gothenburg, Gothenburg, Sweden, 5-8.

2. Sharma, H. (2011). *Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities and Countermeasures.* Paper presented at the International Journal of Computer Science and Management Studies, USA.

3. Sharma, H. (2011). *Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities and Countermeasures.* Paper presented at the International Journal of Computer Science and Management Studies, USA.

4. Sharma, H. (2011). *Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities and Countermeasures.* Paper presented at the International Journal of Computer Science and Management Studies, USA.

5. Widegren, E. (2018). *Fully Homomorphic Encryption: A Case Study*, Master's Thesis, University of Gothenburg, Gothenburg, Sweden, 10-11.

6. Gentry, C. (2009). *A fully homomorphic encryption scheme,* PhD Thesis, Stanford University, USA, 60-150.

7. Singh, K. (2017). *Performance evaluation of block cipher's for wireless sensor networks.* Paper presented at Advanced computing and communication technologies Conference, USA.

8. Singh, K. (2017). *Performance evaluation of block cipher's for wireless sensor networks.* Paper presented at Advanced computing and communication technologies Conference, USA.

9. Singh, K. (2017). *Performance evaluation of block cipher's for wireless sensor networks.* Paper presented at Advanced computing and communication technologies Conference, USA.

10. Singh, K. (2017). *Performance evaluation of block cipher's for wireless sensor networks.* Paper presented at Advanced computing and communication technologies Conference, USA.

11. Wagner, D. (2003,November). *Cryptanalysis of an algebraic privacy homomorphism.* Paper presented at the Information Security, Korea.

12. Gentry, C. (2009). *A fully homomorphic encryption scheme,* PhD Thesis, Stanford University, USA, 30-50.

13. Gentry, C. Sahai, A. and Waters B. (2013). *Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based*. Paper presented at the in Advances in Cryptology–CRYPTO 2013, CA,USA.

14. Menezes, A., Oorschot, P. and Vanstone, S. (2011*). Handbook of Applied Cryptography* (Fifth Edition). Canada; CRC Press, 6-30.

15. Brakerski, Z. and Perlman, R. (2016*). Lattice-based fully dynamic multi-key fhe with short ciphertexts*. Paper presented at the in Annual Cryptology Conference, Santa Barbara, USA.

16. Gentry, C. (2009). *A fully homomorphic encryption scheme*, PhD Thesis, Stanford University, USA, 90-98.

17. Widegren, E. (2018). *Fully Homomorphic Encryption: A Case Study*, Master's Thesis University of Gothenburg, Gothenburg, Sweden, 21-40.

18. Brakerski, Z. and Perlman, R. (2016*). Lattice-based fully dynamic multi-key fhe with short ciphertexts*. Paper presented at the in Annual Cryptology Conference, Santa Barbara, USA.

19. Gentry, C. Sahai, A. and Waters B. (2013). *Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based*. Paper presented at the in Advances in Cryptology–CRYPTO 2013, CA,USA.

20. Lyubashevsky, V. and Regev O. (2010, May). *On ideal lattices and learning with errors over rings.* Paper presented at Annual International Conference on the Theory and Applications of Cryptographic Techniques, USA.

21. Rivest, R, Shamir, A and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, no. 2, 120–126.

22. Tibouchi, M. (2014), *Arbitrary functions computed on encrypted data*. Paper present at Proceedings of the International Conference on Cloud Security, USA.

23. Menezes, A., Oorschot, P. and Vanstone, S. (2011*). Handbook of Applied Cryptography* (Fifth Edition). Canada: CRC Press, 35-54.

24. Tibouchi, M. (2014), *Arbitrary functions computed on encrypted data*. Paper present at Proceedings of the International Conference on Cloud Security, USA.

25. Persichetti, E. (2012). *Improving the Efficiency of Code-Based Cryptography*, PhD Thesis, University of Auckland, Auckland, New Zealand, 20-80.

26. Kessler, G. (1998,). An Overview of Cryptography. In J. P. Slone (Eds.), *Handbook on Local Area Networks.* Auerbach Publications, US; 400-430.

27. Paillier, P. (1999). *Public-key cryptosystems based on composite degree residuosity classes*. Paper presented in International Conference on the Theory and Applications of Cryptographic Techniques, Prague, Czech Republic.

28. Roe, M. (2010). *Cryptography and Evidence*, PhD Thesis, University of Cambridge, Cambridge, England, 50-70.

29. Sekar, G. (2011). *Cryptanalysis and Design of Symmetric Cryptographic Algorithms*, PhD Thesis, Katholieke Universitet Leuven, Leuven, Belgium, 50-90.

30. Persichetti, E. (2012). *Improving the Efficiency of Code-Based Cryptography*, PhD Thesis, University of Auckland, Auckland, New Zealand, 60-90.

31. Widegren, E. (2018). *Fully Homomorphic Encryption: A Case Study*, Master's Thesis University of Gothenburg, Gothenburg, Sweden, 36-43.

32. Menezes, A., Oorschot, P. and Vanstone, S. (2011*). Handbook of Applied Cryptography* (Fifth Edition).Canada; CRC Press, 100-120.

33. Internet: Barthelemy, L. (June,2016). A brief survey of Fully Homomorphic Encryption, computing on encrypted data. Web: https://blog.quarkslab.com/a-brief-survey-of-fully-homomorphic-encryption-computing-on-encrypted-data.html.Taken at 23 May 2019.

34. Sekar, G. (2011). *Cryptanalysis and Design of Symmetric Cryptographic Algorithms*, PhD Thesis, Katholieke Universitet Leuven, Leuven, Belgium, 60-100.

35. Persichetti, E. (2012). *Improving the Efficiency of Code-Based Cryptography*, PhD Thesis, University of Auckland, Auckland, New Zealand, 80-110.

36. Roe, M. (2010). *Cryptography and Evidence*, PhD Thesis, University of Cambridge, Cambridge, England, 20-50.

37. Kessler, G. (1998,). An Overview of Cryptography. In J. P. Slone (Eds.), *Handbook on Local Area Networks.* Auerbach Publications, US; 380-450.

38. Brakerski, Z. and Perlman, R. (2016*). Lattice-based fully dynamic multi-key fhe with short ciphertexts*. Paper presented at the in Annual Cryptology Conference, Santa Barbara, USA.

39. Rivest, R, Shamir, A and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, no. 2, 122–124.

40. Gentry, C.(2009). *A fully homomorphic encryption scheme*, PhD Thesis, Stanford University, USA, 151-157.

41. Sekar, G. (2011). *Cryptanalysis and Design of Symmetric Cryptographic Algorithms*, PhD Thesis, Katholieke Universitet Leuven, Leuven, Belgium,100-150.

42. Roe, M. (2010). *Cryptography and Evidence*, PhD Thesis, University of Cambridge, Cambridge, England, 21-32.

43. Gentry, C. (2009). *A fully homomorphic encryption scheme*, PhD Thesis, Stanford University, USA, 160-191.

44. Speelman, F. (2017). *Quantum homomorphic encryption for polynomial-sized circuits.* Paper presented at the 23rd International Conference on the Theory and Applications of Cryptology Conference, USA.

45. Gentry, C. (2009). *A fully homomorphic encryption scheme*, PhD Thesis, Stanford University, USA, 180-211.

CURRICULUM VITAE

Personal Information

Surname, Name	: BOZDUMAN, Hanife Çağıl
Nationality	: T.C.
Date and Place of Birth	: 24.10.1989, Çanakkale
Marital status	: Single
e-mail	: bozduman2425@gmail.com

Education

Degree	School/ Program	Graduation Date
MSc	Gazi University / EE Engineering	Ongoing
Undergraduate	Anadolu University / E&E Engineering	2014
High School	Çorum Anadolu Öğretmen Lisesi	2008

Professional Experience

Year	Place of Work	Position
2018-Ongoing	OYAK-RENAULT Vehicle Factory	Process Engineer
2015-2018	IPA DEFENCE	R&D Engineer

Foreign Language

English

Publications

1. Bozduman; H.Ç.(2014). *Wind Energy Storage Methods*. Paper presented at Turkish-German Conference on Energy Technologies 2014, Ankara, Turkey

2. Bozduman; H.Ç.(2019). *Simulation of a Homomorphic Encryption.* 4th International Conference on Computational Mathematics and Engineering Sciences-CMES 2019,Antalya:Turkey

Hobbies

Travelling, Writing,

GAZİ GELECEKTİR…