CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

A modified version of Secret Sharing Scheme with general access structure based on Elliptic curve and pairing

by

Syed Burhan

A thesis submitted in partial fulfillment for the degree of Master of Philosophy

in the

Faculty of Computing

Department of Mathematics

2018

Dedicated to my beloved parents

CAPITAL UNIVERSITY OF SCIENCE & TECHNOLOGY
ISLAMABAD

CERTIFICATE OF APPROVAL

A modified version of Secret Sharing Scheme with general access structure based on Elliptic curve and pairing

by

Syed Burhan

MMT161008

THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Organization
(a)	External Examiner	Dr. Ayesha Rafiq	IST, Islamabad
(b)	Internal Examiner	Dr. Dur e Shehwar Sagheer	CUST, Islamabad
(c)	Supervisor	Dr. Rashid Ali	CUST, Islamabad

Supervisor Name
Dr. Rashid Ali
October, 2018

Dr. Muhammad Sagheer
Head
Dept. of Mathematics
October, 2018

Dr. Muhammad Abdul Qadir
Dean
Faculty of Computing
October, 2018

Author's Declaration

I, **Syed Burhan** hereby state that my MS thesis titled "**A modified version of Secret Sharing Scheme with general access structure based on Elliptic curve and pairing**" is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/abroad.

At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my M. Phil Degree.

(Syed Burhan)

Registration No: MMT161008

Plagiarism Undertaking

I solemnly declare that research work presented in this thesis titled "*A modified version of Secret Sharing Scheme with general access structure based on Elliptic curve and pairing*" is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been dully acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of M. Phil Degree, the University reserves the right to withdraw/revoke my M. Phil degree and that HEC and the University have the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized work.

(Syed Burhan)

Registration No: MMT161008

Acknowledgements

All praise to Almighty Allah, the most Benevolent and Merciful, the Creator of universe and man, who gave me the vision and courage to accomplish this work successfully. A research project at any level is very difficult to be accomplished alone. The contributions of many people have made it possible for me to complete this work. I would like to extend my appreciation especially to the following.

First and foremost I extend my sincerest gratitude to my thesis supervisor Dr. Rashid Ali who has supported me throughout my thesis work in every possible way. He presented the things in simplest way. His efforts and encouragement is really appreciable. He gave me self-belief, and confidence and provided me his support and guidance all way along. Under the supervision of him I never felt anything difficult in my thesis. Dr. Rashid Ali provided me the friendly and comfortable environment to work in, without his guidance and support I would not have been able to put this topic together.

I am also thankful to all the faculty of CUST in particular the faculty of Mathematics department. Their ability to simplify the most difficult things helped me to understand the things that I would not have been able to understand.

The research oriented environment provided to students by Mian Amer Mehmood, Chancellor at CUST, Prof. Dr. Muhammad Mansoor Ahmed, Vice Chancellor at CUST and Dr.Muhammad Sagheer, Head of Department of Mathematics made the research work easier and more pleasant. It is their positive attitude that enables such excellent research work from staff and students.

My friends, class mates and fellow research workers Tahir Bhai, Malik Zia, Suleman Liaquat, Mujtaba Azim, Sohail Abid and Sultan Mehmood also helped me a lot and guided me whenever I needed it.

Last but not least, I would like to pay high regards to my parents and all family members for their sincere encouragement and inspiration throughout my research work and lifting me uphill this phase of life. I owe everything to them. Besides this, I would also like to thank all of my friends who supported me in writing, and incented me to strive towards my goal.

Abstract

Secret sharing scheme is a way to share a secret with 'n' participants and then a setup is made for predetermined $t \leq n$ or more number of participants who must contribute to reveal the secret. In secret sharing schemes 't' is known as a threshold which must be achieved for secret reconstruction. In this thesis, a secret sharing scheme with general access structure based on elliptic curve and pairing is analyzed. Elliptic curve makes secret sharing more secure with less computational complexity. Bilinear pairing is used in the reconstruction phase of the secret for verifying the shares delivered by the participants. It has been observed that Sreekumar and Binu's scheme [2] does not provide security to shares when they are delivered for reconstructing the secret. Participants have to deliver their shares to combiner via a secure channel. Participants can not verify combiner and hence increases the chance of being attacked. Also, the reconstructed secret can not be verified by the participants. We have used the idea of public key cryptography in the reconstruction phase through which participants can authenticate combiner and avoid being attacked by an attacker. This also addresses the problem of using a secure channel. Moreover, the idea of hash function is used for verification of the reconstructed secret to avoid being deceived by a dishonest combiner. The revised scheme is more secure as compared to [2] and addresses the weak security points highlighted in the analysis.

Contents

List of Figures

List of Tables

Abbreviations

DES	Data Encryption Standard
AES	Advanced Encryption Standard
RSA	Rivest Shamir and Adelman
DLP	Discrete Logarithm Problem
ECDLP	Elliptic Curve Discrete Logarithm Problem
RC4	Rivest Cipher 4
GF(q)	Galois Field or Finite Field
ECC	Elliptic Curve Cryptography
SHA	Secure Hash Algorithm
MD5	Message Digest 5
SSS	Secret Sharing Scheme
SSSS	Shamir's Secret Sharing Scheme

Chapter 1

Introduction

The meaning of the word cryptography is to communicate securely. It is one of the two branches of cryptology [30]. To achieve the desired security extent, there have been different algorithms introduced over the years. There exist many cases in the literature where secret communication was required, for example, in military, diplomatic service, government in general. But its main use desperately came into consideration to protect the information in digital form when people started communicating over a network. It allowed people to communicate with each other in a secure way even in the presence of an adversary, without compromising the security of their information or message (usually in hidden or encrypted form). But, then the second branch of cryptology known as cryptanalysis which works to break or check the security strength of cryptographic system in order to obtain some information or original message from the encrypted message. Cryptanalysis is also important for analyzing the security of a cryptosystem. There are many attacks available in the literature on cryptographic systems for example brute-force attack. In this attack, the attacker tries every possible key to break the system. We will see different types of attacks on cryptosystems in Chapter 2 of this thesis. There are many cryptographic techniques presented to secure the data but when found weak are replaced by some other improved technique.

Next, we will discuss some techniques which are available in the literature and have been useful for many years. The most worldwide well-known cryptographic

mechanism in literature is DES (Data Encryption Standard) [15], published in 1977. It is a block cipher in which data undergoes 16 stages. It was used for a long time but then found insecure and replaced by AES (Advanced Encryption Standard) [36]. AES was considered as a more secure algorithm than DES. It has three different number of rounds in which information undergoes depending on the key size. Data is encrypted in blocks and in each round, message block goes through four different layers. So far, the schemes that have been mentioned use same key for data encryption and decryption.
Basically, there are two types of cryptographic methods based on keys used in them.

1. Symmetric key cryptosysytems
2. Asymmetric key cryptosysytems

In symmetric key cryptosystems, both parties (sender and receiver) agree to use the same key for encryption and decryption. The main disadvantage of symmetric key cryptosystems is that the sender after encryption has to share the secret key with the receiver which then enables them to decrypt the message. It sometimes compromises the security of the system. To overcome this problem, a new technique known as asymmetric key cryptosystem [5] was proposed in 1976 by Whitfield Diffie and Martin Hellman. It uses two different keys, one for encryption and the other for decryption. The encryption key is public and can be used by anyone and decryption is kept secret that is only known to its owner. Examples of asymmetric key crytposystems are RSA [33], ElGamal [7], Elliptic curve cryptosystems [18] etc. Here, an obvious question arises that how are asymmetric systems made? They are built from one common principle, the one-way function. In mathematics, one-way functions are those functions that can be computed easily in one direction but they are hard to compute in the other direction unless a special information called **trapdoor** is known. The two commonly used one-way functions in public key cryptography are integer factorization problem and Discrete Logarithm Problem (DLP) [29]. RSA is based on integer factorization problem. We will discuss Discrete logarithm problem in Chapter 2.

For the security of information we encrypt the information using some cryptographic method in which we use encryption and decryption keys. It suggests that information security relies highly on these cryptographic keys and securing them is also necessary as far as the protection of information is concerned. To address this issue, Adi Shamir [34] and George Blakley [3] independently invented a secret sharing scheme in 1979.

1.1 Secret sharing

In secret sharing scheme, an information is divided into n number of shares and when specified conditions meet, original information is revealed. This idea helped to secure the key because instead of storing it at one place, we can store it in parts(usually called shares) and when the specified number of shares are combined the key can be obtained. Shamir [34] and Blakely [3] gave the idea of secret sharing scheme independently. Shamir's scheme is based on polynomial interpolation and Blakley's scheme is based on geometry.
Later, many researchers have proposed and investigated new secret sharing schemes. Some of them are described in the next section.

1.2 A look into the literature

A scheme based on general access structure was proposed by Ito, Saito and Nishizeki [17] in 1989. In their scheme, they discussed the general access structure that was difficult to realize in Shamir's scheme [34]. They presented a scheme in which any access structure can be realized. The access structure basically differentiates between authorized and unauthorized sets. An authorized set is a set of participants who can recover the secret by joining their shares and an unauthorized set is the set of participants who cannot get any clue about the secret by pooling their share together. As in (t, n)-threshold scheme, t or more participants can get the secret back by combining their shares. So, it intuitively means that if a set is an authorized set then any superset of it will always be an authorized set. In short, an access structure can be made by finding the minimal set that is eligible to recover the secret and this is the monotone property of the access structure. Similarly, if a set is an unauthorized set then any subset of it will also be unauthorized set. In the scheme proposed by Ito et. al. [17], they implemented the generalized access structure in secret sharing scheme by giving some information related to the shares to participants which enabled the efficient and easy implementation of general access structure in secret sharing scheme. Tompa and Woll in [38] proposed a scheme in 1989 which is cheating resistant. They first showed that Shamir's scheme is less secure and then modified it to provide a better security against the cheaters. There are also schemes which are based on signature to identify the cheaters like in [31]. In these schemes, dealer has to first sign the shares before sending them to the respective participant so that the cheating in shares is avoided. In 1988, a scheme for the verification of the secret was proposed by Ben-or et. al. [1]. Another scheme to identify the cheaters is proposed by Harn and Lin in 2009 [13]. Stadler in [35] worked to propose a publicly verifiable scheme in which shares submitted by the dealer can be verified publicly by anyone. Through this approach, not only the participant can check that the shares are consistent and valid or not but anyone can check the validity and consistency of the shares given by dealer. Secret sharing schemes have been extensively studied in the literature and people worked on them to improve them in every aspect of security and computation.

Many researchers have worked on a new idea of sharing multiple secrets. In multi secret sharing many secrets are shared in such a way that they are distributed among the participants and when the authorized set of participants gather their shares to reconstruct the secrets, the secrets are revealed. He and Dawson [14], in 1994, proposed a multi stage secret sharing scheme that reconstructs several secrets stage by stage. The beauty of this scheme is that the participants do not have to store many shares for several secrets. They used the idea of one way function so that only a single share can be made useful for getting all the secrets without revealing the shares to anyone. They added shift values to participants secret shares to obtain a true share and then the true shares are used instead of secret share to reveal the secret but secret share are kept hidden in order to use them several times. This enabled participants to use only a single share for revealing several secrets. For the practical implementation of these schemes, many schemes use a public bulletin board. On this public bulletin board only the dealer can add or delete data. Dealer publishes all the public parameters that are required to properly run the scheme on the bulletin board so that the participants can access the information from there. There are different multi secret sharing schemes proposed in [9, 19, 40]. So far, the schemes mentioned above are highly reliable on the dealer. Dealer sends the secret shares of the participants to them. It can be risky to use the secret shares again for the reconstruction of some other secret because it is already known to the dealer. In 2006 Pang et. al. [20] used Shamir's scheme and Discrete Logarithm Problem (DLP) to present a new scheme that is independent from getting the shares from dealer. They used the term pseudo share, pseudo share is generated by the participant's private share through some method. Participants can select their shares themselves and then send the pseudo shares to the dealer. If someone wants to obtain the secret share from pseudo share they have to face the difficulty of DLP. It was a great breakthrough in secret sharing schemes because participants are no more dependent on the dealer to select their shares. Every participant can now select their share themselves. In 2008, Zhong et. al. [39] proposed almost a same kind of scheme that was proposed by Yumin et.al. but Zhong et. al. used first degree polynomial instead of using $(t-1)$ degree polynomial. It was the first time when someone had changed the degree of basic Shamir's scheme otherwise all the scheme proposed before it, used $(t-1)$ degree polynomial. It reduced the computational cost as only has to solve first degree Lagrange interpolation in order to recover the secret. It also reduced the storage cost. The security level of the scheme was same as of Shamir's scheme. There were two drawbacks of this scheme. First one is in the phase of share generation where scheme does not allow participants to generate their shares themselves and the second is in the reconstruction phase where there is no criterion that checks the validity of the shares sent by the participants and verify that the shares are correct. Although it improved secret sharing scheme in many aspects but at the same time lacked in some security aspects.

As discussed earlier, due to the invention of elliptic curve, a great change has been occurred in cryptography. It not only reduced the computational complexity but also provided more security to cryptographic applications. Recently, it has also found applications in secret sharing schemes. Elliptic curve is introduced in secret sharing so that DLP is made more harder to solve and it also reduced the computational complexity. It was first introduced in secret sharing scheme by Liu et. al. [22] in 2008. They implemented the idea of elliptic curve in secret sharing in which they used points on elliptic curve to represent multiple secrets using self pairing. Hua and Aimin [16] in 2010 proposed a scheme which uses points of elliptic curve to share a secret and in their scheme, shares are not generated by the dealer instead they are selected by the participants. This scheme does not require any private communication between the participants and the dealer in the secret distribution phase. For reconstruction of secret, participants have to combine their shares. This scheme has good property that the participants are no more bound to reveal their secret share for reconstruction of the secret instead they revealed pseudo shares for reconstructing the secret. This property allowed the users to use secret shares multiple times. If someone wants to find the secret share from pseudo share they have to solve the ECDLP which is assumed to be a hard problem. A field of atleast 160 bits must be used in order to avoid any attack on elliptic curve. Hence, it improves the security level and is efficient to use. Further, multiple secrets can be shared and participants do not need to change their secret shadow for different secrets to be shared.

The idea of pairing was introduced by Meneze's et. al. [23]. Pairing in cryptography is a helpful tool to check an attack on elliptic curve discrete logarithm problem. To improve the security level in the secret distribution phase many scheme made the participants free for getting the shares from the dealer. So that an untrusted dealer does not send inconsistent shares and also to use the shares several times without compromising the security. But, in the reconstruction phase when participants send their shares to the dealer or combine their shares there is a heavy risk of getting inconsistent or invalid shares from the participants. There had to be a way out of this issue to enhance the security and make secret sharing scheme more efficient. Pairing is used to overcome this problem because paring has a good property to verify the shares which were initially selected and later delivered for the reconstruction are same or not. If there is a mismatch between them then combiner asks to give the right share or can make sure that whether a participant is valid or is it an intruder.

Das and Adhikari [4] in 2010 propsed a secret sharing scheme that is based on a collision resistant hash function. Their scheme has a great influence because it does not use any polynomial and hence there is no need to use Lagrange interpolation for revealing the secret. It works only using a collision resistant hash function and "XOR" operation to reveal the secret. What is more in their scheme is that the participants can also check the secret revealed by the combiner is the actual secret or not, by just applying the hash function on the secret. This scheme is also a multi secret sharing scheme and as it does not require exponentiation, modular multiplication and inversion so it is more efficient and also saves space and time.

Recently, Sreekumar and Binu proposed a multi secret sharing scheme based on elliptic curve and pairing [2]. This scheme is a multi secret sharing scheme that uses a general access structure. Elliptic curve made their scheme more secure and also reduced the computational cost. They used only first degree polynomial that made secret sharing more simpler and efficient to implement. The idea of bilinear pairing helps the combiner to verify shares of participants in the reconstruction phase. In this thesis, our goal is to analyze the security of the proposed scheme [2] and present a way out. During the analysis, we found a couple of weak points in the security of the scheme, which are as follows:

- The proposed scheme does not make participants authenticate the combiner when they deliver the shares for reconstructing the secret.

- The proposed scheme also does not make sure that the reconstructed secret given by the combiner is correct or not.

We also proposed a method to improve the security of the proposed scheme [2] in this thesis.

If we consider cryptographic technique as a building then the blocks to construct that building are the basic terminologies that are necessary for the best understanding of any cryptographic technique. They are given in Chapter 2 along with the idea of secret sharing. In Chapter 3, we will be discussing a secret sharing scheme [2] which uses a general access structure based on Elliptic curve and pairing.

It has been observed that the scheme [2] is less secure in the secret reconstruction phase where participants submit their shares to the combiner using a secure channel. Here, participants can not authenticate the combiner. Due to which an attacker can get the benefit and attack the secret by collecting participants shares. Another flaw in the scheme is that the validity of revealed secret can not be checked. A method to tackle the challenges in the security of the scheme [2] is mentioned in Chapter 4.

Chapter 2

Preliminaries

In this chapter, we will first discuss some basics of cryptography along with some of its applications. Then we will move to all the necessary ingredients that will be helpful for the best understanding of the proposed scheme [2]. After that secret sharing scheme and its properties will be studied in detail so that a reader can understand the use of secret sharing scheme in cryptography. We will also highlight some of the secret sharing schemes that are available in the literature.

2.1 Cryptography

Cryptography is the art of hiding the original information in some encrypted/coded form in the presence of an adversary. It is used to hide the original information into a coded form so that it cannot be read by anyone who is not intended to read it. In cryptography, our ultimate focus is to make a system that encrypts and decrypts the data and such a system is known as cryptosystem.

In cryptography, for our best understanding we usually name the two parties who share information with each other as Alice and Bob. There are some technical terms that are used in cryptography when Alice and Bob want to communicate securely over a public network. The original message that is to be sent by Alice to Bob is called **plaintext**. Plaintext is not sent in its original form rather it is

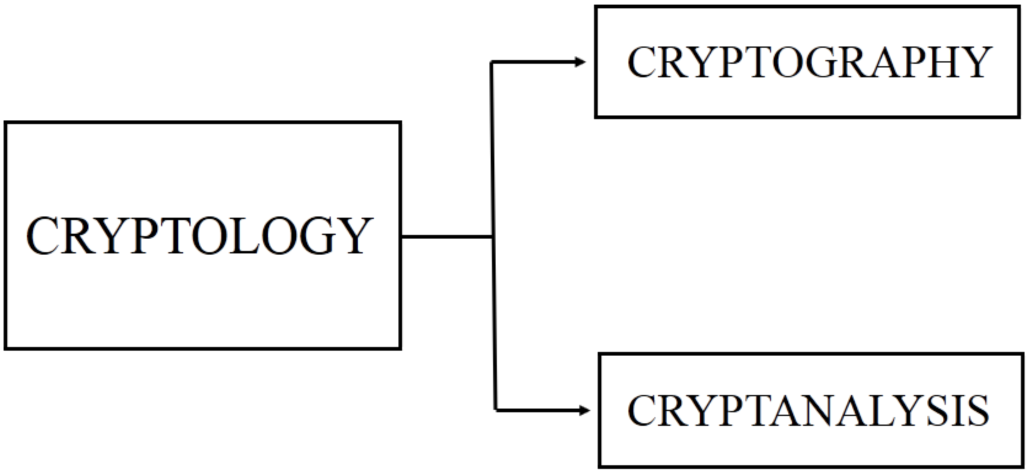

Figure 2.1: Cryptology diagram

first altered into a form that can not be read and then it is sent to the intended receiver. The coded message which can not be understood by anyone is called **ciphertext**. Obviously, there is an algorithm that is used to alter the plaintext into ciphertext such an algorithm is known as **encryption algorithm**. A ciphertext can not be understood as long as it is transformed back into the plaintext and the algorithm that is used to get the plaintext from ciphertext is called **decryption algorithm**. There is a highly sensitive information used in encryption and decryption algorithm for conversion of plaintext and ciphertext, called **key**. The key should be kept hidden throughout in the communication because the security of a secure communication completely depends on it.
Following are some of the applications of cryptography [24].

Confidentiality

Confidentiality refers to a state where the information that has been sent by a sender can only be understood by intended receiver. Suppose Alice sends some encrypted information to Bob. If an adversary, say, George obtains the encrypted information then he can not understand it. It can be said that confidentiality keeps the original information secret.

Integrity

Integrity ensures that there is no change occurred in the original data during the storage or transmission to the designated receiver. For example, if Alice sends some encrypted information to Bob then integrity assures that the information is not altered after sending and helps Bob to believe that the information is in its original form.

Authentication

It enables the sender and receiver to confirm each other's identity. Let us consider the same scenario where Alice and Bob makes a setup to communicate securely. For secure communication, Alice and Bob must be able to verify each others identity. This property helps participants to make sure that the other participants are valid and not any attacker.

Non-Repudiation

Non-repudiation means that the originator of the information cannot refuse at a later stage. If the information is sent from Bob, there is no way out for Bob to deny it at any later stage. This helps the receiver to gain the trust of sender in any cryptographic protocol.

These applications establish a strong and secure communication protocol in cryptography that is more reliable and practical, and hence provides a better platform that fulfills the basic necessities of a secure communication.

There are two types of cryptographic methods as shown in Figure 2.2. They are differentiated on the basis of keys used in them.

1. Symmetric key cryptography

2. Asymmetric key cryptography

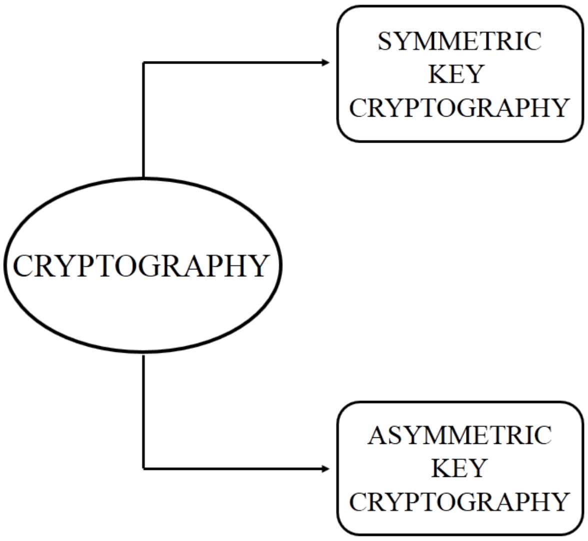

FIGURE 2.2: Cryptographic protocols

2.1.1 Symmetric key cryptography

Symmetric key cryptography [37] is also known as secret key cryptography. In this method, two parties use a single key for the encryption and decryption of the data. Here single key is in a sense that either the keys for encryption and decryption are same or there must be an easy way to obtain the decryption key from encryption key. This was the only method which was used for secure communication until 1976. The two parties who communicate with each other share the secret key through a secure channel [28]. Next, we will see the working of symmetric key cryptography as shown in Figure 2.3. Let the two parties be Alice and Bob. Alice and Bob want to communicate with each other on an insecure channel e.g. Internet. First, Alice uses the secret key to encrypt the information which she wants to share with Bob using an encryption algorithm. After that she sends the encrypted message to Bob over an insecure channel and also sends the key to Bob through a secure channel. When Bob receives the key he can easily decrypt the message using the decryption algorithm. In this method, first party after encrypting the message must send the encryption key to the other party otherwise the ciphertext can not be deciphered. This requires the use of a secure channel to send the key to the designated receiver. The use of secure channel is must, else the higher security

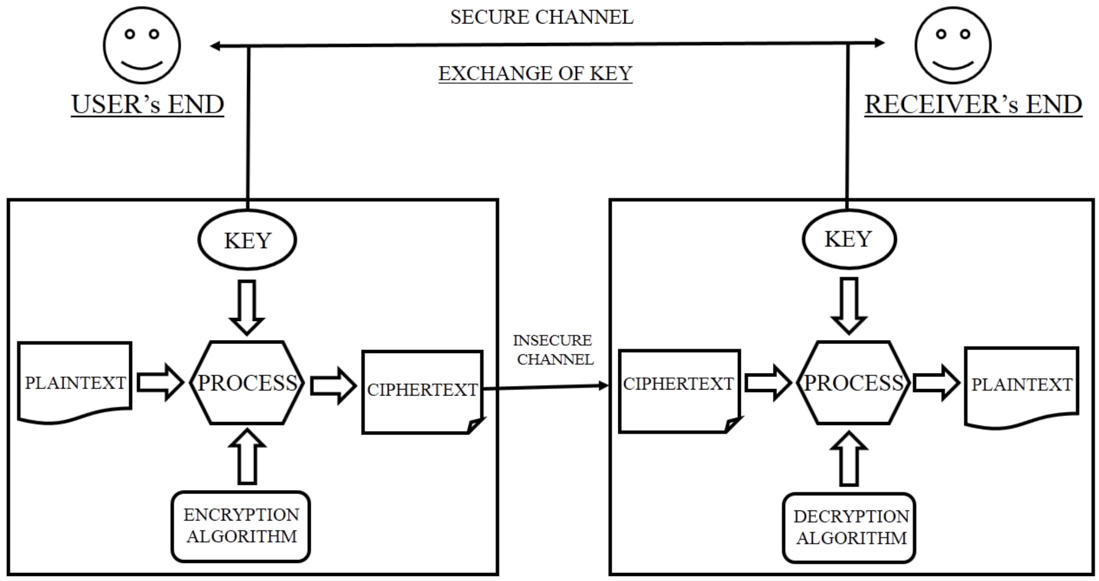

Figure 2.3: Symmetric Key Protcol

can not be achieved. Examples of symmetric key cryptography are AES [36], DES [15], 3DES [30], RC4 [27] etc.

It appears that when two parties share the same key the security can be compromised. There are basically two issues with symmetric key cryptography; one is key management and the other is security.

2.1.2 Asymmetric key cryptography

As mentioned above that symmetric key cryptography had some drawbacks. Due to which there was a need of an improved method which solves the issues of sharing a key. To overcome this problem, Whitfield Diffie and Martin Hellman in 1976 gave the idea of public key cryptography also known as asymmetric key cryptography [5]. This method is based on one way trapdoor function which are easy to compute in one direction but can not be inverted back unless a special information called trapdoor is known. In this approach, two different keys are used; one for encryption and the other for decryption. Encryption key is known as public key and decryption key is called private key which is kept secret. Anyone can use public key to send the message to someone but to decrypt the message only private key can be used which is only known to the owner of it. This way

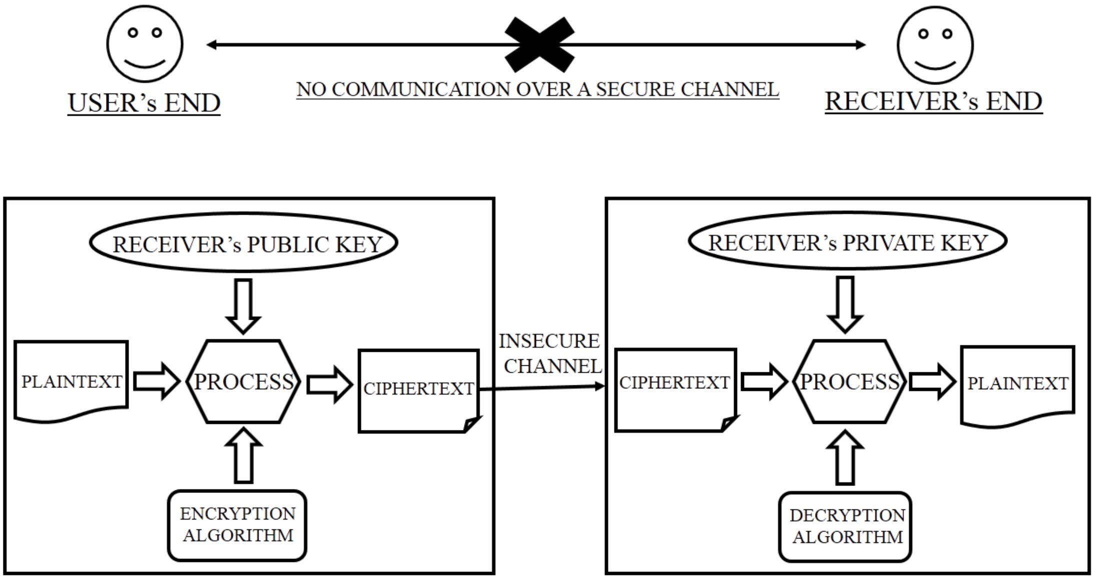

FIGURE 2.4: Asymmetric Key Protcol

public key cryptography addresses the issues of secret key cryptography that can be seen in Figure 2.4. The use of secure channel to share the decryption key was no more required and hence improved the security. Anyone who is the owner of the private key can now make sure that they have the full authority to decipher the message. Examples of such a system are Diffie-Hellman key exchange protocol [5], ElGamal [7], RSA [33].

2.2 Cryptananlysis

The method which is used to obtain the plaintext from ciphertext without having a key or obtaining a key from ciphertext is called cryptanalysis [37]. Cryptanalyst is someone who does this job. It can be said that a cryptanalyst does cryptanalysis on a cryptographic protocol by finding out some weaknesses in one of the following four properties; confidentiality, integrity, authenticity and non-repudiation. If either one of then is found weak then the security of the communication is not strong and can be attacked successfully. Cryptanalysis is used when someone wants to attack a secret communication or to check how much secure a cryptographic system is. There are many types of cryptographic attacks [37] available in the literature, some of them are as follows:

1. **Ciphertext only attacks**

   In this kind of attack, cryptanalyst either try to reveal a meaningful message (original message) from ciphertext or work for obtaining the key. He originally does not know any thing about the plaintext but he uses ciphertext to attack the original message. Frequency of the letters is also analyzed to attack the message.

2. **Known plaintext attacks**

   This is the attack performed by a cryptanalyst when he knows some ciphertext and their corresponding plaintext. On the basis of the previously known information he either tries to recover the key or makes an efficient algorithm to decrypt any further ciphertext.

3. **Chosen plaintext attacks**

   Chosen plaintext attack is the one in which attacker can get the cipher text for the arbitrarily chosen plaintext. Using these plaintext and ciphertext, he tries to recover the key.

4. **Chosen ciphertext attacks**

   In chosen ciphertext attack, the attacker gets the plaintext of the chosen ciphertext and then using these results he tries to figure out the secret key or tries to obtain as much information as he can regarding the key.

5. **Man-in-the-middle attacks**

   In man-in-the-middle attack, the attacker sits between two parties who are secretly communicating with each other and gets control over communication at both ends that is sender and receiver ends. To make this kind of attack possible, the attacker first choses two secret keys. After choosing the keys, he starts the communication with first party using the first key and when he gets the reply in encrypted form he can easily decrypt the message since the key is known to him. Then, he encrypts the received message again using the second key and sends this message to second party and when he gets the reply from second party he can decrypt it using the key. This is how this attack works and the attacker gets control over the communication between two parties by sitting between them.

6. **Brute force attacks**

   In this attack, the attacker tries every possible key in order to guess the original plaintext from ciphertext. With bigger key space, this attack can be made hard.

2.3 Mathematical background

Next, we see some basic ideas from mathematics which are helpful to understand the rest of the chapters presented in this thesis.

Definition 2.3.1. (Groups)

A non-empty set G is called a group [21] under a binary operation '$*$', if for any three elements $a, b, c \in G$, the following axioms are satisfied:

1. **Closure law**: $a * b \in G$
2. **Associativity**: $(a * b) * c = a * (b * c)$
3. **Identity element**: There is an identity element $e \in G$ such that $a * e = e * a = a$, for all $a \in G$
4. **Inverse element**: For all $a \in G$, there exists an element $a' \in G$ such that

   $$a * a' = a' * a = e,$$

   then a' is called the inverse of a.

Definition 2.3.2. (Abelian group)

If further the group G verifies

$$a * b = b * a$$

for all $a, b \in G$ then G is called an **abelian group**. The set of real numbers $\mathbb{R}$ and set of integers $\mathbb{Z}$ are the examples of abelian group with respect to addition. The set of real numbers $\mathbb{R} \setminus \{0\}$ is an example of an abelian group with respect to multiplication.

Definition 2.3.3. (Cyclic group)
A group G is called a cyclic group, if it can be generated by a single element $g \in G$. This g is called the generator of G and notation for it is $\langle g \rangle$. Thus every $a \in G$ has the form g^r for some integer r. Further, every cyclic group is an abelian group.
Let G be a cyclic group and $a, b \in G$ then $a = g^r$ and $b = g^s$
$\Rightarrow ab = g^r g^s = g^{r+s} = g^{s+r} = g^s g^r = ba$
The set of integers $\mathbb{Z}$ is cyclic under addition with 1 or -1 as generators of it.

Definition 2.3.4. (Ring)
A non-empty set R together with two algebraic operations '+' and '.', is called a ring [21]. If for all $a, b, c \in R$, the following conditions hold:

1. **Abelian**: R is an abelian group under addition.

2. **Associativity**: $(a.b).c = a.(b.c)$

3. **Distributivity**: R holds both right and left distributive laws:

$$(b + c).a = b.a + c.a$$

$$a.(b + c) = a.b + a.c \quad .$$

Definition 2.3.5. (Commutative ring)
The ring R is said to be a commutative ring, if it satisfies:

$$a.b = b.a \quad ; \quad \forall \quad a, b \in \mathbb{R}$$

Examples of commutative ring are $(\mathbb{Z}, +, .), (\mathbb{R}, +, .)$ and $(\mathbb{Z}_n, +, .)$.

Definition 2.3.6. (Field)

A commutative ring $\mathbb{F}$ is called a field [21], if the non-zero elements of $\mathbb{F}$ form a group under multiplication. Examples of field include $\mathbb{R}$, $\mathbb{Q}$ and $\mathbb{C}$.

Definition 2.3.7. (Order of a field)

The number of elements that a finite field contains is called its **order**. Further, a field with finite number of elements is called a finite field.

Definition 2.3.8. (Galois field)

A finite field of order q is also called Galois field [21]. It is denoted as $GF(q)$, where q is some prime power p^n, n is some positive integer, and p is called characteristic of the field.

In cryptography, generally our main focus is on two types of Galois fields. First, when $n = 1$, that is, $GF(p)$ and second, when $p = 2$, that is, $GF(2^n)$. In the later case, we have polynomials of degree at most $n - 1$ with coefficients from $GF(2) = \{\bar{0}, \bar{1}\}$.

Definition 2.3.9. (Extension field)

A field $\mathbb{K}$ is said to be the extension field of a field $\mathbb{F}$ if $\mathbb{F}$ is contained by $\mathbb{K}$. It can also be said that $\mathbb{F}$ is subfield of $\mathbb{K}$. Extension field is denoted as $\mathbb{K}/\mathbb{F}$. Moreover, for a polynomial $p(x)$ in $\mathbb{F}$ there exists an extension field $\mathbb{K}$ of $\mathbb{F}$ which contains the roots of the polynomial $p(x)$.

Examples of extension field include $\mathbb{R}$ as extention field of $\mathbb{Q}$ denoted as $\mathbb{R}/\ \mathbb{Q}$, $\mathbb{C}$ as extension field of $\mathbb{R}$ denoted as $\mathbb{C}/\mathbb{R}$ etc.

When we are dealing with a large finite field, it is not very easy to find the multiplicative inverses, so there is an algorithm known as **extended Euclidean algorithm** which is taken into account for finding the inverse of all the numbers lying in the field.

Algorithm 2.3.10.

The extended Euclidean algorithm for finding the inverse of a number **b** under modulo **a** is as follows:

Input: a and b

Output: $b^{-1} \mod a$

1. Start by setting $(A, B, C) = (1, 0, a)$ and $(Q, R, S) = (0, 1, b)$
2. **If** $S = 0$ then **return** that the inverse of b does exist and C as the *gcd* of (a, b).
3. **If** $S = 1$ then **return** that R is the inverse of b and S as the *gcd* of (a, b)
4. Store $Z = \lfloor C/S \rfloor$, where $\lfloor . \rfloor$ represents the floor value.
5. $(L, M, N) = (A - ZQ, B - ZR, C - ZS)$
6. $(A, B, C) = (Q, R, S)$
7. $(Q, R, S) = (L, M, N)$
8. Go back to step no.2

Now we explain tables of addition and multiplication in finite fields. Let us consider the finite field $\mathbb{F}_{13}$ whose elements are $\{\bar{0}, \bar{1}, \bar{2}, \bar{3}, \bar{4}, \bar{5}, \bar{6}, \bar{7}, \bar{8}, \bar{9}, \bar{10}, \bar{11}, \bar{12}\}$. Addition and multiplication in $\mathbb{F}_{13}$ are shown in Tables 2.1 and 2.2 respectively.

$+$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
0	0	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	3	4	5	6	7	8	9	10	11	12	0
2	2	3	4	5	6	7	8	9	10	11	12	0	1
3	3	4	5	6	7	8	9	10	11	12	0	1	2
4	4	5	6	7	8	9	10	11	12	0	1	2	3
5	5	6	7	8	9	10	11	12	0	1	2	3	4
6	6	7	8	9	10	11	12	0	1	2	3	4	5
7	7	8	9	10	11	12	0	1	2	3	4	5	6
8	8	9	10	11	12	0	1	2	3	4	5	6	7
9	9	10	11	12	0	1	2	3	4	5	6	7	8
10	10	11	12	0	1	2	3	4	5	6	7	8	9
11	11	12	0	1	2	3	4	5	6	7	8	9	10
12	12	0	1	2	3	4	5	6	7	8	9	10	11

Table 2.1: Addition in $\mathbb{F}_{13}$

In Table 2.1 and Table 2.2 basic arithmetic modulo operation is used. Additive identity is the class 0 (mod 13) and multiplicative identity is the class 1 (

×	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10	11	12
2	0	2	4	6	8	10	12	1	3	5	7	9	11
3	0	3	6	9	12	2	5	8	11	1	4	7	10
4	0	4	8	12	3	7	11	2	6	10	1	5	9
5	0	5	10	2	7	12	4	9	1	6	11	3	8
6	0	6	12	5	11	4	10	3	9	2	8	1	7
7	0	7	1	8	2	9	3	10	4	11	3	10	4
8	0	8	3	11	6	1	9	4	12	7	2	10	5
9	0	9	5	1	10	6	2	11	7	3	12	8	4
10	0	10	7	4	1	11	8	5	2	12	9	6	3
11	0	11	9	7	5	3	1	12	10	8	6	4	2
12	0	12	11	10	9	8	7	6	5	4	3	2	1

Table 2.2: Multiplication in $\mathbb{F}_{13}$

mod 13). We can observe the additive and multiplicative identities of any number in $\mathbb{F}_{13}$ from these tables.

Definition 2.3.11. (Discrete logarithm problem)

The logarithms which are defined using multiplicative cyclic groups are called discrete logarithms, that is, $g^x \equiv y \mod p \Rightarrow \log_g y \equiv x \mod p$. Then discrete logarithm problem is to find 'x' when 'g' and 'y' are known under modulo p. The discrete logarithm [29] is a hard problem when p is a large prime.

Definition 2.3.12. (Polynomial interpolation)

Polynomial interpolation is a method for approximation of a continuous function whose values on some points are known and the values between the points are not known. Polynomial interpolation is thus used to approximate the function's values between the known values. So, the interpolating polynomial provides us the exact function's values at given points and approximation for the points at which the function's value is not known. For example, according to population census in 1997, Pakistan population was 132,352,279. If in 2002, it was 152,287,108. In 2007, it was 171,517,103 and in 2012 if the population was 190,483,542. According to latest census of 2017, it is 207,774,521. In this case, the values of population census are known for the years 1997, 2002, 2007, 2012 and 2017. But, if we want to know some approximated value for population between 1998 and 2017 apart form the already known values, say, on 2010 then polynomial interpolation is good tool to get an approximation for the year 2010. The most common example of interpolation is linear interpolation that uses a straight line for interpolation between the given points. Polynomial interpolation is thus a method that uses a polynomial, of degree less or equal to the number of known points, to approximate the function. There are different method to determine the polynomial for approximating the function. In our thesis, we will use Lagrange interpolation.

Definition 2.3.13. (Lagrange interpolation)

Suppose a function $f(x)$ is known on n data points that is $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ are known. Then the Lagrange interpolating polynomial of degree $n-1$ is

$$P(x) = \phi_1(x)y_1 + \phi_2(x)y_2 + ... + \phi_n(x)y_n = \sum_{i=1}^{n} \phi_i(x)y_i \tag{2.1}$$

where

$$\phi_i(x) = \frac{(x-x_1)(x-x_2)...(x-x_n)}{(x_i-x_1)(x_i-x_2)...(x_i-x_n)}.$$

Note that, in the denominator $x_i \neq x_j$ for $j = 1, 2, ..., n$, or

$$\phi_i(x) = \prod_{1\leq j\leq n} \frac{x-x_j}{x_i-x_j},\ j \neq i.$$

Equation 2.1 can now be written as,

$$P(x) = \sum_{i=1}^{n} y_i \prod_{1\leq j\leq n} \frac{x-x_j}{x_i-x_j},\ j \neq i. \tag{2.2}$$

which is the required Lagrange interpolation formula to approximate a function $f(x)$.

Example 2.3.14. In this example, we approximate the function $f(x) = x^2$ at $x = 1$ using **Lagrange interpolation**.

Let $f(x)$ be defined at points $x_1 = 0, x_2 = 2$ and $x_3 = 4$ as,

$$y_1 = f(x_1) = 0$$

$$y_2 = f(x_2) = 4$$

$$y_3 = f(x_3) = 16.$$

To construct the Lagrange interpolating polynomial $P(x)$ of degree 2, we first find $\phi_i(x)$ for $i = 1, 2, 3$

$$\phi_1(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)}$$

$$\phi_2(x) = \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)}$$

$$\phi_3(x) = \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)}.$$

Putting the values of x_1, x_2 and x_3, $\phi_1(x), \phi_(x)$ and $\phi_3(x)$ take the form

$$\phi_1(x) = \frac{(x - 2)(x - 4)}{(0 - 2)(0 - 4)} = \frac{(x - 2)(x - 4)}{8}$$

$$\phi_2(x) = \frac{(x - 0)(x - 4)}{(2 - 0)(2 - 4)} = \frac{(x)(4 - x)}{4}$$

$$\phi_3(x) = \frac{(x - 0)(x - 2)}{(4 - 0)(4 - 2)} = \frac{(x)(x - 2)}{8}.$$

Putting $\phi_1(x), \phi_2(x)$ and $\phi_3(x)$, and y_1, y_2, y_3 in Equation 2.1, gives us $P(x)$,

$$P(x) = 0.\frac{(x - 2)(x - 4)}{8} + 4.\frac{(x)(4 - x)}{4} + 16.\frac{(x)(x - 2)}{8}$$

$$P(x) = (x)(4 - x) + 2(x)(x - 2).$$

So, $P(1)$ is

$$P(1) = 1.$$

2.4 Elliptic curve cryptography

As mentioned earlier, computation of discrete logarithm problem is considered a hard problem. This problem can be made much harder if it is defined using elliptic curves and which is then known as "elliptic curve discrete logarithm problem" [18]. Elliptic curves provide more security to cryptographic schemes and it also results a reduction in computational complexity. There are many attacks on discrete logarithm mentioned in the literature, one of them is attack of sub exponential algorithm. This attack can be avoided if at least 1024 bit field is used. Elliptic curve can provide a same level of security by using a field of 160 bits and that is why elliptic curve is considered to provide more security with less number of bits and as a consequent, it saves both the time and the space. We now see what are elliptic curves and how they are used in cryptography.

Elliptic curve is defined by a generalized Weierstrass equation as

$$y^2 + c_1xy + c_2y = x^3 + c_3x^2 + c_4x + c_5 \tag{2.3}$$

where the variables x and y and constants c_1, c_2, c_3, c_4 and c_5 belong to a field. This curve should be non-singular that is satisfied by the determinant which must not be equal to zero (i.e. $\triangle \neq 0$). But, mostly our interest is in the simplified form of the Weierstrass equation which is

$$y^2 = x^3 + c_1x + c_2. \tag{2.4}$$

Here c_1, c_2 are again some constants, together with an extra point which is located at infinity and denoted as $\mathcal{O}$, called the point at $infinity$. Here, non singularity is ensured by $\triangle : 4c_1^3 + 27c_2^{/} = 0$ In the above equations, both the variables x, y and all the constants are the elements of some field $\mathbb{F}$. If we consider the field of real numbers then the variables $x, y \in \mathbb{R}$ and the constants $c_1, c_2 \in \mathbb{R}$ and the set of points which satisfy Equation 2.4 is denoted as $E_{\mathbb{R}}(c_1, c_2)$. For the the geometric characteristics of elliptic curve over real numbers $\mathbb{R}$, let us consider an elliptic curve

$$E : y^2 = x^3 + x + 6. \tag{2.5}$$

In the above equation $c_1 = 1$ and $c_2 = 6$ and the variables $(x, y) \in \mathbb{R}$. The values of (x, y) that satisfy Equation 2.5 are shown through a graph in Figure 2.5. Further, group operations can be defined on elliptic curve $E_{\mathbb{R}}(c_1, c_2)$ and to avoid having vertices or edges, the discriminant $4a^3 + 27b^2 \neq 0$. We are now going to observe geometrical addition of points on elliptic curve and then give the algebraic description of point addition on an elliptic curve. This will help us to define a group on an elliptic curve under the addition operator.

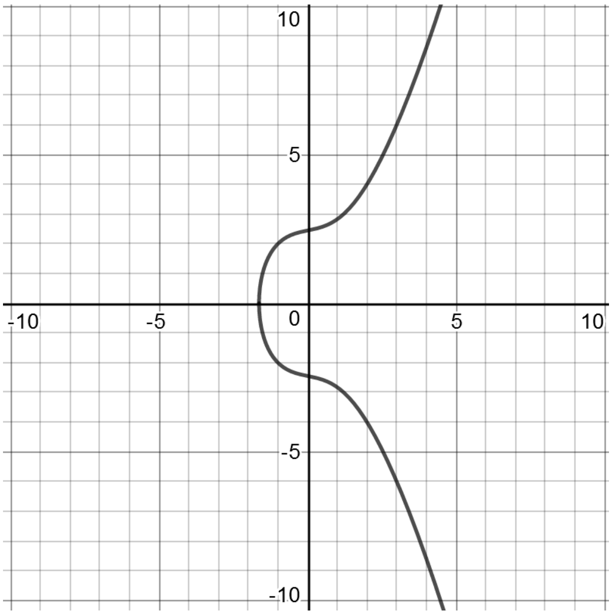

Figure 2.5: Graph of $E(1, 6)$ over $\mathbb{R}$

2.5 Group operations on elliptic curves

In this section, we define group operations on the points of elliptic curve P and Q with x and y coordinates.

1. The point at infinity $\mathcal{O}$ is taken as additive identity element, such that

$$P + \mathcal{O} = P$$

2. The negative of any given point P can be computed by just taking the negative of $y-coordinate$. Which means the negative of $P(x, y)$ is $P(x, -y)$.

3. The operation for adding two different points can also be defined. Suppose, points P and Q are to be added. The algorithm to add these points is as follows:

   a) Draw straight line through points P and Q.

   b) The line intersects the elliptic curve at point R.

   c) Take negative of R.

   d) This $-R$ is the point which we obtain from $P + Q$.

   It is clear from the above procedure that $P - P$ gives us the point $\mathcal{O}$ at infinity.

4. We can also add a point P to itself which means we want to double the point P. Here is the procedure to do so

   a) Draw a tangent line through point P.

   b) Intersection of tangent line with elliptic curve gives us point S.

   c) The negative of point S is the point which is obtained when P is added to itself.

Next, we plot a graph for an elliptic curve and also discuss the point addition for elliptic curve using the procedure mentioned in Section 2.5.

Geometric description for adding points of elliptic curve

Again using Equation 2.5 and taking two points $P(2, 4)$ and $Q(-1, 2)$ of it, we get a new point $R(x, y)$ as follows:

Figure 2.6 elaborates the points addition on elliptic curve. Suppose the points $P(2, 4)$ and $Q(-1, 2)$ are to be added. Draw a tangent line through P and Q that intersected the curve at point S. Then to get the required point R which is the

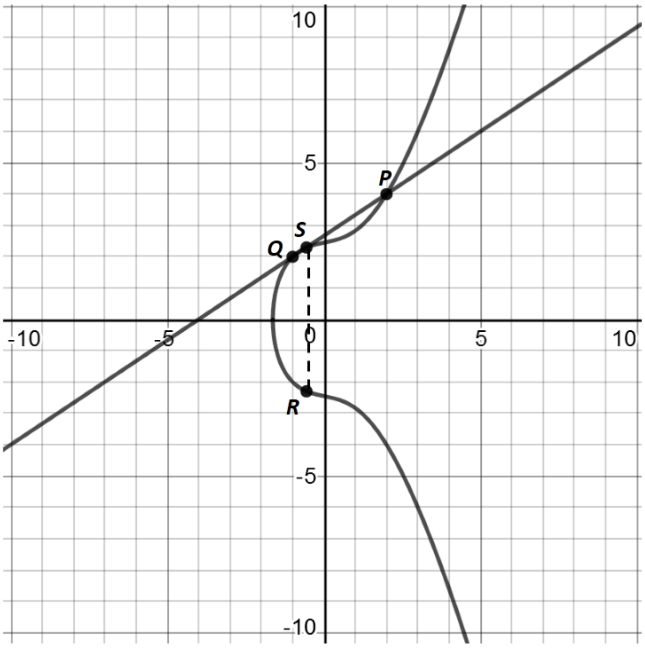

Figure 2.6: Graph of elliptic curve points addition over $\mathbb{R}$

sum of P and Q, Translate the $y - coordinate$ of S and we are done with the addition. Also, we know that if additive inverse of a point say P is added to itself then we get the additive identity. Now, let us see in the case of elliptic curve that how the identity $\mathcal{O}$ is obtained. We already know that the negative of $P(x, y)$ is $P(x, -y)$ so, for $P(2, 4)$ its negative $-P$ is $(2, -4)$, then by adding P into $-P$ graphically. We can observe from Figure 2.7 that the line that has been drawn through P and $-P$ goes to infinity and gives us point at infinity.

We are now all set to define the formulae to add points on elliptic curve. Suppose that the line through points P and Q be L, then

$$L : y = sx + c.$$

Slope s of the line L is given as

$$s = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}, & \text{if points are distinct} \\ \frac{3x_1^2 + a}{2y_1}, & \text{for point doubling.} \end{cases}$$

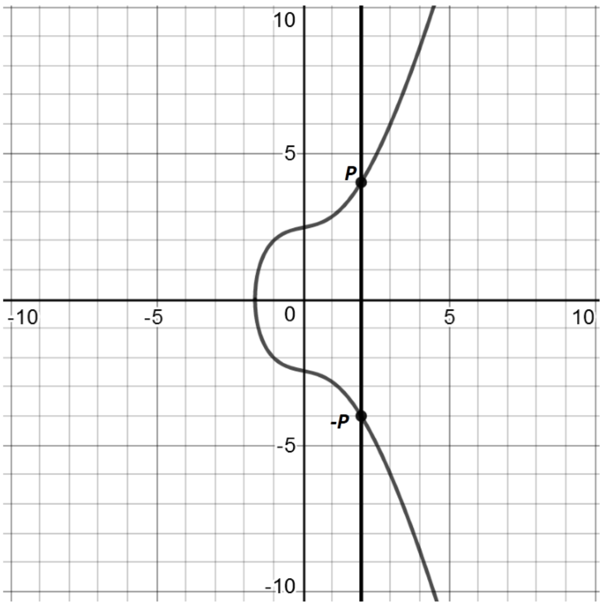

Figure 2.7: Elliptic curve point at infinity

then $(x_3, y_3) = P + Q$, in which

$$x_3 = s^2 - x_1 - x_2$$

$$y_3 = s(x_1 - x_3) - y_1$$

In cryptography, we are generally interested to define a group on elliptic curve over a finite field $\mathbb{F}_p$ defined by $E_{\mathbb{F}_p}(a, b)$. The idea to use elliptic curve in cryptography was given independently by Neal Koblitz [18] and Victor Miller [26] in 1985. Elliptic curve is defined by a cubic equation in two variables along with some coefficients over a finite field. The equation for elliptic curve over $GF(p)$ is as follows:

$$y^2 = (x^3 + ax + b) \mod p$$

together with a point at infinity $\mathcal{O}$. This curve should not be singular and has distinct roots which is ensured by the determinant $4a^3 + 27b^2 \neq 0$. It basically makes sure that the curve has no vertices and self-intersections. All the variables and coefficients belong to $\mathbb{Z}_p$. The operations that have been mentioned earlier to add points are more or less same in the case of a finite field $\mathbb{F}_p$. The extra thing for working in $\mathbb{F}_p$ as compared to $\mathbb{R}$ is that the modulo p has to be used in each operation. Geometrically, it is clear from Figure 2.8 that over a finite field

$\mathbb{F}_p$ the graph does not produce a curve rather we have scattered points this time. Consider an elliptic curve:

$$E : y^2 = (x^3 + x + 6) \mod 13 \tag{2.6}$$

The points $P(x, y)$ along with their additive inverses $P'(x, y)$ which satisfy Equation 2.6 are shown in Table 2.3:

x	y^2	$y_{1,2}$	$P(x,y)$	$P'(x,y)$
0	6	-	-	-
1	8	-	-	-
2	3	4,9	(2,4)	(2,9)
3	10	6,7	(3,6)	(3,7)
4	9	3,10	(4,3)	(4,10)
5	6	-	-	-
6	8	-	-	-
7	7	-	-	-
8	6	-	-	-
9	3	4,9	(9,4)	(9,9)
10	2	-	-	-
11	9	3,10	(11,3)	(11,10)
12	4	2,11	(12,2)	(12,11)

TABLE 2.3: Points of $E_{\mathbb{F}_{13}}(1, 6)$

The graph for Equation 2.6 is shown in Figure 2.8.

Elliptic curve points addition for $E_{\mathbb{F}_{13}}(1, 6)$ is given in Table 2.4.

The points on elliptic curve form a cyclic group which is always abelian as proved on page 17 of this thesis. It follows from the definition of cyclic group that all the elements of the group can be generated by a single element of the group which leads us to define the discrete logarithm problem in elliptic curve cryptography.

2.5.1 Elliptic curve discrete logarithm problem

Since elliptic curve on a finite field forms a cyclic group. So, for an elliptic curve E over $\mathbb{F}_p$ and for points P and Q belong to the additive group formed by points

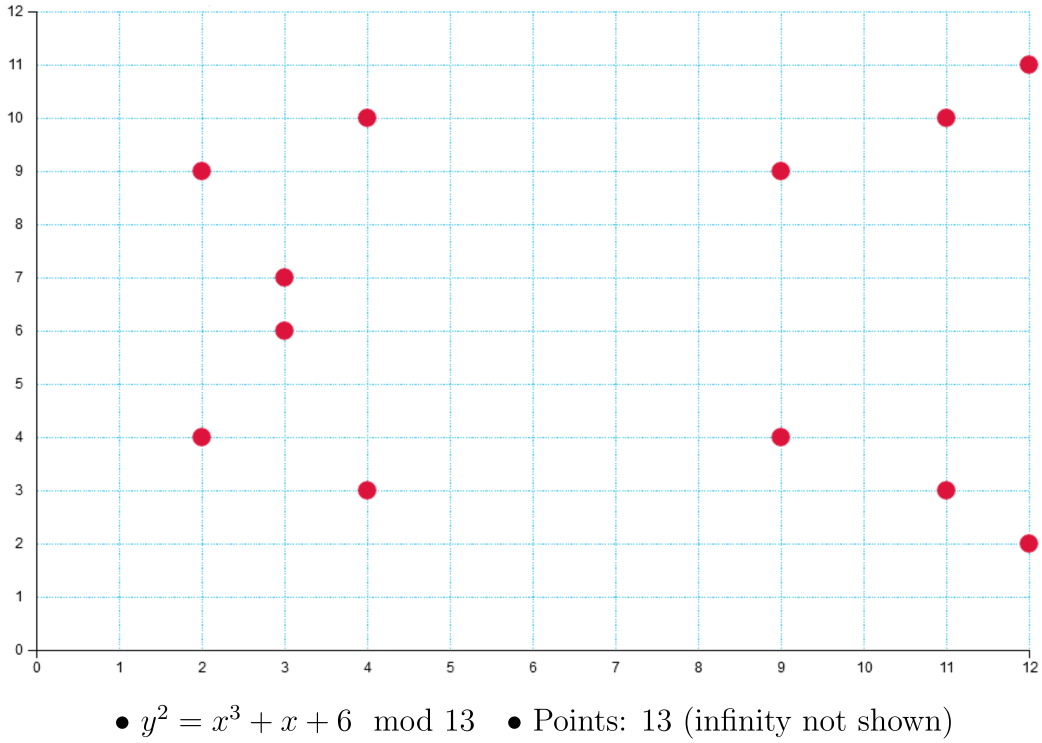

Figure 2.8: Graph of $E_{\mathbb{F}_{13}}(1, 6)$

+	∞	**(2,4)**	**(2,9)**	**(3,6)**	**(3,7)**	**(4,3)**	**(4,10)**	**(9,4)**	**(9,9)**	**(11,3)**	**(11,10)**	**(12,2)**	**(12,11)**
∞	∞	(2,4)	(2,9)	(3,6)	(3,7)	(4,3)	(4,10)	(9,4)	(9,9)	(11,3)	(11,10)	(12,2)	(12,11)
(2,4)	(2,4)	(9,9)	∞	(12,2)	(4,3)	(4,10)	(3,6)	(2,9)	(11,10)	(9,4)	(12,11)	(11,3)	(3,7)
(2,9)	(2,9)	∞	(9,4)	(4,10)	(12,11)	(3,7)	(4,3)	(11,3)	(2,4)	(12,2)	(9,9)	(3,6)	(11,10)
(3,6)	(3,6)	(12,2)	(4,10)	(11,10)	∞	(2,4)	(9,9)	(4,3)	(11,3)	(3,7)	(9,4)	(12,11)	(2,9)
(3,7)	(3,7)	(4,3)	(12,11)	∞	(11,3)	(9,4)	(2,9)	(11,10)	(4,10)	(9,9)	(3,6)	(2,4)	(12,2)
(4,3)	(4,3)	(4,10)	(3,7)	(2,4)	(9,4)	(2,9)	∞	(12,11)	(3,6)	(11,10)	(12,2)	(9,9)	(11,3)
(4,10)	(4,10)	(3,6)	(4,3)	(9,9)	(2,9)	∞	(2,4)	(3,7)	(12,2)	(12,11)	(11,3)	(11,10)	(9,4)
9,4	(9,4)	(2,9)	(11,3)	(4,3)	(11,10)	(12,11)	(3,7)	(12,2)	∞	(3,6)	(2,4)	(4,10)	(9,9)
(9,9)	(9,9)	(11,10)	(2,4)	(11,3)	(4,10)	(3,6)	(12,2)	∞	(12,11)	(2,9)	(3,7)	(9,4)	(4,3)
(11,3)	(11,3)	(9,4)	(12,2)	(3,7)	(9,9)	(11,10)	(12,11)	(3,6)	(2,9)	(4,10)	∞	(4,3)	(2,4)
(11,10)	(11,10)	(12,11)	(9,9)	(9,4)	(3,6)	(12,2)	(11,3)	(2,4)	(3,7)	∞	(4,3)	(2,9)	(4,10)
(12,2)	(12,2)	(11,3)	(3,6)	(12,11)	(2,4)	(9,9)	(11,10)	(4,10)	(9,4)	(4,3)	(2,9)	(3,7)	∞
(12,11)	(12,11)	(3,7)	(11,10)	(2,9)	(12,2)	(11,3)	(9,4)	(9,9)	(4,3)	(2,4)	(4,10)	∞	(3,6)

Table 2.4: Point additions in $E_{\mathbb{F}_{13}}(1, 6)$

of elliptic curve, we can write

$$\underbrace{P + P + P + ... + P}_{\text{t times}} = tP = Q.$$

From the above mentioned equation, it can be observed that for given t and P it is easy to compute Q but it is relatively hard to find t knowing only P and Q.

$$t = \log_P Q$$

and the problem of finding this t is called Elliptic Curve Discrete Logarithm Problem (ECDLP) [18] in cryptography.

Definition 2.5.1. (**Bilinear Pairing**)
Let G_1 be an additive cyclic group of order n, where n is prime, with identity $\mathcal{O}$ and G_2 be a multiplicative cyclic group of order n with identity 1. The mapping $\hat{e}$

$$\hat{e} : G_1 \times G_1 \rightarrow G_2$$

is called bilinear mapping if it satisfies the following three properties:

1. **Bilinearity**: For all $P_1, P_2, P_3 \in G_1$,

   a. $\hat{e}(P_1 + P_2, P_3) = \hat{e}(P_1, P_3).\hat{e}(P_2, P_3)$

   b. $\hat{e}(P_1, P_2 + P_3) = \hat{e}(P_1, P_2).\hat{e}(P_1, P_3)$.

2. **Non-Degeneracy**: It must not map to the identity of G_2 for all $P \in G_1$. That is, $\hat{e}(P, P) \neq 1$, for all $P \in G_1$.

3. **Computability**: There should be a polynomial time algorithm that computes $\hat{e}$.

Weil pairing [23] and Tate pairing [10] are the examples of such a construction. These both pairings are computed using Miller's algorithm [25]. We will use bilinear pairing to secure our cryptosystem later in Chapter 3.

2.6 Hash functions

Hash functions have a great importance in cryptography. They are used to map any length of message to a fixed length output called the digest of the message. Figure 2.9 illustrates the phenomena of a hash function.

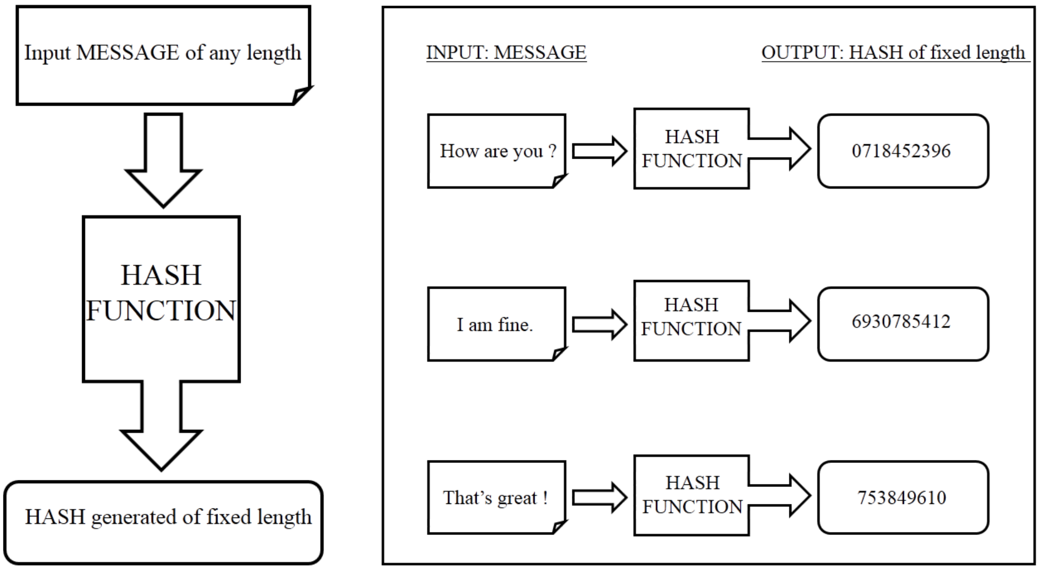

FIGURE 2.9: Hash function

The message digest should be unique for a particular message so that it can be considered as a finger print of the message. A good hash function contains the following properties:

1. It can be applied to any length message.
2. It must always produce a fixed length message digest.
3. It is relatively easy to compute the hash value of the message.
4. It should be one way in a sense that it must be impossible to find the input message from the message digest.
5. The hash of two different messages must not be same.
6. It must provide integrity to the message. i.e, a minor change in the message results major change in the hash value.

Examples of famous hash algorithms are SHA1 [6], MD5 [32] and SHA-512 [12].

2.7 Secret sharing

Secret information or data cannot be shared with everyone and only a trustworthy person can keep it secret. Unfortunately, there is no mechanism which guarantees a person being honest and trustworthy. So when it comes to keeping something secret; it is always better to share some parts of secret information with people rather sharing the complete secret with only one person. In this way, each person has a less knowledge or control to reveal the secret. When we share a secret with one person, we allow them to have all the power. They might deceive us by sharing the secret with those who are not intended to know about it.

Let us consider an example. Suppose there is a secret key that is used to open many important files. As this key is very important and if it is lost or placed in the wrong hands, all the important information might be known to everyone. Here an important question arises that how can we make a setup which fulfills these basic necessities of keeping the information secret. Have a look on another problem. suppose there are eleven scientists working on a secret project. Obviously, they wish to store the important documents somewhere may be in a cabinet to make sure that they are safe. But here the problem is that they cannot trust a single person to keep the key of the cabinet. Now suppose they wish to set up a mechanism through which eleven scientists must be present to unlock the cabinet. For this purpose, they would need to have 462 locks and 252 keys per scientist. This is clearly impractical with these big numbers of locks and keys. So, what should be done to overcome all these difficulties for making a better system that guarantees our secret data being safe.

We often keep our secret information in human brain or in a computer system so that it is well guarded. But any misfortune like a sudden death of the person or computer breakdown may result in lose of the secret information. We can overcome this issue by keeping the multiple copies of the secret information at different places but then we are compromising the security of the secret information. So

we need to find an approach which provides security to our secret information and which is reliable. As we cannot give authority to a single person to have the complete knowledge of the secret because they can take advantage of their authority. Also, we can not keep the multiple copies of the secret because that results in compromising the security of the information. There can be the solution to these problems, if a proper set-up is built that do not give authority to a single person and rather it requires the participation of a few people in revealing the secret.
As far as cryptography is concerned, we always tend to secure our data so that it is not placed in the wrong hands. We know that encrypting our data makes it confidential for the third party and for encryption, we use different schemes which use a secret key to encrypt the data. Here, it is very much clear that our scheme is dependent on the private key. If this key is lost or placed in the wrong hands the whole scheme can become insecure. Cryptographic keys are often of a big length that is why it is not easy to remember them or to store them in someone's mind. The owners often keep their private keys in a computer system or somewhere, where they can be accessed easily. But we have already discussed that these scenarios could increase the security breaches. In the next paragraph, the solution to all these difficulties is discussed.

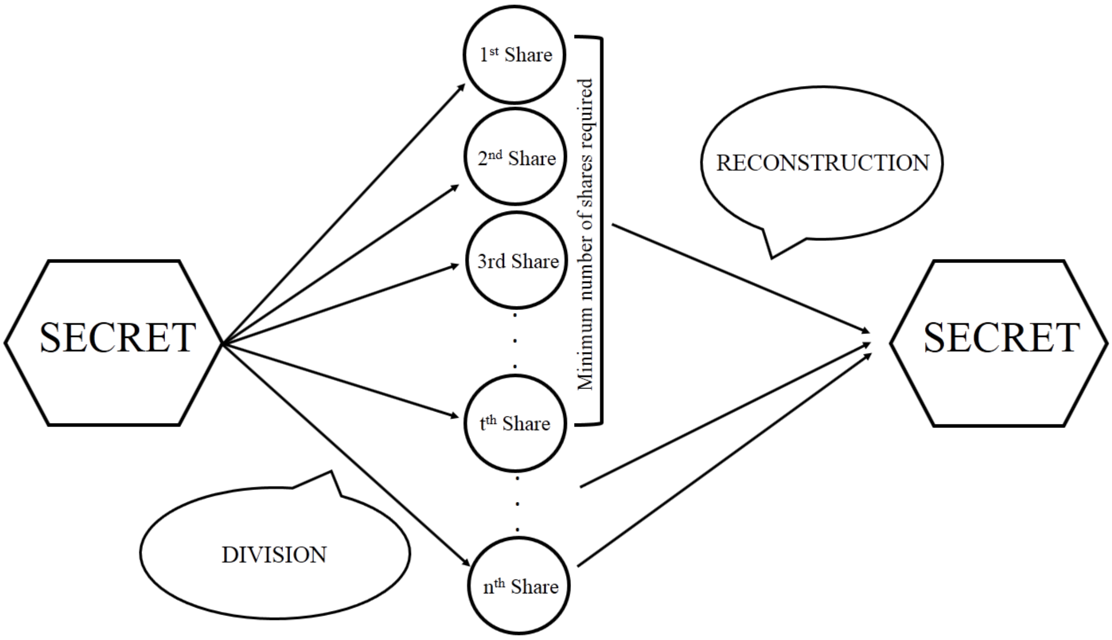

Figure 2.10: Secret Sharing (t, n)-**threshold** Protocol

In 1979, Shamir [34] and Blakley [3] independently invented secret sharing scheme. They proposed schemes in which they made it possible to save cryptographic keys or any highly important secret information. In a secret sharing scheme, a secret is split into n number of pieces such that these pieces individually do not give any information about the secret information. When it is necessary to get back the secret from these pieces, any t or more pieces are combined and the secret is obtained. This scenario is also explained through Figure 2.10. Here, one thing is important that any less than t pieces of shares must not give any knowledge of the secret otherwise the technique will be of no use. A secret sharing scheme that uses t out of n pieces to recover the secret is known as (t, n) **-threshold scheme**.

2.7.1 Shamir secret sharing scheme (SSSS)

Shamir scheme [34] is a (t, n) **-threshold** scheme. The secret 'K' can only be recovered when 't' out of 'n' shares are available to reconstruct the secret 'K'. The idea was based on polynomial interpolation for example Lagrange interpolation that we saw earlier in Definition 2.3.13. First, a polynomial of degree $(t - 1)$ is constructed in which secret K is placed as the first co-efficient of it and rest of the co-efficients are picked at random from $\mathbb{Z}_p$. By using the constructed polynomial, shares are generated. When it is required to get back the secret, t shares are joined together. This scheme achieves the goal for secret K to be easily computed by having a knowledge of t shares but knowledge of less than t shares do not give any idea about it. We will shorlty give the complete procedure of Shamir scheme and see how does it work. But, before proceeding to it let us fix some notations.

As we know that there will be participants or members who take part in secret sharing. Let 'P' be the set of n participants and the secret to be shared be 'K'. The secret 'K' is chosen by a special participant who is trusted by everyone, called dealer 'D'. Now, the dealer 'D' splits the secret 'K' into 'n' number of pieces called shares. Each share should be delivered to the designated participant using a secure channel so that any other participant does not know the share of the other participant. Once the shares are secretly delivered, any subset of 't' participants can reconstruct the secret by combining their shares. Any subset of less than 't' participants cannot obtain any information or clue about the secret. Further, it should be made clear that which set of participants can recover the secret and which cannot. The set of participants which can recover the secret must be equal to or greater than t is known as **authorized set**. The set of participants which are less in numbers than 't' cannot recover the secret or get any clue about it is known as **unauthorized set**. Shamir's scheme uses modular arithmetic instead of real arithmetic. Public parameters in SSSS are $\mathbb{Z}_p$ and x_i's which are selected by the dealer. The scheme has three phases as mentioned below.

A. Initialization

To initialize the scheme, n distinct non-zero elements $x_i \in \mathbb{Z}_p$ are randomly chosen by the dealer D and then they are made public.

B. Share Distribution

1. The aim of this scheme is to share the secret K among the participants P_i. In order to achieve this aim, D has to formulate a polynomial of degree $t-1$ for which the coefficients $a_1, a_2, a_3, ..., a_{t-1}$ are chosen at random and first coefficient a_0 is the secret K.

2. After constructing the polynomial, each participant share y_i is generated from it. By putting the values of x_i in the polynomial we get each participant's share as $y_i = q(x_i)$, for $1 \leq x \leq n$, where

$$q(x) = \sum_{j=0}^{t-1} a_j x^j \mod p. \tag{2.7}$$

3. Each participant P_i is given their respective share y_i.

C. Secret Reconstruction

To reconstruct the secret, the combiner uses Lagrange interpolation. The

Lagrange interpolation formula is given below,

$$q(x) = \sum_{i=1}^{t} y_i \prod_{1 \leq j \leq t} \frac{x - x_j}{x_i - x_j}, \text{ and } j \neq i \tag{2.8}$$

As first coefficient of the polynomial is our secret K. To obtain the secret from Lagrange interpolation we have to evaluate it at $x = 0$. Evaluating Lagrange interpolation for $x = 0$ reduces Equation 2.8 to

$$K = q(0) = \sum_{i=1}^{t} y_i \prod_{1 \leq j \leq t} \frac{x_j}{x_j - x_i}, \text{ and } j \neq i \tag{2.9}$$

The formula in Equation 2.9 is the required formula for reconstructing the secret K.

The procedure of SSSS is further explained step by step in next example.

Example 2.7.1. Suppose we want to share a secret $K = 4$ using Shamir's scheme. We fix $\mathbb{Z}_{19}$ and proceed as follows:

A. Initialization

Suppose that $p = 19$ that means we will be working throughout in the field $\mathbb{Z}_{19}$. Let the secret be $K = 4$ and total number of participants be $n = 5$ and let the number of participants which can reconstruct the secret be $t = 3$. For each participant P_i, an integer x_i is chosen at random from $\mathbb{Z}_{19}$. In this example, we let $x_i = i$, for $1 \leq i \geq 5$.

B. Share distribution

1. As $t = 3$ and we know that the polynomial degree is one less than t, so, it will be degree two polynomial. To construct the degree two polynomial, three coefficients are required. Two coefficients $a_1 = 11, a_2 = 15$ are chosen at random from $\mathbb{Z}_{19}$ and first coefficient $a_0 = K$, then the polynomial gets the form:

$$q(x) = 4 + 11x + 15x^2 \mod 19; \tag{2.10}$$

2. The shares of the participants are generated by putting x_i in polynomial 2.10, we get
    (a) $y_1 = q(1) = 11$
    (b) $y_2 = q(2) = 10$
    (c) $y_3 = q(3) = 1$
    (d) $y_4 = q(4) = 3$
    (e) $y_5 = q(5) = 16.$
3. A secure channel will be used here for delivering the share to corresponding partcipant.

C. Secret reconstruction

The secret is calculated using the Lagrange formula mentioned in Equation 2.9 inverses of the numbers below are obtained through algorithm 2.3.10

$$\begin{aligned} K &= \Bigg[11\Big(\frac{2}{2-1}\Big)\Big(\frac{3}{3-1}\Big) + 10\Big(\frac{1}{1-2}\Big)\Big(\frac{3}{3-2}\Big) \\ &+ 1\Big(\frac{1}{1-3}\Big)\Big(\frac{2}{2-3}\Big)\Bigg] \mod 19 \\ &= \big(11.2.3.2^{-1} - 10.3 + 1.2^{-1}.2\big) \mod 19. \end{aligned}$$

The inverse of $2^{-1} \mod 19 = 10$ is computed using Extended Euclidean Algorithm 2.3.10.

$$\begin{aligned} K &= (660 - 30 + 20) \mod 19 \\ K &= 650 \mod 19 \\ K &= 4. \end{aligned}$$

We can observe that the secret value chosen initially and the secret value obtained by Lagrange formula are same. So, the reconstructed secret is correct and the example ends with it.

2.8 Summary

This chapter is the base of the thesis because it consists many simple and some advanced concepts. Without uderstanding them it might be a little difficult to have a good understanding of this thesis. First, we briefly gave an overview of basics of cryptography and the bottom line is that it provides security to highly confidential information. Afterwards, we discussed the important characteristics of two different cryptographic protocols; their drawbacks and strong areas. Mathematical background related to groups, rings and field have also been included to have a good grip to understand cryptographic protocols. One of the most important things in cryptography on which many cryptosystems are based known as discrete logarithm problem, has also been taken into the consideration in this chapter. Then we elaborated the use of elliptic curve in cryptography and discussed how the security is improved and computations are reduced. Elliptic curve makes DLP more harder for attackers. We also explained the concept of hash functions because they are being used extensively in many cryptographic application. Finally after discussing some basics, we moved towards the theme of the thesis that is secret sharing scheme and there we discovered the importance of secret sharing and its working. Shamir scheme's algorithm along with an example was presented and discussed. In a nutshell, all the basics related to the secret sharing scheme were mentioned and discussed in this chapter.

Chapter 3

Secure and efficient SSS based on elliptic curve and pairing

In this chapter, we will look into the scheme proposed by Sreekumar and Binu [2] that is based on Shamir's secret sharing scheme but it is highly secured and efficient as compared to Shamir's secret sharing scheme [34]. The complete procedure of the scheme is mentioned in this chapter. Further, this chapter also covers the security as well as the computational aspects of the scheme. During the analysis, we noticed that there are some security flaws in this scheme which we will discuss in Chapter 4 along with the counter-measures.

3.1 Introduction

From the discussion of previous chapter we know that elliptic curve plays a vital role in cryptography. It takes the security of cryptographic applications to the next level. Since it provides a great security using only 160 bits field so the computational complexity which was a challenging problem in finite field to get a desired security has also been reduced to a great extent. That is why it caught the attentions of many researchers in cryptography and many different applications based on elliptic curve are being introduced. The scheme that is proposed by Sreekumar and Binu [2] is based on Shamir's secret sharing scheme and elliptic curve pairing. Use of elliptic curve makes their scheme more efficient and bilinear pairing is used for verification purposes in the reconstruction so that a cheater or a disloyal participant can be caught. The scheme also allows participants to choose their shares and hence it makes the participants independent from the involvement of dealer in selecting the shares. Participants evaluate pseudo shares from their secret shares and then the pseudo share are used for the reconstruction of the secret. Pseudo shares allow participants to keep their actual shares hidden and use them to reveal multiple secrets. The scheme is also dynamic in nature because participants can be added or deleted without any change in the secret share of the participants. Moreover, the secret or access structure can also be modified without having any influence on the existing participants secret shadow. The access structure contains the monotone property which means a superset of an authorized set is also authorized and can recover the secret as well and if a set of participants is unauthorized than any of its subset will also be unauthorized. Multiple secrets can also be reconstructed using the proposed scheme and it does not require participants to change their shares. Now, we move towards the construction of the scheme. Here, we only mention the process for sharing a single secret but it can also be extended to share multiple secrets.

3.2 The proposed scheme

To initialize the scheme a public bulletin board is used for the convenience of participants to access publicly available values. Any kind of modification on the bulletin board can only be made by the dealer. For the convenience of reader, all the global parameters used in the scheme are mentioned in Table 3.1.

Let there be n participants $P_1, P_2, ..., P_n$ taking part in secret sharing and a monotone access structure $ß^0 = \{ß_1, ß_2, ..., ß_t\}$ is taken. Participants choose their shares without any interference of the dealer and these shares are kept secret for using

Symbols	**Description**
K	Secret
n	Total number of Participants
P_i	i^{th} Participant
p_i	public identity of i^{th} participant
D	A trusted Dealer
t	Total number of qualified subsets
d	Number of particpants in each qualified subset
$ß^0$	Monotone access structure
$ß_i$	i^{th} qualified subset
p	A large prime
r	An integer
E	Elliptic curve
G_1	Additive group of points of an Elliptic curve $E_{\mathbb{F}_p}(a, b)$
G	Generator of G_1
G_2	Multiplicative subgroup of extension of finite field $\mathbb{F}^*_{p^2}$
$\hat{e}$	Modified Weil pairing
H	Secure hash function
X_i	Secret share of i^{th} participant
X_{ij}	Secret share of i^{th} participant in the j^{th} qualified subset
Y_i	Pseudo share i^{th} participant
Y_{ij}	Pseudo share of i^{th} participant in the j^{th} qualified subset
$r(x)$	Polynomial of degree one
z_i	Identity of i^{th} qualified subset
X_0	Dealer's secret share
Y_0	Dealer's pseudo share

Table 3.1: Global parameters in the scheme

several times. Next, the complete procedure of the scheme is given.

The four important stages of the scheme are mentioned below:

1. Initialization of the scheme
2. Generation of shares
3. Distribution of secret
4. Verification and reconstruction of secret

3.2.1 Initialization of the scheme

In the initialization phase some public parameters necessary for initializing the scheme are posted on the bulletin board by the dealer so that they can be accessed by every participant $P_1, P_2, P_3, ..., P_n$.

1. A trusted dealer **D** chooses an elliptic curve E over $GF(q)$, where $q = p^r$, with p being a large prime such that DLP and ECDLP are hard in $GF(q)$. Let G_1 be an additive cyclic group of the points of elliptic curve over $\mathbb{F}_p$. Let G_2 be a multiplicative subgroup of an extension of finite field $\mathbb{F}^*_{p^2}$. To map the elements of G_1 to G_2, elliptic curve pairing is used e.g. modified Weil pairing ($\hat{e}$) [**?**].

2. Let G a generator of G_1 be chosen by the dealer and a hash function H is defined to map $H : G_1 \mapsto \{0, 1\}^\ell$, where ℓ is the bit length of the field.

3. All these public parameters $\{E, G_1, G_2, q, G, \hat{e}, H\}$ are published in the notice board to access them for later stages.

3.2.2 Generation of shares

In the second stage, participants select their shares and send them to dealer. Dealer first verifies the shares sent by participants and then each share is assigned to the respective participant. Shares are then published in the public bulletin board.

1. Each participant P_i selects X_i at random in $\mathbb{Z}^*_q$ as their secret share which is not to be revealed and it must be kept secret to use for the reconstruction of multiple secrets. Each participant then computes their pseudo share

$$Y_i = X_i G.$$

   The pseudo shares are now used to reconstruct the secret. Each participant P_i submits their pseudo share Y_i to the dealer. Here, dealer has to make sure that all the pseudo shares are distinct so that no participant has a share that some else participant also has. If dealer finds that two pseudo shares are same then they request the participants to chose some other share. No one can access the secret share of the participant because in order to obtain it, ECDLP must be solved which is assumed a hard problem.

2. Each participant P_i's pseudo share Y_i is published on the bulletin board along with the public identity p_i that is chosen at random from $\mathbb{Z}_q^*$.

3.2.3 Distribution of secret

1. Let the secret K to be shared among n participants. Here, degree one polynomial is only needed to be set up which reduces the computational cost of this scheme as compared to many other schemes available in the literature. Polynomial $r(x)$ of degree one is as follows:

$$r(x) = K + cx, \quad \text{where } c \in \mathbb{Z}_q^*$$

2. For giving the identity to each minimal qualified subset in $ß^0$, the dealer selects an integer $z_1, z_2, ..., z_t \in \mathbb{Z}_q^*$ at random.

3. Here, dealer also needs to add their part in secret sharing so that the scheme can be made more secure. It will also help the combiner to easily verify shares. To add some contributuion of dealer, a random number $X_0 \in \mathbb{Z}_q^*$ is chosen to compute:

$$Y_0 = X_0 G \quad \text{and} \quad Y_i' = X_0 Y_i \quad \text{for} \quad i = 1, 2, ..., n.$$

4. Compute $r(1)$ and for each qualified subset $ß_j = \{P_{1j}, P_{2j}, ..., P_{dj}\}$ in $ß^0$, here P_{ij} means participant P_i in the j^{th} subset, compute

$$B_j = r(z_j) \oplus H(Y_{1j}^{'}) \oplus H(Y_{2j}^{'}) \oplus ... \oplus H(Y_{dj}^{'}), \quad 1 \leq j \leq t$$

d is total number of participants in each qualified access set.

The purpose to compute $r(1)$ is to enable the combiner to easily reconstruct the secret. As we know that to compute a degree one polynomial, two points must be required. Hence, $r(1)$ will be made available on the bulletin board and then in the Lagrange interpolation phase, it will be accessed from there to reveal the secret.

5. Values of $Y_0, r(1), (z_1, B_1), (z_2, B_2), ..., (z_t, B_t)$ are published on the public bulletin board.

3.2.4 Verification and reconstruction of secret

There are t qualified subsets in the access structure $ß^0$ and each qualified subset $ß_j, 1 \leq j \leq t$ has the access to reconstruct the secret. Secret shares of the participants and the publicly available values on the public bulletin board are used for retrieving the secret. Combiner has also given the authority to verify the shares and also identify cheaters.

1. Each participant P_{ij} of the qualified subset $ß_j$ obtains Y_0 from the public bulletin board and use their secret share X_{ij} to compute:

$$Y_{ij}^{'} = X_{ij} Y_0.$$

   Then $Y_{ij}^{'}$ is delivered to the designated combiner for obtaining the secret.

2. Before obtaining the secret, combiner first verifies the share using bilinear pairing that is to check

$$\hat{e}(G, Y_{ij}') = \hat{e}(Y_0, Y_{ij}).$$

   If this condition is satisfied then the participant has sent the valid share and if it is not met then the corresponding participant is invalid and their share can not be accepted and that is how an invalid share or an a intruder can be identified.

3. The combiner after receiving all the valid shares can retrieve

$$r(z_j) = B_j \oplus H(Y_{1j}^{'}) \oplus H(Y_{2j}^{'}) \oplus ... \oplus H(Y_{dj}^{'}).$$

4. Lagrange interpolation can now be used to reconstruct the secret $K = r(0)$ using the values of $r(1)$ and $r(z_j)$.

$$r(x) = r(1).\frac{x - z_j}{1 - z_j} + r(z_j).\frac{x - 1}{z_j - 1}.$$

   or we may write it for $r(0)$

$$r(0) = r(1).\frac{z_j}{z_j - 1} + r(z_j).\frac{1}{1 - z_j}.$$

5. The shared secret K is reconstructed.

We now present a toy example that illustrates the working of the scheme and see how does the secret is obtained.
Note that it is just a toy example that highlights the important functions of the scheme for reconstructing the secret. In this example, we will not use a big prime field just to make the computations easy but for practical purposes one must use a big prime field to avoid any cheating from an unauthorized person. The information about each step of each phase will be explained in the next example unless otherwise stated.

Example 3.2.1. We fix the field $\mathbb{Z}_{13}$ and an elliptic curve E over it will be defined in the initialization phase. The four phases of the scheme work for sharing and reconstructing the secret are as follows:

A. Initialization of the scheme

A1. Let us suppose a trusted dealer (**D**) who chooses an elliptic curve

$$E : x^3 + x + 6 \mod 13.$$

A2. let us consider a generator $G = (4, 10)$.

A3. Then, we have defined the following hash values for all the elliptic curve points. Note that these hash values are assumed because our main intention is to only show the working of the scheme. For proper execution of the scheme, one can use any secure hash function and then to keep the bit size in the given field, one can transform the output bit string in it. Here, we have used $\mathbb{Z}_{13}$ whose bit size is 4 and therefore the assumed hash values are taken of size 4 bits.

Point $P(x, y)$	Hash value	Point $P(x, y)$	Hash value
$(2, 4)$	0000	$(4, 3)$	1001
$(2, 9)$	1011	$(4, 10)$	0001
$(3, 6)$	1010	$(9, 4)$	0101
$(3, 7)$	1010	$(9, 9)$	0010
$(11, 3)$	1011	$(12, 2)$	1011
$(11, 10)$	0111	$(12, 11)$	0111

Table 3.2: Hash values of elliptic curve points

B. Generation of shares

Let the number of participants be 6 and the integers selected by participants $P_1, P_2, P_3, P_4, P_5, P_6$ as their secret share are $1, 4, 5, 8, 10, 12$ respectively. Then each participant computes the pseudo share from the secret share as $Y_i = X_i G$:

$$
\begin{aligned}
Y_1 &= 1(4, 10) = (4, 10) = 0001 \\
Y_2 &= 4(4, 10) = (9, 9) = 0010 \\
Y_3 &= 5(4, 10) = (12, 2) = 1011 \\
Y_4 &= 8(4, 10) = (12, 11) = 0111 \\
Y_5 &= 10(4, 10) = (3, 7) = 1011 \\
Y_6 &= 12(4, 10) = (4, 3) = 1001.
\end{aligned}
$$

These pseudo shares play important role in the scheme for revealing K. They also keep the secret shares hidden.

C. Distribution of secret

C1. Let 7 be a secret to be shared. Dealer sets up a polynomial $f(x)$ of degree 1.
i.e.; $f(x) = 7 + 2x \mod 13$

C2. Here, if we let the number of partcipants must contribute to reveal the secret be 4 and let the qualified set be $ß_1 = \{Y_1, Y_2, Y_3, Y_4\}$ then an integer $a_1 = 5$ is chosen to represent $ß_1$.

C3. Dealer also adds some contribution of them to make the scheme more secure. Here is given the dealer's part.
Let $X_0 = 6$ be chosen as dealer secret share. Now, compute dealer's pseudo share by $Y_0 = X_0G$ also $Y_i^{'} = X_0Y_i$ for $i = 1, 2, 3, 4, 5, 6$. The following values are obatianed then

$$Y_0 = 6(4, 10) = (11, 10)$$
$$Y_1^{'} = 6(4, 10) = (11, 10)$$
$$Y_2^{'} = 6(9, 9) = (2, 9)$$
$$Y_3^{'} = 6(12, 2) = (9, 9)$$
$$Y_4^{'} = 6(12, 11) = (9, 4)$$
$$Y_5^{'} = 6(3, 7) = (12, 11)$$
$$Y_6^{'} = 6(4, 3) = (11, 3).$$

C4. Compute $f(1) = 7 + 2(1) \mod 13 = 9 = 1001$.
and,

$$\begin{aligned} A_1 &= f(a_1) \oplus H(Y_1^{'}) \oplus H(Y_2^{'}) \oplus H(Y_3^{'} \oplus H(Y_4^{'}) \\ &= f(5) \oplus H(11, 10) \oplus H(2, 9) \oplus H(9, 9) \oplus H(9, 4) \\ &= 0100 \oplus 1110 \oplus 1000 \oplus 0010 \oplus 0101 \\ A_1 &= 0101 = 5. \end{aligned}$$

C5. Publish $Y_0 = (11, 10), f(1) = 5, (a_1, A_1) = (5, 5)$ on the public bulletin.

D. Verification and reconstruction of secret

D1. Each participant P_i in the qualified subset $ß_1$ can access Y_0 from the bulletin board and computes $Y_i^{'} = Y_0X_i$, using the secret share X_i. The participant then delivers $Y_i^{'}$ to the designated combiner. The combiner first verifies the shares using bilinear pairing e.g. modified weil pairing. We do not need to compute modified weil pairing manually. Thanks to Miller's algorithm who does this job for us (for Miller's algorithm see [25]). If the shares received are valid then proceeds further. Here, we suppose that the received shares are valid.

D2. Once all the valid shares are received, the combiner can retrieve

$$\begin{aligned} f(a_1) &= A_1 \oplus H(Y_1^{'}) \oplus H(Y_2^{'}) \oplus H(Y_3^{'}) \oplus H(Y_4^{'}) \\ f(5) &= 0101 \oplus 1110 \oplus 1000 \oplus 0010 \oplus 0101 \\ f(5) &= 0100 = 4. \end{aligned}$$

D3. Using $f(1)$ and $f(5)$, the polynomial can be reconstructed using the Lagrange Interpolation. The inverses of the numnbers are computed using Algorithm 2.3.10.

$$\begin{aligned} f(x) &= f(1).\frac{x-5}{1-5} + f(5).\frac{x-1}{5-1} \\ &= 9.\frac{x-5}{-4} + 4.\frac{x-1}{4} \\ &= -9 \times (x-5) \times 4^{-1} + 4 \times x - 1 \times 4^{-1} \mod 13 \\ &= -9 \times (x-5) \times 10 + 4 \times x - 1 \times 10 \mod 13 \\ f(x) &= -90 \times (x-5) + (x-1) \mod 13. \end{aligned}$$

D4. The shared secret K is $f(0)$.

$$\begin{aligned}
f(0) &= -90 \times (-5) - 1 \mod 13 \\
f(0) &= -90 \times (-5) - 1 \mod 13 \\
f(0) &= 449 \mod 13 \\
f(0) &= 7 = K.
\end{aligned}$$

3.3 Security analysis of the scheme

Secret sharing scheme highly depends on the distribution of shares to the participants securely. If the shares are not being distributed securely to the participants then the secret sharing scheme is not secure and can have an attack on it and as a result secret can be gone into the wrong hands. This can be a disastrous situation in cryptography because protection of the secret is the only thing for which all this setup is made. In traditional secret sharing schemes, dealer selects the shares for the participants. If, in case, a dealer is not a trustworthy then we can always expect to have inconsistent shares which make the scheme weak in terms of its security. But, thanks to verifiable secret sharing schemes which allow us to verify that the shares are consistent. This intuitively means that each authorized set of participants will construct the same secret when the participants in authorized set combine their shares.

Strong security in share distribution phase

In this scheme the participants are totally made free from the intervention of the dealer in selecting the shares. Choosing a share for themselves allows the participants to use their secret share multiple times and that is how the problem of choosing a new share every time to reconstruct a secret has been avoided. Since each participant's actual share/ secret share does not participate in the reconstruction phase rather it is being used at back end to produce pseudo share which then actually takes part to reconstruct the secret. Finding actual share/ secret share of the participant from pseudo share is equivalent to solve ECDLP. Anybody cannot calculate secret share from pseudo share because of ECDLP and cannot guess any thing about the secret share.

Role of ECDLP and bilinear pairing in secret reconstruction phase

In the reconstruction phase, dealer can also verify the shares of every participants taking part in the reconstruction of the secret by their pseudo shares. Again, security of X_0 depends on ECDLP as Y_0 cannot reveal any information about X_0. Only the birthday paradox method solves the ECDLP but it has a limit and once that limit is exceeded this method also fails to solve ECDLP. Birthday paradox [11] method running time is $O(n)$, where n is the order of the group. Using a field of 160 bits allows to avoid this attack. Here, one can observe the difference between DLP and ECDLP as DLP requires a field of 1024 bits to avoid the attack of sub exponential algorithm [8]. Hence, elliptic curve is more suitable for providing a better security with less number of bits. As a result it saves the time as well as the space. Bilinear pairing in the reconstruction phase has a great importance in the scheme since it allows combiner to check the validity of participant share. For computation of Weil pairing, Miller has made an algorithm that is given in [25] and its complexity is only polynomial time.

Theorem 3.3.1. *In the reconstruction phase of a secret, probability of distributing invalid shares from the participant is negligible.*

Proof. As we already have a lot of discussion on verification of shares by the combiner in the verification phase. Now, it is time to prove that how does the equation $\hat{e}(G, Y'_{ij}) = \hat{e}(Y_0, Y_{ij})$ make sure that the shares sent by the participants are valid. Here, we will make use of properties of bilinear pairing to prove the theorem.

$$\begin{aligned}
\hat{e}(G, Y'_{ij}) &= \hat{e}(Y_0, Y_{ij}) \\
\hat{e}(G, X_{ij}Y_0) &= \hat{e}(Y_0, Y_{ij}) && \because Y'_{ij} = X_{ij}Y_0 \\
\hat{e}(G, X_{ij}Y_0) &= \hat{e}(X_0G, X_{ij}G) && \because Y_0 = X_0G \quad \text{and} \quad Y_{ij} = X_{ij}G \\
\hat{e}(G, X_{ij}X_0G) &= \hat{e}(X_0G, X_{ij}G) && \because Y_0 = X_0G \\
\hat{e}(G, G)^{X_0X_{ij}} &= \hat{e}(G, G)^{X_0X_{ij}}
\end{aligned}$$

□

It is now clear from the above proof that if the share initially sent by the participant and the share later sent for the reconstruction do no match then the participant is a cheater. As the scheme is based on Shamir's secret sharing scheme which allows only authorized participants to reconstruct the secret so any unauthorized set of participants can not obtain the secret because they are not eligible to full fill the criteria of recovering the secret. But, there is no hard mathematical problem on which Shamir's scheme is based whereas the proposed scheme depends on ECDLP. Moreover, the degree one polynomial is just needed to be constructed unlike other schemes which use $(t-1)$ degree polynomial. This also reduced the computational complexity since only two points are required to reconstruct the polynomial, one point $r(1)$ is displayed on the notice board and the other point $r(b_j)$ can only be obtained when only authorized participants pool their shares together. Therefore, any information for secret K cannot be obtained by the participant who is not in the access structure or any unauthorized set cannot retrieve the secret value K. The computational cost of Lagrange interpolation is $O(nlog^2n)$. There are only four multiplication and an inverse computation required to reconstruct the degree one polynomial.

Parameters for multiple secrets

Further, the scheme can also be used to share several secrets. Let $K_1, K_2, ..., K_m$ be the multiple secrets to be shared then for each secret K_i there must be a polynomial $r_i(x)$ that reveals the corresponding secret value K_i. For multiple secret, public parameters on the bulletin board get the form as $r(1)_i, Y_{0i}$ and $B_{ji}, 1 \leq i \leq m$ along with some other parameters related to the particular secret. It is now clear that participant's share does not change for sharing multiple secret only more public parameters are needed. But, if a participant feels like having an insecure share then they can change their secret share by just sending a new pseudo share to the dealer. Once the new pseudo share is received, the dealer update it on the bulletin board and it does not affect others participants secret shadow. We also discuss a case when a new participant is added in the system. A new participant does not affect the shares of other participants and all the work dealer has to do on the bulletin board is in access structure and the public parameters. There is no secret communication happened between dealer and participants so there is no need to have a secure channel in the share distribution phase.

Computational cost of the scheme

Computational cost of the proposed scheme is also low. We are now going to set some notations to denote the time taken for the execution of each operation used in the scheme. Let n be the total number of participants and d be the number of participants in each qualified set. Let T_{ECPA} defines the time consumed for the 'n' addition of a point X of a elliptic curve, T_{BP} defines the time taken for the execution of bilinear pairing, T_{HASH} is defined to show the time consumed for hash function H and time taken for polynomial reconstruction is defined by T_R.
To initialize the scheme, each participant and dealer compute their pseudo share. To compute the pseudo share, everyone has to use his secret share along with the generator of an elliptic curve addictive group. This takes $(n+1)$ point multiplications and the computational cost for it is $(n+1)T_{ECPA}$. The dealer has to apply

the hash function on the pseudo share of each participant and its computational cost is just dT_{HASH}. So, $O((n+1)T_{ECPA}+dT_{HASH})$ is the total cost for the execution of initialization and share distribution phase. In the final stage of the scheme where combiner verifies the shares and then reconstruct the secret, first, d participants in the authorized sets send their shares to the combiner by just doing a point multiplication. Its computational cost is dT_{ECPA}. The combiner then applies the hash function on the received shares which has a computational cost of dT_{HASH}. Dealer only has to do two pairing operations for verifying the shares and that takes computational cost of $2T_{BP}$ and finally using the Lagrange interpolation, secret is revealed. This is achieved by T_R computational cost. So, the computational cost for the final stage is $dT_{ECPA} + dT_{HASH} + 2T_{BP} + T_R$ .Hence, the computational cost for overall scheme is $O(((n + 1) + d)T_{ECPA} + 2dT_{HASH} + 2T_{BP} + T_R)$. The XOR operation and polynomial evaluation do not cost much and hence ignored in the computational cost.

3.4 Summary

This chapter explains the scheme of A. Sreekumar and V.P. Binu [2] which was proposed in 2017. It is a secure and efficient secret sharing scheme which is based on elliptic curve and pairing. Elliptic curve is used in the scheme to reduce the computations and because of its good security property. The scheme does not rely on the dealer for collecting the shares and hence participants are made free in choosing their shares. Degree one polynomial has also reduced the computations without compromising the security of secret sharing. Bilinear pairing sets up a platform for the combiner to verify the shares for reconstruction of the secret. In the end, we discussed the security of the scheme and analyzed its computational cost for evaluating each phase. Overall, it is good secret sharing scheme that provides a good security to a secret message.

Chapter 4

Modification for improving the security of the proposed scheme

In this chapter, we are going to look into some weak points related to the security of the proposed scheme [2] and then try to tackle them. The strong points with respect to the security of the scheme have been mentioned in Chapter 3. Taking those points into account and proposing some security related improvements, we try to improve the security. At the end of this chapter concluding remarks to wrap up the scheme are also mentioned.

4.1 Security loopholes

In the previous chapter, we explained a scheme [2] that is based on elliptic curve and pairing. We also discussed the security of the scheme and observed that how did it improve the security and computational complexity. Many secret sharing schemes in the literature have a problem in the share distribution stage where dealer distributes the share to participants but then to tackle this difficulty, a scheme [20] based on finite field and DLP is proposed that allowed participants to choose their shares without any dealer dependent procedure. This step improved the security of the share distribution phase and the same share can further be used for retrieving many secrets without compromising the security. Many researcher have also worked to improve the security of the reconstruction phase when participants submit their shares to the combiner there was a great risk that a cheater or an attacker sends a fake share to the combiner in order to cheat or to attack the secret. The idea of bilinear pairing is then used to overcome this problem so that a combiner can verify the shares of the participants and avoid any chance of cheating. Remember, it all started when Shamir proposed a scheme to share a secret but since then we have been in a state of improving the scheme more and more so that it could be made more secure and efficient to implement. We have seen some improvements in the share distribution phase and share reconstruction phase. Over the years, researchers have also worked to reduce the computational cost and complexity of secret sharing schemes. Now, we are going to look into a couple of points in the proposed scheme [2] where there is still a chance for attackers to deceive the participants and combiner and we know that if an attacker successfully deceives them then our secret will no longer be secured. We analyzed the following security loop-holes in the scheme:

1. In the reconstruction phase of Section 3.2.4 when participants deliver Y'_{ij} to combiner there can be an intruder, who sits between combiner and participants to get Y'_{ij}'s from participants and if an attacker successfully collects all the Y'_{ij}'s then it will be just a piece of cake for him to obtain the secret. As Shamir's secret sharing scheme says; when sufficient number of shares are combined, secret is revealed. This problem occurs because participant cannot authenticate the combiner and hence the attacker can get the benefit of it.

2. There is another problem in the reconstruction phase of Section 3.2.4 that is the scheme does not allow participants to verify that the secret K given back to them from the combiner after the reconstruction was the actual one. By actual secret, we mean the secret which was initially distributed by the dealer that is K. To make our claim more strong we denote the secret to be shared by K and the recovered secret by K'. Now, suppose the complete

scheme works smoothly and combiner in the reconstruction phase retrieves the secret K' and reveals it. There might be a dishonest combiner who does not give the actual secret K back after its reconstruction because there is no criterion set in the scheme which verifies that the recovered secret K' and K are same. This can also lead to a serious problem in secret sharing scheme where we are not getting the actual secret K instead getting a fake/wrong secret.

There must be set some criteria which help to improve the scheme. Participants must be able to authenticate combiner in order to deliver Y'_{ij}'s to the right combiner otherwise the scheme is not secure. There must also be a mechanism that verifies that the reconstructed secret K' is correct and did not alter it after it is reconstructed. The solutions to these problems are discussed in the next section.

4.2 Countermeasures

In this section, the weak points which we mentioned in the previous section are going to be tackled. The first problem is in the reconstruction phase 3.2.4 where participants can not authenticate the combiner. So an attacker can pretend to be a combiner in order to cheat the participants and collect all the Y'_{ij}'s from them to obtain the secret K. So, we have to find a way out of this issue and make the scheme more secure. All the research in the past have been done to improve the security as well as the computational cost of the scheme and if such points in the reconstruction phase 3.2.4 of [2] will not be tackled then the security cannot be improved and an attacker can take the benefit of it in order to know about the secret K. Once the secret K is known to a person who is not intended to know about it, there can be a huge disastrous occur for us. Now, we present a way out of this problem.

Countermeasure 4.2.1.

We are going to use the idea of public key cryptography and see how does it overcome the security loophole 1.

Let the public key of the combiner be PK_C and secret key of the combiner be SK_C. Let a secure asymmetric algorithm be $\mathcal{E}$ that uses public key PK_C and secret key SK_C of the combiner to encrypt and decrypt the data respectively. Then, the following steps can improve the security:

1. In share distribution phase 3.2.3 when the secret K is distributed. Dealer D publishes the public key PK_C of the combiner along with other public parameters on the bulletin board.

2. Participants then use the public key PK_C of the combiner available on the bulletin board and algorithm $\mathcal{E}_{PK_C}$ to send the encrypted $\mathcal{E}_{PK_C}(Y'_{ij})$ for retrieving the secret K in the reconstruction phase 3.2.4.

3. When combiner receives the encrypted Y_{ij} he can decrypt it using his secret key SK_C via algorithm $\mathcal{E}_{SK_C}$.

4. When all the shares Y'_{ij}'s are received, combiner retrieves the secret K.

Basically, we use the idea of public key cryptography where publicly available key of the recipient is used to encrypt a message. Once the encryption is done, the ciphertext is sent to the recipient. Then the receiver can use his private key to decrypt the message. Here, participants send Y'_{ij} using the public key of the combiner PK_C and algorithm $\mathcal{E}$. Then the combiner can decrypt the received information using his private key SK_C and algorithm $\mathcal{E}$ to obtain Y'_{ij} and then he can use these decrypted values to recover the secret K. The use of public key cryptography avoids the use of private channel for the communication between the participants and combiner. It also makes participants eligible to authenticate the combiner. In the proposed scheme, participants deliver their shares to the designated combiner via a secure channel but we implemented the idea of public key cryptography to avoid the use of a secure channel. There can be cases where use of secure channel is not possible so our counter-measure addresses the problem of communication through a secure channel between combiner and participants.

The second issue in the scheme occurs when a dishonest combiner does not give participants original secret K. There is no criterion set in this scheme which enables participants to confirm that the secret after its reconstruction is returned in its original form. It is very natural to have a dishonest combiner who cheats with participants. To make the scheme more secure and avoid any dishonesty from the combiner following improvement is made:

Countermeasure 4.2.2.

A secure function can be used here to check the validity of the reconstructed secret K'. The following steps are included:

1. In the secret distribution phase 3.2.3, dealer D applies a secure hash function H on the secret and publish the hash value $H(K)$ of the secret K on the public notice board. This will not affect the security of the secret K.

2. Once the combiner recovers the secret K, anyone who knows the recovered secret K' can now check its validity by just applying hash function on it.

3. If the hash value of the recovered secret K' matches the previously available hash value of the secret K then the secret revealed by the combiner is correct else the combiner is trying to cheat by showing a fake secret. Participants just need to check
$$H(K) = H(K').$$
If the above equation does not hold then the recovered secret is fake.

This step does not allow a dishonest combiner to cheat with participants and also helps the scheme to run smoothly. Note that, there must be used a collision resistant hash function e.g. SHA-512 [12]. It can now be publicly verified whether the recovered secret K' displayed is original or fake.

Overall, from the initialization of the scheme to the reconstruction of the secret, the scheme has become more secure. The security of the scheme mainly depends on the hard problem of ECDLP. The use of a secure channel for communication between combiner and participants is avoided using the idea of public key cryptography.

Moreover, the scheme is now more practical as the participants can now verify the recovered secret. Our counter-measures address the available loopholes in the scheme and improves the security of the scheme.

4.3 Concluding remarks

In this thesis we thoroughly revised the proposed scheme [2] and figured out a couple of security loopholes in Section 4.1. We then overcome these difficulties in Section 4.2. The following concluding remarks wrap up the revised scheme and shows the important characteristics of it.

- It is a multi secret sharing scheme where a same share can be used to reconstruct several secrets.
- Participants do not depend on dealer for their secret shares. They choose their secret shares themselves.
- First degree polynomial reduces the computational complexity.
- Bilinear pairing helps the combiner to check the validity of secret shares of participants and avoids being deceived by an intruder or participants.
- Publishing public key of the combiner on the public bulletin board avoids the use of a secure channel and also helps them to check the authenticity of the combiner.
- Publishing hash value of the secret on the public bulletin board helps participants to verify the reconstructed secret.

The revised scheme is more secure as the use of public key cryptography and a secure hash function H do not require the scheme to rely on any secret channel and verification of the secret can also be done efficiently. All the previously mentioned parameters will be retained on the public bulletin board along with public key PK_C and $H(K)$, and no one can cheat participants now.

Bibliography

[1] Ben-Or, M., Goldwasser, S., and Wigderson, A. (1988). Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 1–10, New York, NY, USA. ACM.

[2] Binu, V. P. and Sreekumar, A. (2017). Secure and efficient secret sharing scheme with general access structures based on elliptic curve and pairing. *Wireless Pers Communication*, 92:1531–1543.

[3] Blakley, G. R. (1979). Safeguarding cryptographic keys. In *Managing Requirements Knowledge, International Workshop on(AFIPS)*, volume 00, page 313.

[4] Das, A. and Adhikari, A. (2010). An efficient multi-use multi-secret sharing scheme based on hash function. *Applied Mathematics Letters*, 23(9):993 – 996.

[5] Diffie, W. and Hellman, M. (2006). New directions in cryptography. *IEEE Trans. Inf. Theor.*, 22(6):644–654.

[6] Eastlake, 3rd, D. and Jones, P. (2001). Us secure hash algorithm 1 (sha1).

[7] El Gamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. In *Proceedings of CRYPTO 84 on Advances in Cryptology*, pages 10–18, New York, NY, USA. Springer-Verlag New York, Inc.

[8] Enge, A. (2001). A general framework for subexponential discrete logarithm algorithms in groups of unknown order. In *Finite Geometries*, pages 133–146. Springer.

[9] Fatemi, M., Eghlidos, T., and Aref, M. (2009). A multi-stage secret sharing scheme using all-or-nothing transform approach. In Qing, S., Mitchell, C. J., and Wang, G., editors, *Information and Communications Security*, pages 449–458, Berlin, Heidelberg. Springer Berlin Heidelberg.

[10] Frey, G. and Rck, H.-G. (1994). A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Mathematics of Computation*, 62(206):865–874.

[11] Girault, M., Cohen, R., and Campana, M. (1988). A generalized birthday attack. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 129–156. Springer.

[12] Gueron, S., Johnson, S., and Walker, J. (2011). Sha-512/256. In *Proceedings of the 2011 Eighth International Conference on Information Technology: New Generations*, ITNG '11, pages 354–358, Washington, DC, USA. IEEE Computer Society.

[13] Harn, L. and Lin, C. (2009). Detection and identification of cheaters in (t, n) secret sharing scheme. *Des. Codes Cryptography*, 52(1):15–24.

[14] He, J. and Dawson, E. (1995). Multisecret-sharing scheme based on one-way function. *Electronics Letters*, 31:93–95(2).

[15] Hellman, M. (1977). An extension of the shannon theory approach to cryptography. *IEEE Transactions on Information Theory*, 23(3):289–294.

[16] Hua, S. and Aimin, W. (2010). A multi-secret sharing scheme with general access structures based on elliptic curve. In *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, volume 2, pages V2–629–V2–632.

[17] Ito, M., Saito, A., and Nishizeki, T. (1989). Secret sharing scheme realizing general access structure. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 72(9):56–64.

[18] Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):203–209.

[19] Lee, N.-Y. and Hwang, T. (2001). New multistage secret sharing scheme based on factorization problem. 17:525–529.

[20] Liaojun, P., Huixian, L., and Yumin, W. (2006). An efficient and secure multi-secret sharing scheme with general access structures. *Wuhan University Journal of Natural Sciences*, 11(6):1649–1652.

[21] Lidl, R. and Niederreiter, H. (1986). *Introduction to Finite Fields and Their Applications*. Cambridge University Press, New York, NY, USA.

[22] Liu, D., Huang, D., Luo, P., and Dai, Y. (2008). New schemes for sharing points on an elliptic curve. *Comput. Math. Appl.*, 56(6):1556–1561.

[23] Menezes, A. J., Okamoto, T., and Vanstone, S. A. (1993). Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5):1639–1646.

[24] Menezes, A. J., Oorschot, P. C. V., Vanstone, S. A., and Rivest, R. L. (1997). Handbook of applied cryptography.

[25] Miller, V. S. (1986a). Short programs for functions on curves. In *IBM THOMAS J. WATSON RESEARCH CENTER*.

[26] Miller, V. S. (1986b). Use of elliptic curves in cryptography. In Williams, H. C., editor, *Advances in Cryptology — CRYPTO '85 Proceedings*, pages 417–426, Berlin, Heidelberg. Springer Berlin Heidelberg.

[27] Mousa, A. and Hamad, A. Evaluation of the rc4 algorithm for data encryption.

[28] Niels, F., Bruce, S., and Tadayoshi, K. (2015). *The Secure Channel*, chapter 7, pages 99–114. Wiley-Blackwell.

[29] Odlyzko, A. M. (1985). Discrete logarithms in finite fields and their cryptographic significance. In *Proc. Of the EUROCRYPT 84 Workshop on Advances*

in Cryptology: Theory and Application of Cryptographic Techniques, pages 224–314, New York, NY, USA. Springer-Verlag New York, Inc.

[30] Paar, C. and Pelzl, J. (2009). *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer Publishing Company, Incorporated, 1st edition.

[31] Rabin, M. O. (1983). Randomized byzantine generals. In *24th Annual Symposium on Foundations of Computer Science (sfcs 1983)*, pages 403–409.

[32] Rivest, R. (1992). The md5 message-digest algorithm.

[33] Rivest, R. L., Shamir, A., and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126.

[34] Shamir, A. (1979). How to share a secret. *Commun. ACM*, 22(11):612–613.

[35] Stadler, M. (1996). Publicly verifiable secret sharing. In Maurer, U., editor, *Advances in Cryptology — EUROCRYPT '96*, pages 190–199, Berlin, Heidelberg. Springer Berlin Heidelberg.

[36] Stallings, W. (2002). The advanced encryption standard. *Cryptologia*, 26(3):165–188.

[37] Stallings, W. (2010). *Cryptography and Network Security: Principles and Practice*. Prentice Hall Press, Upper Saddle River, NJ, USA, 5th edition.

[38] Tompa, M. and Woll, H. (1989). How to share a secret with cheaters. *Journal of Cryptology*, 1(3):133–138.

[39] Wei, Y., Zhong, P., and Xiong, G. (2008). A multi-stage secret sharing scheme with general access structures. In *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1–4.

[40] Yong-Jun, G., Li-Zheng, G., and Ming-Hui, Z. (2014). Improved multi-secret sharing scheme based on one-way function. 12.