CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

Generalized Blind Signcryption Scheme For E-Voting System Based on Elliptic Curve Cryptography

by

Tooba Muhammad

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

in the

Faculty of Computing
Department of Mathematics

2023

To my loving family for their persistent support and encouragement

CERTIFICATE OF APPROVAL

Generalized Blind Signcryption Scheme For E-Voting System Based on Elliptic Curve Cryptography

by

Tooba Muhammad

(MMT213035)

THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Organization
(a)	External Examiner	Dr. Munazza Naz	FJWU, Rawalpindi
(b)	Internal Examiner	Dr. Abdul Rehman Kashif	CUST, Islamabad
(c)	Supervisor	Dr. Rashid Ali	CUST, Islamabad

Dr. Rashid Ali
Thesis Supervisor
October, 2023

Dr. Muhammad Sagheer
Head
Dept. of Mathematics
October, 2023

Dr. M. Abdul Qadir
Dean
Faculty of Computing
October, 2023

Author's Declaration

I, **Tooba Muhammad** hereby state that my MPhil thesis titled "**Generalized Blind Signcryption Scheme For E-Voting System Based on Elliptic Curve Cryptography**" 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 MPhil Degree.

(Tooba Muhammad)

Registration No: MMT213035

Plagiarism Undertaking

I solemnly declare that research work presented in this thesis titled "**Generalized Blind Signcryption Scheme For E-Voting System Based on Elliptic Curve Cryptography**" is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly 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 MPhil Degree, the University reserves the right to withdraw/revoke my MPhil 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.

(Tooba Muhammad)

Registration No: MMT213035

Acknowledgement

Firstly, I would like to thank **Allah Almighty**, the most Merciful and most Gracious, who gave me enough strength to preserve through all the highs and lows. His blessings provided me with the power to successfully complete my thesis work. Next, I extend my gratitude to my research supervisor **Dr. Rashid Ali** who guides me whenever needed, motivates me and provides constant support throughout the journey of completing this thesis. His ability to explain complex concepts in a clear and simple manner is invaluable.

Primarily, my appreciation goes to my family. Without the support and love of my amazing family, this achievement would not have been possible. I am particularly thankful to my mother, who consistently motivates and stands by me. Her words of wisdom, relentless beliefs in my abilities have given me the strength and courage to overcome challenges and strive to achieve my goals. I consider myself fortunate to have a family that supports me throughout this journey and believes in my aspirations.

(Tooba Muhammad)

Registration No: MMT213035

Abstract

As more aspects of our daily lives migrated to the digital sphere, the shift towards digital voting platforms becomes a sensible progression. Even though, some of the private institutions or companies have accepted the e-voting but public elections did not adopt e-voting due to some security concerns. For a good e-voting scheme, accuracy, data anonymity, and untraceability are the most important characteristics. Many cryptographic schemes are introduced which offers such properties. Unlike other schemes, elliptic curve cryptography is computationally less expensive scheme. Blind signatures offers the characteristics of data anonymity and untraceability. In this work, a generalized blind signcryption scheme based on elliptic curve cryptography is proposed. Generalized blind signcryption operates in three different modes: blind signcryption mode, blind signature only mode and encryption only mode. If the voter needs both authenticity and confidentiality, then the scheme operates in the blind signcryption mode. If the voter needs only confidentiality, then it operates in the encryption only mode, and if only authenticity is needed to the voter, then it operates in the blind signature only mode. The security analysis shows that the proposed scheme is safe against different cryptanalysis attacks. The proposed scheme gives the security features of blindness, integrity, unforgeability, untraceability, unlinkability, non-repudiation and forward secrecy. The security of the proposed scheme depends on the elliptic curve discrete logarithm problem and the hash function random oracle property. The security analysis of the scheme shows that the scheme is efficient. The computational cost of different modes of the scheme demonstrates that the scheme is cost efficient.

Contents

List of Figures

List of Tables

Abbreviations

ECC	Elliptic Curve Cryptography
RSA	Rivest-Shamir-Adleman
ECDLP	Elliptic Curve Discrete Logarithmic Problem
ECDSA	Elliptic Curve Digital Signature Algorithm
AES	Advanced Encryption Standard
DES	Data Encryption Standard
BSS	Blind Signcryption Scheme
GBSS	Generalized Blind Signcryption Scheme

Symbols

c	Ciphertext
m	Message
k_{e^*}	Secret key
E	Encryption algorithm
D	Decryption algorithm
$\mathcal{O}$	Point at infinity
$\mathbb{R}$	Set of real numbers
$\mathbb{Z}$	Set of integers
$\mathbb{F}$	Field
G	Base point of ECC

Chapter 1

Introduction

In todays developing world, securing a message or any confidential information from the unauthorized access or potential adversaries has become increasingly challenging. Sending messages to the other person without revealing to any other unintended person is very challenging. The demand for information and electronic services continue to grow steadily. The importance of ensuring the security of information and electronic systems in our daily lives cannot be overstated. Through cryptography, a person can transmit a message securely to the other person. Around 1900 B.C., cryptography was used for the first time in documented form [1]. Cryptography means to communicate in a secure manner. Many cryptographic techniques are used for securing information or data against threats. For this purpose, first encrypt the original message using some specific algorithm called encryption algorithm, i.e. convert the original message to the unreadable form called ciphertext. The ciphertext is then sent to the receiver. The intruder cannot find/guess the original message without knowing any key or any hints from this ciphertext. The key used is kept confidential and known only to the sender and the receiver. The receiver then deciphers the ciphered message using a specific algorithm and restores the original message with the help of a decryption key. In any medium, using cryptography provides integrity, authentication, confidentiality and non-repudiation with other basic properties [1]. For providing these properties, one is to ensure that he is using a secure algorithm.

1.1 Cryptology

Cryptology is the art and science of secure communication over insecure channel [2]. It is the art of breaking and designing a secure system. The term 'cryptography' refers to the designing of secure systems, while the term 'cryptanalysis' refers to the breaking of such cryptosystem.

Cryptography

Cryptography means secret communication. Cryptography is the study of 'mathematical' techniques to resolve the security issues of privacy and authentication [3]. The purpose of using cryptography is to secure your message or data from unauthorized person. Around 50 BC, a person named Julius Caesar [4] introduced the scheme called caesar cipher. In this scheme, to cipher the message, shifts the alphabets to the right or left using modular arithmetic. To decipher, the message is shifted back to the left or right. This transformation can also be done through computers by indicating the alphabets digitally and using modular arithmetic operations. This scheme has no secrecy at all because of exhaustive cryptanalysis can use 25 keys that can be easily carried out. Later on, many other schemes have been presented and used over time to send secure messages such as, Polyalphabetic cipher [5], Playfair cipher [6], Hill cipher [5] etc. With the passage of time, these schemes turn out to be insecure against cryptanalysis attacks.

Cryptography is of two types: symmetric key cryptography and asymmetric key cryptography. In symmetric key cryptography, only one key is used for both encryption and decryption. It is also known as secret key cryptography. Symmetric key cryptography is good at managing the large files. In symmetric key cryptography, the difficulty is to transmit the secret key with the sender and receiver. The examples of symmetric key cryptography are Data Encryption Standard (DES) [7], Advanced Encryption Standard (AES) [8].

While, in asymmetric key cryptography, two keys are used. One key is used for encryption and the other is used for decryption. Asymmetric key crptography is

also called public key cryptography. In 1976, Diffie and Hellman [9] gave the idea of asymmetric key cryptography. The examples of asymmetric key cryptography is Rivest Shamir Adleman (RSA) [10], ElGamal Cryptosystem [11], and Elliptic Curve Cryptosystem (ECC) [12]. One of the keys in asymmetric key cryptography is referred to as the public key, which is made available to the general public, and the other key is referred to as the private key, which is preserved confidential.

Cryptanalysis

For analyzing the strength of a scheme, cryptanalysis schemes are used. In addition, for breaking a system, cryptanalysis is done to find the weaknesses of the scheme. Ciphers, codes, and encrypted text can be studied and decrypted using a method known as cryptanalysis, which involves analyzing and exploiting weaknesses in cryptographic techniques. In short, cryptanalysis is the method in which we find out the plaintext without knowing the decryption key [13]. The attacks will be successful only if the scheme is weak or is vulnerable to different attacks. The main goal of cryptanalysis is to collect a maximum amount of information about the plaintext so that a cryptanalyst can break the system and get the ciphertext of plaintext and the keys to reveal other messages that are decrypted using the same key [14]. Cryptanalysis often involves mathematical techniques to analyze and break cryptographic systems. In the digital age, cryptanalysis is crucial for evaluating the security of cryptographic systems in applications like secure communications, data protection and online transactions. Cryptanalysis focuses on unravelling the secrets concealed within encrypted information.

Cryptanalysis attacks can be categorized based on the type of information the attacker has access to. There are many cryptanalysis attacks that are used to check the vulnerability of different schemes. These cryptanalysis attacks are classified as: Ciphertext only attack (COA), Known plaintext attack (KPA), Chosen plaintext attack (CPA), Chosen ciphertext attack (CCA), Man in the middle attack (MITM), Man at the end attack (MATE) and Brute force attacks. These attacks are further explained in Section 2.6.

1.2 Blind Signcryption

Digital signatures are the signatures that are attached to the document that the sender has to send to some other party. Digital signatures are used for the authenticity of the document. If the signatures are attached to the document, the receiver will be sure that the document was sent by a legitimate user. Digital signatures are asymmetric key cryptography, that uses two keys, public and a private key. Some schemes of digital signatures offer non-repudiation property, by which a signer/sender cannot claim that they did not sign the document [15]. Digital Signatures are first introduced by the Diffie and Hellman [3] in 1976.

Signcryption scheme is first proposed by Zheng [16] in 1997. Signcryption is an asymmetric key cryptography, which is a way more effective than signature-then-encryption. By combining the signature and encryption into a single step, the computational cost and communication overhead are reduced compared to the signature-then-encryption scheme. Zheng [17] proposed a scheme of an elliptic curve based signcryption scheme that saves 58% of computational expenditures and 40% of communication expenditures. In signcryption scheme, both signature and encryption are combined into a single step. This scheme ensure the authenticity and confidentiality of a message between sender and receiver. An application of the signcryption scheme is the secure transmission of emails. Signcryption schemes provides the property of efficiency, correctness, confidentiality, unforgeability, non-repudiation, integrity, public verifiability, forward secrecy [18].

Blind signcryption schemes are the extension of digital signatures. Blind signature schemes are a type of digital signature that generates a signature on the message and does not know any information about a message or the sender, and generates only one signature on a single interchange with the requester. Blind signcryption scheme is first introduced by Chaum [19] in 1984. Blind signcryption scheme is used for the security and privacy of sender from the signer and receiver in electronic voting and electronic cash payment systems [19]. In e-voting system many properties like authentication, confidentiality, integrity and blindness are needed. In the blind signcryption schemes, the message is blinded before being

sent to the signer or receiver. So, the signer cannot see the message content and sign the message without knowing the content. Blind signatures are frequently used when the sender of the message and the signer are both different individuals or parties in privacy-related protocols. Examples are e-voting and electronic cash payment systems. Blind signatures provide the characteristic of unlinkability in which a signer cannot link the messages of the signer with their previous messages. To check if the signer or sender of a message is legitimate or not, signatures can also be verified by a third judge or verifier. By sending the signature parameters to the third verifier, he/she can verify the authenticity of the signature. Elliptic curve cryptography is used in the e-voting system because it is less expensive, less key size and its security depends on the elliptic curve discrete logarithm problem (ECDLP), which is infeasible to compute.

1.3 Analysis of the Literature

In 1983, Chaum [19] first introduced the idea of blind signature scheme. In his scheme, he gave the idea of untraceable payments using blind signatures. This scheme makes it possible to implement untraceable payment systems that provide better auditability and control. Further, the scheme provides more personal privacy at the same time. In this scheme, generating signatures for multiple documents requires numerous computations; therefore, Chaum [20] presented another scheme for multiple signatures with maximum number of calculations. In 1994, Brand [21] proposed a new scheme called the restrictive blind signature scheme together with the so-called representation problem in groups of prime order. This scheme results in extremely efficient online cash systems that can be expanded to wallets with observers under the strictest privacy constraints for essentially no additional cost. Stern and Pointcheval [22] introduced a new blind signature scheme based on the factorization problem. The first scheme proved secure relative to factorization. This scheme offers unforgeability under even a parallel attack. Fan and Lei [23] proposed a partially blind signature scheme for electronic cash system. This scheme contains 98% less computation for requester. In this scheme,

the requester performs modular addition and modular multiplication to obtain and validate a signature. The scheme is suitable for mobile clients and smart card applications. Chang and Huang [24] proposed a scheme for untraceable electronic cash system, which is a user-efficient and a signer-efficient fair blind signature scheme. The signer or banker cannot establish a connection between a signature and the instance of the signing protocol, that produced the signature without the assistance of a government or the judge, when the unlinkability property is exploited improperly. This scheme cuts down the computational workload of an online judge. The scheme is appropriate for the user's limited computing power. Nikooghadam and Zakerolhosseini [25] presented an untraceable blind signature scheme. The security of the proposed scheme is based on the complexity of discrete logarithm over an elliptic curve. Awasthi and Lal [26] proposed an efficient scheme for sensitive message transmission blind signcryption scheme. In this scheme, the signcryption and blind signature features are combined. This scheme provides anonymity, untraceability and unlinkability but lacks public verifiability and involves high computational cost. Yu and He [27] proposed an efficient blind signcryption scheme which is based on the discrete logarithmic problem (DLP). This scheme offers public verifiability, confidentiality, integrity, non-repudiation and unforgeability. The scheme has a relatively high computational cost. Chakraborty and Mehta [28] proposed a blind signature scheme that is based on the elliptic curve discrete logarithm problem (ECDLP) and cryptographic hash functions. The proposed scheme has three participants; the requester, the signer and the verifier and consists of two protocols: the signing protocol and the verification protocol. In this scheme, the signer blinded the message twice. This scheme offers blindness and nonforgeability. Dhanashree and Agrawal [29] presented a scheme using zero knowledge protocol. The requester/sender can verify his/her identity. The scheme is based on the elliptic curve cryptosystem. In elliptic curve cryptosystem, solving discrete logarithm problem (DLP) is difficult. The elliptic curve cryptography is used while applying blind signature scheme and the zero knowledge protocol is apply when the requester/sender identity is needed. The zero knowledge protocol also used in this scheme for hiding the identity of the sender. This scheme offers

the properties of anonymity and untraceability. Ullah et al [30] proposed a scheme that is based on the elliptic curve cryptography. This scheme provides the characteristics of confidentiality, integrity, unforgeability and non-repudiation. This scheme is appropriate for mobile phone voting and mobile commerce.

The notion of generalized signcryption scheme was termed by Yiliang and Xiaoyuan [31] in 2006, which includes the function of encryption, signature and signcryption in a single primitive. The idea is that using a special "signcryption", one not only can simultaneously get confidentiality and authentication, but also obtain confidentiality or authentication alone. This is called generalized signcryption. Han and Gui [32] proposed an adaptive secure multicast in wireless networks in 2009. They described a multireceiver generalized signcryption scheme and applied it to wireless multicast communication. In 2010, Yu et al. [33] proposed a provable secure generalized signcryption scheme. They proposed an improved generalized signcryption scheme based on ECDSA. In 2011, Kushwah and Lal [34] proposed a more efficient identity-based generalized signcryption scheme. They also simplify the security notions for identity based generalized signcryption and prove the security of the proposed scheme. In 2014, Zhou et al. [35] proposed a certificateless generalized signcryption scheme that can resist a malicios-but-passive key generation center (KGC) attack. Later, in 2016, Zhou et al. [36] extended generalized signcryption scheme and introduced generalized proxy signcryption scheme by considering sharing the same key pair and algorithm between the proxy signature and proxy signcryption. In 2016, Zhang et al. [37] proposed a lightweight certificateless generalized signcryption scheme and applied it to a mobile health system.

1.4 Thesis Contribution

In this thesis, the blind signcryption scheme of Waheed et al [38] is reviewed. In this scheme, elliptic curve cryptography (ECC) is used for encryption and decryption of the message. The elliptic curve cryptography is a low cost scheme with less key size of 160 bits as compared to the ElGamal cryptosystem and the

RSA cryptosystem where the key size is 2048 bit and 1024 bit. The three participants participated in this scheme: the requester, the signer and the verifier. The presented scheme provides some additional properties like forward secrecy, unlinkability, and nonrepudiation to the fundamental properties of confidentiality, authenticity, integrity, and unforgeability. They claim that their scheme, compared to the other schemes, is more effective in computational and communicational costs. The scheme security is depend on the ECDLP and it works best in the environment, which has little resources, such as mobile commerce transactions, and any citizen's portal. In this thesis, the scheme is extended to the generalized blind signcryption scheme. Generalized blind signcryption scheme works in three modes, blind signcryption mode, blind signature-only mode, encryption-only mode. Three participants are involved in this scheme; the requester/voter, the signer/polling station and the polling server/verifier. The voter/requester is the one who wants to cast the vote, the signer is the one who generates the signature blindly on the message, the verifier, who receives the vote and after verification accepts the vote. The proposed scheme gave same properties as the presented scheme [38]. The security of the proposed scheme depends on the hardness of elliptic curve discrete logarithm problem (ECDLP). The scheme is resistant to various cryptanalysis attacks.

1.5 Thesis structure

The structure of the rest of the thesis is described as:

In **Chapter** 2, the mathematical models that are related to our scheme are discussed, including basic definitions of cryptography, digital signatures, signcryption and blind signcryption. Additionally, the section examines various cryptanalysis attacks that are used in our scheme.

In **Chapter** 3, a detailed analysis of the blind signcryption scheme of Waheed et al [38] based on elliptic curve cryptography is presented. Security analysis of the scheme is comprehensively examined.

In **Chapter** 4, The proposed generalized blind signcryption scheme is presented.

Correctness of the scheme is also examined. A toy example to illustrate the scheme is also presented.

In **Chapter** 5, The analysis of the generalized blind signcryption scheme is discussed in detail. Also the computational cost is provided and compared with the other scheme.

Chapter 2

Preliminaries

In this chapter, the fundamental definition within algebra, number theory and cryptography are introduced. It additionally explores the mathematical framework and the associated terminology in the field of cryptography.

2.1 Mathematical Background

Definition 2.1.1. A non-empty set $\mathbb{G}$ together with a binary operation $*$ is known as a group $(\mathbb{G}, *)$ if the following conditions are satisfied [39]:

1. Closure Property: For all $a, b \in \mathbb{G}$,

$$a * b \in \mathbb{G}.$$

2. Associative Property: For all $a, b, c \in \mathbb{G}$,

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

3. Identity Property: For all $a \in \mathbb{G}$, there exists an element e in $\mathbb{G}$, such that,

$$a * e = e * a = a.$$

4. Inverse Property: For all $a \in \mathbb{G}$, there exists an element $a' \in \mathbb{G}$ such that

$$a * a' = a' * a = e.$$

Moreover, for all $a, b \in \mathbb{G}$, the group is called commutative or abelian if it satisfies the following:

$$a * b = b * a.$$

Example 2.1.2. To illustrate the concept of group, consider the following examples:

1. The set of real numbers, $\mathbb{R}$, is a group under addition of real numbers.
2. The set $\mathbb{R}$ is also a group under multiplication of real numbers.
3. The set of integers $\mathbb{Z}$ is a group under addition of integers, but is not a group under the multiplication of integers.
4. The set $\mathbb{Z}_n$ of integers modulo n is a group under addition of integers modulo n, but it is not a group under multiplication modulo n if n is not prime. That is, $\mathbb{Z}_p$ is a group under multiplication of integer modulo p.

Definition 2.1.3. A set $\mathbb{F}$ under two binary operations $(+, *)$ is known as a field $(\mathbb{F}, +, *)$, if the following axioms are satisfied [40]:

1. Set $\mathbb{F}$ is abelian group under addition.
2. A non-zero elements of set $\mathbb{F}$ form an abelian group under multiplication.
3. Distributivity Property: For all $a, b, c \in \mathbb{F}$

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

Example 2.1.4. To substantiate the concept of field, consider the following example:

1. The set of rational numbers, $\mathbb{Q}$, is a field under addition of rational numbers.

2. The set of rational numbers $\mathbb{Q}$ is also a field under multiplication of rational numbers.

3. The set of complex numbers, $\mathbb{C}$, is a field under addition with additive identity.

4. The set $\mathbb{C}$ is also a field under multiplication with multiplicative identity.

5. The set of integers $\mathbb{Z}$ is not a field because multiplicative inverse does not exist in the set of integer $\mathbb{Z}$.

Definition 2.1.5. A set with a finite number of elements is a finite field, also called a Galois field $GF(p)$ [41]. "A field with order m only exists if m is a prime power, i.e., $m = p^n$, for some positive integer n and prime integer p, p is called the characteristic of the finite field."
In cryptography, the most important cases are:

1. $GF(p)(n-1)$

2. $GF(2^n)(p-2)$

Definition 2.1.6. One way trapdoor function is a one way function f, in which it is feasible to find the value in one way but it is infeasible to find it in the opposite way [42].

$$y = f(x)$$

In the above equation, if the information of x is given it is easy to find the value of y, but given y it is infeasible to find the value of x.

Definition 2.1.7. Suppose g is the generator of $\mathbb{Z}_p$ where p is the prime number. Finding x is difficult when y is known. i.e. Computing x from $y = g^x \mod p$. The process of finding the x is known as Discrete Logarithmic Problem [43]. Solving a discrete logarithm problem is hard.

Definition 2.1.8. Factorization of a number is defined as writing the number as a product of its prime factors. Such that, factorizing a number m into the product of two prime numbers a and b i.e., $m = ab$. The integer factorization

problem can be formally defined as: Given a composite number n, the problem is to find two integers x and y such that their product, $x\,y = n$ [41]. Factorizing a large number is quite difficult.

Definition 2.1.9. An integer-based arithmetic system is known as Modular Arithmetic. Modular arithmetic depends on the congruence relation. Suppose an integer p is our modulus under a binary operation. Suppose the integer is from $\{0, \ldots, p-1\}$. To find the modular value a, the number $b \mod p$ is the remainder when b is divided by p [44].

$$a \equiv b \mod p$$

Example 2.1.10. Some examples of modular arithmetic are as follows:

1. Let $a = 23, b = 73$ and $p = 37$

$$\begin{aligned} & ab \mod p \\ & (23)(73) \mod 37 \\ & 1679 \mod 37 \\ & 14 \mod 37 \end{aligned}$$

2. Let $a = 2, b = 5$ and $p = 3$

$$\begin{aligned} a &\equiv b \mod p \\ 2 &\equiv 5 \mod 3 \end{aligned}$$

Definition 2.1.11. An Extended Euclidean Algorithm is used to find modular inverses [4]. The steps for finding the inverse of a and b are as follows:
Input: $a \mod p$
Output: $a^{-1} \mod p$

1. Initialize: $(u, v, w) = (1, 0, p)$, $(r, s, t) = (0, 1, a)$.
2. If $t = 0$; return $\gcd(a, p) = w$, There is no inverse for $a \mod p$.
3. If $t = 1$; return $\gcd(a, p) = t$ and $s = a^{-1} \mod p$.

4. Now find the quotient Q using: $Q = w$ divided by t.

5. $(b1, b2, b3) = (u - Qr, v - Qs, w - Qt)$.

6. $(u, v, w) = (r, s, t)$.

7. $(r, s, t) = (b1, b2, b3)$..

8. Go to step 2.

Definition 2.1.12. "An elliptic curve is typically a two-space graph defined by the square roots of cubic equation" [45]. Elliptic curves can be defined over other field, such as the prime modulo integer field $GF(p)$, and also over extended fields generated by various bases, such as $GF(2^k)$.

Suppose an elliptic curve E based on $\mathbb{F}_p$ is illustrated as the cubic equation of the kind:

$$y^2 = x^3 + cx + d \tag{2.1}$$

where c and d are constant values based on the $E(\mathbb{F}_p)$. The equation is also known as Weierstrass equation. In terms of cryptography, the elliptic curve must necessitate non-singularity. Condition for the distinct single roots is:

$$4c^3 + 27d^2 \neq 0 \tag{2.2}$$

Fundamentally, it ensures that the curve has no vertices and self intersections. Geometrically, in a real elliptic curve, the curve exists over $\mathbb{R}$. However, if the field $\mathbb{F}$ is finite, such as $\mathbb{F}_p$, then we obtain a finite elliptic curve, which is a collection of a finite number of points satisfying the equation (2.3).
Suppose an equation defined in a finite modulo $p = 17$,

$$y^2 = x^3 + 7 \mod 17 \tag{2.3}$$

where $c = 0$ and $d = 7$.
The points (x, y) satisfying equation (2.3) are represented by scattered points on the graph, as shown in figure 2.1.

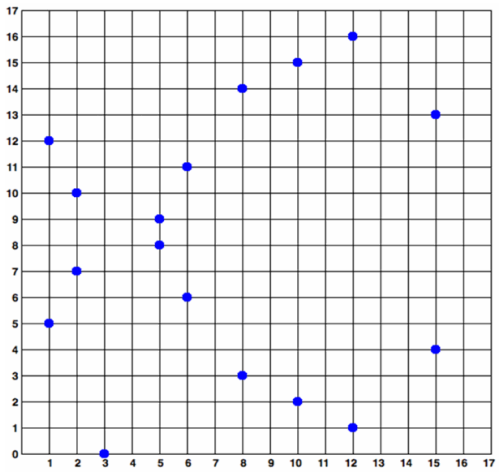

FIGURE 2.1: Elliptic Curve: $y^2 = x^3 + 7 \mod 17$

To verify the equation: $p = 17$, $c = 0$ and $d = 7$

$$4c^3 + 27d^2 \neq 0 \mod p$$

$$4(0) + 27(7)^2 \mod 17$$

$$14 \neq 0$$

The given elliptic curve in eq: (2.3) contain 17 points. Collectively with infinity point $\mathcal{O}$, they make a group with number of points = 18.

x	y^2	y^2 belongs to $\mathbb{F}_p$	$P_1(x, y)$	$P_2(x, y)$
0	7	-	-	-
1	8	5 , 12	(1,5)	(1,12)
2	15	7 , 10	(2,7)	(2,10)
3	0	0	(3,0)	-
4	3	-	-	-
5	13	8 , 9	(5,8)	(5,9)
6	2	6 , 11	(6,6)	(6,11)
7	10	-	-	-
8	9	3 , 14	(8,3)	(8,14)
9	5	-	-	-
10	4	2 , 15	(10,2)	(10,15)
11	12	-	-	-
12	1	1 , 16	(12,1)	(12,16)
13	11	-	-	-
14	14	-	-	-
15	16	4 , 13	(15,4)	(15,13)
16	6	-	-	-

TABLE 2.1: Elliptic Curve Points

For an effective and safe encryption system, elliptic curve cryptography performs effectively. The operations defined on elliptic curve are:

1. Point Doubling.
2. Point Addition.

Point Doubling

The point doubling is the scalar multiplication of points to itself.

$$S^n = nS = S \cdot S \cdot S \cdot \ldots \cdot S \tag{2.4}$$

Graphically

For the point $S(u, v)$ on elliptic curve, the coordinates of $2S$ are determined as follows:

1. Draw a tangent line at point S.
2. The line intersect at point Z of elliptic curve E.
3. Trace the reflection $Z^{'}$ of Z.
4. The point $Z^{'}$ is the point doubling.

The algebraic formulas to find the point doubling of point $S(u, v)$ are:

$$\alpha = \frac{3u_S^2 + c}{2v_S} \tag{2.5}$$

$$v_0 = v_S - \alpha u_S \tag{2.6}$$

To find the coordinates of point Z:

$$u_Z = \alpha^2 - 2u_S \tag{2.7}$$

$$v_Z = -(\alpha u_Z + v_0) \tag{2.8}$$

Example 2.1.13. Consider the elliptic curve:

$$y^2 = x^3 + 7 \mod 17 \tag{2.9}$$

$S = (15, 13)$ is the point of elliptic curve. Here $c = 0$ and $d = 7$ and $p = 17$. Then for the point doubling, find $S^2 = 2S$.

First, to calculate α put the values in the following formula (2.5):

$$\begin{aligned}
\alpha &= \frac{3u_S^2 + c}{2v_S} \\
&= \frac{3(15)^2 + 0}{2(13)} \mod 17 \\
&= \frac{3(225)}{26} \mod 17 \\
&= (675)(26)^{-1} \mod 17
\end{aligned}$$

Find inverse by using Extended Euclidean Algorithm.

$$\begin{aligned}
&= (12)(2) \mod 17 \\
\alpha &= 7 \mod 17
\end{aligned}$$

Then find v_0 by using formula (2.6):

$$\begin{aligned}
v_0 &= v_S - \alpha u_S \\
&= 13 - (7)(15) \mod 17 \\
v_0 &= 10
\end{aligned}$$

Further, find the points (u, v) of Z. To find the points of u_Z from the formula (2.7):

$$\begin{aligned}
u_Z &= \alpha^2 - 2u_S \\
&= (7)^2 - (2)(15) \mod 17 \\
&= 19 \mod 17 \\
u_Z &= 2 \mod 17
\end{aligned}$$

To find the points v_Z from the formula (2.8):

$$\begin{aligned} v_Z &= -(\alpha u_Z + v_0) \\ &= -((7)(2) + 10) \mod 17 \\ v_Z &= 10 \mod 17 \end{aligned}$$

So, the point doubling for $S^2 = 2 \cdot S = (2, 10)$.

Point Addition

By adding two points that are located on an elliptic curve we get new point. Following are the norms for point addition:

1. If $S + \mathcal{O} = \mathcal{O} + S = S$ here $\mathcal{O}$ is the point at infinity.
2. Inverse element: If $S^{'}(u, -v) = S(u, v)$ is reflected on x-axis and $S^{'}(u, -v) = -S$ then $S + S^{'} = 0$

Graphically

All points $S(u, v)$ located on elliptic curve. Suppose two points S and T, by adding these two points we get a new point. For graphically representation of point addition following are the steps to follow:

1. Draw a line between two points P_1 and P_2.
2. The Line intersect at some point P_3 of elliptic curve E.
3. The reflection of P_3 is $P_3{'}$ which is the addition of two points.

The algebraic formulas to find the point addition are:

$$\alpha = \frac{v_T - v_S}{u_T - u_S} \tag{2.10}$$

$$v_0 = v_S - \alpha u_S \tag{2.11}$$

To find the coordinates of point Z:

$$u_Z = \alpha^2 - u_S - u_T \tag{2.12}$$

$$v_Z = -(\alpha x_Z + v_0) \tag{2.13}$$

Example 2.1.14. Consider the elliptic curve:

$$y^2 = x^3 + 7 \mod 17 \tag{2.14}$$

$P(15, 13)$ is the point of elliptic curve. $S^2 = 2S = (2, 10)$, and $E_{17}(0, 7)$. Here $c = 0$, $d = 7$ and $p = 17$.

Compute $S^3 = 3S$ to find the point addition between two points S and T. Now $T = (15, 13)$ and $S = (2, 10)$.

At first, calculate the slope α by putting the values in the following formula (2.10):

$$\begin{aligned}
\alpha &= \frac{v_T - v_S}{u_T - u_S} \\
&= \frac{13 - 10}{15 - 2} \mod 17 \\
&= \frac{3}{13} \mod 17 \\
&= 3(13)^{-1} \mod 17
\end{aligned}$$

Using Extended Euclidean Algorithm calculate the inverse.

$$\begin{aligned}
&= 3(4) \mod 17 \\
\alpha &= 12 \mod 17
\end{aligned}$$

Next, calculate v_0 using the following formula (2.11):

$$\begin{aligned}
v_0 &= v_S - \alpha u_S \\
&= (10 - (12)(2)) \mod 17 \\
&= -14 \mod 17 \\
v_0 &= 3 \mod 17
\end{aligned}$$

Further, find the coordinate points of Z. To find the u_Z using the formula (2.12)

$$\begin{aligned} u_Z &= \alpha^2 - u_S - u_T \\ &= 12^2 - 2 - 15 \mod 17 \\ &= 127 \mod 17 \\ &= 8 \mod 17 \end{aligned}$$

So the value of $u_Z = 8$. Find v_Z by using following formula (2.13):

$$\begin{aligned} v_Z &= -(\alpha \cdot x_Z + v_0) \\ &= -((12)(8) + 3) \mod 17 \\ v_Z &= 3 \mod 17 \end{aligned}$$

So, the point addition of $S^3 = 3S = (8, 3)$.

Definition 2.1.15. Suppose E is an elliptic curve defined over a finite field $\mathbb{F}_p$. Consider two points, A and B on elliptic curve $E_{\mathbb{F}_p}$ such that k is an unknown integer. Calculate k from $A = k.B$ is hard as it is equivalent to the calculation of Elliptic Curve Discrete Logarithm Problem (ECDLP) [46].

2.2 Cryptograhic Background

This section presents some fundamental definitions of cryptography. It also discusses basic cryptographic techniques that will be utilized in the thesis.

2.2.1 Cryptology

Cryptology is the science of secure transmission [47]. Cryptology originated from Greek words "Kryptos" meaning "hidden" and "Logos" meaning "word". Cryptology is the joint study of cryptography and cryptanalysis. In cryptology, we study about the cipher and decipher techniques.

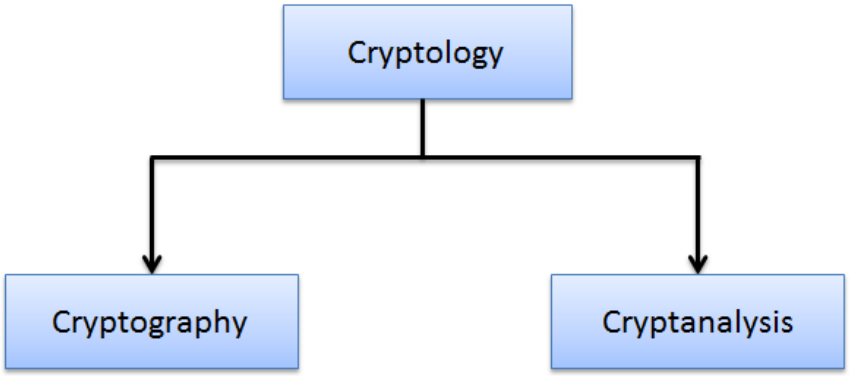

FIGURE 2.2: Cryptology

2.2.2 Cryptography

Cryptography is derived from Greek word, "Kryptos" means "Hidden or secret" and "Graphein" means "writing". Cryptography is a confidential composing of essential information. To prevent from attacking while transmission, messages are converted to unreadable text. In cryptography, the sender sends the ciphered message while the receiver deciphers the ciphered message. The process is known as encryption and decryption respectively. In encryption, the sender changes a message into an unrecognizable text and sends the encrypted text to the receiver. Later, the receiver recovers the original message using the process known as decryption. Cryptography offers the properties like authenticity, integrity, confidentiality and non-repudiation.

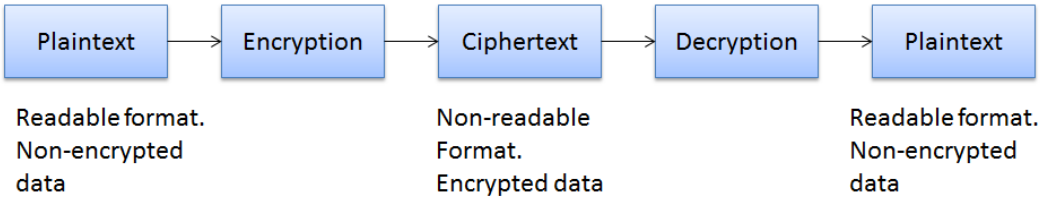

FIGURE 2.3: Cryptography

Cryptosystem has five main components:

1. Message space M.

2. Ciphertext space C.

3. Key space K.

4. Encryption algorithm space E.

5. Decryption algorithm space D.

Cryptography is of two types based on key distribution:

1. Symmetric Key Cryptography.

2. Asymmetric Key Cryptography.

2.2.3 Symmetric key cryptography

Symmetric key cryptography is also called secret key cryptography. In symmetric key cryptography, both parties use only one key for both process known as encryption and decryption. Both parties interchange the key before the transmission of information. Given a message (plaintext) and the key, encryption process produces unintelligible data, which is about the same length as the plaintext was [48]. Decryption process is the reverse of encryption process.

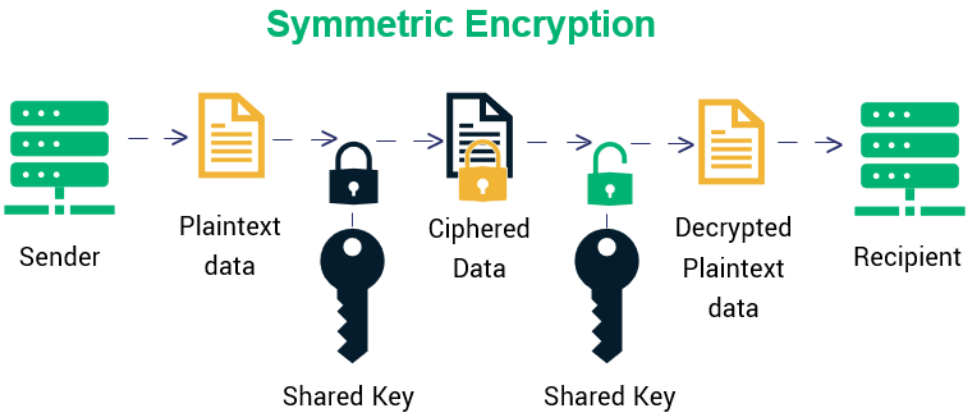

FIGURE 2.4: Symmetric key Cryptography

The technique of symmetric encryption is two way, a message block and specified key will always give same ciphertext, and same key will always give same original

message after applying on the ciphertext block.

Symmetric key cryptography is used to secure information between two parties using particular shared private key. it is used for protecting sensitive information [48].

Two notable methods of symmetric key cryptography are:

1. Data Encryption Standard (DES) [7].
2. Advanced Encryption Standard (AES) [8].

Data Encryption Standard (DES)

In 1973, first symmetric encryption technique DES is proposed. This is the first scheme which is published by NIST (National Institute of Standards and Technology) as a most effective scheme in 1976 [49]. The schematic diagram of DES is shown below:

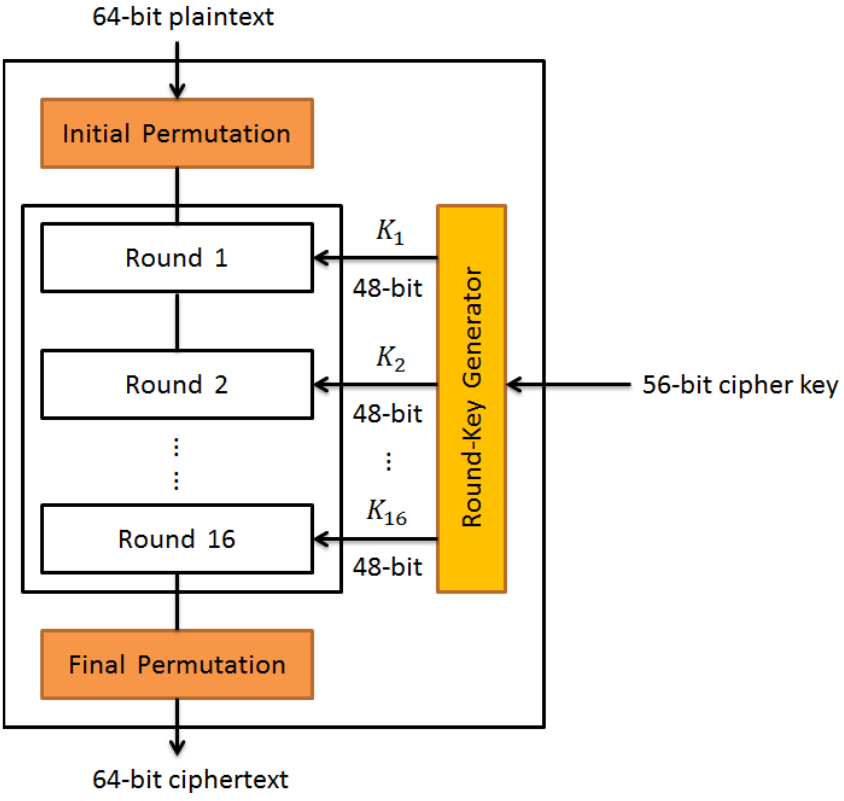

FIGURE 2.5: Data Encryption Standard Scheme

In this cryptographic scheme, same key is used for both encryption and decryption. The ciphertext can only be restore to the plaintext by those who have any information about the key. In Data Encryption Standard (DES), block encryption algorithm is used. For decryption, just reverse the order of encryption rounds and will get the original plaintext.

Simplified-Data Encryption Standard (S-DES)

Simplified DES or S-DES is an encryption algorithm used for educational purposed rather than being a secure encryption algorithm [50]. The scheme shares similar properties and structure with DES, but employs smaller settings.
In S-DES, an 8-bit block of plaintext is taken along with a 10-bit key as input. It then produces an 8-bit block of ciphertext as output. To decrypt, S-DES takes an 8-bit block of ciphertext and uses the same 10-bit key that was used before. it then reverts the ciphertext back to the original 8-bit block of plaintext.

Advanced Encryption Standard(AES)

Advanced Encryption Standard (AES) called as Rijndael algorithm a symmetric block cipher. AES is an iterative scheme which is also published by National Institute of Standards and Technology (NIST) in 2000 as an efficient scheme after the evidence that many theoretical attacks can break the DES cipher [8].
In this scheme, the DES 64 bit block size is extended to 128 bit block size and the DES 128 bits key size is broaden to the 256 bits key size. AES algorithm is compatible for different key length of size 128, 192 and 256 bit.

Disadvantages of symmetric key cryptography

Symmetric key cryptography has several disadvantages, here are some of the disadvantages of symmetric key cryptography:

1. The security depends on the shared secret key.

2. When two parties want to communicate securely for the first time, they need to find a secure way to exchange the initial secret key. This can be challenging, especially in untrusted or insecure environments.

3. Since both the sender and recipient use the same key, messages cannot be definitely verified to have come form a particular person.

2.2.4 Asymmetric Key Cryptography

Asymmetric Key Cryptography was introduced by the Diffie and Hellman in 1976 to solve the key exchange problem. In asymmetric key cryptography, every user has two keys; a public key and a private key. The public key is known to everybody and private key is confidential. If encryption is done by one key then the decryption process will be done by the other key. Therefore, encryption is done by public key and decryption by private key then no one other than who knows private key can decipher the ciphered message.

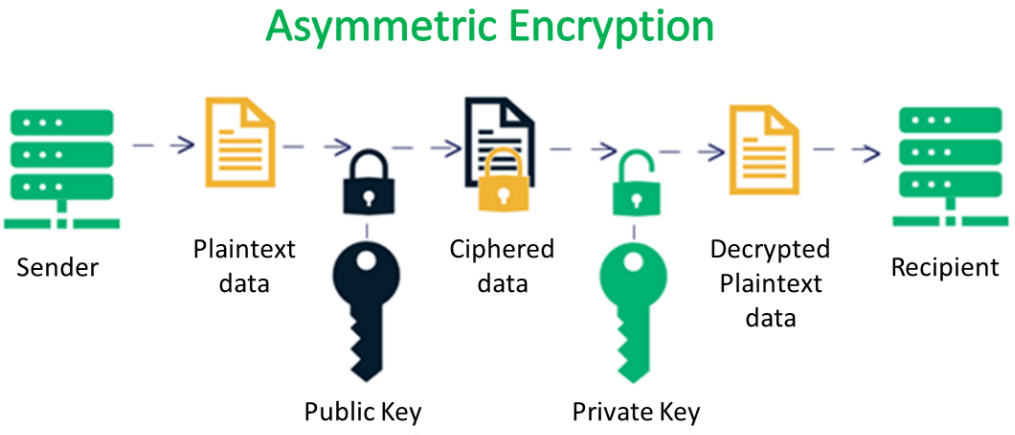

FIGURE 2.6: Asymmetric Key Cryptography

Some of the asymmetric cryptography techniques are as follows:

1. Digital Signature Algorithm (DSA) [51].

2. Rivest-Shamir-Adleman (RSA)[10].

3. ElGamal Cryptosystem [11].

4. Elliptic Curve Cryptosystem(ECC) [12].

2.2.5 Elliptic Curve Cryptosystem

Elliptic curve cryptosystem is a public key cryptosystem. This scheme was proposed in 1985 by the Neil Koblitz and Victor Miller. ECC can be used for digital signatures, data encryption and key exchange. When compared to RSA, it has been found that ECC can employ a key size that is significantly less. In ECC, a smaller key size of 160 bits is required compared to other cryptosystems. Hence, employing ECC with a smaller key size offers a computational benefit over using a similarly secure RSA. The outcomes demonstrate that ECC is effective in terms of the size of data files and encrypted files [52]. Because of less communication and computation cost ECC is widely used for security purposes.

Symmetric Scheme	**ECC based Scheme**	**RSA/DSA**
56	112	512
80	160	1024
112	224	2048
128	256	3072
192	384	7680
256	512	15360

TABLE 2.2: Comparing Key sizes [53]

The security of elliptic curve cryptosystem relies on the difficulty of solving the elliptic curve discrete logarithm problem (ECDLP), which is known to be a computationally hard problem.

Elliptic Curve Encryption Decryption

Elliptic Curve Cryptography is an asymmetric cryptographic scheme. Elliptic curve cryptography involves the use of elliptic curves for various cryptographic

operations. Every user has its own keys for secure transformation. The encryption and decryption using ECC is described in this section [54].

Global Parameters

The global parameters of this scheme are as follows:

1. The elliptic curve's generator point G a very large of order n. That is, $nG = \mathcal{O}$
2. The constant parameters a and b of curve.
3. The prime integer p.

Key Generation

Suppose that Ayesha wants to send a message M to Badar. First, both Ayesha and Badar calculate their keys using the following manner:

1. Ayesha selects arbitrarily a number as her private key $K_{pr} = k_A$ such that

$$k_A \in \{1, \ldots, n-1\}.$$

   Then, calculates her public key K_A by using the generator point of elliptic curve as:

$$K_A = k_A \cdot G.$$

2. Badar selects arbitrarily a number as his private key $K_{pr} = k_B$ such that

$$k_B \in \{1, \ldots, n-1\}.$$

   Then, calculates his public key K_B by using the generator point of elliptic curve as:

$$K_B = k_B \cdot G.$$

Encryption

Ayesha wants to send a message M to Badar that is guess as a point on the elliptic curve $E_p(a, b)$

1. Ayesha choose arbitrarily a positive integer α.
2. Then, create ciphertext by using public key of Badar.

$$C_M = \{\alpha G, B_M + \alpha K_B\} \tag{2.15}$$

Decryption

After obtaining the ciphertext C_M from Ayesha. Badar convert the ciphertext into its original plaintext by using the formula:

$$\begin{aligned} B_M &= B_M + dK_B - k_B(dG) \\ &= B_M + d(k_B G) - k_B(dG) \\ &= B_M \end{aligned} \tag{2.16}$$

Example 2.2.1. Suppose the elliptic curve $y^2 = x^3 + 7 \mod 17$. Let the generator point be $G = (15, 13)$. The order of the elliptic curve point is 18, that is, $18G = \mathcal{O}$ where $\mathcal{O}$ is the point at infinity.

Let Ayesha wants to send a message B_M to Badar by using Elliptic Curve Cryptography.

Suppose Ayesha selects her private key $k_A = 3$ and calculate her public key as: $3G = (8, 3)$. Badar selects his private key $k_B = 5$ and calculate his public key as: $5G = (6, 6)$.

Ayesha selects a random integer $d = 2$ to convert the original message $B_M = (11, 3)$ into ciphertext.

$$\begin{aligned} C_M &= \{dG, B_M + dK_B\} \\ &= [2(15, 13), (11, 3) + 2(6, 6)] \end{aligned}$$

$$
\begin{aligned}
&= [(2,10),(11,3)+(3,7)] \\
C_M &= [(2,10),(16,8)] \qquad (2.17)
\end{aligned}
$$

Ayesha deliver the ciphertext to Badar. After obtaining the ciphertext, Badar decrypt the ciphertext and obtain the original message B_M.

$$
\begin{aligned}
B_M &= B_M + dK_B - k_B(dG) \\
&= B_M + d(k_B \cdot G) - k_B(dG) \\
&= [(11,3) + 2(5(15,13)) - 5(2(15,13))] \\
&= [(11,3) + (3,7) - (3,7)] \\
B_M &= [(11,3)] \qquad (2.18)
\end{aligned}
$$

2.3 Digital Signature

Digital signature resemble digital fingerprints. Digital signature is used to ensure that the message is sent by a verified sender. They guarantee the confidentiality of the message, assuring that the message will not altered during transmission. Digital signature is a mathematical code that is attached to the message that needs to be signed. When an individual digitally signs a document or a message, the algorithm generates a unique digital signature that corresponds to that content.

In 1976, Diffie and Hellman [55] gave the idea of digital signature. However, they hypothesize that such scheme exists on the basis of one way trapdoor function. Following are the components of digital signature scheme:

1. A security parameter k, chooses by the user which determines the quantities (signatures length, signable messages length, computational time of the signing algorithm etc).
2. A message space $\mathcal{M}$, set of messages that is needed to be signed.
3. A signature bound B, is an integer that limits the signature created by the algorithm.

4. A key generation algorithm G, every user can create both public and private keys.

5. A signature algorithm σ, creates the signature for a message M.

6. A verification algorithm V, which verifies the authentic signature for message $M \in \mathcal{M}$.

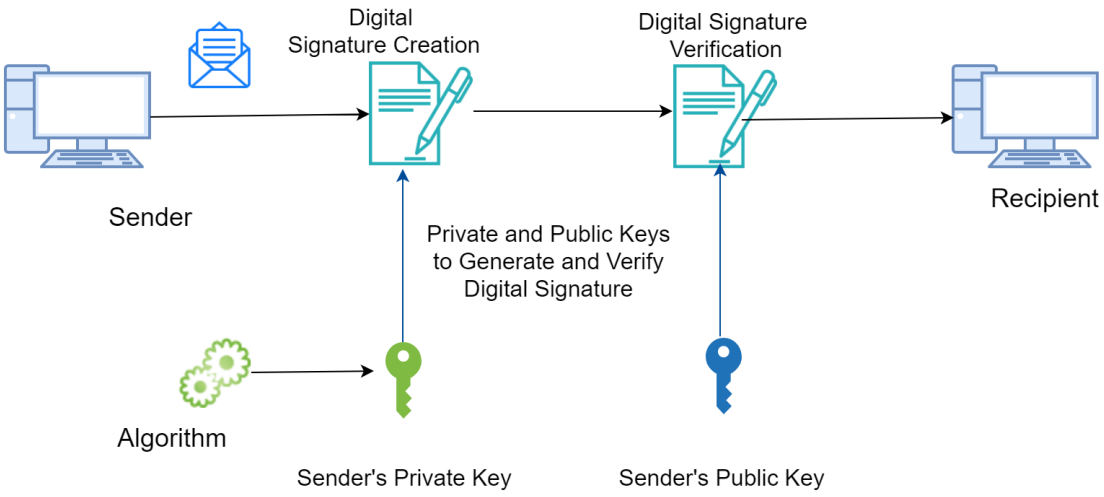

FIGURE 2.7: Digital Signatures

The security parameters that are proposed by the digital signatures:

1. **Authenticity:** Digital signatures used to verify the identification of the message source. An authentic signature verifies that the message is sent by the authorized sender.

2. **Integrity:** Integrity means that the message is not altered during communication. In digital signatures, if an attacker alters the message during transmission, even a small change invalidate the signature.

3. **Non Repudiation:** In non repudiation, a signer cannot deny his/her signature after a while.

4. **Unforgeability:** In unforgeability, only a signer can create a valid signature.

Suppose Ayesha wants to send a digitally signed document to Badar.

Ayesha

1. Choose a document Δ that is needed to be signed.

2. Create hash value H of the document Δ, that is,

$$H = h(\Delta)$$

3. To sign the document, encrypt the hash value of the document with her private key K_{pr}.

$$C = E_{K_{pr}}(H).$$

4. Transmit the signature s and hash value of document H alongside with the document to Badar.

Badar

1. Badar receive the Document Δ along with digital signature s and hash value of document H.

2. He then uses Ayesha's public key to decrypt the digital signature s and finds the hash value of the document H.

3. If the received hash values of document H match with the hash value of step 2 then the message is authentic.

4. Otherwise, someone changed it during transmission.

2.4 Signcryption

In cryptography, signcryption schemes offer both the confidentiality and authentication at the same time. That is, in a single-step signature and encryption done on the messages or documents that need to be shared. The first signcryption scheme was proposed by Zheng [16] in 1997. Signcryption offers same characteristics as

the digital signature and encryption.

In signcryption, the sender uses his/her private key for signature, and uses public key of the receiver to create the secret key that is used to cipher the message. While, in unsigncryption receiver uses the public key of sender to verify the digital signatures. Receiver uses his/her private key for creating the secret key that is used for unsigncryption. The cost of the signcryption scheme is less than the cost of traditional signature-then-encryption schemes.

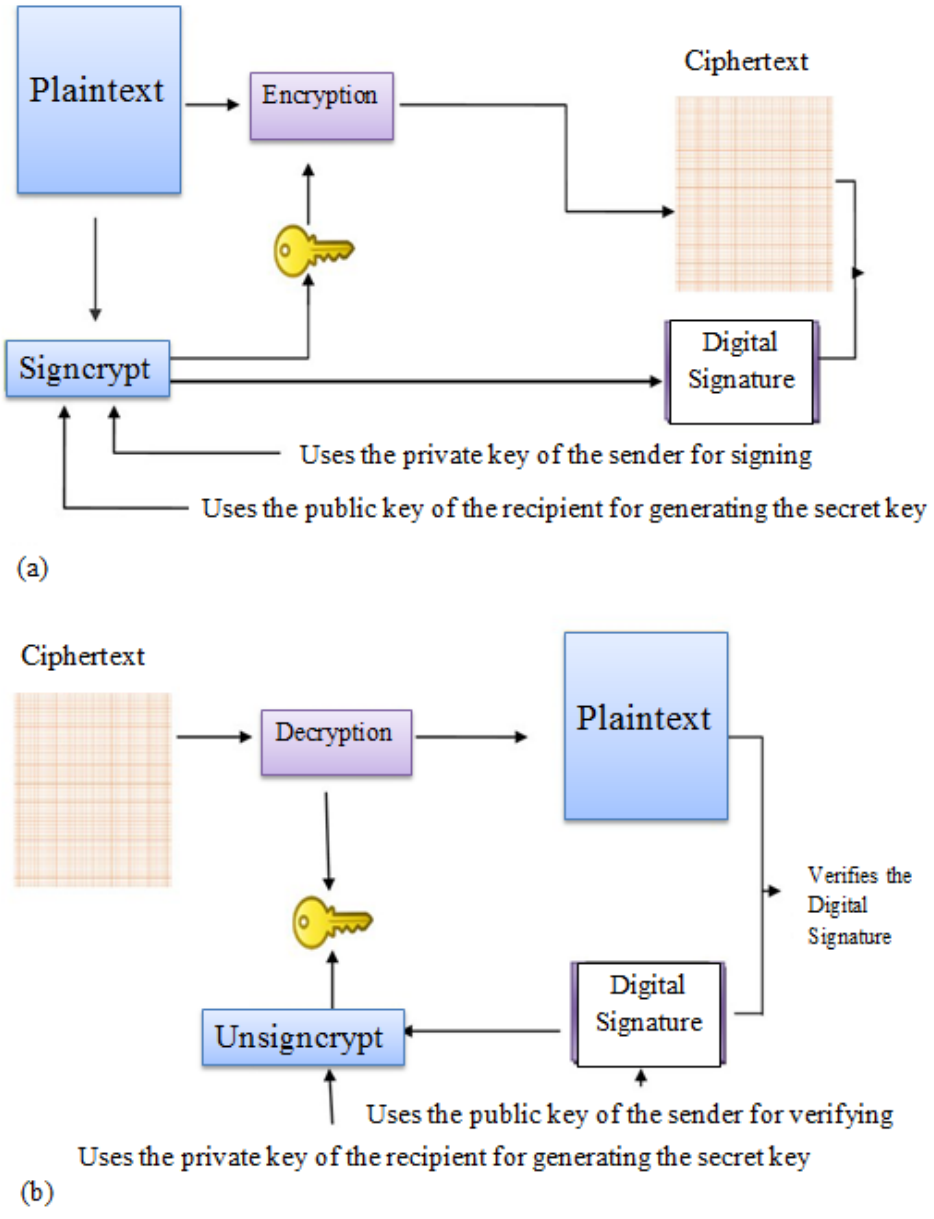

FIGURE 2.8: Signcryption Model (a) Signcryption (b) Unsigncryption

Signcryption schemes provides the following properties [18]:

1. **Confidentiality:** Without knowing any secret key of sender and recipient, it is computationally impossible for an attacker to obtain the contents of the message.

2. **Unforgeability:** It is computationally impossible for an attacker to create a valid signature or signcrypted text.

3. **Non-Repudiation:** In this property a sender can not negate his/her signature. The sender of the signcrypted text is verified by the third party/judge after obtaining the parameters by the receiver.

4. **Integrity:** The receiver can validate the original message that message is transmit by the authentic sender.

5. **Public Verifiability:** In this property, a third party or judge can verify that for a message, signcrypted message is the authentic signcryption without knowing any private keys of both the sender and the receiver.

6. **Forward Secrecy:** After misplacing the long lasting private keys, nobody can retrieve the original message by unsigncryption of the preceding signcrypted text. Furthermore, the preceding signatures also have become duplicitous.

A signcryption typically comprises the three algorithm:

- Key generation algorithm.
- Signcryption algorithm.
- Unsigncryption algorithm.

2.4.1 Zheng's Signcryption scheme

In 1997, Zheng [11] introduces signcryption scheme based on ElGamal digital signature scheme.

Global Parameters

Let user A and user B agreed on the following global parameters:

Variables	Description
p	A large prime number.
q	A prime factor of $p-1$.
g	An arbitrary integer from $\{1,\ldots,p-1\}$ with order q modulo p.
h	One way hash function.
KH	Keyed one way hash function.
E_K	Encryption algorithm with the secret key.
D_K	Decryption algorithm with the secret key.

TABLE 2.3: Global Parameters

Key Generation

Both users A and B can generate their public and private keys in the following manner:

1. User A selects arbitrarily a number α_a from $\{1,...,q-1\}$ as its private key u_A. Then, calculates its public key β_a using:

$$\beta_a = g^{\alpha_a} \mod p.$$

2. User B selects arbitrarily a number from $\{1,...,q-1\}$ as its private key α_b. Then, calculates its public key β_b using:

$$\beta_b = g^{\alpha_b} \mod p.$$

Signcryption

To obtain the signcrypted text, User A performs the following steps:

1. User A chooses randomly $\gamma \in \{1, \ldots, q-1\}$.

2. Calculate keys with using hash functions:

$$K = h(\beta_b{}^{\gamma}) \mod p.$$

3. Split K into two parts, K_1 and K_2, ensuring each part has the suitable length.

4. Then, use K_1 to encrypt the message m:

$$c = E_{K_1}(m).$$

5. Then, by using keyed hash function calculate x as:

$$x = KH_{K_2}(m).$$

6. If shortened digital signature standard 1 (SDSS1) is used then create signature by using:

$$s = \frac{\gamma}{(x + \alpha_a)} \mod q.$$

7. Otherwise, if shortened digital signature standard 2 (SDSS2) is used then create signature by using:

$$s = \frac{\gamma}{(1 + \alpha_a x)} \mod q.$$

8. Send (c, x, s) to the User B.

Unsigncryption

1. If shortened digital signature standard 1 (SDSS1) is used then find keys using the following formula:

$$K = h((\beta_a g^x)^{s\beta_b}) \mod p.$$

2. Otherwise, if shortened digital signature standard 2 (SDSS2) is used then find keys using:

$$K = h((q{\beta_a}^x)^{s\alpha_a}) \mod p.$$

3. Then decrypt the message M using:

$$m = D_{K_1}(c).$$

4. The message is real if $KH_{K_2}(m) = x$ then accept it.

2.5 Hash Function

In cryptography, hash function is an important mechanism. It has been well known for a long time in the discipline of computer science. In hash function, a bit string of random size is mapped into a fixed-size value. This value is known as hash value [4]. The hash value also known as hash codes, digest or simply hash. A well-designed hash function ensures that a small modification in the input leads to a substantially different hash function.

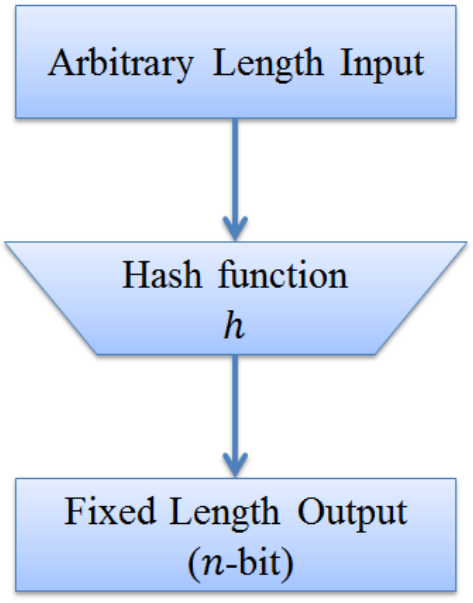

FIGURE 2.9: Hash Value

Hash function gives the property of integrity and uniqueness. To generate a hash value, a one-way trapdoor function is used, which is simple to compute, while the reverse process of recomputing the input from the generated hash value is extremely challenging. Making any change to a message will result in a different hash value. Even if the single bit of message is changed, the hash value will be completely changed. Hash function gives the properties of authentication and efficiency. Hashes are helpful in sending files or any messages and gives integrity. While sending files or any message, hash value delivered along with the message and the recipient can verify the integrity of the message. Hash function gives the property of integrity in digital signatures.

There are many hash functions that are of different fixed-length. Examples are: SHA-1 [13], MD5 [13],CRC-8 [56], CRC-32 [57], SHA-2 [4] and SHA-3 [41].

Cyclic Redundancy Check

Cyclic redundancy check, abbreviated as CRC, represents a technique for detecting errors during decryption [57]. CRC can directly examine whether an error is occurred. In such cases, the recipient can inform the sender to resend the data [56]. A CRC is a widely employed error-detection code in digital networks and storage devices, serving to identify unintended alterations in digital data. CRC are popular due to their simplicty in binary hardware implementation, easy to analyze mathematically, and particularly good in identifying typical errors arising from transmission channel noise.

The most commonly used polynomial lengths are 9-bits (CRC-8), 17 bits (CRC-16), 33-bits (CRC-32) and 65-bits (CRC-64).

Properties of hash function

The following characteristics are found in a good hash function:

1. Comparatively easy to compute the hash value of any message.

2. It must be difficult to find the inverse value of a hash function, i.e.

$$r = h(m)$$

   In the above equation, finding r is easy, but it is infeasible to find m when r is known.

3. The hash value must be unique. For different values different output should be given.

4. It should offers the property of integrity. No one can forge the value of the message. A slight change in the message should change the whole output value.

5. It should generate the output of fixed-length.

2.6 Cryptanalysis

The practice of analysing a cryptographic scheme is known as cryptanalysis and it entails doing so in order to access and improve the system's security as well as to learn from the flaws that are found and prevent them from occurring again in the future. Those who carry out this function is known as cryptanalyst. Attacker has the main goal to find out key or the plaintext. In this section, different cryptanalysis attacks will be over-viewed.

2.6.1 Brute force attack

In brute force attack, "Trial and error" technique is used for predicting the keys [58]. By attacking a system or data, the attacker wants to find out the secret information of a data or a system. He/She uses every possible combination of letters, symbols and numbers to guess the correct key. But it will take a very long time to guess the key of a system or data. The time depends on the length of the key and its complicatedness.

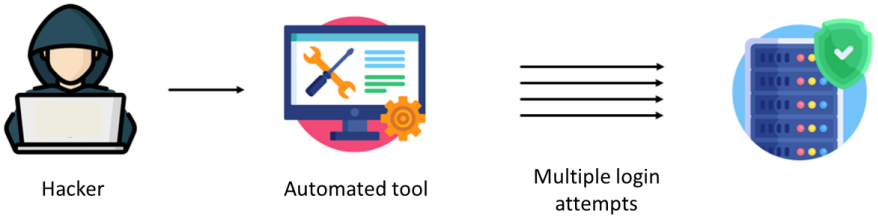

FIGURE 2.10: Brute Force Attack

When the attacker relies on the exact dictionary words for their attack, it is referred to as a "Dictionary attack". On the other hand, if the attacker makes slight modifications to dictionary words while performing the attack, it is known as a "Hybrid Brute-force attack".

2.6.2 Ciphertext Only Attack

In ciphertext only attack (COA), the attacker has the only knowledge of the ciphertext, but he/she does not know about the plaintext or the secret key. The main objective of an attacker is to get the plaintext or the key. In some cases attacker has some knowledge about the plaintext, such that, the plaintext language in which plaintext is written or the language redundancy, but in most cases he/she has no access to plaintext.

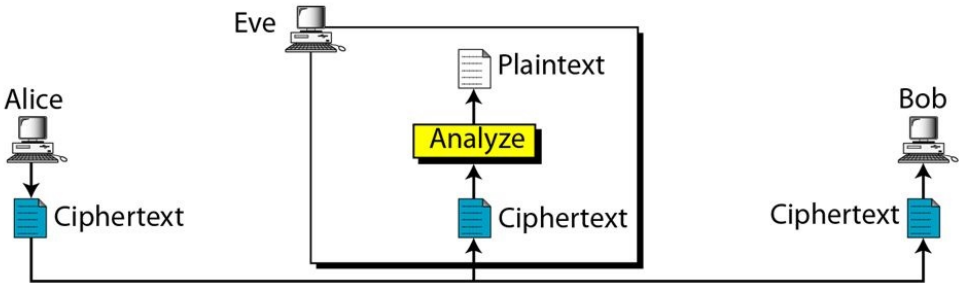

FIGURE 2.11: Ciphertext-Only-Attack

If the attacker figures out the key, he/she can retrieve all the messages using that key. The success of these attacks depends upon both the robustness of the encryption algorithm and the length of the encryption key.

2.6.3 Known Plaintext Attack

In known plaintext attack (KPA), attacker only has access to some of the section of the ciphertext. His objective is to discover the plaintext of the remaining ciphertext or to discover the key. If the attacker gets the key then he/she can decrypt the whole ciphertext. Otherwise, he/she measures the relationship between the ciphertext and the known plaintext. The attacker then uses that information and tries to decrypt the whole ciphertext.

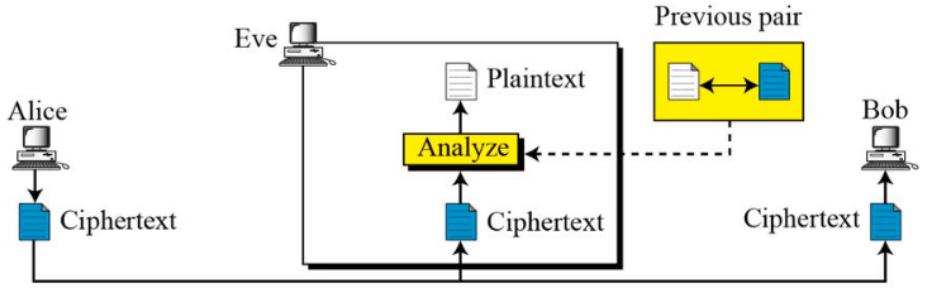

FIGURE 2.12: Known Plaintext Attack

2.6.4 Chosen Ciphertext Attack

Chosen Ciphertext Attack (CCA) is the most severe among among all other types of attacks. In CCA, the attacker has access to the decrypting algorithm [59]. He randomly chooses some ciphertext and then obtains their decrpytion. From this, attacker attempt to obtain key or some details about the system.

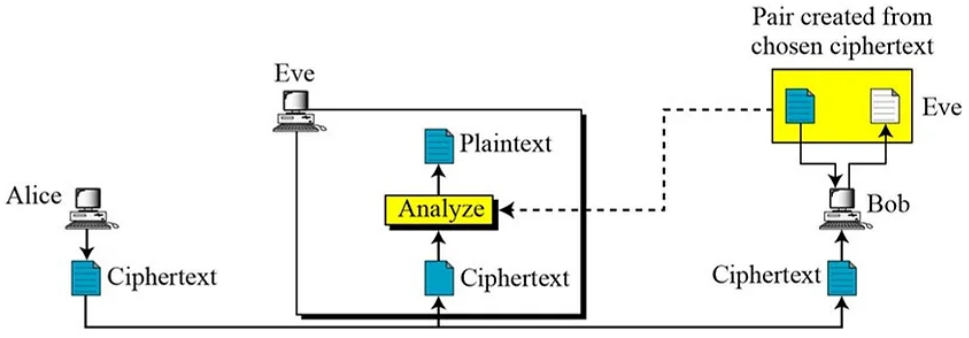

FIGURE 2.13: Chosen Ciphertext Attack

The chosen ciphertext attack is only helpful in public key cryptosystem, because the attacker only has the approach to the public enciphering algorthim [59].

2.6.5 Chosen Plaintext Attack

The Chosen plaintext attack is used rarely but it is still harmfull [60]. In chosen plaintext attack, the attacker chooses an arbitrary plaintext to be encrypted and then get ciphertext. The main aim is to obtain the secret key or to make an encryption algorithm through this information so that he can decrypt the ciphertext.

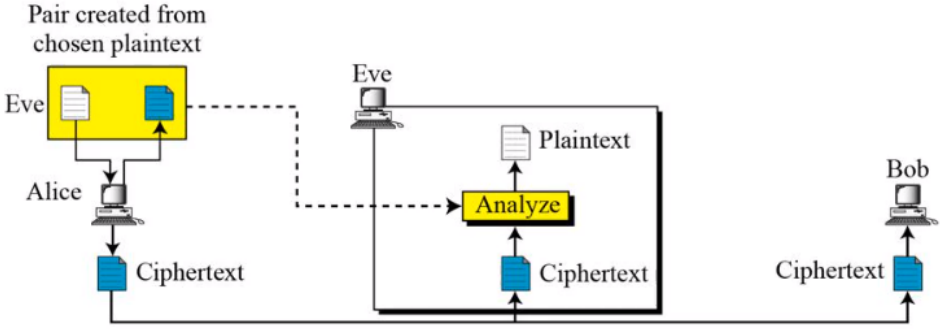

FIGURE 2.14: Chosen Plaintext Attack

Chosen plaintext attack is applicable to the public key cryptosystem, because the attacker can utilize public key for encrypting the chosen plaintext.

2.6.6 Man In The Middle Attack

Man In The Middle Attack is a general term where an attacker or third party takes the control over the secret broadcasting system between two persons/parties. In this attack, the attacker can eavesdrop on or to masquerade as one of the parties that are involved in the exchange, presenting it as if a normal conversation, or information transformation is going.

The attacker first chooses two fake keys to carry out this attack and then initiates communication with the first participant using one of those keys [61]. the attacker establishes a successful combination with the first participant. Similarly, another

successful connection with the second participant. Now, the attacker sends a message of his/her own choice to both participants. Both participants believe that they are communicating with each other.

Man In The Middle Attack short form is MIIM, Miim, MITMA, MitM.

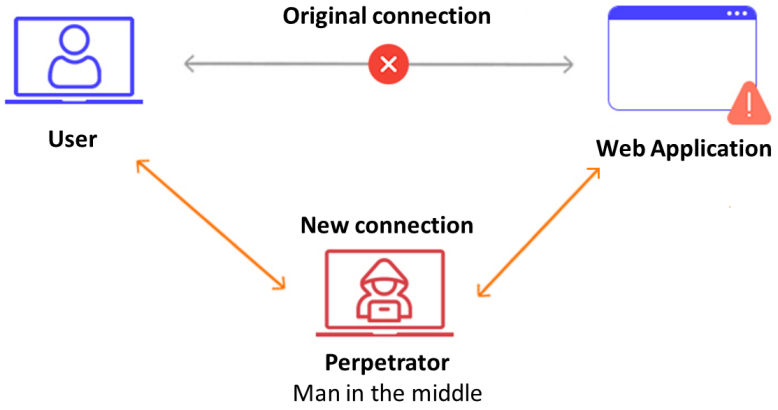

FIGURE 2.15: Man In The Middle Attack

Chapter 3

Blind Signcryption

Because of high demand of privacy in different applications, digital signatures play a vital role for offering authentication and confidentiality. Recall that, in blind signatures, information in the message is blinded before signing. Blind signatures are generally used where the privacy is required between the different parties, the signer and the requester. Examples of this cryptographic scheme is E-Voting and Digital Cash System.

Blind Signature schemes are generally executing by utilizing the public key cryptosystem such as RSA [62], Elliptic Curve Cryptography [63] and ElGamal Cryptography [64]. To implement the signature, message is blinded first by the requester then combined with some random 'Blinding Factors', and send it to the signer. The signer without seeing any message content sign the message with the corresponding scheme and then send it to the requester. The signatures then verified by the verifier.

Blind signature scheme typically consists of two participants:

1. **Requester:** The requester is the one who asks for signatures, requesting the signer to sign the message. The requester first blinds the message and then sends it to the signer.

2. **Signer:** Signer is the one who sign the message without seeing any message content.

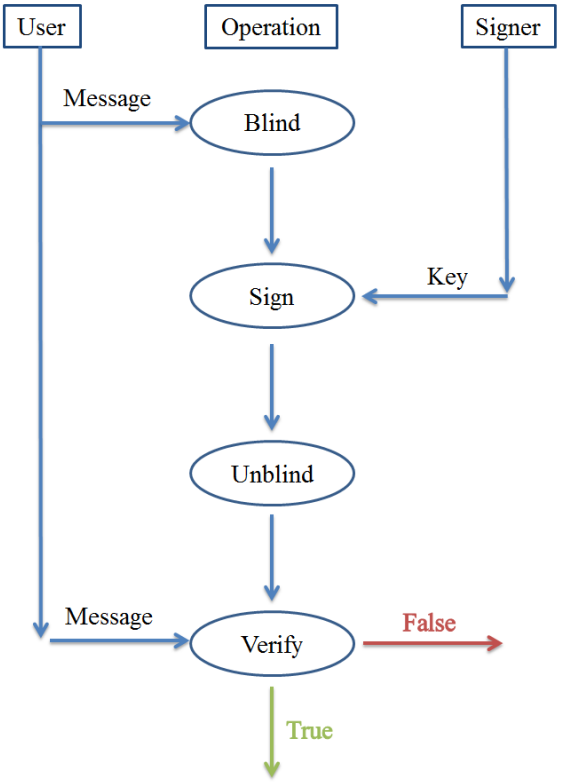

FIGURE 3.1: Blind Signature

3.1 Chaum's Blind Signature Scheme

In this section, we will discuss the blind signature scheme of Chaum [19]. In the scheme, the third party cannot trace the sender of the message. The characteristics that construct the blind signature scheme are as follows:

1. **The signing function:** Only signer knows the signing function s, and its publicly known inverse s^{-1} such that $s^{-1}(s(t)) = t$ and through s^{-1} one cannot find s.

2. **Commuting Function:** Only the requester knows the commuting function $\bar{s}$ and its inverse $\bar{s}^{-1}$ such that $\bar{s}^{-1}(s(\bar{s}(t))) = s(t)$. Both $\bar{s}(t)$ and s give no information about t.

3. **Redundancy:** A redundancy checking mechanism u, that ensure there is enough redundancy to make it difficult to find legitimate signatures.

These function are used in scheme as follows:

1. The requester selects arbitrarily t such that $u(t)$, forms $\bar{s}(t)$ and then transmits $\bar{s}(t)$ to the signer.

2. Signer uses s to sign $\bar{s}(t)$ and sends back the signed content $s(\bar{s}(t))$ to the requester.

3. The requester then unblinds the signed content by applying $\bar{s}^{-1}$ such that $\bar{s}^{-1}(s(\bar{s}(t))) = s(t)$.

4. To verify the signature, any body can use the signer's public key s^{-1} and verify that the unblinded factor $s(t)$ is signed by the signer, and examine: $u(s^{-1}(s(t)))$.

The security properties are as follows:

1. Unlinkability:

This scheme provides unlinkability, in which the signer cannot link the blind message he signed to the previous messages.

2. Blindness:

The signer does not know the message contents. The requester sends the blinded message to the signer, who signs the message blindly.

For further details about the scheme we refer to [19].

3.2 Elliptic Curve Cryptography Based Blind Signcryption Scheme for E-Voting System.

In this section, we will discuss the scheme of Waheed et al. [38] "A Novel Blind Signcryption Scheme for E-Voting System Based on Elliptic Curve". In this scheme, the approved voter can cast vote via electronic devices in the presence of

internet. Polling server will count all the votes at the end and check the anonymity of voter. The proposed scheme consists of three participants: signer/polling station, voter, and polling server. The suggested procedure is further divided into four phases:

1. Key generation.
2. Establish communication between two parties.
3. Blind signcryption.
4. Unsigncryption.

This scheme uses elliptic curve cryptography (ECC). In this scheme, elliptic curve is used because it is inexpensive and use less bit sized key. The security of this scheme depends on the hardness of ECDLP. This scheme gives the property of authenticity, forward secrecy, unlinkability and non-repudiatuion, unforgeability, confidentiality, integrity.

Scheme Participants

1. **Signer/Polling Station:**
   The voter/requester sends the blinded message to the signer at the polling station to sign the message. The signer will sign the message blindly without seeing the original message. After blindly signing the message, polling station sends back the signed blinded message to the voter/requester.
2. **Voter/Requester:**
   Voter is the one who wants to cast a vote. The voter corresponds with the polling station/signer and requests them to sign his blinded message. After receiving the signed blinded message from the signer/polling station, voter sends a signcrypted vote/data to the polling server.
3. **Verifier/Polling Server:**
   The polling server verifies the validity of a voter and acts as an authentic

voter's data verifier, checking the voter's validity after unsigncrypting the received signcrypted message. Polling server accepts the vote if the vote is valid or else repudiate the vote.

3.2.1 Proposed Scheme

In this section, scheme proposed by Waheed et al. [38] is presented. After mutually agreeing on the global parameters, the scheme has the setup phase, key generation phase, blind signcryption, unsigncryption as expressed below.

Global Parameters

The global parameters that are used in this scheme are given below:

Symbol parameters	**Description**
p	p is a large prime number $\geq 2^{224}$
n	Order of elliptic curve, which is $> p$
G	The base point on ECC of order $n \geq 2^{224}$
$\mathbb{F}_p$	The Finite Field of order p
$E(\mathbb{F}_p)$	Elliptic points on ECC curve $\mathbb{F}_p$
$E_K(\cdot)$	Symmetric encryption operation using secret key K.
$D_K(\cdot)$	Symmetric decryption operation using secret key K.
h	A hash function
m	Message.
c	Ciphered text.
$\|\|$	Concatenation symbol.

TABLE 3.1: Global Parameters

Setup phase

In this phase, the Elliptic Curve Cryptosystem is used to define the security parameters. With former $p \geq 2^{224}$ and choose $a, b \in \mathbb{F}_p$ to form elliptic curve E over

$\mathbb{F}_p$. That is,

$$E(\mathbb{F}_p) : y^2 = x^3 + ax + b \mod p$$

where a and b are chosen such that

$$4a^3 + 27b^2 \neq 0$$

The base point G in $E(\mathbb{F}_p)$ is of order $n \geq 2^{224}$; Hash function is denoted by h, the message is given by m and $E_k(\cdot)/D_k(\cdot)$ are functions for symmetric key encryption and decryption respectively with a secret key k. The ciphered message is denoted by c.

3.2.2 Key Generation Phase

Every voter selects his/her private key Pr_k and calculates its public key Pub_k. Afterwards, gets certificate from relevant certificate authority. Other participants, signer/polling station and verifier/polling server, calculate the public and private key using the same method. The methodology is described as follows:

1. Signer/polling station chooses a number $d_{sign} \in \{1, \ldots, n-1\}$ arbitrarily as a private key. Then calculates his/her public key P_{sign} using:

$$P_{sign} = d_{sign} \cdot G$$

2. Voter/requester chooses a number $d_{req} \in \{1, \ldots, n-1\}$ arbitrarily as a private key. Then calculates his/her public key P_{req} using:

$$P_{req} = d_{req} \cdot G$$

3. Verifier/Polling server chooses a number $d_{ver} \in \{1, \ldots, n-1\}$ arbitrarily as a private key. Then calculates his/her public key P_{ver} using:

$$P_{ver} = d_{ver} \cdot G$$

3.2.3 Signcryption Algorithm

Any voter/requester desires to send a message without revealing its identity to the polling server/verifier in a trustworthy and secret manner. The procedure described below is to produce a blind signcrypted message.

A. Voter/Requester

The following steps have to be calculated at the voter/requester's end:

1. Choose an integer arbitrarily as: $\beta \in \{1, \ldots, n-1\}$.
2. Calculate elliptic curve point:

$$A = \beta \cdot G \tag{3.1}$$

3. Calculate hash value r using the hash function h as:

$$r = h(m||A) \tag{3.2}$$

4. Send r to signer/polling station.

B. Signer/Polling Station

After receiving r, the signer/polling station arbitrarily chooses an integer γ. The following steps have to perform at the signer/polling station's end.

1. Choose an integer arbitrarily $\gamma \in \{1, \ldots, n-1\}$.
2. Calculate elliptic curve point using:

$$X = \gamma \cdot G$$

3. Generate signature $\bar{s}$ by using signer private key as:

$$\bar{s} = (d_{sign} + r \cdot \gamma) \mod n \tag{3.3}$$

4. Signer forwards $(X, \bar{s})$ to voter/requester.

C. Requester/Voter

After receiving $(X, \bar{s})$ from the signer/polling station, the requester/voter has to perform the following steps:

1. Choose an integer arbitrarily $\varphi \in \{1, \ldots, n-1\}$.
2. Find the secret key k_e for encryption using:

$$k_e = h(\varphi \cdot P_{ver}) \tag{3.4}$$

3. Calculate ciphertext C using secret key k_e found in the previous step:

$$c = E_{k_e}(m||\bar{s})$$

4. Calculate signature using arbitrarily chosen integers φ and β and signature blind factor $\bar{s}$:

$$s = \left(\frac{\varphi}{r + \beta + \bar{s}}\right) \mod n \tag{3.5}$$

5. Forwards (c, r, s, A, X) to polling server/verifier.

3.2.4 Unsigncryption Algorithm

The polling server/verifier checks the authentication after receiving the message from blind signcrypted text. If it verifies the authentication of the message, then accept it and put it in the voter bank; if not, then reject the vote.

1. Verifier first calculates q by using his/her secret key d_{ver} as :

$$q = d_{ver} \cdot s \mod n \tag{3.6}$$

2. Calculate secret encryption key using signer's public key P_{sign} as:

$$k_e = h(q.(P_{sign} + r \cdot (X + G) + A)) \tag{3.7}$$

3. Decipher the message and the set blinded signature $\bar{s}$ using the above secret key k_e as:

$$m||\bar{s} = D_{k_e}(c)$$

4. Next calculates the hash value r' of the message m and A as:

$$r' = h(m||A)$$

5. If $r' = r$, then the message m is original and can be accepted safely otherwise vote will be rejected by the verifier.

3.2.5 Correctness

For the correctness of the scheme, we have to make sure that the encryption/decryption key k_e computed in equation (3.4) and (3.7) are same.

Theorem 3.2.1. If the following equation is authenticated by the sender and the receiver then the above scheme is accurate:

$$q \cdot (P_{sign} + r(X + G) + A) = \varphi \cdot P_{ver} \tag{3.8}$$

where all calculations are performed under modulo n.

Proof:

$$\begin{aligned}
L.H.S. &= q.(P_{sign} + r(X + G) + A) \\
&= q \cdot (P_{sign} + r \cdot \gamma \cdot G + rG + A) \\
&= d_{ver} \cdot s \cdot (P_{sign} + r \cdot \gamma \cdot G + r \cdot G + A) \\
&= \left(\frac{\varphi}{r + \beta + \bar{s}}\right)(d_{ver} \cdot (d_{sign} \cdot G + r \cdot \gamma \cdot G + r \cdot G + \beta \cdot G)) \\
&= \left(\frac{\varphi \cdot P_{ver}}{r + \beta + \bar{s}}\right)(d_{sign} + r \cdot \gamma + r + \beta) \\
&= \left(\frac{\varphi \cdot P_{ver}}{r + \beta + d_{sign} + r \cdot \gamma}\right)(d_{sign} + r \cdot \gamma + r + \beta)
\end{aligned}$$

$$= \varphi \cdot P_{ver}$$

This proves that the scheme described above is authentic.

3.3 Security Analysis

The scheme provides security against numerous attacks. The security of the scheme relies on Elliptic Curve Discrete Logarithm Problem (ECDLP). Below are the security characteristics of the scheme, which will be examined to authenticate the safety of the presented signcryption scheme.

1. Confidentiality

The scheme provides security against numerous attacks to ensure the confidentiality of message information. If an intruder wants to finds out the private key d_{sign} from the Step (1) of Section 3.2.2, he/she cannot easily solve the ECDLP, as it is computationally infeasible.

Case-I: If an attacker wants to calculate k_e using the equation from the Step (3) of Section 3.2.2 and from the equations (3.6) and (3.7) in the Section 3.2.4 as follows:

$$\begin{aligned} P_{ver} &= d_{ver} \cdot G \\ q &= d_{ver} \cdot s \\ k_e &= h(q \cdot (P_{sign} + r \cdot X + G) + A)) \end{aligned}$$

finding d_{ver} is hard for an attacker because of elliptic curve discrete logarithm problem (ECDLP).

Case-II: For an attacker to calculate k_e using the equations (3.1), (3.5) and (3.4) from the Section 3.2.3, as follows:

$$\begin{aligned} A &= \beta \cdot G \\ s &= \frac{\varphi}{r + \beta + \bar{s}} \end{aligned}$$

$$k_e = h(\varphi \cdot P_{ver})$$

the value of β is required which is hard to find for an attacker because of ECDLP.

2. Integrity

The integrity property assures that the message content does not change throughout the communication via the noisy medium. The voting server is authenticated using the proposed scheme. Voter evaluates $r = h(m||A)$ and impulsively signs the contents to create $\bar{s}$ then sends r to the polling station. Later, the voter creates his/her own signature s after the polling station/signer sends back the signature, and then the voter sends it for verification to the polling server.

If an intruder attacks and swaps the contents of the encrypted message from c to $\overline{c}$ while communicating, this indicates that the value of m also alters to $\overline{m}$ on the server's end too. Accordingly, the values of r, s will also be altered to $\overline{r}, \bar{s}$, which is impossible because of the random oracle properties of the hash function such that $h(m||A) \neq h(\overline{m}||A)$. For an authentic signature, the attacker requires β which is calculated from the equation (3.1) in the Section 3.2.3 as follows:

$$A = \beta \cdot G$$

and d_{sign} from the Step (1) of Section 3.2.2 as follows:

$$P_{sign} = d_{sign} \cdot G$$

respectively that are hard to find because of the ECDLP.

3. Unforgeability

In unforgeability, neither the attacker nor the receiver can fabricate values of signature (c, r, s) while communicating over a noisy medium. In the presented scheme, the attacker requires β, d_{sign} and m to create an authentic signature in the equation (3.5). For this, the attacker has to calculate β from

the equation (3.1) and d_{sign} from the Step (1) of Section 3.2.2 respectively:

$$A = \beta \cdot G$$
$$P_{sign} = d_{sign} \cdot G$$

that are difficult to find because of ECDLP.

4. Authentication

Authenticity makes sure that the received message or the sender is authorized or unauthorized. The proposed scheme gives validity at two stages: first, it gives signer's authenticity, and secondly, it also authenticates the registered vote collected to the polling server. After obtaining data, the polling server validates the signature by utilizing the polling station/signer's public key P_{sign} by utilizing the public key P_{sign} related with a signature key (private key d_{sign}). In the case that signatures are authenticated, it indicates that they were created by the authorized signer.

On the contrary, if they were changed during transmission or at some other location by anyone, the polling station will not accept them. Calculating d_{sign} from the Step (1) of Section 3.2.2 as following:

$$P_{sign} = d_{sign} \cdot G$$

is hard to compute because of ECDLP.

5. Public Verifiability

If a conflict arises, the judge (third verifer) can confirm the contents of the message by providing the parameters of the signature to the judge (third verifer) without seeing any message secret. The proposed scheme makes certain public verifiability. If a conflict arises, the polling server transmits $(m,\bar{s},X,A)$ to a third verifier to resolve the dispute and confirm the genuine signer. The third verifier authentication procedure is described below:

Judge: (Third Verifier)

Verify$(m,\bar{s},X, A, P_{sign})$

1. Authenticate the public key of signer P_{sign} with the certificate.
2. Calculate:
$$r = h(m||A)$$
3. Calculate:
$$\mathbf{y} = (\bar{s} \cdot G - r \cdot X)$$
4. If $\mathbf{y} = P_{sign}$ then the sign created by the authentic person with the public key P_{sign}

Theorem 3.3.1. The following equation validates the correctness of the public verifiability described earlier, if the following condition satisfied

$$\bar{s} \cdot G - r \cdot X = P_{sign}$$

Proof:

$$\begin{aligned} \bar{s} \cdot G - r \cdot X &= (d_{sign} + r \cdot B)G - r \cdot X \\ &= d_{sign} \cdot G + r \cdot B \cdot G - r \cdot X \\ &= d_{sign} \cdot G + r \cdot B \cdot G - r \cdot B \cdot G \\ &= d_{sign} \cdot G \\ &= P_{sign} \end{aligned}$$

This means that the authentication process is valid.

6. Non-Repudiation

If a conflict arises, the judge (third verifier) can confirm the contents of the message by providing the parameters of the signature to the judge without seeing any message secret. In case of any conflict, the proposed scheme assures public verifiability. If a conflict arises, the polling server transmits $(m, \bar{s}, Z, A)$ to the judge to resolve the dispute. The judge confirms the authenticity of the signer and the content of the message, ensuring that it was signed by the genuine person and not by somebody else.

7. Un-Traceability

Un-traceability ensures that the recipient of the message cannot trace down the sender. For computing parameters $(c_i, r_i, A_i, s_i, X_i)$, the voter utilizes arbitrary numbers as a private key, as described in the Section 3.2.3, and then transmits them to the polling station. As a result, neither the verifier nor the polling station can verify the legitimacy of the sender.

8. Unlinkability

Unlinkability refers to the inability to connect earlier messages with the sender of the current message. For instance, a voter transmits $r_1 = h(m_1||A)$, as mentioned in the equation (3.2) of the Section 3.2.3, to the polling station for signature. If another voter transmits $r_2 = h(m_2||A)$ to the polling station. The signer at the polling sation then includes it in the record list $L_i(r_1, r_2, ..., r_i)$. Subsequently, the pair (m_i, r_i) is produced by either the polling server or the polling station, so the signer/polling station cannot link r_i back to the original m_i.

9. Forward Secrecy

After the breakdown of a sustained relationship, an attacker cannot obtain the private keys d_{req} and d_{sign}, from Steps (1) and (2) of Section 3.2.2 of any contributor, nor they can access old messages. If the private key is misplaced, the presented e-voting scheme offers forward secrecy. The intruder cannot find out the private key from the old signcrypted message (c, r, s, A, X). Furthermore, the attacker has to calculate β from the equation (3.1) in the Section 3.2.3 as follows:

$$A = \beta \cdot G,$$

which is difficult due to the hardness of the elliptic curve discrete logarithm problem (ECDLP).

Chapter 4

New Generalized Blind Signcryption Scheme Based on Elliptic Curve for E-Voting Scheme

In this chapter, the Blind Signcryption Scheme presented in Chapter 3 will be extended to a Generalized Blind Signcryption Scheme. Particularly, a generalized blind signcryption scheme for e-voting system based on ECC is proposed. To make an e-voting system more secure, ECC is used. The scheme gives authenticity, integrity, unlinkability, unforgeability, confidentiality, forward secrecy and nonrepudiation, similar to the scheme presented in Chapter 3. In a generalized blind signcryption scheme, we have three modes: blind signcryption mode, encryption-only mode and blind signature-only mode. In the encryption-only mode, only the message will be encrypted without applying a signature. In the blind signature-only mode, only the signature will be generated. If both signing and encryption are required, then it will perform generalized blind signcryption. In this chapter, the three modes of operation will be discussed and the correctness of the scheme will be examined. Furthermore, a toy example to illustrate the scheme will also be presented.

4.1 Generalized Blind Signcryption Scheme

In this section, the Generalized Blind Signcryption Scheme is proposed, which divides the signcryption scheme into three modes: Blind Signcryption Mode, Blind Signature Only Mode, and Encryption Only Mode. The proposed scheme for the e-voting system utilizes elliptic curve cryptography and consists of three main steps:

1. Key Generation.
2. Signcryption.
3. Unsigncryption.

Scheme Participants

The participants in the proposed scheme, all of which are described in Section 3.2 of Chapter 3, are as follows:

1. Voter/Requester.
2. Signer/Polling station.
3. Verifier/Polling server.

Proposed scheme

The proposed scheme is structured in the following manner: Global Parameters, Key Generation Phase, Blind Signcryption Phase, Blind Unsigncryption Phase.

Global Parameters

The global parameters that are used in the proposed scheme are same as described in the Table 3.1 in Section 3.2.1 of Chapter 3.

4.1.1 Key Generation Phase

Every participant of the proposed scheme will choose their private key and calculate their public key. The steps for key generation for every participant of the proposed scheme are described below:

1. Voter/Requester chooses his/her private key $d_{req} \in \{1, \ldots, n-1\}$ arbitrarily. Then calculates his/her public key P_{req} using:

$$P_{req} = d_{req} \cdot G \mod p.$$

2. Signer/Polling station chooses his/her private key $d_{sign} \in \{1, \ldots, n-1\}$ arbitrarily. Then calculates his/her public key P_{sign} using:

$$P_{sign} = d_{sign} \cdot G \mod p.$$

3. Verifier/Polling server chooses his/her private key $d_{ver} \in \{1, \ldots, n-1\}$ arbitrarily. Then calculates his/her public key P_{ver} using:

$$P_{ver} = d_{ver} \cdot G \mod p.$$

4.1.2 Generalized Blind Signcryption Scheme

In this section, the steps for the Generalized Blind Signcryption Scheme will be discussed. In the signcryption phase, the voter/requester and the signer/polling station are involved. In the unsigncryption phase, the polling server/verifier is involved.

Signcryption Phase

Any voter/requester wants to send a message to the polling server. The steps to generate blind signcrypted message are described below:

Voter/Requester

After choosing secret keys as very large random integers β_1, β_2, and the message m the following computations will be performed at the voter/requester end.

1. Choose random integers β_1, β_2 or both $\in \{1, \ldots, n-1\}$.
2. If $\beta_2 =$ null, then take $A =$ null and calculate $r = h(m)$. Then go to step 8, else
3. Calculate elliptic curve point A by using β_2, provided $\beta_2 \neq$ Null:

$$A = \beta_2 \cdot G \mod p.$$

4. Calculate hash value r of message m concatenated with A:

$$r = h(m \,||\, A).$$

5. Use private key d_{req} of voter and public key P_{ver} of verifier and calculate key K using:

$$K = d_{req} \cdot P_{ver} = (K_1, K_2) \mod p. \tag{4.1}$$

6. Use key K_1 and elliptic curve point A to calculate secret encryption key k_{e^*}:

$$k_{e^*} = h(K_1 \cdot A).$$

7. If $\beta_1 =$ null, then take $T =$ null and calculate $c = E_{k_{e^*}}(m)$. Then go to step 13, else
8. Another elliptic curve point T will be calculated by using public key of verifier P_{ver} and β_1, provided $\beta_1 \neq$ Null:

$$T = (T_1, T_2) = \beta_1 \cdot P_{ver} \mod p.$$

9. Transmit (r, T) to signer/polling station.

Signer/Polling Station

After receiving (r, T) from the voter/requester, the signer/polling station has to perform the following steps to generate the signature. After choosing the secret key as very large random integer γ the following computations will be performed at the signer/polling station end.

1. Choose arbitrarily an integer $\gamma \in \{1, \ldots, n-1\}$.
2. Generate the signature $\bar{s}$ using the private key d_{sign} of the signer, blinding factor r and the random integer γ as:

$$\bar{s} = T_2 \cdot (d_{sign} + r \cdot \gamma) \mod n.$$

3. Transmit $\bar{s}$ to Voter/Requester.

Voter/Requester

After receiving $\bar{s}$ from the signer/polling station, the following computations will be performed at the voter/requester end:

10. Calculate the elliptic curve point Q by:

$$Q = \bar{s} \cdot G \mod p.$$

11. Calculate the ciphertext c for the message m together with $\bar{s}$ using the secret encryption key k_{e^*}:

$$c = E_{k_{e^*}}(m||\bar{s}).$$

OR if $\beta_2 =$ null then calculate c by concatenating message m with $\bar{s}$:

$$c = (m||\bar{s}).$$

12. Calculate signature s by using the voter's private key d_{req} and random integer β_1:

$$s = \frac{\beta_1}{\bar{s} + d_{req}} \mod n.$$

13. Transmit (c, r, A) to the polling server if $\beta_1 =$ null or transmit (s, Q, T) to the polling server if $\beta_2 =$ null, else transmit (c, r, s, Q, A, T) to the polling server/verifier.

Unsigncryption Phase

The verifier/polling server unsigncrypt the signcrypted text to verify the authenticity of the message. If it is verified, then it is accepted.
If the verifier receives (s, Q, T) then go to Step (6); if the verifier receives (c, r, A) then go to Step (1) and return after completing Step (5). If the verifier receives (c, r, s, A, Q, T), the following computation will be performed at the verifier/polling server's end:

1. First, find the key K using the verifier's private key d_{ver} and the voter/requester's public key P_{req}:

$$K = d_{ver} \cdot P_{req} = (K_1, K_2) \mod p \tag{4.2}$$

2. Using K_1 and A calculate the shared secret key k_{e^*} :

$$k_{e^*} = h(K_1 \cdot A).$$

3. Calculate plaintext message m by decrypting the ciphertext c using secret key k_{e^*}:

$$m||\bar{s} = D_{k_{e^*}}(c)$$

4. Calculate:

$$r^{'} = h(m\,||\,A)$$

5. If $r^{'} = r$ accept the message m as original; otherwise, reject it.

6. Verify the signature s by calculating the elliptic curve point T':

$$T' = d_{ver} \cdot s(Q + P_{req}) \mod n$$

7. If $T' = T$ then signature are authentic otherwise rejected.

The Generalized blind signcryption scheme is summarized in Table 4.1 given below:

Signcryption	**Unsigncryption**
Voter/Requester:	$K = d_{ver} \cdot P_{req} = (K_1, K_2)$
If $\beta_2 =$ Null take $A =$ Null;	$k_{e^*} = h(K_1 \cdot A)$
else calculate: $A = \beta_2 \cdot G$	$m\|\|\bar{s} = D_{k_{e^*}}(c)$
$r = h(m\|\|A)$	$r' = h(m\|\|A)$
$K = d_{req} \cdot P_{ver} = (K_1, K_2)$	Accepts if $r' = r$
$k_{e^*} = h(K_1 \cdot A)$	$T' = d_{ver} \cdot s(Q + P_{req})$
If $\beta_1 =$ Null take $T =$ Null;	Accepts if $T' = T$
else calculate: $T = \beta_1 \cdot P_{ver} = (T_1, T_2)$	
Signer/Polling Station:	
$\bar{s} = T_2 \cdot (d_{sign} + r \cdot \gamma)$	
Voter/Requester:	
$Q = \bar{s} \cdot G$	
$c = E_{k_{e^*}}(m\|\|\bar{s})$	
$s = \dfrac{\beta_1}{\bar{s} + d_{req}}$	
$(c, r, s, A, Q, T) \rightarrow$	

TABLE 4.1: Generalized Blind Signcryption Scheme

4.1.3 Blind signature only mode

If only authentication is required, then the voter/requester will perform the operations of the blind signature only mode. In blind signature only mode, to perform only operations for signatures on the message, he/she will set $\beta_2 =$ Null in Section 4.1.2 of Step (1) of voter/requester. For signatures, both the voter/requester and the signer/polling station are involved. Verification, on the other hand, is performed by the verifier/polling station. The scheme for blind signature-only mode is summarized in Table 4.2 given below:

Signature	**Verification**
Voter/Requester:	$T' = d_{ver} \cdot s(Q + P_{req})$
$r = h(m)$	Accepts if $T' = T$
$T = \beta_1 P_{ver} = (T_1, T_2)$	
Signer/Polling Station:	
$\bar{s} = T_2(d_{sign} + r\,.\gamma)$	
Voter/Requester:	
$Q = \bar{s} \cdot G$	
$s = \dfrac{\beta_1}{\bar{s} + d_{req}}$	
$(s, Q, T) \rightarrow$	

TABLE 4.2: Blind Signature Only Mode when β_2 = Null

4.1.4 Encryption Only mode

If only confidentiality is required then voter/requester will perform the operations of the encryption-only mode In encryption only mode, The voter/requester will set β_1 = Null in Section 4.1.2 of Step (1) of voter/requester. For encryption-only mode, only voter/requester is involved. However, verification is performed by the verifier/polling station. The scheme for encryption-only mode is summarized in Table 4.3 given below:

Voter/Requester	**Unsignryption**
$A = \beta_2 \cdot G$	$K = d_{ver} \cdot P_{req} = (K_1, K_2)$
$r = h(m\|\|A)$	$k_{e^*} = h(K_1.A)$
$K = d_{req} \cdot P_{ver} = (K_1, K_2)$	$m = D_{k_{e^*}}(c)$
$k_{e^*} = h(K_1 \cdot A)$	$r' = h(m\|\|A)$
$c = E_{k_{e^*}}(m)$	Accepts if $r' = r$
$(c, r, A) \rightarrow$	

TABLE 4.3: Encryption only mode when β_1 = Null

4.2 Correctness

The correctness of both the signature and encryption phases is proved in this section.

Theorem 4.2.1. The signatures in the above generalized blind signcryption scheme are correct if the verifier proves the following:

$$\beta_1 \cdot P_{ver} = d_{ver} \cdot s(Q + P_{req})$$

Proof:

$$\begin{aligned} d_{ver} \cdot s(Q + P_{req}) &= d_{ver} \cdot \frac{\beta_1}{\bar{s} + d_{req}}(\bar{s} \cdot G + d_{req} \cdot G) \\ &= d_{ver} \cdot \frac{\beta_1 \cdot G}{\bar{s} + d_{req}}(\bar{s} + d_{req}) \\ &= d_{ver} \cdot \beta_1 \cdot G \\ &= \beta_1 \cdot (d_{ver} \cdot G) \\ &= \beta_1 \cdot P_{ver} \end{aligned}$$

So, the signatures are authentic.

Now, it will be demonstrated that the key K plays the role of the shared secret key.

Theorem 4.2.2. The above proposed signcryption and unsigncryption are correct if the verifier proves the following:

$$d_{ver} \cdot P_{req} = d_{req} \cdot P_{ver}$$

Proof:

$$\begin{aligned} d_{ver} \cdot P_{req} &= d_{ver} \cdot (d_{req} \cdot G) \\ &= d_{req} \cdot (d_{ver} \cdot G) \\ &= d_{req} \cdot P_{ver} \end{aligned}$$

This shows that key K computed in equation (4.1) and (4.2) are the same.

4.3 Toy Example

The above generalized blind signcryption scheme is now illustrated with a toy example.

Example 4.3.1. A voter wants to send a message $m = 15$ to the polling server in a safe and trustworthy approach. Suppose an elliptic curve $y^2 = x^3 + 4x + 229$ where $a = 4$, $b = 229$ and $p = 503$. The points $E_{\mathbb{F}_p}(a, b) = E_{503}(4, 229)$ generate the elliptic curve group. Suppose the base point $G = (220, 174)$ and the order of base point is $n = 541$ where $541 \cdot G = \mathcal{O}$. There are 542 total points in the elliptic curve indicates that $|E_{503}(4, 229)| = 541$.

S-DES will be used for encryption/decryption, and CRC-8 will be utilized to calculate the hash value.

First, generate keys by following the key generation algorithm in section 4.1.1.

Key Generation

1. Voter selects his private key as: $d_{req} = 10$, and then calculates his public key P_{req} as:

$$\begin{aligned} P_{req} &= d_{req} \cdot G \mod p \\ &= 10\,(220, 174) \mod 503 \\ &= (138, 275) \end{aligned}$$

2. Signer selects his private key as: $d_{sign} = 5$, and then calculates his public key P_{sign} as:

$$\begin{aligned} P_{sign} &= d_{sign} \cdot G \mod p \\ &= 5\,(220, 174) \mod 503 \end{aligned}$$

$$= (381, 476)$$

3. Verifier selects his private key as: $d_{ver} = 2$, and then calculate his public key P_{ver} as:

$$\begin{aligned} P_{ver} &= d_{ver} \cdot G \mod p \\ &= 2\,(220, 174) \mod 503 \\ &= (385, 480) \end{aligned}$$

Generalized Blind Signcryption

Voter/Requester:

1. Chooses arbitrarily $\beta_1 = 3, \beta_2 = 4$.

2. Compute elliptic point A using:

$$\begin{aligned} A &= \beta_2 \cdot G \mod p \\ &= 4\,(220, 174) \mod 503 \\ &= (480, 374) \end{aligned}$$

3. Compute hash value of message m together with A by using CRC-8 algorithm:

$$\begin{aligned} r &= h(m||A) \mod n \\ &= h(15(480, 374)) \mod 541 \\ r &= 127 \end{aligned}$$

4. Compute keys K by using private key d_{req} of voter and public key P_{ver} of verifier:

$$K = d_{req} \cdot P_{ver} \mod p$$

$$\begin{aligned} &= 10 \cdot (385, 480) \mod 503 \\ K = (K_1, K_2) &= (279, 343) \end{aligned} \tag{4.3}$$

5. Compute secret encryption key k_{e^*}:

$$\begin{aligned} k_{e^*} &= h(K_1 \cdot A) \mod n \\ &= h(279 \cdot (480, 374)) \\ &= h(147, 186) \\ k_{e^*} &= 402 \end{aligned}$$

6. Compute elliptic point T using:

$$\begin{aligned} T &= \beta_1 \cdot P_{ver} \mod p \\ &= 3(385, 480) \mod 503 \\ T = (T_1, T_2) &= (430, 285) \end{aligned}$$

7. Transmit (r, T) to signer.

Signer/Polling station

1. Chooses arbitrarily $\gamma = 6$.

2. Compute signature $\bar{s}$ using:

$$\begin{aligned} \bar{s} &= T_2 \cdot (d_{sign} + r \cdot \gamma) \mod n \\ &= 285 \cdot (5 + 127 \cdot 6) \mod 541 \\ &= 285 \cdot (5 + 762) \mod 541 \\ &= 31 \end{aligned}$$

3. Transmit $\bar{s}$ to voter.

Voter/Requester

1. Compute ciphertext c using S-DES [50] a symmetric key encryption:

$$\begin{aligned} c &= E_{k_{e^*}}(m||\bar{s}) \mod n \\ &= E_{402}(15||31) \mod 516 \\ c &= 0011001011110111 \end{aligned}$$

2. Compute signatures s using:

$$\begin{aligned} s &= \frac{\beta_1}{\bar{s} + d_{req}} \mod n \\ &= \frac{3}{31 + 10} \mod 541 \\ &= 3(41)^{-1} \mod 541 \\ s &= 198 \end{aligned}$$

3. Compute elliptic point Q using:

$$\begin{aligned} Q &= \bar{s} \cdot G \mod p \\ &= 31 \cdot (220, 174) \mod 503 \\ Q &= (158, 12) \end{aligned}$$

4. Transmit (c, r, s, Q, A) to verifier.

Unsigncryption

1. Compute K using:

$$\begin{aligned} K &= d_{ver} \cdot P_{req} \mod p \\ &= 2(138, 275) \mod 503 \\ K = (K_1, K_2) &= (279, 343) \end{aligned}$$

Note that this K is same as computed by the voter/requester in equation (4.3).

2. Compute secret key k_{e^*} using:

$$\begin{aligned} k_{e^*} &= h(K_1 \cdot A) \\ &= h(279(480, 374)) \\ &= h(147, 186) \\ k_{e^*} &= 402 \end{aligned}$$

3. Decrypt the ciphertext c by using the decryption algorithm of S-DES [50]:

$$\begin{aligned} m||\bar{s} &= D_{k_{e^*}}(c) \mod n \\ &= D_{402}(0011001011110111) \\ m||\bar{s} &= 15||31 \end{aligned}$$

4. Compute T using:

$$\begin{aligned} T' &= d_{ver} \cdot s \cdot (Q + P_{req}) \mod p \\ &= 2 \cdot 198\,((158, 12) + (138, 275)) \mod 503 \\ &= 396\,((158, 12) + (138, 275)) \mod 503 \\ T' = (T_1', T_2') &= (430, 285) \end{aligned}$$

5. Compute r' using:

$$\begin{aligned} r' &= h(m||A) \\ &= h(15||(480, 374)) \\ r' &= 127 \end{aligned}$$

6. As $r = 127$ and $r' = 127$, so the vote is acceptable.

Chapter 5

Security Analysis

In this chapter, the security analysis of the generalized blind signcryption scheme proposed in Chapter 4 is discussed. The security of the scheme relies on the Elliptic Curve Discrete Logarithm Problem (ECDLP). Afterwards, the cost analysis is presented and compared with the other existing schemes.

5.1 Security Characteristics

In this section, the security characteristics will be discussed. The proposed scheme provides confidentiality, integrity, unforgeability, authenticity, untraceability, unlinkability and public verifiability. The security of this scheme depends on the ECDLP.

1. Confidentiality

In this property, the attacker is unable to access the message contents. To access the message contents, the attacker needs the secret encryption key k_{e^*}, which is hard to find due to the ECDLP.

Case-I: To attack the system, the attacker needs the private key of voter/ requester d_{req} from Step (1) of Key Generation Phase 4.1.1,

$$P_{req} = d_{req} \cdot G$$

and the key K from the Step (5) of voter/requester in Signcryption Phase 4.1.2 to find the secret key for encryption k_{e^*}.

$$K = (K_1, K_2) = d_{req} \cdot P_{ver}$$
$$k_{e^*} = h(K_1 \cdot A)$$

But finding d_{req} is difficult for an attacker due to ECDLP.

Case-II: To calculate the secret encryption key k_{e^*}, the attacker needs A from the Step (3) of voter/requester in Section (4.1.2), which is infeasible due to ECDLP.

$$A = \beta_2 \cdot G$$
$$k_{e^*} = h(K_1 \cdot A)$$

2. Integrity

In this property, the attacker cannot alter the message during transmission over a noisy medium. In the proposed scheme, a hash function is used for blinding the message contents. To find the value of r in the Section 4.1.2 of Step (4) of voter/requester the hash function is used.

$$r = h(m||A)$$

If the attacker alters $m \rightarrow m'$, then the values of $(c, s, r, \bar{s})$ will also change to new values. That is, $r \rightarrow r'$, $c \rightarrow c'$, $s \rightarrow s'$ and $\bar{s} \rightarrow \bar{s}'$. As a result, the value of r' from the Section 4.1.2 of Step (4) of unsigncryption changes to r'' on the server side as well due to the hash function random oracle property.

$$r' = h(m||A)$$
$$r'' = h(m'||A)$$

So, if the message contents are changed during transmission, the verifier will recognize that the message was altered along the way.

3. Unforgeability

The proposed scheme also offers unforgeability. In this property, the attacker cannot forge values of (c, r, s) in Section 4.1.2. To forge the values, attacker needs $(d_{sign}, \beta_1, \beta_2, r)$ from Section 4.1.2 to calculate:

$$s = \frac{\beta_1}{\bar{s} + d_{req}}$$
$$\bar{s} = T_2(d_{sign} + r \cdot \gamma)$$

However, it is difficult to find an authentic signature because the signer's private key d_{sign} and the requester's private key d_{req} from the Key Generation Phase of Section 4.1.1 is used for generating the signature in Step (12) of voter/requester in Section 4.1.2.

$$P_{sign} = d_{sign} \cdot G$$
$$P_{req} = d_{req} \cdot G$$

Finding the private key d_{sign} and d_{req} of the signer and the requester is hard to compute due to ECDLP.

4. Authentication

The proposed scheme provides authenticity by generating the signature s using the requester's private key d_{req} from the Key Generation Phase 4.1.2.

$$P_{req} = d_{req} \cdot G$$
$$s = \frac{\beta_1}{\bar{s} + d_{req}}$$

Then the receiver/verifier verifies the signature in Step (6) of the Unsigncription Phase 4.1.2 using his/her private key d_{ver} and the public key P_{req} of requester from the Key Generation Phase 4.1.2.

$$P_{req} = d_{req} \cdot G$$
$$T' = d_{ver} \cdot s(Q + P_{req})$$

Finding d_{req} and d_{ver} is difficult to calculate due to ECDLP. Therefore, the attacker cannot forge the signatures.

5. Un-Traceability

This scheme provides un-traceability. In the proposed scheme, the attacker or the receiver of the message cannot trace the message sender. The voter/requester uses arbitrary private integers β_1, β_2 to calculate parameters (c, r, s) in the Signcryption Phase 4.1.2. Finding β_1 and β_2 is hard because of ECDLP. So, the attacker cannot calculate the parameters (c, r, s). As a result, the receiver cannot trace the message sender.

6. Unlinkability

The voter sends r to the polling station for generating the signature $\bar{s}$ in the Step (2) of the signer/polling station in the Section 4.1.2,

$$r = h(m||A)$$
$$\bar{s} = T_2(d_{sign} + r \cdot \gamma)$$

The signer then keeps the record $(r_1, r_2, ..., r_j)$ for messages $(m_1, m_2, ..., m_j)$. Later, if the signer/polling station wants to link the old messages, he/she cannot link r_j with the message m_j. Because the same pair (m_j, r_j) is produced by either the polling station or the polling server.

7. Public Verifiability

In this property, if there occurs any misunderstanding, a third party can verify if the signature are authentic or not and that the document has not been altered. Anyone can verify the signature by sending the signature parameters (c, r, s) to the judge. In the proposed scheme, the signer generates the signature $\bar{s}$ using his/her private key d_{sign} in Step (2) of the Signer/Polling Station in Section 4.1.2, and then the secret arbitrary integer β_1 is used to generate the signature s that only voter/requester knows in Step (9) of the Section 4.1.2. Therefore, neither the signer nor the voter/requester can deny their signatures.

8. Non-Repudiation

In this property, if there is any dispute, the sender cannot deny that he/she did not send the message. One can prove this by sending the parameters to the third verifier/judge. In the proposed scheme, the signer generates the signature in Step (2) of the signer/polling station in Section 4.1.2 using his/her private key d_{sign} which is hard to compute due to ECDLP.

$$P_{sign} = d_{sign} \cdot G$$

Therefore, no one other then the signer can generate the signature. Furthermore, the voter/requester uses an arbitrary number β_1 to generate the signature s in Step (9) of the voter/requester in Section 4.1.2, the arbitrary number β_1 is known only to the voter/requester. As a result, no one other than the voter/requester can sign the message content. Consequently, the sender cannot deny his/her message.

9. Forward Secrecy

In this property, the attacker cannot find the secret keys d_{req} and d_{sign} of voter/requester and signer/polling station from the Key Generation Phase 4.1.1 after the long term communication has ended or is lost. Whenever a new message is being encrypted, the value of β_2 will be changed. As a result, the attacker cannot recover the message from the previous communications. To retrieve the old messages, one need β_2 from the Step (2) of voter/requester of Section 4.1.2, but finding β_2 is hard due to ECDLP.

5.2 Computational Cost

In this section, the computational cost will be discussed. For the computational cost the approach presented in [38] is used. That is, we focus on the number of various operations involved in the proposed scheme.

The operations used in the proposed Generalized Blind Signcryption scheme are

described in comparison to the previous scheme, as shown in Table 5.1. In this table, the 'M-Exp' represents 'exponentiation multiplication' and 'Mul' represents 'scalar multiplication'. The displayed numbers indicate that how frequently an operation is performed across all sections, requester, signer and verifier of the scheme.

Scheme		**M-Exp**	**Mul**
Ullah et al. [30]	Requester	-	3
	Signer	-	1
	Verifier	-	2
Yu and He [27]	Requester	7	-
	Signer	2	-
	Verifier	2	-
Awasthi and Lal [26]	Requester	4	-
	Signer	1	-
	Verifier	2	-
Waheed et al. [38]	Requester/Voter	-	2
	Signer/Polling Station	-	1
	Verifier/Polling Server	-	2
Han et al. [65]	Requester/Voter	-	2
	Verifier/Polling server	-	3
Zhou [66]	Signcryption	-	5
	Unsigncryption	-	7
Proposed Scheme	Requester/voter	-	5
	Signer/Polling Station	-	-
	Verifier/Polling Server	-	3

TABLE 5.1: Comparison of operations of other schemes with proposed Generalized Blind Signcryption Scheme

In this scheme, using security controller Infineon SLE66CUX640P [67], the Computation time for for exponentiation multiplication (M-Exp) is 220 milliseconds

(ms), and for scalar multiplication (Mul), it is 83 milliseconds (ms). The table below provides a cost comparison between various existing blind signcryption schemes (BSS) and the generalized blind signcryption schemes (GBSS) with the proposed Generalized Blind Signcryption scheme.

Schemes	**Feature**	**Computational time**	
		Mul (83 ms)	M-Exp (220 ms)
Ullah et al. [30]	BSS	$6 \times 83 = 498$	-
Yu and He [27]	BSS	-	$11 \times 220 = 2420$
Awashthi and Lal [26]	BSS	-	$7 \times 220 = 1540$
Waheed et al. [38]	BSS	$5 \times 83 = 415$	-
Han et al. [65]	GBSS	$5 \times 83 = 415$	-
Zhou, Caixue [66]	GBSS	$12 \times 83 = 996$	-
Proposed Scheme	GBSS	$8 \times 83 = 664$	-

TABLE 5.2: Computational Cost

The computational cost in the proposed scheme is higher compared to some other schemes due to its nature as a Blind Signcryption Schemes. While certain schemes involve only two participants, our scheme involves three participants, resulting in a Generalized Blind Signcryption scheme. In the generalized blind signcryption scheme, the computational cost varies due to the involvenment of three different modes. Consequently, by summing up the costs associated with these three modes, blind signcryption mode, encryption-only mode and the blind signature-only mode, the computational cost of the Generalized Blind Signcryption Scheme exceeds that of other schemes. The computational cost for the different mode is described below:

Encryption Only Mode

In the encryption mode, compute the cost of the operations that will be used for encryption in algorithm 4.1.4. The number of operation that are used for the

encryption mode and the cost is described below:

Scheme participants	**M-Exp**	**Mul**
Voter/Requester	-	3
Verifier/polling server	-	2
Total operations		5
Total cost		$5 \times 83 = 415$

TABLE 5.3: Computational Cost for encryption mode

Signature Only Mode

In the Signature only mode, compute the cost of the operations that are used for signature in Section 4.1.3. The number of operations that are used and the cost is described below:

Scheme participants	**M-Exp**	**Mul**
Voter/Requester	-	2
Signer/Polling station	-	-
Verifier/Polling server	-	1
Total operations	-	3
Total cost		$3 \times 83 = 249$

TABLE 5.4: Computational Cost of Signature Only Mode

5.3 Cryptanalysis attack

In this section, various cryptanalysis attacks will be discussed, and their analyses will be presented in detail. These attacks will determine whether the proposed scheme is resilient to attack or not.

5.3.1 Brute Force Attack

In this attack, the attacker uses a hit or trial method to find the secret key. In the proposed scheme, to find the secret keys of the signer/polling station d_{sign} or of the voter/requester d_{req} in the Section (4.1.1), the attacker needs to compute the ECDLP, which is difficult to compute. The elliptic curve cryptosystem employs a 160 bit key size. Thus, if the order of the elliptic curve is $\geq 2^{163}$ then the scheme is considered to be on the safe side [4]. The order of the elliptic curve is $\geq 2^{224}$ which means the key space is very large. Consequently, it will take a significant amount of time for an attacker to find the secret keys of the signer or the voter. Thus, the scheme is secure against Brute Force Attack.

5.3.2 Ciphertext only Attack

In this attack, the attacker has access to the ciphertext through publicly available data. The primary aim of the attacker is to find out the private key or the original message. If the attacker discovers the private key, he/she can determine all the plaintexts that are encrypted using that key.

Let's assume the attacker has information about ciphertext c, and his/her goal is to calculate the private key or the original message. To obtain the private key d_{req} from the Step (1) of the voter/requester in Key Generation Phase 4.1.1, the attacker must solve the elliptic curve discrete logarithm problem (ECDLP), which is a hard problem to solve. Therefore, the attacker cannot find out the private key d_{req} and, as a result, he/she cannot find the plaintext message m from the ciphertext c. Consequently, the proposed scheme is secure against Ciphertext Only Attack.

5.3.3 Known Plaintext Attack

In this attack, the attacker has access to some information about the plaintext and its corresponding ciphertext. The primary goal of the attacker is to find the secret

key so that he/she can obtain the plaintext or gain additional information about it. To find the secret encryption key k_{e^*}, from the Step (6) of the voter/requester in Section (4.1.2), the attacker needs K_1 from the Step (5) of the voter/requester from Section (4.1.2). However, finding K_1 is difficult because it involves the private key d_{req} of requester/voter, which is computationally difficult to derive from the Step (1) of Key Generation Phase 4.1.1, due to ECDLP. Additionally, the secret shared key k_{e^*} involves a hash function, and due to the hash function random oracle property, it is impossible to find the secret shared key k_{e^*}. As a result, the proposed scheme is safe against this attack.

5.3.4 Chosen Ciphertext Attack

In this attack, the attacker chooses random ciphertext of his/her own choice and obtains their corresponding plaintext. The primary aim of the attacker is to find the secret key or the secret parameters. In the proposed scheme, the attacker chooses his/her ciphertext c of his/her own choice and obtains the plaintext message m corresponding to the chosen ciphertext. However, given c and m, it is impossible to find the shared secret key k_{e^*} because it involves another parameter A, which in turn involves the secret integer β_2 from the Step (3) of voter/requester in the Section 4.1.2. Finding the secret integer β_2 from the Step (3) of the voter/requester in the Section 4.1.2 is hard to find for the attacker due to ECDLP. Therefore, the scheme is secure against Chosen Ciphertext Attack.

5.3.5 Forgery Attack

In forgery attack, an attacker targets the network communication between the requester/voter and the polling server/verifier. To achieve this, the attacker forges the values of message m in Step (4) of the voter/requester or the signature s in Step (12) of the voter/requester in the Section 4.1.2, in order to generate his/her own signature or message. In the proposed scheme, if an attacker attempts to forge the values of message m, the values of (c, s, r) are changed to (c', s', r') in the

Signcryption Phase 4.1.2. Subsequently, the attacker sends these altered values to the verifier/polling server. However, the verifier/polling station cannot verify the values. In the Signcryption Phase 4.1.2, to forge the values of the signature s in Step (12) of the voter/requester, an attacker would require to know the private key d_{ver} of voter/requester from the Step (1) of Key Generation Phase 4.1.1, which he/she cannot find out due to ECDLP. Moreover, the attacker would also need to determine the value of r from Step (4) performed by voter/requester in Signcryption Phase 4.1.2, which is computationally infeasible due to the involvement of a hash function with a unique output in calculating r. Consequently, the Unsigncryption Phase 4.1.2 would be unable to verify any falsified values within the signcrypted text in Step (4) of the unsigncryption phase. Thus, the overall scheme remains secure against forgery attacks.

5.3.6 Man In The Middle Attack

In this attack, an attacker places themselves between the communication of two parties and eavesdrops on their communication. He/She wants to alter the message, either completely or partially, while pretending that the normal conversation is ongoing. The objective of the attacker is to create mutual shared secret keys for both the verifier/polling server and the voter/requester. To achieve this, the attacker selects the secret key d_{att} and calculate the public key $P_{att} = d_{att} \cdot G$ which is an elliptic point. To establish honest communication with both parties, the attacker needs to create mutual secret keys. The attacker uses his/her public key P_{att} to create a secret key, but he/she cannot create a valid key because the attacker has no knowledge about the secret number β_2 from the Step (3) of the voter/requester in Section4.1.2, and the hash function is involved for generating the secret key k_{e^*} in Step (6) of the voter/requester in Section4.1.2. Consequently, the attacker cannot create a valid secret key. Therefore, the scheme is safe from this attack.

Chapter 6

Conclusion

In this thesis, Waheed et al.'s blind signcryption scheme [38] "Novel Blind Signcryption Scheme for E-Voting System Based on Elliptic Curves" is examined. Their scheme provides the characteristics of forward secrecy, unlinkability, and non-repudiation along with the basic characteristics of confidentiality, authentication, integrity and unforgeability. The scheme simultaneously offers both confidentiality and authenticity. In this scheme, some notational mistakes are made, and some parameters/variables used in their scheme are not properly defined. For instance, T is not defined in their scheme [38]. In this thesis, the generalized blind signcryption scheme is proposed by enhancing the blind signcryption scheme presented in Section 3.2 of Chapter 3. In the proposed generalized blind signcryption scheme, blind signcryption mode, encryption mode and blind signature mode are adaptable as needed. If the user wants only authenticity the blind signature mode will be used, if the user wants confidentiality then encryption mode will be used, and if both the confidentiality and authenticity is required then blind signcryption will be used. The security of this scheme depends on the hardness of ECDLP and the hash function random oracle property. This scheme provides the security features of blindness, integrity, unforgeability, untraceability, unlinkability, non-repudiation and forward secrecy. The correctness of the scheme shows that our scheme is accurate. The resistance of the scheme against known ciphertext attacks is also highlighted in Chapter 5, which demonstrates that the scheme is resilient

against from cryptanalysis attacks and is both efficient and authentic. The cost computation of the scheme is computed for the different modes. The computational cost for the different mode is not the same because of the different number of operations involved in those modes.

As a future work, this study can be extended further in some interesting directions. For instance, one can extend the proposed scheme to the following:

- ID-based generalized blind signcryption scheme in the setting of hyperelliptic or elliptic curve.
- Proxy-based generalized blind signcryption scheme.

Bibliography

[1] G. C. Kessler, "An overview of cryptography," *Handbook on Local Area Networks*, 2003.

[2] G. Brassard, *Modern Cryptology: A Tutorial*, vol. 325. Springer, 1988.

[3] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory,*, vol. 22, no. 6, 1976.

[4] W. Stallings, *Cryptography and network security, 4/E.* Pearson Education India, 2006.

[5] D. E. R. Denning, *Cryptography and data security*, vol. 112. Addison-Wesley Reading, 1982.

[6] R. Deepthi, "A survey paper on playfair cipher and its variants," *Int. Res. J. Eng. Technol*, vol. 4, no. 4, pp. 2607–2610, 2017.

[7] W. Tuchman, "A brief history of the data encryption standard," in *Internet besieged: countering cyberspace scofflaws*, pp. 275–280, 1997.

[8] A. M. Abdullah *et al.*, "Advanced encryption standard (AES) algorithm to encrypt and decrypt data," *Cryptography and Network Security*, vol. 16, pp. 1–11, 2017.

[9] W. Diffie and M. E. Hellman, "Multiuser cryptographic techniques," in *Proceedings of the June 7-10, 1976, National Computer Conference and Exposition*, pp. 109–112, 1976.

[10] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[11] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE transactions on information theory*, vol. 31, no. 4, pp. 469–472, 1985.

[12] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[13] V. Pachghare, *Cryptography and information security*. PHI Learning Pvt. Ltd., 2019.

[14] J. F. Dooley, *History of cryptography and cryptanalysis: Codes, Ciphers, and their Algorithms*. Springer, 2018.

[15] J. Chia, J. J. Chin, and S. C. Yip, "Digital signature schemes with strong existential unforgeability," 2021.

[16] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption) $\ll$ cost (signature) + cost (encryption)," in *Advances in Cryptology CRYPTO 97: 17th Annual International Cryptology Conference Santa Barbara, California, USA August 17-21, 1997 Proceedings 17*, pp. 165–179, Springer, 1997.

[17] Y. Zheng and H. Imai, "How to construct efficient signcryption schemes on elliptic curves," *Information processing letters*, vol. 68, no. 5, pp. 227–233, 1998.

[18] M. Toorani and A. Beheshti, "Cryptanalysis of an efficient signcryption scheme with forward secrecy based on elliptic curve," in *2008 International Conference on Computer and Electrical Engineering*, pp. 428–432, IEEE, 2008.

[19] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology: Proceedings of Crypto 82*, pp. 199–203, Springer, 1983.

[20] D. Chaum, "Blinding for unanticipated signatures," in *Advances in Cryptology - EUROCRYPT 87, Workshop on the Theory and Application of Cryptographic Techniques, Amsterdam, The Netherlands, April 13-15, 1987, Proceedings*, vol. 304 of *Lecture Notes in Computer Science*, pp. 227–233, Springer, 1987.

[21] S. Brands, "Untraceable off-line cash in wallet with observers," in *Advances in Cryptology Crypto93: 13th Annual International Cryptology Conference Santa Barbara, California, USA August 22-26, 1993 Proceedings 13*, pp. 302–318, Springer, 1994.

[22] D. Pointcheval and J. Stern, "New blind signatures equivalent to factorization," in *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pp. 92–99, 1997.

[23] C. I. Fan and C. L. Lei, "Low-computation partially blind signatures for electronic cash," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 81, no. 5, pp. 818–824, 1998.

[24] H. F. Huang and C. C. Chang, "An untraceable electronic cash system using fair blind signatures," in *2006 IEEE International Conference on e-Business Engineering (ICEBE 06)*, pp. 39–46, IEEE, 2006.

[25] M. Nikooghadam and A. Zakerolhosseini, "An efficient blind signature scheme based on the elliptic curve discrete logarithm problem," *ISeCure-The ISC International Journal of Information Security*, vol. 1, no. (2), pp. 125–131, 2009.

[26] A. K. Awasthi and S. Lal, "An efficient scheme for sensitive message transmission using blind signcryption," *arXiv preprint cs/0504095*, 2005.

[27] X. Yu and D. He, "A new efficient blind signcryption," *Wuhan University Journal of Natural Sciences*, vol. 13, no. 6, pp. 662–664, 2008.

[28] K. Chakraborty and J. Mehta, "A stamped blind signature scheme based on elliptic curve discrete logarithm problem.," *Int. J. Netw. Secur.*, vol. 14, no. 6, pp. 316–319, 2012.

[29] M. Dhanashree and M. Agrawal, "Implementation of blind digital signature using ECC," *International Journal of Computer Science and Network*, vol. 1, no. 5, 2012.

[30] R. Ullah, A. I. Umar, N. ul Amin, and Nizamuddin, "Blind signcryption scheme based on elliptic curves," in *2014 Conference on Information Assurance and Cyber Security (CIACS)*, pp. 51–54, IEEE, 2014.

[31] Y. Han and X.-Y. Yang, "New ecdsa-verifiable generalized signcryption," *CHINESE JOURNAL OF COMPUTERS-CHINESE EDITION-*, vol. 29, no. 11, p. 2003, 2006.

[32] Y. Han and X. Gui, "Adaptive secure multicast in wireless networks," *International Journal of Communication Systems*, vol. 22, no. 9, pp. 1213–1239, 2009.

[33] G. Yu, X. Ma, Y. Shen, and W. Han, "Provable secure identity based generalized signcryption scheme," *Theoretical Computer Science*, vol. 411, no. 40-42, pp. 3614–3624, 2010.

[34] P. Kushwah and S. Lal, "An efficient identity based generalized signcryption scheme," *Theoretical Computer Science*, vol. 412, no. 45, pp. 6382–6389, 2011.

[35] C. Zhou, W. Zhou, and X. Dong, "Provable certificateless generalized signcryption scheme," *Designs, Codes and Cryptography*, vol. 71, pp. 331–346, 2014.

[36] C. X. Zhou, "Identity based generalized proxy signcryption scheme," *Information Technology and Control*, vol. 45, no. 1, pp. 13–26, 2016.

[37] A. Zhang, L. Wang, X. Ye, and X. Lin, "Light-weight and robust security-aware d2d-assist data transmission protocol for mobile-health systems," *IEEE*

Transactions on Information Forensics and Security, vol. 12, no. 3, pp. 662–675, 2016.

[38] A. Waheed, N. Din, A. I. Umar, R. Ullah, and U. Amin, "Novel blind signcryption scheme for e-voting system based on elliptic curves," *Mehran University Research Journal Of Engineering & Technology*, vol. 40, no. 2, pp. 314–322, 2021.

[39] J. B. Fraleigh, *A first course in abstract algebra*. Pearson Education India, 2003.

[40] D. Cox, J. Little, and D. OShea, *Ideals, Varieties, and Algorithms: An introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2013.

[41] C. Paar and J. Pelzl, *Understanding Cryptography: A textbook for students and practitioners*. Springer Science & Business Media, 2009.

[42] G. Maze, *Algebraic methods for constructing one-way trapdoor functions*. University of Notre Dame, 2003.

[43] B. Kapoor and P. Pandya, "Chapter 2 - Data Encryption," in *Cyber Security and IT Infrastructure Protection* (J. R. Vacca, ed.), pp. 29–73, Boston: Syngress, 2014.

[44] M. Kamarzarrin, S. E. Hosseini, M. H. M. Zavareh, and M. Kamarzarrin, "Designing and implementing of improved cryptographic algorithm using modular arithmetic theory," *Journal of Electrical Systems and Information Technology*, vol. 2, no. 1, pp. 14–17, 2015.

[45] T. St Denis, *Cryptography for developers*. Elsevier, 2006.

[46] P. Wang and F. Zhang, "Computing elliptic curve discrete logarithms with the negation map," *Information Sciences*, vol. 195, pp. 277–286, 2012.

[47] D. Liu, *Next generation SSH2 implementation: Securing data in motion*. Syngress, 2011.

[48] J. Thakur and N. Kumar, "DES, AES and Blowfish: Symmetric key cryptography algorithms simulation based performance analysis," *International journal of emerging technology and advanced engineering*, vol. 1, no. 2, pp. 6–12, 2011.

[49] T. Nie and T. Zhang, "A study of DES and Blowfish encryption algorithm," in *Tencon 2009-2009 IEEE Region 10 Conference*, pp. 1–4, IEEE, 2009.

[50] I. R. Sharma and V. Gupta, "Comparative analysis of des and s-des encryption algorithm using verilog coding," *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, vol. 1, no. 9, pp. 469–473, 2013.

[51] D. W. Kravitz, "Digital signature algorithm," July 27, 1993. US Patent 5,231,668.

[52] G. Raju and R. Akbani, "Elliptic curve cryptosystem and its applications," in *SMC 03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483)*, vol. 2, pp. 1540–1543, IEEE, 2003.

[53] D. Mahto and D. K. Yadav, "Performance Analysis of RSA and Elliptic Curve Cryptography.," *Int. J. Netw. Secur.*, vol. 20, no. 4, pp. 625–635, 2018.

[54] L. D. Singh and K. M. Singh, "Implementation of text encryption using elliptic curve cryptography," *Procedia Computer Science*, vol. 54, pp. 73–82, 2015.

[55] S. Goldwasser, S. Micali, and R. L. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM Journal on computing*, vol. 17, no. 2, pp. 281–308, 1988.

[56] N. Liu, Y. Yin, X. Wu, and L. Ye, "Rfid cryptographic protocol based on cyclic redundancy check for high efficiency," *Journal of Sensors & Transducers*, vol. 168, no. 4, pp. 197–202, 2014.

[57] A. Grover and S. Singh, "Comparative Analysis of CRC-32 and SHA-1 Algorithms in WEP," *Advanced Engineering Technology and Application*, vol. 4, no. 1, pp. 5–10, 2015.

[58] K. T. Dave, "Brute-force attack seeking but distressing," *Int. J. Innov. Eng. Technol. Brute-force*, vol. 2, no. 3, pp. 75–78, 2013.

[59] Y. Zheng and J. Seberry, "Practical approaches to attaining security against adaptively chosen ciphertext attacks," in *Advances in Cryptology CRYPTO 92: 12th Annual International Cryptology Conference Santa Barbara, California, USA August 16-20, 1992 Proceedings*, pp. 292–304, Springer, 2001.

[60] R. Anderson, *Security engineering: A guide to building dependable distributed systems*. John Wiley & Sons, 2020.

[61] G. N. Nayak and S. G. S, "Different flavours of man-in-the-middle attack, consequences and feasible solutions," in *2010 3rd International Conference on Computer Science and Information Technology*, vol. 5, pp. 491–495, IEEE, 2010.

[62] D. Chaum, "Blind signature system," in *Advances in Cryptology: Proceedings of Crypto 83*, pp. 153–153, Springer, 1983.

[63] R. Ma and L. Du, "Attribute-based blind signature scheme based on elliptic curve cryptography," *IEEE Access*, vol. 10, pp. 34221–34227, 2022.

[64] E. Mohammed, A.-E. Emarah, and K. El-Shennaway, "A blind signature scheme based on elgamal signature," in *Proceedings of the Seventeenth National Radio Science Conference. 17th NRSC'2000 (IEEE Cat. No. 00EX396)*, pp. C25–1, IEEE, 2000.

[65] Y. Han, X. Yang, P. Wei, Y. Wang, and Y. Hu, "ECGSC: Elliptic curve based generalized signcryption," in *International conference on ubiquitous intelligence and computing*, pp. 956–965, Springer, 2006.

[66] C. Zhou, "An improved lightweight certificateless generalized signcryption scheme for mobile-health system," *International Journal of Distributed Sensor Networks*, vol. 15, no. 1, p. 1550147718824465, 2019.

[67] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pp. 62–73, 1993.

Turnitin Originality Report

Generalized Blind Signcryption Scheme For E-Voting System Based on ECC by Tooba Muhammad

From CUST Library (MS Thesis)

- Processed on 17-Aug-2023 14:16 PKT
- ID: 2146993053
- Word Count: 16407

Graduate Studies Office
16%
Within Limits

Similarity Index
16%
Similarity by Source

Internet Sources:
10%
Publications:
11%
Student Papers:
5%

sources:

1 1% match (Internet from 11-Jan-2023)
https://oaji.net/articles/2021/2712-1616870727.pdf

2 < 1% match (student papers from 04-Apr-2019)
Submitted to Higher Education Commission Pakistan on 2019-04-04

3 < 1% match (student papers from 11-May-2011)
Submitted to Higher Education Commission Pakistan on 2011-05-11

4 < 1% match (student papers from 08-Apr-2011)
Submitted to Higher Education Commission Pakistan on 2011-04-08

5 < 1% match (student papers from 17-Dec-2022)
Submitted to Higher Education Commission Pakistan on 2022-12-17

6 < 1% match (student papers from 04-Jul-2023)
Submitted to Higher Education Commission Pakistan on 2023-07-04

7 < 1% match (student papers from 06-Dec-2017)
Submitted to Higher Education Commission Pakistan on 2017-12-06

8 < 1% match (Internet from 03-Feb-2023)
https://thesis.cust.edu.pk/UploadedFiles/Syed%20Burhan--MMT161008.pdf

9 < 1% match ("Encyclopedia of Cryptography and Security", Springer Science and Business Media LLC, 2005)
"Encyclopedia of Cryptography and Security", Springer Science and Business Media LLC, 2005

10 < 1% match (student papers from 25-Mar-2019)
Submitted to Universidad de La Laguna on 2019-03-25

11 < 1% match (Malik Zia Ullah Bashir, Rashid Ali. "Cryptanalysis and improvement of a blind multi-document signcryption scheme", Cryptologia, 2020)
Malik Zia Ullah Bashir, Rashid Ali. "Cryptanalysis and improvement of a blind multi-document signcryption scheme", Cryptologia, 2020

12 < 1% match (student papers from 13-Feb-2023)
Submitted to Indian Institute of Technology, Madras on 2023-02-13

13 < 1% match (Internet from 14-Nov-2018)
https://es.scribd.com/document/197524076/A-Pairing-Free-Identity-Based-Tripartite-Signcryption-Scheme

14 < 1% match (Internet from 11-Apr-2023)
https://archive.org/stream/cissp-all-in-one-exam-guide-8th-edition/CISSP%20All-in-One%20Exam%20Guide%2C%208th%20Edition_djvu.txt

15 < 1% match (Internet from 03-Jul-2018)