CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD


# Efficient and Secure Certificateless Aggregate Signcryption Scheme Based on Hyperelliptic Curve

by

Attiya Karim

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

in the

Faculty of Computing
Department of Mathematics

2025

*To my parents, teachers and friends for their support and love.*


# CERTIFICATE OF APPROVAL

## Efficient and Secure Certificateless Aggregate Signcryption Scheme Based on Hyperelliptic Curve

by

Attiya Karim

(MMT223006)

### THESIS EXAMINING COMMITTEE

| S. No. | Examiner | Name | Organization |
|---|---|---|---|
| (a) | External Examiner | Dr. Ghulam Murtaza | NUML, Islamabad |
| (b) | Internal Examiner | Dr. Abdul Rehman Kashif | CUST, Islamabad |
| (c) | Supervisor | Dr. Rashid Ali | CUST, Islamabad |

Dr. Rashid Ali
Thesis Supervisor
May, 2025

Dr. Muhammad Sagheer
Head
Dept. of Mathematics
May, 2025

Dr. Muhammad Abdul Qadir
Dean
Faculty of Computing
May, 2025

# *Author's Declaration*

I, **Attiya Karim** hereby state that my MS thesis titled "**Efficient and Secure Certificateless Aggregate Signcryption Scheme Based on Hyperelliptic Curve**" is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/abroad.

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

**(Attiya Karim)**

Registration No: MMT223006

# *Plagiarism Undertaking*

I solemnly declare that research work presented in this thesis titled "**Efficient and Secure Certificateless Aggregate Signcryption Scheme Based on Hyperelliptic Curve**" 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 MS Degree, the University reserves the right to withdraw/revoke my MS 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.

**(Attiya Karim)**

Registration No: MMT223006

# *Acknowledgements*

All praise be to Almighty ALLAH who has been bestowed me with his great bounties, gifted me a loving family and excellent teachers and enabled me to complete my dissertation.

I would like to express my special gratitude to my kind supervisor Dr. Rashid Ali for his constant motivation. He was always there whenever I found any problem. I really appreciate his efforts and guidance throughout my thesis and proud to be a student of such kind supervisor.

I am deeply grateful to all the teachers at CUST Islamabad, Dr. Muhammad Sagheer, Dr. Abdul Rehman Kashif, Dr. Sabeel, Dr. Muhammad Afzal, Dr. Dur-e-Shehwar and Dr. Samina Batul for conveying the excellent lectures.

I am grateful to the management staff of Capital University of Science and Technology, Islamabad for providing a friendly environment for studies.

I am thankful to all of my family members for becoming a bulwark of patience and support during my research work. However, my deepest gratitude is reserved for my Parents for their earnest prayers, unconditional love and unflinching support in completing my degree program. They supported and encouraged me throughout my life.

Finally, I am obliged to all people who pray for me, share their knowledge during my degree program and support me.

**(Attiya Karim)**

Registration No: MMT223006

# *Abstract*

By merging certificateless aggregate signcryption scheme and elliptic curve cryptography, a certificateless elliptic curve aggregate signcryption scheme is reviewed and presented in this thesis. In the context of 5G wireless networks, it provides confidentiality and authenticity of transmission of data. Unlike traditional encryption schemes, it provides robust security and simultaneously enables an efficient aggregation of multiple encrypted messages. To lower the computational cost and communication overhead, a signcryption scheme is used that combine digital signature and encryption in a single logical step. In this thesis, an efficient certificateless aggregate signcryption scheme based on hyperelliptic curve has been introduced. Hyperelliptic curves are suitable for cryptographic purposes because of smaller key size and fast communication, thus best to use in IoTs devices. Instead of using 160-bit ECC for security and performance, the extended approach make use of 80-bit HEC. It provides non-repudiation, confidentiality, mutual authentication, integrity and resilience against various security threats. Security of the extended scheme relies on the hardness of hyperelliptic curve discrete logarithm problem.

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| **AES** | Advance Encryption Standard |
| **AS** | Aggregate Signature |
| **DES** | Data Encryption Standard |
| **DLP** | Discrete Logarithm Problem |
| **ECC** | Elliptic Curve Cryptography |
| **ECDH** | Elliptic Curve Diffie-Hellman |
| **HECC** | Hyperelliptic Curve Cryptography |
| **HECDLP** | Hyperelliptic Curve Discrete Logarithm Problem |
| **IFP** | Integer Factorization Problem |
| **IoT-CS** | Internet of Things Certificateless Signcryption |
| **RSA** | Rivet Shamir Adleman |

# Symbols

| | |
|---|---|
| $c$ | Ciphertext Message |
| $m$ | Plaintext Message |
| $D$ | Divisor of Hyperelliptic Curve |
| $\gcd(a, b)$ | Greatest Common Divisor of $a$ and $b$ |
| $E$ | Elliptic Curve |
| $H$ | Hyperelliptic Curve |
| mod | Modular Operator |
| $\alpha$ | Order of Base Point G |
| $\mathcal{O}$ | Point at infinity |
| $p$ | Large Prime Number |
| $\mathbb{F}_p$ | Finite Field |
| $Z$ | Set of Integers |

# Chapter 1

# Introduction

From ancient time till today, the secure data transmission over the public network is a big issue. Throughout history, a variety of strategies have been used to conceal information from spies in order to maintain secure communication during military operations, world wars etc. Many cryptographic techniques were used to prevent the enemy from obtaining crucial military data when kings and generals had conversations with their troops. In today's digital age, the rapid growth of communication networks and electronic transactions has created an essential need for secure data transmission and protection. Cryptography is the science of keeping data safe from unauthorized access.

Egyptians applied cryptography for the first time around 1900 BC [1]. In Egyptian society, it was applied in a variety of ways. To hide the secret messages from those who were unaware of them, Egyptian scribes used hieroglyphic symbols. Hieroglyphic symbols are pictorial symbols written in an extremely complex script. Certain symbols stand for things, ideas, sounds, or activities. These are the most well-known Egyptian scripts that are used to protect military and governmental data. Julius Caesar, the Roman ruler, later invented the cipher, which is called the Caesar cipher because of his name [2]. For the purpose of conveying codes or secret messages, later on many cryptographic ciphers were designed. These comprise the Playfair cipher [3], the mono alphabetical cipher [3], the poly alphabetical cipher [4], the Hill ciphers of various orders, etc.

## 1.1 Cryptology

The art and science of secure communication across unreliable channels is known as cryptology [5]. It is the skill of creating secure systems and breaking them. The creation of secure systems is referred to as "cryptography" while the cracking of such a cryptosystem is referred to as "cryptanalysis".

## 1.2 Cryptography

Cryptography is the process of secret communication in the presence of third party. The word "cryptography" comes from the Greek words "kryptos" and "graphein", kryptos means hidden and graphein means writing. The primary goal of early cryptography was to secure the content of messages while they were being transported from one location to another by turning them into an unintelligible series of numbers. From simple message confidentiality to message integrity, sender/receiver identity authentication and digital signatures, cryptography has advanced in the modern era. It includes encrypting and decrypting textual data. Mathematical functions are used for encryption and decryption. In encryption plaintext (original message) is converted into ciphertext (coded message) whereas, in decryption ciphertext is converted back into the corresponding plaintext.

Cryptography is further divided into two main branches, the symmetric key cryptography and asymmetric key cryptography. In symmetric key cryptography, both the sender and receiver use a single key for encryption and decryption. This single key is only accessible by the sender and receiver. The main drawback of symmetric key cryptography is key distribution. To resolve this problem, in 1976 Diffie and Hellman [6] proposed a public key cryptography or asymmetric key cryptography. In public key cryptography, both the sender and receiver use two different keys, one is known as public key that is known to everybody and the other is known as private key that is kept secret. RSA [7], El-Gamal [8] and ECC [9] (elliptic curve cryptography) are the public key cryptosystems. Public key cryptography

involves encryption and digital signature that ensure the confidentiality and authentication of communication. Public key cryptography operations are complex as compared to private key cryptography but it provides highest level of security as compared to others. Public key cryptography has great advantages over the private key cryptosystem. Public key cryptosystem publish a directory of public keys so that when somebody wants to send a message then he simply look the public key of specific user and use the agreed algorithm for message sending. Then the authentic receiver has the decrypting key and able to read the content of original message.

Modern cryptography [10] provides security features such as confidentiality, authentication, integrity, non-repudiation, unforgeability, forward secrecy of messages and public verification. Now a days, the modern science of cryptography is not so limited, it has been used in many ways like random numbers, electronic signature, secure key exchange, electronic voting, electronic money and many more.

## 1.3 Cryptanalysis

By examining and taking advantage of flaws in cryptographic methods, a process known as cryptanalysis can be used to study and decrypt ciphers, codes and encrypted material. To put it briefly, cryptanalysis is the process of obtaining the plaintext without decryption key [11].

Cryptanalysis methods are used to evaluate a scheme's strength against known attacks. Additionally, cryptanalysis is carried out to identify the scheme's vulnerabilities in order to break the system. The attacks will be successful if the scheme is weak or open to several attacks. The primary objective of cryptanalysis is to gather as much information as possible about the plaintext in order to break the system, obtain the plaintext's ciphertext and obtain the keys necessary to decrypt more messages that share the same key [11]. The goal of cryptanalysis is to reveal the secrets that are hidden in encrypted data. The categorization of cryptanalysis attacks is based upon the nature of the information that the attacker possesses. Numerous cryptanalysis attacks are employed to assess the weakness of various

schemes. Known plaintext attacks (KPA), chosen plaintext attacks (CPA), chosen ciphertext attacks (CCA), man in the middle attacks (MITM), man at the end attacks (MATE) and brute force attacks are the different categories for these cryptanalysis attacks. For detail on these attacks, the reader is referred to [12].

## 1.4 Signcryption Scheme

In 1997, Zheng [13] proposed a new cryptographic technique called Signcryption. It combines the capabilities of an encryption algorithm with a digital signature for confidentiality and authentication. Therefore, one can concurrently execute the function of both digital signature and encryption in a single logical step. It reduces the computational cost as compared to signature-then-encryption scheme. In the signcryption design set up by Zheng, the sender uses the receiver's public key to make a secret key for symmetric encryption. After the receiver gets the ciphertext and digital signature he uses his private key to make the same secret key.

Over time, a number of signcryption methods [13–19] have been established, each technique has advantages and disadvantages of its own, as well as varying levels of computational costs and security services.

## 1.5 Literature Survey

Although fifth generation (5G) technology is still in its beginnings, there are a number of security concerns with it. The 5G network of the future will be extremely diversified, open and adaptable. Since 5G communication's Internet of Things (IoTs) terminal access method mostly uses wireless access, in large-scale situations it will result in excessive system resource consumption and signal congestion. The IoTs framework makes use of cloud storage, user-facing apps, web-enabled smart devices with embedded technologies and underlying internet structure [20]. The benefits of IoTs have demonstrated potential in a number of industries, including

supply chains, healthcare, banking, agriculture and many more [21]. There are significant issues associated with the growing number of apps and the prediction of more device connections [22]. One of the main issues for everyone is security [23, 24]. It also inherits the security vulnerabilities associated with wireless networks [25]. Therefore, in order to solve the problem of terminal and network authentication, aggregate signcryption scheme (ASC) [26] was introduced. This approach eliminates the need for a reliable third party by combining the signcrypted ciphertexts produced by various terminals into a single ciphertext. Once the aggregation scheme is adopted, numerous IoT devices have access to certified communication in a 5G environment. A short signature is created by aggregating the various signatures of $\eta$ signers in some fashion, and aggregate signatures (AS) [27] can guarantee non-repudiation for distinct messages. AS lowers the processing costs and storage space needed for the verification process. Across 5G environments, cloud computing, wireless networks, IoTs, and electronic medical devices, AS has many uses. By incorporating aggregate signatures in signcryption scheme, ASC scheme can be established [28–32].

The certificateless ASC (CL-ASC) scheme was proposed by Lu and Xie [33] in 2011. For wireless sensor networks, Yu and Ren [34] developed a new certificateless elliptic curve aggregate signcryption scheme. Later, Zhang et al. [35] developed an effective and publicly verifiable CL-ASC scheme. Eslami and Pakniat [36] developed a CL-ASC scheme that is provably secure through pairing work. With inside security, Zhang et al. [37] developed an effective CL-ASC method. IoTs can use the CL-ASC framework described in [38]. To create a reliable data transmission model, Elhoseny and Shankar [39] combine aggregation with signcryption approach.

There are several benefits to the ECC [9, 40–43], including high security, low storage space needs and in-expensive bandwidth needs. ECC has several benefits, including 5G communication, digital economies, smart medical care, embedded systems and intelligent card systems. Most CL-ASC systems have been designed so far using bilinear pairings. Therefore, developing a secure and effective ECC based CL-ASC method is a crucial issue. As a result, IoTs must take into account

the cost and effectiveness of communication secrecy and authentication. In [34], the certificateless elliptic curve aggregate signcryption technique (CL-ECASC) for IoTs is suggested to address the aforementioned issues and achieve data secrecy while enhancing authentication efficiency. Because CL-ECASC scheme has a lower communication cost and a faster calculation efficiency, it can be used for secure information transmission.

In 1989, Koblitz [44] proposed that encryption systems based on the discrete logarithm problem (DLP) be designed utilizing hyperelliptic curves rather than simple elliptic curves. Extended versions of elliptic curves are called hyperelliptic curves. Hyperelliptic curves [45] have one major advantage, that is, their key size is small. In comparison to an elliptic curve, a hyperelliptic curve requires a smaller finite field to get a certain security level. Hyperelliptic curves of genus 2 are presented in Chapter 2.

## 1.6 Thesis Contribution

In this thesis, "Certificateless elliptic curve aggregate signcryption scheme (CL-ECASC)" proposed by Yu and Ren [34] is reviewed. They proposed the scheme by merging ECC and CL-ECASC scheme [23, 46, 47] in the context of 5G wireless networks. For the transmission of secure and authenticated messages, the scheme is based on ECC, which offers both digital signature and encryption with low computational cost in comparison to signature-then-encryption schemes. This method offers strong security because its security depends on the elliptic curve discrete logarithm problem (ECDLP) and elliptic curve Diffie-Hellman problem (ECDHP). Compared to other cryptosystems, this scheme offers integrity, forward secrecy, message confidentiality, authenticity, unforgeability, verification and non-repudiation. It offers quicker processing and shorter key lengths.

The reviewed scheme is further extended using hyperelliptic curves. It enhance the security of existing scheme, making it more resistant to cryptographic attacks. Using hyperelliptic curve in extended scheme, provides the same security level as elliptic curves but with smaller key size which is more faster and provides low

computational cost. Security of the extended scheme relies on the hardness of hyperelliptic curve discrete logarithm problem.

## 1.7 Thesis Layout

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

**Chapter 2**, presents mathematical background that is related to our scheme, including basic definitions of cryptography, digital signatures and signcryption. Additionally, the section includes various cryptanalysis attacks that are used in our scheme.

**Chapter 3**, covers a detailed analysis of elliptic curve aggregate signcryption scheme of Yu and Ren [34] based on certificateless elliptic curve cryptography. A comprehensive analysis about security of the scheme is also presented.

**Chapter 4**, presents aggregate signcryption scheme which is based on certificateless hyperelliptic curve cryptography. The working of the proposed scheme is covered in detail.

**Chapter 5**, covers the security analysis of proposed scheme and comparison with other schemes. Also the comparison of communication analysis and computational cost analysis with other schemes is provided.

**Chapter 6**, provides the overall conclusion.

# Chapter 2

# Preliminaries

Some fundamental definitions from the field of cryptography and related mathematics are presented in this chapter. For the benefit of the reader, a few basic definitions, notations, algebraic and cryptographic results are further included. This chapter also covers several hard cryptographic problems and the mathematical framework.

## 2.1 Mathematical Background

### Definition 2.1.1. Group

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

1. **Closure Law:** $a * b \in G$ for all $a, b \in G$
2. **Associativity:** $(a * b) * c = a * (b * c)$ for all $a, b, c \in G$
3. **Identity element:** There is an identity element $e \in G$ such that

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

for all $a \in G$

4. **Inverse element:** For all $a \in G$, there exists an element $a' \in G$ such that

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

Then $a'$ is called inverse of $a$

A group $G$ is called **abelian** if it satisfies

$$a * b = b * a$$

for all $a, b \in G$ [48]."

### Definition 2.1.2. Field

"A non-empty set $\mathbb{F}$ together with two binary operations '$+$' and '$*$' defined on $\mathbb{F}$ is said to be field if it satisfies following axioms:

1. $\mathbb{F}$ is an abelian group under addition.
2. $\mathbb{F}\backslash\{0\}$ is an abelian group under multiplication.
3. Left and right distributive laws are hold in $\mathbb{F}$ [49]."

### Definition 2.1.3. Extension Field

If $\mathbb{F}_p$ and $\mathbb{F}_q$ are two fields and $\mathbb{F}_p \subseteq \mathbb{F}_q$, $\mathbb{F}_p$ is a subfield of $\mathbb{F}_q$, then we say that $\mathbb{F}_q$ is the extension field of $\mathbb{F}_p$.

**Example:** The set of complex numbers $\mathbb{C}$ is the extension field of real numbers $\mathbb{R}$ and real numbers is an extension field of rational field $\mathbb{Q}$.

### Definition 2.1.4. Algebraic over the field

If $\mathbb{F}_q$ is an extension field of $\mathbb{F}_p$, an element $\alpha \in \mathbb{F}_q$ is called algebraic over $\mathbb{F}_p$ if there is a non zero polynomial $f$ such that $f(\alpha) = 0$, where $f \in \mathbb{F}_p$.

### Definition 2.1.5. Algebraically closed

A field $\mathbb{F}_p$ is algebraically closed if every polynomial with coefficient in $\mathbb{F}_p$ has roots in $\mathbb{F}_p$. For example, the field $\mathbb{C}$ is algebraically closed.

**Definition 2.1.6. Galois Field**

"For every prime $p$ and a positive integer $n$, there is exactly one finite field of order $p^n$. This field $GF(p^n)$ usually referred as Galois field of order $p^n$ [50]."

$\mathbb{F}_2$, $\mathbb{F}_3$, $\mathbb{F}_5$ are all finite fields and their Galois fields are $\mathbb{F}_{2^3}$, $\mathbb{F}_{3^2}$, $\mathbb{F}_{5^2}$ respectively.

### 2.1.1 One way Function

A function that is easy to compute in one direction, but computationally infeasible to invert in the other direction. More specifically, for given $x$ it is easy to compute $f(x)$ but for given $f(x)$, it is highly challenging to get $x$. An essential component of cryptography is one way function.

### 2.1.2 One way Trapdoor Function

A one way trapdoor function $f$ is a function that satisfies the following properties.

1. Computing $f(x)$ is simple for given $x$ in the domain of $f$.
2. Unless particular information is given, it is computationally impossible to determine $x$ such that $f(x) = y$, for given $y$ in the range of $f$.

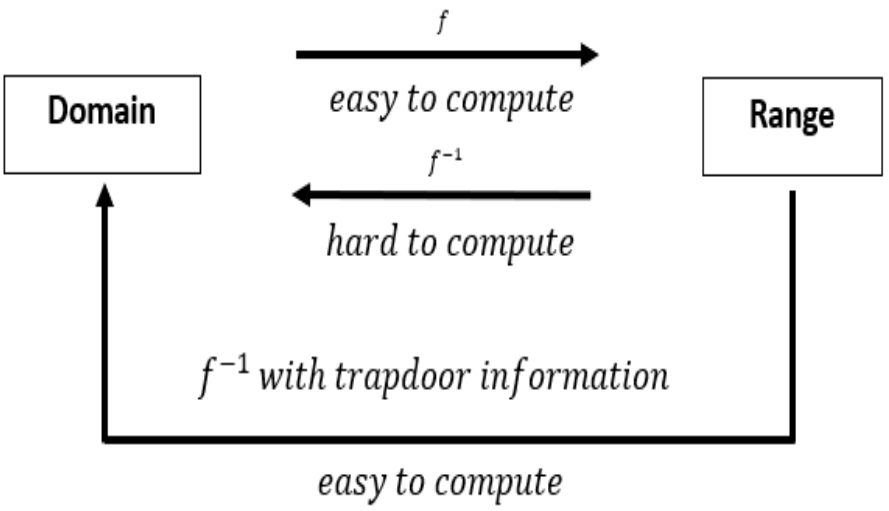


FIGURE 2.1: One Way Trapdoor Function

### 2.1.3 Integer Factorization Problem(IFP)

"An **integer factorization problem** is defined as, let $m$ be a given number and $m \in \mathbb{Z}$, the problem of decomposition of $m$ into the product of primes $p_a$ and $q_b$ such that $m = p_a q_b$ [51]."

### 2.1.4 Discrete Logarithm Problem (DLP)

"Given $x, y \in \mathbb{Z}$ such that

$$x^n = y \mod p$$

then finding $n$ is known as **integer factorization problem** [51]."

## 2.2 Cryptology

The word "cryptology" originates from the Greek terms kryptos and logos. Kryptos means hidden, while logos refers to words. The science that studies encrypted and secret communications is called cryptology. There are two main branches of cryptology:

1. Cryptography
2. Cryptanalysis

### 2.2.1 Cryptography

The science of private communication through a public network is called cryptography. It is the science of encrypting and decrypting messages using mathematical functions. Plaintext or cleartext is the term used in cryptography to describe the initial message. The term "encryption" refers to the process of hiding contents of plaintext message from others. Decryption is the procedure that turns the ciphertext back into plaintext. A cryptosystem consists of plaintext, ciphertext,

encryption algorithms, decryption algorithms and the key $k$, where the key is an entity utilized by the communicating parties for the encryption and decryption processes. Written communications had previously been made private through the application of cryptography. It was just a set of encryption mechanism. It had the same principles to apply equally for securing data flow between computers or to encrypting television signals. The applications of the modern (mathematical) science of cryptography are numerous and include random numbers, data integrity, electronic signatures, electronic voting, electronic money, secure key exchange and many more.

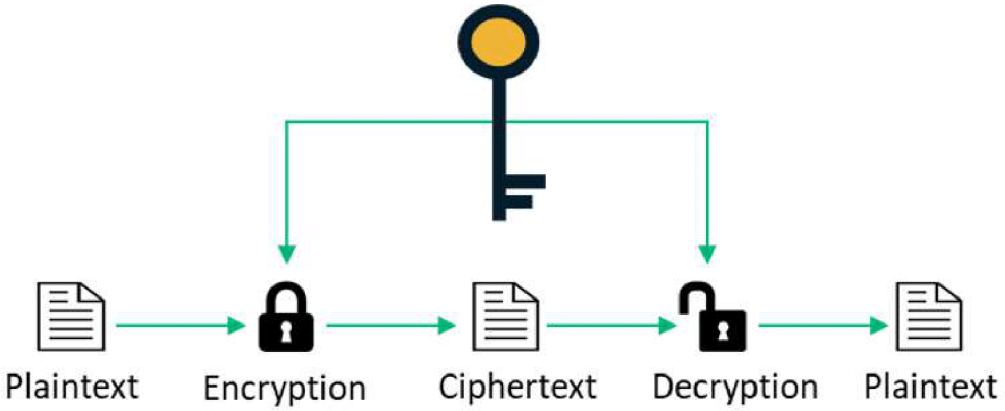


FIGURE 2.2: A Typical Cryptosystem

### 2.2.2 Security Attributes of Cryptography

The following security parameters are addressed by modern cryptography.

1. **Confidentiality:** It is the process of keeping information confidential so that only the person who is supposed to get it can understand it. It refers to securing the data against unauthorized or uninvited parties. Many important pieces of information have become available in recent years, including trade secrets, credit card numbers, bank account statements, government and military paperwork, and many more from many aspects of daily life. Everybody wants to protect their personal information so that secrets are kept private and unknown to other parties.

2. **Authentication:** It is the process of providing the recipient with confirmation of the sender's identification so they may be guaranteed the communication was sent by the actual recipient.

3. **Integrity:** It is a technique to guarantee that data is not read or altered while being transmitted or stored.

4. **Non-repudiation:** Nobody can claim not to be the sender of information. It is a way to make sure that data cannot be rejected.

5. **Unforgeability:** The signature verification algorithm makes it computationally impossible for an attacker to produce a signature.

6. **Forward secrecy:** When someone discovers the sender's private key but the ciphertext cannot be decrypted by anyone. By employing random numbers in the ciphers, this feature can typically be preserved.

7. **Public Verification:** Any third party can confirm the signature without knowing the sender's or recipient's secret key. Therefore, to verify the signature anyone can use the public key without having to read the contents of the original message.

## 2.3 Types of Cryptography

Two main areas of cryptography are:

- Symmetric Key Cryptography
- Asymmetric Key Cryptography

### 2.3.1 Symmetric (Private) Key Cryptography

In symmetric key cryptography (which is also known as private key cryptography), for encryption and decryption during transmission of a message both the sender and the receiver use the same key known as secret key or private key. It is quick

and very simple [52]. Advanced Encryption Standard (AES) [52], Data Encryption Standard (DES), Blowfish [53], Double Data Encryption Standard (2DES) [48], and Triple Data Encryption Standard (3DES) [48] are few examples of private key cryptography. Key sharing in symmetric key cryptography, which requires that the private key be sent to every communication partner, is the main drawback.

Symbolically, the encryption and decryption in symmetric key cryptography is expressed as

$$c = E_k(m)$$

and

$$m = D_k(c)$$

where $m$, $c$ and $k$ are plaintext, ciphertext and private key respectively.

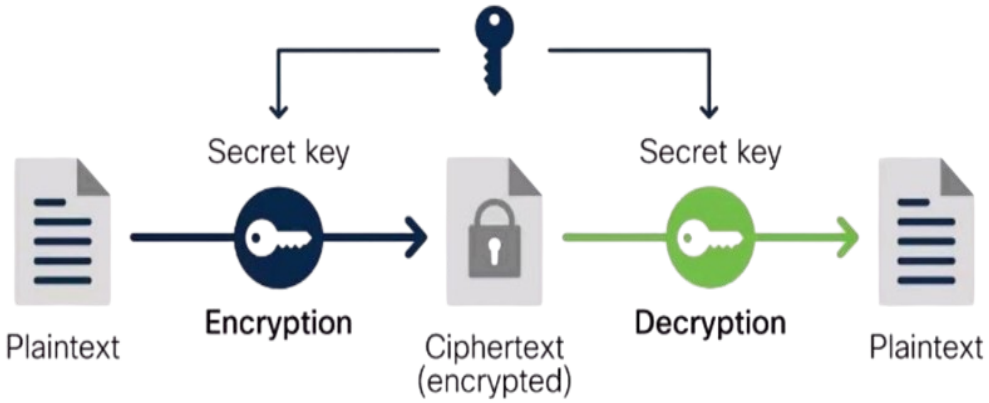


FIGURE 2.3: Symmetric Key Cryptography

### 2.3.2 Asymmetric (Public) Key Cryptography

In 1976, Diffie and Hellman developed public key cryptography [54] by introducing the idea of using two keys, the public key and the private key. When sending a plaintext, the sender encrypts it with the recipient's public key. The recipient then decrypt the ciphertext using his/her private key to recover the plaintext. The public key is known to all and the private key is kept secret, but they are mathematically related. The idea is based on one way trapdoor function. That

is, the process of generating the public and private keys is quite simple, however from the public key it is challenging to determine the private key.

Symbolically, the encryption and decryption in asymmetric key cryptography is expressed as

$$c = E_{k_1}(m)$$

and

$$m = D_{k_2}(c)$$

where $m$, $c$, $k_1$ and $k_2$ are plaintext, ciphertext, public key and private key respectively.

El-Gamal [8], ECC [9] and RSA [55] are few examples of public key cryptosystems. Encryption and digital signatures are used in public key cryptography to guarantee communication's confidentiality and authenticity. While the processes of public key cryptography are more complicated than those of private key cryptography. A directory of public keys is published by public key cryptosystems, allowing users to send messages by simply looking up the public key of a particular user and using the pre-agreed messaging method. The original message's content can then be read by the authentic receiver who also has the decrypting key.

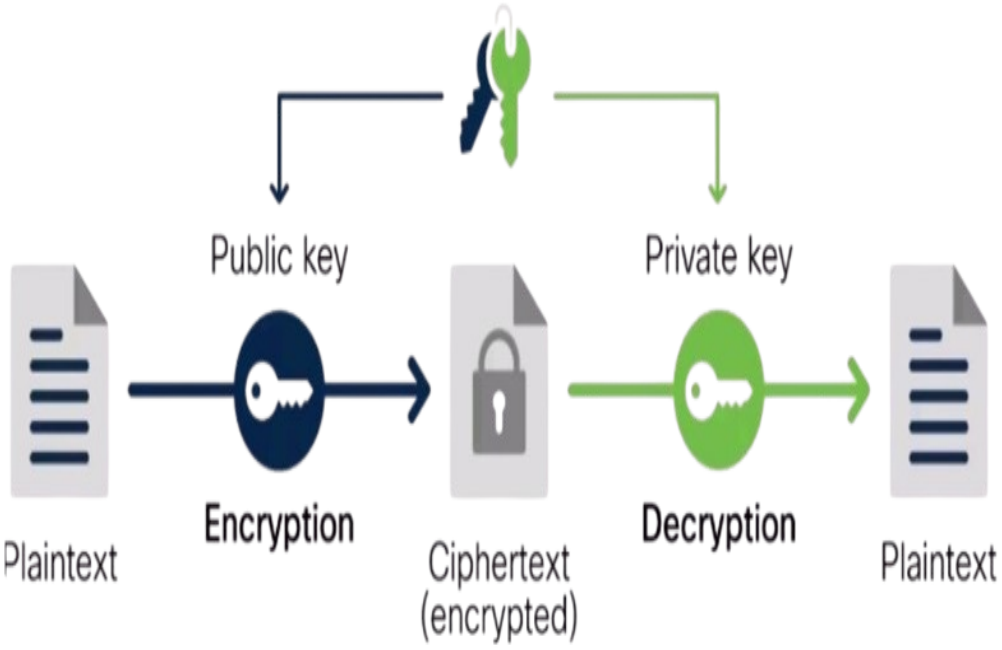


FIGURE 2.4: Asymmetric Key Cryptography

## 2.4 Cryptanalysis

The study of techniques for deciphering encrypted data without possessing the secret key is known as cryptanalysis. There are numerous methods for carrying out cryptanalysis. An attack is the term for any such attempt. These include the widely used methods of differential cryptanalysis, linear cryptanalysis, ciphertext only analysis, man-in-the-middle attack and known plaintext analysis [56]. The original plaintext and ciphertext are represented by $m$ and $c$ respectively in the figure below. After observing the pattern of $m$ and $c$, the cryptanalyst analyzes it and finds another pair, $m^{'}$ and $c^{'}$.

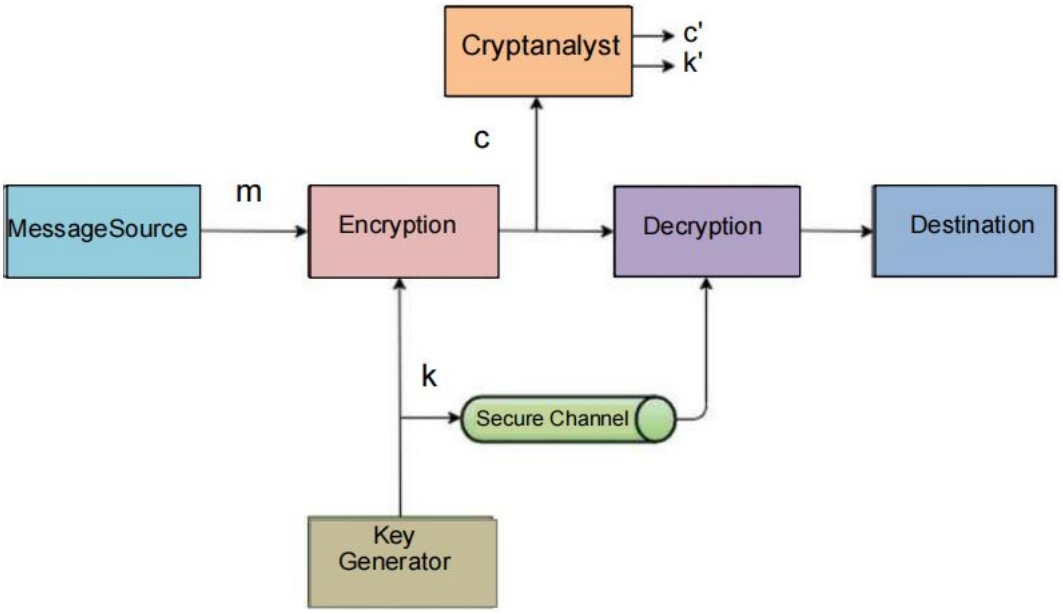


FIGURE 2.5: Cryptanalysis Model

### 2.4.1 Cryptographic Attacks

Following are some common attacks on cryptographic system.

#### 2.4.1.1 Chosen Ciphertext Attack:

Cryptanalysts use the chosen ciphertext attack to select a ciphertext and then look for the matching plaintext. Public key cryptosystems are typically vulnerable to this kind of attack.

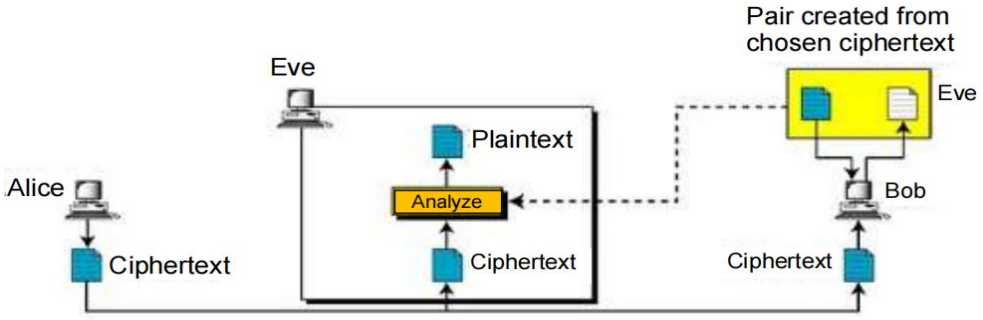


FIGURE 2.6: Chosen Ciphertext Attack

#### 2.4.1.2 Chosen Plaintext Attack:

An attack known as "chosen plaintext" occurs when an attacker selects any plaintext and then obtains the corresponding ciphertext. The aim of this attack is to obtain more data in order to crack the cryptosystem.

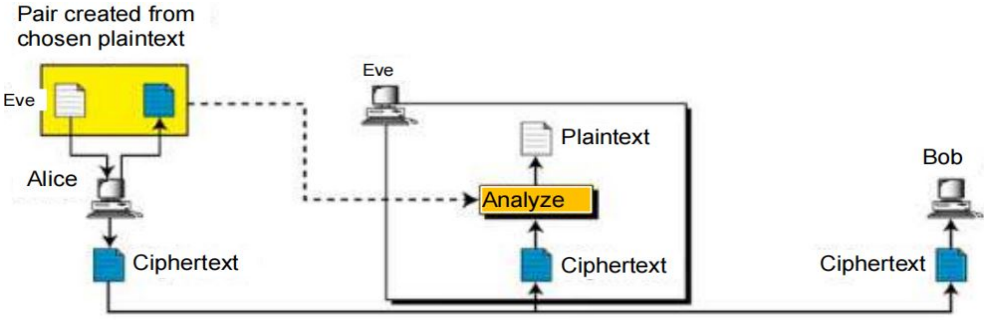


FIGURE 2.7: Chosen Plaintext Attack

#### 2.4.1.3 Known Plaintext Attack:

A known plaintext attack is one in which the attacker has both the plaintext and the corresponding ciphertext. By getting more information from the pair, the attacker tries to break the cryptosystem.

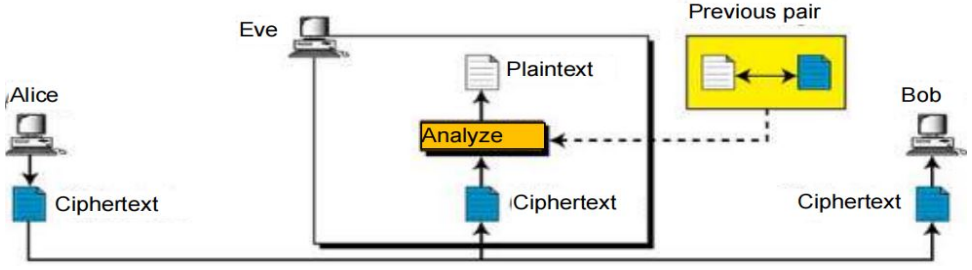


FIGURE 2.8: Known Plaintext Attack

#### 2.4.1.4 Ciphertext Only Attack:

The attacker can only obtain further cryptosystem knowledge by utilizing the ciphertext in this attack. In practical cryptography problems, this is the most popular attack.

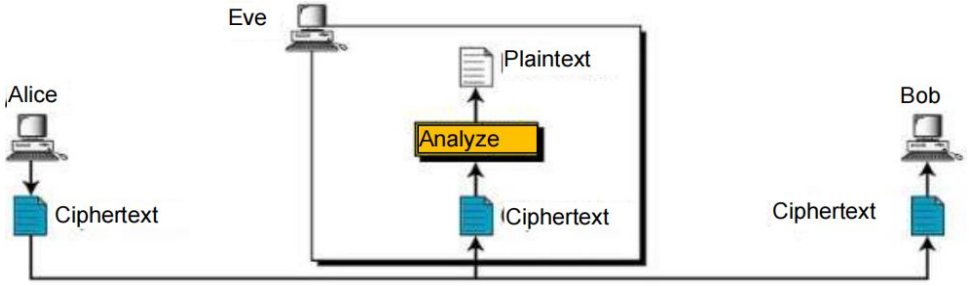


FIGURE 2.9: Ciphertext Only Attack

#### 2.4.1.5 Man in the Middle Attack:

This attack involves the attacker interfering with the sender and receiver's communications in order to obtain information from the received data. Data will be sent and received by the attacker without either party knowing about it. To carry out this kind of attack, the attacker first chooses two fake keys, after which he uses his single key to initiate communication with the first participant. The attacker successfully establishes a link with the first person. Similarly, a connection is successfully established with the second participant. The attacker then sends a message of their choice to both participants. Both participants think they are communicating with one another.

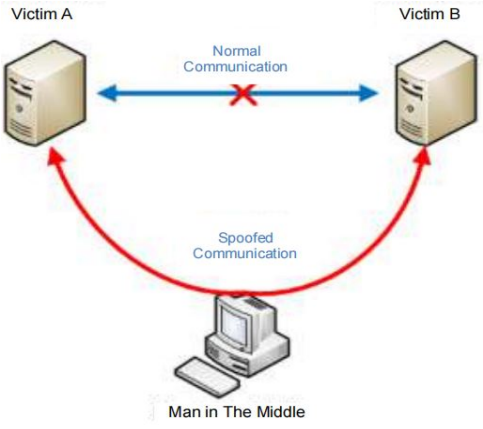


FIGURE 2.10: Man in the Middle Attack

## 2.5 Elliptic Curve Cryptography

Elliptic curves serve as the foundation for elliptic curve cryptography (ECC). Miller and Koblitz [9] used the elliptic curve group over the finite field in 1985. The public key cryptosystem was primarily developed by constructing elliptic curves over finite fields algebraically [57].

An elliptic curve $E(a, b)$ is a cubic curve made up of the points $(a, b)$ that satisfy the following equation. Another name for this type of equation is a Weierstrass equation [58].

$$y^2 = x^3 + ax + b \text{ where } a, b \in \mathbb{F} \tag{2.1}$$

The field can be a real field or finite field. If $\mathbb{F} = \mathbb{R}$, then the curve is called real elliptic curve. Real numbers are utilized to better comprehend how the visual representation of the curve works and how the geometry of their points functions. An elliptic curve over $\mathbb{R}$ is defined by (2.1) where $a$ and $b$ are real numbers.
The elliptic curve needs to satisfy the following equation and require non-singularity from a cryptography perspective.

$$4a^3 + 27b^2 \neq 0 \tag{2.2}$$

Consider

$$y^2 = x^3 + x + 1 \mod 19 \tag{2.3}$$

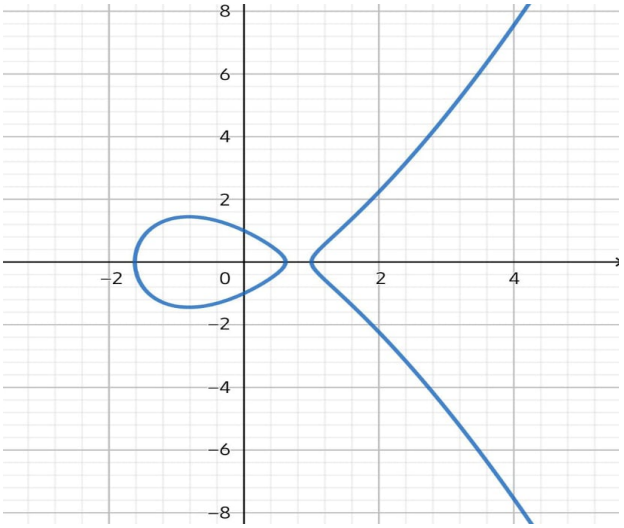


FIGURE 2.11: Elliptic Curve Over $\mathbb{R}$ : $y^2 = x^3 - 2x + 1$

In the above equation , $a = 1$, $b = 1$ and $p = 19$. We first verify that:

$$\begin{aligned} 4a^3 + 27b^2 \mod p &= 4(1)^3 + 27(1)^2 \mod 19 \\ &= 4 + 27 \times 1 \mod 19 \\ &= 31 \mod 19 \\ &= 12 \mod 19 \neq 0. \end{aligned}$$

From reduced set of residues, we will find the quadratic residues $\mathbb{Q}_{19}$.

$\mathbb{Z}_{19} = \{0, 1, 2, ..., 18\}$

TABLE 2.1: Quadratic Residues in $\mathbb{Q}_{19}$

| $x^2 \mod p$ | $(p-x)^2 \mod p$ |
|---|---|
| $1^2 \mod 19$ | $18^2 \mod 19 = 1$ |
| $2^2 \mod 19$ | $17^2 \mod 19 = 4$ |
| $3^2 \mod 19$ | $16^2 \mod 19 = 9$ |
| $4^2 \mod 19$ | $15^2 \mod 19 = 16$ |
| $5^2 \mod 19$ | $14^2 \mod 19 = 6$ |
| $6^2 \mod 19$ | $13^2 \mod 19 = 17$ |
| $7^2 \mod 19$ | $12^2 \mod 19 = 11$ |
| $8^2 \mod 19$ | $11^2 \mod 19 = 7$ |
| $9^2 \mod 19$ | $10^2 \mod 19 = 5$ |

Therefore set of quadratic residues $\{\frac{p-1}{2} = 9\}$ is

$\mathbb{Q}_{19} = \{1, 4, 5, 6, 7, 9, 11, 16, 17\}$.

We will compute $y^2 = x^3 + x + 1 \mod 19$ for $0 \leq x < 19$ to find the points on elliptic curve and check whether $y^2$ lies in quadratic residues set.

TABLE 2.2: Points of Elliptic Curve

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| $y^2$ | 1 | 3 | 11 | 12 | 12 | 17 | 14 | 9 | 8 | 17 |
| $y^2 \in Q_{19}$? | Yes | No | Yes | No | No | Yes | No | Yes | No | Yes |
| $y_1$ | 1 | - | 7 | - | - | 6 | - | 3 | - | 6 |
| $y_2$ | 18 | - | 12 | - | - | 13 | - | 16 | - | 13 |

| $x$ | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|
| $y^2$ | 4 | 13 | 12 | 7 | 4 | 9 | 9 | 10 | 18 |
| $y^2 \in Q_{19}$? | Yes | No | No | Yes | Yes | Yes | Yes | No | No |
| $y_1$ | 2 | - | - | 8 | 2 | 3 | 3 | - | - |
| $y_2$ | 17 | - | - | 11 | 17 | 16 | 16 | 18 | - |

The points of elliptic curve $E_{19}(1,1)$ are as follows:

$$\begin{aligned} E_{19}(1,1) \;=\; \{&(0,1),\,(0,18),\,(2,7),\,(2,12),\,(5,6),\,(5,13),\,(7,3),\,(7,16),\\ &(9,6)\,(9,13),\,(10,2),\,(10,17),\,(13,8),\,(13,11),\,(14,2),\\ &(14,17)\,(15,3),\,(15,16),\,(16,3),\,(16,16),\,\mathcal{O}\}\,. \end{aligned}$$

ECC is a public key cryptography. Compared to RSA and Elgamal, ECC has smaller key size and achieve the same level of security as other cryptosystems. In contrast to 1024 bit RSA, it employs a 160 bit key size [59]. ECC reduces the computational cost of the cryptosystem while maintaining its security. It is faster than RSA because it requires less storage and has a smaller key size [60].

TABLE 2.3: Comparison of Standard Key Size in bits

| **Security Bit Level** | **RSA** | **ECC** |
|---|---|---|
| 80 | 1024 | 160 |
| 112 | 2048 | 224 |
| 128 | 3072 | 256 |
| 192 | 7680 | 384 |
| 256 | 15360 | 512 |

### 2.5.1 Computations on Elliptic Curves

Elliptic curves over prime numbers are used in cryptography. For the purpose of the identity element, a unique point $\mathcal{O}$ is added to the elliptic curve. The elliptic

curve group definition can therefore include identity element $\mathcal{O}$ (not lies on the curve). Elliptic curve is symmetric with respect to the $x$-axis. So inverse of point $P = (x, y)$ is $-P = (x, P - y)$ [9].

#### 2.5.1.1 Point Addition

The generation of group points on elliptic curve is based upon point addition. Let $P(X_P, Y_P)$ and $Q(X_Q, Y_Q)$ be points on the elliptic curve. Then addition of $P$ and $Q$ is calculated in following steps.

1. Draw the straight line passing through the points $P$ and $Q$ (secant line) if $P$ and $Q$ are different points. If $P$ and $Q$ are same points then draw the tangent line.
2. Find the third point of intersection between the line and the elliptic curve, and then reflect that point across the $x$-axis.

The addition [9] of elliptic curve points is $P + Q = (X_3, Y_3)$, where

$$\begin{aligned} X_3 &= m^2 - X_P - X_Q \mod p, \\ Y_3 &= m(X_P - X_3) - Y_P \mod p, \\ m &= \frac{Y_Q - Y_P}{X_Q - X_P}. \end{aligned}$$

#### 2.5.1.2 Point Doubling

If $P$ and $Q$ are same points then the addition is calculated as

1. Draw the tangent line passing through the point $P$.
2. Find the second point of intersection between the drawn line and the elliptic curve, and then reflect that point across the x-axis.

The formula for point doubling on elliptic curve

$$X_3 = m^2 - 2X_P \mod p,$$

$$Y_3 = m(X_P - X_3) - Y_P \bmod p,$$

$$m = \frac{3X_P^2 + a}{2Y_P}.$$

**Example: Points on Elliptic Curve**

Consider an elliptic curve $E : y^2 = x^3+2x+3 \bmod 17$ using base point $P = (5, 11)$. This elliptic curve has 22 points that are as follows:

$$2P = 2(5, 11) = (5, 11) + (5, 11) = (15, 5).$$

Similarly,

$3P = (13, 4)$, $10P = (14, 2)$, $17P = (2, 7)$,

$4P = (8, 15)$, $11P = (16, 0)$, $18P = (8, 2)$,

$5P = (2, 10)$, $12P = (14, 15)$, $19P = (13, 13)$,

$6P = (12, 15)$, $13P = (11, 9)$, $20P = (15, 12)$,

$7P = (9, 11)$, $14P = (3, 11)$, $21P = (5, 6)$,

$8P = (3, 6)$, $15P = (9, 6)$, $22P = \mathcal{O}$

$9P = (11, 8)$, $16P = (12, 2)$,

So, $P = (5, 11)$ is the generator of cyclic group.

## 2.6 Elliptic Curve Diffie-Hellman Key Exchange

Elliptic Curve Diffie-Hellman (ECDH) is a variation of Diffie-Hellman that enables two parties to establish a shared secret key via an unsecured channel without any prior knowledge of one another. It is utilized to protect a range of Internet applications and was among the initial public key protocols.

Two parties, $A$ and $B$, can establish a shared secret key via an unsecured channel using the anonymous key agreement technique known as ECDH, provided that

each party has an elliptic curve public-private key pair. The protocol is described below for the production of the same secret key $k$ [54].

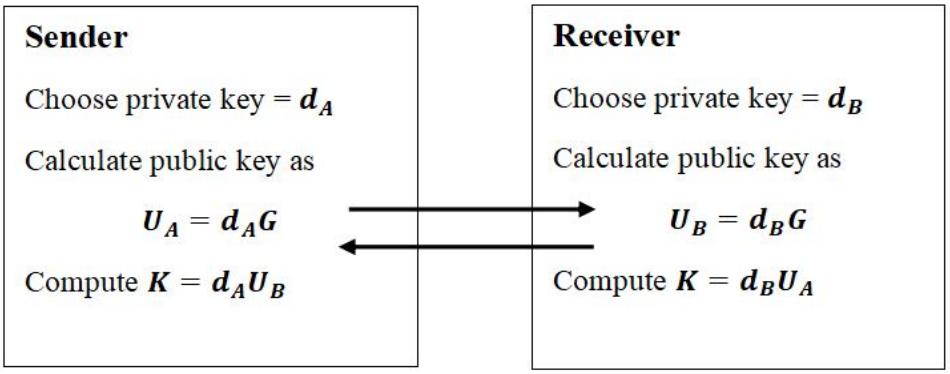


FIGURE 2.12: Diffie-Hellman Key Exchange

For implementing ECC, the main step is converting a message into the points of an elliptic curve $E_p(a, b)$. Converting plaintext message into points of an elliptic curve involves the following steps:

1. Select a curve which has N points.
2. Our alphabets consist of $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, ..., Z\}$.
3. The above step will encode our message into numbers between $\{0, 1, 2, ..., 35\}$.
4. Now select a number $k$ which must be shared between both the parties.
5. For each number $mk$, calculate $x = mk + 1$ and find the corresponding value of $y$.
6. If we cannot find any value of $y$ then try for $x = mk + 2$ and then for $x = mk + 3$ uptil $x = mk + (k - 1)$.

For encryption and decryption domain parameters of an elliptic curve $E_p(a, b)$ are required.

### 2.6.1 Domain Parameters

The domain parameters [9] are:

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

### 2.6.2 Key Generation

For the elliptic curve $E_p(a, b)$ with the domain parameters $(p, a, b, G, n')$ the key pair [9] is computed as follows:

1. The sender of a message choose a random integer $z_A$ as his private key such that $z_A < n'$.
2. Public key of sender is then calculated by multiplying base point $G$ with private key $z_A$ as $X_A = z_A G$.
3. The receiver of a message choose a random integer $z_B$ as his private key such that $z_B < n'$.
4. Public key of receiver is then calculated by multiplying the base point $G$ with private key as $X_B = z_B G \bmod p$.

Here $z_A$, $z_B$ are integers and $X_A$, $X_B$ are points of elliptic curve.

### 2.6.3 Key Sharing

1. Sender first chooses a random number $r \leq n'$.
2. Then multiply $r$ with base point $G$ to get $T = rG$.
3. Multiply the random number $r$ with public key of receiver $X_B$ to get $S = rX_B$.
4. Send T to receiver through public channel.

So the receiver will calculate $S$ by using his private key $z_B$ as $S = Tz_B$. So sender and receiver have same secret key $S$ [19].

### 2.6.4 Elliptic Curve Cryptosystem

With the use of the recipient's public key, the elliptic curve cryptography encryption algorithm encrypts the original message. The recipient will then receive this message. For encryption we use

$$c^{'} = [(rG), (m' + rX_B)].$$

The message is decrypted by multiplying the private key $z_B$ of the receiver with first part of ciphertext $c^{'}$.

$$c^{'} = [(z_B(rG), (m^{'} + rX_B)].$$

$$c^{'} = [(X_B r), (m^{'} + rX_B)].$$

Then the second part of ciphertext is subtracted from first part to get original message $m^{'}$ [61].

$$[(m^{'} + rX_B) - (X_B r)] = m^{'}.$$

### 2.6.5 Elliptic Curve Discrete Logarithm Problem (ECDLP)

For given points $A$ and $B$ in elliptic curve group $E_p(a, b)$, elliptic curve discrete logarithm problem is to find the integer $Q$ such that $QA = B$. The number $Q$ (if it exist) is then called discrete logarithm of $B$ to the base $A$.

## 2.7 Hyperelliptic Curves

A hyperelliptic curve $H$ of genus $g$ defined over $\mathbb{F}_q$ is of the form

$$H : y^2 + h(x)y = f(x) \tag{2.4}$$

where $\mathbb{F}_q$ be a finite field and curve has rational points in its algebraic closure $\overline{\mathbb{F}}_q$ where $f(x) \in \mathbb{F}_q$ is a monic polynomial of degree $2g+1$ and $h(x) \in \mathbb{F}_q$ is a polynomial whose degree is at most $g$.

The curve should be non singular i.e no point on $H$ satisfies $2y + h(x) = 0$ and $h^{'}(x)y - f^{'}(x) = 0$.

If $h(x) = 0$ then (2.4) becomes $y^2 = f(x)$ and degree of $f(x)$ is $2g+1$ with the condition that $f(x)$ has no repeated roots.

**Genus of a curve:**

The genus of a curve is the maximum number of non-intersecting, simple closed curves (loops) that can be drawn on the surface without touching each other. In simple words, it means the number of holes a curve has. It decides the polynomial type and the processing time required for implementation.

For secure and efficient implementation, a genus two curve is known to be the best whereas curves of genus greater than 2 are vulnerable to some known attacks like index calculus attack, for more details see [62]. In this thesis we will only discuss curves of genus 2.

In the figure below, the curve that intersects the hyperelliptic curve with genus 2 is known as Jacobian variety curve. This Jacobian variety curve meets the hyperelliptic curve at 6 points $P_1, P_2, Q_1, Q_2, R_1, R_2$. Within the quotient group, the combined sum of these points equals to zero. The points $R_1^{'}$ and $R_2^{'}$ are the reflection of $R_1$ and $R_2$ respectively. The resulting group operation is

$$(P_1 + P_2) + (Q_1 + Q_2) = (R_1^{'} + R_2^{'})$$

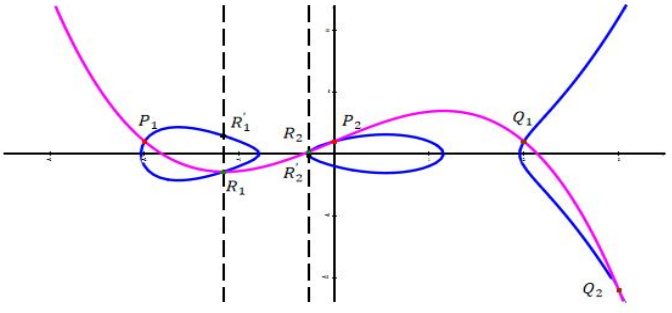


FIGURE 2.13: Hyperelliptic Curve Over the Real Field: $y^2 = x^5 - 5x^3 + 4x - 1$

**Definition 2.7.1. (Opposite Point, Special and Ordinary Points)**

If $P = (x, y)$ is a point on the curve then opposite of $P$ is $\overline{P} = (x, -y - h(x))$. If $P = \overline{P}$ then that point is called a special point and the rest are ordinary points.

**Example 2.7.2.** Consider the curve $y^2 = x^5 + 3x^4 - 7x^3 - 27x^2 - 18x \mod 11$ with $g = 2$, $h(x) = 0$ and then $f(x) = x^5 + 3x^4 - 7x^3 - 27x^2 - 18x$ and deg $f(x) = 5$. The multiplication table is given below.

TABLE 2.4: Multiplication Table

| $\times$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 2 | 0 | 2 | 4 | 6 | 8 | 10 | 1 | 3 | 5 | 7 | 9 |
| 3 | 0 | 3 | 6 | 9 | 1 | 4 | 7 | 10 | 2 | 5 | 8 |
| 4 | 0 | 4 | 8 | 1 | 5 | 9 | 2 | 6 | 10 | 3 | 7 |
| 5 | 0 | 5 | 10 | 4 | 9 | 3 | 8 | 2 | 7 | 1 | 6 |
| 6 | 0 | 6 | 1 | 7 | 2 | 8 | 3 | 9 | 4 | 10 | 5 |
| 7 | 0 | 7 | 3 | 10 | 6 | 2 | 9 | 5 | 1 | 8 | 4 |
| 8 | 0 | 8 | 5 | 2 | 10 | 7 | 4 | 1 | 9 | 6 | 3 |
| 9 | 0 | 9 | 7 | 5 | 3 | 1 | 10 | 8 | 6 | 4 | 2 |
| 10 | 0 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

The rational points on the curve are the points $(x, y)$ satisfy the above equation and are given below

$$\{(0,0), (2,1), (2,10), (3,0), (4,2), (4,9), (5,4), (5,7), (8,0), (9,0), (10,0), (\mathcal{O})\}.$$

The special and ordinary points are shown by the table:

TABLE 2.5: Opposite and Ordinary Points in Hyperelliptic Curve

| $P(x, y)$ | Opposite point $\overline{P}(x, -y - h(x))$ | Special Point $P = \overline{P}$ |
|---|---|---|
| $(2, 1)$ | $(2, -1)$ | No |
| $(2, 10)$ | $(2, -10)$ | No |
| $(3, 0)$ | $(3, 0)$ | Yes |
| $(4, 2)$ | $(4, -2)$ | No |
| $(4, 9)$ | $(4, -9)$ | No |
| $(5, 4)$ | $(5, -4)$ | No |
| $(5, 7)$ | $(5, -7)$ | No |
| $(8, 0)$ | $(8, 0)$ | Yes |
| $(9, 0)$ | $(9, 0)$ | Yes |
| $(10, 0)$ | $(10, 0)$ | Yes |

### 2.7.1 Divisors

In the study of hyperelliptic curves, divisors play a central role in understanding the curve's geometry, arithmetic, and applications in cryptography. Divisors are formal sums of points on the curve, and they provide a powerful tool for analyzing the curve's properties. Divisors are used to define the group law on the Jacobian. Given two divisors, their sum is computed using the Mumford representation, which provides a compact way to represent divisors on hyperelliptic curves.

**Definition 2.7.3. Divisors, Degree, Order**

Let $H$ be a hyper elliptic curve of genus 2, then the divisor on the curve is a linear combination $r_1P_1 + r_2P_2 + ... + r_nP_n$ of distinct points $P_1, P_2, ...P_n$ on $H$ and $r_1, r_2, ...r_n \in \mathbb{Z}$ and $n \in \mathbb{N}$ i.e.

$$D = \sum_{n \in \mathbb{N}} r_n P,$$

with only finitely many $r_n = 0$. The degree $\deg(D)$ and order $\text{ord}_P(D)$ of the divisor D at point $P$ is defined as: $\deg(D) = \sum_n r_n$ and $\text{Ord}_P(D) = r_n$.

**Example 2.7.4.** Consider the curve $H : y^2 = x^5 + 3 \bmod 7$ with $g = 2$.

Let $P_1 = (1, 2)$, $P_2 = (1, 5)$, $P_3 = (3, 1)$, $P_4 = (3, 6)$, $P_5 = (6, 4)$.

$$\begin{aligned} D_1 &= P_1 + P_2 - 2P_\infty \\ &= (1,2) + (1,5) - 2P_\infty \\ D_2 &= P_3 + P_4 + P_5 - 3P_\infty \\ &= (3,1) + (3,6) + (6,4) - 3P_\infty \end{aligned}$$

Here $\deg(D_1) = \sum_n r_n = 1 + 1 = 2$ and $\deg(D_2) = \sum_n r_n = 1 + 1 + 1 = 3$.

**Representation of the divisors:**

Cantor has employed the Mumford representation of the divisors, which is an effective method of representing divisors and is also appropriate for computations, because working with the divisors is difficult from an implementation point of view.

**Mumford Representation:** The divisors of hyper elliptic curve can be written in the form of polynomials $u(x)$ and $v(x)$ where $u(x), v(x) \in \mathbb{F}_q[x]$. As the polynomial belongs to the polynomial field $\mathbb{F}_q[x]$ however, it must satisfy the following three conditions [63]:

1. $u(x)$ must be a monic polynomial
2. deg $v(x) <$ deg $u(x) \leq g$
3. $u(x)|v(x)^2 + v(x)h(x) - f$

The polynomial expression of $u(x)$ is represented as

$$u(x) = \prod_{i=1}^{r}(x - x_i) \tag{2.5}$$

**Example 2.7.5.** Consider a hyperelliptic curve $y^2 = x^5+3x^4-7x^3-27x^2-18x$ with $g = 2$ over $\mathbb{F}_{11}$.
The rational points on the curve are

$\{(0,0), (2,1), (2,10), (3,0), (4,2), (4,9), (5,4), (5,7), (8,0), (9,0), (10,0), (\mathcal{O})\}$.

Let $P_1 = (2,1)$, $P_2 = (3,0)$, $P_3 = (2,1)$, $P_4 = (4,2)$ be the points of hyperelliptic curve. We defines the divisors $\mathrm{D}_1$ and $\mathrm{D}_2$ as follows

$$\mathrm{D}_1 = P_1 + P_2 = (2,1) + (3,0)$$
$$\mathrm{D}_2 = P_3 + P_4 = (2,1) + (4,2)$$

To express $\mathrm{D}_1$ in the polynomial form (Mumford representation), we use (2.5) i.e

$$\begin{aligned} u(x) &= (x - x_1)(x - x_2) \bmod 11 \\ &= (x - 2)(x - 3) \bmod 11 \\ &= x^2 + 6x + 6 \bmod 11. \end{aligned}$$

We find the number of all possible combinations of $\mathbb{F}_{11} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ to find $v_x$. For this purpose we fix the $y$-component as 0 and make all the combinations then likewise fix $y$-component as 1 to make all possible combinations and

so on.

$\{(0,0)$, $(1,0)$, $(2,0)$, $(3,0)$, $(4,0)$, $(5,0)$, $(6,0)$, $(7,0)$, $(8,0)$, $(9,0)$, $(10,0)$, $(0,1)$, $(1,1)$, $(2,1)$, $(3,1)$, $(4,1)$, $(5,1)$, $(6,1)$, $(7,1)$, $(8,1)$, $(9,1)$, $(10,1)$, $(0,2)$, $(1,2)$, $(2,2)$, $(3,2)$, $(4,2)$, $(5,2)$, $(6,2)$, $(7,2)$, $(8,2)$, $(9,2)$, $(10,2)$, $(0,3)$, $(1,3)$, $(2,3)$, $(3,3)$, $(4,3)$, $(5,3)$, $(6,3)$, $(7,3)$, $(8,3)$, $(9,3)$, $(10,3)$, $(0,4)$, $(1,4)$, $(2,4)$, $(3,4)$, $(4,4)$, $(5,4)$, $(6,4)$, $(7,4)$, $(8,4)$, $(9,4)$, $(10,4)$, $(0,5)$, $(1,5)$, $(2,5)$, $(3,5)$, $(4,5)$, $(5,5)$, $(6,5)$, $(7,5)$, $(8,5)$, $(9,5)$, $(10,5)$, $(0,6)$, $(1,6)$, $(2,6)$, $(3,6)$, $(4,6)$, $(5,6)$, $(6,6)$, $(7,6)$, $(8,6)$, $(9,6)$, $(10,6)$, $(0,7)$, $(1,7)$, $(2,7)$, $(3,7)$, $(4,7)$, $(5,7)$, $(6,7)$, $(7,7)$, $(8,7)$, $(9,7)$, $(10,7)$, $(0,8)$, $(1,8)$, $(2,8)$, $(3,8)$, $(4,8)$, $(5,8)$, $(6,8)$, $(7,8)$, $(8,8)$, $(9,8)$, $(10,8)$, $(0,9)$, $(1,9)$, $(2,9)$, $(3,9)$, $(4,9)$, $(5,9)$, $(6,9)$, $(7,9)$, $(8,9)$, $(9,9)$, $(10,9)$, $(0,10)$, $(1,10)$, $(2,10)$, $(3,10)$, $(4,10)$, $(5,10)$, $(6,10)$, $(7,10)$ $(8,10)$, $(9,10)$, $(10,10)\}$.

From above combinations, we check which pair will satisfy the above mentioned condition (3). The pair (10,3) will satisfy i.e

$$u(x)|v(x)^2 + v(x)h(x) - f(x) \bmod 11$$
$$(x^2 + 6x + 6)|(10x + 3)^2 + 0 - x^5 - 3x^4 + 7x^3 + 27x^2 + 18x \bmod 11$$
$$(x^2 + 6x + 6)|10x^5 + 8x^4 + 7x^3 + 6x^2 + 1x + 9 \mod 11.$$

Now we can write $v(x)$ as

$$v(x) = v_1(x) + v_2 = 10x + 3,$$

then

$$\mathrm{D}_1 = [x^2 + 6x + 6, 10x + 3].$$

Similarly to express $\mathrm{D}_2 = (2,1) + (4,2)$, we find $u(x)$ and $v(x)$ by same method.

$$\begin{aligned} u(x) &= (x-2)(x-4) \mod 11 \\ &= x^2 - 6x + 8 \mod 11. \\ &= x^2 + 5x + 8 \mod 11. \end{aligned}$$

To find $v(x)$ we will check which pair from the combinations of $\mathbb{F}_{11}$ satisfy the condition (3). The only pair $(6, 0)$ will satisfy, so $v(x)$ will be

$$\begin{aligned} v(x) &= v_1(x) + v_2 \\ &= 6x. \end{aligned}$$

Hence $\mathrm{D}_2 = [x^2 + 5x + 8, 6x]$

### 2.7.2 Hyperelliptic Curve Discrete Logarithm Problem

Let $\mathrm{D}_1$ and $\mathrm{D}_2$ be two divisors of hyperelliptic curve H, the problem of finding the integer $n \in \mathbb{Z}$, such that $n\mathrm{D}_1 = \mathrm{D}_2$, is known as the hyperelliptic curve discrete logarithm problem (HECDLP). Solving for $n$ is computationally infeasible and the security of hyperelliptic curve cryptography (HECC) relies entirely on the difficulty of HECDLP [45].

### 2.7.3 Digital Signature

To confirm the sender's identity on a message, an electronic signature is used [64]. It is a procedure to make sure data has not been altered. It has the same effect as a written signature. Digital signatures are created using hash functions and digital documents. Every entity in a proposed digital signature scheme has a private key and a public key of their own. The sender's private key is used to create the digital signature and the sender's public key is used to verify it [65]. Two algorithms are used to generate a digital signature, one for signature generation and the other for signature verification. The following security qualities are obtained because of digital signatures.

1. **Correctness:** Without the public key of sender it is impossible to verify the signatures.

2. **Authenticity:** This confirms that the right individual signed the message.

3. **Unforgeability:** It indicates a message can only have one distinct, authentic signature from an authorized signer.

4. **Non-repudiation:** The message's signer can not deny their signature.

5. **Integrity:** This means that there was no change in the message during transmission.

6. **Non-reusability:** It is not possible to sign several messages using the same signature that was used on one.

The sender first computes a hash value from the original data using a hash function. This hash value is then encrypted using the sender's private key, creating the digital signature. To verify the signature, the receiver decrypts it using the sender's public key, recovering the original hash value. The receiver then applies same hash function to the received message to generate a new hash value. By comparing the decrypted hash with the newly computed hash, the receiver can authenticate the message. If the two hash values match, the message is confirmed as authentic.

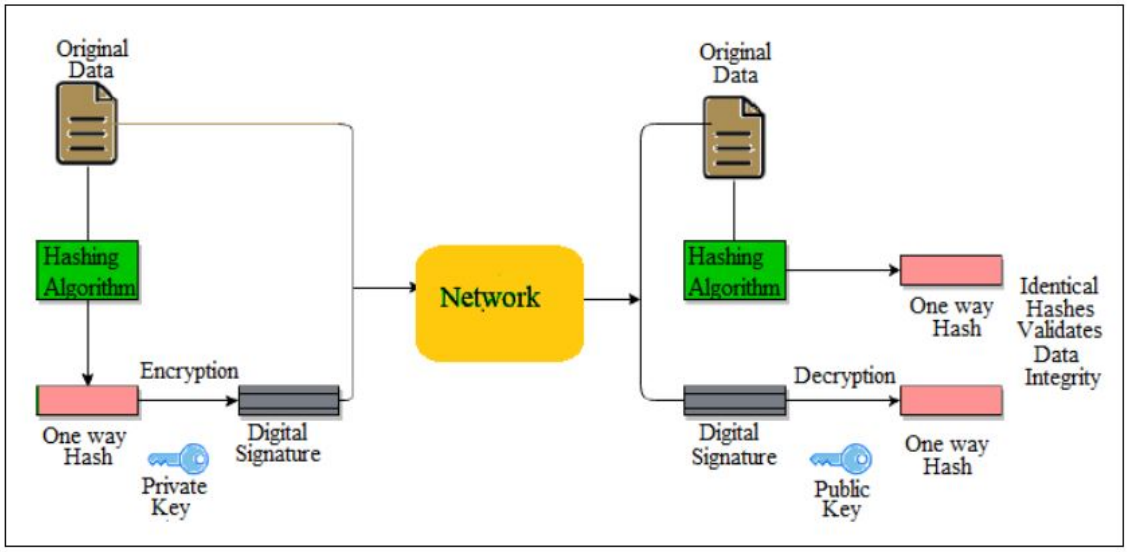


FIGURE 2.14: Digital Signature

### 2.7.4 Signcryption

Signcryption is a new cryptographic technique that combines the security requirements of digital signature and encryption into a single logical step. In comparison

to the traditionally applied signature-then-encryption technique, it is useful in minimizing the computational and communicational cost. The many security features of public verifiability, authentication, non-repudiation, confidentiality, unforgeability and forward secrecy are provided by signcryption techniques. Depending on the need, a variety of signcryption methods with different levels of security and computational cost were previously introduced, each with own challenges and flaws, while offering different computing costs and security. Zheng [13] discovered a new technique in 1997, called signcryption that at the same time performs both the functions of encryption and digital signatures. A typical signcryption scheme consists of three algorithms: Key generation, Signcryption (SC) and Unsigncryption (USC).

#### 2.7.4.1 Key Generation

1. Sender chooses a private key $z_A$ from the range $\{1, 2, ...p-1\}$ and compute public key $X_A = x^{z_A} \text{mod } p$.
2. Receiver chooses private key $z_B$ from the range $\{1, 2, ...p-1\}$ and compute public key $X_B = x^{z_B} \text{mod } p$.

#### 2.7.4.2 Signcryption

1. Sender chooses an integer $x \in \{1, 2, ...p-1\}$.
2. He use public key of receiver $X_B$, the integer $x$ and one way hash function $h$ to compute
$$u = h(X_B x) \text{ mod } p.$$
3. A 128 bit value is divided into two parts of 64 bits, numbered as $u_1$ and $u_2$.
4. Receiver use the public key encryption scheme E to encrypt the message $m$ with the key $u_1$. It will give ciphertext
$$c = Eu_1(m).$$

5. He uses the key $u_2$ message $m$ and one way keyed hash $H_k$ value to compute $r$

$$r = H_k u_2(m).$$

6. The signature parameter $s$ is computed by sender. By using $x$, the secret key $z_A$, the large prime number $p$ and $r$ to get

$$s = \frac{x}{r + z_A} \mod p.$$

7. The values $c$, $r$ and $s$ are now available to sender. In order to complete the task, he send these values to receiver.

#### 2.7.4.3 Unsigncryption

1. Receiver recieves the values $c$, $r$ and $s$ from sender. He uses the values $r$ and $s$, his secret key $z_B$, sender's public key $X_A$, and $u$ to calculate a hash value of 128 bits

$$z = h(X_A.x^r)^s z_B \mod p.$$

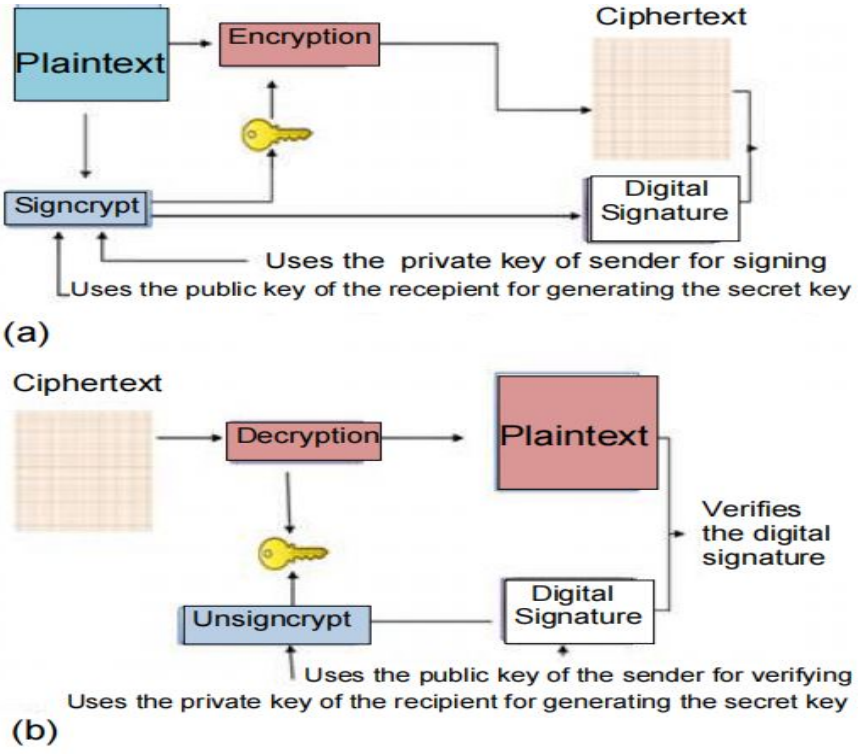


FIGURE 2.15: Signcryption Model (a)Signcryption (b)Unsigncryption

The same key pair $\{u_1, u_2\}$ is generated by dividing 128 bit hash value into two 64 bit pieces.

2. For decryption of ciphertext c, receiver uses the key $u_1$, which will give him the message $m$.

$$m = Du_1(c).$$

3. Receiver will verify the evaluation

$$r = H_k u_2(m).$$

If its match, the message $m$ was sent and signed by the sender.

### 2.7.5 Public Key Generator (PKG)

It serves as a trusted third party responsible for generating public keys based on the user's identification. In identity-based public key cryptography, the user's public keys are created using the Public Key Generator (PKG).

### 2.7.6 Certificate Based Cryptography

Public key infrastructure is used by certificate-based public key systems, which require less data for certificate distribution and validation. Only the decryption process uses the certificates in certificate-based encryption schemes. Hence, digital certificates or authentication are not required in order to encrypt a communication. The encryption requires the user's identity.

### 2.7.7 Certificateless Cryptography

Certificateless cryptography does not require certificates in order to confirm the legitimacy of public keys. The Certificate Authority (CA), often referred to as a trusted third party (TTP), is not necessary for CL cryptography [66]. CL-PKC

combines the benefits of identity-based cryptography and traditional PKC. ID-based encryption uses user identities, such as email addresses and IP addresses, as the public key instead of digital certificates, with a trusted third party creating the secret key. In contrast, the secret key in CL-PKC is generated jointly by the PKG and the user.

## 2.8 Hash Function

A function that converts an arbitrary length of data to a fixed length of data is called a hash function. Hashing values, or simply hashes, are the results of the hash function. Hash functions in cryptography are fairly simple to compute but challenging to reverse. There is a chance that two distinct messages will have the same hash values since hash functions decrease the input size to a fixed length.

### 2.8.1 Properties of Cryptographic Hash Functions

Hash function has following properties.

1. **Efficiency**

   For any given input it is easy to calculate hash value.

2. **Pre-image resistance**

   For given any hash value it is infeasible to find corresponding input.

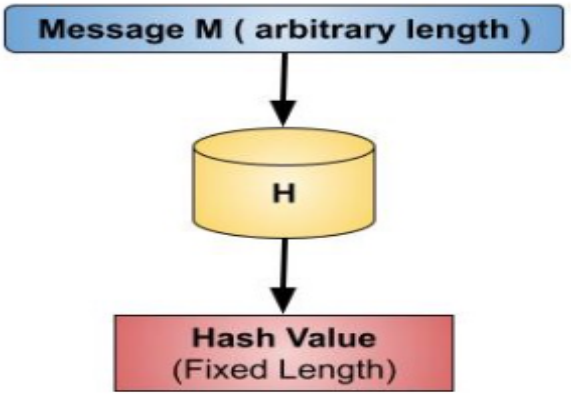


FIGURE 2.16: Cryptographic Hash Function

3. **Second pre-image resistance** For any given message $m_1$ it is infeasible to find another message $m_2$ such that both $m_1$ and $m_2$ have the same hash value.

4. **Collision resistance**

   With same hash value it is infeasible to find two different messages. In general for $n$ possibilities, it require $n^{\frac{1}{2}}$ trials to find collision.

5. **Sensitivity**

   It means minor changes in data produce major changes in output.

### 2.8.2 Cryptographic Hash Algorithms

There are different versions of cryptographic hash functions each uses different length of input and produces some fixed output. Most common hash functions are message digest (MD4), MD5, Secure hash algorithm (SHA), SHA-1, SHA-2 and SHA-3. SHA-2 has different versions SHA-224, SHA-256, SHA-384 and SHA-512 [67].

### 2.8.3 Characteristic of Hash Algorithms

SHA-1 uses 512 bit block size and gives 160 bit hash value with 80 rounds. SHA-256/224 uses 512 bit block size and gives 256/224 bit hash value. They have total 64 rounds. SHA-512/384 uses 1024 block size and gives 512/384 bit hash value. They have total 80 rounds.

| **Algorithm** | **Output Size** | **Block Size** | **Message Size** | **Rounds** | **Collision** |
|---|---|---|---|---|---|
| SHA-0 | 160 | 512 | $2^{64}-1$ | 80 | Yes |
| SHA-1 | 160 | 512 | $2^{64}-1$ | 80 | $2^{63}$ Attack |
| SHA-256 | 256 | 512 | $2^{64}-1$ | 64 | No |
| SHA-224 | 224 | 512 | $2^{64}-1$ | 64 | No |
| SHA-512 | 512 | 1024 | $2^{128}-1$ | 80 | No |
| SHA-384 | 384 | 1024 | $2^{128}-1$ | 80 | No |

TABLE 2.6: Comparison of Cryptographic Hash Function

# Chapter 3

# Certificateless Elliptic Curve Aggregate Signcryption Scheme

In this chapter, the work of Yu and Ren [34] on "Certificateless Elliptic Curve Aggregate Signcryption Scheme(CL-ECASC)" is presented. The CL-ECASC represents a cryptographic protocol designed to address the challenges of secure data transmission in modern communication systems. Unlike traditional encryption schemes, it provides robust security while simultaneously enabling efficient aggregation of multiple encrypted messages. This chapter will cover the mechanism and security of CL-ECASC.

## 3.1 Introduction

In the Internet of Things (IoTs), where a variety of devices are linked to the network, authentication threats assume a new dimension. As a result, the security system needs to be strong enough to give the network enough power without sacrificing efficiency.

A rational idea of being free from risks is security. The growing number of attacks and their creative methods for obtaining access to network data are evident in today's world of internet technology. Furthermore, data centers frequently have to

send various encrypted messages to various departments on time and with proper encryption. In this case, conventional multi-receiver encryption techniques are unsuitable. Using the conventional encryption method to encrypt separate messages will result in significantly higher computing overhead. Consequently, the issue of terminal and network authentication is addressed by aggregate signcryption (ASC), which combines the signcrypted ciphertexts produced by several terminals into a single ciphertext.

In public key cryptography, secure and authenticated communication is required. The traditional approach to guarantee the message confidentiality and authentication is signature-then-encryption. That is first the sender of a message would sign the message with some digital signature scheme and then encryption is performed with the help of private key cryptography. By using recipient's public key, the message encryption key is then encrypted. This two step approach is called sign-then-encryption. The generation of signature and encryption requires more machine cycles and then extra bits are added to the original message. Similarly same amount of computational cost is required for decryption and verification of signature. The attacks on the cryptographic scheme based upon sign-then-encryption are increasing very fast.

## 3.2 Aggregate Signcryption

Compared to the conventional signature-then-encryption techniques, signcryption is used to lower the computational cost and communication overhead. It combines digital signature and encryption in a single step. Some most important characteristics of signcryption include accuracy, security in terms of forward secrecy, efficiency and unforgeability [68]. In recent years, several signcryption methods have been created. It is also important to note a few other noteworthy applications of signcryption in various IoT-based infrastructures [69–76]. Short signature is created by aggregating the various signatures of $n$ signers in some fashion. An aggregate signature (AS) can guarantee non-repudiation for distinct messages. The amount of storage space needed for signatures and the processing cost of

the verification process is decreased using aggregate signatures. Application areas for AS include cloud computing, wireless networks, IoTs, 5G environments, and electronic medical devices.

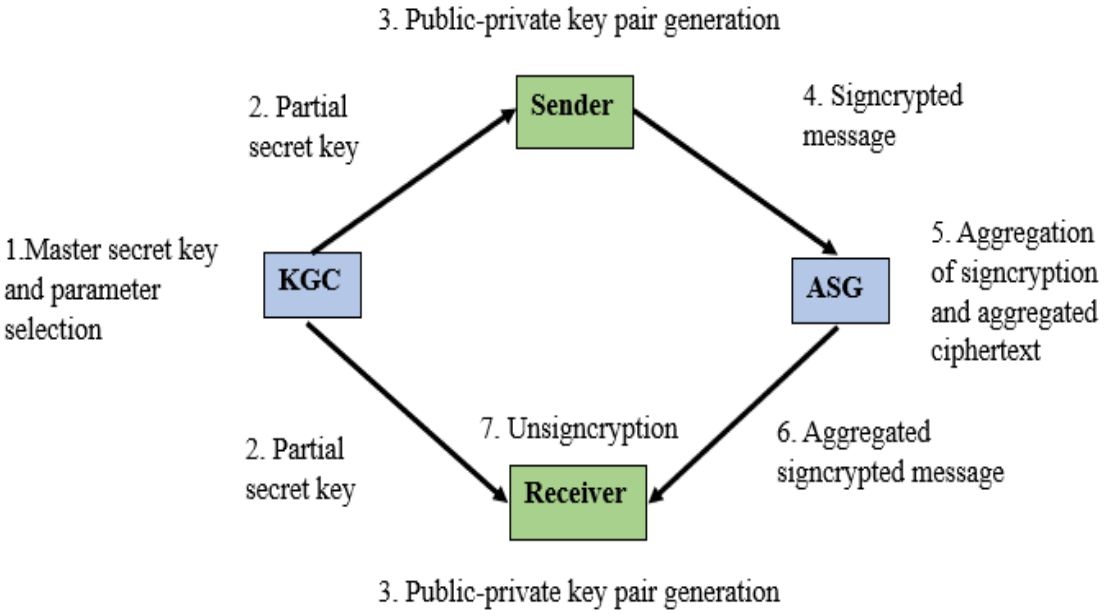


FIGURE 3.1: Aggregate Signcryption Model

## 3.3 Certificateless Aggregate Signcryption Scheme (CL-ECASC)

A new certificateless elliptic curve aggregate signcryption scheme (CL-ECASC) of Yu and Ren [34] is developed in the context of 5G wireless networks by fusing ECC with CL-ASC. The confidentiality and authenticity of transmission data can be guaranteed by CL-ECASC, and it simplifies several verifications of various signatures into a single verification. Under the difficulties of the elliptic curve discrete logarithm problem (ECDLP) and elliptic curve computation Diffie-Hellman (ECCDH), CL-ECASC is able to achieve strong security. Its advantages over other cryptosystems with the same level of security include lower key lengths and faster processing. The conflict between the sender and the recipient can be settled by any third party. Without access to the sender or recipient's private key, a third

party can confirm the authenticity of information conveyed using signcryption. The proposed scheme, includes key generation centre (KGC), an aggregate set of $\eta$ users with identity $I_{i=1}^{\xi}$, a receiver with identity $I_R$ and an aggregate generator. In this scheme the elements of $Z_\rho^*$ are denoted by lowercase letters and elliptic curve points in $E_\rho(a, b)$ are denoted by uppercase letters.

### 3.3.1 Global Parameters

Following are the global parameters for this scheme.

$\rho$ : A prime number

$G$ : A generator of an elliptic curve $E_\rho(a, b)$ with large order $\alpha$ such that

$$\alpha G = \mathcal{O},$$

where

$\alpha$ : Order of base point $G$

$\mathcal{O}$ : Point at infinity

$\hbar_1^*, \hbar_2^*, \hbar_3^*$ : Hash functions

### 3.3.2 Notations

The scheme includes the following symbols.

$\rho$ : A prime number

$x$ : The master key

T : Identity length

$\ell$ : The message length

$G_\rho$ : Additive cyclic group

$\mathbb{F}_\rho$ : Finite field of order $\rho$

$a_i$ : Private key

$I_i$ : User's identity

$c$ : Aggregate ciphertext

$\gamma$ : The system parameter

Below are the algorithms used in our proposed scheme.

1. Setup
2. Key generation
3. Extract
4. Signcrypt
5. Aggregate
6. Unsigncrypt

**Algorithm 3.3.1.** (**Setup**) It contains four steps which are given below.

**Step 1:** KGC selects an elliptic curve $E_\rho(a, b)$ over a finite field $\mathbb{F}_\rho$, a large prime $\rho$ of $k$ bits and elliptic curve base point $G$ of prime order $\alpha$.

**Step 2:** To calculate the system public key, a master secret key $x$ is chosen by KGC from $Z_\alpha^*$.

$$P_{pub} = xG \mod p$$

**Step 3:** KGC selects cryptographic hash functions as

1. $\hbar_1^* : \{0,1\}^{\mathrm{T}} \times G_\rho \longrightarrow Z_\rho^*$
2. $\hbar_2^* : \{0,1\}^{\mathrm{T}} \times G_\rho \longrightarrow Z_\rho^*$
3. $\hbar_3^* : G_\rho \times G_\rho \longrightarrow \{0,1\}^\ell$

**Step 4:** The following system parameters are published.

$$\gamma = (\mathbb{F}_\rho, E_\rho(a,b), \rho, G_\rho, G, P, \ell, \hbar_1^*, \hbar_2^*, \hbar_3^*)$$

### Algorithm 3.3.2. (**Key Generation**)

It consists of two steps.

**Step 1:** A random value $a_i \in (1, \alpha)$ is chosen which act as user's private key with identity $I_i$.

**Step 2:** The user's public key is calculated as

$$\wp_i = a_i G.$$

### Algorithm 3.3.3. (**Extract**)

Extraction of partial public and private key is performed in two steps.

**Step 1:** To get partial public key and partial private key respectively, KGC chooses a random value $b_i \in (1, \alpha)$.

$$W_i = b_i G.$$

$$\delta_i = b_i + x.\hbar_1^*(I_i, \wp_i) \mod \alpha$$

**Step 2:** KGC determines

$$R_i = \delta_i G + b_i \wp_i \mod \alpha,$$

and transmits $(\delta_i, R_i, W_i)$ to the user with identity $I_i$. The authenticity of $(W_i, \delta_i)$ can be verified by

$$\begin{aligned} \delta_i G &= (b_i + x.\hbar_1^*(I_i, \wp_i) \mod \alpha).G \\ &= b_i G + xG.\hbar_1^*(I_i, \wp_i) \mod \alpha \\ &= W_i + \hbar_1^*(I_i, \wp_i) P_{pub} \end{aligned}$$

and

$$\begin{aligned} R_i &= \delta_i G + b_i \wp_i \\ &= \delta_i G + b_i (a_i G) \\ &= \delta_i G + a_i (b_i G) \\ &= \delta_i G + a_i W_i \end{aligned}$$

**Algorithm 3.3.4.** (**Signcrypt**)

Signcryption is performed through the following three steps.

**Step 1:** The user with identity $I_i$ chooses a random value $t_i \in (1, \alpha)$ to find

$$\mathrm{T}_i = t_i G.$$

**Step 2:** The user then calculates

$$B_i = t_i(W_R + \hbar_1^*(I_R, \wp_R)P_{pub} + \wp_R) \mod \alpha,$$

$$r_i = M_i \oplus \hbar_3^*(B_i, \mathrm{T}_i).$$

**Step 3:** The user calculates $\sigma_i$, $\phi_i$ and produce ciphertext $c$ as

$$\sigma_i = \hbar_2^*(M_i, \mathrm{T}_i),$$

$$\phi_i = \sigma_i(a_i + \delta_i) + t_i.$$

$$c = (\mathrm{T}_i, r_i, \phi_i).$$

**Algorithm 3.3.5.** (**Aggregate**)

Aggregation is performed by two steps.

**Step 1:** The users with identity $I_{i=1}^{\alpha}$ generate a collection of distinct ciphertexts.

$$\{c_i = (\mathrm{T}_i, r_i, \phi_i)\}_{i=1}^{\alpha}.$$

**Step 2:** The aggregator then calculates

$$\phi = \Sigma_{i=1}^{\alpha}\phi_i,$$

and outputs an aggregate ciphertext

$$c = (\mathrm{T}_i, ..., \mathrm{T}_\xi, r_i, ..., r_\alpha, \phi).$$

**Algorithm 3.3.6.** (**Unsigncrypt**) Unsigncryption is performed by the following step.

**Step 1:** The recipient of the public key $(\wp_i, W_i)$ of users with identity $I_{i=1}^{\alpha}$ and calculates

$$B_i = (\delta_R + a_R)\mathrm{T}_i,$$

and recovers

$$M_i = r_i \oplus \hbar_3^*(B_i, \mathrm{T}_i).$$

The receiver accept the message $M_{i=1}^{\alpha}$ if

$$\phi G = \Sigma_{i=1}^{\alpha}\hbar_2^*(M_i, \mathrm{T}_i).(W_i + \hbar_1^*(I_i, \sigma_i)P_{pub} + \sigma_i) + \Sigma_{i=1}^{\alpha}\mathrm{T}_i$$

holds.

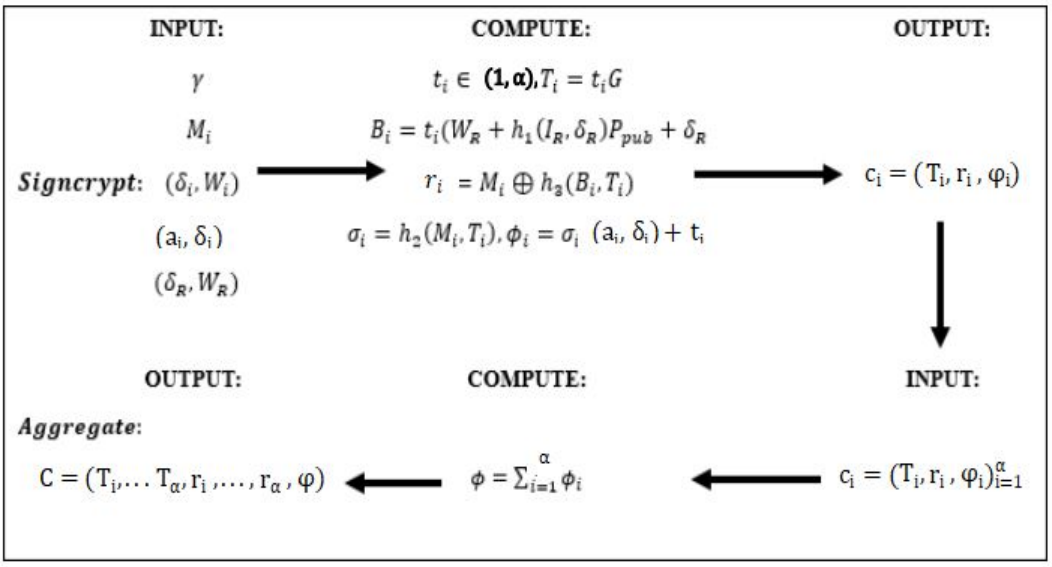


FIGURE 3.2: Flow Process of Signcryption and Aggregation

## 3.4 Computational Cost

CL-ECASC has a significantly reduced computational cost. Compared to other existing schemes, this scheme ensures both high computation efficiency and a high level security. The comparison between computational cost of CL-ECASC and the

existing schemes [77–80] (by making several experiments on a Windows 10 system with an Inter(R) Core(TM) i7 – 9750H CPU at 2.60 GHz and 16.00 GB RAM) is given below.

| **Schemes** | [77] | [78] | [79] | [80] | **ECC-CLASC** |
|---|---|---|---|---|---|
| Signcryption cost | $n(4O_M + O_P)$ | $2nO_M + nO_P$ | $7nO_M$ | $6nO_M$ | $5nO_M$ |
| Unsigncryption cost | $nO_M + 3O_P$ | $nO_M + (n+2)O_P$ | $5nO_M$ | $8nO_M$ | $2nO_M$ |
| Execution time | $\frac{5nO_M+3O_P+nO_P}{49608.13ms}$ | $\frac{3nO_M+2nO_P+2O_P}{75565.42ms}$ | $\frac{12nO_M}{40320ms}$ | $\frac{14nO_M}{47040ms}$ | $\frac{7nO_M}{23520ms}$ |
| Confidentiality | Yes | Yes | Yes | Yes | Yes |
| Unforgeability | Yes | Yes | Yes | Yes | Yes |

TABLE 3.1: Computational Cost Analysis

where $O_M$ is the number of a multiplication operation (one multiplication operation $O_M$ takes 3.36 ms) and $O_P$ is the number of a bilinear pairing operations (one multiplication operation $O_P$ takes 32.71 ms)

# Chapter 4

# Certificateless Hyperelliptic Curve Aggregate Signcryption Scheme

In order to maintain same level of security while reducing computational cost, the hyperelliptic curve is proposed to replace the elliptic curve for the development of a certificateless signcryption system. By extending the concepts of elliptic curve cryptography to curves of larger genus, it provides a vast mathematical landscape. Hyperelliptic curve is a field for research and exploration because of its expansion, which presents new difficulties and opportunities for cryptographic applications.

## 4.1 Introduction

Reliable and efficient cryptographic foundations are a constant quest in the field of modern cryptography, where safeguarding sensitive data is of utmost importance. Of them, elliptic curve cryptography (ECC) has become an essential, providing safe and economical solutions for a range of cryptographic applications. As cryptographic requirements change and threats grow increasingly complex, the need for alternative mathematical approaches that ensure equivalent security with improved efficiency becomes more evident. An IoT certificateless signcryption scheme

(IoT-CS) using hyperelliptic curve cryptography, is designed to enhance security while reducing computational and communication cost in IoT environments.

## 4.2 Proposed Scheme

This proposed scheme focuses on building safe and efficient methods to use IoT devices. One way to guarantee security criteria like integrity, confidentiality, non-repudiation and authenticity is to use a signature-then-encryption process. However, because this method generates the message's encryption and signature in two different processes, it is not suitable for low-processing IoT platforms. In order to increase efficiency, sincryption techniques were introduced by Zheng [13], which involve encryption and signature in a single logical step. However, public key cryptography (PKC) is the base of Zheng's method. The idea of public key infrastructure (PKI), that uses a certificate authority (CA) to connect the public key with certificates, was established to overcome the shortcomings in PKC-based schemes [81]. Nevertheless, there are issues with certificate distribution, storage, and manufacture with this approach [82]. Identity-based cryptography (IBC), a solution to these drawbacks, was proposed in [83]. Without the use of a CA, IBC allows participants to generate public keys straight from their identities, including phone numbers and email addresses. The reliable server that serves as the key generation center (KGC) generates each participating entity's private key. To combine the functions of encryption and signing in one step, the signcryption principle was implemented [84]. Nevertheless, IBC-based schemes face the issue of key escrow, whereby the KGC possesses complete knowledge of all participants' private keys. A solution to this issue was proposed in [85] with the concept of Certificateless Public Key Infrastructure (CPKI). Within CPKI, a private key of the participant consists of two components: the private key supplied by the KGC and a value created by the participant that is secret. In order to combine the functions of encryption and signature in one step, the idea of certificateless signcryption was presented [86]. A 160-bit key size is the foundation for the ECC's efficiency and security [87]. The concept of hyperelliptic curve cryptography was used in order to

improve the efficiency of ECC based schemes [88]. By using 80-bit tiny key sizes, the HECC provides an equivalent level of security to that of the ECC [89–91].

### 4.2.1 Global Parameters

Following are the global parameters for this scheme:

$H$ : Hyperelliptic curve

$D$ : A divisor of hyperelliptic curve

$$\alpha D = P_{\infty},$$

where

$\alpha$ : Order of divisor

$P_{\infty}$ : Point at infinity

$\hbar_1^*, \hbar_2^*, \hbar_3^*$ : Hash functions

### 4.2.2 Notations

The scheme includes following symbols:

$\rho$ : A prime number

$x$ : The master key

T : Identity length

$\ell$ : The message length

$D_\rho$ : Additive cyclic group of divisors

$\mathbb{F}_\rho$ : Finite field of order $\rho$

$a_i$ : Private key

$I_i$ : User's identity

$c$ : Aggregate ciphertext

$\gamma$ : The system parameter

$D$ : Divisor of hyoperelliptic curve

Below are the algorithms that are used in the proposed scheme.

1. Setup
2. Key generation
3. Extract
4. Signcrypt
5. Aggregate
6. Unsigncrypt

## Algorithm 4.2.1. (**Setup**)

It consists of four steps given below.

**Step 1:** KGC selects an hyperelliptic curve $H_\rho(a, b)$ over a finite field $\mathbb{F}_\rho$ with a large prime $\rho$ of $k$ bits and a divisor $D$ on hyperelliptic curve $H_\rho$ with $\alpha$ as prime order of hyperelliptic curve.

**Step 2:** KGC selects a master key $x$ from $Z_\alpha^*$ to calculate the public key.

$$P = xD \mod p$$

**Step 3:** KGC selects cryptography hash functions as

$\hbar_1^* : \{0,1\}^{\mathrm{T}} \times D_\rho \longrightarrow Z_\rho^*$

$\hbar_2^* : \{0,1\}^l \times D_p \longrightarrow Z_\rho^*$

$\hbar_3^* : D_\rho \times D_\rho \longrightarrow \{0,1\}^l$

**Step 4:** The following set of system parameter is published.

$$\gamma = (\mathbb{F}_\rho, H, \rho, D_\rho, D, P, \ell, \hbar_1^*, \hbar_2^*, \hbar_3^*) \tag{4.1}$$

## Algorithm 4.2.2. (**Key Generation**)

It consists of two steps as stated below:

**Step 1:** A random value $a_i \in (1, \alpha)$ is chosen which act as user's private key.

**Step 2:** User calculate his public key as

$$\delta_i = a_i D_i \mod \alpha.$$

## Algorithm 4.2.3. (Extract:)

It contains the following two steps.

**Step 1:** KGC selects a random value $b_i \in (1, \alpha)$ to get partial public key and partial private key respectively.

$$W_i = b_i D_i$$

$$p_i = b_i + x\hbar_1^*(I_i, \delta_i) \mod \alpha$$

**Step 2:** KGC computes

$$R_i = p_i D_i + b_i \delta_i,$$

to transmits $(p_i, R_i, W_i)$ to the user. The authenticity of $(W_i, p_i)$ can be verified by

$$p_i D = W_i + \hbar_1^*(I_i, \delta_i) P_{pub} \mod \alpha \tag{4.2}$$

$$R_i = p_i D_i + a_i W_i.$$

## Algorithm 4.2.4. (Signcrypt:)

Signcryption is performed through the following three steps.

**Step 1:** The user chooses a random value $t_i \in (1, \alpha)$ with identity $I_i$ and finds

$$\mathrm{T}_i = t_i D_i.$$

**Step 2:** The user then calculates

$$B_i = t_i(W_R + \hbar_1^*(I_R, \delta_R) P_{pub} + \delta_R) \mod \alpha$$

$$r_i = M_i \oplus \hbar_3^*(B_i, \mathrm{T}_i).$$

**Step 3:** Using the hash function $\hbar_2^*$, the user calculates the ciphertext $c_i^{'}$ as :

$$\sigma_i = \hbar_2^*(M_i, \mathrm{T}_i)$$

$$\phi_i = \sigma_i(a_i + p_i) + t_i \mod \alpha$$

$$c_i^{'} = (\mathrm{T}_i, r_i, \phi_i).$$

**Algorithm 4.2.5.** (**Aggregate:**)

Aggregation is performed through the following two steps:

**Step 1:** The user with the identity $I_{i=1}^{\alpha}$ produce a set of different ciphertexts.

$$\{\sigma_i = (\mathrm{T}_i, r_i, \phi_i)\}_{i=1}^{\alpha}$$

**Step 2:** The aggregator then calculates $\phi$ and outputs an aggregate ciphertext $c$

$$\phi = \Sigma_{i=1}^{\alpha}\phi_i$$

$$c = (\mathrm{T}_i, ..., \mathrm{T}_\alpha, r_i, ..., r_\alpha, \phi)$$

**Algorithm 4.2.6.** (**Unsigncrypt:**)

It consists of one step.

**Step 1:** The public key $(\delta_i, W_i)$ of users with identity $I_{i=1}^{\alpha}$ is received by the recipient and recipient then calculates

$$B_i = (p_R + a_R)\mathrm{T}_i \mod \alpha$$

and recovers

$$M_i = r_i \oplus \hbar_3^*(B_i, \mathrm{T}_i)$$

The receiver accept the message $M_{i=1}^{\alpha}$ if

$$\phi D = \Sigma_{i=1}^{\alpha}\hbar_2^*(M_i, \mathrm{T}_i).(W_i + \hbar_1^*(I_i, \delta_i)P_{pub} + \delta_i) + \Sigma_{i=1}^{\alpha}\mathrm{T}_i \mod \alpha$$

holds.

**Correctness:** The correctness is obtained by using the following method.

$$\begin{aligned}
\phi D &= \Sigma_{i=1}^{\alpha} \phi_i D \\
&= \Sigma_{i=1}^{\alpha} [\sigma_i (a_i + p_i) + t_i] D \\
&= \Sigma_{i=1}^{\alpha} [\sigma_i (a_i D + p_i D) + t_i D] \\
&= \Sigma_{i=1}^{\alpha} [h_2^*(M_i, T_i)(a_i D + p_i D) + t_i D] \\
&= \Sigma_{i=1}^{\alpha} [h_2^*(M_i, T_i)(a_i D + (W_i + h_1^*(I_i, \delta_i) P_{pub})) + t_i D] \\
&= \Sigma_{i=1}^{\alpha} h_2^*(M_i, T_i).(\delta_i + W_i + h_1^*(I_i, \delta_i) P_{pub}) + \Sigma_{i=1}^{\alpha} t_i D \\
&= \Sigma_{i=1}^{\alpha} h_2^*(M_i, T_i).(\delta_i + W_i + h_1^*(I_i, \delta_i) P_{pub}) + \Sigma_{i=1}^{\alpha} T_i
\end{aligned}$$

CL-HECASC provides an efficient and secure solution for combining the functionalities of encryption, authentication and signature aggregation in a single cryptographic framework. By utilizing the mathematical characteristics of hyperelliptic curves, this proposed scheme achieves high security with relatively small key sizes, making it suitable for resource-constrained environments such as IoT devices and mobile networks.

# Chapter 5

# Analysis of the Proposed Scheme

The security analysis of the aggregate signcryption technique based on hyperelliptic curves, which was proposed in Chapter 4, is covered in this chapter. The hyperelliptic curve discrete logarithm problem (HECDLP) is the foundation for the scheme's security. Subsequently, an analysis of communication and computational cost is shown and compared with other existing schemes.

## 5.1 Security Analysis

The security features will be covered in this section. The security is dependent upon the HECDLP. The security analysis was conducted with consideration for the following assumptions. The private key and secret values of the $m^{th}$ and $n^{th}$ nodes are unknown to the attacker, only the relevant participating entity (KGC and IoT nodes) is known to them. An attacker is unable to break $c$ and authenticated message since the encryption scheme is sufficiently secure.

### 5.1.1 Confidentiality

An assurance that sensitive data would remain hidden while being sent is known as confidentiality. The $m^{th}$ node and $n^{th}$ node initially exchange plaintext versions of their public keys and identities with each other since secrecy is not necessary. The

message $\{T_m, c, c^{'}, W_m\}$ is then sent to the $n^{th}$ node via the $m^{th}$ node. Because the ciphertext $c$ is dependent on the secret random number $\alpha$, the adversary is unable to decipher it. An attacker cannot compute $\alpha$ given $P$ and $D$ (4.2.1), because finding $\alpha$ from $D$ is equivalent to HECDLP. Since adversary is dependent on the private values ($a_i$ and $b$) of $m^{th}$ node, it is unable to extract any knowledge from signature. Additionally, nothing is revealed in the messages that the $n^{th}$ node sends to the $m^{th}$ node. An attacker cannot deduce any information from time stamp and hash messages. Consequently, confidentiality aspects are successfully provided by the existing protocol.

### 5.1.2 Authentication

IoT nodes must authenticate one another at the beginning of every session and vice versa to provide safe communication.

$m^{th}$ **node authentication:** Once the $m^{th}$ node has sent the message $\{c, c^{'}, W\}$, the $n^{th}$ node computes the session key, the $n^{th}$ node verifies the signature of the $m^{th}$ node. The $n^{th}$ node properly authenticates the $m^{th}$ node if step 2 (4.2) holds true. In the scenario that an adversary serves as a genuine node, it would have to produce a valid signature. An adversary could not, however, obtain the correct value of signature since it is dependent on $m^{th}$ private node values.

$n^{th}$ **node authentication:** It is computed by the $m^{th}$ node upon obtaining the authenticated message. If messages of both nodes are same, $n^{th}$ node has authenticated by the $m^{th}$ node successfully. An attacker must convey the correct message if it poses as a genuine node. But since authenticated message is based upon the $n^{th}$ node private key, an adversary finds it difficult to transmit the right message.

### 5.1.3 Non-repudiation

The $m^{th}$ node private key determines the value of signature that it transfers to the $n^{th}$ node. The $n^{th}$ node private key is also the basis for the message that $m^{th}$ node receives from the $n^{th}$ node. The transmission of the message to the $n^{th}$ node will not be denied by the $m^{th}$ node if the $n^{th}$ node verified $m^{th}$ node signature in

step 2 (4.2), and similarly the transmission of the message to the $m^{th}$ node will not be denied by the $n^{th}$ node if $m^{th}$ node verified $n^{th}$ node signature.

### 5.1.4 Integrity

Equation (4.2) can be used in the proposed scheme to check whether or not a cipher text $c$ was altered throughout the transmission. This equation is true unless an adversary changes $c$; in that case, it is false. Similar to this, if an enemy alters the message, it can be immediately discovered. The authentication will fail in both cases and the session will end. Therefore, the proposed method guarantees integrity.

### 5.1.5 Unforgeability

In the proposed IoT-CS method, adversary attempts to generate a valid signature. The private key pair $\{a_m, \delta_m\}$ of the $m^{th}$ node would be required for adversary to accomplish this. Adversary must solve the unfeasible HECDLP in order to compute the private keys. Thus, protection against unforgeability is provided by the proposed IoT-CS methods.

### 5.1.6 Security against Eavesdropping Attacks

The messages are sent in hashed text, plaintext and ciphertext formats in the proposed IoT-CS protocol. The plaintext messages give the enemy no advantages and do not contain any sensitive information. Additionally, HECDLP, a one-way hash function and encryption techniques are used to safeguard any message containing secret information, making it computationally impossible for an attacker to retrieve the information. Thus, eavesdropping attacks are prevented by the proposed IoT-CS method.

### 5.1.7 Security against Denial of Service (DoS) Attack

The participating nodes in the proposed IoT-CS system first confirm the validity of the time stamps they received. The messages are rejected if the time stamps are invalid. Additionally, the encrypted message always includes the most recent time stamp, and integrity checks (in the form of signatures) are added to the sent data. Thus, by effectively ending the session, the proposed approach may detect erroneous messages and prevent denial of service attacks.

### 5.1.8 Security against Man in the Middle (MITM) Attack

In Man in the Middle attack, an attacker tries to alter messages from the $m^{th}$ node to the $n^{th}$ node and vice versa. By pretending as an authorized participating entity, the adversary delivers the modified messages to either node. The scheme uses messages $\{c, c^{'}, W\}$ and authentication tags to perform mutual authentication. Adversary can only accurately display any of these messages in order to pretend to be a valid participant. HECDLP, however, states that it is not computationally possible to retrieve the private key. Therefore, the proposed scheme is easily resistant to MITM attacks.

## 5.2 Comparative Analysis

The comparative analysis of security features, communication overhead and computational cost is presented in this section.

### 5.2.1 Computational Cost

The rate at which various cryptographic procedures in an authentication scheme are executed, determines the computing overhead. According to Garg et al. [92], utilizing MIRACL (Multiprecision Integer and Rational Arithmetic C/C++ Library) [93], the time needed to complete hash-to-point (HtP) and elliptic curve

scalar multiplication (ECSM) operations is 14.293 ms and 0.986 ms, respectively. Hyperelliptic curve divisor multiplication (HECDM) has an estimated execution time of 0.48 ms [94]. When compared against the time requirements of ECSM and HECDM, cryptographic activities have a very tiny time consumption and can be neglected.

Each sender and receiver node in the proposed scheme carry out two HECDM operations. As a result, $4 \times 0.48 = 1.92$ ms is the total amount of time that the sender and recipient nodes consumed. To verify each other's identities, the KGC carries out three HECDM procedures for a minimum of two IoT nodes within the system. As a result, the KGC took $2 \times 0.48 = 0.96$ ms to complete. The KGC and nodes need a total of 0.96 ms + 1.92 ms, or 2.88 ms, to complete the mutual authentication process. Table 5.1 compare the IoT-CS scheme's computational cost to that of the existing schemes [92, 95] and [96]. The findings demonstrate that, in comparison to other existing schemes, the IoT-CS scheme has lower computing costs.

TABLE 5.1: Computational Cost Analysis

| **Schemes** | **KGC** | **IoT nodes** | **Total** |
|---|---|---|---|
| [95] | 1ECSM= 0.086ms | 3ECSM=2.958ms | 4ECSM=3.944ms |
| [92] | 6ECSM= 5.916ms | 3ECSM=2.958ms | 9ECSM=8.874ms |
| [96] | 4HtP= 57.172ms | 2ECSM+1HtP=16.265ms | 2ECSM+5HtP=73.437ms |
| Proposed | 2HCDM= 0.96ms | 4HCDM=1.92ms | 6HCDM=2.88ms |

### 5.2.2 Communication Overhead

The number of sent and received bits during authentication process by the participating IoT nodes can be used to compute communication overhead. The hash function, SHA-256, produced an output of 256 bits. The encryption technique uses 128-bit AES and produced a ciphertext of 128 bits. An IoT node must transmit two messages $\{a_n, \delta_n\}$ and $\{T_m, c, c^{'}, \delta\}$ and receive two messages $\{a_n, \delta_n\}$

and $\{T_n, Auth\}$ in the proposed IoT-CS scheme. Communication overhead of IoT nodes to convey the messages $\{a_n, \delta_n\}$ and $\{c^{'}, \delta\}$ is 160+80+80+128+256+80 = 784 bits. On the other hand, $160 + 80 + 80 + 128 = 448$ bits are required for an IoT node to receive the messages $\{a_n, \delta_n\}$ and $\{T_j, Auth\}$. An IoT node's total communication overhead is $784 + 448 = 1232$ bits. In Table 5.2 the comparison of the IoT-CS communication overhead with the current techniques [92, 95, 96] is given.

TABLE 5.2: Communication Overhead Analysis

| **Schemes** | **Sent (bits)** | **Received (bits)** | **Total (bits)** |
|---|---|---|---|
| [95] | 832 | 1536 | 2344 |
| [92] | 864 | 928 | 1792 |
| [96] | 1120 | 800 | 1920 |
| Proposed | 784 | 448 | 1232 |

## 5.3 Comparison of Security Attributes

The security of the proposed approach is compared to the current state-of-the-art schemes [92, 95, 96]. As indicated in Table 5.3, the proposed technique provides authentication, non-repudiation, unforgeability and security against eavesdropping, DoS and MITM attacks.

TABLE 5.3: Comparison of the Security Features

| **Schemes** | [95] | [92] | [96] | Proposed |
|---|---|---|---|---|
| **Authentication** | Yes | Yes | Yes | Yes |
| **Non-repudiation** | - | - | - | Yes |
| **Unforgeability** | No | No | No | Yes |
| **Eavesdropping attack** | - | Yes | - | Yes |
| **DoS attack** | Yes | Yes | No | Yes |
| **MITM attack** | Yes | Yes | No | Yes |

# Chapter 6

# Conclusion

In this thesis, the certificateless aggregate signcryption scheme based on ECC was reviewed, as proposed by Yu and Ren [34]. This very efficient technique satisfies various security features including availability, unforgeability, confidentiality, integrity and forward secrecy. The ECC based certificateless aggregate signcryption scheme (Section 3.1) is expanded to a HEC based scheme in this thesis. Instead of using 160-bit ECC for security and performance, the proposed approach makes use of 80-bit HEC. According to the analysis, the proposed method provides non-repudiation, confidentiality, mutual authentication, integrity, and resilience against various security threats like eavesdropping, DoS, impersonation, replay, MITM and key compromise attacks, among others.

Our proposed scheme as compared to the state-of-the-art is comparatively less costly. In terms of communication overhead and computing cost, our suggested approach is better than the most recent approaches by 31.25% and 51.31%, respectively. By using hyperelliptic curves, this new approach not only increases security but also improves efficiency by being faster and requiring lower key sizes. The approach is suitable for IoT devices with limited resources since it lowers the computational and communication complexity.

Our approach also incorporates formal security validation, which lowers the security hurdles and gives confidence in its practical execution. IoTs security and

efficiency is guaranteed by the well-balanced solution offered by the suggested methodology, which satisfies security and efficiency requirements.

Future work on the certificateless hyperelliptic curve aggregate signcryption scheme could explore several promising directions to enhance its applicability and security. Investigating the scheme's resilience against advanced cryptographic attacks, including quantum computing threats, could be a valuable area of research, ensuring its long-term viability in a post-quantum era. The scheme can also be extended to an identity-based signcryption framework within the setting of elliptic or hyperelliptic curves, simplifying key management while maintaining efficiency. Furthermore, the proposed scheme may be adapted into a blind signcryption scheme, which could be implemented in privacy-preserving applications such as electronic voting systems and electronic cash payment systems. Additionally, optimizing the scheme for resource-constrained IoT devices and evaluating its performance in real-world IoT deployments, such as smart cities and industrial automation, would provide practical insights and strengthen its potential for widespread adoption.

# Bibliography


[1] T. M. Damico, "A brief history of cryptography," *Inquiries Journal*, vol. 1, no. 11, 2009.

[2] O. Abraham and G. O. Shefiu, "An improved caesar cipher (icc) algorithm," 2012.

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

[4] S. Som, M. Kundu, and S. Ghosh, "A simple algebraic model based polyalphabetic substitution cipher," *International Journal of Computer Applications*, vol. 975, p. 8887, 2012.

[5] G. Brassard *et al.*, *Modern cryptology: A tutorial*. Springer, 1988, vol. 325.

[6] J. K. Grewal, "Elgamal: public-key cryptosystem," *Math and Computer Science Department, Indiana State University*, 2015.

[7] R. L. Rivest, "A method for obtaining digital signature and public-key cryptosystems," *ACM*, vol. 21, p. 2, 1987.

[8] 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.

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

[10] J. Katz and Y. Lindell, *Introduction to modern cryptography: principles and protocols*. Chapman and hall/CRC, 2007.

[11] J. F. Dooley, "History of cryptography and cryptanalysis," *History of Computing*, 2018.

[12] H. Delfs, H. Knebl, and H. Knebl, *Introduction to cryptography.* Springer, 2002, vol. 2.

[13] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption) 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.* Springer, 1997, pp. 165–179.

[14] 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.

[15] H. Y. Jung, K. Chang, D. Lee, and J. Lim, "Signcryption schemes with forward secrecy," *Proceeding of Information Security Application-WISA*, vol. 1, pp. 403–475, 2001.

[16] F. Bao and R. H. Deng, "A signcryption scheme with signature directly verifiable by public key," in *International workshop on public key cryptography.* Springer, 1998, pp. 55–59.

[17] C. Gamage, J. Leiwo, and Y. Zheng, "Encrypted message authentication by firewalls," in *Public Key Cryptography: Second International Workshop on Practice and Theory in Public Key Cryptography, PKC99 Kamakura, Japan, March 1–3, 1999 Proceedings 2.* Springer, 1999, pp. 69–81.

[18] Y. Han, X. Yang, and Y. Hu, "Signcryption based on elliptic curve and its multi-party schemes," in *Proceedings of the 3rd international conference on Information security*, 2004, pp. 216–217.

[19] R.-J. Hwang, C.-H. Lai, and F.-F. Su, "An efficient signcryption scheme with forward secrecy based on elliptic curve," *Applied Mathematics and computation*, vol. 167, no. 2, pp. 870–881, 2005.

[20] J. Y. Khan and M. R. Yuce, *Internet of Things (IoT): systems and applications*. CRC Press, 2019.

[21] E. de Matos, R. T. Tiburski, C. R. Moratelli, S. Johann Filho, L. A. Amaral, G. Ramachandran, B. Krishnamachari, and F. Hessel, "Context information sharing for the internet of things: A survey," *Computer Networks*, vol. 166, p. 106988, 2020.

[22] A. Čolaković and M. Hadžialić, "Internet of things (IoT): A review of enabling technologies, challenges, and open research issues," *Computer networks*, vol. 144, pp. 17–39, 2018.

[23] H. HaddadPajouh, A. Dehghantanha, R. M. Parizi, M. Aledhari, and H. Karimipour, "A survey on internet of things security: Requirements, challenges, and solutions," *Internet of Things*, vol. 14, p. 100129, 2021.

[24] S. Rizvi, R. Orr, A. Cox, P. Ashokkumar, and M. R. Rizvi, "Identifying the attack surface for IoT network," *Internet of Things*, vol. 9, p. 100162, 2020.

[25] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019.

[26] O. R. M. Boudia, S. M. Senouci, and M. Feham, "A novel secure aggregation scheme for wireless sensor networks using stateful public key cryptography," *Ad Hoc Networks*, vol. 32, pp. 98–113, 2015.

[27] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Advances in CryptologyEUROCRYPT 2003: International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4–8, 2003 Proceedings 22*. Springer, 2003, pp. 416–432.

[28] H. Yu, L. Bai, M. Hao, and N. Wang, "Certificateless signcryption scheme from lattice," *IEEE Systems Journal*, vol. 15, no. 2, pp. 2687–2695, 2020.

[29] J. Li, H. Yu, and Y. Xie, "Elgamal broadcasting multi-signcryption protocol with uc security," *J. Comput. Res. Develop.*, vol. 56, no. 5, pp. 1101–1111, 2019.

[30] E. Abouelkheir and S. El-sherbiny, "Pairing free identity based aggregate signcryption scheme," *IET Information Security*, vol. 14, no. 6, pp. 625–632, 2020.

[31] J. Song, Y. Liu, J. Shao, and C. Tang, "A dynamic membership data aggregation (dmda) protocol for smart grid," *IEEE Systems Journal*, vol. 14, no. 1, pp. 900–908, 2019.

[32] D. Wang and P. Wang, "Two birds with one stone: Two-factor authentication with security beyond conventional bound," *IEEE transactions on dependable and secure computing*, vol. 15, no. 4, pp. 708–722, 2016.

[33] H. Lu and Q. Xie, "An efficient certificateless aggregate signcryption scheme from pairings," in *2011 International Conference on Electronics, Communications and Control (ICECC)*. IEEE, 2011, pp. 132–135.

[34] H. Yu and R. Ren, "Certificateless elliptic curve aggregate signcryption scheme," *IEEE Systems Journal*, vol. 16, no. 2, pp. 2347–2354, 2021.

[35] X. Zhang *et al.*, "Certificateless aggregate signcryption scheme with public verifiability," *Journal of Computer Applications*, vol. 33, no. 07, p. 1858, 2013.

[36] Z. Eslami and N. Pakniat, "Certificateless aggregate signcryption: Security model and a concrete construction secure in the random oracle model," *Journal of King Saud University-Computer and Information Sciences*, vol. 26, no. 3, pp. 276–286, 2014.

[37] Y. Zhang, Y. Zhang, and C. Wang, "Certificateless aggregate signcryption scheme with internal security and const pairings," , vol. 40, no. 2, pp. 500–508, 2018.

[38] T.-H. Kim, G. Kumar, R. Saha, M. Alazab, W. J. Buchanan, M. K. Rai, G. Geetha, and R. Thomas, "Cascf: Certificateless aggregated signcryption

framework for internet-of-things infrastructure," *IEEE Access*, vol. 8, pp. 94 748–94 756, 2020.

[39] M. Elhoseny and K. Shankar, "Reliable data transmission model for mobile ad hoc network using signcryption technique," *IEEE transactions on reliability*, vol. 69, no. 3, pp. 1077–1086, 2019.

[40] S. A. Vanstone, "Elliptic curve cryptosystemthe answer to strong, fast public-key cryptography for securing constrained environments," *Information security technical report*, vol. 2, no. 2, pp. 78–87, 1997.

[41] M. Toorani and A. A. Beheshti, "An elliptic curve-based signcryption scheme with forward secrecy," *arXiv preprint arXiv:1005.1856*, 2010.

[42] H.-f. Yu and B. Yang, "Low-computation certificateless hybrid signcryption scheme," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 7, pp. 928–940, 2017.

[43] M. Zia and R. Ali, "Cryptanalysis and improvement of an elliptic curve based signcryption scheme for firewalls," *PloS one*, vol. 13, no. 12, p. e0208857, 2018.

[44] N. Koblitz, "Hyperelliptic cryptosystems," *Journal of cryptology*, vol. 1, pp. 139–150, 1989.

[45] H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, and F. Vercauteren, *Handbook of elliptic and hyperelliptic curve cryptography*. CRC press, 2005.

[46] J. Cao, P. Yu, M. Ma, and W. Gao, "Fast authentication and data transfer scheme for massive NB-IoT devices in 3GPP 5G network," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1561–1575, 2018.

[47] I. Ullah, N. Ul Amin, M. Zareei, A. Zeb, H. Khattak, A. Khan, and S. Goudarzi, "A lightweight and provable secured certificateless signcryption approach for crowdsourced IIoT applications," *Symmetry*, vol. 11, no. 11, p. 1386, 2019.

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

[49] J. J. Rotman, *Journey into mathematics: An introduction to proofs.* Courier Corporation, 2013.

[50] C. J. Benvenuto, "Galois field in cryptography," *University of Washington*, vol. 1, no. 1, pp. 1–11, 2012.

[51] K. S. McCurley, "The discrete logarithm problem," in *Proc. of Symp. in Applied Math*, vol. 42. USA, 1990, pp. 49–74.

[52] M. S. Iqbal, S. Singh, and A. Jaiswal, "Symmetric key cryptography: Technological developments in the field," *International Journal of Computer Applications*, vol. 117, no. 15, 2015.

[53] B. Schneier, "Description of a new variable-length key, 64-bit block cipher (blowfish)," in *International Workshop on Fast Software Encryption.* Springer, 1993, pp. 191–204.

[54] W. Diffie and M. E. Hellman, "New directions in cryptography," in *Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman*, 2022, pp. 365–390.

[55] 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.

[56] T.-C. Wu, *Information Security: 11th International Conference, ISC 2008, Taipei, Taiwan, September 15-18, 2008, Proceedings.* Springer Science & Business Media, 2008, vol. 5222.

[57] S. Bose and P. Vijaykumar, *Cryptography and network security.* Pearson Education India, 2016.

[58] M. England, "Elliptic curve cryptography," *M. Sc Applied Mathematical Science, Heriot-Watt University, Summer*, 2006.

[59] F. Mallouli, A. Hellal, N. S. Saeed, and F. A. Alzahrani, "A survey on cryptography: comparative study between RSA vs ECC algorithms, and RSA vs El-Gamal algorithms," in *2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*. IEEE, 2019, pp. 173–176.

[60] "Comparison of ECC and RSA algorithms in IoT devices, author=Vahdati, Zeinab and Yasin, Sharifah and Ghasempour, Ali and Salehi, Mohammad, journal=Journal of Theoretical and Applied Information Technology, volume=97, number=16, pages=4293, year=2019."

[61] R. Ahirwal, A. Jain, and Y. Jain, "Signcryption scheme that utilizes elliptic curve for both encryption and signature generation," *International Journal of Computer Applications*, vol. 62, no. 9, 2013.

[62] N. Thériault, "Index calculus attack for hyperelliptic curves of small genus," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2003, pp. 75–92.

[63] R. A. Asif, "Efficient computation for hyper elliptic curve based cryptography," 2016.

[64] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. CRC press, 2018.

[65] 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.

[66] G. K. Verma, B. Singh, and H. Singh, "Provably secure certificate-based proxy blind signature scheme from pairings," *Information Sciences*, vol. 468, pp. 1–13, 2018.

[67] P. P. Pittalia, "A comparative study of hash algorithms in cryptography," *International Journal of Computer Science and Mobile Computing*, vol. 8, no. 6, pp. 147–152, 2019.

[68] Y. Yuan, "Security analysis of an enhanced certificateless signcryption in the standard model," *Wireless Personal Communications*, vol. 112, pp. 387–394, 2020.

[69] S. Belguith, N. Kaaniche, M. Hammoudeh, and T. Dargahi, "Proud: Verifiable privacy-preserving outsourced attribute based signcryption supporting access policy update for cloud assisted IoT applications," *Future Generation Computer Systems*, vol. 111, pp. 899–918, 2020.

[70] A. Karati, C.-I. Fan, and R.-H. Hsu, "Provably secure and generalized signcryption with public verifiability for secure data transmission between resource-constrained IoT devices," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 431–10 440, 2019.

[71] J. Qiu, K. Fan, K. Zhang, Q. Pan, H. Li, and Y. Yang, "An efficient multi-message and multi-receiver signcryption scheme for heterogeneous smart mobile IoT," *IEEE Access*, vol. 7, pp. 180 205–180 217, 2019.

[72] V.-H. Hoang, E. Lehtihet, and Y. Ghamri-Doudane, "Password-based authenticated key exchange based on signcryption for the internet of things," in *2019 Wireless Days (WD)*. IEEE, 2019, pp. 1–8.

[73] E. Ahene, Z. Qin, A. K. Adusei, and F. Li, "Efficient signcryption with proxy re-encryption and its application in smart grid," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9722–9737, 2019.

[74] C. Peng, J. Chen, M. S. Obaidat, P. Vijayakumar, and D. He, "Efficient and provably secure multireceiver signcryption scheme for multicast communication in edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6056–6068, 2019.

[75] S. Mandal, B. Bera, A. K. Sutrala, A. K. Das, K.-K. R. Choo, and Y. Park, "Certificateless-signcryption-based three-factor user access control scheme for IoT environment," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3184–3197, 2020.

[76] X. Fan, T. Wu, Q. Zheng, Y. Chen, M. Alam, and X. Xiao, "Hse-voting: A secure high-efficiency electronic voting scheme based on homomorphic signcryption," *Future Generation Computer Systems*, vol. 111, pp. 754–762, 2020.

[77] S. Niu, Z. Li, and C. Wang, "Privacy-preserving multi-party aggregate signcryption for heterogeneous systems," in *Cloud Computing and Security: Third International Conference, ICCCS 2017, Nanjing, China, June 16-18, 2017, Revised Selected Papers, Part II 3*. Springer, 2017, pp. 216–229.

[78] G. Swapna and P. V. Reddy, "Efficient identity based aggregate signcryption scheme using bilinear pairings over elliptic curves," in *Journal of Physics: Conference Series*, vol. 1344, no. 1. IOP Publishing, 2019, p. 012010.

[79] W. Luo and W. Ma, "Secure and efficient data sharing scheme based on certificateless hybrid signcryption for cloud storage," *Electronics*, vol. 8, no. 5, p. 590, 2019.

[80] M. Zia Ullah Bashir and R. Ali, "A multi recipient aggregate signcryption scheme based on elliptic curve," *Wireless Personal Communications*, vol. 115, no. 2, pp. 1465–1480, 2020.

[81] N. C. Kumar, A. Basit, P. Singh, and V. C. Venkaiah, "Lightweight cryptography for distributed PKI based manets," *arXiv preprint arXiv:1804.06313*, 2018.

[82] S. Ullah, L. Marcenaro, and B. Rinner, "Secure smart cameras by aggregate-signcryption with decryption fairness for multi-receiver IoT applications," *Sensors*, vol. 19, no. 2, p. 327, 2019.

[83] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology: Proceedings of CRYPTO 84 4*. Springer, 1985, pp. 47–53.

[84] J. Malone-Lee, "Identity-based signcryption," *Cryptology ePrint Archive*, 2002.

[85] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *International conference on the theory and application of cryptology and information security*. Springer, 2003, pp. 452–473.

[86] M. Barbosa and P. Farshim, "Certificateless signcryption," in *Proceedings of the 2008 ACM symposium on Information, computer and communications security*, 2008, pp. 369–372.

[87] A. A. Omala, A. S. Mbandu, K. D. Mutiria, C. Jin, and F. Li, "Provably secure heterogeneous access control scheme for wireless body area network," *Journal of Medical Systems*, vol. 42, pp. 1–14, 2018.

[88] C. Tamizhselvan and V. Vijayalakshmi, "An energy efficient secure distributed naming service for IoT," *International Journal of Advanced Studies of Scientific Research*, vol. 3, no. 8, 2018.

[89] V. S. Naresh, R. Sivaranjani, and N. VES Murthy, "Provable secure lightweight hyper elliptic curve-based communication system for wireless sensor networks," *International Journal of Communication Systems*, vol. 31, no. 15, p. e3763, 2018.

[90] A. ur Rahman, I. Ullah, M. Naeem, R. Anwar, H. Khattak, S. Ullah *et al.*, "A lightweight multi-message and multi-receiver heterogeneous hybrid signcryption scheme based on hyper elliptic curve," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 5, 2018.

[91] U. Ali, M. Y. I. B. Idris, M. N. B. Ayub, I. Ullah, I. Ali, T. Nandy, M. Yahuza, and N. Khan, "RFID authentication scheme based on hyperelliptic curve signcryption," *IEEE Access*, vol. 9, pp. 49 942–49 959, 2021.

[92] S. Garg, K. Kaur, G. Kaddoum, and K.-K. R. Choo, "Toward secure and provable authentication for internet of things: Realizing industry 4.0," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4598–4606, 2019.

[93] S. Yu and Y. Park, "A robust authentication protocol for wireless medical sensor networks using blockchain and physically unclonable functions," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 20 214–20 228, 2022.

[94] M. A. Khan, I. Ullah, S. Nisar, F. Noor, I. M. Qureshi, F. U. Khanzada, and N. U. Amin, "An efficient and provably secure ateless key-encapsulated signcryption scheme for flying ad-hoc network," *IEEE Access*, vol. 8, pp. 36 807–36 828, 2020.

[95] A. Braeken, "PUF based authentication protocol for IoT," *Symmetry*, vol. 10, no. 8, p. 352, 2018.

[96] U. Chatterjee, R. S. Chakraborty, and D. Mukhopadhyay, "A PUF-based secure communication protocol for IoT," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 3, pp. 1–25, 2017.