CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD


# Genomic Data Protection using Hyperelliptic Curve Cryptography and Blockchain Technology

by

Ammad Jahangir

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

## Genomic Data Protection using Hyperelliptic Curve Cryptography and Blockchain Technology

by

Ammad Jahangir

(MMT223008)

### 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
September, 2025

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

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

# *Author's Declaration*

I, **Ammad Jahangir** hereby state that my MPhil thesis titled "**Genomic Data Protection using Hyperelliptic Curve Cryptography and Blockchain Technology** " is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/abroad.

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

**(Ammad Jahangir)**

Registration No: MMT223008

# *Plagiarism Undertaking*

I solemnly declare that research work presented in this thesis titled "**Genomic Data Protection using Hyperelliptic Curve Cryptography and Blockchain Technology**" is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged and that complete thesis has been written by me.

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

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

**(Ammad Jahangir )**

Registration No: MMT223008

# *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. Dure-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.

**(Ammad Jahangir)**
Registration No: MMT223008

# *Abstract*

Protecting genomic data is a critical requirement in Healthcare 4.0, particularly for applications such as microarray gene profile analysis used in early cancer detection. However, Healthcare 4.0 standards pose challenges for secure and efficient data processing and storage. While several blockchain-based solutions have been proposed, many lack essential features such as scalability, computational efficiency, and strong security. To address these issues, this work introduces a Healthcare 4.0 architecture that integrates blockchain with lightweight cryptography. The system incorporates edge devices, fog computing, cloud storage, and blockchain to ensure secure data exchange. Hyperelliptic Curve Cryptography (HECC) replaces traditional ECC to provide stronger security with smaller key sizes. Case studies demonstrate that the proposed model delivers improved efficiency, scalability, and resistance against various security threats. The security of the extended scheme relies on the hardness of the hyperelliptic curve discrete logarithm problem.

# Contents

**Author's Declaration** iv

**Plagiarism Undertaking** v

**Acknowledgements** vi

**Abstract** vii

**List of Figures** xi

**List of Tables** xii

**Abbreviations** xiii

**Symbols** xiv

**1 Introduction** 1
- 1.1 Literature Review . . . 2
- 1.2 Thesis Contribution . . . 7
- 1.3 Thesis Layout . . . 8

**2 Preliminaries** 9
- 2.1 Mathematical Background . . . 9
- 2.2 Cryptographic Background . . . 13
  - 2.2.1 Cryptography . . . 14
  - 2.2.2 Symmetric Key Cryptography . . . 15
  - 2.2.3 Asymmetric Key Cryptography . . . 16
- 2.3 Cryptanalysis . . . 16
  - 2.3.1 Cryptography Attacks . . . 17
    - 2.3.1.1 Brute Force Attack . . . 17
    - 2.3.1.2 Ciphertext Only Attack . . . 17
    - 2.3.1.3 Chosen Ciphertexts Attack . . . 17
    - 2.3.1.4 Chosen Plaintext Attack . . . 17
    - 2.3.1.5 Known Plaintext Attack . . . 18
    - 2.3.1.6 Man-in-the-Middle Attack . . . 18
    - 2.3.1.7 Forgery Attack . . . 18

2.4 ECC-based Cryptography . . . 18
2.4.1 Weierstrass Equation . . . 19
2.4.2 Elliptic Curve over $\mathbb{F}_p$ . . . 19
2.4.2.1 Point Addition . . . 20
2.4.2.2 Point Doubling . . . 20
2.4.2.3 Point at Infinity . . . 20
2.4.2.4 Mathematical Representation . . . 21
2.4.3 Elliptic Curve Discrete Logarithm Problem . . . 24
2.4.4 Elliptic Curve Diffie-Hellman Key Exchange Protocol . . . 24
2.5 Eliptic Curve Encryption and Decryption . . . 25
2.5.1 Global Setting . . . 25
2.5.2 Key Generation Phase . . . 26
2.5.3 Encryption Phase . . . 26
2.5.4 Decryption Phase . . . 26
2.6 Digital Signature . . . 27
2.7 Elliptic Curve Digital Signature Algorithim (ECDSA) . . . 28
2.7.1 Use of ECDSA Algorithim . . . 29
2.7.1.1 Domain Parameter Validation . . . 29
2.7.1.2 Digital Signature ECDSA Generation . . . 30
2.7.1.3 Digital Signature ECDSA Verification . . . 31
2.8 Hyperelliptic Curve . . . 32
2.9 HealthCare 4.0 . . . 32
2.9.1 Smart Devices Network . . . 33
2.9.2 Cloud . . . 33
2.9.3 Blockchain . . . 34
2.9.4 IoT Device (IN) . . . 34
2.9.5 Medical user (MU) . . . 34
2.9.6 Fog node (FN) . . . 35
2.9.7 Cloud storage (CS) . . . 35
2.9.8 Private blockchain (PB) . . . 35

**3 Secure Gene Profile Data Processing Using Lightweight Cryptography and Blockchain 36**
3.1 Hybrid Technique . . . 36
3.2 The Proposed Encryption Scheme . . . 40
3.2.1 Global parameters . . . 41
3.2.2 Notations . . . 41
3.3 Secuirty Analysis . . . 45
3.3.1 Confidentiality . . . 46
3.3.2 Integrity . . . 46
3.3.3 Authentication . . . 46
3.3.4 Scalability . . . 46

**4 Secure Gene Data Storage using Hyperelliptic Curve Cryptography and Blockchain Technology 47**

4.1 Hyperelliptic Curves . . . 48
4.1.1 Genus of a Curve . . . 49
4.1.2 Ordinary, Special and Opposite point . . . 49
4.1.3 Divisor . . . 50
4.1.4 Mumford Representation . . . 51
4.1.5 Hyperelliptic Curve Discrete Logarithm Problem . . . 54
4.2 Proposed Scheme . . . 54
4.2.1 Global paramters . . . 55
4.2.2 Notations . . . 55
4.3 Secuirty analysis . . . 57
4.3.1 Confidentiality . . . 58
4.3.2 Integrity . . . 58
4.3.3 Authentication . . . 58
4.3.4 Scalability . . . 58
4.3.5 Computational Efficiency . . . 59
4.3.6 Attack Resistance . . . 59

**5 Conclusion** **60**

**Bibliography** **62**

# List of Figures


2.1 Hash Function . . . . . 13
2.2 Cryptology . . . . . 14
2.3 Cryptosystem . . . . . 15
2.4 ECDSA signature and verification steps . . . . . 31

3.1 ECC and ECDH-based hybrid encryption process for enhanced security . . . . . 37
3.2 Hybrid Decryption Procedure Using ECDH and ECC . . . . . 38
3.3 Proposed secured gene data storage . . . . . 39
3.4 Proposed technique of gene data searching . . . . . 45

4.1 Geometrical Representation of Divisor . . . . . 51

# List of Tables


2.1 Addition in $GF(13)$ . . . . . 12
2.2 Multiplication in $GF(13)$ . . . . . 12
2.3 Points of elliptic curve . . . . . 22
2.4 Addition of points of $E_{F_{13}}(7, 4)$ . . . . . 24

3.1 Security bits and encryption key sizes . . . . . 40
3.2 Symbols used in the Scheme . . . . . 41

4.1 Symbols Used in the Proposed Scheme . . . . . 55

# Abbreviations

| | |
|---|---|
| **AES** | Advance Encryption Standard |
| **CS** | Cloud Storage |
| **DES** | Data Encryption Standard |
| **DLP** | Discrete Logarithm Problem |
| **ECC** | Eliptic Curve Cryptography |
| **ECDH** | Elliptic Curve Diffie-Hellman |
| **HECC** | Hyperelliptic Curve Cryptography |
| **HECDLP** | Hyperelliptic Curve Discrete Logarithm Problem |
| **IoT** | Internet of Things |
| **PB** | Private Blockchain |
| **RSA** | Rivest Shamir Adleman |

# Symbols

| | |
|---|---|
| $D$ | Divisor of Hyperelliptic Curve |
| $E$ | Elliptic Curve |
| $H$ | Hyperelliptic Curve |
| $mod$ | Modular Operator |
| $o$ | Order of curve point $G$ |
| $\mathcal{O}$ | Point at infinity |
| $p$ | Prime Number |
| $\mathbb{F}_p$ | Finite Field |
| $Z$ | Set of Integers |

# Chapter 1

# Introduction

Cryptography [38] is the study of the transmission of a message in a manner that prevents unauthorized parties from reading it. It is a method of safeguarding data or information from attackers by using mathematical functions. The original communication, known as plaintext, is converted into a ciphertext for public network transmission using an encryption method [32]. The recipient or another authorized person then uses the decryption procedure to transform the ciphertext back to plaintext. Both the sender and the recipient employ secret information which is only known to them for encoding and decoding. The term "key" refers to this exclusive information. Entire system is referred to as a cryptosystem. The secret key determines the cryptosystem's level of security.

The cryptographic technique comes into one of the two main categories of cryptography, which are symmetric key and asymmetric key. In "symmetric key" encryption only the sender and the receiver are aware of the single key that is used. Examples of symmetric key encryption are DES [1] and AES [26]. Key delivery between the sender and the recipient is the primary concern with this technique.

Assigning the code to each participant becomes extremely difficult when there are thousands of users connecting with one another. To solve this issue, Diffie and Helman [7] developed the idea of asymmetric key cryptography, also referred to as public key cryptography. Participants in public key cryptography possess two

types of encryption keys: a private key and a public key that is made public. Examples of "asymmetric key" cryptography are RSA [24] and ELGamal [37].

A digital signature is the electronic counterpart of a person's physical signature and is an authentication method that enables the sender of a message to include a code that serves as a signature [23]. Digital signatures consist of three algorithms: key generation, generation of signatures, and verification of the signatures. In order to guard against forgeries and preserve the confidentiality of a letter's contents, it has been customary for decades for the person who created the message to sign it, seal it in an envelope, and then give it to a deliverer. Since 1976, public key cryptography has completely changed how people conduct secure and verified interactions. Through both open and inaccessible networks, like the internet, people who have never interacted before can now safely and authentically communicate with one another. The recipient public key is then used to lock (encrypt) the random message key. We refer to two-step process as signature-then-encryption. The number of bits and computational complexity are the drawbacks of encryption compared to signatures.

## 1.1 Literature Review

Healthcare 4.0 [17] is a term inspired by Industry 4.0. Both terms refer to the use of modern technologies, like the "Internet of Things (IoT)", to improve healthcare and industries. Researchers have placed significant emphasis on developing ITS, advanced health tracking, smart home automation, surveillance systems along with IoT-powered intelligent city applications [18, 30]. The cloud service providers (CSPs) are essential to healthcare 4.0.

The fundamental cloud-based approach to sharing and keeping medical records among many providers, assisting each in managing their data, offering a dependable means of communication, and maybe providing a unified picture of each persistent's human services records and ensuring data consistency between Personal Health Records (PHR) and Electronic Health Records (EHRs). Patients can save their health information, like genetic data, in the cloud and keep it in

PHRs. Healthcare providers can use Electrronic medical records (EMRs) to get a patient's reports from the cloud. With EHRs, which are stored in the cloud, both patients and clinics can access medical history from anywhere, no matter where they are located.

Healthcare 4.0-based monitoring systems will store genetic data and share it via cloud service providers. However, CSP face hurdles when exchanging this information because there are risks and security threats involved in making the information public. Data owners and administrators are worried about careless or malicious users putting sensitive information at risk of being exposed or compromised [9, 10, 21, 46]. For the past 20 years, many encryption methods have been used to protect data in the cloud. However, with the rise of medical applications and Healthcare 4.0, these methods are no longer effective or good enough [52]. Traditional cryptography has raised concerns about trusting cloud servers and protecting data privacy. It also prevents users from searching through encrypted data, leading to a poor user experience. To fix this, searchable encryption methods have been created for both symmetric-key and public-key systems to safely store, share, and search gene data. Although symmetric-key systems are faster and more efficient than public-key ones, patients and specialists face hurdles when searching for keywords in encrypted genetic data. The blockchain approach has proven to be a successful substitute for conventional cryptographic techniques in cloud data processing, offering higher security and computational performance. Although there have been some recent attempts to use blockchain technology for secure data processing for CPS-related e-healthcare applications, their reach has been limited and they lack adequate research. Blockchain technology enables to create an open and distributed online database utilizing a chain of data structures known as obstructs. These bricks are transported between the hubs of a foundation. A timestamp for production, cryptographic hash of the preceding block, sharing data, patient data, and healthcare provider data are all included in each block. Blockchain technology comes in public, private, and federated forms. In order to offer thorough, scalable, and reliable data security and privacy preservation, A framework for processing gene profile data with the help of Healthcare 4.0 is introduced, utilizing blockchain and cloud service provider. A thorough

framework for patient privacy and data protection was created and methodically implemented [20, 34]. The suggested framework integrates four distinct layers: Edge Layer, Cloud Storage Layer, Fog Computing Layer, and Blockchain Layer. The suggested architecture incorporates lightweight cryptography techniques for gene data encryption and decryption. The meta-data of periodically created gene data is kept on the distributed blockchain network to guard against risks including forgery, manipulation, and quantum attacks. The gene data storage and search method provided privacy preservation and data security protections with the ability to audit meta-data.

In the healthcare sector, blockchain main goal is to safeguard patient data from various threats. With the growing role of blockchain in smart healthcare, there is not much research yet on how to securely process medical data while working alongside CSPs. In [29] the use of blockchain technology in healthcare for social insurance programs has been covered by writers. They interfaced with insurance systems and secured medical records using blockchain technology. They came to the conclusion that was not solely to blame for the current problems with medical records.

The initial combined approach for combining blockchain and cloud service providers CSPs had been introduced in [5]. This method secures the management of medical data on cloud servers by utilizing blockchain technology. The approach suggests encrypting the medical data using easily accessible cryptographic algorithms in order to safeguard Electronic Health Records (EHRs). The blockchain then securely stores the encrypted data. In order to securely store and transmit medical data in CSP, the blockchain technique was developed to construct a linked blockchain that connects patient networks, hospitals, health agencies, and medical services.

The use of CSPs in smart healthcare raises important concerns about data security. Researching how blockchain can improve the security and privacy of data in CSPs is a challenging task. This problem has been addressed lately in [15]. The blockchain-based data-sharing solution in [48] addresses the challenges of controlling access to cloud-stored medical data by leveraging the blockchain's inherent immutability and secure features.

To manage shared data pools, strong cryptographic methods were used in conjunction with a permissioned blockchain. In [47] MeDShare addresses the problem of pharmaceutical big-data brokers disclosing health information in untrustworthy ways. The solution manages access control for shared data in cloud containers, tracks data provenance, and ensures data audits using blockchain technology. A third-party framework is no longer required according to the TKSE proposal in [35], utilizing blockchain alongside CSPs for secure file sharing and storage solutions. The encrypted data is indexed with a digital signature, allowing end-users to access the encrypted content securely. In [5], the authors propose a framework for storing personal medical data using both cloud and blockchain technologies. They outline a method for securely exchanging medical data and analyze how blockchain can enhance traditional medical data methods. In [42], the authors solve privacy problems in Personal Health Records (PHRs) by using access control algorithms based on "blockchain technology". They utilize proxy re-encryption and pairing-based cryptographic techniques to ensure the secure transmission and storage of medical data. The system offers features such as fine-grained access control, the ability to revoke consent, tamper resistance, and audit capabilities. Finally, MedChain, a blockchain-based platform for sharing and storing medical data, is introduced in [35]. The authors developed features like blockchain combining and digest chains to enhance the security and functionality of medical data management. To make sure medical data from IoT devices is correct, a method called chain digest construction is used. For security tasks, a lightweight version of Elliptic Curve Cryptography (ECC) is used. However, there are some problems, such as the need to enter data manually and issues related to servers.

In [49] a new model for EHR has been proposed that uses traditional methods for storing data and keeping it secure. They developed a blockchain-based approach to enhance system interoperability and address medical data integrity. The blocks were manufactured using a novel incentive system. However, this method was unable to perform tasks including searching, distributing, and storing medical data under a variety of risks. Additionally, they employed cryptography techniques that were computationally inefficient. Smart healthcare systems that integrate blockchain and cloud computing face significant security challenges, and robust

solutions are still evolving. Blockchain is being used more in healthcare with new technologies like Healthcare 4.0. But there is not enough research on how it works in healthcare. This study offers a better solution because the current ones have many problems. The challenges with current methods are:

1. Lack of Generality: There is limited use of blockchain for data storage, auditing, sharing, and searching across Cloud-based Processing Systems (CPSs).

2. Inefficient Cryptography: Many of the current methods use traditional cryptography techniques, which require more time and resources for processing gene data. Using lightweight cryptography could improve the performance of secure gene data processing.

3. Limited Risk Protection: The current solutions mainly protect against issues like cloud server misbehavior, end-user misbehavior, and various types of attacks (e.g., tampering, unauthorized access, etc.). However, they do not address newer risks like those from quantum computing.

4. Blockchain Tool Dependence: Existing systems rely on specific blockchain types for gene data. These systems need to be more dynamic and flexible, independent of the blockchain used for gene data processing.

5. Lack of Experimental Validation: Current experimental assessments have not fully proven the efficiency, scalability, and security of smart healthcare systems.

Smart hospitals have adopted a reliable and scalable system to protect security and privacy. This system allows safe storage, sharing, and searching of gene data using blockchain, cloud computing, and fog nodes, while using very little space and time. To save time and space when using cryptography, lightweight ECC (Elliptic Curve Cryptography) algorithms were created [27]. These are used to store and share gene data over blockchain and cloud systems. A lightweight digital signature algorithm was also developed using ECDSA (Elliptic Curve Digital Signature Algorithm). Utilizing blockchain technology to protect against hazards of forging,

manipulation, and quantum. The blockchain layer is created, which holds the metadata retrieved from the cloud computing layer and connects to cloud storage.

In 1989, Koblitz [12] suggested creating encryption systems using hyperelliptic curves instead of regular elliptic curves, relying on DLP(Discrete Logarithm Problem). Hyperelliptic curves [6] are extended versions of elliptic curves. Their main advantage is that they use smaller key sizes. Compared to elliptic curves, hyperelliptic curves need a smaller finite field to reach the same level of security. Hyperelliptic curves of genus 2 are explained in Chapter 2.

## 1.2 Thesis Contribution

In this thesis, "Secure gene profile data processing using lightweight cryptography and blockchain" proposed by Mahajan and Reddy [17] is reviewed. They proposed the scheme by merging lightweight cryptography and blockchain technologies [49]. Smart healthcare applications require a centralized system to safely store and exchange sensitive data. Healthcare 4.0 is built on four main components: the edge layer (which includes patients or gene data users), fog nodes, cloud storage, and blockchain. Fog nodes handle the storage and searching of gene data from the edge layer within cloud storage that is secured using blockchain and lightweight cryptography. ECC is used to protect data privacy and ensure secure processing, with ECDH for key exchange and ECDSA for digital signatures. To prevent tampering, forgery, and future quantum attacks, metadata from the cloud is stored on the blockchain. Compared to other cryptosystems, this scheme provides several security features such as data integrity, forward secrecy, message confidentiality, authenticity, unforgeability, verification, and non-repudiation. It also offers faster processing speeds and uses shorter key lengths, making it more efficient.

The reviewed scheme is improved by using hyperelliptic curves, which makes it more secure, lightweight and better protected against hacking or cryptographic attacks. The same security level as elliptic curves is achieved by using hyperelliptic curves in extended schemes, but with a lower key size that is faster and requires less computational cost.

## 1.3 Thesis Layout

Rest of the thesis is organised as follows :

**Chapter 2**, presents mathematical background that is related to our scheme, including basic definitions of cryptography. Additionally, the section covers different types of attacks that are used in our scheme.

**Chapter 3**, covers a detailed analysis of securing gene profile data using lightweight cryptography and blockchain [17]. A detailed study of the scheme's security is also provided.

In **Chapter 4**, scheme presented in Chapter 3 is modified with the use of hyperelliptic curve, and covers the security analysis of the proposed scheme.

**Chapter 5**, provides the overall conclusion.

# Chapter 2

# Preliminaries

The purpose of this chapter is to present some fundamental definitions and cryptographic foundations from algebra and number theory that are necessary for a proper understanding of the work carried out in this thesis. The concepts related to elliptic curve cryptography will also be discussed in the subsequent sections.

## 2.1 Mathematical Background

In this part we examine some mathematical background from algebra which are important for a good understanding of the chapters in this thesis.

### 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 the *inverse* of $a$.

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

$$a * b = b * a$$

for all $a, b \in G$ [38]."

### Definition 2.1.2. Cyclic Groups

"A group $G$ is cyclic if every element of $G$ is a power $x^k$ (where $k$ is an integer) of a fixed element $x \in G$. The element x is said to generate the group $G$, or to be a generator of G. [25]"

**Example 2.1.3.** The following are examples of a group and a cyclic group:

1. "The set of real numbers $\mathbb{R}$, complex numbers $\mathbb{C}$, and integers $\mathbb{Z}$ are all groups under addition $(+)$."
2. Under multiplication, the set of integers $\mathbb{Z}$ is not a group.

### Definition 2.1.4. Field

"A nonempty set $(F, +, \cdot)$ together with binary operations '$+$' and '$\cdot$' is called a *field* $F$ if the following properties hold:

1. $F$ is abelian group, under addition.
2. $F \setminus \{0\}$ is an abelian group under multiplication.
3. The left and right distributive laws hold in $F$ [50]."

**Example 2.1.5.** Some examples of fields are given below:

1. The sets $\mathbb{R}$ for real numbers and $\mathbb{C}$ for complex numbers are fields.

2. The multiplicative inverse does not hold in $\mathbb{Z}$, hence the set of integers $\mathbb{Z}$ is not a field.

### Definition 2.1.6. Order

A *field* order is determined by how many elements it contains. For example, if a field has 4 elements, then its order is 4.

### Definition 2.1.7. Finite or Galois Field

"For every prime $p$ and positive integer $n$, there exists exactly one finite field of order $p^n$. This field, denoted by GF($p^n$), is usually referred to as the *Galois field* of order $p^n$ [2]."

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

### Algorithm 2.1.8. Extended Euclidean Algorithm

To find the inverse of $b$ under modulo $n$, below mentioned steps are to be followed:

**Input:** $b$ **mod** $n$

**Output:** $b^{-1}$ **mod n**

1. Set $(x, y, z) = (1, 0, n)$ and $(A, B, C) = (0, 1, b)$

2. If C = 0, return $gcd(b, n) = z$, There is no inverse for b mod n.

3. If C = 1, return $gcd(b, n) = C$ and $B = b^{-1} mod n$

4. Now find quotient T using: $T = z$ divided by $C$

5. $(S, P, D) = (x - TA, y - TB, z - TC)$

6. $(A, B, C) = (x, y, z)$

7. $(x, y, z) = (S, P, D)$

8. Go to step 2

The addition and multiplication procedures for the fields $GF(13)$ are displayed in Tables 2.1 and 2.2, respectively.

TABLE 2.1: Addition in $GF(13)$

| + | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 0 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 0 | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 0 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 0 | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 0 | 1 | 2 | 3 | 4 | 5 |
| 6 | 7 | 8 | 9 | 10 | 11 | 12 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 7 | 8 | 9 | 10 | 11 | 12 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 11 | 12 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 12 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

TABLE 2.2: Multiplication in $GF(13)$

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

In Table 2.1 the elements which give 0 are additive inverses of each other and in Table 2.2 the elements which give 1 are multiplicative inverses of each other.

## Definition 2.1.9. Hash Function

A one-way function is easy to compute in the forward direction but computationally difficult to invert. In other words, given $x$, it is simple to calculate $f(x)$, but given $f(x)$, it is difficult to determine x. A one-way hash function is a special type

of one-way function that maps data of arbitrary length to a fixed-length output. The result of this process is called a hash value. Hash values are easy to compute but practically infeasible to reverse [22].

There are some properties of hash function are as follows:

1. It is easy to compute $H(t)$, where $t$ is the message.
2. Given $H(t)$, it is computationally infeasible to determine $t$.
3. Given an output (hash value), it is computationally infeasible to find a matching input.
4. It is computationally infeasible to find two distinct inputs $t_1$ and $t_2$ such that $H(t_1) = H(t_2)$.

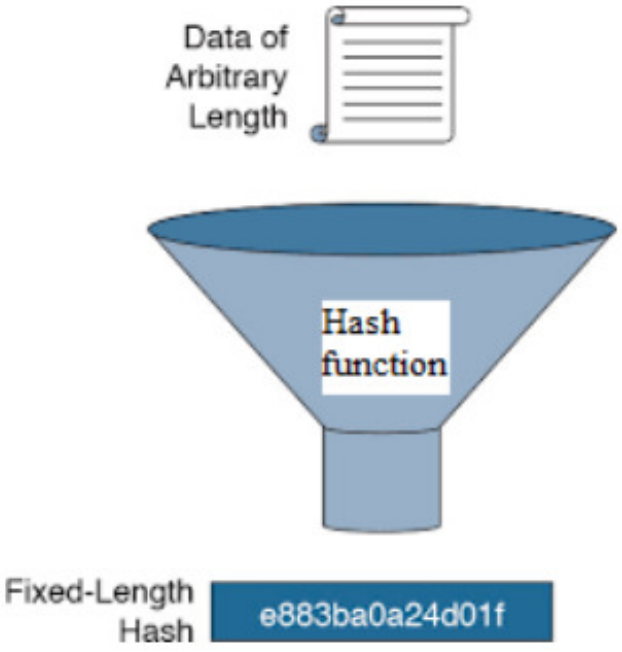


FIGURE 2.1: Hash Function

## 2.2 Cryptographic Background

The Greek terms kryptos (hidden) and logos (words) are combined to form the word "cryptology". Cryptography and cryptanalysis are its two more varieties.

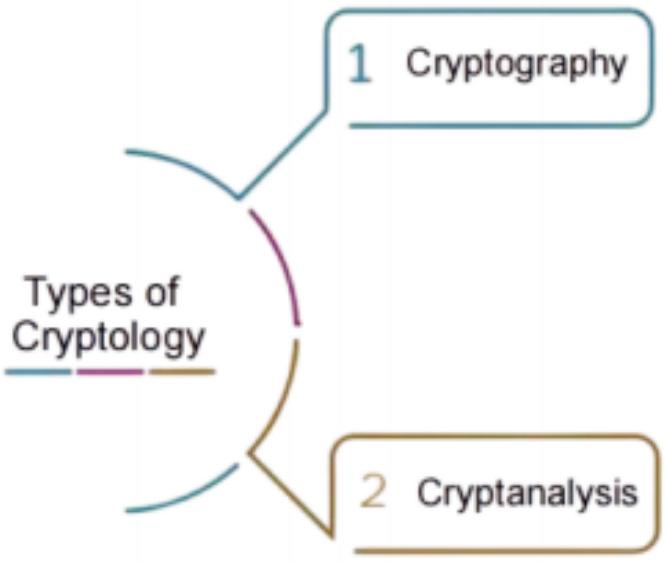


FIGURE 2.2: Cryptology

### 2.2.1 Cryptography

Cryptography is the science of keeping messages private, even when they are sent over public networks. It works by turning messages into secret code using math. The original message is called plaintext or cleartext. Encryption is the process of converting a communication into a secret code, so that others cannot read it.

To read the original message again, the secret code is changed back into normal text, this process is called decryption.

A cryptosystem includes plaintext, ciphertext, encryption algorithms, decryption algorithms, and a key called $k$, which is used by the communicating parties during the encryption and decryption processes. In the past, people used cryptography to keep written messages private through simple encryption methods.

These same principles are now used to secure the flow of data between computers or to encrypt television signals.

Modern cryptography, which relies on mathematics, has many important uses such as generating random numbers, ensuring data integrity, creating electronic signatures, enabling electronic voting and digital money, securing key exchanges, and much more.

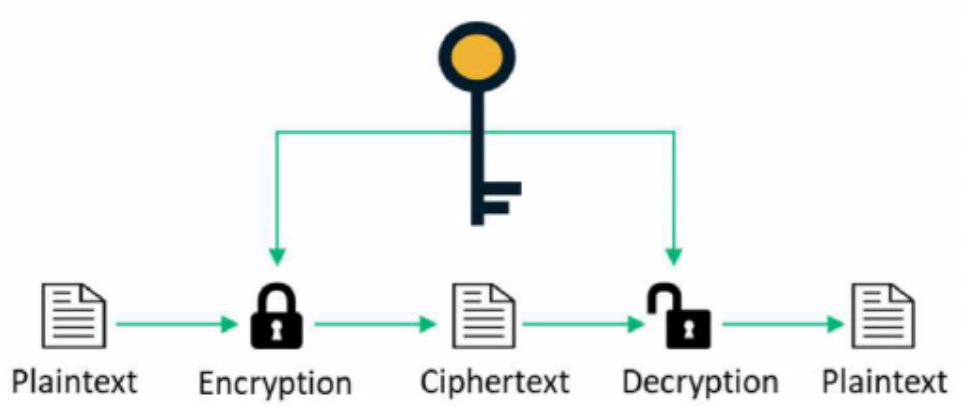


FIGURE 2.3: Cryptosystem

Cryptography is divided into two primary groups according to key distribution. Symmetric key cryptography (Secret key cryptography)and Asymmetric key cryptography (Public key cryptography.

### 2.2.2 Symmetric Key Cryptography

Symmetric key Cryptography encrypts and decrypts data using a single key. In symmetrical encryption, a secret key is a collection of data that is used to encrypt and decrypt data.

It is frequently called a private key.It can be sent between two parties over a secure channel. Both parties interchange the key before the transmission of information.

Given a message (cleartext) and the key, encoding procedure produces unreadable data, which is exactly equal to the length as the cleartext was[41]. The process of decryption is the opposite of that of encryption.

Symmetric key cryptography is used to secure information between two parties using particular shared private key. It is used for protecting sensitive information.

Two notable methods of "symmetric key cryptography" are:

1. DES : Data Encryption Standard [43].

2. AES : Advanced Encryption Standard [16].

### 2.2.3 Asymmetric Key Cryptography

Asymmetric Key Cryptography was presented by the Diffie and Hellman in 1976 to solve the key exchange problem. A public key is used for encryption in asymmetric cryptography, and a private key is used for decryption.

The private key is kept secret, whereas the public key is known to all. If encryption is done by one key then the decryption process will be done by the other key. Only the individual with the private key may decode the ciphered communication.

Some of the asymmetric cryptography techniques are as follows:

1. Digital Signature Algorithm (DSA) [13].
2. Rivest-Shamir-Adleman (RSA) [14].
3. ElGamal Cryptosystem [8].
4. Elliptic Curve Cryptosystem (ECC) [27].

## 2.3 Cryptanalysis

Cryptanalysis is a technique used to crack the cryptosystem in order to extract plaintext.

Additionally, it is being examined to confirm the stability and effectiveness of a cryptosystem.

A cryptanalyst is the person who performs cryptanalysis. Cryptanalysis became necessary if a cryptosystem lacks any of the following characteristics.

1. Integrity
2. Confidentiality
3. Authentication
4. Non-repudiation

### 2.3.1 Cryptography Attacks

There are several kinds of attacks, some of them are listed below:

#### 2.3.1.1 Brute Force Attack

The ciphertext and the decryption algorithm are known to the attacker in this attack. Using each key from the set of all possible keys, the attacker tries to extract the plaintext from the ciphertext. This attack takes a long time to complete because the attacker must look for every key in the key space.

This attack is impossible if it is not possible to try every key in a fair amount of time. In other words, the key space should be sufficiently large to withstand such an attack [39].

#### 2.3.1.2 Ciphertext Only Attack

In this attack, the attacker just knows the ciphertext. Most of the time, the accompanying plaintexts are unknown. He obtains the corresponding plaintexts by using these well-known ciphertexts [45].

#### 2.3.1.3 Chosen Ciphertexts Attack

In this attack, the attacker attempts to gain plaintext by decrypting some ciphertexts that they have access to. Using this information, he might attempt to obtain the key or the plaintexts of other ciphertexts [31].

#### 2.3.1.4 Chosen Plaintext Attack

In this attack , the attacker knows the plaintext and the corresponding ciphertext through which he tries to guess the key or obtains as much as information possible [44].

#### 2.3.1.5 Known Plaintext Attack

When the attacker has both the plaintext and the matching ciphertext, it is referred to as a known plaintext attack. The attacker attempts to crack the cryptosystem by obtaining additional information from the pair [44].

#### 2.3.1.6 Man-in-the-Middle Attack

Two parties when try to agree on a key for safe communication. In order to agree on a key without understanding them, the attacker positions himself between them. The attacker chooses two keys to deceive the two groups. He uses one of the keys by pretending to be the second party to make the first party agree on the exchange of information. To misguide the second group, the other key is used. In fact , the two sides assume that they are communicating with each other, but it is the attacker who gets the data from both ends and then attacks the conversation [3].

#### 2.3.1.7 Forgery Attack

In this attack, the attacker attempted to forge a signature for the message without knowing the signer's secret signing key. The term "forgery" typically refers to an attack that makes use of digital communications and signatures [51].

## 2.4 ECC-based Cryptography

ECC , a public-key encryption method, is founded on the mathematical concept of elliptic curves over finite fields. ECC ability to offer the same level of safety has led to its steady rise in popularity over the last few decades. This trend is likely to continue as the need for safe devices grows as a result of larger keys, which use less mobile resources. This is why it is crucial to comprehend ECC in its context. Because it is more stable than RSA, As a next-generation method of public key cryptography, ECC provides a very effective and safe solution.

The major advantage of the use of elliptic curves is that the same security level can be achieved by working in a field of 160 bits and hence elliptic curve solves the problem of computational complexity to achieve the desired security extent. A detailed explanation of elliptic curves, including their mathematical foundation and importance in cryptographic systems, will be provided in next section.

### 2.4.1 Weierstrass Equation

The equation of the form:

$$y^2 + z_1xy + z_3y = x^3 + z_2x^2 + z_4x + z_5 \tag{2.1}$$

defined over a some field $\mathbb{F}$ , known as Weierstrass equation [36].

Where $z_1, z_2, z_3, z_4, z_5, z_6$ are called *Weierstrass coefficients*. This curve is called a smooth curve if the discriminant is

$$-v_2^2v_8 - 8v_4^3 - 27v_6^2 + 9v_2v_4v_6^6 \neq 0$$

where

$$v_2 = u_1^2 + 4z_2 \tag{2.2}$$

$$v_4 = 2l_4 + z_1z_3 \tag{2.3}$$

$$v_6 = z_3^2 + 4z_6 \tag{2.4}$$

$$v_8 = z_1^2z_6 + 4z_2z_6 - z_1z_3z_4 + z_2z_3^2 - z_4^2 \tag{2.5}$$

### 2.4.2 Elliptic Curve over $\mathbb{F}_p$

Our primary focus in cryptography is on the simplified version of the Weierstrass equation, which is

$$y^2 = x^3 + bx + d \pmod{p} \tag{2.6}$$

Where $b$ and $d$ are Weierstrass coefficients selected from a finite field $\mathbb{F}_p$. The curve is said to be *smooth* if the discriminant $4b^3 + 27d^2 \neq 0$ and this curve is called elliptic curve.

#### 2.4.2.1 Point Addition

Suppose we have two points on an elliptic curve $E$, $A$ and $B$. To add such points, the procedures listed below need to be followed.

1. $A$ straight line is passed from points $A$ and $B$ .
2. At some point, the straight line intersects the curve, say at $S$ of $E$.
3. Next, we get the point $W$ as the product of $A$ and $B$. It is only appropriate to take the negative of $S$, which is $-S = (x, -y)$.

#### 2.4.2.2 Point Doubling

To add a point $A$ to itself, these steps are applied :

1. Draw a tangent on $A$.
2. At some point, it intersects the curve, again considered as $S$ of $E$.
3. Next, to get the point $W = 2A$ as the product of adding $A$ to itself, it is only necessary to take $S$ negative.

#### 2.4.2.3 Point at Infinity

For the addition of $A$ to $-A$ , the same method can be used. We know that $-A$ is the reflection of $A$ So, it approaches infinity as straight line passes from them.

To describe a particular point located at infinity that is recognised as a point towards infinity.

#### 2.4.2.4 Mathematical Representation

To add $A(x_1, y_1)$ and $B(x_2, y_2)$ on the curve (2.6 ) a line must be drawn through them. This line intersects the curve and defines the geometric point addition. Let the line $L$ pass through points $A$ and $B$. The point slope form of L is:

$$L : y = sx + c$$

Now, we are only left to define the slope $s$, it includes the following cases:

**Case 1** If $A$ and $B$ are two different points, then

$$s = \frac{y_2 - y_1}{x_2 - x_1} \tag{2.7}$$

**Case 2**

If $A$ and $B$ are same points, then

$$s = \frac{3x_1^2 + b}{2y_1} \tag{2.8}$$

Using basic algebra, the new point $W(x_3, y_3)$ acquired by adding $A(x_1, y_1)$ and $B(x_2, y_2)$ has the following coordinates:

$$x_3 = s^2 - x_1 - x_2 \tag{2.9}$$

$$y_3 = s(x_1 - x_3) - y_1 \tag{2.10}$$

**Example 2.4.1.** Let us consider the elliptic curve over $\mathbb{F}_{13}$, given by,

$$y^2 = x^3 + 7x + 4 \pmod{13} \tag{2.11}$$

The points that lie on the given curve are shown in Table(2.3). Let $A(9, 4)$ and $B(10, 12)$ be two points on elliptic curve (2.11). Then the formula in (2.9) and (2.10) give us with the co-ordinates of new point $W(x, y)$. The elliptic curve points addition for $E_{F_{13}}$ is shown in Table (2.4). In the table the first column shows the

values of $x$ and the second column gives the corresponding values of $y^2$. The table is organized in such a way that for each value of $x$, the corresponding value of $y^2$ is calculated. This table essentially enumerates all valid points of the elliptic curve over the finite field $F_{13}$.

TABLE 2.3: Points of elliptic curve

| $x$ | $y^2$ | $y_1$ | $y_2$ | $A(x,y)$ | $\bar{A}(x,y)$ |
|---|---|---|---|---|---|
| 0 | 4 | 2 | 11 | (0,2) | (0,11) |
| 1 | 12 | 5 | 8 | (1,5) | (1,8) |
| 2 | 0 | 0 | 0 | (2,0) | - |
| 3 | 0 | 0 | 0 | (3,0) | - |
| 4 | 5 | - | - | - | - |
| 5 | 8 | - | - | - | - |
| 6 | 2 | - | - | - | - |
| 7 | 6 | - | - | - | - |
| 8 | 0 | 0 | 0 | (8,0) | - |
| 9 | 3 | 4 | 9 | (9,4) | (9,9) |
| 10 | 8 | - | - | - | - |
| 11 | 8 | - | 3 | - | - |
| 12 | 9 | 3 | 10 | (12,3) | (12,10) |

Evaluate the slope s by

$$\begin{aligned} s &= \frac{10-4}{12-9} \bmod 13 \\ &= \frac{6}{3} \bmod 13 \\ &= 6 \cdot 3^{-1} \bmod 13 \end{aligned}$$

By using Extended Euclidean algorithm,

$$\begin{aligned} s &= (6)(9) \bmod 13 \\ s &= 2 \bmod 13 \end{aligned}$$

Put the value of s in (2.9) and (2.10), gives us the coordinates of new point.

$$x_3 = (2)^2 - 9 - 12 \mod 13$$
$$x_3 = 4 - 21 \mod 13$$
$$x_3 = -17 \mod 13$$
$$x_3 = 9 \mod 13$$

$$y_3 = 2(9 - 9) - 4 \mod 13$$
$$y_3 = 2(0) - 4 \mod 13$$
$$y_3 = -4 \mod 13$$
$$y_3 = 9 \mod 13$$

So, $W(x_3, y_3) = (9, 9)$ is the addition of points. Now, let us add a point A(9,4) to itself. To compute 2.7 as :

$$s = \frac{3(9)^2 + 7}{2(4)} \mod 13$$
$$s = \frac{243 + 7}{8}$$
$$s = \frac{250}{8}$$
$$s = 250 \cdot 8^{-1}$$
$$s = 250 \cdot 5$$
$$s = 1250$$
$$s = 2$$

Put the value of s in (2.9) and (2.10),then we can find the co-ordinates $x_3$ and $y_3$ as :

$$x_3 = (2)^2 - 2(9) \mod 13$$
$$x_3 = 12$$
$$y_3 = 2(9 - 12) - 4 \mod 13$$
$$y_3 = 3$$

so, $W(x_3, y_3) = (12, 3)$

TABLE 2.4: Addition of points of $E_{F_{13}}(7, 4)$

| + | $\mathcal{O}$ | $(0,2)$ | $(0,11)$ | $(1,5)$ | $(1,8)$ | $(2,0)$ | $(3,0)$ | $(8,0)$ | $(9,4)$ | $(9,9)$ | $(12,3)$ | $(12,10)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{O}$ | $\mathcal{O}$ | $(0,2)$ | $(0,11)$ | $(1,5)$ | $(1,8)$ | $(2,0)$ | $(3,0)$ | $(8,0)$ | $(9,4)$ | $(9,9)$ | $(12,3)$ | $(12,10)$ |
| $(0,2)$ | $(0,2)$ | $(12,3)$ | $\mathcal{O}$ | $(8,0)$ | $(9,9)$ | $(12,10)$ | $(9,4)$ | $(1,8)$ | $(1,5)$ | $(3,0)$ | $(2,0)$ | $(0,11)$ |
| $(0,11)$ | $(0,11)$ | $\mathcal{O}$ | $(12,10)$ | $(9,4)$ | $(8,0)$ | $(12,3)$ | $(9,9)$ | $(1,5)$ | $(3,0)$ | $(1,8)$ | $(0,2)$ | $(2,0)$ |
| $(1,5)$ | $(1,5)$ | $(8,0)$ | $(9,4)$ | $(12,10)$ | $\mathcal{O}$ | $(9,9)$ | $(12,3)$ | $(0,11)$ | $(2,0)$ | $(0,2)$ | $(1,8)$ | $(3,0)$ |
| $(1,8)$ | $(1,8)$ | $(9,9)$ | $(8,0)$ | $\mathcal{O}$ | $(12,3)$ | $(9,4)$ | $(12,10)$ | $(0,2)$ | $(0,11)$ | $(2,0)$ | $(3,0)$ | $(1,5)$ |
| $(2,0)$ | $(2,0)$ | $(12,10)$ | $(12,3)$ | $(9,9)$ | $(9,4)$ | $\mathcal{O}$ | $(8,0)$ | $(3,0)$ | $(1,8)$ | $(1,5)$ | $(0,11)$ | $(0,2)$ |
| $(3,0)$ | $(3,0)$ | $(9,4)$ | $(9,9)$ | $(12,3)$ | $(12,10)$ | $(8,0)$ | $\mathcal{O}$ | $(2,0)$ | $(0,2)$ | $(0,11)$ | $(1,5)$ | $(1,8)$ |
| $(8,0)$ | $(8,0)$ | $(1,8)$ | $(1,5)$ | $(0,11)$ | $(0,2)$ | $(3,0)$ | $(2,0)$ | $\mathcal{O}$ | $(12,10)$ | $(12,3)$ | $(9,9)$ | $(9,4)$ |
| $(9,4)$ | $(9,4)$ | $(1,5)$ | $(3,0)$ | $(2,0)$ | $(0,11)$ | $(1,8)$ | $(0,2)$ | $(12,10)$ | $\mathcal{O}$ | $(12,3)$ | $(8,0)$ | $(9,9)$ |
| $(9,9)$ | $(9,9)$ | $(3,0)$ | $(1,8)$ | $(0,2)$ | $(2,0)$ | $(1,5)$ | $(0,11)$ | $(12,3)$ | $\mathcal{O}$ | $(12,10)$ | $(9,4)$ | $(8,0)$ |
| $(12,3)$ | $(12,3)$ | $(2,0)$ | $(0,2)$ | $(1,8)$ | $(3,0)$ | $(0,11)$ | $(1,5)$ | $(9,9)$ | $(8,0)$ | $(9,4)$ | $(12,10)$ | $\mathcal{O}$ |
| $(12,10)$ | $(12,10)$ | $(0,11)$ | $(2,0)$ | $(3,0)$ | $(1,5)$ | $(0,2)$ | $(1,8)$ | $(9,4)$ | $(9,9)$ | $(8,0)$ | $\mathcal{O}$ | $(12,3)$ |

### 2.4.3 Elliptic Curve Discrete Logarithm Problem

In the Elliptic Curve Discrete Logarithm Problem (ECDLP), the objective is to determine an integer $Q$, given two points $A$ and $B$ on an elliptic curve $E_p(a, b)$, such that:

$$B = Q \cdot A$$

If such a $Q$ exists, it is called the discrete logarithm of $B$ to the base point $A$ on the curve.

### 2.4.4 Elliptic Curve Diffie-Hellman Key Exchange Protocol

In order to communicate securely, Ayesha and Bilal must exchange keys so they may encrypt and decrypt the communications. In 1976, the concept was given by Diffie and Hellman [7] to exchange keys over a public network without compromising the security. The scheme is designed with the help of a cyclic group of elliptic curve points and safety relies on the complexity of overcoming ECDLP.The following method tells the whole story of Ayesha and Bilal exchanging keys using Diffie-Hellman key exchange protocol.

1. An elliptic curve $E$ over a finite field $\mathbb{F}_q$ is mutually chosen by Ayesha and Bilal, with a base point $G$ on the curve $E$.

2. Ayesha choose a random number $e_A \in \{1, 2, 3, \ldots, n-1\}$ as her private key, and calculates her public key as:

$$P_A = e_A G$$

3. Bilal selects his private key $e_B \in \{1, 2, 3, \ldots, n-1\}$ and computes his public key as:

$$P_B = e_B G$$

4. Ayesha and Bilal exchange their public keys $P_A$ and $P_B$, with each other.

5. Ayesha computes the shared secret as:

$$P_{AB} = e_A P_B = e_A(e_B G) = e_A e_B G$$

Bilal also computes:

$$P_{AB} = e_B P_A = e_B(e_A G) = e_B e_A G$$

6. $P_{AB}$ can serve as a session key for secure communication between Ayesha and Bilal.

## 2.5 Eliptic Curve Encryption and Decryption

One method of asymmetric key cryptography is based on the use of elliptic curve, every user has a public and private key for secure communication.

### 2.5.1 Global Setting

Ayesha, the sender, and Bilal, the recipient, communicate using these global parameters.

1. The base point $G$ such that $nG = \mathcal{O}$, where $\mathcal{O}$ is the point at infinity and $n$ is the smallest prime number.

2. A prime integer modulo $q$ and constants $u$ and $v$.

### 2.5.2 Key Generation Phase

1. Ayesha choose a secret key $e_A$ based on the set $\{1, 2, \ldots, n-1\}$ and calculate her public key as:

$$P_A = G \cdot e_A$$

2. Bilal determines his public key as follows after selecting his private key, $e_B < n$.:

$$P_B = e_B \cdot G$$

### 2.5.3 Encryption Phase

Ayesha sent Bilal a message $t$ using the ECC method. Then t is converted into an elliptic curve point $Q_t$ . Ayesha uses Bilal public key $P_B$ to find the ciphertext $Q_C$ as the elliptic curve pair of points after selecting a random number k.

$$Q_C = (kG,\ Q_t + kP_B) \mod p \tag{2.12}$$

### 2.5.4 Decryption Phase

Bilal decrypt the message back into original form after receiving ciphertext $Q_C$, by multiplying $kG$ with his private key $e_B$ and then add the result into second ciphertext pair $Q_t + kP_B$

$$\begin{aligned} Q_t + kP_B - e_B(kG) &= Q_t + ke_BG - ke_BG \\ &= Q_t \end{aligned}$$

which is plaintext point, corresponding to plaintext message $t$.

**Example 2.5.1.** Let us consider an elliptic curve $y^2 = x^3 - x + 188 \mod 751$. Let $G = (0, 376)$ be the base point. Total number of points and order of this curve

is 727 where $727(0, 376) = \mathcal{O}$. Let Ayesha wants to send a message $Q_t$ to Bilal by using ECC encryption. Ayesha selects her secret key $e_A = 6$ and compute public key as $P_A = 6(0, 376) = (6, 390)$. Bilal selects his secret key $e_B = 5$ and compute public key as $P_B = 5(0, 376) = (188, 657)$. Ayesha chooses the secret random number $k = 113$ to encrypt the original message $Q_t = (443, 253)$, as follows :

$$\begin{aligned} Q_C &= [kG, Q_t + kP_B] \mod p \\ &= [113(0, 376), (443, 253) + 113(188, 657)] \mod 751 \\ &= [(34, 633), (443, 253) + (529, 254)] \mod 751 \\ &= [(34, 633), (418, 18)] \end{aligned}$$

Ayesha sends $Q_C = [(34, 633), (418, 18)]$ to Bilal. After receiving $Q_C$ Bilal decrypt it to get $Q_t$.

$$\begin{aligned} Q_t &= [Q_t + kP_B - e_B(kG)] \mod p \\ &= [(443, 253) + 113(188, 657) - 5(34, 633)] \mod 751 \\ &= [(443, 253) + (529, 254) - (529, 254)] \mod 751 \\ &= (443, 253) + \mathcal{O} \\ &= (443, 253) \end{aligned}$$

## 2.6 Digital Signature

A digital signature is a mathematical code that is attached to an electronic document for its verification.The digital signature provides the assurance that the message is not tampered or changed during the whole communication.

It is computationally infeasible to develop a valid signature without knowing the sender's private key.

In 1976, Diffie and Hellman [7] gave the idea of digital signature.

Suppose Ayesha (sender) aims to generate a digital signature for a document and sends it to Bilal.

**Ayesha**

1. Choose a digital document to be signed.
2. Produces the hash value of this document
3. She encrypts the hash value using her private key in order to calculate the digital signature.
4. Sends the original digital document as well as its signature to Bilal.

**Bilal**

1. Bilal will use Ayesha public key to obtain the hash value that was calculated at Ayesha end in order to decrypt the digital signature.
2. Determines a document's hash value after receiving it from Ayesha.
3. Accepts the digital document as a valid if the hash value computed in Step 1 and Step 2 are same.
4. It is assumed that the received document was altered or tampered with during transmission if these two hash values disagree.

## 2.7 Elliptic Curve Digital Signature Algorithim (ECDSA)

ECDSA is a simple transposition of the DSA (Digital Signature Algorithm) to the elliptic curve. It was proposed in 1992 by Scott Vanstone [11].

It offers the benefit of providing smaller keys with an equivalent level of security. Its unquestionable benefit explains why it has emerged as the most used numerical signature across several fields.

### 2.7.1 Use of ECDSA Algorithim

A signer creates a digital signature for data using ECDSA, and a verifier confirms the signature's legitimacy. Every signatory has both a public and a private key. Actually, the only way for someone to confirm the legitimacy of the message's signature is to use the signer's public key.

A third party is unable to provide the proper signature if they are unaware of the approver secret key. While the public key is used for signature verification, the private key is utilized for signature generation[33] . The message $M$ is hashed using a hash function prior to the creation and verification of the signature.

The set up of an ECDSA required the implementation of four essential algorithms, These four algorithims are[19].

1. Domain parameter generation, this function is responsible for producing safe domain parameters.
2. Key Pair generation, this method creates a key pair $(Q, d)$ by accepting a set of domain parameters $D$ as input.
3. Signature generation, this process creates a signature $S(r, s)$ by taking as inputs a private key, a message $m$, and a collection of domain parameters $D$.
4. Signature verification, this method approves or rejects the suggested signing after receiving as inputs a message m, a public key $Q$, a domain parameter $D$, and an allegedly acceptable signature $s$.

#### 2.7.1.1 Domain Parameter Validation

The following conditions must be verified by the generator of parameters of the elliptic curve.

1. Verify that $p$ is an odd prime number.
2. Verify that $Q(x_G, y_G) \neq \mathcal{O}$.

3. Check that $a, b, x_G, y_G \in \mathbb{F}_q$.

4. Check that $a$ and $b$ are parameters of the curve, and verify the equation:

$$4a^3 + 27b^2 \not\equiv 0 \pmod{p} \quad \text{over } \mathbb{F}_q$$

5. Verify that $Q$ is a point on the elliptic curve defined by $a$ and $b$, i.e., it satisfies the curve equation.

6. Verify that the order $n$ of the point $P$ is a prime number.

7. Verify that $nQ = \mathcal{O}$ and that $n \neq p$.

#### 2.7.1.2 Digital Signature ECDSA Generation

1. Ayesha generates a random number $k$ such that $0 < k < n$.

2. She computes the elliptic curve point:

$$kP = (x_1, y_1)$$

3. She calculates:

$$r = x_1 \bmod n$$

   If $r = 0$, she returns to Step 1.

4. She computes:

$$e = H(m)$$

   where $H(m)$ is the hash of the message $m$.

5. She calculates:

$$s = k^{-1}(e + dr) \bmod n$$

   where $d$ is her private key.

6. If $s = 0$ or $r = 0$, she returns to Step 1 (tries again).

7. Finally, Ayesha takes the pair $(r, s)$ as the digital signature and sends it to Bob.

#### 2.7.1.3 Digital Signature ECDSA Verification

1. Bilal obtains an authentic copy of Ayesha public key: $(E, P, n, Q)$.
2. He verifies that $r$ and $s$ are integers in the interval $[1, n-1]$. If not, he rejects the signature.
3. He computes:
$$e = H(m)$$
where $H(m)$ is the hash of the received message $m$.
4. He calculates:
$$u_1 = e \cdot s^{-1} \bmod n \quad \text{and} \quad u_2 = r \cdot s^{-1} \bmod n$$
5. Using Alice's public key and the computed values, he evaluates:
$$u_1 P + u_2 Q = (x_1, y_1)$$
6. He sets:
$$v = x_1 \bmod n$$
7. Finally, he checks whether:
$$v = r$$
If the equality holds, the signature is accepted. Otherwise, it is rejected.

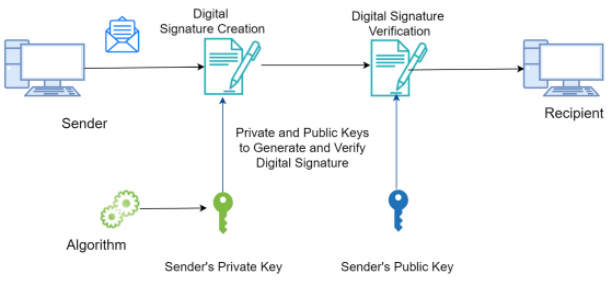


FIGURE 2.4: ECDSA signature and verification steps

## 2.8 Hyperelliptic Curve

A hyperelliptic curve $C$ of genus $g$, defined over a finite field $\mathbb{F}_q$, has the following form:

$$C : y^2 + h(x)y = f(x) \tag{2.13}$$

where $h(x)$ and $f(x)$ are polynomials with coefficients in $\mathbb{F}_q$.The degree of $h(x)$ is at most $g$ and $f(x)$ has degree at least $2g + 1$. For non-singularity, no points of curve $C$ should simultaneously satisfy the equations:

$$2y + h(x) = 0 \quad \text{and} \quad h'(x) - f'(x) = 0.$$

The *genus* of a curve is the number of non-intersecting curves that are drawn on a surface without coming into contact with one another. It determines the amount of computing time required for the curve's implementation.

It is believed that a curve of genus 2 is appropriate for safe and effective calculations. Following are the curve of different genus.

1. $y^2 = x^3 + x + 1$ has genus one and is called an elliptic curve.
2. $y^2 + xy = x^5 + b_1x^3 + b_2x^2 + b_3x + b_4$ has genus 2, and $h(x) = x$.
3. $y^2 = x^9 + b_1x^7 + b_2x + b_3$ has genus 4.

A detailed discussion on hyperelliptic curves is presented in Chapter 4.

## 2.9 HealthCare 4.0

Healthcare 4.0 refers to the integration of advanced technologies, such as artificial intelligence (AI), Internet of Things (IoT), and data analytics, into healthcare systems to improve patient care, outcomes, and efficiency. Scientists have focused a lot of emphasis on intelligent transportation systems (ITSs), intelligent health monitoring, intelligent home automation, security checking, and other IoT-enabled smart city applications[18].

Industry 4.0 and the fourth health care revolution are currently developing concurrently. In this situation, the process of delivering health care turns into a cyber-physical system that is outfitted with wearable technology, RFID (radio-frequency identification), the Internet of Things, and various medical devices, intelligent sensors, medical robots, and other gadgets.

These are combined with cloud computing, big data analysis, artificial intelligence, and decision support methods to create a smart and connected health care delivery system.

In such a system, not only the health care organizations and facilities ( hospital, clinics, and long-term care facilities) are connected; but also all the equipment and devices, as well as the patients' home and communities are linked together. Patient-related information, e.g. medication history, diagnostic notes, lab results, treatment plans, pharmacy refills, billing, and insurance claims, can be potentially shared through adequate protocols.

In addition, through AI techniques, we can envision proactive treatment, disease prediction and prevention, personalized medicine, and enhanced patient-centered care. Thus, a pervasive, smart, and interconnected health care community emerges, which leads to the paradigm of Health Care 4.0.

### 2.9.1 Smart Devices Network

The Internet of Things (Smart Devices Network) is a network of physical objects, such as vehciles, appliances, and other goods, that are equipped with sensors, software, and network connectivity to collect and share data.

### 2.9.2 Cloud

Cloud storage store the data that has been gathered from various devices. A collection of servers that operate continuously over the internet is known as cloud computing or cloud infrastructure. Users can make decisions at any moment by remotely accessing the data from the cloud storage.

### 2.9.3 Blockchain

A decentralized, digital ledger that records transactions across a network of computers in a secure and transparent manner [40]. Blockchain technology has several important characteristics that make it secure, reliable, and widely used in fields like finance and healthcare. One of its core principles is decentralization, meaning no single authority or organization controls the entire network. Instead, it operates on a peer-to-peer model where all participants have equal control. Alongside this, the blockchain uses a distributed ledger, where multiple copies of the transaction records are stored across many computers (called nodes). This ensures that even if one node fails or is compromised, the data remains safe and accessible elsewhere. Another key feature is immutability, which means that once a transaction is recorded on the blockchain, it cannot be changed or deleted. This makes the system highly trustworthy and resistant to fraud. In addition to these foundational features, blockchain is also known for its security. It protects user identities and secures transaction data using advanced cryptographic techniques. Transparency is another important aspect, as all confirmed transactions are visible to every participant in the network, making the system open and accountable. Finally, blockchain operates using consensus mechanisms, which are rules that help network nodes agree on which transactions are valid.

### 2.9.4 IoT Device (IN)

It is a part that want to store or retrieve its gene data from the healthcare system. In our system, someone who has total authority over their gene data is often referred to as a data owner. IoT nodes can provide the cryptographic keys and metadata for their data by defining access control policies. Using newly issued cryptography keys, this node can also update its data locally for auditing purposes.

### 2.9.5 Medical user (MU)

This component tries to access gene data from the CSP that is connected to a specific IN's blockchain. It can only do this if the IN gives permission. A medical

practitioner (such as a physician, nurse, radiologist, or pathologist), a caretaker, or a health insurance company typically owns a medical user (MU). After sending a request to the CSP, MU can receive both the metadata from the blockchain and the original gene data in readable form.

### 2.9.6 Fog node (FN)

FN is a part of fog computing that uses physical devices, like access points with sensors, to provide fog services. After checking the information, FN sends the encrypted data it gets from IN or MU's search requests to the CSP. It also receives data back from the CSP, checks it, and sends it to IN or MU, who are the intended users. In this setup, FN acts only as a middle layer between the edge devices and the CSP — it doesn't keep or store any data itself.

### 2.9.7 Cloud storage (CS)

Either a public or private Cyber-Physical System (CPS) provides the CS. IoT device-derived encrypted gene data is stored there. Additionally, it ensures that every encrypted file is stored on the blockchain together with its access history and additional information (metadata).

### 2.9.8 Private blockchain (PB)

For newly created data, PB encrypts access logs and meta-data. PB distributes the system's meta-data and access log to enable search functionality and protect against quantum risks from parallel computing services and other threats like forging. Well-known decentralized blockchains include Hyperledger and Ripple (XRP).

# Chapter 3

# Secure Gene Profile Data Processing Using Lightweight Cryptography and Blockchain

This chapter explores the design and implementation of a secure, scalable, and efficient approach for processing gene data, as proposed by Mahajan and Reddy [17]. To demonstrate the effectiveness of their model, they showcase various threat-based case studies, highlighting its resilience against multiple opponent models. The study of their work demonstrate that the new method achieves a significant speedup in encryption and decryption, making it ideal for secure and efficient data processing in healthcare.

## 3.1 Hybrid Technique

ECC is a method of encryption that uses a pair of keys one private and one public to lock (encrypt) and unlock (decrypt) data. It is a more modern alternative to the traditional RSA (Rivest-Shamir-Adleman) encryption. Unlike RSA, which relies on prime numbers, ECC uses elliptic curve mathematics to protect key pairs. ECC is known for its strong security and the ability to use smaller key sizes, making it more efficient.

Due to its use of elliptic curves, ECC keys are difficult to crack, offering robust protection for public-key cryptography.Applications in Industry 4.0 and Healthcare 4.0, where efficiency and security are crucial, now require ECC.

Asymmetric key cryptography and robust security with short keys are features of ECC cryptography. Asymmetric key pairs are used in ECC cryptography, which provides robust security with smaller keys.

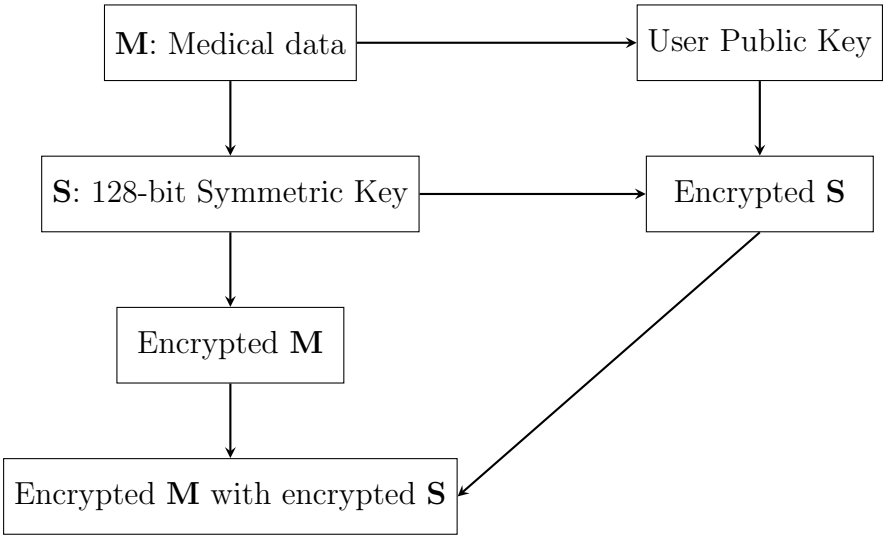


FIGURE 3.1: ECC and ECDH-based hybrid encryption process for enhanced security

The process depicted in Figure 3.1 utilizes a hybrid encryption approach to securely encrypt gene data. A 128-bit symmetric key is generated and used with the AES algorithm to encrypt the gene data. The AES algorithm transforms the plaintext gene data into ciphertext, resulting in encrypted gene data.

The security and integrity of the sensitive genetic data are guaranteed by this encryption procedure. The symmetric key, used for encrypting the gene data, will later be encrypted using an ECC public key, but the primary focus here is on the secure encryption of the gene data itself using the symmetric key and AES.

The outcome of this process is the encrypted gene data, which is now protected and ready for secure transmission.

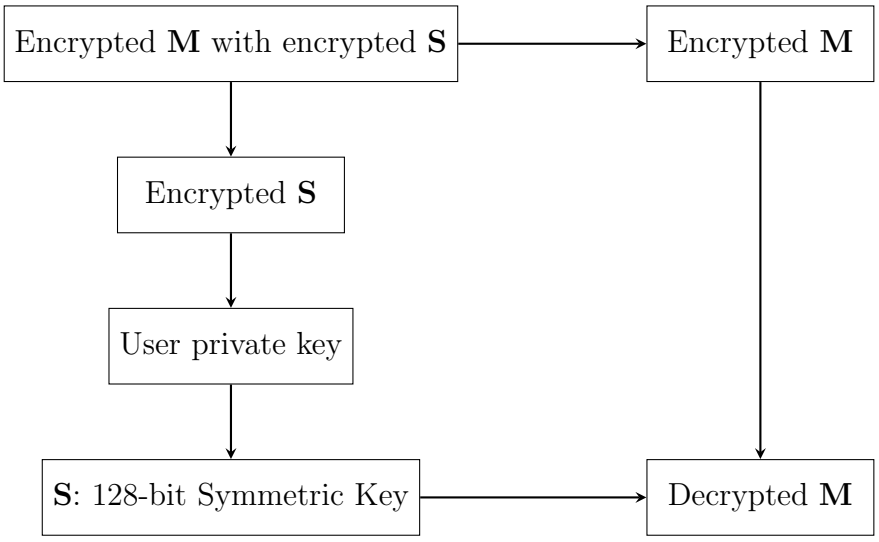


FIGURE 3.2: Hybrid Decryption Procedure Using ECDH and ECC

The hybrid decryption process is a secure method that combines the strengths of asymmetric and symmetric cryptography to protect sensitive gene data during transmission and storage. As shown in Figure 3.2, this process begins when the recipient receives two main components: the encrypted gene data and the encrypted symmetric key. The symmetric key is originally used to encrypt the gene data using a fast and efficient symmetric encryption algorithm like AES (Advanced Encryption Standard).

However, to securely share this symmetric key, it is itself encrypted using the recipient's public key based on Elliptic Curve Cryptography (ECC). Upon receiving the data, the recipient uses their ECC private key to decrypt the encrypted symmetric key, recovering the original symmetric key used for AES encryption. With this symmetric key now available, the recipient applies the AES algorithm to decrypt the gene data, converting the ciphertext back into its original plaintext form, which reveals the actual genetic information. A critical part of this process is the use of ECDH protocol, which securely facilitates the exchange of the symmetric key between sender and recipient. With the use of ECDH, an attacker cannot obtain the symmetric key without the private ECC key, even if they manage to intercept the transmission. By combining ECC and ECDH for secure key handling, and AES for efficient data decryption, this hybrid approach guarantees

both the confidentiality and integrity of highly sensitive gene data, making it ideal for modern Healthcare 4.0 environments where data privacy is a top priority.

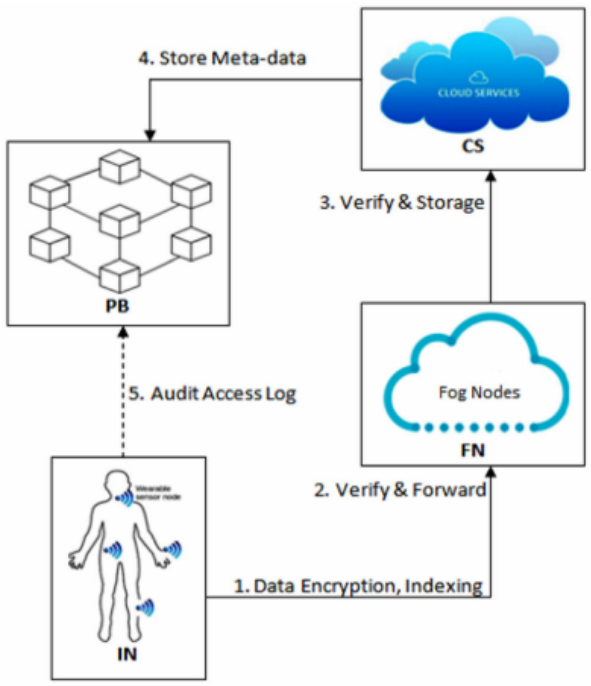


FIGURE 3.3: Proposed secured gene data storage

Figure 3.3 illustrates the proposed method for securely storing patient genomic data on CSP (Cloud Storage Platform) in conjunction with Private Blockchain (PB). The periodic gene data gathered at IN is encrypted using the ECC-based hybrid encryption technique, as seen in Fig 3.1. The digital signature for the encrypted message is then produced using the ECDSA algorithm, and index creation comes next. The index is created using the current timestamp associated with IN. To stop attacks, the FN confirms the digital signature of the encrypted transmission. CS saves the data after verification. CS confirms that the message was received before saving it in CSP. PB creates and stores encrypted data metadata in a distributed manner. This technique offers great security against a number of risks, including quantum risk, and lightweight gene data storage with short transmission times. If any part of the digital signature verification process for an encrypted transmission fails, the hospital is informed and the message and associated keys are erased. The concerned IN and the administration are cautioned to act. Gene data audits can also be conducted using the access logs provided

by PB by IN. ECC and RSA key sizes for symmetric encoding are contrasted in

TABLE 3.1: Security bits and encryption key sizes

| Security bits | Symmetric encryption | RSA public key size (bits) | ECC public key size (bits) |
|---|---|---|---|
| 80 | DES | 1024 | 160 |
| 112 | 3DES | 2048 | 224 |
| 128 | AES-128 | 3072 | 256 |
| 192 | AES-192 | 7680 | 384 |
| 256 | AES-256 | 15,360 | 521 |

Table 3.1. With tiny keys, ECC cryptography offers robust security and asymmetric key cryptography. Cryptography is accelerated by smaller key sizes. ECC (Elliptic Curve Cryptography) makes cryptographic tasks like creating signatures, encrypting and decrypting data, and verifying information faster and more secure. ECC is an example of asymmetric cryptography that uses a private key for encryption and a public key for decryption. ECDH is advised for hybrid encryption and decryption since ECC does not support direct encryption and decryption. Using a shared secret key created by ECDH, messages can be encrypted and decrypted using the same key. Verifying encrypted data and keys is essential to ensuring message integrity against potential assaults, in addition to encryption and decryption. The recipient verifies the signature of encrypted communications using public keys. Using ECC keys, ECDSA creates and validates signatures. This process will be explained in more detail in the upcoming sections.

## 3.2 The Proposed Encryption Scheme

This scheme discusses healthcare 4.0 architecture based on blockchain and lightweight cryptography. Latest developments in online storage data sharing and storing using blockchain technology point to possible security aspects. A central system is needed to keep data safe when sharing and storing it in smart healthcare. Healthcare 4.0 includes distributed edge computing, $CS$, blockchain, and the edge layer (like patients or gene users).

To store and retrieve gene data from the edge layer of blockchain-protected cloud storage, fog nodes employ basic cryptography. To keep information safe, ECC uses two methods: ECDSA for digital signatures and ECDH for secure key sharing.

Blockchain protects against forging, manipulation, and quantum attacks by storing meta-data from cloud-stored data.

### 3.2.1 Global parameters

Following are the global parameters for this scheme.

$p$ : A prime number

$G$ : A generator of an elliptic curve with order $o$

$oG = \mathcal{O}$

$\mathcal{O}$ : point at infinity

$H$ : Hash function

### 3.2.2 Notations

TABLE 3.2: Symbols used in the Scheme

| Symbol | Description | Symbol | Description |
|---|---|---|---|
| $IN$ | Gene data holder (IoT node) | $FN$ | Nearby computers (Fog nodes) |
| $CS$ | Online storage (Cloud storage) | $MU$ | Client (Medical User) |
| $PB$ | Closed blockchain (Private blockchain) | $G$ | Base point of the elliptic curve |
| $\alpha_r$ | Private key in ECC | $X_u$ | Public key in ECC |
| $ST$ | Timestamp | $m$ | Gene data collected at node $IN$ |
| $i$ | Index linked with $m$ and $IN$ | $(r, s)$ | ECDSA signature pair |
| $X_h$ | Shared secret key from ECDH | $m_{\text{encrypt}}$ | Encrypted gene data |
| $m_{\text{hash}}$ | Hash of $m_{\text{encrypt}}$ | $o$ | Order of curve point $G$ |

Below is algorithim used in proposed scheme.

The first step in algorithim is key generation $(\alpha_r, Xu) = Keygen\_ECC$. 256-bit random private key is used in the symmetric AES-128 encryption method. The private key was generated at random between 1 and G multiplicative order o. Calculated by,

$$\alpha_r = \texttt{rand}\{1, 2, \ldots, o-1\} \tag{3.1}$$

public key can be generated as :

$$Xu = \alpha_r.G \tag{3.2}$$

Using the ECC public and private keys linked to the multimedia data ($m$) at the current timestamp, ECDH creates a shared secret key.

$$Xh = Xu.\alpha_r \tag{3.3}$$

When we encrypt data using AES-128, a secret key is used and a unique index number is created, linked to the user's ID and timestamp, for tracking and auditing. A digital signature is then generated using ECDSA, involving a SHA-2 hash of the encrypted message and a private-public key pair. The cloud storage and our system verify the message using this signature, processing the data if verified, and discarding it otherwise.

The encrypted data is saved with its index and keys after successful verification, and metadata is shared with a public blockchain to guard against malicious activity, manipulation, and quantum attacks.

## Algorithm 3.2.1.

**Input** $m$: Gene data collected at $IN$

$sT$: Current Timestamp for index generation

**Step 1 ($IN$)** ECC is used to generate the public and private key pair in the first stage. Following the creation of the ECC-based public and private key pair for the timestamp currently linked with the multimedia data $m$, a shared secret key is obtained using the ECDH protocol. Data is encrypted using AES-128 with a shared secret key. A unique index, based on the user ID and timestamp, is created

for storage and auditing. A lightweight ECDSA signature is then generated by hashing the encrypted data (using SHA-2) and signing it with the user key pair.

1.1. $(\alpha_r, Xu) =$ Keygen_ECC

1.2. $Xh$: ECDH shared secret key computation from(3.3)

1.3. If $m \neq null$, then:

$$(m_{\text{encrypt}}, i) = \text{encryption}(m, Xh, sT)$$

Else

$$\text{discard}(m)$$

End if

1.4. $m_{\text{hash}} = \text{SHA2}(m_{\text{encrypt}})$

1.5. $(r, s) = \text{signing}(m_{\text{hash}}, \alpha_r, Xu)$

1.6. Forward $(m_{\text{encrypt}}, i, r, s)$

**Step 2 ($FN$)** At the Fog Node ($FN$), the following steps are performed. First, the associated public key ($Xu$) is retrieved. Next, the validity of the public key ($Xu$) is checked to ensure it is genuine and trustworthy. Then, the encrypted data ($m_encrypt$) is hashed using the SHA2 algorithm to produce a message hash ($m_hash$). After that, the digital signature (r, s) is verified using the message hash ($m_hash$) and public key (Pu). If the verification is successful ($f == 1$), the encrypted data ($m_encrypt$), index ($i$), and digital signature ($r, s$) are forwarded to the next step. Otherwise, the data is discarded, and the process is terminated. Before the data is processed and stored further, $FN$'s verification procedure guarantees its integrity and validity.

2.1 Get associated $Xu$

2.2 Check the $Xu$ validity

2.3 $m_{\text{hash}} = \text{SHA2}(m_{\text{encrypt}})$

2.4 $f = \text{verify}(m_{\text{hash}}, r, s, Xu)$

2.5 If $f == 1$, then:

Forward $(m_{\text{encrypt}}, i, r, s)$

Else

Discard $m_{\text{encrypt}}$ and break

End if

**Step 3 ( $CS$ and $PB$)** The signature pair $(r, s)$ is used by the $FN$ (Fog Node) and $CS$ (Cloud Server) to validate the encrypted communications. The encrypted data is handled if verification is successful; if not, it is thrown away with the keys and indexes. Following $CS$ verification, metadata is distributed onto the Private Blockchain ($PB$) and the data is saved with matching index numbers and keys.

As previously mentioned, $PB$ stores metadata to safeguard data against quantum attacks, tampering, and other malicious activities.

3.1 Get associated $Xu$

3.2 Check the $Xu$ validity

3.3 $m_{\text{hash}} = \text{SHA2}(m_{\text{encrypt}})$

3.4 $f = \text{verify}(m_{\text{hash}}, r, s, Xu)$

3.5 If $f == 1$, then:

Forward $(m_{\text{encrypt}}, i, r, s)$

Else

Discard $m_{\text{encrypt}}$ and break

End if

The search and retrieval process involves several steps to ensure secure and authorized access to gene data. Firstly, the Medical User (MU) uses metadata to search the Private Blockchain (PB) for gene data. Then, using the data owner

($IN$) ID and index ($i$), the MU sends a request to the Cloud Server ($CS$). Using the current timestamp, the $MU$ generates a signature and uses shared private and public keys to validate the metadata. The authenticity of this signature is then confirmed at the Fog Nodes ($FN$) and $CS$.

If the verification is successful, the $CS$ fetches the requested encrypted gene data, signs it, and sends it to the $MU$ through the $FN$. The received encrypted data is decrypted by the $MU$ using AES-128-bit symmetric decryption and the shared secret key (Xh). Throughout this process, Secure authentication and defense against unwanted access are ensured by the use of public-private key pairs, signature creation, and verification. A secure and auditable record of every transaction is also provided via the $PB$ metadata storage.

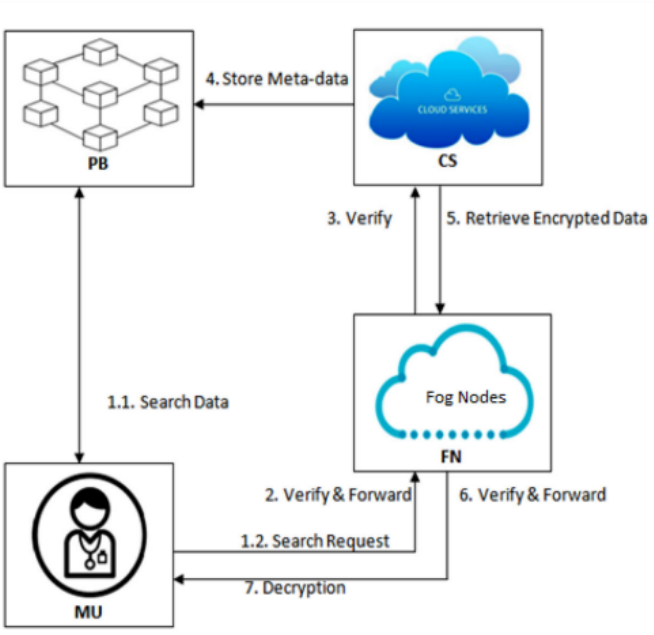


FIGURE 3.4: Proposed technique of gene data searching

The functionality of the suggested information model is shown in Figure 3.4

## 3.3 Secuirty Analysis

Protecting genetic profile data requires strong security mechanisms due to its sensitivity and long-term significance. A combination of lightweight cryptography

and blockchain ensures confidentiality, integrity, and authentication while maintaining efficiency. The use of Elliptic Curve Cryptography (ECC) and its related schemes, ECDH and ECDSA, provides an optimal balance between security and computational efficiency.

### 3.3.1 Confidentiality

**ECC** Provides strong encryption with smaller key sizes, ensuring secure storage and transmission of genetic data.

**ECDH** Facilitates secure key exchange for encrypting gene profile data, preventing unauthorized access.

**Blockchain** Stores cryptographic hashes of genetic data rather than the raw data, ensuring privacy while enabling verification.

### 3.3.2 Integrity

**ECDSA** Ensures data integrity by generating digital signatures for genetic profiles, allowing verification of unaltered data. **Blockchain** Uses hash-based linking of blocks, making genetic data records immutable and tamper-proof. Any modification to a record is detectable.

### 3.3.3 Authentication

**ECDSA** Provides authentication by verifying the identity of users accessing genetic data, ensuring only authorized entities can make changes. **Blockchain** Implements decentralized identity management, reducing reliance on a single point of failure for authentication.

### 3.3.4 Scalability

**ECC** Lightweight and computationally efficient, making it suitable for IoT-based healthcare systems and cloud environments. **ECDH** Ensures fast and secure key exchanges with minimal overhead, ideal for large-scale genetic databases.

# Chapter 4

# Secure Gene Data Storage using Hyperelliptic Curve Cryptography and Blockchain Technology

In this chapter, a proposed generalized scheme will be discussed. The proposed scheme is the modification of the scheme presented in chapter 3. In order to maintain same level of secuirty while reducing computational cost , ECC is replaced with hyperelliptic curve for securing gene data using leightweight cryptography and blockchain. 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. Hyperelliptic Curve Cryptography (HECC), as a generalization of elliptic curve cryptography, offers smaller key sizes and reduced calculation costs, making it a promising candidate for lightweight cryptographic solutions. When we use blockchain technology, which is a system that keeps records safe and cannott be changed easily, it creates a strong and flexible foundation. This is perfect for modern Internet of Things (IoT) networks and other apps that don't rely on a central control. Be aware that

hyperelliptic curves, which belong to the genus $g \geq 2$, can also be thought of as a particular kind of elliptic curve. HECC offers the benefit of a reduced "key" size while retaining the same degree of security as ECC.

Furthermore, unlike ECC, it lacks subexponential algorithms for solving HECDLP. Hyperelliptic curves are also an excellent option for lightweight cryptosystems because of the smaller size of the base field. A theoretically better level of security is provided by Hyper Elliptic Curve Cryptography (HECC) than by any of the well-known public key cryptosystems [28]. This is because, even when compared to Elliptic Curve Cryptosystems with equivalent key lengths, the mathematical complexity is considerable.

## 4.1 Hyperelliptic Curves

A Hyperelliptic curve of genus g, over a finite field has the following form $H$:

$$y^2 + h(x) \cdot y = f(x) \tag{4.1}$$

Here, $h(x)$ and $f(x)$ are math expressions (polynomials) with numbers from a certain finite field.

The size (degree) of $f(x)$ is at least $2g+1$, and the size of $h(x)$ is at most $g$, where $g$ is a number called the genus.

The curve should be non-singular, i.e., no point on $H$ satisfies both of the following conditions:

$$2y + h(x) = 0 \quad \text{and} \quad h'(x)y - f'(x) = 0.$$

If $h(x) = 0$, then equation (4.1) becomes:

$$y^2 = f(x)$$

where the degree of $f(x)$ is $2g + 1$, with the condition that $f(x)$ has no repeated roots.

### 4.1.1 Genus of a Curve

The genus of a curve refers to the number of "holes" it has, or the maximum number of non-intersecting loops that can be drawn on its surface.

This characteristic plays a significant role in determining the type of polynomial used to define the curve and the computational complexity and processing time required for its implementation.

The genus is a key factor that influences the mathematical properties and computational requirements of a curve, making it an important consideration in various applications.

### 4.1.2 Ordinary, Special and Opposite point

Assuming that $P = (x, y)$ is a finite point on the hyperelliptic curve, $\overline{P}(x, -y - h(x))$ is the curve's opposite point.

A point $\mathcal{O}$ is referred to as a point at infinity, and $\overline{\mathcal{O}}$ represents its opposite, so that $\mathcal{O} = \overline{\mathcal{O}}$.

If $P = \overline{P}$, then a point P is special; otherwise, it is called an ordinary point.

Assume that the curve $H$ is defined over the finite field $\mathbb{Z}_7$ and provided by:

$$H : y^2 + xy = x^5 + x^4 + x^2 + x + 2$$

It is obvious that $H$ does not have a singular point. Hence, $H$ is a **hyperelliptic curve** over $\mathbb{Z}_7$.

The set of points on $H$ over $\mathbb{Z}_7$, including the point at infinity, is:

$$H(\mathbb{Z}_7) = \{\mathcal{O}, (1,1), (1,5), (2,2), (2,3), (5,3), (5,6), (6,4)\}$$

The point $(6, 4)$ is a **special point** and the remaining points are ordinary points.

### 4.1.3 Divisor

An arbitrary linear combination of different points $P_1, P_2, P_3, \ldots, P_n$ on $C$ is the divisor $D = \sum m_n P$ for the hyperelliptic curve $H$, where $m_1, m_2, m_3, \ldots, m_n \in \mathbb{Z}$ has only some $m_n = 0$.

The integer

$$\deg(D) = \sum m_n$$

is called the **degree** of the divisor $D$.

The **order** of the divisor $D$ at a point $P$ is the integer

$$\mathrm{ord}_P(D) = m_n,$$

The divisor of any point on the hyperelliptic curve that is $P(x, y)$ is determined as

$$D = \begin{cases} P + \bar{P} - 2\infty & \text{if } P \neq \bar{P} \\ P - 2\infty & \text{if } P = \bar{P} \end{cases}$$

Take several points $P_1, P_2$ and $Q_1, Q_2$ on hyperelliptic curve $H$. By using the interpolation, find a curve that passes through these four points and also two additional points $R_1, R_2$ on the hyperelliptic curve. The image below displays the reflection of these two extra points on the curve, $\bar{R}_1$ and $\bar{R}_2$. These divisors are represented as

$D_1 = Q_1 + Q_2 - 2\infty$
$D_2 = P_1 + P_2 - 2\infty$ From these divisors , third divisor can be calculated as

$$D_3 = (Q_1 + Q_2 - 2\infty) + (P_1 + P_2 - 2\infty) = R_1 + R_2 - 2\infty$$

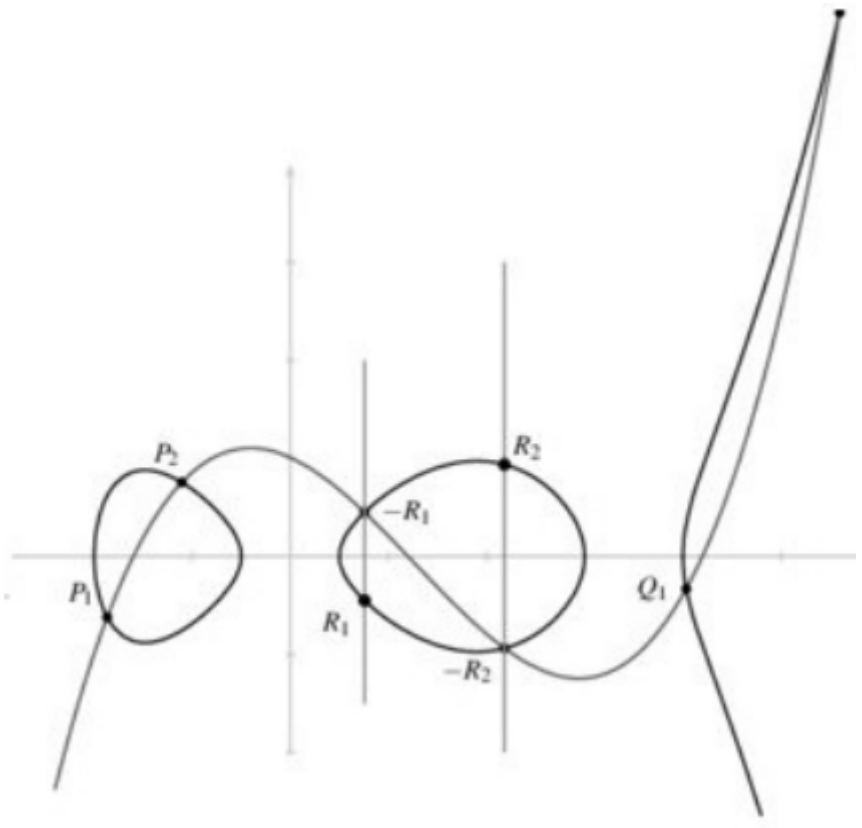


FIGURE 4.1: Geometrical Representation of Divisor

### 4.1.4 Mumford Representation

Mumford representation clearly converts Cartesian points into a polynomial divisor form. For implementation point of view, working with divisors is not easy. Cantor [4] used Mumford representation of the divisors for efficient computation and representation of the divisors. Mumford representation make working with the divisors easier in computations, especially in cryptography. One main reason to use the Mumford Representation is because it makes calculations easier. Consider a hyperelliptic curve $H$ of genus $g$. The curve $H$ is shown by.

$$y^2 + h(x)y = f(x)$$

The polynomials $f(x)$ and $h(x)$ are from the polynomial field $\mathbb{F}_q[x]$. The degree of $f(x)$ is $2g + 1$, and the degree of $h(x)$ is less than or equal to $g$. As previously

mentioned, a divisor class over the field $\mathbb{F}_q$ can be represented by two polynomials $u(x)$ and $v(x)$, both belonging to $\mathbb{F}_q[x]$. Although the polynomials $u(x)$ and $v(x)$ come from the polynomial field $\mathbb{F}_q[x]$, they must satisfy the following three conditions:

**Conditions**

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

The polynomial $u(x)$ for the divisor class $D$ is found by multiplying terms of the form $(x - x_i)$, where each $x_i$ comes from the divisor. This means:

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

The pair of polynomials $u(x)$ and $v(x)$ form the divisor class $D$, expressed as:

$$D = [u(x), v(x)]$$

**Example 4.1.1.** In this example, we consider the hyperelliptic curve $H$ defined by:

$$y^2 = x^5 + 3x^4 + 2x^3 + 3$$

of genus $g = 2$ over the field $\mathbb{F}_5$.

The Cartesian points are:

$$P_1 = (3, 0), \quad P_2 = (1, 2), \quad Q_1 = (4, 1), \quad Q_2 = (3, 0)$$

Taking the points $P_1 = (3, 0)$ and $P_2 = (1, 2)$, where $x_1 = 3$ and $x_2 = 1$. The representation of the polynomial expression $u(x)$ of the divisor class $D$ is:

$u(x) = (x - x_1)(x - x_2) = (x - 3)(x - 1) = x^2 - 4x + 3 = x^2 + x + 3 \in \mathbb{F}_5[x]$.

The polynomial $v(x)$ must satisfy the second and third conditions:

1. $\deg(v(x)) < \deg(u(x)) \leq g$
2. $u(x) \mid \big(v(x)^2 + h(x)v(x) - f(x)\big)$

Since the degree of $v(x)$ is less than the degree of $u(x)$, the polynomial $v(x)$ can be expressed as:

$$v(x) = v_{11}x + v_{14}$$

The coefficients $v_{11}$ and $v_{14}$ each take values from the set $\{0, 1, 2, 3, 4\}$. The possible combinations $(v_{11}, v_{14})$ are:

$$\begin{array}{lllll} (0,0), & (1,0), & (2,0), & (3,0), & (4,0) \\ (0,1), & (1,1), & (2,1), & (3,1), & (4,1) \\ (0,2), & (1,2), & (2,2), & (3,2), & (4,2) \\ (0,3), & (1,3), & (2,3), & (3,3), & (4,3) \\ (0,4), & (1,4), & (2,4), & (3,4), & (4,4) \end{array}$$

Any combination of $(v_{11}, v_{14})$ will satisfy the third condition:

$$u(x) \mid \big(v(x)^2 + h(x)v(x) - f(x)\big)$$

In this case, the combination $(v_{11}, v_{14}) = (4, 3)$ satisfies this condition.

We have two points $P_1 = (3, 0)$ and $P_2 = (1, 2)$ on the curve

$$y^2 = x^5 + 3x^4 + 2x^3 + 3.$$

This curve has genus 2 and is defined over the field $\mathbb{F}_5$.

Using the Mumford representation, these points are written as a pair of polynomials:

$$D_1 = (x^2 + x + 3,\ 4x + 3).$$

Similarly, the points $Q_1 = (4, 1)$ and $Q_2 = (3, 0)$ can also be represented by:

$$D_2 = (x^2 + 3x + 2,\ x + 2).$$

### 4.1.5 Hyperelliptic Curve Discrete Logarithm Problem

Given that $D_1$ and $D_2$ are two divisors of hyperelliptic curve H, the discrete logarithm problem for the hyperelliptic curve is to find the number c such that $cD_1 = D_2$ (HECDLP). It is computationally infeasible to find c. The entire security of HECC depends upon HECDLP.

## 4.2 Proposed Scheme

We introduce a method for safely processing microarray gene expression data using the highly secure, scalable, and computationally efficient Healthcare 4.0 standards. Our new Healthcare 4.0 architecture is based on blockchain technology and lightweight cryptography.

Recent developments in cloud data sharing and storage using blockchain technology point to possible security aspects. To securely exchange and store data in smart healthcare, we require a central system.

Cloud storage, fog computing, blockchain, and patient devices are all part of healthcare 4.0. Fog nodes use lightweight cryptography and blockchain-enabled cloud storage to provide safe and scalable gene data storage. As more secure and computationally effective options, we suggest utilizing Hyperelliptic Curve Cryptography (HECC) for data protection and privacy, utilizing Hyperelliptic Curve Diffie–Hellman (HECDH) and Hyperelliptic Curve Digital Signature Algorithm (HECDSA). A 160-bit key size is the foundation for the ECC's efficency and secuirty. The concept of hyperelliptic curve cryptography was used in order to improve the efficency of ECC based scheme. By using 80-bit key sizes, the HECC provides an equivalent level of secuirty to that of the ECC.

### 4.2.1 Global paramters

Following are the global parameters for this scheme

$H$ : Hyperelliptic curve

$D$ : A divisor of hyperelliptic curve

$\beta D = P_\infty$

where

$\beta$ : Order of divisor

$P_\infty$ : Point at infinity

$h$ : Hash function

### 4.2.2 Notations

TABLE 4.1: Symbols Used in the Proposed Scheme

| Symbol | Description | Symbol | Description |
|---|---|---|---|
| $IN$ | Gene data holder (IoT node) | $FN$ | Nearby computers (fog nodes) |
| $CS$ | Online storage (cloud storage) | $MU$ | Client (Medical User) |
| $PB$ | Closed blockchain (private) | $D$ | Divisor of hyperelliptic curve |
| $\alpha_r$ | Private key of HECC | $Xu$ | Public key of HECC |
| $sT$ | Timestamp | $m$ | Collecting gene data at IN |
| $i$ | Index linked with $m$ and $IN$ | $(r, s)$ | HEDSA signature pair |
| $Xh$ | Shared secret key of HECDH | $m_{\text{encrypt}}$ | Encrypted genetic data |
| $m_{\text{hash}}$ | Hash of $m_{\text{encrypt}}$ | $\beta$ | Multiplicative order of divisor |

**Algorithm 4.2.1.** All steps are described in detail below.

**Input:**

$m$: Input gene data of $IN$

$sT$: Current Timestamp for index generation

**Step 1 (Key Generation)** The entire operation is performed at the IoT node ($IN$).

1.1 Consider a hyperelliptic curve $H_p(a, b)$ over a finite field $\mathbb{F}_p$, where $p$ is a large prime of $k$ bits. Let $D$ be a divisor on the hyperelliptic curve $H_p$, and let $\beta$ be the prime order of the hyperelliptic curve.

A random value $\alpha_r \in (1, \beta)$ is chosen, which acts as the user's private key. Now the public key can be calculated by the help of equation.

$$Xu = \alpha_r.D \tag{4.2}$$

$(\alpha_r, Xu) =$ Keygen_HECC

1.2. $Xh$: HECDH shared secret key computation from the equation

$$Xh = Xu.\alpha_r \tag{4.3}$$

1.3. If $m \neq null$, then:

$(m_{\text{encrypt}}, i) = \text{encryption}(m, Xh, sT)$

Else

discard$(m)$

End if

1.4. $m_{\text{hash}} = \text{SHA2}(m_{\text{encrypt}})$

1.5. $(r, s) = \text{signing}(m_{\text{hash}}, \alpha_r, Xu)$

1.6. Forward $(m_{\text{encrypt}}, i, r, s)$

**Step 2** This step is performed at fog node $FN$

2.1 Get associated $Xu$

2.2 Check the $Xu$ validity

2.3 $m_{\text{hash}} = \text{SHA2}(m_{\text{encrypt}})$

2.4 $f = \text{verify}(m_{\text{hash}}, r, s, Xu)$

2.5 If $f == 1$, then:

Forward $(m_{\text{encrypt}}, i, r, s)$

Else

Discard $m_{\text{encrypt}}$ and break

End if

**Step 3** The computation is performed at both the cloudstorage($CS$) layer and the private blockchain($PB$).

3.1 Get associated $Xu$

3.2 Check the $Xu$ validity

3.3 $m_{\text{hash}} = \text{SHA2}(m_{\text{encrypt}})$

3.4 $f = \text{verify}(m_{\text{hash}}, r, s, Xu)$

3.5 If $f == 1$, then:

Forward $(m_{\text{encrypt}}, i, r, s)$

Else

Discard $m_{\text{encrypt}}$ and break

End if

**Stop**

The search data request procedure is intelligently built to protect against numerous security concerns and prevent unauthorized access by using a lightweight technique for signature formation and verification.

## 4.3 Secuirty analysis

In this section, the security attributes “confidentiality, Integrity, Authentication, Scalability , computational efficency, Attack resistance will be discussed.

### 4.3.1 Confidentiality

Hyperelliptic Curve Cryptography (HECC), which shields private data from unwanted access, guarantees confidentiality. Because of the difficulty of the Hyperelliptic Curve Discrete Logarithm Problem (HECDLP), HECC's encryption methods ensure the secure transfer and storage of data. Because of this, attackers will find it very difficult to decrypt the data without the private key. Confidentiality is achieved through encryption using the shared secret key $Xh$, derived from the Hyperelliptic Curve Diffie-Hellman (HECDH) key exchange. This occurs at $IN$, where the input gene data m is encrypted to produce $m_encrypt$ 4.2.1.

### 4.3.2 Integrity

Data integrity is maintained using digital signatures based on HECC. These signatures prevent unauthorized modification or tampering and provide a reliable means of verifying authenticity. Due to the hardness of HECDLP, such signatures are computationally resistant to forgery, ensuring the trustworthiness of transmitted data.

### 4.3.3 Authentication

Authentication 4.2.1 is provided through the secure generation and verification of keys using HECC. At step 1.1 at $IN$, the key pair $(\alpha_r, Xu)$ is generated by using HECC. Later, at steps 2.2 and 3.2 at $FN$ and $CS$ respectively, the validity $Xu$ is checked, ensuring authentication.

### 4.3.4 Scalability

HECC supports efficient key generation and encryption, enabling the system to manage large datasets and serve multiple users simultaneously. This scalability makes the proposed framework suitable for modern Healthcare 4.0 environments that process vast amounts of genomic data.

### 4.3.5 Computational Efficiency

HECC requires smaller key sizes compared to ECC for equivalent security levels, which reduces computational costs. Its efficient encryption and decryption mechanisms enable fast and secure data processing, making it appropriate for real-time healthcare applications where responsiveness is critical.

### 4.3.6 Attack Resistance

The framework provides resistance against various threats, including brute-force, side-channel, and quantum-based attacks. The reliance on HECDLP ensures that breaking the encryption is computationally infeasible, thereby guaranteeing the long-term security of genomic data.

# Chapter 5

# Conclusion

This thesis proposed a secure and scalable framework for genomic data protection in Healthcare 4.0 environments by integrating blockchain technology with lightweight cryptography. The architecture brings together edge devices, fog computing, cloud storage, and blockchain to ensure reliable data collection, efficient processing, scalable storage, and tamper-resistant data exchange. This layered approach addresses major challenges in healthcare data management, including confidentiality, integrity, authentication, and non-repudiation.

The novelty of the proposed system lies in its adoption of Hyperelliptic Curve Cryptography (HECC) in place of conventional Elliptic Curve Cryptography (ECC). HECC offers stronger security with shorter key sizes, making it particularly suitable for resource-constrained devices such as wearable sensors and IoT-based healthcare applications. This advantage reduces computational overhead, improves processing efficiency, and enhances security, thereby aligning with the real-time requirements of genomic data analysis and healthcare monitoring systems.

In addition, the framework incorporates an Intermediate Node (IN) that facilitates secure communication with blockchain nodes. The IN validates metadata, monitors access logs, and ensures trustworthy updates without revealing sensitive genomic information. This mechanism improves transparency, reliability, and auditability across the system while maintaining privacy. The proposed hybrid

cryptographic method based on HECC also enables secure searches within encrypted datasets, ensuring that sensitive information remains confidential while still supporting legitimate data queries.

The strength of the proposed system is supported by theoretical arguments. The security of HECC is based on the hardness of the hyperelliptic curve discrete logarithm problem, which provides a stronger mathematical foundation compared to ECC while requiring smaller key sizes. These features position HECC as a promising cryptographic technique for modern healthcare systems where both performance and security are equally critical.

This research contributes in three key ways:

1. It presents a layered Healthcare 4.0 architecture that integrates edge, fog, cloud, and blockchain for secure genomic data processing.
2. It demonstrates the theoretical advantages of adopting HECC over ECC, emphasizing smaller key sizes and reduced computational costs.
3. It introduces mechanisms for secure searching and auditing of genomic data without exposing sensitive information.

Although this work is largely theoretical, it sets a foundation for practical implementation. Future work may focus on developing prototypes to validate the framework in real-world environments, benchmarking HECC against other cryptographic methods in terms of speed and resource consumption, and extending the model to cover broader categories of healthcare data. Further exploration of integration with artificial intelligence (AI) and machine learning could also enhance data analysis while maintaining strong security guarantees.

# Bibliography


[1] W. G. Barker, *Introduction to the analysis of the Data Encryption Standard (DES)*. Aegean Park Press, 1991.

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

[3] F. Callegati, W. Cerroni, and M. Ramilli, "Man-in-the-middle attack to the https protocol," *IEEE Security & Privacy*, vol. 7, no. 1, pp. 78–81, 2009.

[4] D. G. Cantor, "Computing in the jacobian of a hyperelliptic curve," *Mathematics of computation*, vol. 48, no. 177, pp. 95–101, 1987.

[5] L. Chen, W.-K. Lee, C.-C. Chang, K.-K. R. Choo, and N. Zhang, "Blockchain based searchable encryption for electronic health record sharing," *Future generation computer systems*, vol. 95, pp. 420–429, 2019.

[6] 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.

[7] W. Diffie, "New direction in cryptography," *IEEE Trans. Inform. Theory*, vol. 22, pp. 472–492, 1976.

[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] S. Fugkeaw, "A fine-grained and lightweight data access control model for mobile cloud computing," *IEEE Access*, vol. 9, pp. 836–848, 2020.

[10] C. Hsu, L. Harn, Z. Xia, L. Bai, and Z. Zhang, "Information-theoretic secure rational secret sharing in asynchronous networks for untrusted cloud environments," *Journal of Cloud Computing*, vol. 11, no. 1, p. 89, 2022.

[11] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ecdsa)," *International journal of information security*, vol. 1, pp. 36–63, 2001.

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

[13] D. W. Kravitz, "Digital signature algorithm," Jul. 27 1993, uS Patent 5,231,668.

[14] R. R. L., 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.

[15] X. Liu, Z. Wang, C. Jin, F. Li, and G. Li, "A blockchain-based medical data sharing and protection scheme," *IEEE Access*, vol. 7, pp. 118 943–118 953, 2019.

[16] A. A. M., "Advanced encryption standard (AES) algorithm to encrypt and decrypt data," *Cryptography and Network Security*, vol. 16, no. 1, p. 11, 2017.

[17] H. Mahajan and K. Reddy, "Secure gene profile data processing using lightweight cryptography and blockchain," *Cluster computing*, vol. 27, no. 3, pp. 2785–2803, 2024.

[18] H. B. Mahajan, A. Badarla, and A. A. Junnarkar, "Cl-iot: cross-layer internet of things protocol for intelligent manufacturing of smart farming," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 7, pp. 7777–7791, 2021.

[19] D. Manel, O. Raouf, H. Ramzi, and A. Mtibaa, "Hash function and digital signature based on elliptic curve," in *14th International Conference on Sciences and Techniques of Automatic Control & Computer Engineering-STA'2013*. IEEE, 2013, pp. 388–392.

[20] D. Marbouh, T. Abbasi, F. Maasmi, I. A. Omar, M. S. Debe, K. Salah, R. Jayaraman, and S. Ellahham, "Blockchain for covid-19: review, opportunities, and a trusted tracking system," *Arabian journal for science and engineering*, vol. 45, pp. 9895–9911, 2020.

[21] H. S. H. M.Ayachi, Nacer, "Cooperative game approach to form overlapping cloud federation based on inter-cloud architecture," *Cluster Computing*, vol. 24, pp. 1551–1577, 2021.

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

[23] R. C. Merkle, "A certified digital signature," in *Conference on the Theory and Application of Cryptology.* Springer, 1989, pp. 218–238.

[24] E. Milanov, "The rsa algorithm," *RSA laboratories*, vol. 1, no. 11, 2009.

[25] A. Mousa and A. Hamad, "Evaluation of the rc4 algorithm for data encryption." *Int. J. Comput. Sci. Appl.*, vol. 3, no. 2, pp. 44–56, 2006.

[26] M. A. Musa, E. F. Schaefer, and S. Wedig, "A simplified aes algorithm and its linear and differential cryptanalyses," *Cryptologia*, vol. 27, no. 2, pp. 148–177, 2003.

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

[28] J. Pelzl, T. Wollinger, J. Guajardo, and C. Paar, "Hyperelliptic curve cryptosystems: Closing the performance gap to elliptic curves," in *Cryptographic Hardware and Embedded Systems-CHES 2003: 5th International Workshop, Cologne, Germany, September 8–10, 2003. Proceedings 5.* Springer, 2003, pp. 351–365.

[29] C. Pirtle and J. Ehrenfeld, "Blockchain for healthcare: The next generation of medical records?" *Journal of Medical Systems*, vol. 42, no. 9, p. 172, 2018.

[30] V. K. Quy, N. V. Hau, D. V. Anh, N. M. Quy, N. T. Ban, S. Lanza, G. Randazzo, and A. Muzirafuti, "Iot-enabled smart agriculture: architecture, applications, and challenges," *Applied Sciences*, vol. 12, no. 7, p. 3396, 2022.

[31] C. Ran., S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption," in *Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23*. Springer, 2004, pp. 207–222.

[32] R. L. Rivest, "The rc5 encryption algorithm," in *International Workshop on Fast Software Encryption*. Springer, 1994, pp. 86–96.

[33] J. R. Shaikh, M. Nenova, G. Iliev, and Z. Valkova-Jarvis, "Analysis of standard elliptic curves for the implementation of elliptic curve cryptography in resource-constrained e-commerce applications," in *2017 IEEE international conference on microwaves, antennas, communications and electronic systems (COMCAS)*. IEEE, 2017, pp. 1–4.

[34] A. Sharma, S. Bahl, A. K. Bagha, M. Javaid, D. K. Shukla, and A. Haleem, "Blockchain technology and its applications to combat covid-19 pandemic," *Research on Biomedical Engineering*, pp. 1–8, 2020.

[35] B. Shen, J. Guo, and Y. Yang, "Medchain: Efficient healthcare data sharing via blockchain," *Applied sciences*, vol. 9, no. 6, p. 1207, 2019.

[36] J. H. Silverman, *The arithmetic of elliptic curves*. Springer, 2009, vol. 106.

[37] R. Singh and S. Kumar, "Elgamal's algorithm in cryptography," *International Journal of Scientific & Engineering Research*, vol. 3, no. 12, pp. 1–4, 2012.

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

[39] V. Svetlana, S. De. Sutter, S. Iliopoulos, D. Aggelis, and T. Tysmans, "Experimental structural analysis of hybrid composite-concrete beams by digital image correlation (dic) and acoustic emission (ae)," *Journal of Nondestructive Evaluation*, vol. 35, pp. 1–10, 2016.

[40] M. Swan, *Blockchain: Blueprint for a new economy.* " O'Reilly Media, Inc.", 2015.

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

[42] T. T. Thwin and S. Vasupongayya, "Blockchain-based access control model to preserve privacy for personal health record systems," *Security and Communication Networks*, vol. 2019, no. 1, p. 8315614, 2019.

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

[44] H. C. Van Tilborg and S. Jajodia, *Encyclopedia of cryptography and security.* Springer Science & Business Media, 2014.

[45] W. C. Vu. Minh. A. Headley and K. P. Heaslip, "A comparative overview of automotive radar spoofing countermeasures," in *2022 IEEE International Conference on Cyber Security and Resilience (CSR).* IEEE, 2022, pp. 245–252.

[46] B. Wang and L. Li, "Research progress in biomedical big data," *Progress in China Epidemiology: Volume 1*, pp. 391–400, 2023.

[47] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "Medshare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE access*, vol. 5, pp. 14 757–14 767, 2017.

[48] Q. Xia, E. B. Sifah, A. Smahi, S. Amofa, and X. Zhang, "Bbds: Blockchain-based data sharing for electronic medical records in cloud environments," *Information*, vol. 8, no. 2, p. 44, 2017.

[49] G. Yang, C. Li, and K. E. Marstein, "A blockchain-based architecture for securing electronic health record systems," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 14, p. e5479, 2021.

[50] A. Zhang, L. Wang, X. Ye, and X. Lin, "Light-weight and robust security-aware d2d-assist data transmission protocol for mobile-health systems," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 3, pp. 662–675, 2016.

[51] K.-J. Zhang, W.-W. Zhang, and D. Li, "Improving the security of arbitrated quantum signature against the forgery attack," *Quantum information processing*, vol. 12, pp. 2655–2669, 2013.

[52] L. Zhang, M. Peng, W. Wang, Z. Jin, Y. Su, and H. Chen, "Secure and efficient data storage and sharing scheme for blockchain-based mobile-edge computing," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 10, p. e4315, 2021.