CAPITAL UNIVERSITY OF SCIENCE AND
TECHNOLOGY, ISLAMABAD

Double Parameter Hyperelliptic Curve Digital Signature Scheme for Blockchain Security

by

Jahanzaib Tariq

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

in the

Faculty of Computing
Department of Mathematics

2025

CERTIFICATE OF APPROVAL

Double Parameter Hyperelliptic Curve Digital Signature Scheme for Blockchain Security

by

Jahanzaib Tariq

(MMT223009)

THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Organization
(a)	External Examiner	Dr. Atta Ullah	NUTECH, Islamabad
(b)	Internal Examiner	Dr. M. Sabeel Khan	CUST, Islamabad
(c)	Supervisor	Dr. Rashid Ali	CUST, Islamabad

Dr. Rashid Ali
Thesis Supervisor
November, 2025

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

Dr. M. Abdul Qadir
Dean
Faculty of Computing
November, 2025

Author's Declaration

I, **Jahanzaib Tariq**, hereby state that my MPhil thesis titled "**Double Parameter Hyperelliptic Curve Digital Signature Scheme for Blockchain Security**," is my 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.

(Jahanzaib Tariq)

Registration No. MMT223009

Plagiarism Undertaking

I solemnly declare that the research work presented in this thesis titled "**Double Parameter Hyperelliptic Curve Digital Signature Scheme for Blockchain Security**", 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 awarding 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.

(Jahanzaib Tariq)

Registration No. MMT223009

Acknowledgement

I am grateful to Almighty ALLAH for providing me with a loving family, outstanding teachers, and the opportunity to complete my dissertation.

I am grateful to my supervisor, Dr. Rashid Ali, for his consistent motivation. He was always there whenever I found any problem. I really appreciate his efforts and guidance during my thesis and pleased to a student of such kind supervisor.

I sincerely thank 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 delivering such valuable and clear lectures.

I am thankful to the management team at Capital University of Science and Technology, Islamabad, for creating a supportive and welcoming environment for learning.

I am truly grateful to all my family members for being a strong source of support and patience during my research. Above all, I owe my deepest thanks to my parents for their sincere prayers, constant love, and unwavering support in helping me complete my degree. They have always stood by me and encouraged me throughout my life.

Finally, I am grateful to everyone who prayed for me, shared their knowledge, and supported me throughout my degree program.

(Jahanzaib Tariq)
Registration No: MMT223009

Abstract

The traditional elliptic curve digital signature algorithm (ECDSA) requires an inversion operation during both signing and verification, which lowers its efficiency. To address this limitation, the present study proposes employing two random values instead of a single one. This dual-parameter strategy strengthens the security of digital signatures and improves resistance against potential attacks. The same concept is extended to hyperelliptic curve cryptography (HECC), an advanced variant of elliptic curve cryptography that operates on genus 2 curves. HECC provides robust security with comparatively shorter key lengths, making it well suited for applications such as blockchain where storage and computational resources are constrained. In the proposed scheme, the structural properties of hyperelliptic curves are utilized to construct a signature mechanism based on two random parameters. Although HECC is inherently more complex, recent algorithmic developments have enhanced its speed and practicality. The findings indicate that integrating dual parameters into HECC can yield improved security and efficiency for digital signatures, particularly in contemporary technologies like blockchain.

Contents

List of Figures

List of Tables

Abbreviations

AES	Advance Encryption Standard
DES	Data Encryption Standard
DLP	Discrete Logarithm Problem
ECC	Elliptic Curve Cryptography
HECC	Hyper Elliptic Curve Cryptography
HECDLP	Hyperelliptic Curve Discrete Logarithm Problem
RSA	Rivest Shamir Adleman

Symbols

E	Elliptic Curve
$HECC$	Hyperelliptic Curve
c	Ciphertext Message
n	Starting Point
h	Cofector
D	Divisor
$\mathbb{F}_p$	A Discrete Prime Field

Chapter 1

Introduction

In today's rapidly evolving digital landscape, protecting messages and sensitive information from unauthorized access and potential threats has become increasingly complex [1]. Ensuring that communication remains private and inaccessible to unintended recipients is a significant challenge. As the reliance on digital services and the exchange of electronic information continues to grow, the need for robust information security is more critical than ever [2].

Cryptography is essential for ensuring communication security. It enables individuals to transmit messages securely, making sure that only the right person can read the message. The first reputed usage of cryptography dates back to roughly 1900 BCE [3].

Basically, cryptography means keeping communication safe and private. Modern cryptographic techniques are designed to protect data from a wide range of threats. This process usually changes readable information into a secret code called cipheryext using a mathod known as cipher [4].

The locked message is then delivered to the person it is meant for. Without decryption key or any indications, it is almost hard for illegal people to decipher the original text. The secret key is known only to the sender and receiver, and it lets them access and read the original message [5]. This ensures that the communication remains confidential.

In addition to confidentiality, cryptography also provides other essential security features such as data integrity, authentication, and non-repudiation. To ensure these properties, it is imperative to use strong, reliable encryption algorithms [6]. Cryptography is the study and practice protecting knowledge by transforming in a secure structure.

It facilitates discreet conversation in the presence of third persons or possible opponents. The main purpose of cryptography is to ensure that information can be accessed and understood only by those for whom it is intended. Mathematical techniques are used to change the data into an unreadable code called ciphertext, and anyone with the right key can convert it back to its original form, called plaintext.

1.1 Literature Review

Blockchain technology [7] has greatly changed how businesses and industries work by using cryptography to make sure that transactions can be tracked, cannot be changed or faked, and that no one can deny making them.

Before elliptic curve cryptography (ECC) [8] became widely used, most public key methods such as RSA [9], DSA [10], which use modular arithmetic for encryption. Most cryptocurrency [11] and blockchain platforms use the ECDSA with a specific curve called secp256k1 [12].

Unlike most commonly used curves that are randomly structured, secp256k1 has a special non-random design that allows for faster and more efficient computations. Frequent cases of bitcoin theft on major trading platforms, along with anonymous transactions, have made people worry about the security of blockchain systems.

Bitcoin is built on traditional cryptography and distributed system methods [13] to avoid needing trusted third parties for online transactions. It uses tools like proof-of-work, protection against sybil attacks [14], the ECDSA, consensus methods, and Merkle tree proof.

One problem with bitcoin is that it uses weak randomness in its ECDSA system. Bellare et al [15] supported the use of weak randomness in digital signatures. The industry has raised two major mathematical issues concerning elliptic curve digital signatures (EDCSA), first is inverse operations and second is scalar multiplication.

According to literature [16], inverse operations take ten times longer than multiplication, although scalar multiplication [17] takes less time. To save time, the inverse operation is given priority over multiplication, even though it usually needs to be repeated at least once.

Researchers in the industry have created better methods for performing inverse and scalar multiplication operations. To improve speed, new algorithms that avoid using inverse operations and instead use scalar and modular multiplication have been introduced in recent research [18], [19].

The shortcomings of the two procedures indicated above have been emphasized in the literature [20], [21]. While lowering the inverse technique improves computation performance, it can result in forged signatures and does not provide forward security. Literature [22] aims to fix security weaknesses found in digital signature systems that use elliptic curves.

Research [23] proposes a safe and efficient method for creating ECDSA signatures between two parties. Literature [24] introduced a shared elliptic curve digital signature method that works without needing a trusted central authority. However, the shared parts in our method might not work well in some situations. Literature [25] looks at how hamming distance is connected to hash values. Literature [20] highlighted the drawbacks of the aforementioned two approaches. While decreasing the inverse procedure improves computing performance, it can lead to forged signatures and does not ensure forward security. Literature [26] proposes an elliptic curve digital signature technique capable of recovering messages. This approach is resistant to forged signature attacks and provides forward security.

Literature [23] focuses on addressing security vulnerabilities in elliptic curve based digital signature schemes. In the literature [24], a threshold elliptic curve digital signature scheme was proposed that does not rely on any trusted center. However,

our scheme's shared fragments may not be suitable for certain cases. The study [27] talks about hoe hamming distance and hash values are connected.

We present a double parameter elliptic curve digital signature scheme that simplifies verification, improves security, and effectively resists weak randomness attacks. Replacing the hash value with the hamming weight of the hash function in signatures decreases scalar multiplications and increases the efficiency of the signature.

ECC has various benefits including excellent security, reduced storage space requirements, and affordable bandwidth costs. In 1989, Koblitz [28] suggested using hyperelliptic curves instead of regular elliptic curves to create encryption methods for solving the DLP.

Hyperelliptic curves are the extended forms of elliptic curves. Hyperelliptic curves [29] have a major benefitthey require only a very small key size. To achieve the same security level, hyperelliptic curves use a smaller finite field compared to elliptic curves.

1.2 Thesis Contribution

In this thesis, double parameter ECDSA scheme of Liu and Chen [30], is reveiwed. Liu, and Chen contributes significantly to the area of cryptographic, notably in cycle of improving digital signature techniques for blockchain applications. ECDSA has performance constraints because of the inversion operation required in both the signature and verification operations. To overcome these difficulties, they suggest a new ECDSA variant that uses double parameters during the signature creation process.

This method improves the randomness generation process, making it more resistant to forgery and nonce reuse attacks, which are widespread in blockchain contexts. This method offers strong security because it relies on the difficulty of solving the ECDLP. To send secure and verified messages, system uses ECC, which provides digital security with low processing power. This method, unlike

other cryptosystems, ensures the message is safe, can not be changed or faked, stays private, proves who sent it, can be checked for accuracy, and the sender can nott deny sending it. It works faster and uses shorter keys.

This scheme is further extended by using hyperelliptic curve. Hyperelliptic curves are a generalization of elliptic curves and are particularly usefull in environments with limited computational resources. The proposed approach enhances the resistance of current methods to hacking and cryptographic attacks. By employing hyperelliptic curves instead of elliptic curves, the scheme attains an equivalent security level with reduced key sizes, leading to faster operation and lower computational overhead. [31]

1.3 Thesis Structure

In **Chapter 2**, the mathematical models associated with our concept are addressed, including basic definitions of cryptography, digital signatures, and blockchain.

In **Chapter 3**, a detailed examination of the digital signature algorithm technique developed by Liu and Chen using elliptic curve cryptography is presented. The security analysis of the system is thoroughly explored.

In **Chapter 4**, work of Liu and Chen hyperelliptic curve double parameter method is introduced. The correctness of this method is also examined. A toy example demonstrating the technique is also provided. A detailed analysis of the generalized double parameter ECDSA method is provided. Computational cost is reported and compared to other schemes.

In **Chapter 5**, conclusions and future directions are discussed.

Chapter 2

Preliminaries

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

2.1 Mathematical Background

Definition 2.1.1. "A non empty set 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: $a * b \in G$
2. Associativity: $(a * b) * c = a * (b * c)$
3. Identity Element: There is an identity element $e \in G$ such that for all $a \in G$,

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

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

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

Then a' is called inverse of a

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

$$a * b = b * a$$

for all $a, b \in G$" [32].

Example 2.1.2. Here are several examples that demonstrate the concept of group:

1. The real numbers $\mathbb{R}$ make a group when you combine them using addition.
2. The set $\mathbb{R}$ also forms a group when using multiplication of real numbers.
3. Set of integers $\mathbb{Z}$ is a group for adding integers but not for multiplying integers.
4. The $\mathbb{Z}_n$ of integers modulo n is a group for integer addition modulo n, but not for multiplication modulo n when n is not prime. That example, $\mathbb{Z}_p$ is a group formed by multiplying integers modulo p.

Definition 2.1.3. "A set $\mathbb{F}$ under two binary operations $(+, *)$ is known as a field $(F, +, *)$, if the following axioms are satisfied [33]:

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

$$a \cdot (b + c) = a \cdot b + a \cdot c". \tag{2.1}$$

Example 2.1.4.

Consider the following examples to understand the concept of a field:

1. The set of rational numbers $\mathbb{Q}$ forms a field under the usual addition and multiplication of rational numbers.

2. The collection of complex numbers $\mathbb{C}$ makes a field when we use addition and multiplication, where 0 is the identity for addition and 1 is the identity for multiplication.

3. The set of integers $\mathbb{Z}$, however, is not a field because most integers do not have multiplicative inverses within $\mathbb{Z}$.

Definition 2.1.5. "In integer arithmetic, if we divide a by n, we can get a quotient q and a remainder r. The relationship between these four integers can be shown as

$$a = q \times n + r$$

In this relation, a is called the dividend, q the quotient, n the divisor, and r the remainder. Note that this is not an operation because the result of dividing a by n is two integers, q and r. We can call it a division relation."[34]

Definition 2.1.6. "One way trapdoor function is a one way function $\mathbb{F}$, in which it is feasible to find the value in one way but it is infeasible to find it in the opposite way" [35].

$$x + y = a \tag{2.2}$$

In the above equation, if you know the value of x , it is easy to figure out y. But if you only know y, its very hard to find out what x from y, but given y it is infeasible to find the value of x.

Definition 2.1.7. "Suppose g is the generator of Zp where p is the prime number. Finding x is difficult when y is known. i.e. Computing x from $y = gx$ mod p. The process of finding the x is known as Discrete Logarithmic Problem [36]. Solving a discrete logarithm problem is hard."

Definition 2.1.8. Factorization of a number means expressing it as a product of its prime factors. For example, if a number m can be written as the product of two prime numbers a and b, then $m = ab$. The integer factorization problem can be formally described as follows: given a composite number n, the task is to find

two integers x and y such that their product satisfies $xy = n$. Determining the factors of large numbers is a challenging process.[34]

Definition 2.1.9. A number system that operates on integers is called modular arithmetic. This system is based on the concept of congruence relation. If p is taken as the modulus in a binary operation, then the expression b mod p represents the remainder obtained when b is divided by p. [37]

$$a \equiv b \pmod{p}$$

Algorithm 2.1.10. "An Extended Euclidean Algorithm is used to find modular inverses [38]. The steps for finding the inverse of a and b are as follows:

Input: $a \mod p$

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

1. Initialize: $(u, v, w) = (1, 0, p)$, $(r, s, t) = (0, 1, a)$
2. If $t = 0$; return $gcd(a, p) = w$, There is no inverse for a mod p.
3. If $t = 1$; return $gcd(a, p) = t$ and $s = a$ Now find the quotient Q using: $Q = w$ divided by t.
4. $(b_1, b_2, b_3) = (u - Qr,\ v - Qs,\ w - Qt)$
5. $(u, v, w) = (r, s, t)$
6. $(r, s, t) = (b_1, b_2, b_3)$
7. Go to step 2

Definition 2.1.11. A trapdoor function is a special type of mathematical function that can be easily computed in one direction but is hard to invert unless certain secret information, known as the trapdoor. [39].

Which functions are not trapdoor?
Suppose we define a function $x + y = a$. In this case, if x and y are given than

a can be computed easily. However, it is also easy to x and y if any two of three values are given.

Therefore this is not a trapdoor function because a trapdoor function requires some special knowledge to make the reverse computation feasible. Without such special knowledge, the reverse direction should be computationally hard.

Definition 2.1.12. "A Galois Field, also known as a finite field, is a mathematical system with a limited number of elements in which you can perform addition, subtraction, multiplication, and division (excluding division by zero), and the result always remains within the set. These operations follow the standard rules of arithmetic". [40]

For the field $\mathbb{GF}(13)$, the operation for addition is shown in the table given below:

TABLE 2.1: Addition in $\mathbb{GF}(13)$

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

2.2 Cryptographic Background

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

2.2.1 Cryptology

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

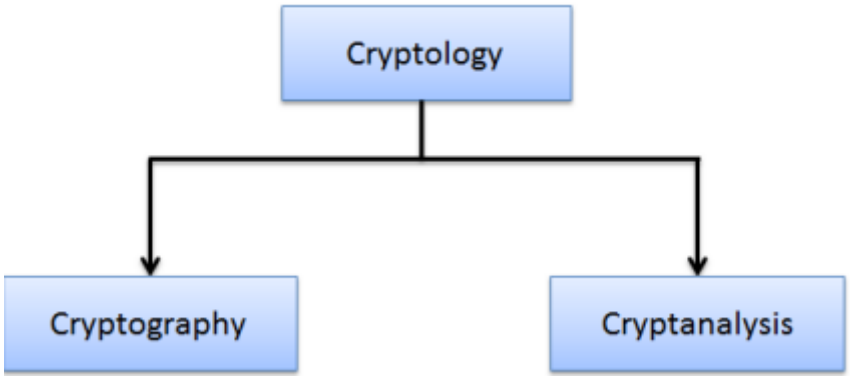

FIGURE 2.1: Cryptology

2.2.2 Cryptography

The term cryptography derived from Greek words "kryptos" (meaning hidden) and "graphein" (meaning to write). It refers to the secure writing of sensitive information. To safeguard data from potential attacks during transmission, messages are transformed into an unreadable format. In this system, sender encrypts message, while receiver uses decryption to retrieve the original content. Encryption converts plain text into an unintelligible form, and decryption restores it back to its original state. Cryptography provides key security features like confidentiality, integrity, authenticity, and non-repudiation.

There are five components of cryptosysytem:

1. Message space
2. Ciphertext space
3. Key space

4. Encryption algorithm space

5. Decryption algorithm space

For further details see [41].

2.2.2.1 Objectives of Cryptography

Cryptography gives various key objectives that are necessary to modern information security:

Confidentiality: Ensures that data is applicable only to those authorized to prove it.

Integrity: It gives surity that the data is altered or tampered with during transmission.

Authentication: Verifies who is taking part in the communication.

Repudiation: It protects against any party denying their actions, such as sending a message or signing a document.

2.2.2.2 Components of Cryptography

Plaintext: It is that data which needs to be protected.

Ciphertext: The version which is encypted of the plaintext, which is not readable without a decryption key.

Encryption Algorithm: It is a method used to change readable text into a secret code.

Decryption Algorithm: A method which is used to revert ciphertext back to plaintext.

Key: A piece of information that is used during the encryption and decryption processes.

There are types of cryptography:

1. Symmetric Key Cryptography
2. Asymmetric Key cryptography

2.2.3 Symmetric Key Cryptography

Symmetric key cryptography [42] is also called secret key cryptography. Both parties use only one key for both process known as encryption and decryption in symmetric key cryptography. Both parties shift or interchange the key before the transmission of information. Given a message (plaintext) and the key, encryption process produces unintelligible data, which is about the same length as the plaintext was [43]. Decryption process is the reverse of encryption process.

There are two methods of symmetric key cryptogrphy:

1. Data Encryption Standard [44]
2. Advanced Encryption Standard [45]

2.2.4 Asymmetric Key Cryptography

Diffie and Hellman [46] created asymmetric key cryptography in 1976 to tackle the key exchange issue. In asymmetric key cryptography, each user has two keys; public key and private key.Everyone has access to the public key, while the private key is kept private. If one key encrypts, the other key will decrypt. Encryption [47] is done using a public key and decryption with a private key. Only those who have access the private key can decipher the message that is encrypted.

Some of the following cryptographic tchniques are:

1. Digital Signature Algorithm(DSA) [48]
2. Rivest-Shamir Adleman(RSA) [49]

3. ElGamal Cryptosystem [50]

4. Elliptic Curve Cryptosystem [51]

2.2.5 Applications of Cryptography

Cryptography has numerous real-world uses, such as secure communication for example email encryption. Digital commerce for example, save credit card transactions. Banking system for example,save internettransactions and PINs. Government and military for example,protecting confidential messages. Blockchain currency such as Bitcoin uses cryptographic hashing and digital signatures. Digital signatures and certificates enable to authenticate identities.

2.3 Hash Function

In cryptography, the hash function [52] is very important tool. This idea has been widely recognized in computer science for many years. In a hash function, a random length bit string is converted into a fixed length value. This value is called as the hash value. Hash values are often called as hash codes, digests, or just hashes. A good hash function makes sure that even small changes in the input create a completly different output. The hash function provides the property of integrity and uniqueness. A one way trapdoor function is used to generate a hash value, which is easy to compute. However, recalculating the input from the generated hash value is complex. The hash function provides security and efficiency. Hashes are useful for transmitting files or data and provide integrity. While giving files or messages, the recipient can verify the message's integrity by receiving a hash value. The hash function gives surity for integrity in digital signatures.

2.3.1 Properties of Hash Function

The hash function has the following properties:

1. It is quite simple to calculate hash value of any text.

2. Finding the inverse value of a hash function, $r = h(m)$, is likely challenging.

3. In the given equation, obtaining r is easy, however it is impossible to get m when r is known.

4. Each hash value must be unique. Output should vary based on the input value.

5. It should offers the property of integrity. No one can forge the value of the message.

   It should be nearly impossible to find two different inputs that create the same hash value.

6. The output should be of set length.

7. Each hash value must be unique. Output should vary based on the input value.

8. A slight change in the message should change the whole output value.

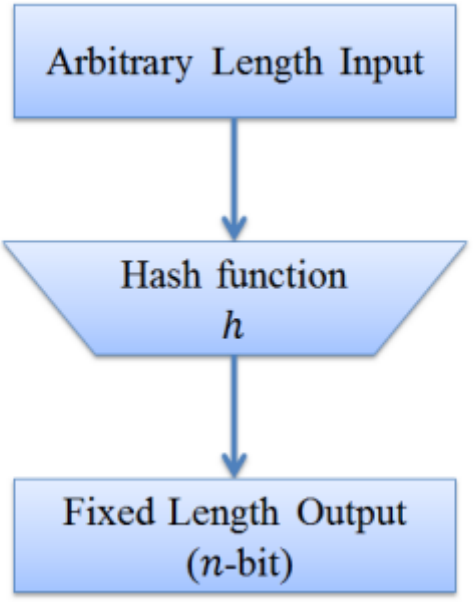

FIGURE 2.2: Hash Function

2.4 Cryptanalysis

For analyzing the strength of a scheme, cryptanalysis schemes are used. In addition, for breaking a system, cryptanalysis [53] is done to find the weaknesses of the scheme. Ciphers, codes, and encrypted text can be studied and decrypted using a method known as cryptanalysis, which involves analyzing and exploiting weaknesses in cryptographic techniques. In short, cryptanalysis is the method in which we find out the plaintext without knowing the decryption key [54]. The attacks will be successful only if the scheme is weak or is vulnerable to different attacks.The purpose of cryptanalysis is to gather as much information as possible about the plaintext. This allows a cryptanalyst [55] to break the system and obtain the ciphertext and keys, which may then be used to decrypt other message. Cryptanalysis uses mathematical techniques toanalyze and break cryptographic systems. Cryptanalysis is required for determining the security of cryptographic systems used in digital applications such as communication, data protection, and online transactions. Cryptanalysis focuses on unravelling the secrets concealed within encrypted information.

Cryptanalysis attacks can be classified according to the sort of information that the attacker has access to. Cryptoanalysis attacks [56] test the susceptibility of various schemes. Cryptanalysis is like attempting to solve a secret puzzle without knowing all of the components.

2.5 Digital Signature

Digital signatures [57] mimic digital fingerprints. A digital signature confirms the message was sent by a confirmed sender. They protect the message's confidentiality, ensuring that it is not altered during transmission. A digital signature is a mathematical code appended to the message to be signed. When someone digitally signs a document or message, the algorithm creates a unique digital signature that matches the content. The concept of digital signature was first introduced

by Diffie and Hellman in 1976. Authors propose a one way trapdoor function to explain the working of this scheme.

The following are the components of the digital signature scheme.

1. A security parameter k, selected by the user which determines the numbers (signatures length, signable messages length, computing time of the signing algorithm etc).
2. A message space M, a collection of messages that need to be signed.
3. A signature bound B is an integer that limits the signature generated by the method.
4. A key generation algorithm by which each user can generate both public and private keys.
5. A signature algorithm generates the signature for a message M
6. $V$is a verification algorithm that confirms the authenticity of a message.

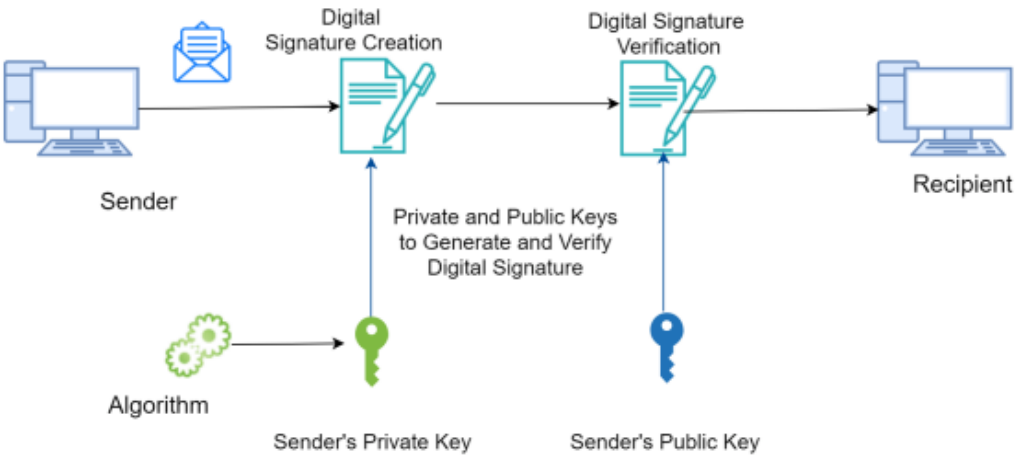

FIGURE 2.3: Digital Signature

The digital signatures addresses the following security parameters:

1. Authenticity: Digital signatures validate the message source's identity. A real signature proves that the message was senr by correct person.

2. Integrity: Integrity implies that the message is not altered during transmission. Even minor changes to a message during transmission can render a digital signature invalid.

3. Non repudiation: Non repudation means the person who signed somethong can not claim later that they did not sign it.

4. Unforgeability: Only the person who is supposed to sign can create a valid signature.

2.6 Elliptic Curve Cryptography

In 1985, Elliptic Curve Cryptography (ECC) [51] was separately introduces by Miller and Koblitz. ECC has the advantage of using fewer parameter sizes than ElGamal and RSA. Longa and Miri [58] created a flexible method to make calculating points of elliptic curve using only a limited set of numbers.

This adaptable method improves parallel processes for example 160-bit separated from scalar multiplication and parallel SIMD (Single Instruction Multiple Data) process, decreasing computing costs by 63% to 70%. Bailey [59] proposed an efficient method for working with elliptic curve arithmetic using finite fields. Rao and Setty [60] proposed two approaches for mapping alphanumeric characters to the xy coordinates of an elliptic curve.

The ciphertext is broadcast over public channel as an elliptic curve point. The proposed technique uses two secret keys to improve security when compared to one secret key-based encryption scheme. In their approach, even if the sender's long-term private key is compromised, an adversary cannot access the message without the second secret key. Athena et al. [61] proposed a new elliptic curve-based approach to secure personal health records. The suggested technique uses the elliptic curve Diffie-Hellman key exchange protocol to generate secret keys and identity-based encryption to secure cloud data. They analyzed the proposed strategy and found it to be both secure and efficient.

He et al. [62] proposed a novel elliptic curve-based certificateless encryption technique for multiple recipients. The suggested network is more efficient than exitesented a new system for secretly transferring content. The technique utilizes data concealment and secret sharing methods to create a letter based secret sharing scheme.

Avci [63] presented a new system for secretly transferring content. The technique utilizes data concealment and secret sharing methods to create a letter-based secret sharing scheme. Li et al.

[64] developed an anonymous user technique for the industrial internet of things (IIoT) using elliptic curves. Their study shows that the suggested method is both safe and effictive for industrial internet of things.

The general equation of elliptic curive is:

$$y^2 = x^3 + \alpha x + \beta$$

2.6.1 Elliptic Curve Over Real Field

In 1985, Victor Miller and Neal Koblitz each came up with elliptic curve cryptography [51] on their own.the points that lie on an elliptic curve using real numbers follow this equation:

$$y^2 = x^3 + \alpha x + \beta$$

With an additional point O at infinity, as described in Figure (2.4), α and β belong to R and have the condition

$$4\alpha^3 + 27\beta^2 \neq 0$$

The set E contains all points (x, y) that fulfill the above equation , which constitutes the elliptic curve group.

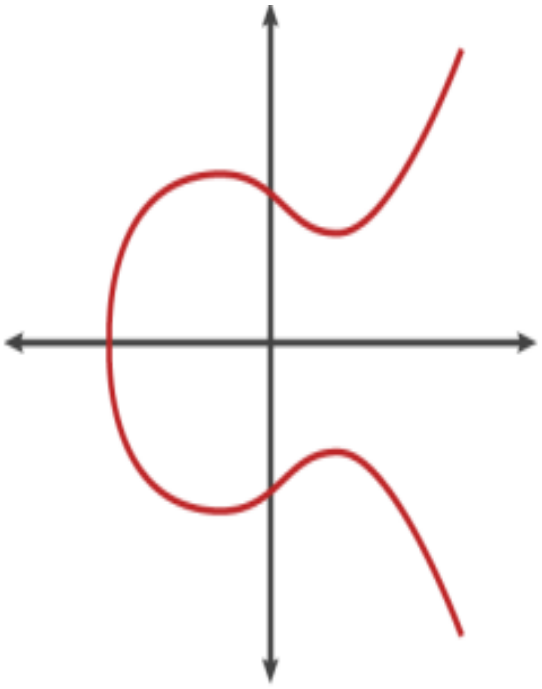

FIGURE 2.4: Elliptic Curve

The next part describes the addition operation of points on EC.

2.6.1.1 Addition of Points on Elliptic Curve

Consider two unique points P and R that lie on the elliptic curve E.

The addition of points P and R is indicated by Q and can be evaluated as follows:

1. Make a straight line from points P and R to a third point on elliptic curve E.

2. Adding two points on an elliptic curve gives a third point that is reflected across the x-axis.

Q represents sum of points $P$and R. Its coordinates are determined as follows:

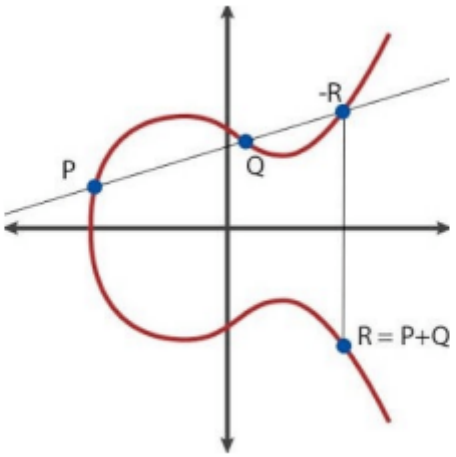

FIGURE 2.5: ECC Point Addition

Given the equations:

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

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

$$\text{where} \quad m = \frac{y_2 - y_1}{x_2 - x_1}$$

2.6.1.2 Elliptic Curve Point Doubling

Let E be an elliptic curve defined over a field as:

$$E : y^2 = x^3 + ax + b$$

Let $P = (x_1, y_1) \in E$. The process of point doubling $2P = P + P = (x_3, y_3)$ is as follows:

1. Find the slope of the line that just touches the curve at point P:

$$\lambda = \frac{3x_1^2 + a}{2y_1}$$

2. Compute the coordinates of $2P$:

$$x_3 = \lambda 1^2 - 2x_1$$

$$y_3 = \lambda 1(x_1 - x_3) - y_1$$

3. Thus, doubling a point gives:

$$2P = (x_3, y_3)$$

Let $P = (x_1, y_1)$ be a point on the elliptic curve

$$E : y^2 = x^3 + ax + b$$

When a point is added to itself the result is called $Q = 2P = (x_2, y_2)$, and its coordinates are calculating using the following method:

$$m = \frac{3x_1^2 + a}{2y_1}$$
$$x_2 = m^2 - 2x_1$$
$$y_2 = m(x_1 - x_2) - y_1$$

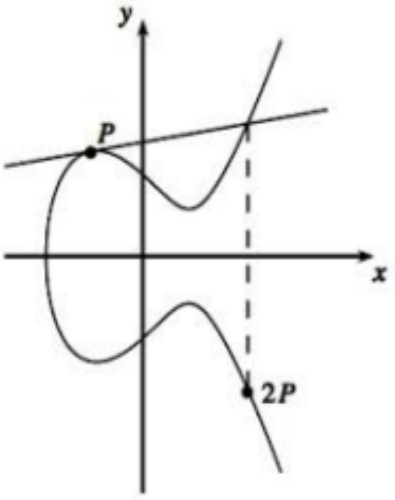

FIGURE 2.6: ECC Point Doubling

2.6.2 Elliptic Curve Cryptosystem

The elliptic curve cryptosystem is a type publickey encryption system. Neil Koblitz and Victor Miller [65] proposed this method back in 1985. ECC can be used for creating digital signatures, securing data through encryption, and sharing keys safely. When compared to RSA, ECC can use substantially smaller key sizes. ECC, unlike other cryptosystems, requires a smaller key size of 160 bits.

Using ECC with a reduced key size has a computational advantage over RSA, albeit providing identical security. The results show that ECC is successful in terms of data file size and encryption [66]. ECC is a popular security method due to its low communication and calculation costs. Elliptic curve cryptosystems are secure because solving the ECDLP is extremely difficult. .

2.6.3 Elliptic Curve Encryption Decryption

Elliptic Curve Cryptography (ECC) is an asymmetric cryptography technique. Elliptic curve cryptography (ECC) uses elliptic curves to perform cryptographic operations.

Each user has their own keys for secure transformation. This section [67] describes how to encrypt and decrypt using ECC.

2.6.3.1 Global Parameters

The global parameters for this scheme are as follows:

1. The elliptic curve's generating point G has a relatively large order n. That is, $nG = O$.

2. The curve's constant parameters (a and b).

3. The prime number p.

2.6.3.2 Key Generation

Assume Ayesha wishes to send a message M to Badar. Ayesha and Badar calculate their keys using the following method:

1. Ayesha chooses a random integer for her private key, $k_p = k_A$

$$k_A \in \{1, \ldots, n-1\} \tag{2.3}$$

The public key K_A is calculated using generator point of elliptic curve as:

$$K_A = k_A \cdot G \tag{2.4}$$

2. Badar randomly chooses a number as his private key, $k_P = k_B$, such that

$$k_B \in \{1, \ldots, n-1\} \tag{2.5}$$

The public key (k_B) is calculated using the generator point of elliptic curve as:

$$k_B = k_B \cdot G \tag{2.6}$$

2.6.3.3 Encryption

Ayesha want to send a message M to Badar, which is considered to be a point on the elliptic curve E_p(a, b).

1. Ayesha randomly selects a positive integer β.

2. Create ciphertext using Badar's public key.

$$C_M = \{\beta G,\ B_M + \beta K_B\} \tag{2.7}$$

2.6.3.4 Decryption

Badar converts the ciphertext C_M from Ayesha to its original plaintext using the following formula:

$$B_M = B_M + dK_B - k_B(dG) \tag{2.8}$$

$$= B_M + d(k_B G) - k_B(dG) \tag{2.9}$$

$$= B_M \tag{2.10}$$

Example 2.6.1.

Consider the elliptic curve $y^2 = x^3 + 7 \bmod 17$. Assume the generating point is $G = (15, 13)$. The elliptic curve point has an order of 18, which is equivalent to $18G = O$, where O represents the point at infinity.

Assume Ayesha wishes to deliver a message B_M to Badar using Elliptic Curve Cryptography.

Assume Ayesha chooses $k_A = 3$ for her private key and calculates her public key as $3G = (8, 3)$. Badar chooses $4k_B = 5$ for his private key and calculates his public key as $5G = (6, 6)$.

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

$$C_M = \{dG, B_M + dK_B\} \tag{2.11}$$

$$= [2(15, 13), (11, 3) + 2(6, 6)] \tag{2.12}$$

$$= [(2, 10), (11, 3) + (3, 7)] \tag{2.13}$$

$$C_M = [(2, 10), (16, 8)] \tag{2.14}$$

Ayesha delivered the ciphertext to Badar. Badar decrypts the ciphertext to recover the original message (B_M).

$$B_M = B_M + dK_B - k_B(dG) \tag{2.15}$$

$$= B_M + d(k_B \cdot G) - k_B(dG) \tag{2.16}$$

$$= [(11, 3) + 2(5(15, 13)) - 5(2(15, 13))] \tag{2.17}$$

$$= [(11, 3) + (3, 7) - (3, 7)] \tag{2.18}$$

$$B_M = [(11, 3)] \tag{2.19}$$

2.7 Elliptic Curve Discrete Logarthim Problem

The Elliptic Curve Discrete Logarithm Problem (ECDLP) is about finding a secret number A, given two points M and N on an elliptic curve, where $N = AM$. All these points belong to a group, called H. The security of elliptic curve cryptography (ECC) depends on how hard it is to solve this kind of problem, known as the Discrete Logarithm Hardness (DLH). The main rule in ECC is $N = AM$, where both M and N are points on the curve, and a is a private value. When A is a very large number, it's easy to use a and M to get N. But if someone only knows M and N, it's extremely hard for them to figure out A. So, while it's simple to go from A and M to N, it's very difficult to go backwards and find a just from M and N. This difficulty is what makes ECC secure.

2.8 Basics Of Hyperelliptic Curve

Hyperelliptic curves [68] are widely explored in cryptology. Compared to elliptic curve-based cryptosystems, this method provides comparable security with shorter key lengths. In 1987, Koblitz [69] claimed that Jacobians of hyperelliptic curves can generate abelian groups appropriate for cryptography. Hyperelliptic curves are a type of algebraic curve that can be seen as a mode advanced version of of elliptic curves. A hyperelliptic curve of genus 1 is also known as an elliptic curve. Hyperelliptic curves are a type of algebraic curve that might be considered

a generalization of elliptic curves. A hyper elliptic curve with genius $g = 1$ is also known as an elliptic curve. As a result, any curves with genus g higher than or equal to one are hyperelliptic.

Elliptic curves have simple and straightward shape, On the other hand hyperelliptic curve have curve shape and complex geometry. Following equation describe hyperelliptic curve

$$y^2 = f(x) \tag{2.20}$$

Here, $f(x)$ is a polynomial, which is a mathematical expression containing powers of x, such as

$$f(x) = x^5 + 2x^3 + 7 \tag{2.21}$$

In contrast to elliptic curves, $f(x)$ has a degree of 3 like $(x^3 + ax + b)$. However, with hyperelliptic curves, the degree of $f(x)$ is typically 5 or greater. For further details see section 4.2.

Chapter 3

Efficient Double Parameter Elliptic Curve Digital Signature Algorithm for Blockchain

The traditional ECDSA relies on a single inversion operation for signature and verification, resulting in significantly lower efficiency. Currently, research focuses on decreasing reverse operations to improve efficiency, but neglects forgery and signature attacks.

The weak randomness of ECDSA [13] in blockchain can lead to attacks forging random numbers, posing a possible concrn for buying, selling, or sending money using digital money. Therefore, this techniques approches a better and proven to be secure version of ECDSA method.

The novel signature strategy [23] uses double parameters to prevent weak randomness attacks, making it applicable to online platforms that use blockchain to trade digital money. The random oracle model [24] is as hard to solve as ECDLP, even for two kind of powerfull atackers.

3.1 Introduction

Blockchain technology [70] has significantly revolutionized industries and commercial organizations by leveraging cryptography to assure transaction traceability, cannot be changes,cannot be dained, and cannot be faked. Prior to ECC's popularity, most public key algorithms relied on RSA, DSA, and DH, which are modular encryption schemes [71].

However, recent episodes of bitcoin theft and anonymous transactions have raised concerns about blockchain security. Bitcoin's core technology relies on conventional cryptography and allocate structure technology to make online payment work without needing someone else to be trusted in the middle.

Bitcoin's core technology [72] relies on classical cryptography and distributed system technologies to reduce reliance on trusted third parties in electronic transactions. This covers how the system works, protects against fake identities, uses ECDSA for digital signature, agreed on share decsions and checks data with Merkle tree proofs.

Bellare et al. [15] advocated weak randomness for signatures. Researchers discovered that using an insecure password to create a random number can disclose DSS vulnerabilities.Users can create the ECDSA private key without using a truly random number, or they might use the same arbitrary number more than once. Failure to encrypt random integers risks exposing the private key.

Inadequate protection of Bitcoin users funds can result in significant economic losses. The industry has identified two major issues with elliptic curve digital signatures: inverse operation [16], which can take up to ten times longer than multiplication, and dot product. To save time, the inverse operation is prioritized above the multiplication operation, which requires at least one repetition. Researchers have proposed several improved methods for performing inverse and scalar multiplication operations on this topic.

New algorithms without inverse operations, such as scalar and modular multiplication, have been presented in the literature [18] to enhance operation speed.

3.2 Elliptic Curve Digital Signature Algorithm

This section provides an overview of the Elliptic Curve digital Signature Algorithm Signature, Elliptic Curve Digital Logarthmic Problem, Digital payment structure, relevant background information.

3.2.1 Notations

$\mathbb{F}_p$: A discrete prime field

E_p : An elliptic curve

P : Generator point

m : Mod value

h : Cofactor

$H(M)$: Hash function

HV : Hash value

M : Message

d : Private key

Q : Public key

3.3 Preliminaries

The ECDSA is like DSA but works on elliptic curve. The users agree as the follows elliptic curve group $E(\mathbb{F}_p)$. The equation below defines an ECC over a limited prime field $F(p)$,

$$E(\mathbb{F}_p) : y^2 = x^3 + \alpha x + \beta \pmod{m} \tag{3.1}$$

where p is a prime number, G is a group of points on elliptic curve with order m.

3.3.1 Elliptic Curve Digital Signature Generatiom

A Signer signs the message as follows:

1. Pick a arbitrary integer a from $\{1, 2, \ldots, m-1\}$
2. Calculate $aG = (u, v)$, and then calculate $t = u \bmod m$. If $t = 0$, go back and start again from step 1.
3. Calculate $e = H(M)$, where $H(M)$ represents the secure hash algorithm.
4. Calculate $s = a^{-1}(e + tD) \mod m$. If $s = 0$, go return to step 1 and again select a.

The pair (t, s) is then sent by A to verifier B as the signature on the message M.

3.3.2 Elliptic Curve Digital Signature Verification

The receiver B verifies similar Hash function using the public key $Q = dG$ after receiving A's signature for message M

1. Signature is true if t and s are integers from the interval $\{1, 2, \ldots, m-1\}$ otherwise signature will be invalid.
2. Calculate $e = H(M)$, where $H(M)$ represents the secure Hash Function.
3. Calculate $u = ew \mod m$, $v = sw \mod m$.
4. Calculate $u', v' = uG + vQ$; and calculate $t' = u' \mod m$.
5. Signature will be valid if $t = t'$ otherwise it is wrong.

3.3.3 Elliptic Curve Digital Signature Algorithm Complexity Analysis

In ECDSA, public key is generated as $Q = dP$. During both the signature generation and verification processes, the modular inverses $a^{-1} \pmod m$ and s^{-1}

(mod m) must be computed separately. When the bit length of the operands involved in modular multiplication is m, the computational complexity of a single modular multiplication operation is $O(m^2 \ln m)$. It is evident that both the signature generation and verification stages are computationally demanding because each involves modular inverse operations, which contribute substantially to the total computation cost.

3.4 Blockchain Cryptography

Blockchain constitutes a method for secure and organized data record-keeping utilizing distributed computing systems. This concept may be envisioned as a digital ledger, where, instead of singular custodianship, multiple copies are distributed across a global network. This architecture dictates that the record's content is visible to all participants, and no individual may alter the data without the change being universally noted. The data is stored in discrete units referred to as blocks. Each block contains a compilation of information, such as transactions or records. Upon being filled, a block is cryptographically linked to its predecessor, thereby establishing a chain of blocks, which gives rise to the term blockchain [73]. Furthermore, every block incorporates a unique identifier known as a hash, functioning as its digital fingerprint. Any attempt to modify the information contained within a block will result in a change to this hash, which breaks the sequence and indicates a compromise. Prior to the addition of a new block to the chain, the computers in the network must collectively validate and agree upon the accuracy of the information. This critical validation procedure is termed consensus. Different blockchain implementations employ varied methodologies for achieving consensus, such as the resolution of complex mathematical problems or the demonstration of asset ownership [74]. This mechanism prevents the incorporation of erroneous or fraudulent data. Given that the blockchain is distributed across numerous computers and not under the control of a single entity, it is exceptionally resistant to unauthorized tampering or hacking. The security of ECDSA signatures in Bitcoin relies heavily on the use of unique random numbers for each transaction. However, if

these random numbers are reused, it can lead to a security vulnerability, allowing an attacker to compromise the private key.

As discussed earlier, an ECDSA signature takes the form $M = (t, s)$. All signatures are visible on Bitcoin's public ledger. In Bitcoin, reusing the same random number allows one public key holder to potentially calculate another user's private key. Reusing either the random number or the public key can expose the private key to anyone who can access the blockchain.

In the third step of the ECDSA signing process, a random number a is securely chosen from $G(F_p)$. Suppose there are two transactions, u_i for $(i = 1, 2)$. Each transaction u_i has a signature t_i, s_i, a public private key pair $G_i = d_i Q_i$, and an intermediate variable a_i. The hashes of these transactions are H_i for $(i = 1, 2)$. In this section, we will examine two scenarios in which an attacker can take advantage of reused random numbers.

3.4.1 Random Number Reuse under Different Public Keys

If two transactions have different public keys but use the same random number a, an attacker can calculate the private key of one transaction owner based on the signature of the other. Since a is the same, $t_1 = t_2 = t = u \mod m$, where $(u, v) = a * G$. The attacker can then use the following equation to calculate the private key d_i of one transaction owner based on the signature (t, s_i) and the hash H_i of the other transaction:

$$d_i = t^{-1}(s_i a - H_i) \mod m \tag{3.2}$$

3.4.2 Reuse Random Numbers and Public Keys

When two transactions share the same public key $d_1 = d_2 = d$ and reuse the same random value a, the private key d can be determined by anyone using the following approach. :

$$k = (s_1 - s_2)^{-1}(H_1 - H_2) \mod m \tag{3.3}$$

$$d = t^{-1}(s_1 a - H_1) \mod m \tag{3.4}$$

In both cases, reusing the random number a leads to a security vulnerability, allowing an attacker to compromise the private key. To avoid these vulnerabilities, it's essential to use unique random numbers a for each transaction, even if the public keys are different, and to use unique public keys for each transaction, or ensure that the random numbers a are uniformly random and not reused.

In 2013, Thomas et al. [75] proposed a deterministic technique for creating digital signatures. Instead, a deterministic algorithm will compute using secure keys and note. A sincere client is unlikely to select the similar pseudo arbitrary number numerous pattern, and there should be no several sign in a deal using similar arbitrary number. A 20-year analysis of Bitcoin transactions revealed that 0.48 percent of transactions still had this vulnerability, and 1331 private keys had been hacked. Inadequate Bitcoin security can lead to significant economic losses.Spam transaction attacks can take advantage of ECDSA's weak randomness, as transactions connected with several addresses follow similar patterns. The researchers examined popular Digital wallets to see if they susceptible to poor stray of elliptic curve digital signature algorithm. Since conclusion is hopeful, one of impacted location disclosed in april 2014 was still use in august 2017.

3.5 Proposed Two Parameter ECDSA

The proposed scheme works by using the following four algorithms

1. Setup is an initial scheme. Created EC variable $T = (\alpha, \beta, , m, h)$, created a key points (d, G), and choose a hash function.

2. KeyGeneration is key generating method. Choose arbitrary whole number (a, a_1, a_2) and ensure that $a = a_1t + a_2n$.

3. Signcryption is a method used to both sign and encrypt information. Use a hash to find w and make signature.

4. Verify is an algorithm used to check authenticity. Calculate $v = (s+a_2n)\text{mod } m$, $u = (w + t)\text{mod } m$ and then confirm if the eqaution holds true.

3.6 Security Model

In this section, we present the security models of our scheme through a series of games between the challenger $\mathcal{C}$ and two types of attackers, $\mathcal{A}I$ and $\mathcal{A}II$. The security of our scheme is based on the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP).

3.6.1 Elliptic Curve Discrete Logarithm Problem

Given an elliptic curve E over a finite field $\mathbb{F}_p$, let $G \in E(\mathbb{F}_p)$ be a point of order m. The ECDLP is to find the integer $d \in [0, m-1]$ such that $Q = dG$, where $Q \in E(\mathbb{F}_p)$.

3.6.1.1 Security Definition

Let $\mathcal{C}$ be a polynomial-time algorithm. The advantage $\alpha_{\mathcal{C}}^{\text{ECDLP}}$ of $\mathcal{C}$ is defined as the probability Pr:

$$\alpha_{\mathcal{C}}^{\text{ECDLP}} = \Pr[\mathcal{C}(Q, G) = d \mid d \in [0, m-1] \text{ and } Q = dG]$$

3.6.1.2 Elliptic Curve Discrete Logarithm Difficulty Hypothesis

The ECDLP difficulty hypothesis states that if there is no polynomial-time algorithm $\mathcal{C}$ that can solve the ECDLP with a non negligible probability ϵ in polynomial time, then the ECDLP is considered hard. "'

Adversary A_I may adaptively issue a polynomial number of the following queries:

User-generated queries: The challenger maintains a list L_c, initially empty. Each entry of L_c is a tuple (I, t, s, a).

Private-key queries: If A_I requests the private key of user I, the challenger C runs the key-generation algorithm and returns the private key D_I to A_I.

Public-key replacement queries: On a public-key replacement query for user I, C updates the corresponding entry of L_c, replacing a with a new public key a'.

Signature queries: On a signature query, C computes the signature (or ciphertext) σ and returns it to A_I.

H_1 queries: C simulates the random oracle H_1 and maintains a list L_{H_1} (tuples of the form (I, b, t)), initially empty.

H_2 queries: C simulates the random oracle H_2 and maintains a list L_{H_2} (tuples of the form (M, I, Z_A, b, a, w)), initially empty.

3.7 Elliptic Curve Initialization and Key Setup

To provide strong security for the elliptic curve system, we first choose a large prime number q, which defines the finite field $\mathbb{F}_q$. The elliptic curve is described over this field by the equation:

$$y^2 = x^3 + \alpha x + \beta$$

Next, we identify the key domain parameters for the curve. These parameters are grouped as:

$$T = (\alpha, \beta, G, m, h)$$

where:

- α and β are constants from the field $\mathbb{F}_q$.
- G is the base point on the curve.
- m is a prime number representing the order of point G.
- h is the cofactor.

The signer, referred to as A, selects a private key d, where d is randomly chosen from the range $[1, m-1]$. The corresponding public key is calculated as $Q = dG$

To introduce randomness in further operations (e.g., for signing), two random values a_1 and a_2 are also selected, both chosen from the interval $\{1, 2, \ldots, m-1\}$

3.7.1 Signature Generation

A sender signs a message M by following these steps:

1. Select a random integer a from interval $\{1, 2, \ldots, m-1\}$. Compute $aG = (u, v)$, then compute $t = u \bmod m$. If $t = 0$, repeat this step.
2. Choose two more random integers a_1, a_2 from interval $\{1, 2, \ldots, m-1\}$. Then compute:
$$a = a_1 \cdot t + a_2 \cdot n$$
3. Compute the hash of the message as $e = H(M)$, where $H(M)$ is a secure hash function. Then calculate the Hamming weight w.
4. Calculate
$$s = t \cdot a_1 + (w + t) \cdot d \bmod m$$
If $s = 0$, return to Step 1.

The resulting signature on the message M is the tuple (t, s, a_2), which is then sent to the verifier.

3.7.2 Signature Verification

To verify the validity of a signature (t, s, a_2) for a message M, follow these steps:

1. Ensure that t, s, and a_2 are integers in the range $\{1, 2, \ldots, m-1\}$. If any value falls outside this range, the signature is considered invalid.

2. Compute the hash of the message: $e = H(n)$, where H is a secure hash function. Then, determine the Hamming weight w.

3. Calculate
$$v = (s + a_2 \cdot n) \bmod m, \quad u = (w + t) \bmod s$$

4. Compute the elliptic curve point:
$$(u', v') = vG - uQ \bmod m$$
and then calculate:
$$t' = u' \bmod m$$

5. If $t = t'$, the signature is valid. Otherwise, it is invalid.

3.7.3 Correctness of the Signature Scheme:

The following equations demonstrate that the signature generation and verification algorithms are consistent and valid:

$$(u', v') = vG - uQ = (s + a_2 n)G - (w + t)DG \bmod m$$

$$= [ta_1 + (w + t)D + a_2 n - (w + t)x]D \bmod m$$

$$= (ta_1 + a_2 n)B \bmod m \tag{6}$$

$$(u, v) = aG = (v - uD)G$$

$$u' = v - ux = b - (w + a)x \bmod s$$

$$= [(s + a_2 n) - (w + t)D]G$$

$$= vG - uQ = (u', v') \tag{7}$$

This proves that the proposed scheme is correct.

3.8 Unforgeability

3.8.1 Theorem 1

Assume an attacker $\mathcal{A}_I$ can make a limited number of queries (q_c user-generated queries and q_k private key queries) in polynomial time. If $\mathcal{A}_I$ can break our proposed scheme with a significant probability $\beta\mathcal{A}_I$, then we can construct an algorithm $\mathcal{C}$ that can solve the ECDLP with a probability of at least $\frac{1}{q_c}\left(1 - \frac{1}{q_c}\right)^{q_k} \cdots \beta\mathcal{A}_I$ in polynomial time.

In simpler terms:

If an attacker can break our scheme, then we can use that attacker to solve a well-known hard problem (ECDLP) with a certain probability. This implies that our scheme is secure, assuming the ECDLP is hard to solve.

3.8.2 Theorem 2

Assume an attacker $\mathcal{AII}$ can make a limited number of queries (q_c user-generated queries and q_s secret value queries) in polynomial time. If $\mathcal{AII}$ can break our proposed scheme with a significant probability $\beta_{\mathcal{AII}}$, then we can construct an algorithm $\mathcal{C}$ that can solve the Elliptic Curve Discrete Logarithm Problem (ECDLP) with a probability of at least $\frac{1}{q_c}\left(1-\frac{1}{q_c}\right)^{q_s} \times \beta\mathcal{A}_{II}$ in polynomial time.

In simpler terms:

If an attacker can break our scheme, then we can use that attacker to solve a well known hard problem (ECDLP) with a certain probability. The probability of solving ECDLP depends on the number of queries made by the attacker, but it remains non-negligible if the attacker can break the scheme For the proof of the above theorems see [30].

3.9 Avoid Reusing Arbitrary Number

This feature of method uses a random number k from the interval $[1, m-1]$ to model n functions. Since n is whole number and r, m are known, k was freely chosen from the interval. The equation $a = a_1t + a_2n$ requires a_2 to be within $[1, m-1]$. The improved scheme's structure is reasonable. We limit the reuse of random numbers by selecting the same number twice.

3.10 Prevent Data Tampering

The hash function and hamming weight ensure the integrity of the message. Tampering with data might lead to inaccurate findings when using hamming weight. The preceding proof demonstrates that data manipulation can be efficiently prevented.

3.11 Future Security Proof

If an unauthorized person obtains the private key for A or B, it becomes difficult to retrieve the session key. The ECDLP states that even if an attacker obtains Q_A and G, they are unlikely to gain dA and thus the private key, because $Q_A = dA_G$. Thus, the strategy offers forward security.

3.12 Man in the Middle Attack

Setting up a session key without verifying the other party's identity can leave the system open to man in the middle attacks. To prevent this, the system employs digital signatures for two way authentication. This ensures that unauthorized users cannot impersonate any participant, effectively blocking man in the middle attacks.

3.13 Non Repudation

The receiver relies on the values (s, w, t) provided by the sender to ensure that the sender cannot later deny their involvement.

This table presents a comparison of the time required by the proposed scheme versus the traditional ECDSA.

TABLE 3.1: Security comparison

Method	Forgery	Data Tampering	Forward safety	Man-in-the-middle attack Non-repudiation
Zhang	×	×	×	✓
Wu	×	✓	✓	✓
our algorithm	✓	✓	✓	✓

Chapter 4

Double-Parameter Hyperelliptic Curve Digital Signature Algorithm for Blockchain

The hyperelliptic curve [76] is proposed as a replacement for the elliptic curve in certificateless encryption systems, providing the same level of security but reducing the computational costs.

Extending elliptic curve encryption to wider curves opens up new mathematical possibilities. The growth of hyperelliptic curves creates new challenges and potential for cryptographic applications, making it a promising study area.

4.1 Introduction

Modern cryptography prioritizes reliable and efficient foundations to protect sensitive data. Elliptic curve cryptography (ECC) [77] offers secure and cost effective solutions for various cryptographic applications. As cryptographic requirements evolve and threats become more complex, there is a growing demand for new mathematical approaches that provide comparable security while improving efficiency. The IoT-CS technique, which utilizes hyperelliptic curve cryptography, aims to

improve security while lowering computational and communication costs in IoT contexts.

4.2 Hyper Elliptic Curve Cryptography

Hyperelliptic Curve Cryptography (HECC) [29] is an advanced variety of general key cryptography that extends Elliptic Curve Cryptography (ECC) to work with hyperelliptic curves. HECC, like ECC, relies on hardness to solve the discrete logarithm problem, but it employs more complicated mathematical objects. When we defined a hyperellitic curve over a finite field and often gets the structure:

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

The polynomials $h(x)$ and $f(x)$ have coefficients in equation. The degree of $h(x)$ is at most g, while degree of $f(x)$ is at least $2g + 1$. Non singularity occurs when no points of curve link the calculations simultaneously. $2y + h(x) = 0$, and $h - f(x) = 0$. The curve genus is the number of simple curves that draw on a face that do not touch each other. A curve of genus 2 is ideal for secure and efficient computations. Here are the curves for various genus.

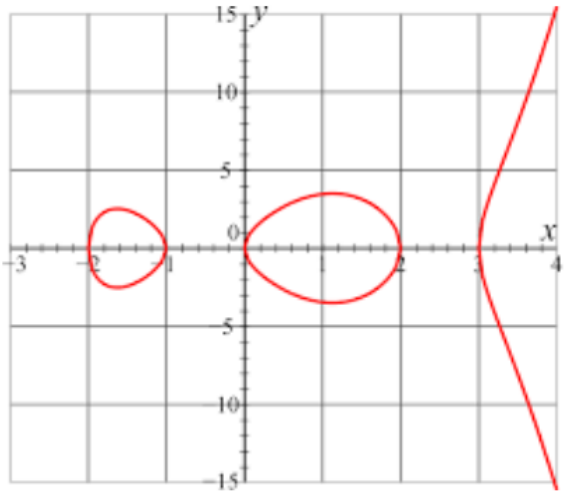

FIGURE 4.1: Hyper Elliptic Curve

1. $y^2 = x^3 + x + 1$ has genus 1 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.

For more information on hyperelliptic curves, please refer to [29]

Encrypted communication on online network requires IT security solutions like isolation and validation. Among these tactics, public key cryptography is crucial in our everyday life. This IT security technology serves one of most important aspects, such as electronic payment infrastructure. Diffie and Hellman [78] created public key cryptography in 1976. In 1977, Rivest,Shamir,and Adleman [79] pioneered practical applications. The extensively used public key cryptosystem is Rivest Shamir Adleman (RSA) [79]. The elliptic curve cryptosystem is now the greatest option for public key cryptography,despite the popularity of RSA. The elliptic curve was first proposed by Koblitz [51] and Miller [80] separately. The security of the RSA is dependent on the difficulty of solving the integer. The factorization problem and ECC rely to solve discrete logarithm problem on an elliptic curve. Analytical points on an ECC can form group using Diffie-Hellman's [78] generalized notions. This group is eventually utilized to create ECC. Similar to ECC, Hyper Elliptic Curve (HECC) can form group structures over the Jacobian of a hyperelliptic curve specified over a finite field.

4.2.1 Global Parameters

Following are the global parameters for this scheme:

H : Hyperelliptic curve
D : A divisor of hyperelliptic curve
$ND = P_\infty$

where

m : Order of divisor
P_∞ : Point at infinity

H(M) : Hash function

4.2.2 Notations

F_p : A discrete prime field

H_p : Hyperelliptic curve

D : Divisor

m : Mod value

h : Cofector

$H(M)$: Hash function

HV : Hash value

M : Message

Q : Public key

4.3 Ordinary, Opposite and Special Points

Consider $P = (x, y)$ is a finite point on HEC, its opponent point on curve is $\overline{P}(x, -y - h(x))$. A point $\mathcal{O}$ is called point at infinity and its opponent is represented by $\overline{\mathcal{O}}$ such that $\mathcal{O} = \overline{\mathcal{O}}$. A point P is called special if $P = \overline{P}$ otherwise called ordinary point.

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

$$C : y^2 + xy = x^5 + x^4 + x^2 + x + 2 \tag{4.2}$$

It is clear that curve lacks a unique point. Hence, C is a **hyperelliptic curve** over $\mathbb{Z}_7$.

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

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

The point $(6, 4)$ is a special point

4.3.1 Divisor

The divisor $D = \sum m_n P$ for the hyperelliptic curve H is an arbitrary linear combination of distinct points

$P_1, P_2, P_3, \ldots, P_n$ on C,

where $m_1, m_2, m_3, \ldots, m_n \in \mathbb{Z}$

with 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, $P(x, y)$, is determined as follows:

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

Suppose several points P_1, P_2 and Q_1, Q_2 are hyperelliptic curves 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 observation of these two further points on the C are $\bar{R}_1$ and $\bar{R}_2$

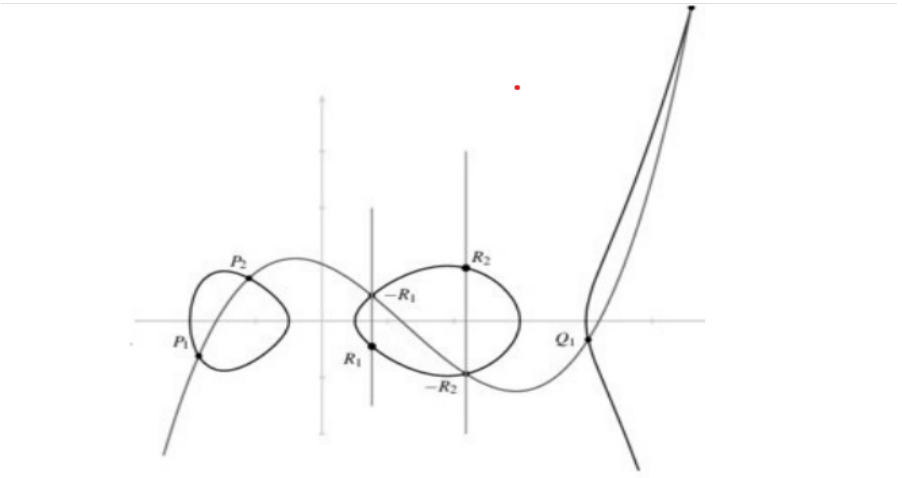

FIGURE 4.2: Geometrical Representation of Divisor

The divisors are expressed as

$D_1 = P_1 + P_2 - 2\infty$
$D_2 = Q_1 + Q_2 - 2\infty$

Third divisor of these divisors are evaluated as

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

4.3.2 Divisor Group

All the divisors of a hyperelliptic curve H together make a group using addition, and this group is written as $D = D(H)$. The addition is defined as

$$\sum m_i P_i + \sum n_i P_i = \sum (m_i + n_i) P_i.$$

The subgroup which containing the divisors of degree zero is denoted as D^0.

4.4 Hyperelliptic Curve Discrete Logarithm Problem

Let D_1 and D_2 are two divisors of the hyperelliptic curve H, finding an integer c such that c$D_1 = D_2$ is a HECDLP. Finding c is not computationally feasible. The complete security of HECC depends on HECDLP [81].

4.4.1 GCD of Divisors

The greatest common divisor of $D_1 = \sum m_i P_i$ and $D_2 = \sum n_i P_i$ is defined as

$$\gcd(D_1, D_2) = \sum \min(m_i, n_i) P_i$$

where $P_i \in C$ and $m_i, n_i \in \mathbb{Z}$.

4.4.2 Semi Reduced Divisor

A divisor $D = \sum m_i p_i - (\sum m_i)\,\infty$ is called a **semi-reduced divisor** if:

1. $m_i \geq 0$ for each $i \in \mathbb{N}$,
2. If $p = \overline{p}$, then each $m_i = 1$,
3. Only p or $\overline{p}$ is used in the sum if $p \neq \overline{p}$.

4.4.3 Reduced Divisor

$$D = \sum m_i p_i - \left(\sum m_i\right) \infty$$

is called a **reduced divisor** if $\sum m_i \leq g$, where g is the genus of the hyperelliptic curve. The group operation is carried out on the decreased divisors for the practical implementation of Hyperelliptic Curve Cryptography (HECC).

4.5 Mumford Representation

Working with divisors on hyperelliptic curves poses significant challenges for implementation. Cantor, building on Mumford's foundational work [82], introduced a representation that enables efficient computation and manipulation of divisors.

Let

$$D = \sum m_i p_i - \left(\sum m_i\right) \infty,$$

be a semi-reduced divisor, where each point $p_i = (x_i, y_i) \in H$ lies on the hyperelliptic curve. Define the monic polynomial

$$\alpha(u) = \prod (u - x_i)^{m_i}.$$

There is a single polynomial $\beta(u)$ that meets the following conditions:

1. $\deg \alpha > \deg \beta$.
2. $\beta(x_i) = y_i$ for each i such that $m_i \neq 0$.
3. $\alpha(u) \mid (\beta(u)^2 + \beta(u)h(u) - f(u))$.

Then the Mumford representation of the divisor is given by:

$$D = \gcd\left(\operatorname{div}(\alpha(u)), \operatorname{div}(\beta(u)h(u)) - y\right).$$

Example 4.5.1.

Consider the hyperelliptic curve over $\mathbb{F}_{11}$:

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

This curve has genus $g = 2$.

Let the divisor be:

$$D_1 = (1, 3) + (2, 1)$$

We aim to compute the Mumford representation $[u(x), v(x)]$ of D_1.

Step 1: The polynomial $u(x)$ is the monic polynomial whose roots are the x-coordinates of the points in the divisor:

$$u(x) = (x-1)(x-2) = x^2 - 3x + 2 \equiv x^2 + 8x + 2 \mod 11$$

Step 2: We seek a polynomial $v(x) = ax + b$ of degree less than 2 such that:

$$v(1) = 3, \quad v(2) = 1$$

Solving:

$$a + b \equiv 3 \mod 11$$
$$2a + b \equiv 1 \mod 11$$

$\Rightarrow$ Subtracting: $a \equiv -2 \equiv 9 \mod 11$
$\Rightarrow b \equiv 3 - a \equiv 3 - 9 \equiv -6 \equiv 5 \mod 11$ Thus:

$$v(x) = 9x + 5$$

Step 3: (Verify Interpolation): Note that,

$$v(1) = 9(1) + 5 = 14 \equiv 3 \mod 11 \quad v(2) = 9(2) + 5 = 23 \equiv 1 \mod 11$$

Both conditions are satisfied.

Final Result: The Mumford representation of the divisor $D_1 = (1, 3) + (2, 1)$ is:

$$[u(x), v(x)] = \left(x^2 + 8x + 2,\ 9x + 5\right)$$

4.6 Proposed Scheme

The scheme suggests an upgrade to the classic Elliptic Curve Digital Signature Algorithm (ECDSA) by proposing a double parameter signature scheme to increase

the security and efficiency of digital signatures used in blockchain systems [30]. Traditional ECDSA [16] relies on a single random parameter during signature generation, which can be exploited if weak randomness is utilized, potentially exposing private keys.

The revised approach introduces a second parameter, increasing unpredictability and lowering the risk of attacks that use predictable or reused nonces [83]. It also improves signature generation and verification operations, resulting in improved performance in resource-constrained contexts such as blockchain networks [84].

When we apply this notion to Hyperelliptic Curve Cryptography (HECC), the same principles can be used, potentially with much higher benefits [85]. HECC is a type of elliptic curve cryptography that operates on curves of larger genus (2) and provides similar security with shorter key lengths [86].

This makes HECC particularly appealing in contexts with limited storage and bandwidth. A hyperelliptic curve's Jacobian group structure could be used to implement a double parameter signature method that includes two independent secret values and random parameters [87].

This not only increases resilience to weak randomness, but also makes use of HECC's structural complexity and great security-per-bit efficiency. While HECC operations are computationally more difficult than those on elliptic curves, advances in algorithm design are constantly improving their efficiency [88].

Thus, the fundamental concept of using double parameters for increased security is quite consistent with, and potentially useful in hyperelliptic curve encryption.

Algorithm 4.6.1. Consider a hyperelliptic curve H over a finite field F. It has a big prime and a divisor D of m prime order. Let $\alpha, \beta \in \mathbb{F}_p$ define parameters over the finite field $\mathbb{F}_p$. Let D be a divisor on the Jacobian of a hyperelliptic curve H, and suppose the order of D is a prime order m.

The private key of signer A is an integer d from interval $\{1, 2, \ldots, m-1\}$. The corresponding public key is:

$$Q = dG,$$

where the operation is scalar multiplication in the Jacobian group of the hyperelliptic curve. To generate a signature, introduce random parameters a_1 and a_2 from interval $\{1, 2, \ldots, m-1\}$

Algorithm 4.6.2.

Let H be a hyperelliptic curve over a finite field $\mathbb{F}_p$ of genus g, and let D be a base divisor on the Jacobian $\mathrm{Jac}(H)$ of order m. Let A be the signer, with private key d from interval $\{1, 2, \ldots, m-1\}$ and public key $Q = dG$. To sign a message M, signer A performs the following steps:

1. Random divisor generation: Select a random integer a from interval $\{1, 2, \ldots, m-1\}$. Compute the scalar multiple

$$aG = D = (u, v),$$

   where D is a reduced divisor in Mumford representation. Extract the u-coordinate from the first affine point of D, and compute

$$t = u \bmod m.$$

   If $r = 0$, repeat this step with a new a.

2. Double randomization: Select two random integers a_1, a_2 $\{1, 2, \ldots, m-1\}$, and compute an aggregated nonce:

$$a = a_1 t + a_2 n.$$

3. Hashing the message: Compute the hash of the message:

$$e = H(M),$$

   where HM is a secure hash function (e.g., SHA-256). Then calculate the Hamming weight w of e.

4. Signature generation: Compute the signature component:

$$s = ta_1 + (w + t)d \pmod{m}.$$

If $s = 0$, return to Step 1 and choose a new a.

Then the signature of A on message M is the tuple:

$$(t, s, a_2),$$

which is transmitted to the verifier B.

Algorithm 4.6.3.

Let H be a hyperelliptic curve defined over the finite field $\mathbb{F}_p$, and let D be a base divisor of prime order m in the Jacobian group $\mathrm{Jac}(H)$. Given the signer's public key $Q = dG$ and a signature (t, s, a_2) for a message M, any verifier can check its validity as follows:

1. Range Check: Verify that t, s, and a_2 are integers from the interval $\{1, 2, \ldots, m-1\}$. If not, the signature is invalid.

2. Hash Computation: Compute

$$e = H(M),$$

where $H(M)$ is a secure hash function (e.g., SHA-2). Then compute the Hamming weight w of e.

3. Compute Scalars: Calculate:

$$v = (s + a_2 n) \bmod m, \quad u = (w + t) \bmod m.$$

4. Jacobian Operation: Compute the divisor

$$D' = vG - uQ \in \mathrm{Jac}(H).$$

Let (u_0, v_0) be the affine representation of the first point in the reduced form of D' (e.g., from the Mumford representation of D'). Then compute:

$$t_0 = u_0 \bmod m.$$

5. Validation: If $t = t_0$, then the signature is valid otherwise, it is invalid.

Algorithm 4.6.4.

In the proposed digital signature scheme based on hyperelliptic curve cryptography (HECC), all operations are performed over the Jacobian group $\mathrm{Jac}(H)$ of a hyperelliptic curve H defined over a finite field $\mathbb{F}_p$. Let D be a base divisor of prime order m, and let $Q = dG$ be the public key of the signer, where d from interval $\{1, 2, \ldots, m-1\}$ is the private key.

Let w be the Hamming weight of the hash $e = H(M)$ of the message M. Let t, s, and a_2 be the components of the signature.

The verifier computes the following:

$$\begin{aligned}(u_0, v_0) &= vD - uQ \\ &= (s + a_2 n)D - (w + t)dGD \mod m \\ &= \left[ta_1 + (w + t)d + a_2 n - (w + t)d\right] D \mod m \\ &= (ta_1 + a_2 n)D \mod m \end{aligned} \tag{7}$$

This shows that the divisor reconstructed during verification corresponds to the one used in signature generation.

From the signature generation phase, the ephemeral scalar k is defined as:

$$a = ta_1 + a_2 n$$

Now, using the relation:

$$ud = v - a \quad \Rightarrow \quad a = v - ud = v - (w + r)d \mod m$$

Thus, we compute:

$$\begin{aligned}(u, v) = aD &= (v - ud)GD \\ &= [(s + a_2 n) - (w + t)d]\, GD \\ &= vG - uQ = (u_0, v_0) \end{aligned} \tag{8}$$

Therefore, the value (u_0, v_0) computed during verification is identical to the value used in signature generation. As a result, the derived value of t from (u_0, v_0) matches the original signature component t, and the signature is deemed valid. This proves the correctness of the signature scheme in the hyperelliptic curve setting.

4.7 Analysis of the Proposed Scheme

This section presents a security and performance evaluation of the proposed Double Parameter Hyperelliptic Curve Digital Signature Scheme. The security of the method is grounded in the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP), while the cost analysis compares its computational efficiency to existing signature schemes.

The proposed scheme guarantees key security properties including confidentiality, integrity, authenticity, unforgeability, and verifiability. Its robustness is based on the HECDLP [89]. The security analysis assumes that attackers do not possess the private keys or secret parameters of the m^{th} and n^{th} nodes; they only have access to public information related to the communicating entities. Due to the strength of the encryption mechanism, adversaries are unable to decrypt the ciphertext c or forge the authenticated message.

This implies that the system is built on a strong cryptographic foundation, making unauthorized access or tampering extremely difficult. Even if an attacker knows which users are communicating, they cannot intercept or alter the actual message content. The encryption method effectively prevents brute force guessing or decoding of both the encrypted data and the authentication tags. Consequently, the scheme provides robust protection against various attack vectors and ensures that only legitimate users can access and verify the exchanged information, maintaining privacy and trust throughout the communication process.

4.7.1 Authentication

To establish a secure session, IoT nodes must mutually authenticate each other at the beginning of communication. When the m^{th} node sends the message $\{c, c', W\}$, the n^{th} node first derives the session key and then verifies the signature of the m^{th} node. If an adversary attempts to impersonate a legitimate node, they must generate a valid signature. However, this is infeasible without access to the m^{th} node's private key.

Upon receiving the authenticated message, the m^{th} node performs a verification of the n^{th} node. If the received message matches the expected format and values, the n^{th} node is successfully authenticated. An attacker would need to forge a valid message to impersonate the n^{th} node, but since the authentication relies on the node's private key, such forgery is computationally impractical.

4.7.2 Non-repudiation

The proposed scheme achieves non-repudiation through the signature components (s, w, r), which are generated using the sender's private key over a hyperelliptic curve. These components are cryptographically bound to both the message and the sender's identity. Due to the computational difficulty of solving the HECDLP [90], it is infeasible for any entity other than the legitimate sender to produce a valid (s, w, r) signature.

As a result, the receiver can verify the authenticity of the message using the sender's public key and the signature components. This ensures that the sender cannot deny having sent the message after transmission. The non-repudiation property provides strong assurance of accountability and trust in the communication protocol.

Since the scheme relies on the complexity of the HECDLP, no external party can replicate the signature without knowledge of the sender's private key. Even if someone attempts to deny authorship of a message, the signature serves as irrefutable proof of origin. This guarantees that once a message is signed and transmitted, the sender is bound to its content, reinforcing the integrity and reliability of secure communications.

4.7.3 Unforgeability

In the proposed IoT-CS scheme, an adversary may attempt to generate a valid signature. To do this, they would need access to the private key pair $\{a_m, \delta_m\}$ of the m^{th} node. However, retrieving these keys requires solving the hard HECDLP problem, which is computationally infeasible. Therefore, the scheme ensures strong protection against signature forgery.

4.7.4 Security Against Eavesdropping Attacks

In the proposed IoT-CS protocol, messages are transmitted in hashed, plaintext, and ciphertext formats. The plaintext messages do not contain sensitive information and offer no advantage to an attacker.

For confidential data, the scheme uses HECDLP, one-way hash functions, and encryption techniques that make it computationally impossible for an adversary to recover the original message. As a result, the protocol effectively prevents eavesdropping attacks.

4.7.5 Security Against Denial-of-Service Attacks

In the proposed IoT-CS system, each node first checks the validity of received timestamps. If a timestamp is invalid, the message is discarded. Every encrypted message includes the latest timestamp and is signed to ensure data integrity. This mechanism helps detect false messages and terminate the session if needed, thereby preventing denial-of-service attacks.

4.7.6 Security Against Man-in-the-Middle Attacks

When communicating parties are initially unaware of each other's identities, establishing a secure session key becomes vulnerable to man-in-the-middle attacks [91]. The proposed technique uses Hyperelliptic Curve Digital Signature technology to enable strong bidirectional authentication. Each party signs their communication using private keys derived from HECDLP, ensuring that only authorized users can generate valid signatures. Since HECDLP [90] is computationally hard to solve, an attacker cannot forge signatures or impersonate either party.

This means that before any secure session begins, both users must prove their identities by generating and verifying digital signatures. These signatures are based on private keys that cannot be guessed or recreated by outsiders due to the difficulty of HECDLP. Even if a hacker tries to interfere and act like one of the users during communication, they will fail because they cannot produce a valid signature without the actual private key. This ensures mutual trust and prevents unauthorized access, making the exchange of session keys secure and reliable between verified users.

4.7.7 Forward Security

Assume that an adversary obtains the public key Q_A of a user A in the proposed Hyperelliptic Curve technique. Since $Q_A = d_A \cdot D$, where d_A is the private key and D is a base divisor on the hyperelliptic curve, the adversary would need to solve

the HECDLP to extract d_A from Q_A and D. Due to the higher-genus structure of hyperelliptic curves, HECDLP is more complex than standard ECDLP for similar security levels.

Therefore, even if the public key Q_A is exposed, the attacker cannot feasibly derive the private key d_A or reconstruct the session key.

This means that even if a hacker gains access to someone's public key, they still cannot figure out the private key because solving HECDLP is extremely difficult. The special structure of hyperelliptic curves adds extra complexity, making it more secure than regular elliptic curve problems.

Even if the attacker somehow obtains the private key in the future, they still cannot read or modify any previously sent messages. Each session uses its own secure data, independent of the long-term private key. This property is called forward security, and it ensures that past communications remain confidential and unchanged even if private keys are compromised later.

4.8 Computational Cost

The computational overhead of an authentication scheme depends on how frequently various cryptographic operations are executed. According to Garg et al. [92], using MIRACL (Multiprecision Integer and Rational Arithmetic C/C++ Library) [31], the time required to perform hash-to-point (HtP) and elliptic curve scalar multiplication operations is 14.293 ms and 0.986 ms, respectively. The estimated processing time for hyperelliptic curve divisor multiplication is 0.48 ms. Compared to ECSM and HECDM, the time consumed by other cryptographic operations is minimal and can be neglected.

In the proposed scheme, both the sender and receiver perform two HECDM operations each. So, total time consumed by both parties is:

$$4 \times 0.48 = 1.92 \text{ ms}.$$

To verify the identities of at least two IoT nodes, the Key Generation Center (KGC) performs three HECDM operations, which require:

$$2 \times 0.48 = 0.96 \text{ ms}.$$

Hence, the total time required by the KGC and the nodes for mutual authentication is:

$$0.96 + 1.92 = 2.88 \text{ ms}.$$

Chapter 5

Conclusion

The hyperelliptic curve–based digital signature and signcryption scheme discussed in [93] demonstrates a robust and efficient framework for secure communication, particularly in environments with limited computational resources such as blockchain networks and Internet of Things (IoT) systems [94]. By leveraging the computational hardness of the HECDLP [95], scheme offers strong resistance against a wide range of cryptographic attacks, including forgery, impersonation, and replay.

The proposed scheme achieves essential security goals such as unforgeability, authentication, non-repudiation, and forward secrecy. Its use of bidirectional authentication through digital signatures effectively mitigates man-in-the-middle attacks, ensuring that only legitimate parties can participate in secure sessions. The integration of Hamming weight analysis and cryptographic hash functions further strengthens the protocol by enabling tamper detection and preserving message integrity.

One of the key advantages of hyperelliptic curve cryptography (HECC) is its ability to provide equivalent security to elliptic curve cryptography (ECC) while using shorter key lengths. This makes HECC particularly suitable for low-power and bandwidth-constrained devices. Even in scenarios where partial key exposure occurs, the structural complexity of hyperelliptic curves prevents adversaries from reconstructing private or session keys, thereby maintaining forward security [29].

The inclusion of signature components such as (s, w, t) ensures undeniable proof of origin, addressing non-repudiation requirements in secure communication protocols. These components are cryptographically bound to both the message and the sender's identity, It makes it practically impossible for anyone unauthorized to create a valid signature.

Illustrative Summary of Security Features

Unforgeability: Based on HECDLP, attackers cannot generate valid signatures without the private key.

Authentication: Mutual verification between nodes ensures only authorized entities can initiate communication.

Non-repudiation: Signature components (s, w, r) provide cryptographic evidence of message origin.

Forward Secrecy: Session keys stay safe even if the long term private keys are exposed.

Tamper Detection: Hash functions and Hamming weight analysis detect unauthorized modifications.

In conclusion, the hyperelliptic curve–based digital signature scheme offers a compelling blend of security, efficiency, and adaptability. Its strong theoretical foundations and practical performance make it a promising candidate for next-generation secure communication systems. With continued research and refinement, HECC can play a pivotal role in shaping the future of cryptographic protocols across diverse domains.

Bibliography

[1] M. A. I. Mallick and R. Nath, "Navigating the cyber security landscape: A comprehensive review of cyber-attacks, emerging trends, and recent developments," *World Scientific News*, vol. 190, no. 1, pp. 1–69, 2024.

[2] T. C. Glaessner, T. Kellermann, and V. McNevin, *"Electronic Security Risk Mitigation in Financial Transactions Public Policy Issues"*. World Bank Publications, 2002, vol. 2870.

[3] G. C. Kessler, "An overview of cryptography," 2003.

[4] M. Gardner, *"Codes, ciphers and secret writing"*. Courier Corporation, 1984.

[5] N. Unger, S. Dechand, J. Bonneau, S. Fahl, H. Perl, I. Goldberg, and M. Smith, "Sok: secure messaging," in *2015 IEEE Symposium on Security and Privacy*. IEEE, 2015, pp. 232–249.

[6] M. F. Mushtaq, S. Jamel, A. H. Disina, Z. A. Pindar, N. S. A. Shakir, and M. M. Deris, "A survey on the cryptographic encryption algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 11, 2017.

[7] E. Yavuz, A. K. Koç, U. C. Çabuk, and G. Dalkılıç, "Towards secure e-voting using ethereum blockchain," in *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*. IEEE, 2018, pp. 1–7.

[8] A. Jurišic and A. Menezes, "Elliptic curves and cryptography," *Dr. Dobb's Journal*, pp. 26–36, 1997.

[9] S. Y. Yan, *Cryptanalytic attacks on RSA*. Springer Science & Business Media, 2007.

[10] F. Wilman, "The digital services act (dsa)-an overview," *Available at SSRN 4304586*, 2022.

[11] U. Mukhopadhyay, A. Skjellum, O. Hambolu, J. Oakley, L. Yu, and R. Brooks, "A brief survey of cryptocurrency systems," in *2016 14th annual conference on privacy, security and trust (PST)*. IEEE, 2016, pp. 745–752.

[12] A. J. Di Scala, A. Gangemi, G. Romeo, and G. Vernetti, "Special subsets of addresses for blockchains using the secp256k1 curve," *Mathematics*, vol. 10, no. 15, p. 2746, 2022.

[13] A. Narayanan and J. Clark, "Bitcoin's academic pedigree: The concept of cryptocurrencies is built from forgotten ideas in research literature." *Queue*, vol. 15, no. 4, pp. 20–49, 2017.

[14] M. Platt and P. McBurney, "Sybil in the haystack: A comprehensive review of blockchain consensus mechanisms in search of strong sybil attack resistance," *Algorithms*, vol. 16, no. 1, p. 34, 2023.

[15] M. Bellare, S. Goldwasser, and D. Micciancio, ""pseudo-random" number generation within cryptographic algorithms: The dds case," in *Advances in Cryptology—CRYPTO'97: 17th Annual International Cryptology Conference Santa Barbara, California, USA August 17–21, 1997 Proceedings 17*. Springer, 1997, pp. 277–291.

[16] 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.

[17] J. Guajardo and C. Paar, "Efficient algorithms for elliptic curve cryptosystems," in *Annual International Cryptology Conference*. Springer, 1997, pp. 342–356.

[18] Z. Qing-Sheng, G. Bao-An, X. Shu-Min, and C. Deng-Feng, "Fast elliptic curve digital signature and verify algorithm," *Comput. Eng. Des.*, vol. 29, no. 17, pp. 4425–4427, 2008.

[19] C. Liang and Y. Lin, "Optimization and design of elliptic curve digital signature algorithm," *Chin. J. Electron Devices*, vol. 34, no. 1, pp. 93–98, 2017.

[20] C. Yaru, C. Peiqiang, and C. Zhuang, "An improved scheme for elliptic curve digital signature," *J. Inf. Secur. Res.*, vol. 5, no. 3, pp. 217–222, 2019.

[21] X. Shuai, W. An, and P. Feng, "Elliptic curve digital signature algorithm without modular inverse operation," *Comput. Eng. Appl.*, vol. 56, no. 11, pp. 118–123, 2020.

[22] J. Li and X. Miao, "Analysis and improvement of forward-secure digital signature scheme," *Journal of Jilin University (Information Science Edition)*, vol. 6, 2017.

[23] J. Wang, L. Wu, M. Luo, and D. He, "Secure and efficient two-party ecdsa signature scheme," *J. Commun.*, vol. 42, no. 2, pp. 12–25, 2021.

[24] Z. Jian, Q. Ran, and S. Liyan, "Securing blockchain wallets efficiently based on threshold ecdsa scheme without trusted center," in *2021 Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS)*. IEEE, 2021, pp. 47–51.

[25] Y.-Q. Shi and H. Sun, *"Image and video compression for multimedia engineering: Fundamentals, algorithms, and standards"*. CRC press, 2017.

[26] W. J. Caelli, E. P. Dawson, and S. A. Rea, "Pki, elliptic curve cryptography, and digital signatures," *Computers & Security*, vol. 18, no. 1, pp. 47–66, 1999.

[27] X. Liu, Y. Zhang, D. K. Goswami, J. S. Okasinski, K. Salaita, P. Sun, M. J. Bedzyk, and C. A. Mirkin, "The controlled evolution of a polymer single crystal," *Science*, vol. 307, no. 5716, pp. 1763–1766, 2005.

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

[29] 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.

[30] S.-G. Liu, W.-Q. Chen, and J.-L. Liu, "An efficient double parameter elliptic curve digital signature algorithm for blockchain," *IEEE Access*, vol. 9, pp. 77 058–77 066, 2021.

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

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

[33] C. David, "Ideals, varieties, and algorithms-an introduction to computational algebraic geometry and commutative algebra," *Undergraduate Texts in Mathematics*, 1991.

[34] C. Paar and J. Pelzl, "A textbook for students and practitioners," *Understanding Cryptography, Springer*, 2009.

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

[36] J. R. Vacca, *"Cyber security and IT infrastructure protection"*. Syngress, 2013.

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

[38] A. Iliev and N. Kyurkchiev, "The faster extended euclidean algorithm," in *Collection of scientific works from conference*, 2018, pp. 21–26.

[39] M. Bellare, S. Halevi, A. Sahai, and S. Vadhan, "Many-to-one trapdoor functions and their relation to public-key cryptosystems," in *Annual International Cryptology Conference*. Springer, 1998, pp. 283–298.

[40] D. R. Stinson, *"Cryptography: theory and practice"*. Chapman and Hall/CRC, 2005.

[41] J. Buchmann and J. Buchamann, *"Introduction to cryptography"*. Springer, 2004, vol. 335.

[42] H. Delfs and H. Knebl, "Symmetric-key cryptography," in *Introduction to Cryptography: Principles and Applications*. Springer, 2015, pp. 11–48.

[43] 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.

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

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

[46] R. Banoth and R. Regar, "Asymmetric key cryptography," in *Classical and Modern Cryptography for Beginners*. Springer, 2023, pp. 109–165.

[47] R. Davis, "The data encryption standard in perspective," *IEEE Communications Society Magazine*, vol. 16, no. 6, pp. 5–9, 1978.

[48] C. Nist, "The digital signature standard," *Communications of the ACM*, vol. 35, no. 7, pp. 36–40, 1992.

[49] S. J. Muhammad, H. Chiroma, and M. Mahmud, "Cryptanalytic attacks on rivest, shamir, and adleman RSA cryptosystem: issues and challenges," *J Theor Appl Inf Technol*, vol. 61, no. 1, p. 2349, 2014.

[50] A. V. Meier, "The elgamal cryptosystem," in *Joint Advanced Students Seminar*. TU München, 2005.

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

[52] R. Sobti and G. Geetha, "Cryptographic hash functions: a review," *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 2, p. 461, 2012.

[53] C. Swenson, *"Modern cryptanalysis: techniques for advanced code breaking"*. John Wiley & Sons, 2008.

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

[55] S. Prajapat and R. S. Thakur, "Various approaches towards cryptanalysis," *International Journal of Computer Applications*, vol. 127, no. 14, pp. 15–24, 2015.

[56] E. F. Brickell and A. M. Odlyzko, "Cryptanalysis: A survey of recent results," *Proceedings of the IEEE*, vol. 76, no. 5, pp. 578–593, 1988.

[57] R. Kaur and A. Kaur, "Digital signature," in *2012 International Conference on Computing Sciences*. IEEE, 2012, pp. 295–301.

[58] P. Longa and A. Miri, "Fast and flexible elliptic curve point arithmetic over prime fields," *IEEE Transactions on computers*, vol. 57, no. 3, pp. 289–302, 2008.

[59] D. V. Bailey and C. Paar, "Efficient arithmetic in finite field extensions with application in elliptic curve cryptography," *Journal of cryptology*, vol. 14, pp. 153–176, 2001.

[60] R. Srinivasa and P. Setty, "Efficient mapping methods for elliptic curve cryptography," *Int J of Engineering Science and Technology*, vol. 2, pp. 3651–3656, 2010.

[61] J. Athena, V. Sumathy, and K. Kumar, "An identity attribute–based encryption using elliptic curve digital signature for patient health record maintenance," *International Journal of Communication Systems*, vol. 31, no. 2, p. e3439, 2018.

[62] D. He, H. Wang, L. Wang, J. Shen, and X. Yang, "Efficient certificateless anonymous multi-receiver encryption scheme for mobile devices," *Soft Computing*, vol. 21, pp. 6801–6810, 2017.

[63] D. Avci, "A novel meaningful secret image sharing method based on arabic letters," *Kuwait Journal of Science*, vol. 43, no. 4, 2016.

[64] N. Li, M. Ma, and H. Wang, "an anonymous secure authentication protocol for industrial internet of things," *Sensors*, vol. 24, no. 4, p. 1243, 2024.

[65] C. Christensen, "Review of random curves by neal koblitz," 2009.

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

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

[68] J. W. Bos, C. Costello, and A. Miele, "Elliptic and hyperelliptic curves: A practical security analysis," in *Public-Key Cryptography–PKC 2014: 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings 17*. Springer, 2014, pp. 203–220.

[69] T. Juhas, "The use of elliptic curves in cryptography," Master's thesis, Universitetet i Tromsø, 2007.

[70] O. Alphand, M. Amoretti, T. Claeys, S. Dall'Asta, A. Duda, G. Ferrari, F. Rousseau, B. Tourancheau, L. Veltri, and F. Zanichelli, "Iotchain: A blockchain security architecture for the internet of things," in *2018 IEEE*

wireless communications and networking conference (WCNC). IEEE, 2018, pp. 1–6.

[71] Y. M. Dalal, S. Supreeth, K. Amuthabala, T. Satheesha, P. Asha, and S. Somanath, "Optimizing security: A comparative analysis of rsa, ecc, and dh algorithms," in *2024 IEEE North Karnataka Subsection Flagship International Conference (NKCon)*. IEEE, 2024, pp. 1–6.

[72] A. Blundell-Wignall, "The bitcoin question: Currency versus trust-less transfer technology," 2014.

[73] R. H. Lock, P. F. Lock, K. L. Morgan, E. F. Lock, and D. F. Lock, *"Statistics: Unlocking the power of data"*. John Wiley & Sons, 2020.

[74] R. Garg, *"Blockchain for real world applications"*. John Wiley & Sons, 2023.

[75] T. Pornin, "Deterministic usage of the digital signature algorithm (dsa) and elliptic curve digital signature algorithm (ecdsa)," Tech. Rep., 2013.

[76] A. Menezes, R. Zuccherato, and Y.-H. Wu, *"An elementary introduction to hyperelliptic curves"*. Faculty of Mathematics, University of Waterloo, 1996.

[77] M. Amara and A. Siad, "Elliptic curve cryptography and its applications," in *International workshop on systems, signal processing and their applications, WOSSPA*. IEEE, 2011, pp. 247–250.

[78] U. M. Maurer and S. Wolf, "The diffie–hellman protocol," *Designs, Codes and Cryptography*, vol. 19, no. 2, pp. 147–171, 2000.

[79] P. Barrett, "Implementing the rivest shamir and adleman public key encryption algorithm on a standard digital signal processor," in *Conference on the Theory and Application of Cryptographic Techniques*. Springer, 1986, pp. 311–323.

[80] D. Miller, "Miller (1983) revisited: A reflection on eo research and some suggestions for the future," *Entrepreneurship theory and practice*, vol. 35, no. 5, pp. 873–894, 2011.

[81] 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.

[82] D. Mumford, "An analytic construction of degenerating curves over complete local rings," *Compositio Mathematica*, vol. 24, no. 2, pp. 129–174, 1972.

[83] C. Zhou, B. Hu, Y. Shi, Y.-C. Tian, X. Li, and Y. Zhao, "A unified architectural approach for cyberattack-resilient industrial control systems," *Proceedings of the IEEE*, vol. 109, no. 4, pp. 517–541, 2020.

[84] S. Misra, A. Mukherjee, A. Roy, N. Saurabh, Y. Rahulamathavan, and M. Rajarajan, "Blockchain at the edge: Performance of resource-constrained iot networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 1, pp. 174–183, 2020.

[85] A. Yadav, P. Sharma, and Y. Gigras, "A comparative study of elliptic curve and hyperelliptic curve cryptography methods and an overview of their applications," in *2024 International Conference on Intelligent Systems for Cybersecurity (ISCS).* IEEE, 2024, pp. 01–06.

[86] P. Vijayakumar, "Investigations on hyperelliptic curve cryptography over prime field of different genus curves for wireless systems," Ph.D. dissertation, 2015.

[87] G. Frey and T. Shaska, *"Curves, Jacobians, and cryptography"*. American Mathematical Society Providence, 2019, vol. 724.

[88] S. Kavitha, P. Alphonse, and Y. V. Reddy, "An improved authentication and security on efficient generalized group key agreement using hyper elliptic curve based public key cryptography for iot health care system," *Journal of medical systems*, vol. 43, no. 8, p. 260, 2019.

[89] V. S. Naresh and S. Reddi, "Secure lightweight multi-party key agreement based on hyperelliptic curve diffie–hellman for constraint networks," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 13, p. e6921, 2022.

[90] M. Wang, H. Xue, and T. Zhan, "Fault attacks on hyperelliptic curve discrete logarithm problem over binary field," *Science China Information Sciences*, vol. 57, pp. 1–17, 2014.

[91] M. Conti, N. Dragoni, and V. Lesyk, "A survey of man in the middle attacks," *IEEE communications surveys & tutorials*, vol. 18, no. 3, pp. 2027–2051, 2016.

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

[93] S. Ullah, X.-Y. Li, and L. Zhang, "A review of signcryption schemes based on hyper elliptic curve," in *2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)*. IEEE, 2017, pp. 51–58.

[94] A. Panarello, N. Tapas, G. Merlino, F. Longo, and A. Puliafito, "Blockchain and iot integration: A systematic survey," *Sensors*, vol. 18, no. 8, p. 2575, 2018.

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