INTELLIGENT COMPUTING ON COMPLEX NUMBERS FOR CRYPTOGRAPHIC APPLICATIONS

Ni Ni Hla

UNIVERSITY OF COMPUTER STUDIES, YANGON

MAY, 2024

Intelligent Computing on Complex Numbers for Cryptographic Applications

Ni Ni Hla

University of Computer Studies, Yangon

A thesis submitted to the University of Computer Studies, Yangon in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

MAY, 2024

Statement of Originality

I hereby certify that the work embodied in this dissertation is the result of original research and has not been submitted for a higher degree to any other University or Institution.

---------- ----------

Date Ni Ni Hla

ACKNOWLEDGEMENTS

First of all, I would like to thank His Excellency, the Minister for Science and Technology, for full facilities support during the Ph.D. Course at the University of Computer Studies, Yangon.

Secondly, I would like to express very special thanks to **Dr. Mie Mie Khin**, the Rector, the University of Computer Studies, Yangon, for allowing me to develop this dissertation and giving me general guidance during the period of my study.

I would like to express my deepest gratitude to my supervisor **Dr. Htar Htar Lwin**, Pro-Rector, the University of Computer Studies, Yangon, for her tremendous support and guidance during the course of my doctoral study. Her insights into research has inspired me significantly in my pursuit of research.

I would like to express special thanks to my external examiner **Dr. Aung Htein Maw,** Professor, University of Information Technologies, for his useful comments and suggestions.

I would like to extend my special appreciation and thanks to **Dr. Nilar Aye,** Professor, Head of Faculty of Information Science, the University of Computer Studies, for her useful comments, advice and insight which are invaluable to me.

I would also like to thank my dissertation committee. Without their insightful comments and feedback, this dissertation would not be possible.

I would like to express my respectful gratitude to **Daw Aye Aye Khine**, Professor, Head of English Department, the University of Computer Studies, Yangon, for her valuable supports from the language point of view and pointing out the correct usage in my dissertation.

Finally, I want to express sincerest and deepest gratitude to my husband, my son and my daughter for their unconditional love, support and encouragement. I would not have been where I am today without their endless love and tremendous support. My dissertation is dedicated to them.

ABSTRACT

This research makes an effort to examine the mathematical characteristics of the typical attacks on both traditional and modern ciphers, including the Hill cipher and the elliptic curve cryptosystem. Known-plaintext attack and chosen-ciphertext attack often occur in traditional Hill cipher. Baby-Step, Giant-Step Method, Pollard's Rho Method and Pohlig-Hellman Method can solve the hardness of elliptic curve discrete logarithm problem that is the security of elliptic curve cryptosystem. Generally, finite field arithmetic is used to calculate the Hill cipher and the elliptic curve cryptosystem. The research development uses Java Programming Language to examine the arithmetic properties of finite field arithmetic integrated with complex numbers. The study concludes that the finite field arithmetic foundations are followed by the arithmetic properties of finite field combined with complex numbers. For the Hill cipher and the elliptic curve cryptosystem, the research scheme analyzes not only the arithmetic characteristics of residue matrices and elliptic curve arithmetic integrated with them but also cyclic group orders of points on various kinds of elliptic curves to produce more effective secret codes. According to the study, the arithmetic features of residue matrices and elliptic curve arithmetic integrated with complex numbers come after the fundamentals of the arithmetic. The analysis of the complex plane's point order and curve order indicates that they generally have higher cyclic group orders. As a result, the integration of complex numbers makes the time complexity higher and can protect the common attacks. To create cryptographic non-linear transformation approaches for security improvement, classical ciphers and elliptic curve cryptography utilize their computational capabilities in mathematics on the plane constructed of complex numbers. The research task is to extend non-linear cryptographic transformation techniques by using mathematical properties of residue matrices and elliptic curve arithmetic over the complex plane in order to resist the common attacks on traditional ciphers and modern ciphers including the Hill cipher and the elliptic cryptosystem. The proposed technique needs to double the memory areas to store the keys, however, their security levels are generally squared.

Table of Contents

List of Figures

List of Tables

List of Equations

CHAPTER 1
INTRODUCTION

The origins of cryptography are found in Roman and Egyptian cultures. The Egyptians utilized hieroglyphic writing to send messages to one another about 4,000 years ago. Cryptography is derived from the use of hieroglyph, and it is known as the art and science of keeping message secure [104]. Using cryptography, sensitive data may be protected or sent through unreliable networks. It has to do with the process of converting typical plain text into unreadable text and vice versa. It usually uses mathematical transformations for encoding process known as encryption and decoding process known as decryption [16]. Encryption is the process of converting regular text, sometimes known as plain text, into cipher text, which cannot be read. The process of recovering plain text from the unreadable cipher text is known as decryption or doing the opposite [103]. Therefore, it may be understood only by the person who has authorized access. It is a technique for distributing and storing data in a format that a specific individual can understand and use. With the use of cryptography, it can store and send private data over public networks like the Internet without worrying that anybody else will be able to read it.

A cryptosystem is an implementation system that uses cryptographic methods and supporting tools to provide information security services. A cipher is another name for a cryptosystem. A fundamental cryptosystem is made up of the following components: Plaintext, Encryption Algorithm, Ciphertext, Decryption Algorithm, Encryption Key and Decryption Key [103].

Nowadays, two parties may communicate confidential data over a public network with the use of cryptographic methods, which also prohibit an adversarial eavesdropper from reading the communication's contents. If credit card is used to purchase something online, cryptography must undoubtedly be utilized to ensure that an eavesdropper cannot see the credit card number in the interim. In addition to encryption, if Windows or Mac software is updated, it unknowingly relies on digital signatures to verify the authenticity of the update. The newest kind of digital currency, Bitcoin, employs encryption techniques to ensure its security.

1.1 Traditional Ciphers

Cryptography prior to the modern age is related with traditional ciphers. Traditional ciphers come in two different varieties: substitution ciphers and transposition ciphers [11]. The traditional ciphers are organized into the following categories as shown in Figure 1.1.

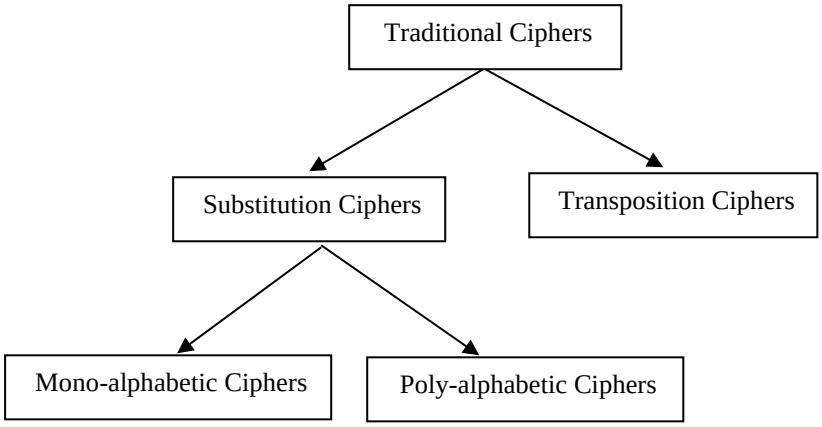

Figure 1.1: Categories of Traditional Ciphers

A substitution cipher replaces one element with another. If the elements in the plaintext are alphabet characters, one character is replaced by another. The substitution ciphers can be grouped into monoalphabetic ciphers and polyalphabetic ciphers. In monoalphabetic ciphers, an element (or a symbol) in the plain text is always altered to the same element (or a symbol) in the cipher text without considering its position in the text. In monoalphabetic ciphers, the mapping between an element in the plain text to an element in the cipher text is always one by one. In polyalphabetic ciphers, each occurrence of an element may have a different substitute. The mapping between an element in the plain text to an element in the cipher text is one-to-many [100,101]. Additive cipher, Shift cipher, Caesar cipher, Multiplicative cipher and Affine cipher are popular monoalphabetic ciphers and Vigenère cipher, Autokey cipher, Playfair cipher, Beaufort cipher, Running key cipher, Porta cipher, Hill cipher, One-Time pad and Rotor cipher are popular polyalphabetic ciphers [11, 107].

A transposition cipher changes the location of the elements. An element in the first place of the plain text may come out in the tenth place of the cipher text. An element in the eight places of the plain text may come out in the first place of the cipher text. In other words, a transposition cipher reorders (transposes) the elements.

There are two methods for permutation of elements. In the first method, the element is written into a table column by column and then transmitted row by row. In the second method, the element is written into a table row by row and then transmitted column by column [101]. Rail-fence cipher, Route cipher, Columnar cipher, Transposition using Matrix and Double transposition are popular transposition ciphers [11, 107].

1.2 Modern Ciphers

Modern ciphers have two main features: diffusion and confusion. The correlation between the plain text and the encrypted text must be concealed via the diffusion concept. This will thwart the opponent's attempt to find the plain text using the encrypted text statistics. The purpose of the confusion concept is to conceal the relationship between the key and the cipher text. This will frustrate the opponent's attempt to find the key using the cipher text. Shannon firstly proposed the idea of a complicated encryption known as a "product cipher", which might create confusion and diffusion [11]. The strength of the cryptographic cipher mechanism depends on how difficult it is for a cryptanalyst to decipher.

Mathematics and computer science have a big impact on modern cryptography. Because cryptographic methods are based on assumptions about computational hardness, it is challenging for an adversary to successfully employ them. Although theoretically possible, it is difficult to break into a well-designed system [115]. These strategies must be continually studied and, if necessary, adjusted considering theoretical developments like faster computer technology and improvements in integer factoring techniques. If effectively developed, these strategies are referred to as "computationally secure". The best theoretically breakable but computationally secure solutions are more difficult to implement in practice than information-theoretically safe protocols, such as the one-time pad, which can be proven to be uncrackable even with infinite computer power.

Modern cryptographic algorithms can be divided into: Symmetric ciphers and Asymmetric ciphers. Symmetric ciphers have the property that the same shared secret key is used for encryption and decryption as shown in Figure 1.2. It is also called private key algorithms. As shown in Figure 1.3, asymmetric ciphers use two different keys: public key for encryption and private key for decryption [1]. There are two types of symmetric-key algorithm: *block cipher* and *stream cipher* [98]. In a stream cipher,

encryption and decryption operate on the basis of one symbol (a bit or byte) at a time. In a block cipher, encryption and decryption operate on the basis of a block of symbols of particular size [99]. Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) are popular modern standard block ciphers used for data encryption. A5 and RC4 are popular standard stream ciphers used in GSM networks and in popular protocols such as SSL and WEP. RSA (Rivest–Shamir–Adleman) and Elliptic Curve Cryptography are popular standard asymmetric ciphers used for smart card security.

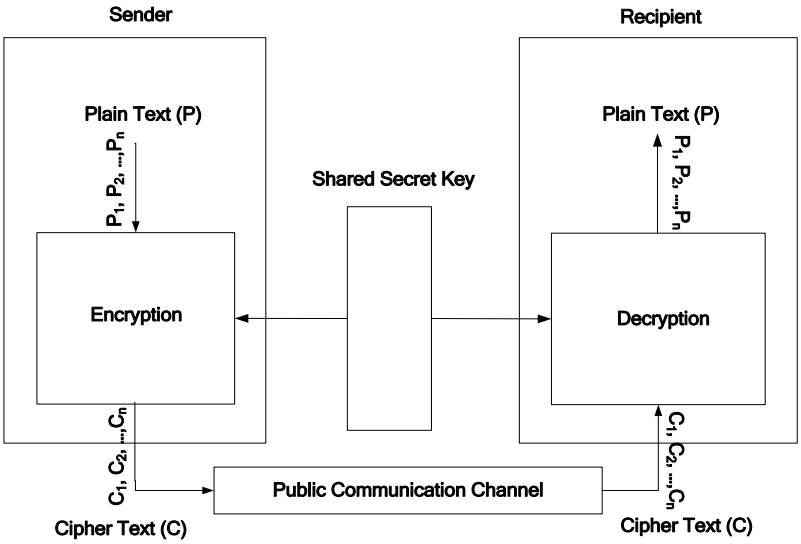

Figure 1.2 : Symmetric Cipher

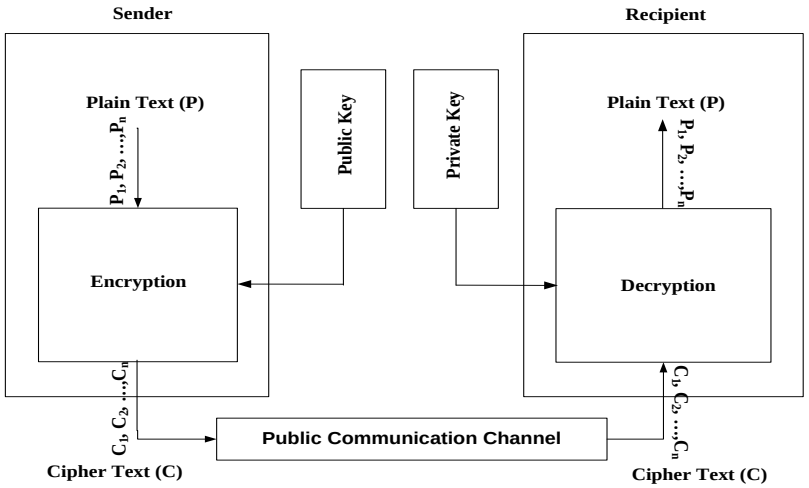

Figure 1.3: Asymmetric Cipher

1.3 Security Goals

There are three different categories of security goals, including availability, integrity, and confidentiality as shown in Figure 1.4. Authorized persons have right to access data are meant confidentiality. Integrity stands as guarantee that data is truthful as well as precise. Availability means an assurance to consistent right to access data by intended persons [9].

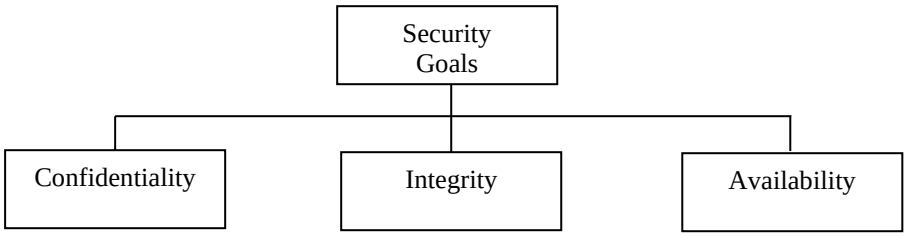

Figure 1.4: Taxonomy of Security Goals

1.3.1 Confidentiality

Confidentiality is possibly the most common feature of information security. It is essential to protect the information that is confidential to the user. An organization must protect against malicious actions that endanger the confidentiality of its information. The main issue in the military is the concealing of sensitive information [9]. In business, keeping some information secret from opponents is essential to the running of the company. Customers' accounts must be kept confidential in banking.

Confidentiality not only applies to the storage of the information, but it also applies to the transmission of information [9]. When a piece of information is transmitted to be stored in a remote computer or when a piece of information is retrieved from a remote computer, it needs to be concealed during transmission [58].

1.3.2 Integrity

Information needs to be often updated. When a client deposits or withdraws money from a bank, her account balance must be updated. Integrity implies that updates can only be made by authorized parties through authorized processes [9]. Integrity violations are not always the consequence of malicious behaviour; unwanted updates to certain information may also result from a system disruption, such a power surge.

1.3.3 Availability

Availability is the third element of information security. The information created and stored by an organization must be accessible to those who have been given permission to access it. If information is unavailable, it is of no use. Since information must be frequently updated, only authorized parties must have access to it. The unavailability of information is harmful for an organization as the lack of confidentiality or integrity [9]. Imagine what would happen to a bank client who were unable to access their accounts to make transactions. Data generated and kept by an association must be accessible to official entities.

1.4 Cryptanalysis and Types of Attacks

Cryptography is the science and art of creating secret codes, and cryptanalysis is the science and art of cracking those codes [57]. Cryptology includes both cryptanalysis and cryptography. Attackers are another name for cryptanalysts. Traditional cryptanalysis requires a unique blend of analytical thinking, using mathematical tools, pattern recognition, patience, effort and luck. In addition to studying cryptography techniques, it also needs to study cryptanalysis techniques. This is essential to fully understand how weak our cryptosystem is, not to decipher someone else's codes. Computer scientists who study cryptanalysis produce more effective secret codes. Cryptanalysis attacks often come in four different types as shown in Figure 1.5 [11, 12, 13].

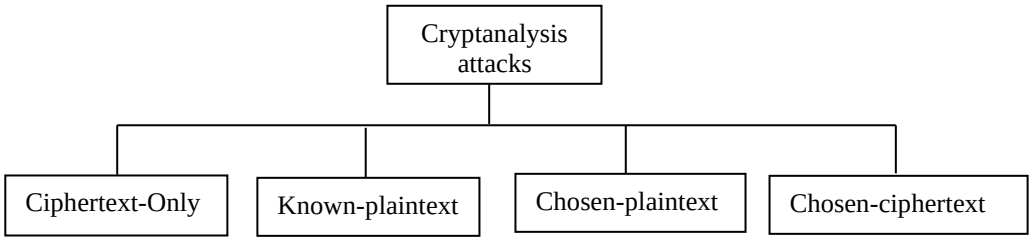

Figure1.5: Cryptanalysis Attacks

1.4.1 Ciphertext-Only Attack

Attacker Eve only has access to certain ciphertext in ciphertext-only attacks. In addition to the plaintext, she looks for the appropriate key. The ciphertext can be intercepted since Eve is said to be aware of the technique. The ciphertext-only attack is the most probable one because the attacker needs only the ciphertext for this attack. A

cipher needs to be particularly resistant to this kind of attack to prevent an adversary from decrypting a message. Figure 1.6 shows the process [11, 12].

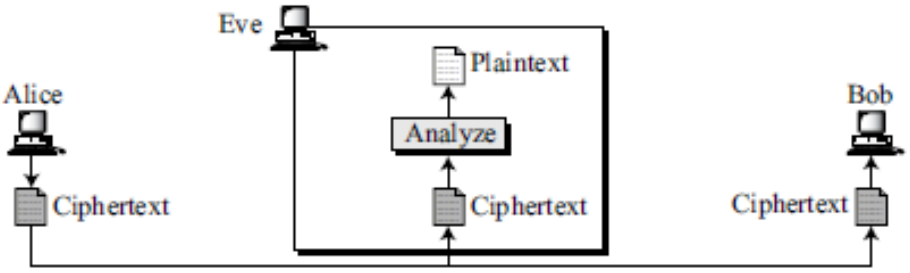

Figure 1.6: Ciphertext-only Attack [11, 12]

Ciphertext-only attacks can employ a variety of techniques. Here, the following are a few typical ones.

1.4.1.1 Brute-Force Attack

In the brute-force approach, also known as the exhaustive-key-search method, the attacker attempts every key that may be used. It is presumed that the attacker is familiar with the algorithm and the key domain (the list of all possible keys). The attacker uses the intercepted cipher to decode the ciphertext using each potential key until the plaintext is readable. Using brute-force attack was a difficult task in the past; it is easier today using a computer. To prevent this type of attack, the number of possible keys must be very large [11].

1.4.1.2 Statistical Attack

A statistical attack can be launched by the cryptanalyst by taking advantage of some built-in features of the plaintext language. For example, it is known that the letter E is the most frequently used letter in English text. The cryptanalyst determines which character appears most frequently in the ciphertext and assumes that it corresponds to the plaintext character E. After finding a few pairs, the analyst can identify the key and use it to decrypt the message. To prevent this type of attack, the cipher should hide the characteristics of the language [11].

1.4.1.3 Pattern Attack

Some ciphers may hide the characteristics of the language but may create some patterns in the ciphertext. A cryptanalyst may use a pattern attack to break the cipher. Therefore, it is important to use ciphers that make the ciphertext look as random as possible [11].

1.4.2 Known-Plaintext Attack

In a known-plaintext attack, attacker Eve has access to several plaintext-ciphertext combinations in addition to the intercepted ciphertext that she is attempting to analyze as shown in Figure 1.7.

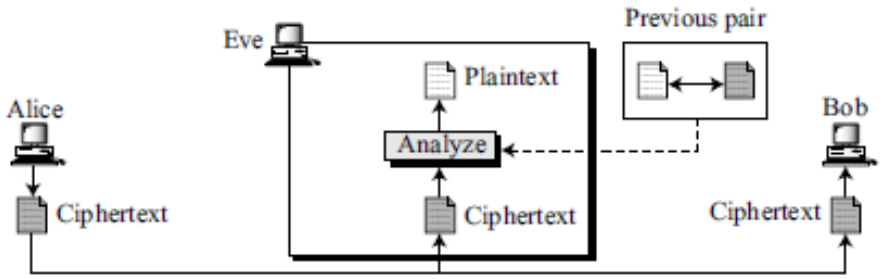

Figure 1.7: Known-plaintext Attack [11, 12]

The combinations of plaintext and ciphertext have been collected earlier. For example, Alice sent Bob a message that was meant to be private, but she later made it public. If Alice has not changed her key, Eve has kept both the ciphertext and the plaintext to use to break the encrypted message sent from Alice to Bob. Eve analyzes the current ciphertext using the relationships between the previous pair. Here, the same techniques that may be utilized in a ciphertext-only attack can be used. This attack is simpler to implement efficiently because Eve has more data to work with when conducting analysis. However, it is less probable to happen since Alice could have changed her key or might not have revealed the contents of any earlier communications [11, 12].

1.4.3 Chosen-Plaintext Attack

Like the known-plaintext attack, the chosen-plaintext attack uses plaintext and ciphertext combinations that the attacker herself has selected. Figure 1.8 shows the procedure of chosen-plaintext attack.

For instance, if Eve obtains access to Alice's computer, this may occur. She can pick a piece of plaintext and intercept the created ciphertext. Since the key is typically integrated in the software used by the sender, she obviously does not have it. Although this kind of attack is significantly simpler to carry out, it is far less probable to appear [11, 12].

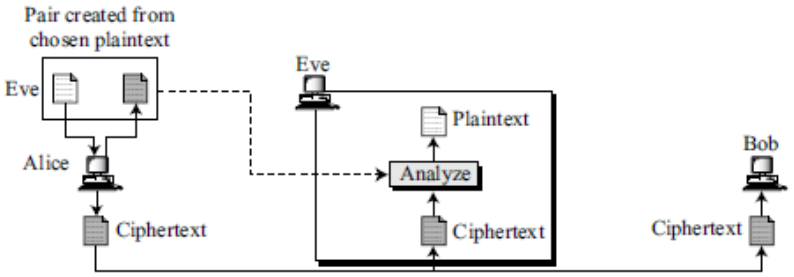

Figure 1.8: Chosen-plaintext Attack [11, 12]

1.4.4 Chosen-Ciphertext Attack

In contrast to the chosen-plaintext attack, the chosen-ciphertext attack involves the attacker selecting some ciphertext and decrypting it to create a pair of ciphertext and plaintext. If Eve gets access to Bob's computer, then this is possible. Figure 1.9 shows the procedure of chosen-ciphertext attack the process [11, 12].

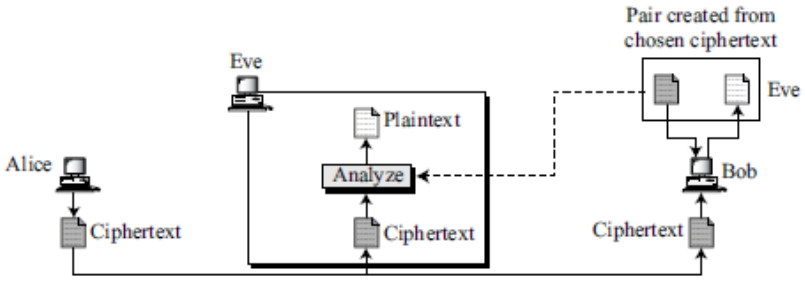

Figure 1.9: Chosen-ciphertext Attack [11, 12]

1.5 Common Attacks on Hill Cipher

It is difficult to analyze Hill ciphers using ciphertext-only. First, since the key to Hill cipher is a $m \times m$ matrix, brute-force attacks on them are exceptionally difficult. For English text, each entry in the matrix may have one of the 26 values. At first glance, this appears to indicate that the size of the key domain is $26^{m \times m}$. However, the multiplicative inverses do not exist for all the matrices. The key domain is smaller, but still huge.

Second, Hill ciphers do not keep the statistics of the plaintext. Attacker is unable to conduct a frequency analysis on single letters, diagrams, or trigrams. A frequency analysis of words of size m could be effective, but it is extremely uncommon for a plaintext to have many strings of the same length.

However, if the attacker knows the value of m and the plaintext-ciphertext combinations for at least m blocks, he/she may perform a known-plaintext attack on the Hill cipher. The blocks should be distinct and might either belong to the same message or to different messages. Attacker can create two $m \times m$ matrices, P for plaintext and C for ciphertext, in which the corresponding rows represent the corresponding known plaintext-ciphertext combinations. Since $C = PK$, attacker can use the relationship $K = CP^{-1}$ to find the key if P is invertible. If P is not invertible, then the attacker needs to use a different set of m plaintext-ciphertext combinations. If the attacker does not know the value of m, she can try different values if m is not very large [11]. The mathematical implementation of known-plaintext attack on Hill cipher is shown in Appendix A.

1.6 Common Attacks on Elliptic Curve

The complexity of solving the Discrete Logarithm Problem (DLP) is deeply important for the security of Public Key Cryptosystem (PKC). PKC is constructed based on the assumption that the DLP is extremely difficult to compute; the more difficult it is, the more security it supports [61]. Therefore, PKC is constructed on a larger group order under large integer to increase the complexity of solving the DLP [68]. General methods of attacking on the Elliptic Curve Discrete Logarithm Problem (ECDLP) can be classified into three groups: methods standing on random walks, methods standing on random walks with special conditions, and methods standing on multiplicative groups. These methods can solve the ECDLP under small integer. The complexity of solving ECDLP determines the security of ECC. Let P and Q be the points on an elliptic curve such that $Q = kP$, where k is an integer number. k is called the discrete logarithm of Q to the base P. Known two points, P and Q, it is unable to compute k, when the group order of the points is enough large [6, 8, 19, 68, 76]. The following general methods of attacking on the ECDLP are studied in this research work.

1.6.1 Baby-Step, Giant-Step Method

Let $P, Q \in E$. Assume that we solve an integer scalar k such that $Q = [k]P$ and P has prime order N.At first, we must compute the order N of P.This method generally performs about $\sqrt{N}$ steps and requires about $\sqrt{N}$ storage. Therefore, this method only works well for memory storage size N . This method follows the procedure below [76]:

1. Fix an integer m such that $m = \lceil \sqrt{N} \rceil$ and compute mP.
2. Compute and store a list of iP for $1 \leq i \leq m$.
3. Compute the points such that $Q - jmP$ for $j = 0, 1, \ldots$ until one of resulting points matches one from the stored list.
4. If $iP = Q - jmP$, then $Q = kP$ with $k \equiv i + jm(mod\ N)$.

The list of points iP are calculated by adding P to $(i-1)P$. It is *the baby-step*.

The list of points $Q - jmP$ are computed by adding $-mP$ to $\mathrm{Q} - (j-1)mP$. It is *the giant-step*. This method may generally perform about *m* steps to find a match and its time complexity is $O(\sqrt{N})$.The mathematical implementation of this method is shown in Appendix B.

1.6.2 Pollard's Rho Method

Let $P, Q \in E$. Assume that we solve an integer scalar k such that $Q = [k]P$ where P has prime order N and $Q \in < P >$. This method generally find two different pair of integers: (a, b) and (a', b') modulo N such that $[a]P + [b]Q = [a']P + [b']Q$. This method follows the procedure below [76]:

1. Select $a, b \in [0, N-1]$ uniformly at random.
2. Compute $[a]P + [b]Q$.
3. Store the triple $(a, b, [a]P + [b]Q)$.
4. Select new pairs (a', b') uniformly at random such that $(a, b) \neq (a', b')$.
5. Compute $[a']P + [b']Q$.
6. Store the new triple $(a', b', [a']P + [b']Q)$.
7. Compute and check the new triple against all previously stored triples until compute a pair (a', b') satisfied with the Eq. (1.1).
8. Compute $k \equiv (a - a')(b' - b)\ mod\ N$.

$$[a]P + [b]Q = [a']P + [b']Q \quad (1.1)$$

The time complexity of this method is $O\left(\sqrt{\pi N/2}\right)$. The diagram of the sequence of resulting points looks like the Greek letter ρ. Therefore, this method is called the Pollard-Rho method. The mathematical implementation of this method is shown in Appendix C.

1.6.3 Pohlig–Hellman Method

Let *P*, $Q \in E$. Assume that we solve an integer scalar *k* such that $Q = [k]P$ where P has prime order *N*.

$$N = \prod_i {q_i}^{e_i} \quad (1.2)$$

The main idea of this method is as follows [76]:

1. Compute the order *N* of *P*.
2. Compute prime factorization of *N* that satisfied the Eq. (1.2).
3. Compute $k(mod\ {q_i}^{e_i})$ for each i.
4. Combine them to obtain $k(mod\ N)$ using the Chinese Remainder theorem.

Let q be a prime and let q^e be the exact power of q dividing N. This method defines k in its base q expansion as the Eq. (1.3).

$$k = k_0 + k_1 q + k_2 q^2 + n \quad (1.3)$$

where $0 \leq k_i < q$. This method evaluates $k(mod\ q_i^e)$ by successively determining $k_0, k_1, k_2, n, k_{e-1}$. This method follows the procedure below:

1. Compute $T = j.\left(\frac{N}{q}.p\right),\ \ 0 \leq j \leq q-1$.
2. Compute $\frac{N}{q}.Q$. It is an element of $k_0\left(\frac{N}{q}.P\right)$ of T.
3. If $e = 1$, stop. Otherwise, continue.
4. Let $Q_1 = Q - k_0 P$.
5. Compute $\frac{N}{q^2}.Q$. It is an element of $k_1\left(\frac{N}{q^2}.P\right)$ of T.
6. If $e = 2$, stop. Otherwise, continue. Assume that we have calculated:
7. $k_1, k_{2,} n, k_{r-1}$ and Q_1, Q_2, n, Q_{r-1}.
8. Let $Q_r = Q_{r-1} - k_{r-1} q^{r-1} P$.
9. Determine k_r such that $\frac{N}{q^{r+1}}.Q_r = k_r\left(\frac{N}{q^r}.P\right)$.

10. If $r = e - 1$, stop. Otherwise, return to step (7).

Then the method computes $k \equiv k_0 + k_1 q + n + k_{e-1} q_{e-1} (mod\ q_e)$. Therefore, early we find k_1. In the same way, the method produces $k_2, k_{3,n}$. We must stop after $r = e - 1$. The time complexity of this method is $O(\sqrt{q})$. In this case, q is the largest prime divisor of N. In practice this method becomes infeasible because N has a large prime divisor. Then it becomes difficult to make and store list *T* to find matches. The mathematical implementation of this method is shown in Appendix D.

1.7 Research Problems and Objectives

Cryptography is the computing science that entails mathematically converting sensitive information from an understandable to a non-understandable format to assure confidentiality, trustworthiness, and accuracy during data transfers over public communication networks. Residue matrices and elliptic curve arithmetic based on modular number arithmetic are often utilized by numerical calculations in traditional and modern cryptographic techniques. In recent years, well-known ciphers such as Hill ciphers and elliptic curve cryptosystems applied non-linear cryptographic transformation methods using on residue matrices and elliptic curve arithmetic over finite fields of integer numbers and binary numbers. Hill ciphers are vulnerable to known-plaintext attacks and chosen-ciphertext attacks while elliptic curve cryptosystems are subject to typical generic attacks like the Baby-Step, Giant-Step, Pollard's Rho, and Pohlig-Hellman methods.

A complex plane which is made of complex numbers over finite fields is becoming more valuable in computing science areas that deal with the applications in cryptography to make them more stable and more secure. As a result, the mathematical properties of residue matrices and elliptic curve arithmetic on complex plane which is made of complex numbers over finite fields are observed to use them in the applications of cryptographic science.

The objective of this research is to extend non-linear cryptographic transformation techniques by using mathematical properties of residue matrices and elliptic curve arithmetic on complex plane which is made of complex numbers over finite fields based on modular arithmetic to improve their security.

1.8 Expected Outcomes

Intelligent computing on the complex plane based on the integration of complex number arithmetic with modular arithmetic is beneficial to cryptographic applications. The proposed techniques need to double the memory areas to store the keys, however their security levels are generally squared. The complex plane supports the non-linear cryptographic transformations not only for traditional ciphers but also elliptic curve cryptography to get more secure for sustainable development. This research points to the importance of complex planes in modern cryptography. The results of this research are expected to become useful for the choice of next-generation cryptographic applications.

1.9 Organization of the Research

The dissertation is organized with nine chapters. The introduction to the research includes the features of traditional ciphers and modern ciphers, the security goals of cryptographic techniques, the concepts of traditional cryptanalysis attacks, common attack on Hill-cipher and common attacks on elliptic curve cryptosystems, the problems and the objectives of the research, and its expected outcomes. The modernizations of classical ciphers, the innovations of modern ciphers and complex number-based innovations on cryptographic techniques are reviewed in the chapter 2. The arithmetic features of finite fields including prime field and binary field and their implementations are described in the chapter 3. The arithmetic features of complex numbers and how to compute complex numbers with prime field and binary field are discussed in the chapter 4. The arithmetic features of matrices and residue matrices are described in the chapter 5. The arithmetic features of elliptic curves and how to compute elliptic curve arithmetic with prime field and binary field are discussed in the chapter 6. The chapter 7 includes the implementations and the analyses of arithmetic properties of residue matrices and elliptic curves by using different arithmetic features. The chapter 8 describes how to apply cryptographic techniques with complex numbers. The chapter 9 presents the discussion of the advantages, the limitation and future work of the research.

CHAPTER 2
LITERATURE REVIEWS

This chapter presents the developed concepts of transforming classical ciphers to modern ciphers and the innovative concepts of modern ciphers. This chapter is structured as follows: The modernization of classical ciphers is reviewed in section 2.1. The section 2.2 studies how to design modern ciphers. The innovations based on complex numbers in computer science are revised in section 2.3. The chapter is summarized in section 2.4.

2.1 Modernization of Classical Ciphers

The study of utilizing mathematics to encrypt and decode data is known as cryptography. Several mathematicians have taken use of mathematics early on and developed implementations for a variety of cryptographic tasks, including encryption, key agreement, authentication, and digital signature. Évariste Galois originally developed Galois Theory and explored a connection between field theory and group theory. The finite field arithmetic called Galois Theory is usually applied to implement cryptographic applications such as classical ciphers and modern ciphers [10, 20]. Julius Caesar created a classical cipher by using an additive group. Additive group and Multiplicative group are combined to create Affine Cipher. Lester S. Hill introduced Hill cipher by using matrix algebra [11]. The Vigenère cipher uses a sequence of interconnected Caesar ciphers that are based on the letters of a keyword to encrypt alphabetic text. It makes use of a polyalphabetic substitution technique. By combining Vigenere and Hill Ciphers, Hamza Touil, Nabil EL Akkad and Khalid Satori developed a hybrid cryptographic approach for text encryption resistant to varying attacks, including statistical attacks [37]. Ashraf A. M. Khalaf, Mona S. Abd El-karim and Hesham F.A. Hamed proposed a triple Hill cipher algorithm to increase the security of encrypted binary data [7]. Deeksha Priya Jha, Rashi Kohli and Archana Gupta developed an encryption algorithm for data security by using matrix properties [22]. Y. S. Yeh, T. C. Wu, C. C. Chang and W. C. Yang provided a more secure number system with different bases to overcome the drawbacks of the Hill cipher [117]. P.L. Sharma and M. Rehan developed the Hill cipher to provide two-fold security to it by using the elements of finite fields and logical operators. They also developed the traditional Hill cipher by using Vandermonde matrix and introducing the elements of finite field which

provides more security [90]. Jessie R. Paragas, Ariel M. Sison and Ruji P. Medina developed the Hill cipher by using multiple rounds of encryption process, cipher block chaining, and hexadecimal substitution box to overcome the drawbacks of the original Hill cipher algorithm [40].

2.2 Innovations in Modern Ciphers

Ron Rivest, Adi Shamir and Leonard Adleman developed a public key encryption algorithm by using modular exponentiation based on the security of discrete logarithm problem [116]. The method of public key encryption known as elliptic curve cryptography (ECC) is based on the algebraic structure of elliptic curves over finite fields. The application of elliptic curves in cryptography was suggested by Victor S. Miller based on the security of elliptic curve discrete logarithm problem. Elliptic curve cryptography is a public key encryption method that was around 20% quicker than the Diffie-Hellman key exchange protocol [105]. For public key cryptosystems, Neal Koblitz also suggested elliptic curves over finite fields. He explained that compared to binary fields, the discrete logarithm issue is more challenging for finite group fields. He also provided a theorem for the existence of no smoothness in cyclic subgroups produced by a global point [85]. The concept of the discrete logarithmic problem utilized in Diffie-public Hellman's key cryptography was expanded to the elliptic curve group by Neal Koblitz, Alfred Menezes, and Scott Vanstone. It offered higher speed, more security, and smaller block sizes [86]. "Guide to Elliptic Curve Cryptography" written by Darrel Hankerson, Alfred Menezes, and Scott Vanstone provides numerous aspects on elliptic curve arithmetic, cryptographic protocols, and implementation challenges [19]. "Elliptic Curves: Number Theory and Cryptography" written by Lawrence C. Washington offers proofs for a variety of elliptic curve understanding theorems [62]. "Computational and Algorithmic Problems in Finite Fields" written by Igor E. Shparlinski provides Polynomial Factorization, Finding Irreducible and Primitive Polynomials, The Distribution of Irreducible and Primitive Polynomials, Bases and Computation in Finite Fields, Coding Theory and Algebraic Curves, Elliptic Curves, Recurrent Sequences in Finite Fields and Cyclic Linear Codes, Recurrent Sequences in Finite Fields and Cyclic Linear Codes, Finite fields and discrete mathematics, and Congruences [38]. "Implementing Elliptic Curve Cryptography" written by Michal Rosing provides step-by-step explanations of basic number theory,

polynomial mathematics, normal basis mathematics, and implementation procedures of elliptic curve cryptography in C programming language and elliptic curve cryptographic protocols [73]. Jorko Teeriaho demonstrated how to use Mathematica to build ECC-DH key exchange, ECC encryption, and Elliptic Curve Digital Signature [44]. By first converting the message into ASCII values and then mapping it into affine points of an elliptic curve by performing point addition, S. Maria Celestin and K. Muneeswaran achieved text cryptography using ECC [93]. With the help of certain additional criteria added to ECC, Sarvana, Suneetha and Chandrasekhar developed a mechanism for safely, non-repudiatively, and authentically communicating with many parties [18]. The delay of point multiplication is lowered by employing a parallel multiple field multiplier approach using Koblitz curves in Jarvinen. K., Helsinki and Skytta J.'s discussion on elliptic curve cryptography [49]. Amara M. and Siad A. examine and compare ECC with RSA to illustrate the function that ECC plays in network security and conclude that ECC is a superior option [65]. Designing an ECC system in the AT89C51 microcontroller with fuzzy modular arithmetic is the work of Gopinath Ganapathy and K. Mani. It was discovered that fuzzy modular arithmetic requires less time for encryption and decryption than non-fuzzy modular arithmetic [33]. Current ECC standards are explained by Scott A. Vansfone, along with their benefits and applications [96]. "Cryptography and Network Security" written by W. Stalling provides a detailed explanation of various cryptographic techniques [108]. When executing encryption and multimedia compression using ECC, Loai Tawalbeh, Moad Mowafi, and Walid Aljoby examined the effectiveness of the encryption, compression, codec compliance, and security [64]. Using a non-singular matrix, Balamurugan, R. Kamalakannan, V. Rahul Ganth, and S. Tamilselvan suggest a quick mapping method. They first converted the message into elliptic curve points, and then they encrypted the points using the ElGamal algorithm utilizing a non-singular matrix. Inverse of the non-singular matrix is employed during decryption [91]. Megha Kolhekar and Anita Jadhav gave a brief history of key exchange and encryption/decryption using ECC. To implement text encryption, they utilized C++. To convert an ASCII value to an elliptic curve coordinate, they employed a mapping table. When decrypting, reverse mapping is employed [72]. "Elliptic Curves in Cryptograph" written by Evan Dummit describes how elliptic curves are used in cryptography for ElGamal encryption, Die-Hellman key exchange, and ElGamal signatures [27].

2.3 Complex Numbers-Based Innovations

"Advanced Engineering Mathematics" written by Erwin Kreyszig provides complex numbers and their functions and mathematical properties [25]. "Analysis for Computer Scientists" written by Michael Oberguggenberger and Alexander Ostermann describes the essential concepts of analysis, covering real and complex numbers, sequences and series, functions, and curves [73]. "Quantum computing for computer scientists" written by N. S. Yanofsky and M. A. Mannucci describes the important role of complex numbers in quantum computing and their mathematical functions and properties [88]. The authors George Stergiopoulos, Miltiadis Kandias and Dimitris Gritzalis used the characteristics of the complex plane and the complex logarithm to approach encryption. They provided a mathematical idea that will be used to cryptography. To ensure robustness against cryptanalysis, the suggested approach uses the features of the complex logarithm along with well-defined procedures from international standards such as AES to implement encryption scheme by converting complex numbers into position vectors in a two-dimensional Cartesian coordinate system known as the complex plane [32]. Khalil Hariss, Maroun Chamoun and Abed Ellatif Samhat provided Somewhat Homomorphic Encryption (SHE) approach that is a novel computation over complex numbers. The suggested system is subsequently made fully homomorphic (FH) and supports an unlimited number of circuit depths using the bootstrapping approach. The key feature of the suggested new scheme is its simplicity because it only relies on addition and multiplication operations over complex numbers, in addition to its homomorphic features and security level. The new scheme is implemented under Python using SAGEMath library and evaluated [54]. Elsayed Mohamed and Hassan Elkamchouchi proposed a new approach that utilizes Gaussian integers instead of rational integers on elliptic curve cryptography. It generates a much larger number of points under the same curve equation and the same prime. The suggested technique uses twice as much storage space as the original prime field to hold cryptographic keys represented by points, but the group order security level is about square and has much greater security [25]. M. Safieh, J. P. Thiers, J. Freudenberger Describe Gaussian Integer Rings and Fields, Point Multiplication over Gaussian Integers, Elliptic Curve Point Multiplication for Complex Expansions, Elliptic Curve Cryptography over Gaussian Integers [69]. Wanarat Juraphanthong and Suradet Jitprapaikulsarn enhanced the security of McEliece cryptosystem with a two-

dimensional finite Gaussian integer. By substituting the one-dimensional linear code with a two-dimensional code employing a finite Gaussian integer, a new system simultaneously increases the key space and the errors to be correct by syndrome decoding. Compared to the classic McEliece cryptosystem, the enhanced cryptosystem achieves a higher security level against key recovering and decoding attacks [113].

2.4 Summary

Nowadays, various scientists made the modifications of classical ciphers and modern ciphers to resist their vulnerable issues by using mathematical non-linear stochastic transformations. Matrix algebra, modular exponentials, elliptic curves, substitution boxes, permutation boxes and logic gates have all been used to create mathematical non-linear stochastic transformations. In many branches of mathematics and engineering, complex numbers are often employed. Complex numbers are now useful to not only computer science but also cryptography to support mathematical non-linear stochastic transformations.

CHAPTER 3
FINITE FIELD ARITHMETIC

This chapter is to present basic concepts of modular arithmetic and to know implementation arithmetic of operations in finite fields by using java *BigInteger* class. This chapter is structured as follows: Finite fields and their characteristics are covered in section 3.2. The implementation of finite field arithmetic operations under prime field and binary field is explained in section 3.3. Field arithmetic procedures on BigInteger under the prime field and binary field are discussed in section 3.4. In section 3.5, a list of some of the algorithms used in the implementation is provided. In section 3.6, the chapter is summarized.

The origins and history of finite fields can be found in the 17^{th} and 18^{th} centuries, however at that time, finite fields were mostly unimportant in mathematics. But, in more recent years, finite fields have played a fundamental role, and they are widely accepted and really growing in significance in a variety of fields, including number theory, algebraic geometry, coding theory, and cryptography [10, 20, 47].

Nowadays, a finite field is an essential structure in cryptography. Finite field arithmetic is frequently used in cryptography applications [31]. To provide structure and effective arithmetic, public key systems based on different discrete logarithm problems are commonly implemented over finite fields [59].

For the development and research of stream ciphers, block ciphers, public key cryptosystems, and cryptographic techniques over elliptic curves, finite field arithmetic operations must be used [17]. The complexity of factoring large composite numbers serves as the foundation for many cryptographic protocols. As a result, the BigInteger class of Java is utilized to provide finite field arithmetic operations for large prime and binary fields [84].

3.1 Finite Fields

A finite field is a field containing a finite number of elements. The representations of the typical number systems (such as the rational numbers Q, real numbers R, and complex numbers C) and their fundamental characteristics are known as *fields* [109, 110]. They are made up of a set F and two operations, addition (denoted by +) and multiplication (denoted by •), that follow to the fundamental principles of arithmetic [15, 31]:

- (F, +) is an additive identity abelian group, represented by the number 0.
- (F\ {0}, ·) is a multiplicative identity abelian group, represented by the number 1.
- The law of commutativity is valid: $x + y = y + x;\quad x.y = y.x,$ for all $x, y, \in F$.
- The law of aassociativity is valid: $(x + y) + z = x + (y + z)$; $(x.y).z = x.(y.z)$ for all $x, y, z \in F$.
- The law of distributivity is valid: $(x + y).z = x.z + y.z$, for all $x, y, z \in F$.

The field is said to be finite if the set F is finite. Galois demonstrated that a field must have p^m elements to be finite, where p is a prime number known as the *characteristic* of F and m is a positive integer. The finite fields typically referred to as Galois Fields and abbreviated as GF (p^m). GF is referred to be a prime field if m = 1. F is referred to be an extension field if $m \geq 2$. The number of elements in a finite field determines its *order*. If the orders of any two fields match, then the fields are said to be *isomorphic* [84, 109].

3.2 Field Arithmetic

Addition and *multiplication* are the two operations available in a field F. It is defined in terms of addition to *subtract* field elements: for $a, b \in$ F, $a - b = a + (-b)$ where $-b$ is the unique element in F that makes $b + (-b) = 0$. $-b$ is referred to as the *opposite or additive inverse* of *b*. The *division* of field elements is similarly defined in terms of multiplication: for $a, b \in$ F with $b \neq 0$, $a/b = a \cdot b^{-1}$ where b^{-1} is the unique element in F that makes $b \cdot b^{-1} = 1 \cdot b^{-1}$ is referred to as the *multiplicative inverse* of *b* [15, 31, 38,47].

3.2.1 Prime Field Arithmetic

Give p a prime value, the integers modulo p are a finite field of order p and consist of the integers $\{0, 1, 2, \ldots, p - 1\}$ with addition and multiplication carried out modulo p. It refers to this field as *GF (p)* and refers to p as the *modulus* of *GF(p)*. For any integer *a*, *a mod p* is the process of dividing one integer (*a*) by another integer (*p*), yielding a unique integer remainder, *r*, where $0 \leq r \leq p-1$. In mathematical term this operation is referred to as *reduction modulo p* [10, 84].

Example 3.1. *Prime field GF (29)* The elements of *GF (29)* are $\{0, 1, 2, \ldots, 28\}$.The following are some examples of arithmetic operations in *GF (29)*.

(i) Addition: 15 + 25 = 11 since 40 mod 29 = 11.

(ii) Subtraction: 15 − 25 = 19 since −10 mod 29 = 19.

(iii) Multiplication: 15 · 5 = 17 since 75 mod 29 = 17.

(iv) Inversion: $17^{-1} = 12$ since 17 · 12 mod 29 = 1.

3.2.2 Binary Field Arithmetic

Binary fields or *characteristic-two finite fields* abbreviated as *GF(2^m)* are terms used to describe finite fields of order 2^m. The elements of *GF (2^m)* are created by using a *polynomial basis representation* shown in the equation (3.1). Here, the elements of *GF (2^m)* are represented as the binary polynomials whose coefficients are in the field *GF (2)* = {0, 1} of degree at most $m - 1$:

$$GF(2^m) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_1x + a_0 : a_i \in \{0,1\}. \quad (3.1)$$

An irreducible binary polynomial *f (x)* of degree *m* is chosen to create the elements of *GF (2^m)*. If *f(x)* is irreducible, it cannot be factored as a product of binary polynomials, each of which must have a degree lower than *m* [2]. When adding field elements, polynomials are added as normal with coefficient arithmetic carried out modulo 2. Field elements are multiplied modulo the *reduction polynomial f(x)*. For any binary polynomial *a(x)*, *a(x)* mod *f(x)* is the process of dividing of *a(x)* by *f(x)*, yielding the unique remainder polynomial *r(x)* of degree less than *m.* In mathematical term, this operation is referred to as *reduction modulo f(x)* [10, 84].

Example 3.2. B*inary field GF (2^4)* The elements of *GF (2^4)* are the 16 binary polynomials of degree at most 3:

0	x^2	x^3	$x^3 + x^2$
1	$x^2 + 1$	$x^3 + 1$	$x^3 + x^2 + 1$
x	$x^2 + x$	$x^3 + x$	$x^3 + x^2 + x$
$x + 1$	$x^2 + x + 1$	$x^3 + x + 1$	$x^3 + x^2 + x + 1$

The following are some examples of arithmetic operations in *GF (2^4)* with reduction Polynomial $f(x) = x^4 + x + 1$.

(i) Addition: $(x^3 + x^2 + 1) + (x^2 + x + 1) = x^3 + x$

(ii) Subtraction: $(x^3 + x^2 + 1) - (x^2 + x + 1) = x^3 + x$

(iii) Multiplication: $(x^3 + x^2 + 1).(x^2 + x + 1) = x^2 + 1$ since
$(x^3 + x^2 + 1).(x^2 + x + 1) = x^5 + x + 1$ and
$(x^5 + x + 1)\ mod\ (x^4 + x + 1) = x^2 + 1.$

(iv) Inversion:$(x^3 + x^2 + 1)^{-1} = x^2$
since $(x^3 + x^2 + 1).x^2\ mod\ (x^4 + x + 1) = 1.$

3.3 Field Arithmetic Operations on BigInteger

Stream ciphers, public key cryptosystems, and cryptographic techniques over elliptic curves all need the implementation of the finite field arithmetic operations: addition, subtraction, division, multiplication, and multiplicative inverse. The Java BigInteger class is utilized to implement finite field arithmetic operations and conduct research under big numbers [84].

3.3.1 Arithmetic Operations of Prime Field

The arithmetic operations of *prime field* need to be implemented to study the research work under prime fields. Therefore, it must implement a *PrimeField* class with methods of arithmetic operations for addition, subtraction, multiplication, and division of elements *(a, b)* in the prime field *GF (p)*. The methods of *PrimeField* class are implemented as follows [84]:

(i) The *addition* method is implemented by *add* and *mod* methods of *BigInteger* class for the logic statement: $a + b = (a + b)\ mod\ p$.

(ii) The *subtraction* method is implemented by *add*, *subtract*, and *mod* methods of *BigInteger* class for the logic statement: $a - b = (a + (-b))\ mod\ p$. In this case, $-b$ is an additive inverse of prime number *p*. The logical statement of additive inverse $-b$ is $(p - b)$.

(iii) The *multiplication* method is implemented by *multiply* and *mod* methods of *BigInteger* class for the logic statement: $a\ .\ b = (a \times b)\ mod\ p$.

(iv) The *division* method is implemented by *multiply* and *modInverse* methods of *BigInteger* class for the logic statement: $a \div b = (a \times b^{-1})\ mod\ p$. In this case, b^{-1} is a multiplicative inverse of prime number *p*.

(v) The *multiplicative inverse* method is adopted from the *modInverse* method.

Example 3.3. *Prime Field (P-192).* Suppose that X and Y are big integers with 192 bits and P is a large prime number with 192 bits. The followings are the results of prime field arithmetic operations for big integers X and Y under a large prime number P.

P= 6277101735386680763835789423207666416083908700390324961279.

X = 188da80eb03090f67cbf20eb43a18800f4ff0afd82ff1012.

Y = 07192b95ffc8da78631011ed6b24cdd573f977a11e794811.

Addition:

Z = X + Y.

Z = 776096614669310688163071032867745522280722465564271335459.

Subtraction:

Z = X - Y.

Z = 427995950082066625353355928307306701552675487709498034177.

Multiplication:

Z = X × Y.

Z = 463980704477630344363893383854114350541460842267886231447 2.

Division:

Z = X % Y.

Z = 102023148406326899839785129772657290128628412720770914977 4.

Multiplicative Inverse of X

Z = 450148766166845920113120162576033894528685541159299270375 0.

3.3.2 Arithmetic Operations of Binary Field

The arithmetic operations of *binary field* need to be implemented to study the research work under binary field. Therefore, it must implement a *BinaryField* class with methods of arithmetic operations for addition, subtraction, multiplication, and division of elements *(a, b)* in the binary field $GF(2^m)$ with reduction polynomial *p*. The methods of *BinaryField* class are implemented as follows [5, 84].

(i) The *addition* method is implemented by *xor* method of *BigInteger* class for the logic statement: $a + b = a \oplus b$. In this case, the *addition* operation is implemented by bitwise XOR operation of all bits of the two operands.

(ii) The *subtraction* method is identical to the *addition* method as above.

(iii) The *multiplication* method is implemented by *shifLeft* and *xor* methods of *BigInteger* class for the logic statement: $a \,.\, b = (a \times b)\ mod\ p$. The

algorithm for multiplication of two polynomials in $GF(2^m)$ is given in Algorithm (3.1).

(iv) The *quotientAndRemainder* method is implemented by *shifLeft* and *setBit* methods of *BigInteger* class for the logic statement: $(q, r) = (a \div b)$. The algorithm to find quotient (q) and remainder (r) from division of two polynomials in $GF(2^m)$ is given in Algorithm (3.2).

(v) The *multiplicativeInverse* method is implemented by *quotientAndRemainder* and *multiplication* methods of *BinaryField* class and *xor* method of *BigInteger* for the logic statement: $b \cdot b^{-1} \bmod p = 1$. The multiplicative inverse b^{-1} is computed by using Extended Euclidean GCD algorithm given in Algorithm (3.3).

(vi) The *division* operation is implemented by *multiplication* and *multiplicativeInverse* methods of *BinaryField* class for the logic statement: $a \div b = (a \times b^{-1}) \bmod p$. In this case, b^{-1} is a multiplicative inverse of prime polynomial p. The multiplicative inverse is adopted from the *multiplicativeInverse* method.

Example 3.4. *Binary Field (K-163).* Suppose that X and Y are big integers with 163 bits and $p(t)$ is a reduction polynomial with 163 bits. The following are the results of binary field arithmetic operations for big integers X and Y under a reduction polynomial $p(t)$.

$p(t) = t^{163} + t^7 + t^6 + t^3 + 1$

X = 2fe13c0537bbc11ac aa07d793de4e6d5e5c94eee8

Y = 289070fb05d38ff58321f2e800536d538ccdaa3d9

Addition:

Z = X + Y.

Z = 7714cfe32684eef49818f913db78b866904e4d31

Subtraction:

Z = X - Y.

Z = 7714cfe32684eef49818f913db78b866904e4d31

Multiplication:

Z = X × Y.

Z = 4d741872162b253d5a381f1f680b47e5c0ad3aa2a

Division:

Z = X % Y.

Z = 498d03bb544d83614e0b5963052f604eb8ec8d0cd

Multiplicative Inverse of X:

Z = 63f514f39f4587684f96c8dd6558e69339a1efed9

3.4 Algorithms

The following algorithms are applied for implementing the multiplication, the division and the multiplicative inverse, the arithmetic operations used in binary field $GF(2^m)$.

Algorithm (3.1). ***shift-and-xor method***

Input: a, b, p as polynomials

Output: result

Begin

Set result= 0;

For (i=0; i<bitLength of b; i++)

begin

If (b_i == 1)

Set result = result xor a.

endIf

Set a = shiftLeft(1) of a.

If (a_{LSB} == 1)

Set a = a xor p.

endIf

end

Return Result

End

Algorithm (3.2). ***shift-and-setBit method***

Input: a, b as polynomials

Output: quotient, remainder

Begin

Set q = 0.

for (term = bitLength of a – bitLength of b;term >= 0; term--)

begin

if (bitLength of a == bitLength of b + term)

Set a = a xor shiftLeft(term of b).

Set quotient = setBit(term of quotient).

endIf

end

Set remainder = a.

Return quotient, remainder

End

Algorithm (3.3). ***Extended Euclidean GCD algorithm***

Input: x, p as polynomials
Output: a

Begin
Set y = x.
Set x = p.
Set a = 0.
Set b = 1.
while $(y \neq 0)$
begin
Set $q = x / y$.
Set $r = x \bmod y$.
Set x = y.
Set y = r.
Set $temp = a \oplus (q \times b)$.
Set a = b.
Set b = temp;
end
if (x = 1) return a.
endIf
End

3.5 Summary

Finite field arithmetic is the fundamental mathematical tool for studying cryptography and related research. This is the first step to study cryptography. Several cryptosystems are mostly constructed using big numbers to increase their security. In comparison to prime field, addition and subtraction operations perform more effectively in binary field shown in Table (3.1). In comparison to binary field, division,

multiplication, and multiplicative inverse operations perform in prime field more effectively shown in Table (3.1). As a result, the software implementation of finite field arithmetic operations in prime field is more effective when using a Java BigInteger class.

Table 3.1 Comparison of Computation Times

Finite Field Arithmetic Operations	**Prime Field (ms/100000times)**	**Binary Field (ms/100000times)**
Addition	31	16
Subtraction	62	16
Division	2, 262	70, 497
Multiplication	156	2808
Multiplicative inverse	2, 028	70, 153

CHAPTER 4
COMPLEX NUMBER ARITHMETIC

This chapter is to know the concepts of complex number arithmetic and to provide the implementations of computing complex numbers integrated with finite fields such as prime field and binary field. This chapter is structured as follows: The mathematical formulas used to the operations on complex numbers are covered in section 4.2. In section 4.3, how to calculate complex numbers by using the arithmetic operations of complex numbers under prime field is described in examples. In section 4.4, how to calculate complex numbers by using the arithmetic operations of complex numbers under binary field is described in examples. The chapter is summarized in section 4.5.

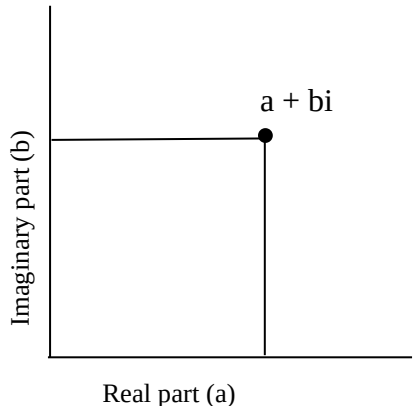

Figure 4.1 : Complex Plane

A complex plane is constructed by using complex numbers integrated with finite fields. A finite field integrated with complex numbers is called Complex Field denoted by $Z(n)$. The complex field over $GF(p)$ is denoted by $Z(GF(p))$. Similarly, the complex field over $GF(2^m)$ is denoted by $Z(GF(2^m))$. A complex field contains a finite number of complex numbers. A complex number is a number that can be expressed in the form $a + bi$, where a and b are integer numbers under one of finite fields, in which a is called the *real part*, and b is called the *imaginary part* [24]. Geometrically, the complex number, $a + bi$, can be identified with the point (a, b) in two-dimensional complex plane by using the horizontal axis for the real part and the vertical axis for the imaginary part [25]. It is demonstrated in Figure 4.1.

4.1 Complex Number Arithmetic

The following rules are applied for addition, subtraction, multiplication, division, reciprocal and scalar multiplication that are the arithmetic operations of complex numbers [3, 25, 81].

Addition: The addition of two complex numbers $x = a_1 + b_1 i$ and $y = a_2 + b_2 i$ is defined by the equation (4.1).

$$x + y = (a_1 + a_2) + (b_1 + b_2)i \tag{4.1}$$

Subtraction. The subtraction of two complex numbers $x = a_1 + b_1 i$ and $y = a_2 + b_2 i$ is defined by the equation (4.2).

$$x - y = (a_1 - a_2) + (b_1 - b_2)i \tag{4.2}$$

Multiplication. The multiplication of two complex numbers $x = a_1 + b_1 i$ and $y = a_2 + b_2 i$ is defined by the equation (4.3).

$$x.y = (a_1 a_2 - b_1 b_2) + (a_1 b_2 - a_2 b_1)i \tag{4.3}$$

Reciprocal. The reciprocal of a nonzero complex number $z = a + bi$ is defined by the equation (4.4).

$$\frac{1}{z} = z^{-1} = \frac{a}{a^2+b^2} - \frac{b}{a^2+b^2}\, i \tag{4.4}$$

Division. The division of two complex numbers $x = a_1 + b_1 i$ and $y = a_2 + b_2 i$ is defined by the equation (4.5).

$$\frac{x}{y} = x.y^{-1} \tag{4.5}$$

Scalar Multiplication. The multiplication of a complex number $z = a + bi$ and the scalar integer k is defined by the equation (4.6).

$$k.z = k.a + k.bi \tag{4.6}$$

4.2 Complex Number Arithmetic on Prime Field *(GF (p))*

Complex numbers in the complex field *Z(GF(p))* can be computed using complex number arithmetic integrated with modular arithmetic over a prime field

GF(p) as shown in Example 4.1. The arithmetic properties of complex numbers in the *Z (GF(p))* are analysed and their experiments are described in section 7.2.2 [99].

Example (4.1). C*omplex Field over GF(p).* Let two complex numbers, $x = 1 + 2i$ and $y = 2 + 1i$, be in $Z(GF(7)$. The followings demonstrate arithmetic operations of complex numbers in $Z(GF(7)$.

Addition.

$$x + y = 3 + 3i \quad since\ (1 + 2)\ mod\ 7 + \big((2 + 1) mod\ 7\big)i$$

Subtraction.

$$x - y = 6 + 1i \quad since\ (1 - 2)\ mod\ 7 + \big((2 - 1) mod\ 7\big)i$$

Multiplication.

$$x.y = 5i \quad since\ (1.2 - 2.1)\ mod\ 7 + \big((1.1 + 2.2) mod\ 7\big)i$$

Inversion.

$$y^{-1} = 6 + 4i \quad since\ \left(\tfrac{2}{4+1} mod\ 7\right) + \left(-\tfrac{1}{4+1}\ mod\ 7\right) i$$

Division.

$$\frac{\mathrm{x}}{\mathrm{y}} = 5 + 2i \quad since \quad (1 + 2i) \times (6 + 4i)$$

Scalar Multiplication.

$$5x = 5 + 3i \quad since \quad (5.1) mod\ 7\ + \big((5.2) mod\ 7\big)i$$

4.3 Complex Number Arithmetic on Binary Field *GF (2^m)*

Complex numbers in the complex field *Z (GF (2^m))* can be computed using complex number arithmetic integrated with modular arithmetic over a binary field *GF (2^m)* as shown in Example 4.3. The arithmetic properties of complex numbers in the *Z (GF (2^m))* are analysed and their experiments are described in section 7.2.3 [99].

Example (4.2). Table 4.1 shows the power representations of g and corresponding binary representations for elements of *GF (2^3)* generated by the reduction polynomial $f(x) = x^3 + x + 1$. The element of g = (010) is a generator of *GF (2^3).*

Table 4.1: Power and Binary Representations

Power	Binary	Power	Binary	Power	Binary	Power	Binary
0	000	g	010	g^3	011	g^5	111
1	001	g^2	100	g^4	110	g^6	101

Example (4.3). *Complex Field over GF (2^m)).* Let two complex numbers, $x = 1 + 2i \ and \ \ y = 2 + 1i,$ in $Z\ (GF(\ f\ (x)))$. They can be represented by the power of g. Then $x = 1 + gi \ \ and \ y = g + 1i$. The following demonstrate arithmetic operations of complex numbers in *Z (GF (f(x)))*.

Addition.

$$x + y = 3 + 3i \ \ since \ (001 \oplus 010) + (010 \oplus 001)i \ \ and \ x + y = g^3 + g^3 i$$

Subtraction.

$$x + y = 3 + 3i \ \ since \ (001 \oplus 010) + (010 \oplus 001)i \ \ and \ x + y = g^3 + g^3 i$$

Multiplication.

$$x.y = 5i$$

$$since \ \ (1 + gi) \times (g + 1i) = (1.g - g.1) + (1.1 + g.g)i = (1 + g^2)i = g^6 i$$

Inversion.

$$y^{-1} = 4 + 2i$$

$$since \ \ y^{-1} = (g + 1i)^{-1} = \frac{g}{g^2 + 1} + \frac{1}{g^2 + 1} i = \frac{g}{g^6} + \frac{1}{g^6} i = g^{-5} + g^{-6} i = g^2 + gi$$

Division.

$$\frac{x}{y} = 1i \quad since \ \ (1 + gi) \times (g + 1i)^{-1} = 1\text{i}$$

Scalar Multiplication.

$$5x = 5 + 1i \quad since \quad 5x = g^6(1 + gi) = g^6 + 1i$$

4.4 Summary

The sections 4.3 and 4.4 prove that complex numbers over finite fields satisfy the arithmetic properties of rational numbers over finite fields and they also perform the same as arithmetic operations of rational numbers over finite fields. Therefore, cryptographic applications can be constructed by using the complex numbers over finite fields such as prime field and binary field. The security of cryptographic applications is increased by the challenge of computing on complex number arithmetic over finite fields.

CHAPTER 5
MATRIX ALGEBRA

This chapter is to provide a detailed implementation for arithmetic operations of matrices computing over finite fields. This work supports to implement, analyze and study the arithmetic properties of residue matrices including complex numbers. The organization of this chapter is as follows. The section 5.1 includes basic concepts of matrix algebra. Section 5.2 describes detailed arithmetic operations of traditional matrices and their arithmetic properties. The section 5.3 describes the procedures to compute additive inverse and multiplicative inverse of a matrix useful for a pair of keys for encryption scheme and decryption scheme. The section 5.4 describes in details arithmetic operations of residue matrices useful for cryptographic techniques. The section 5.5 summarizes this chapter.

5.1 Matrix

A matrix is a collection of i x j components that are arranged in a two-dimensional array, where i is the number of rows and j is the number of columns. Typically, a matrix is denoted by a capitalized letter such as M and the component m_{ij} is allocated in the i^{th} row and the j^{th} column [26, 63]. The following is a matrix.

$$\mathrm{M} = \begin{bmatrix} 1 & 1 \\ 0 & 2 \\ 1 & 3 \end{bmatrix}$$

This matrix is known as a "3 by 2" matrix since it contains three rows and two columns. The components of the matrix are numbered in the following order [41]:

$$\mathrm{M} = \begin{bmatrix} a_{11} & a_{12} \dots & a_{1n} \\ a_{21} & a_{22} \dots & a_{2n} \\ \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} \dots & a_{mn} \end{bmatrix}$$

In the matrix, the first subscript denotes the row, and the second subscript denotes the column. M is $m \times n$ matrix including components a_{ij} [26, 63].

A matrix with only one row (i = 1) is a *row matrix* that is also called a *row vector* while a matrix with only one column (j = 1) is a *column matrix* that is also called a *column vector* [26, 63].

The followings are a row matrix, A and a column matrix, B.

$$A = [a_{i1} \quad a_{i2} \ldots \quad a_{in}],$$

$$B = \begin{bmatrix} b_{1j} \\ b_{2j} \\ \vdots \\ b_{mj} \end{bmatrix}$$

A matrix including the same number of rows and columns (i = j) is a *square matrix* and a square matrix that has $x_{ij} = x_{ji}$ for every i and j is said to be *symmetric* [26, 63]. In the followings matrix A is symmetric but matrix B is not symmetric.

$$A = \begin{bmatrix} 9 & 3 & 5 \\ 3 & 6 & 4 \\ 5 & 4 & 7 \end{bmatrix}, \quad B = \begin{bmatrix} 9 & 2 & 5 \\ 3 & 6 & 2 \\ 5 & 4 & 7 \end{bmatrix}$$

A *diagonal matrix* is a symmetric matrix where all the off-diagonal components are 0 but each of components $a_{11}, a_{22}, \ldots, a_{jj}$ defined as a *main diagonal* is set a value [26, 63]. In the following, matrix A is a diagonal matrix.

$$A = \begin{bmatrix} 9 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 7 \end{bmatrix}$$

A matrix with all rows and columns set to 0's is called an *additive identity matrix* and is signified as 0. A square matrix with 1's on the main diagonal and 0's elsewhere is called an *identity matrix* and is signified as I [26, 63]. The followings are additive identity matrix and identity matrix.

$$0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Two matrices are equal if they belong to the same number of rows and columns and the contents of their corresponding components are equal. In symbols, M = N if they have $m_{ij} = n_{ij}$ for all $\imath`s$ and $\jmath`s$.

The $(i,j)^{th}$ element or entry of A is the element a_{ij}, that is, the number in the i^{th} row and j^{th} column of A. A convenient shorthand notation for expressing the matrix A is to write $A = [a_{ij}]$, which indicates that A is the matrix with its $(i,j)^{th}$ element equal to a_{ij} [26, 63].

5.2 Matrix Arithmetic Operations

To either add or subtract two matrices, they must have both the same number of rows and the same number of columns. Two matrices with the same number of rows and columns can be added and subtracted [53].

1) **Addition.** The addition of two matrices, A and B, is resulted as $\mathrm{R} = \mathrm{A} + \mathrm{B}$ such that $r_{ij} = a_{ij} + b_{ij}$ for all i and j. This implies that the results are simply obtained by adding each member of the two matrices individually. If A and B are $\mathrm{m} \times \mathrm{n}$ matrices, then the result of A + B is calculated in the following procedure [26].

$$\mathrm{A} + \mathrm{B} = \begin{bmatrix} a_{11} & a_{12} \dots & a_{1n} \\ a_{21} & a_{22} \dots & a_{2n} \\ \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} \dots & a_{mn} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \dots & b_{1n} \\ b_{21} & b_{22} \dots & b_{2n} \\ \vdots & \vdots & \vdots \\ b_{m1} & b_{m2} \dots & b_{mn} \end{bmatrix}$$

$$\mathrm{R} = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} \dots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} \dots & a_{2n} + b_{2n} \\ \vdots & \vdots & \vdots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} \dots & a_{mn} + b_{mn} \end{bmatrix}.$$

2) **Subtraction.** The subtraction of two matrices, A and B, is resulted as $\mathrm{R} = \mathrm{A} - \mathrm{B}$ such that $r_{ij} = a_{ij} - b_{ij}$ for all i and j. This implies that the results are simply obtained by subtracting each member of the two matrices individually. If A and B are $\mathrm{m} \times \mathrm{n}$ matrices, then the result of $\mathrm{A} - \mathrm{B}$ is calculated in the following procedure [26].

$$\mathrm{A} - \mathrm{B} = \begin{bmatrix} a_{11} & a_{12} \dots & a_{1n} \\ a_{21} & a_{22} \dots & a_{2n} \\ \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} \dots & a_{mn} \end{bmatrix} - \begin{bmatrix} b_{11} & b_{12} \dots & b_{1n} \\ b_{21} & b_{22} \dots & b_{2n} \\ \vdots & \vdots & \vdots \\ b_{m1} & b_{m2} \dots & b_{mn} \end{bmatrix}$$

$$\mathrm{R} = \begin{bmatrix} a_{11} - b_{11} & a_{12} - b_{12} \dots & a_{1n} - b_{1n} \\ a_{21} - b_{21} & a_{22} - b_{22} \dots & a_{2n} - b_{2n} \\ \vdots & \vdots & \vdots \\ a_{m1} - b_{m1} & a_{m2} - b_{m2} \dots & a_{mn} - b_{mn} \end{bmatrix}.$$

3) **Matrix Multiplication.** Two matrices can be multiplied if the number of columns of the first matrix is the same as the number of rows of the second matrix. In symbols, if A is a $\mathrm{p} \times \mathrm{k}$ matrix and B is a $\mathrm{k} \times \mathrm{q}$ matrix, the product matrix of the two matrices is a matrix R of size $\mathrm{p} \times \mathrm{q}$. Each component of product matrix R is calculated as $r_{ij} = \sum a_{ik} b_{kj}$. The result of $\mathrm{A} \times \mathrm{B}$ is calculated as following. The number of columns in the first matrix must match the number of rows in the second matrix for matrix multiplication to be valid. These criteria mean that matrix multiplication is typically *not commutative*. This implies that $\mathrm{A} \times \mathrm{B} \neq \mathrm{B} \times \mathrm{A}$. And

even if $A \times B$ is a legal operation, there is no assurance that $B \times A$ will also be legal [26].

$$\mathrm{R}_{3\times2} = \mathrm{A}_{3\times3} \times \mathrm{B}_{3\times2} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix}$$

$$\mathrm{R} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} \\ a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} \\ a_{31}b_{11} + a_{32}b_{21} + a_{33}b_{31} & a_{31}b_{12} + a_{32}b_{22} + a_{33}b_{32} \end{bmatrix}$$

4) **Matrix-Scalar Multiplication.** The multiplication of a matrix by a scalar value is defined as scalar multiplication of a matrix. If M is an p × q matrix and s is a scalar value, R = sM is a matrix of size p × q, where $r_{ij} = s \times m_{ij}$. This implies that the scalar is simply multiplied by the matrix component for each component in the product matrix. The result of s × M is calculated as follows. Scalar-based matrix multiplication *is commutative*. Therefore, s × M = M×s [26].

$$\mathrm{M}_{3\times2} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \\ m_{31} & m_{32} \end{bmatrix}$$

$$\mathrm{R} = \mathrm{s} \times \mathrm{M}_{3\times2} = \begin{bmatrix} s \times m_{11} & s \times m_{12} \\ s \times m_{21} & s \times m_{22} \\ s \times m_{31} & s \times m_{32} \end{bmatrix}$$

5.3 Inverses of Matrix

In cryptography, inverses of matrix such as additive inverse and multiplicative inverse are often utilized to generate a pair of keys for encryption scheme and decryption scheme [10, 77].

1) **Determinant.** The determinant of an equal matrix M with size q × q signified as det(M) is a scalar determined recursively as following [26, 77].
   - If $\mathrm{q} = 1, \det(\mathrm{M}) = m_{11.}$
   - If $\mathrm{q} > 1, \det(\mathrm{M}) = \sum_{\mathrm{i}=1}^{\mathrm{q}} (-1)^{\mathrm{i+j}} \times \mathrm{m_{ij}} \times \det(\mathrm{M_{ij}})$.

     In this situation, $\mathrm{M_{ij}}$ is a matrix M determined by the i^{th} row and j^{th} column. The determinant can be calculated only for an equal matrix.

2) **Additive Inverse.** The additive inverse of matrix M is another matrix N such that $\mathrm{M} + \mathrm{N} = 0$. As such, they have $\mathrm{n_{ij}} = -\mathrm{m_{ij}}$ for all $i's$ and $j's$. Generally, the additive inverse of M is characterized by $-\mathrm{M}$. The additive inverse of M is calculated in the following procedure [26, 77].

$$\mathrm{M} = \begin{bmatrix} m_{11} & m_{12} \dots & m_{1j} \\ m_{21} & m_{22} \dots & m_{2j} \\ \vdots & \vdots & \vdots \\ m_{i1} & m_{i2} \dots & m_{ij} \end{bmatrix}$$

$$\mathrm{N} = -\mathrm{M} = \begin{bmatrix} -m_{11} & -m_{12} \dots & -m_{1j} \\ -m_{21} & -m_{22} \dots & -m_{2j} \\ \vdots & \vdots & \vdots \\ -m_{i1} & -m_{i2} \dots & -m_{ij} \end{bmatrix}$$

$$\mathrm{M} + \mathrm{N} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} = 0$$

3) Multiplicative Inverse. The multiplicative inverse of a matrix can be calculated only for square matrices. The multiplicative inverse of a matrix M is a matrix N such that $\mathrm{M} \times \mathrm{N} = \mathrm{N} \times \mathrm{M} = \mathrm{I}$. Generally, the multiplicative inverse matrix of M is characterized by as M^{-1}. The matrices have their reciprocals only if $\det(\mathrm{M}) \neq 0$. The multiplicative inverse of a matrix M is calculated in the following procedure [26, 77].

$$\mathrm{M} = \begin{bmatrix} -1 & 1 & 2 \\ 3 & -1 & 1 \\ -1 & 3 & 4 \end{bmatrix}$$

$$\begin{aligned} \det \mathrm{M} &= (-1)\begin{vmatrix} -1 & 1 \\ 3 & 4 \end{vmatrix} - (1)\begin{vmatrix} 3 & 1 \\ -1 & 4 \end{vmatrix} + 2\begin{vmatrix} 3 & -1 \\ -1 & 3 \end{vmatrix} \\ &= (-1)\{-4-3\} - (1)\{12-(-1)\} + 2\{9-(1)\} \\ &= 7 - 13 + 16 = 10 \neq 0 \end{aligned}$$

$$C_{11} = (-1)^{1+1}\begin{vmatrix} -1 & 1 \\ 3 & 4 \end{vmatrix} = -7.$$

$$C_{12} = (-1)^{1+2}\begin{vmatrix} 3 & 1 \\ -1 & 4 \end{vmatrix} = -13.$$

$$C_{13} = (-1)^{1+3}\begin{vmatrix} 3 & -1 \\ -1 & 3 \end{vmatrix} = 8.$$

$$C_{21} = (-1)^{2+1}\begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} = 2.$$

$$C_{22} = (-1)^{2+2}\begin{vmatrix} -1 & 2 \\ -1 & 4 \end{vmatrix} = -2.$$

$$C_{23} = (-1)^{2+3}\begin{vmatrix} -1 & 1 \\ -1 & 3 \end{vmatrix} = 2.$$

$$C_{31} = (-1)^{3+1}\begin{vmatrix} 1 & 2 \\ -1 & 1 \end{vmatrix} = 3.$$

$$C_{32} = (-1)^{3+2}\begin{vmatrix} -1 & 2 \\ 3 & 1 \end{vmatrix} = 7.$$

$$C_{33} = (-1)^{3+3}\begin{vmatrix} -1 & 1 \\ 3 & -1 \end{vmatrix} = -2.$$

$$M^{-1} = \frac{1}{\det M}\begin{bmatrix} C_{11} & C_{21} & C_{31} \\ C_{12} & C_{22} & C_{32} \\ C_{13} & C_{23} & C_{33} \end{bmatrix}$$

$$= \frac{1}{10}\begin{bmatrix} -7 & 2 & 3 \\ -13 & -2 & 7 \\ 8 & 2 & -2 \end{bmatrix}$$

$$= \begin{bmatrix} -0.7 & 0.2 & 0.3 \\ -1.3 & -0.2 & 0.7 \\ 0.8 & 0.2 & -0.2 \end{bmatrix}$$

$$M \times M^{-1} = \begin{bmatrix} -1 & 1 & 2 \\ 3 & -1 & 1 \\ -1 & 3 & 4 \end{bmatrix}\begin{bmatrix} -0.7 & 0.2 & 0.3 \\ -1.3 & -0.2 & 0.7 \\ 0.8 & 0.2 & -0.2 \end{bmatrix}$$

$$= \begin{bmatrix} .7 - 1.3 + 1.6 & -.2 - .2 + 0.4 & -.3 + .7 - .4 \\ -2.1 + 1.3 + .8 & .6 + .2 + .2 & .9 - .7 - .2 \\ .7 - 3.9 + 3.2 & -.2 - .6 + 0.8 & -.3 + 2.1 - .8 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I$$

$$M^{-1} \times M = \begin{bmatrix} -0.7 & 0.2 & 0.3 \\ -1.3 & -0.2 & 0.7 \\ 0.8 & 0.2 & -0.2 \end{bmatrix}\begin{bmatrix} -1 & 1 & 2 \\ 3 & -1 & 1 \\ -1 & 3 & 4 \end{bmatrix}$$

$$= \begin{bmatrix} .7 + .6 - .3 & -.7 - .2 + .9 & -1.4 + .2 + 1.2 \\ 1.3 - .6 - .7 & -1.3 + .2 + 2.1 & -2.6 - 0.2 + 2.8 \\ -.8 + .6 + .2 & .8 - .2 - .6 & 1.6 + .2 - .8 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I$$

5.4 Residue Matrices

Cryptography frequently utilizes residue matrices in which all components are in Z_p where p is a prime number. Except for the fact that the arithmetic operations are carried out in modular arithmetic, all matrix transformations of residue matrices are carried out in the same way as for the integer matrices. The residue matrices follow the properties of the integer matrices [10, 26, 42, 77].

1) **Addition.** The addition of two residue matrices, A and B, is resulted as $R = A + B$ such as $r_{ij} = a_{ij} + b_{ij}\ mod\ p$ for all i and j in case of the number of their rows and columns. This implies that the results are simply obtained by adding each member of the two matrices individually in Z_p. If A and B are $m \times n$ matrices, then the result of A + B is calculated in the following procedure [77].

$$A + B = \begin{bmatrix} a_{11} & a_{12} \dots & a_{1n} \\ a_{21} & a_{22} \dots & a_{2n} \\ \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} \dots & a_{mn} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \dots & b_{1n} \\ b_{21} & b_{22} \dots & b_{2n} \\ \vdots & \vdots & \vdots \\ b_{m1} & b_{m2} \dots & b_{mn} \end{bmatrix} in\ Z_p$$

$$R = \begin{bmatrix} a_{11} + b_{11}\ mod\ p & a_{12} + b_{12}\ mod\ p \dots & a_{1n} + b_{1n}\ mod\ p \\ a_{21} + b_{21}\ mod\ p & a_{22} + b_{22} mod\ p \dots & a_{2n} + b_{2n}\ mod\ p \\ \vdots & \vdots & \vdots \\ a_{m1} + b_{m1}\ mod\ p & a_{m2} + b_{m2} mod\ p \dots & a_{mn} + b_{mn}\ mod\ p \end{bmatrix}.$$

2) Subtraction. The subtraction of two residue matrices, Aand B, is resulted as $R = A - B$ such as $z_{ij} = a_{ij} - b_{ij}\ mod\ p$ for all i and j in case of the number of their rows and columns. This implies that the results are simply obtained by subtracting each member of the two matrices individually in Z_p. If A and B are $m \times n$ matrices, then the result of $A - B$ is calculated in the following procedure [77].

$$A - B = \begin{bmatrix} a_{11} & a_{12} \dots & a_{1n} \\ a_{21} & a_{22} \dots & a_{2n} \\ \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} \dots & a_{mn} \end{bmatrix} - \begin{bmatrix} b_{11} & b_{12} \dots & b_{1n} \\ b_{21} & b_{22} \dots & b_{2n} \\ \vdots & \vdots & \vdots \\ b_{m1} & b_{m2} \dots & b_{mn} \end{bmatrix} in\ Z_p$$

$$R = \begin{bmatrix} a_{11} - b_{11}\ mod\ p & a_{12} - b_{12}\ mod\ p \dots & a_{1n} - b_{1n}\ mod\ p \\ a_{21} - b_{21}\ mod\ p & a_{22} - b_{22} mod\ p \dots & a_{2n} - b_{2n}\ mod\ p \\ \vdots & \vdots & \vdots \\ a_{m1} - b_{m1}\ mod\ p & a_{m2} - b_{m2} mod\ p \dots & a_{mn} - b_{mn}\ mod\ p \end{bmatrix}$$

3) Matrix Multiplication. Two residue matrices can be multiplied if the number of columns of the first matrix is the same as the number of rows of the second matrix. In symbols, if A is a $p \times k$ matrix and B is a $k \times q$ matrix, the product matrix of the two is a matrix R of size $p \times q$ in Z_p. Each component of product matrix R is calculated as $r_{ij} = \sum a_{ik} b_{kj}\ mod\ p$.The result of $A \times B$ is calculated in the following procedure [77].

$$R = A_{3\times3} \times B_{3\times2} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} \text{ in } Z_p$$

$$R_{3\times2} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} mod\ p & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} mod\ p \\ a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} mod\ p & a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} mod\ p \\ a_{31}b_{11} + a_{32}b_{21} + a_{33}b_{31} mod\ p & a_{31}b_{12} + a_{32}b_{22} + a_{33}b_{32} mod\ p \end{bmatrix}$$

4) Matrix-Scalar Multiplication. The multiplication of a residue matrix by a scalar value is defined as scalar multiplication of a matrix. If M is an $p \times q$ matrix and s is a scalar value, $R = s \times M$ is a matrix of size $p \times q$ in Z_p,where

$r_{ij} = s \times m_{ij}\ mod\ p$. The result of $\mathrm{s} \times \mathrm{M}$ is calculated in the following procedure [77].

$$\mathrm{M}_{3\times2} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \\ m_{31} & m_{32} \end{bmatrix} in\ Z_p$$

$$\mathrm{R} = \mathrm{s} \times \mathrm{M}_{3\times2} = \begin{bmatrix} s \times m_{11}\ mod\ p & s \times m_{12}\ mod\ p \\ s \times m_{21}\ mod\ p & s \times m_{22}\ mod\ p \\ s \times m_{31}\ mod\ p & s \times m_{32}\ mod\ p \end{bmatrix}$$

5) **Additive Inverse.** Additive inverse of a residue matrix A is another matrix B such that $\mathrm{A} + \mathrm{B} = 0$ in Z_p. As such, they have $\mathrm{b_{ij}} = -\mathrm{a_{ij}}\ mod\ p$ for all $i's$ and $j's$. Generally, the opposite of A is characterized by $-\mathrm{A}$. The result of $-\mathrm{A}$ is calculated in the following procedure [77].

$$\mathrm{A} = \begin{bmatrix} a_{11} & a_{12} \dots & a_{1n} \\ a_{21} & a_{22} \dots & a_{2n} \\ \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} \dots & a_{mn} \end{bmatrix} \text{ in } Z_p.$$

$$\mathrm{B} = -\mathrm{A} = \begin{bmatrix} -a_{11}\ mod\ p & -a_{12} mod\ p \dots & -a_{1n}\ mod\ p \\ -a_{21}\ mod\ p & -a_{22} mod\ p \dots & -a_{2n}\ mod\ p \\ \vdots & \vdots & \vdots \\ -a_{m1}\ mod\ p & -a_{m2} mod\ p \dots & -a_{mn}\ mod\ p \end{bmatrix}$$

$$\mathrm{A} + \mathrm{B}\ mod\ p = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} = 0$$

6) **Multiplicative Inverse.** Multiplicative inverse of a residue matrix X is a matrix Y in Z_p such that $\mathrm{X} \times \mathrm{Y} = \mathrm{Y} \times \mathrm{X} = \mathrm{I}\ \mathrm{mod}\ \mathrm{p}$. Generally, the multiplicative inverse of X is characterized by as X^{-1}. The result of X^{-1} in Z_p is calculated in the following procedure [77].

$$\mathrm{X} = \begin{bmatrix} 1 & 1 & 2 \\ 3 & 1 & 1 \\ 1 & 3 & 4 \end{bmatrix} \text{ in } Z_5$$

$$\det \mathrm{X} = (1)\begin{vmatrix} 1 & 1 \\ 3 & 4 \end{vmatrix} - (1)\begin{vmatrix} 3 & 1 \\ 1 & 4 \end{vmatrix} + 2\begin{vmatrix} 3 & 1 \\ 1 & 3 \end{vmatrix}\ mod$$

$$= (1)\{4-3\} - (1)\{12-(1)\} + 2\{9-(1)\}$$

$$= 1 - 11 + 16 = 6\ m = 1\ \neq 0$$

$$C_{11} = (-1)^{1+1}\begin{vmatrix} 1 & 1 \\ 3 & 4 \end{vmatrix} = 1$$

$$C_{12} = (-1)^{1+2}\begin{vmatrix} 3 & 1 \\ 1 & 4 \end{vmatrix} = -11\ mod\ 5 = 4$$

$$C_{13} = (-1)^{1+3}\begin{vmatrix} 3 & 1 \\ 1 & 3 \end{vmatrix} = 8\ mod\ 5 = 3$$

$$C_{21} = (-1)^{2+1}\begin{vmatrix}1 & 2\\ 3 & 4\end{vmatrix} = 2$$

$$C_{22} = (-1)^{2+2}\begin{vmatrix}1 & 2\\ 1 & 4\end{vmatrix} = 2,$$

$$C_{23} = (-1)^{2+3}\begin{vmatrix}1 & 1\\ 1 & 3\end{vmatrix} = -2\ mod\ 5 = 3$$

$$C_{31} = (-1)^{3+1}\begin{vmatrix}1 & 2\\ 1 & 1\end{vmatrix} = -1\ mod\ 5 = 4$$

$$C_{32} = (-1)^{3+2}\begin{vmatrix}1 & 2\\ 3 & 1\end{vmatrix} = 5\ mod\ 5 = 0$$

$$C_{33} = (-1)^{3+3}\begin{vmatrix}1 & 1\\ 3 & 1\end{vmatrix} = -2\ mod\ 5 = 3$$

$$\mathrm{X}^{-1} = \frac{1}{\det \mathrm{X}}\begin{bmatrix}C_{11} & C_{21} & C_{31}\\ C_{12} & C_{22} & C_{32}\\ C_{13} & C_{23} & C_{33}\end{bmatrix} = \frac{1}{1}\begin{bmatrix}1 & 2 & 4\\ 4 & 2 & 0\\ 3 & 3 & 3\end{bmatrix} = \begin{bmatrix}1 & 2 & 4\\ 4 & 2 & 0\\ 3 & 3 & 3\end{bmatrix}$$

$$\mathrm{X} \times \mathrm{X}^{-1} = \begin{bmatrix}1 & 1 & 2\\ 3 & 1 & 1\\ 1 & 3 & 4\end{bmatrix} \times \begin{bmatrix}1 & 2 & 4\\ 4 & 2 & 0\\ 3 & 3 & 3\end{bmatrix} \text{ in } Z_5$$

$$= \begin{bmatrix}1(1)+1(4)+2(3) & 1(2)+1(2)+2(3) & 1(4)+1(0)+2(3)\\ 3(1)+1(4)+1(3) & 3(2)+1(2)+1(3) & 3(4)+1(0)+1(3)\\ 1(1)++3(4)+4(3) & 1(2)+3(2)+4(3) & 1(4)+3(0)+4(4)\end{bmatrix} \bmod 5$$

$$= \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix} = \mathrm{I}$$

$$\mathrm{X}^{-1} \times \mathrm{X} = \begin{bmatrix}1 & 2 & 4\\ 4 & 2 & 0\\ 3 & 3 & 3\end{bmatrix}\begin{bmatrix}1 & 1 & 2\\ 3 & 1 & 1\\ 1 & 3 & 4\end{bmatrix} \text{ in } Z_5$$

$$= \begin{bmatrix}1(1)+2(3)+4(1) & 1(1)+2(1)+4(3) & 1(2)+2(1)+4(4)\\ 4(1)+2(3)+0(1) & 4(1)+2(1)+0(3) & 4(2)+2(1)+0(4)\\ 3(1)++3(3)+3(1) & 3(1)+3(1)+3(3) & 3(2)+3(1)+3(4)\end{bmatrix} mod\ 5$$

$$= \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix} = \mathrm{I}$$

5.5 Summary

In cryptography, the residue matrices based on the modular arithmetic are normally used in cryptographic non-linear mathematical transformations. Hill cipher was a well-known substitution cipher based on residue matrix transformations in cryptosystems of past years. It operates using matrix multiplication and inversion for encryption and decryption. It is vulnerable to the known-plaintext attack. Nowadays, several researchers develop its secure variants to improve the security of Hill cipher.

CHAPTER 6
ELLIPTIC CURVE ARITHMETIC

This chapter aims to provide a detailed implementation for elliptic curve arithmetic operations over prime field and binary field. This work supports to implement, analyze and study any elliptic curve cryptosystems over prime field and binary field under large integers. The organization of this chapter is as follows. The section 6.1 describes the basic concepts of elliptic curve arithmetic. The section 6.2 includes finite field arithmetic operations over prime field and its properties. The section 6.3 describes in details elliptic curve arithmetic operations over binary field and its geometric properties. The section 6.4 illustrates scalar multiplication of a point and its implementation. Finally, this chapter is concluded by discussion about its security and performance in section 6.5.

6.1 Elliptic Curve

The elliptic curve over finite field $E(GF)$ is a cubic curve defined by the general Weierstrass equation (6.1) over GF *where* $a_i \in GF$ and *GF* is a finite field.

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (6.1)$$

Elliptic curves are driven from the general Weierstrass equation (6.1). The elliptic curve $E\big(GF(\mathrm{p})\big)$ is determined by the equation (6.2).

$$y^2 = x^3 + ax + b \quad (6.2)$$

Where $p > 3$ is a prime and $a, b \in GF(p)$ satisfy that $4a^3 + 27b^2 \neq 0$. $(a_1 = a_2 = a_3 = 0,\ a_4 = a\ and\ a_6 = b\ )$ corresponding to the general Weierstrass equation (6.1) [6, 8, 14, 27, 55, 60].

Elements over $GF(2^m)$ must be firstly generated by using a reduction polynomial $f(x)$. These elements are applied to construct an elliptic curve $E\big(GF(2^m)\big)$ over $GF(2^m)$. The elliptic curve $E\big(GF(2^m)\big)$ is determined by the equation (6.3) [6, 8, 14, 27, 55, 60]

$$y^2 + xy = x^3 + ax^2 + b \quad (6.3)$$

Where $a, b \in GF(2^m)\ and\ b \neq 0$.

The addition of two points on an elliptic curve uses the *chord-and-tangent rule* that results a third point on the curve. The addition operations with the points on an

elliptic curve generate a group with point at infinity *O* serving as its identity. It is the cyclic group of points on an elliptic curve that is used in the construction of elliptic curve cryptosystems [27, 43, 55, 94,102]. According to Hasse's Theorem [114], the cyclic group order on the elliptic curves over finite field *GF* may be between $q+1-2\sqrt{q}$ and $q+1+2\sqrt{q}$, where $q = p \text{ or } 2^m$. It is the best way to explain the point addition rule geometrically. Let $P = (x_1, y_1)$ *and* $Q = (x_2, y_2)$ be two distinct points on an elliptic curve. Assume that the point $R = (x_3, y_3)$ is obtained by *addition* of P *and* Q. This point addition is illustrated in Figure 6.1. The line connecting through P *and* Q intersects the elliptic curve at the point called $-R$. R is the reflection of $-R$ with respect to the x-axis. Assume that doubling of P is $R = (x_3, y_3)$ in the case of $P = (x_1, y_1)$. This point doubling is illustrated in Figure 6.2. The tangent line drawing from point P intersects the elliptic curve at the point called $-R$. R is the reflection of $-R$ with respect to the *x*-axis as in the case of addition [102].

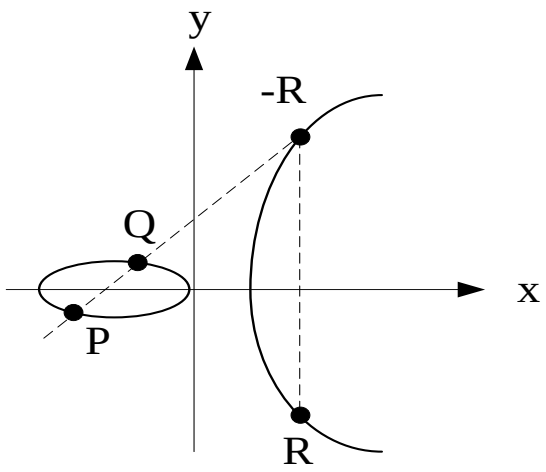

Figure 6.1: Addition (R = P+Q)

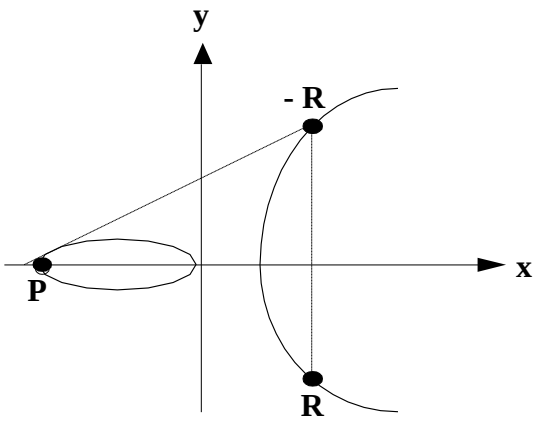

Figure 6 2: Doubling (R = P + P)

6.2 Elliptic Curve Arithmetic Over Prime Field $E(GF(P))$

The following are algebraic methods for the addition of two distinct points on $E(GF(P))$ and the doubling of a point on $E(GF(P))$ [111].

1) **Addition of points.**

Let $P, Q \in E(GF(P)), P = (x_1, y_1)\ and\ Q = (x_2, y_2)\ \ where\ P \neq Q.$

Then $P + Q = (x_3, y_3).$

In this case, $x_3 = \lambda^2 - x_1 - x_2$ and $y_3 = \lambda(x_1 - x_3) - y_1$

where $\lambda = (y_2 - y_1)/(x_2 - x_1)$ [6, 8, 14, 102].

2) **Doubling of a point**.

Let $P = (x_1, y_1) \in E(GF(P))\ \ where\ P \neq -P.$

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

In this case, $x_3 = \lambda^2 - 2x_1\ \ and\ \ y_3 = \lambda(x_1 - x_3) - y_1$

where $\lambda = (3{x_1}^2 + a)/(2y_1)$ [6, 8, 14, 102].

The elliptic curve *E(GF(p))* consists of a set of points $\{P = (x, y) |\ y^2 = x^3 + ax + b, x, y, a, b \in GF(p)\}$ together with a point at infinity defined as *O*. The point at infinity serves as additive identity. Every point on the elliptic curve has its *inverse*. The inverse of a point (*x*, *y*) on *E(GF(p))* is (*x*, *-y*). The point $(x, -y)$ is signified by $(-P).\ P + O = O + P = P\ and\ P + (-P) = O$ for all $P \in E(GF(P))$. The number of points on the curve, including a point at infinity, is called its *order #E* [6, 8, 14, 102]. The pseudocode for finding the points on the elliptic curve *E(GF(p))* is shown in Algorithm 6.1. The elliptic curve arithmetic operations such as addition of points and doubling of a point on $E(GF(P))$ are described in Example 6.1.

Algorithm (6.1). Pseudocode for finding the points on the elliptic curve *E(GF(p))*

Input: a, b, p

Output: $P_i = (x_i, y_i)$

Begin

x = 0;

while (x < p) {

$w = (x^3 + ax + b) mod\ p.$

If (w is perfect square in Z_p*) output* $(x, \sqrt{w})\ (x, -\sqrt{w})$

x = x + 1.

}

End

Example (6.1). Let p = 13 and consider the elliptic curve E: $y^2 = x^3 + x + 1$ defined over $GF(P)$ where $a = 1$ and $b = 1$. Note that $4a^3 + 27b^2 = 4 + 27 = 31\ mod\ 13 = 5$, so E is indeed an elliptic curve. The points on $E\big(GF(P)\big)$ are shown in Figure 6.3 and its graph is illustrated in Figure 6.4. The order of the elliptic curve E: $y^2 = x^3 + x + 1$ over $GF(13)$ is 18.

$(0,1)$	$(0,12)$
$(1,4)$	$(1,9)$
$(4,2)$	$(4,11)$
$(5,1)$	$(5,12)$
$(7,0)$	O
$(8,1)$	$(8,12)$
$(10,6)$	$(10,7)$
$(11,2)$	$(11,11)$
$(12,5)$	$(12,8)$

Figure 6.3: Points on E: $y^2 = x^3 + x + 1$ $GF(13)$

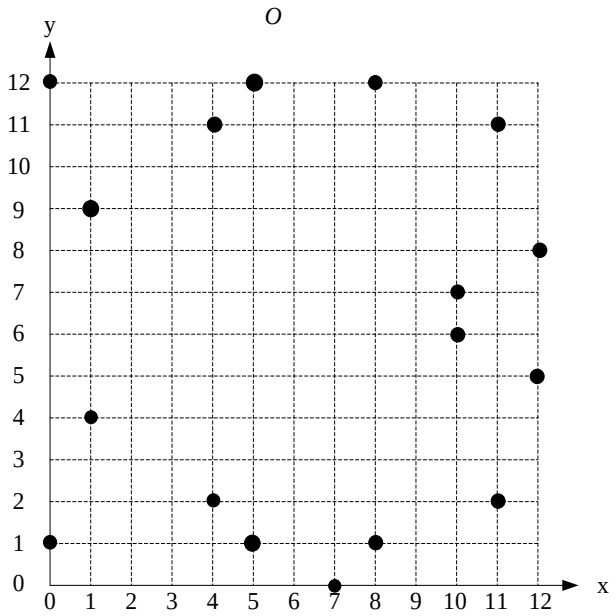

Figure 6.4: Graph Illustrated for Points on E: $y^2 = x^3 + x + 1$ $GF(13)$

Addition of Points: Let $P = (4,2)$ and $Q = (10,6)$.

$x_1 = 4, y_1 = 2, x_2 = 10, y_2 = 6.$

$$\lambda = \frac{(y_2 - y_1)}{(x_2 - x_1)} = \frac{6-2}{10-4} = (6-2)(10-4)^{-1}\ mod\ 13 = 4 \times 6^{-1}\ mod\ 13$$

The Multiplicative Inverse of 6 in Z_{13} is 11.

$$= 4 \times 11\ mod\ 13 = 5\ mod\ 13$$

$$x_3 = \lambda^2 - x_1 - x_2 = (5)^2 - 4 - 10\ mod\ 13 = 11\ mod\ 13.$$

$$y_3 = \lambda(x_1 - x_3) - y_1 = 5(4-11) - 2\ mod\ 13 = 2\ mod\ 13.$$

Then $P + Q = (x_3, y_3) = (11,2)$.

Doubling of a Point: Let P $= (4,2)$. $x_1 = 4, y_1 = 2$

$$\lambda = \frac{(3(x_1)^2 + a)}{(2y_1)} = \frac{3(4)^2 + 1}{2 \times 2} = (49)(4)^{-1}\ mod\ 13$$

$$= 49 \times 10\ mod\ 13$$

$$= 9\ mod\ 13.$$

$x_3 = \lambda^2 - 2x_1 = (9)^2 - 2(4)\ mod\ 13 = (81 - 8) mod\ 13 = 73\ mod\ 13$

$= 8\ mod\ 13.$

$y_3 = \lambda(x_1 - x_3) - y_1 = 9(4 - 8) - 2\ mod\ 13 = -38\ mod\ 13 = 1\ mod\ 13.$

Then $2P = (8, 1)$.

Inverse: Let P $= (1, 6)$. Then $-P = (1, 7)$.

6.3 Elliptic Curve Arithmetic Over Binary Field $E\big(GF(2^m)\big)$

The followings are algebraic methods for the addition of two distinct points on $E\big(GF(2^m)\big)$ and the doubling of a point on $E\big(GF(2^m)\big)$ [95].

1) Addition of points.

Let $P, Q \in E\big(GF(2^m)\big),\ P = (x_1, y_1)\ and\ Q = (x_2, y_2)$ where $P \neq Q$.

Then $P + Q = (x_3, y_3)$.

In this case, $x_3 = \lambda^2 + \lambda + x_1 + x_2 + a\ and\ y_3 = \lambda(x_1 + x_3) + x_3 + y_1$

where $\lambda = (y_2 + y_1)/(x_2 + x_1)$ [6, 8, 14, 102].

2) Doubling of a point.

Let $P = (x_1, y_1) \in E\big(GF(2^m)\big)$ where $P \neq -P$.

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

In this case, $x_3 = \lambda^2 + \lambda + a\ \ and\ \ y_3 = {x_1}^2 + \lambda x_3 + x_3$

where $\lambda = x_1 + (y_1/x_1)$ [6, 8, 14, 102].

The elliptic curve *E(GF(2^m))* consists of a set of points: $\{P = (x, y) | y^2 + xy = x^3 + ax + b,\ x, y, a, b \in GF(2^m)\}$ together with a point at infinity defined as *O*. The point at infinity serves as additive identity [34]. Every point on the elliptic curve has its *inverse*. The inverse of a point (*x*, *y*) on E(GF(2^m)) is $(x, x \oplus y)$. The point $(x, x \oplus y)$ is signified by $(-P)$. $P + O = O + P = P\ \ and\ \ P + (-P) = 0\ \ for\ all\ \ P \in E\big(GF(2^m)\big)$. The number of points on the curve, including a point at infinity, is called its *order #E* [6, 8, 14, 102]. The pseudocode for finding the points on the elliptic curve *E(GF(2^m))* is shown in Algorithm 6.2 [102]. The elliptic curve arithmetic operations such as addition of points and doubling of a point on $E\big(GF(2^m)\big)$ are described in Example 6.2.

Algorithm (6.2). Pseudocode for finding the points on the elliptic curve *E(GF(2^m))*

Input: a, b, $f(x)$

Output: $P_i = (x_i, y_i)$

Begin

$x_i = \{0, 1, g^1, \dots, g^{m-2}\}$

$y_j = \{0, 1, g^1, \dots, g^{m-2}\}$

for (i=0; i<2^m; i++) {

for (j=0; j < 2^m; j++) {

$w_1 = {x_i}^3 \oplus ax_i \oplus b.$

$w_2 = {y_j}^2 \oplus x_i y_j$

If ($w_1 = w_2$*) output* (x_i, y_j) $(x_i, y_j \oplus x_i)$

}

}

End

Example (6.2). Let $f(x) = x^4 + x + 1$ be the reduction polynomial. Then 16 elements of *GF (2^4)* are shown in Table (6.1).

Table 6.1: Elements of $GF(2^4)$

0000	0	1000	x^3
0001	1	1001	$x^3 + 1$
0010	x	1010	$x^3 + x$
0011	$x + 1$	1011	$x^3 + x + 1$
0100	x^2	1100	$x^3 + x^2$
0101	$x^2 + 1$	1101	$x^3 + x^2 + 1$
0110	$x^2 + x$	1110	$x^3 + x^2 + x$
0111	$x^2 + x + 1$	1111	$x^3 + x^2 + x + 1$

Table (6.2) shows the power representation of g for elements of $GF(2^4)$ generated by the polynomial $f(x) = x^4 + x + 1$. The element of $g = x = (0010)$ is a generator of $GF(2^4)$ because its order is 15 $(2^4 - 1)$.

Table 6.2: Power Representation of Elements

g	0010	g^5	0110	g^9	1010	g^{13}	1101
g^2	0100	g^6	1100	g^{10}	0111	g^{14}	1001
g^3	1000	g^7	1011	g^{11}	1110	g^{15}	0001
g^4	0011	g^8	0101	g^{12}	1111		

Using the elliptic curve $E: y^2 + xy = x^3 + gx + 1$, with $a = g$ and $b = 1$, the points on the curve are shown in Figure 6.5 and its graph is illustrated in Figure 6.6. The *order* of the elliptic curve $E: y^2 + xy = x^3 + gx + 1$ over $GF(2^4)$ is 16.

$(0, 1)$	O
$(1, g^7)$	$(1, g^9)$
$(g^3, 0)$	(g^3, g^3)
(g^5, g^{12})	(g^5, g^{14})
(g^6, g^7)	(g^6, g^{10})
(g^9, g^2)	(g^9, g^{11})
(g^{10}, g^2)	(g^{10}, g^4)
(g^{12}, g^2)	(g^{12}, g^7)

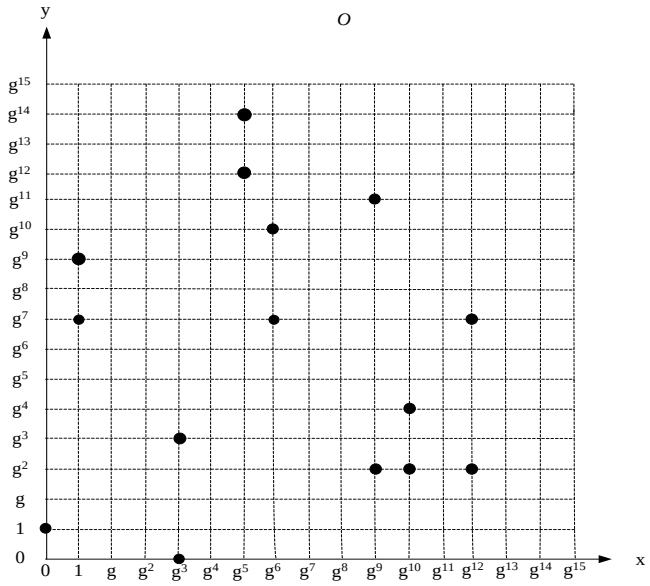

Figure 6.5: Points on $E: y^2 + xy = x^3 + gx + 1$ $GF(2^4)$

Figure 6.6: Graph Illustrated for Points on $E: y^2 + xy = x^3 + gx + 1$ $GF(2^4)$

Addition of Points: Let P $= (g^5, g^{12})$ and $Q = (g^6, g^7)$.

$x_1 = g^5, y_1 = g^{12}, x_2 = g^6, y_2 = g^7.$

$$\lambda = \frac{(y_2 + y_1)}{(x_2 + x_1)} = \frac{g^7 + g^{12}}{g^6 + g^5} = \frac{g^2}{g^9} = g^2 g^{-9} \bmod f(x) = g^2 \times g^6 \bmod f(x)$$

$$= g^8 mod\ f(x) = (g^2 + 1) \bmod f(x)$$

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a$$

$$= (g^2 + 1)^2 + g^2 + 1 + g^5 + g^6 + g = g^4 + 2g^2 + 1 + g^2 + 1 + g^5 + g^6$$

$$= g^6 + g^5 + g^4 + g^2 + g = g^{12}$$

$$y_3 = \lambda(x_1 + x_3) + x_3 + y_1$$

$$= (g^2 + 1)(g^5 + g^{12}) + g^{12} + g^{12} = g^7 + g^{14} + g^5 + g^{12} = g^7$$

Then $P + Q = (x_3, y_3) = (g^{12}, g^7)$

Doubling of a Point: Let P $= (g^5, g^{12})$. $x_1 = g^5, \quad y_1 = g^{12}, a = g$

$\lambda = x_1 + (y_1/x_1)$.

$= g^5 + (g^{12}/g^5) = g^5 + g^{12}g^{-5} = g^5 + g^{12}g^{10} = g^5 + g^7 = g^{13} = g^6 + 1$

$x_3 = \lambda^2 + \lambda + a$

$= (g^6 + 1)^2 + g^6 + 1 + g = g^{12} + 2g^6 + 1 + g^6 + 1 + g = g^{12} + g^6 + g = 1$

$y_3 = {x_1}^2 + \lambda x_3 + x_3$

$= (g^5)^2 + (g^6 + 1)1 + 1 = g^{10} + g^6 = g^7$

Then $2P = (x_3, y_3) = (1,\ g^7)$.

Inverse: Let P $= (g^5, g^{12})$. Then $-P = (g^5, g^{14})$.

6.4 Point Multiplication

The complexity of solving Elliptic Curve Discrete Logarithm Problem (ECDLP) determines the security of ECC [61, 68]. Let *P* and *Q* be the points on an elliptic curve such that $Q = kP$ where *k* is an integer number. *k* is called the discrete logarithm of *Q* to the base *P* [23]. Known two points, *P* and *Q*, it is unable to compute *k*, when the group order of the points is enough large [102].

Point Multiplication is a major operation usually used in ECC. The scalar multiplication operation of an integer scalar *k with* a point *P* on the elliptic curve creates another point *Q* on this curve [35]. The point *Q* is gotten by performing *point addition and point doubling* operations according to bit sequence patterns of integer scalar *k*. The bit sequence patterns of integer *k* are shown as the equation (6.4) [74, 102].

$$k = k_{n-1}2^{n-1} + k_{n-2}2^{n-2} + \cdots + k_1 2 + k_0 \tag{6.4}$$

Where $k_{n-1} = 1$ and $k_i \in \{0, 1\},\ i = 0, 1, 2, \ldots, n-1$.This operation is based on the *binary method* which scans the bit sequence patterns of *k* either from left-to-right or right-to-left. The Algorithm 6.3 illustrates the scalar multiplication operation of a integer scalar *k* with a point *P* on the elliptic curve using binary method [74, 102]. This method can be applied for both elliptic curves over $GF(P)$ and $GF(2^m)$ [30].

Algorithm (6.3). Scalar Multiplication of a Point

Input: point P and integer scalar k

Output: point Q such that $Q = kP$

Begin

$k_i \in \{0, 1\}, i = 0, 1, 2, \dots, n-1$

$Q = P$

$For\ i = n-1\ to\ 0\ do$

{

$Q = Point - Doubling\ of\ Q\ If\ k_i = 1\ then$

$Q = Point - Addition\ of\ P\ and\ Q$

}

$Return\ Q$

End

6.5 Summary

Elliptic Curve Arithmetic was applied on cryptography known as of Elliptic Curve Cryptography (ECC) was discovered in 1985 by Victor Miller and Neil Koblitz (University of Washington) as an alternative mechanism for implementing public-key cryptography (PKC). ECC is a kind of public key cryptography. The use of ECC allows the increasing of security. In the same time, ECC decreases the overloading. ECC security consists in the difficulty to calculate logarithms in discrete fields (discrete logarithms problem): being given *A* (an element from a finite field) and A^x, it is practically impossible to calculate *x* when *A* is big enough. An ECC with 160-bit key offers a security level equivalent with that offered by a cryptosystem based on a 1024-bit *GF(p)* field. Because of this, ECC provides a feasible method of implementation for a high-level security system on a PC card, on an intelligent card or on a mobile communications device.

CHAPTER 7
IMPLEMENTATION

This is a core chapter that describes the implementations, the analyses, and the experimental results for the research work. The purpose of the chapter is to explain how to implement the mathematical methods that need to perform the research work, to analyse the mathematical properties by using the implemented methods that are integrating with complex numbers and to describe and discuss the experimental results. The organization of this chapter is as follows. Section 7.2 describes the step-by-step implementation procedures for computing matrix algebra and elliptic curve arithmetic in complex field. Section 7.3 includes the study on the mathematical characteristics of finite field integrating with complex numbers. Section 7.4 contains the analysis on the mathematical characteristics of residue matrices integrating with complex numbers. Section 7.5 includes the analysis on the mathematical characteristics of elliptic curve arithmetic integrating with complex numbers. The sections 7.6 and 7.7 show the experimental results on the curve order and on the cyclic group which are generated from the elliptic curves integrating with complex numbers. The computational costs on the arithmetic operations with real numbers, complex numbers and elliptic curves with real numbers and complex numbers are studied in section 7.8. Section 7.9 summarizes the chapter with discussions on the research work.

7.1 Design of the Proposed Scheme

In this research, there are two main parts. They are generating and comparing point orders using prime field and binary field with integer and complex number, respectively. The purpose is to proof that point order generated with complex number is greater than one with integer in both prime field and binary field. According to experimental result, point order generated with complex number is greater than one with integer in both prime field and binary field. Elective curve implemented with complex number has more time complexity than one with integer. Therefore, it can be shown that the proposed scheme is more secure than other one. Figure 7.1 and 7.2 shows flowcharts of the proposed scheme in terms of using prime field and binary field, respectively.

7.1.1 Prime Field with Integer and Complex Number

In the proposed scheme, the workflow is outlined through a series of sequential steps aimed at generating and comparing points derived from prime numbers and elliptic curves.

Step 1: Start with a prime number and an elliptic curve, which will be the main parameters for the next steps.

Step 2A: Generate points using the prime field with integers.

Step 2B: Generate points using the prime field with complex numbers.

Step 3: Compare the points generated from the integer and complex number fields.

Step 4: Output the comparison results, showing that the point generated with complex numbers is greater than the point generated with integers.

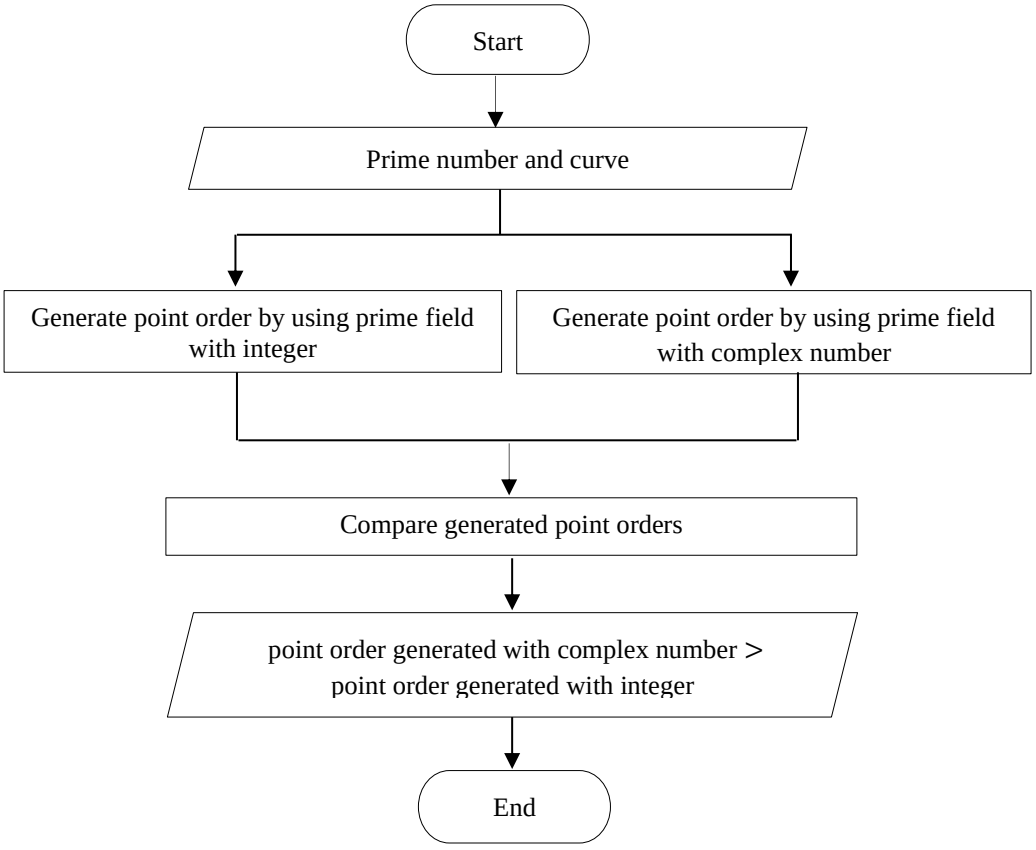

Figure 7. 1: Flowchart of the Proposed Scheme using Prime Field

7.1.2 Binary Field with Integer and Complex Number

In the proposed scheme, the flowchart shows a systematic approach to processing input functions and elective curves, aiming to generate and evaluate points derived from binary fields with integers and complex numbers.

Step 1 initiates the process by acquiring the input function (f(x)) and elective curve parameters, which serve as fundamental components for subsequent computations.

In Step 2A, points are systematically generated utilizing the binary field with integers, while in parallel, Step 2B undertakes a similar process utilizing the binary field with complex numbers. These distinct methodologies allow for a comprehensive exploration of computational strategies within the system.

Following point generation, Step 3 makes comparison of the points derived from the integer and complex number binary fields.

Lastly, in Step 4, the system outputs the comparison results, affirming that the point generated with complex numbers exceeds the point generated with integers.

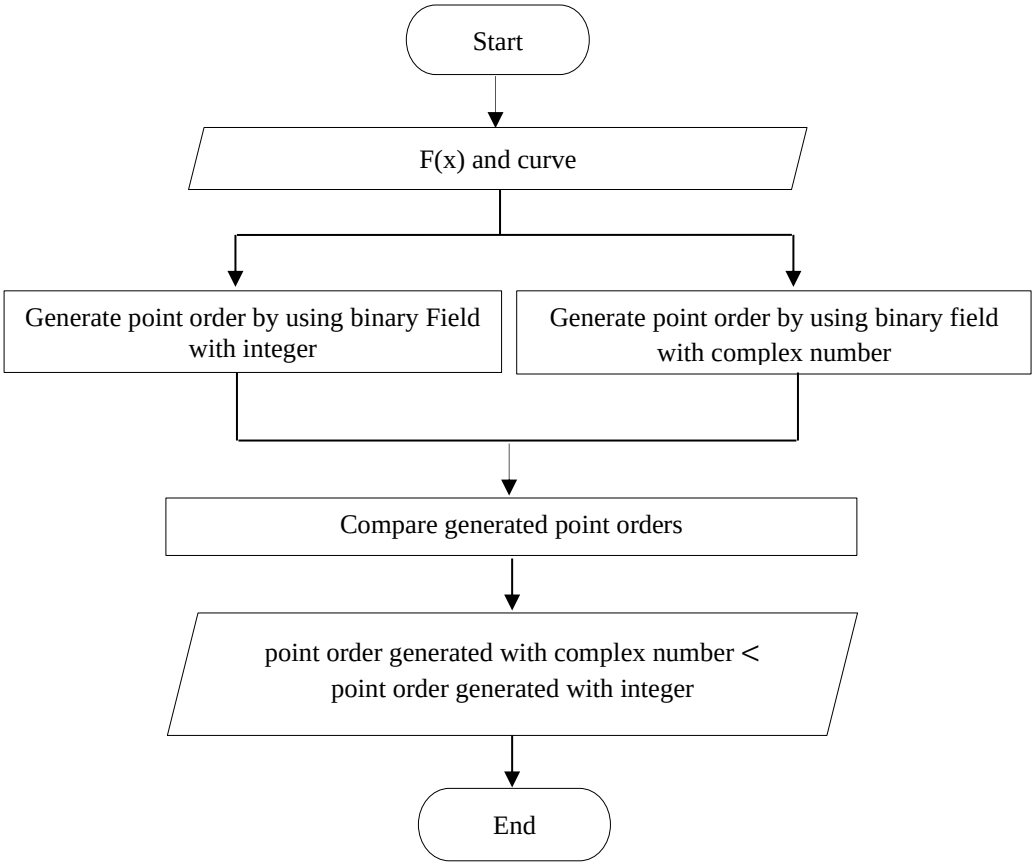

Figure 7.2: Flowchart of the Proposed Scheme using Binary Field

7.2 Implementation

The mathematical characteristics of matrix algebra and elliptic curve arithmetic over complex fields need to be studied in this research work. The results show the methods for arithmetic operations of finite fields are firstly implemented by using java BigInteger class. In second step, it implements the methods that need to use the arithmetic operations of finite fields integrating with complex numbers. Then not only the methods for arithmetic operations of matrix algebra are implemented to analyse the arithmetic properties of reduce matrices over complex field but also the methods for arithmetic operations of elliptic curve arithmetic are implemented to analyse the arithmetic properties of elliptic curves over complex fields.

7.2.1 Implementation for Computing Matrix Algebra in Complex Field

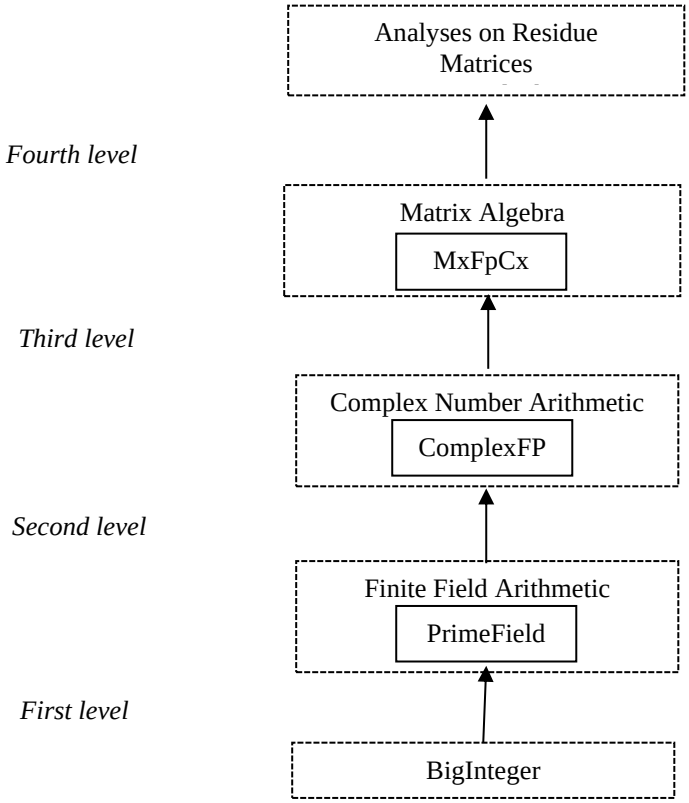

Figure 7. 3: Implementation Logic Design for Residue Matrices [77, 78, 84, 99]

At first level, the PrimeField class including methods for addition, subtraction, multiplication, division, additive inverse and multiplicative inverse, finite field arithmetic operations of $GF(p)$ is implemented by using methods of java BigInteger class. At second level, the ComplexFp class including methods for addition, subtraction, multiplication, division, additive inverse and multiplicative inverse,

complex arithmetic operations of $Z(GF(p))$, complex field based on $GF(p)$, is implemented by using methods of PrimeField class. At third level, the MxFpCx class including methods for addition of two residue matrices, subtraction of two residue matrices, multiplication of two residue matrices and inversion of a residue matrix, residue matrices arithmetic operations based on complex field $Z(GF(p))$, is implemented by using methods of ComplexFp class. At fourth level, analyses on residue matrices in complex fields are performed by using corresponding methods of MxFpCx class. For the implementation logic design of residue matrices in complex fields, the general hierarchy is shown in Figure 7.3.

7.2.2 Implementation for Computing Elliptic Curve Arithmetic in Complex Field

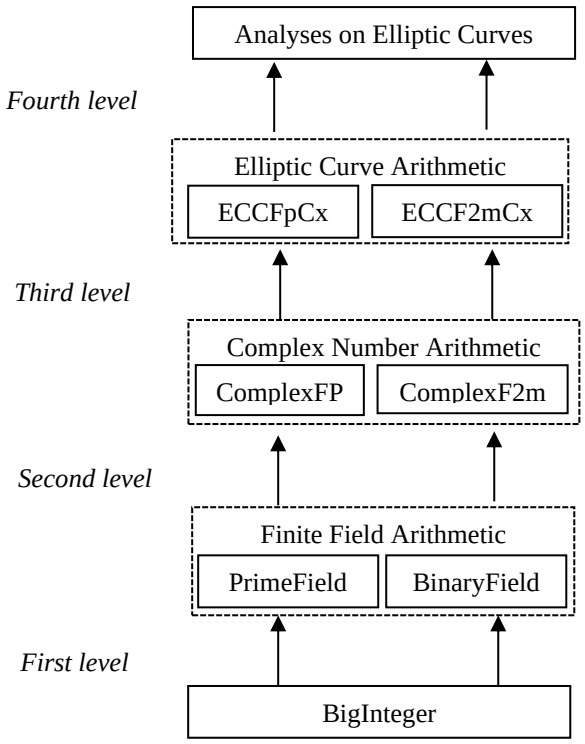

Figure 7.4: Implementation Logic Design for Elliptic Curves [78]

At first level, the PrimeField class including methods for addition, subtraction, multiplication, division, additive inverse and multiplicative inverse, finite field arithmetic operations of $GF(P)$ is implemented by using methods of java BigInteger class. Similarly, the BinaryField class including methods for addition, subtraction, multiplication, division, additive inverse and multiplicative inverse, finite field arithmetic operations of $GF(2^m)$ is implemented by using java BigInteger class. At second level, the ComplexFp class including methods for addition, subtraction,

multiplication, division, additive inverse and multiplicative inverse, complex arithmetic operations of $Z(GF(P))$, complex field based on $GF(p)$, is implemented by using methods of PrimeField class. Similarly, the ComplexF2m class including methods for addition, subtraction, multiplication, division, additive inverse and multiplicative inverse, complex arithmetic operations of $Z(GF(2^m))$, complex field based on $GF(2^m)$, is implemented by using methods of BinaryField class. At third level, the ECCFpCx class including methods for point addition, point doubling and point multiplication, elliptic curve arithmetic operations of $E(Z(GF(P)))$, the elliptic curve based on complex field $Z(GF(P))$, is implemented by using methods of ComplexFp class. Similarly, the ECCF2mCx class including methods for point addition, point doubling and point multiplication, elliptic curve arithmetic operations of $E(Z(GF(2^m)))$, the elliptic curve based on complex field $Z(GF(2^m))$, is implemented by using methods of ComplexF2m. At fourth level, analyses on elliptic curves in complex fields are performed by using corresponding methods of ECCFpCx class and ECCF2mCx class. For the implementation logic design of elliptic curves in complex fields, the general hierarchy is shown in Figure 7.4.

7.3 Analysis on Elements in Complex Field

The following is the study on the fundamental arithmetic properties of elements in complex fields with compared to finite fields.

7.3.1 Arithmetic Properties of Finite Field

There are a set of elements in finite fields, and such elements satisfy the following mathematical characteristics [31, 84, 92, 109].

- The Law of Commutativity: $x + y = y + x; \quad x.y = y.x,$ for all $x, y, \in F$.
- The Law of Associativity: $(x + y) + z = x + (y + z); \ (x.y)z = x(y.z)$ for all $x, y, z \in F$.
- The Law of Distributivity: $(x + y)z = xz + yz$, for all $x, y, z \in F$
- The Law of Identity: Zero, denoted by 0, is the additive identity so that $z + 0 = z$ for all $z \in F$. Besides, one, denoted by 1, is the multiplicative identity so that $z.1 = z$ for all $z \in F$.

- The Law of Additive Inverse. For any $z \in F$, there exists a unique additive inverse $-z \in F$ so that $z + (-z) = 0$.
- The Law of Multiplicative Inverse. For any $z \in F$ where $z \neq 0$, there exists a unique multiplicative inverse $z^{-1} \in F$, so that $zz^{-1} = 1$.

7.3.2 Experiments on Elements in Complex Field $Z(GF(P))$

Suppose that two complex numbers, $x = 1 + 2i,\ y = 2 + 1i\ and\ z = 3 + 2i$ are belongs to complex field $Z(GF(7))$, that is, $x, y, z\ \in Z(GF(7))$. The arithmetic of these complex numbers is computed by using our developed system. The followings are experiments on arithmetic operations of complex numbers in $Z(GF(p))$. The experimental results prove that the mathematical characteristics of complex field $Z(GF(p))$ are consistent with those of finite field $GF(p)$ [99].

- $x + y = (1 + 2i) + (2 + 1i) = s = 3 + 3i$
- $s - y = (3 + 3i) - (2 + 1i) = x = 1 + 2i$
- $s - x = (3 + 3i) - (1 + 2i) = y = 2 + 1i$
- $x \times y = (1 + 2i) \times (2 + 1i) = s = 5i$
- $s/y = (5i)/(2 + 1i) = x = 1 + 2i$
- $s/x = (5i)/(1 + 2i) = y = 2 + 1i$
- $(x + y) + z = x + (y + z) = 6 + 5i$
- $z(x + y) = zx + zy = 3 + 1i$
- Additive inverse of $x = (-x) = 6 + 5i$ and $x + (-x) = 0$.
- Multiplicative inverse of $y = y^{-1} = 6 + 4i$ and $y.y^{-1} = 1$.

7.3.3 Experiments on Elements in Complex Field $Z(GF(2^m))$

Suppose that two complex numbers, $x = 1 + 2i, y = 2 + 1i\ and\ z = 3 + 2i$ are belongs to complex field $Z(GF(2^m))$ with elements generated by the reduction polynomial $f(x) = x^3 + x + 1$, that is, $x, y, z\ \in Z\left(GF(f(x))\right)$. The arithmetic of these complex numbers is computed by using our developed system. The followings are experiments on arithmetic operations of complex numbers in $Z(GF(2^m))$. The experimental results prove that the mathematical characteristics of complex field $Z(GF(2^m))$ are consistent with those of finite field $GF(2^m)$ [99].

- $x + y = (1 + 2i) + (2 + 1i) = s = 3 + 3i$

- $s - y = (3 + 3i) - (2 + 1i) = x = 1 + 2i$
- $s - x = (3 + 3i) - (1 + 2i) = y = 2 + 1i$
- $x.y = (1 + 2i) \times (2 + 1i) = s = 5i$
- $s/y = (5i)/(2 + 1i) = x = 1 + 2i$
- $s/x = (5i)/(1 + 2i) = y = 2 + 1i$
- $(x + y) + z = x + (y + z) = i$
- $z(x + y) = zx + zy = 3 + 3i$
- Additive inverse of $x = (-x) = 1 + 2i$ and $x + (-x) = 0$.
- Multiplicative inverse of $y = y^{-1} = 4 + 2i$ and $y.y^{-1} = 1$.

7.4 Analysis on Residue Matrices in Complex Field

The following is the study on the fundamental arithmetic properties of residue matrices in complex fields with compared to finite fields.

7.4.1 Arithmetic Properties of Residue Matrices

The arithmetic matrix operations of residue matrices satisfy the following properties [10, 77].

- The Commutative Law: $x + y = y + x$, $x.y = y.x$ for all $x, y \in F$.
- The Associative Law: $(x + y) + z = x + (y + z)$; $(x \times y) \times z = x \times (y \times z)$ for all $x, y, z \in F$
- The Distributive Law: $(x + y) \times z = x \times z + y \times z$, for all $x, y, z \in F$.
- The Identities Law: The digit zero (0) is the additive identity such that $z + 0 = z$ for all $z \in F$. In addition, the digit one (1), is the multiplicative identity such that $z \times 1 = z$ for all $z \in F$.
- The Additive Inverse Law: For any $z \in F$, there exists a unique additive inverse $-z \in F$ such that $z + (-z) = 0$.
- The Multiplicative Inverse Law: For any $z \in F$ where $z \neq 0$, there exists a unique Multiplicative inverse $z^{-1} \in F$ such that $z \times z^{-1} = 1$.

7.4.2 Experiments on Residue Matrices in Complex Field

The mathematical characteristics of the residue matrices in complex field are studied by using the developed system. X, Y and Z are the three residue matrices in

complex field $Z(GF(11))$ as shown below. The following are experiments on arithmetic operations of residue matrices in $Z\big(GF(p)\big)$ [77].

$$X = \begin{bmatrix} 5+2i & 8+6i \\ 7i & 4+3i \end{bmatrix} \quad Y = \begin{bmatrix} 8+7i & 5+9i \\ 3+1i & 10 \end{bmatrix} \quad Z = \begin{bmatrix} 5+7i & 6+8i \\ 4+2i & 9+6i \end{bmatrix}$$

The addition operation of residue matrices X and Y in complex field $Z(GF(11))$ satisfies the commutative law as shown below.

$$X + Y = Y + X = \begin{bmatrix} 2+9i & 2+4i \\ 3+8i & 3+3i \end{bmatrix}$$

The addition operation of residue matrices X, Y and Z in complex field $Z(GF(11))$ satisfies the associative law as shown below.

$$(X + Y) + Z = X + (Y + Z) = \begin{bmatrix} 7+5i & 8+1i \\ 7+10i & 1+9i \end{bmatrix}$$

The multiplication operation of residue matrices in complex fields does not satisfy the commutative law, the associative law and the distributive law while the multiplication of complex numbers is commutative, associative and distributive.

The scalar multiplication operation of residue matrices X and Y with scalar K in complex field $Z(GF(11))$ satisfies the distributive law as shown below.

$$K = 2 + 6i$$

$$K \times (X + Y) = K \times X + K \times Y = \begin{bmatrix} 5 + 8\text{i} & 2 + 9\text{i} \\ 2 + 1\text{i} & 10 + 2i \end{bmatrix}$$

The experimental results prove that the mathematical characteristics of the residue matrices in complex field $Z\big(GF(p)\big)$ are consistent with those of finite field $GF(p)$.

7.5 Analysis on Elliptic Curve Points in Complex Field

The following is the study on the fundamental arithmetic properties of elliptic curves in complex fields with compared to finite fields.

7.5.1 Arithmetic Properties of Elliptic Curve

The group made from the points on an elliptic curve for cryptographic purpose is an abelian group. Therefore, the point on elliptic curve satisfies the following arithmetic properties [6, 14, 55, 99, 102].

- Closure: adding two points, using the corresponding method for the addition operations defined in the previous sections 5.2 and 5.3, creates another point on the curve. i.e. $P + Q = R \ \ for\ all\ P, Q, R \in E$.

- Associativity: $(P + Q) + R = P + (Q + R)$.
- Commutativity: $P + Q = Q + P$.
- Existence of identity: The additive identity in this case is the zero point, 0. In other words, $P = P + 0 = 0 + P$.
- Existence of inverse: Each point on the curve has an inverse. The inverse of a point is its reflection with respect to the x-axis. In other words, the point $P = (x_1, y_1)$ and $Q = (x_1, -y_1\ )$ are inverses of each other, which means that $P + Q = 0$. Note that the identity element is the inverse of itself.

7.5.2 Experiments on Elliptic Curve Points in Complex Field $\left(Z\left(GF(P)\right)\right)$

The arithmetic properties of the points on the elliptic curve $E: y^2 = x^3 + x + (1 + 5i)$ over $Z\left(GF(7)\right)$ are studied as following by using our developed system. Suppose that the points are P = (5, 3+2i) and Q = (4i, 4+1i) on the curve. The following are experiments on arithmetic operations of the points on the elliptic curve over $Z\left(GF(p)\right)$. The experimental results prove that the mathematical characteristics of the elliptic curve over complex field $Z\left(GF(p)\right)$ are consistent with those of finite field $GF(p)$ [99].

- Closure: $P + Q = R\ \ for\ all\ P, Q, R \in E$.

  P + Q = R = (5+6i, 5),
- Associativity: $(P + Q) + R = P + (Q + R)$.

  P + Q = R = (5+6i, 5), (P + Q) + R = (3 + 3i, 6 + 4i)

  Q + R = (4 + 4i,5 + 6i), P + (Q + R) = (3 + 3i, 6 + 4i)
- Commutativity: $P + Q = Q + P$.

  P + Q = Q + P = R = (5+6i, 5).
- Existence of identity: $P = P + 0 = 0 + P$.
- Existence of inverse: the point $P = (x_1, y_1)$ and $Q = (x_1, -y_1\ )$ are inverses of each other, which means that $P + Q = O$.

  P = (5, 3+2i)

  Q = (5, -3 - 2i)

  P + Q = *O*.

7.5.3 Experiments on Elliptic Curve Points in Complex Field $\left(Z\left(GF(2^m)\right)\right)$

The arithmetic properties of the points on the elliptic curve $E: y^2 + xy = x^3 + x^2 + (1 + 5i)$ over $Z\left(GF\left(f(x)\right)\right)$ where $f(x) = x^3 + x + 1$ are studied as following by using our developed system. Suppose that the points are P = (1, 2+5i) and Q = (5i, 6+1i) on the curve. The following are experiments on arithmetic operations of the points on the elliptic curve over $Z\left(GF(p)\right)$. The experimental results prove that the mathematical characteristics of the elliptic curve over complex field $Z\left(GF(p)\right)$ are consistent with those of finite field $GF(p)$ [99].

- Closure: $P + Q = R \ \ for\ all\ P, Q, R \in E$.

  P + Q = R = (1 + 4i,7 + 5i)

- Associativity: $(P + Q) + R = P + (Q + R)$.

  P + Q = R = (1 + 4i,7 + 5i), (P + Q) + R = (1, 2+5i)

  Q + R = 0, P + (Q + R) = (1, 2+5i)

- Commutativity: $P + Q = Q + P$.

  P + Q = Q + P = R = = (1 + 4i,7 + 5i)

- Existence of identity: $P = P + 0 = 0 + P$.

- Existence of inverse: the point $P = (x_1, y_1)$ and $Q = (x_1, -y_1\ )$ are inverses of each other, which means that $P + Q = O$.

  P = (1, 2+5i), Q = (1, 3 + 5i), P + Q = *O*.

7.6 Analysis on Curve Orders

The following is the study on the curve order properties of elliptic curves in complex fields with compared to finite fields.

7.6.1 Experiments on Curve Order over Prime Field $GF(P)$

The total number of points on the elliptic curve $E: y^2 = x^3 + x + 1$ over $GF(7)$ is 5 and all the points are shown in Table 7.1 [21, 99].

Table 7.1: All Points on $E: y^2 = x^3 + x + 1$ over $GF(7)$

0, 1	0, 6	2, 2	2, 5	*O*

7.6.2 Experiments on Curve Order over Complex Field $Z(GF(P))$

The total number of points on the elliptic curves $E: y^2 = x^3 + x + 1$ over $Z(GF(7))$ is 55 and all the points are shown in Table 7.2 [101].

Table 7.2: All Points on $E: y^2 = x^3 + x + 1$ over $Z(GF(7))$

0, 1	0, 6	1, 2i	1, 5i	2, 2
2, 5	3, 2i	3, 5i	4,1i	4, 6i
5, 3i	5, 4i	6, 1i	6, 6i	1i, 1
1i, 6	1+1i, 4+3i	1+1i, 3+4i	2+1i, 3+2i	2+1i, 4+5i
3+1i, 3+1i	3+1i, 4+6i	4+1i, 3+1i	4+1i,4+6i	2i, 4+1i
2i, 3+6i	1+2i, 3i	1+2i, 4i	4+2i, 5+2i	4+2i, 2+5i
6+2i, 2	6+2i, 5	3+3i, 6+3i	3+3i, 1+4i	3+4i, 1+3i
3+4i, 6+4i	5i, 3+1i	5i, 4+6i	1+5i, 3i	1+5i, 4i
4+5i, 2+2i	4+5i, 5+5i	6+5i, 2	6+5i, 5	6i, 1
6i, 6	1+6i, 3+3i	1+6i, 4+4i	2+6i, 4+2i	2+6i, 3+5i
3+6i, 4+1i	3+6i, 3+6i	4+6i, 4+1i	4+6i, 3+6i	*O*

7.6.3 Experiments on Curve Order over Binary Field $GF(2^m)$

The total number of points on the elliptic curve $E: y^2 + xy = x^3 + x^2 + 1$ over $GF(2^3)$ is 14 and all the points are shown in Table 7.3 [101].

Table 7.3 All Points on $E: y^2 + xy = x^3 + x^2 + 1$over $GF(f(x))$ where $f(x) = x^3 + x + 1$

0, 1	4, 3	3, 3	6, 3	2, 7
4, 7	7, 7	2, 5	6, 5	5, 5
3, 0	7, 0	5, 0	*O*	

7.6.4 Experiments on Curve Order over Complex Field $Z\left(GF(2^m)\right)$

The total number of points on the elliptic curve $E: y^2 + xy = x^3 + x^2 + 1$ over $Z\left(GF(2^3)\right)$ is 105 and all the points are shown in Table 7.4 [99].

Table 7.4 All Points on $E: y^2 + xy = x^3 + x^2 + 1$ over $Z\left(GF(f(x))\right)$ where $f(x) = x^3 + x + 1$

0, 1	0, 1i	0, 3+2i	0, 2+3i	0, 5+4i
0, 4+5i	0, 7+6i	0, 6+7i	2, 5	2, 7
3, 0	3, 3	4, 3	4, 7	5, 0
5, 5	6, 3	6, 5	7, 0	7, 7
2+1i, 1+2i	2+1i, 3+3i	3+1i, 3+4i	3+1i, 5i	4+1i, 1+4i
4+1i, 5+5i	5+1i, 5+6i	5+1i, 7i	6+1i, 1+6i	6+1i, 7+7i
7+1i, 7+2i	7+1i, 3i	2i, 4+1i	2i, 4+3i	1+2i, 7+4i
1+2i, 6+6i	4+2i, 1+4i	4+2i, 5+6i	5+2i, 5+5i	5+2i, 7i
6+2i, 2+5i	6+2i, 4+7i	7+2i, 1+1i	7+2i, 6+3i	3i, 5+5i
3i, 5+6i	1+3i, 1+4i	1+3i, 7i	4+3i, 6+1i	4+3i, 2+2i
5+3i, 6+5i	5+3i, 3+6i	6+3i, 4+4i	6+3i, 2+7i	7+3i, 2+1i
7+3i, 5+2i	4i, 6+1i	4i, 6+5i	1+4i, 2+2i	1+4i, 3+6i
2+4i, 6+3i	2+4i, 4+7i	3+4i, 1+1i	3+4i, 2+5i	6+4i, 7+2i
6+4i, 1+6i	7+4i, 3i	7+4i, 7+7i	5i, 7+2i	5i, 7+7i
1+5i, 3i	1+5i, 1+6i	2+5i, 4+3i	2+5i, 6+6i	3+5i, 4+1i
3+5i, 7+4i	6+5i, 2+1i	6+5i, 4+4i	7+5i, 5+2i	7+5i,2+7i
6i, 2+1i	6i, 2+7i	1+6i, 5+2i	1+6i, 4+4i	2+6i, 1+2i
2+6i, 3+4i	3+6i, 3+3i	3+6i, 5i	4+6i, 6+3i	4+6i, 2+5i
5+6i, 1+1i	5+6i, 4+7i	7i, 3+3i	7i, 3+4i	1+7i, 1+2i
1+7i, 5i	2+7i, 4+1i	2+7i, 6+6i	3+7i, 4+3i	3+7i, 7+4i
4+7i, 2+2i	4+7i, 6+5i	5+7i, 6+1i	5+7i, 3+6i	*O*

7.7 Analysis on Point Orders

The following is the study on the point order properties of elliptic curves in complex fields with compared to finite fields.

7.7.1 Experiments on Point Order over Prime Field $GF(P)$

The cyclic group order of points on the elliptic curves $E: y^2 = x^3 + x + 1$ over $GF(7)$ is shown in Table 7.5 [99].

Table 7.5 The Order of the Points on the Curve $E: y^2 = x^3 + x + 1$ over $GF(7)$

Points		Group orders
P	(0, 1)	5
2P	(2, 5)	5
3P	(2, 2)	5
4P	(0, 6)	5
5P	*O*	

7.7.2 Experiments on Point Order over Binary Field $GF(2^m)$

The cyclic group order of points on the elliptic curves $E: y^2 + xy = x^3 + x^2 + 1$ over $GF(f(x))$ where $f(x) = x^3 + x + 1$ is shown in Table 7.6 [99].

Table 7.6. The Order of the Points on the Curve $E: y^2 + xy = x^3 + x^2 + 1$ over $GF(f(x))$ where $f(x) = x^3 + x + 1$

Points		Group orders	Points		Group orders
P	(4, 3)	14	8P	(7, 0)	7
2P	(5, 0)	7	9P	(2, 7)	14
3P	(6, 3)	14	10P	(3, 3)	7
4P	(3, 0)	7	11P	(6, 5)	14
5P	(2, 5)	14	12P	(5, 5)	7
6P	(7, 7)	7	13P	(4, 7)	14
7P	(0, 1)	2	14P	*O*	

7.7.3 Experiments on Point Order over Complex Field $Z(GF(P))$

The cyclic group order of points on the elliptic curves $E: y^2 = x^3 + x + (1 + 5i)$over $Z(GF(7))$ is shown in Table 7.7 [99].

Table 7.7 The Order of the Points on the Curve
$E: y^2 = x^3 + x + (1 + 5i)$over $Z(GF(7))$

Points		Group orders	Points		Group orders
P	(5, 3+2i)	47	25P	(2+4i, 1i)	47
2P	(4i, 4+1i)	47	26P	(5+1i, 6+2i)	47
3P	(5+6i, 5)	47	27P	(2+6i, 3i)	47
4P	(4+2i, 3+3i)	47	28P	(2i, 4+6i)	47
5P	(4+4i, 5+6i)	47	29P	(3+6i, 4+6i)	47
6P	(3+3i, 6+4i)	47	30P	(1+1i, 2+2i)	47
7P	(5+4i, 5i)	47	31P	(5+3i, 5+1i)	47
8P	(4+5i, 5+2i)	47	32P	(6+6i, 5+3i)	47
9P	(2+5i, 4+1i)	47	33P	(1+6i, 1+1i)	47
10P	(4+6i, 3+1i)	47	34P	(1+5i, 3+2i)	47
11P	(5+5i, 3+6i)	47	35P	(1+2i, 4+5i)	47
12P	(1+2i, 3+2i)	47	36P	(5+5i, 4+1i)	47
13P	(1+5i, 4+5i)	47	37P	(4+6i, 4+6i)	47
14P	(1+6i, 6+6i)	47	38P	(2+5i, 3+6i)	47
15P	(6+6i, 2+4i)	47	39P	(4+5i, 2+5i)	47
16P	(5+3i, 2+6i)	47	40P	(5+4i, 2i)	47
17P	(1+1i, 5+5i)	47	41P	(3+3i, 1+3i)	47
18P	(3+6i, 3+1i)	47	42P	(4+4i, 2+1i)	47
19P	(2i, 3+1i)	47	43P	(4+2i, 4+4i)	47
20P	(2+6i, 4i)	47	44P	(5+6i, 2)	47
21P	(5+1i, 1+5i)	47	45P	(4i, 3+6i)	47
22P	(2+4i, 6i)	47	46P	(5, 4+5i)	47
23P	(1+3i, 1+2i)	47	47P	*O*	47
24P	(1+3i, 6+5i)	47			

7.7.4 Experiments on Point Order over Complex Field $Z\left(GF(2^m)\right)$

The cyclic group order of points on the elliptic curve $E: y^2 + xy = x^3 + x^2 + (1 + 5i)$ over $Z\left(GF\left(f(x)\right)\right)$ where $f(x) = x^3 + x + 1$ is shown in Table 7.8 [101].

Table 7.8 The Order of the Points on the Curve $E: y^2 = x^3 + x + (1 + 5i)$ over $Z\left(GF\left(f(x)\right)\right)$ where $f(x) = x^3 + x + 1$

Points		Group orders
P	(1, 2+5i)	5
2P	(5i, 6+1i)	5
3P	(1+4i, 7+5i)	5
4P	(3+2i, 6)	5
5P	*O*	

7.7.5 Comparison on Curve Order and Point Order

The comparison on curve orders and point orders based on the 3-bits integer number and the 3-bits complex number shown in Table 7.9 is demonstrated in Figure 7.3 as a comparison chart.

Table 7.9 Comparison on Curve Orders and Point Orders based on the 3-bits Integer Number and the 3-bits Complex Number

	Curve Order	Point Order
Prime Field (Integer)	5	5
Prime Field (Complex)	55	47
Binary Field (Integer)	14	14
Binary Field (Complex)	105	5

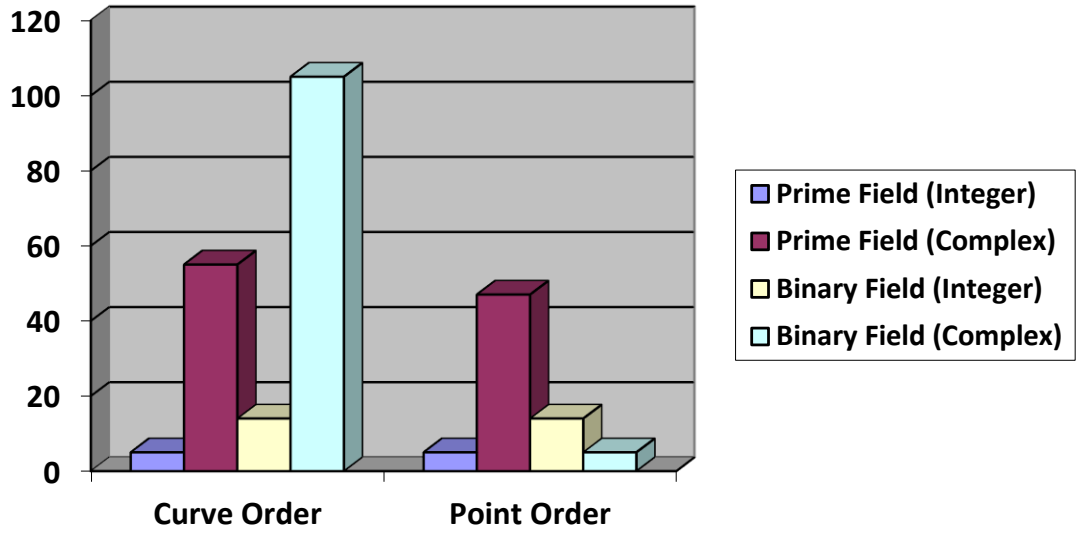

Figure 7.5: Comparison Chart on Curve Orders and Point Orders

7.7.6 Comparison on the Time Complexity

The comparison on the time complexity of the orders of point over prime fields generated by Intel(R) Core (TM) i5-5200U CPU with 2.20GHz is shown in Table 7.10.

Table 7.10 Time Complexity of the Prime fields of Integer and Complex number

Prime Field	Integer (3 bits)	Complex number (3 bits)
The order of point	5	47
Time Complexity (millisecond)	1.2	68.15

The comparison on the time complexity of the orders of point over binary fields generated by Intel(R) Core (TM) i5-5200U CPU with 2.20GHz is shown in Table 7.11.

Table 7.11 Time Complexity of the Binary fields of Integer and Complex number

Binary Field	Binary (3 bits)	Complex Number (3 bits)
The order of point	14	5
Time Complexity(millisecond)	5.93	6.6

7.8 Analysis on Computational Cost

The computational costs of arithmetic operations over finite fields with real numbers are generally specified as A for addition, B for subtraction, M for multiplication and I for inverse to study computational costs on elliptic curves.

7.8.1 Computational Cost on Complex Number Arithmetic

The computational costs of complex number arithmetic operations are generally specified as A_c for addition, B_c for subtraction, M_c for multiplication and I_c for inverse to study computational costs on elliptic curves with complex numbers. The computational costs of complex number arithmetic operations are generally computed on the base of real number arithmetic operations as follows:

Addition (A_c). $x + y = (a_1 + a_2) + (b_1 + b_2)i$ = *3A*

Subtraction (B_c). $x - y = (a_1 - a_2) + (b_1 - b_2)i$ = *A + 2B*

Multiplication (M_c). $x.y = (a_1a_2 - b_1b_2) + (a_1b_2 - a_2b_1)i$ = $4\mathrm{M} + \mathrm{A} + 2\mathrm{B}$

Reciprocal (I_c). $\frac{1}{z} = z^{-1} = \frac{a}{a^2+b^2} + \frac{b}{a^2+b^2}i$ = $3\mathrm{A} + 2\mathrm{I} + 6\mathrm{M}$

7.8.2 Computational Cost on Elliptic Curve Arithmetic Over Prime Field $E(GF(P))$

The computational costs of elliptic curve arithmetic operations over prime field are generally computed as follows:

1) **Addition of points.**

   $P = (x_1, y_1)\ and\ Q = (x_2, y_2)$, $P + Q = (x_3, y_3)$.

   In this case, $x_3 = \lambda^2 - x_1 - x_2$ and $y_3 = \lambda(x_1 - x_3) - y_1$

   where $\lambda = (y_2 - y_1)/(x_2 - x_1)$ = $6\mathrm{B} + 3\mathrm{M} + \mathrm{I}$.

2) **Doubling of a point.**

   $P = (x_1, y_1)$. $2P = (x_3, y_3)$.

   In this case, $x_3 = \lambda^2 - 2x_1$ and $y_3 = \lambda(x_1 - x_3) - y_1$

   where $\lambda = (3{x_1}^2 + a)/(2y_1)$ = $\mathrm{A} + 3\mathrm{B} + 4\mathrm{M} + \mathrm{I}$.

7.8.3 Computational Cost on Elliptic Curve Arithmetic Over Binary Field $E(GF(2^m))$

The computational costs of elliptic curve arithmetic operations over binary field are generally computed as follows:

1) **Addition of points.**

   $P = (x_1, y_1)\ and\ Q = (x_2, y_2)$, $P + Q = (x_3, y_3)$.

In this case, $x_3 = \lambda^2 + \lambda + x_1 + x_2 + a \; and \; y_3 = \lambda(x_1 + x_3) + x_3 + y_1$

where $\lambda = (y_2 + y_1)/(x_2 + x_1)$ = 9A + 3M + I

2) Doubling of a point.

$P = (x_1, y_1)$. $2P = (x_3, y_3)$.

In this case, $x_3 = \lambda^2 + \lambda + a \; and \quad y_3 = {x_1}^2 + \lambda x_3 + x_3$

where $\lambda = x_1 + (y_1/x_1)$ = 5A + 3M + I

7.8.4 Computational Cost on Elliptic Curve Arithmetic Over Prime Field with Complex Number

The computational costs of elliptic curve arithmetic operations over prime field with complex number are generally computed on the base of complex number arithmetic operation costs. Then they are transformed to the computational costs based on the real number arithmetic operations for cost analysis on each of fundamental arithmetic operations.

1) Addition of points.

$P = (x_1, y_1) \; and \; Q = (x_2, y_2), \; P + Q = (x_3, y_3)$.

In this case, $x_3 = \lambda^2 - x_1 - x_2$ and $y_3 = \lambda(x_1 - x_3) - y_1$

where $\lambda = (y_2 - y_1)/(x_2 - x_1)$

$= 3\mathrm{M}_c + \mathrm{I}_c + 6\mathrm{B}_c$

$=3[4M + A + 2B] + [6M + 3A + 2I] + 6[A + 2B]$

$= [12M + 3A + 6B] + [6M + 3A + 2I] + [6A + 12B]$

$= 12A + 18B + 18M + 2I.$

2) Doubling of a point.

$P = (x_1, y_1)$. $2P = (x_3, y_3)$.

In this case, $x_3 = \lambda^2 - 2x_1 \; and \quad y_3 = \lambda(x_1 - x_3) - y_1$

where $\lambda = (3{x_1}^2 + a)/(2y_1)$

$= 4\mathrm{M}_c + \mathrm{I}_c + \mathrm{A}_c + 3\mathrm{B}_c$

$= 4[4M + A + 2B] + [6M + 3A + 2I] + 3[A + 2B]$

$= [16M + 4A + 8B] + [6M + 3A + 2I] + [3A + 6B]$

$= 10A + 14B + 22M + 2I.$

7.8.5 Computational Cost on Elliptic Curve Arithmetic Over Binary Field with Complex Number

The computational costs of elliptic curve arithmetic operations over binary field with complex number are generally computed on the base of complex number arithmetic operation costs. Then they are transformed to the computational costs based on the real number arithmetic operations for cost analysis on each of fundamental arithmetic operations.

1) Addition of points.

$P = (x_1, y_1)\ and\ Q = (x_2, y_2\ ),\ P + Q = (x_3\ , y_3).$

In this case, $x_3 = \lambda^2 + \lambda + x_1 + x_2 + a\ and\ \ y_3 = \lambda(x_1\ + x_3\ ) + x_3 + \ y_1$

$where\ \lambda = (y_2\ + y_1)/(x_2\ + x_1)$

$= 3\mathrm{M}_c + \mathrm{I}_c + 9\mathrm{A}_c$

$= 3[4M + A + 2B] + [6M + 3A + 2I] + 9[3A]$

$= [12M + 3A + 6B] + [6M + 3A + 2I] + 27A$

$= 33A + 6B + 18M + 2I.$

2) Doubling of a point.

$P = (x_1, y_1\ ).\ 2P = (x_3\ , y_3).$

In this case, $x_3 = \lambda^2 + \lambda + a\ \ and\quad y_3 = {x_1}^2 + \lambda x_3 + x_3$

$$where\ \lambda = x_1 + (y_1/x_1)$$

$$= 3\mathrm{M}_c + \mathrm{I}_c + 5\mathrm{A}_c$$

$$= 3[4M + A + 2B] + [6M + 3A + 2I] + 5[3A]$$

$$=[12M + 3A + 6B] + [6M + 3A + 2I] + [15A]$$

$$= 21A + 6B + 18M + 2I.$$

7.8.6 Comparison on Computational Costs for Addition of Points and Doubling of a Point

In actuality, the computational cost is quantified in milliseconds, which is the amount of time it takes to compute an addition of points or a double of a point. However, to analyse computational cost of fundamental arithmetic operations, let 'A' be one millisecond for addition, 'B' be one millisecond for subtraction, 'M' be two milliseconds for multiplication, and 'I' be three milliseconds for inverse. Then the computational costs for an addition of points and a double of a point based on the cost of fundamental arithmetic operations are computed as followings. Table 7.12 shows computational costs for addition of points and doubling of a point. Figure 7.4 illustrates in bar chart for comparison on computational costs of an addition of points and a double of a point that are generated from different curves. Table 7.13 shows computational costs of fundamental arithmetic operations for addition of points. Figure 7.5 illustrates in bar chart for comparison on computational costs of fundamental arithmetic operations for addition of points that are generated from different curves. Table 7.14 shows computational costs of fundamental arithmetic operations for doubling of a point. Figure 7.6 illustrates in bar chart for comparison on computational costs of fundamental arithmetic operations for doubling of a point that is generated from different curves. Let A = 1, B = 1, M = 2, I = 3.

1) Prime Field (Integer)

- *Addition of points:* $6B + 3M + I = 6(1) + 3(2) + 3 = 6 + 6 + 3 = 15.$
- *Doubling of a point:* $A + 3B + 4M + I = 1 + 3(1) + 4(2) + 3$
  $= 1 + 3 + 8 + 3 = 15.$

2) Prime Field (Complex)

- *Addition of points:* $12A + 18B + 18M + 2I = 12(1) + 18(1) + 18(2) + 2(3) = 12 + 18 + 36 + 6 = 72.$
- *Doubling of a point:* $10A + 14B + 22M + 2I = 10(1) + 14(1) + 22(2) + 2(3) = 10 + 14 + 44 + 6 = 74$

3) Binary Field (Integer)

- *Addition of points:* $9\text{A} + 3\text{M} + \text{I} = 9(1) + 3(2) + 3 = 9 + 6 + 3 = 18$
- *Doubling of a point:* $5\text{A} + 3\text{M} + \text{I} = 5(1) + 3(2) + 3 = 5 + 6 + 3 = 14$

4) Binary Field (Complex)

- *Addition of points:* $33A + 6B + 18M + 2I = 33(1) + 6(1) + 18(2) + 2(3) = 33 + 6 + 36 + 6 = 81$
- *Doubling of a point:* $21A + 6B + 18M + 2I = 21(1) + 6(1) + 18(2) + 2(3) = 21 + 6 + 36 + 6 = 69.$

Table 7.12 Computational Costs for Addition of Points and Doubling of a Point

	Prime (Integer)	**Binary (Integer)**	**Prime (Complex)**	**Binary (Complex)**
Addition	15	18	72	81
Doubling	15	14	74	69

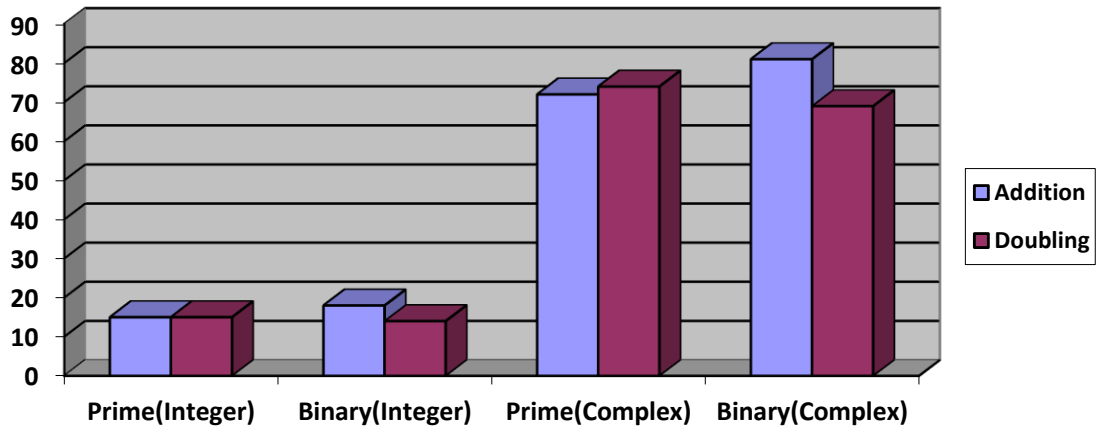

Figure 7.6 : Comparison on Computational Costs for Addition of Points and Doubling of a Point

Table 7.13. Computational Costs of Arithmetic Operations for Addition of Points

	Prime (Integer)	Binary (Integer)	Prime (Complex)	Binary (Complex)
A (Addition)		9	12	33
B (Subtraction)	6		18	6
M(Multiplication)	3	3	18	18
I (Inverse)	1	1	2	2

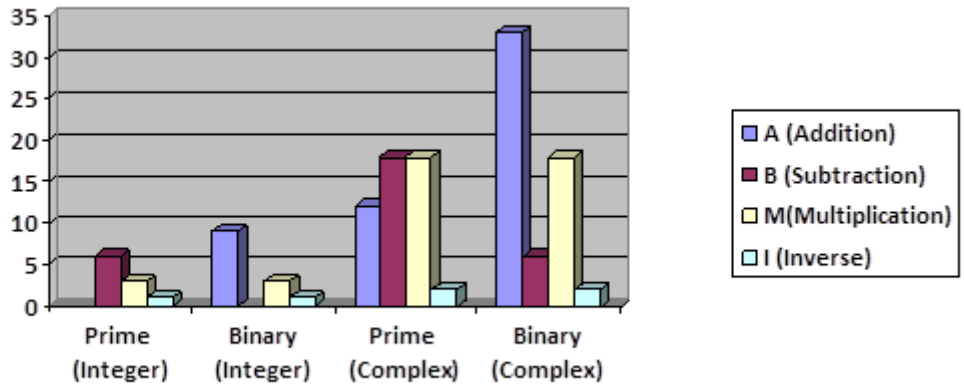

Figure 7.7 Comparison on Computational Costs of Arithmetic Operations for Addition of Points

Table 7. 14 Computational Costs of Arithmetic Operations for Doubling of a Point

	Prime (Integer)	Binary (Integer)	Prime (Complex)	Binary (Complex)
A(Addition)	1	5	10	21
B (Subtraction)	3		14	6
M(Multiplication)	4	3	22	18
I (Inverse)	1	1	2	2

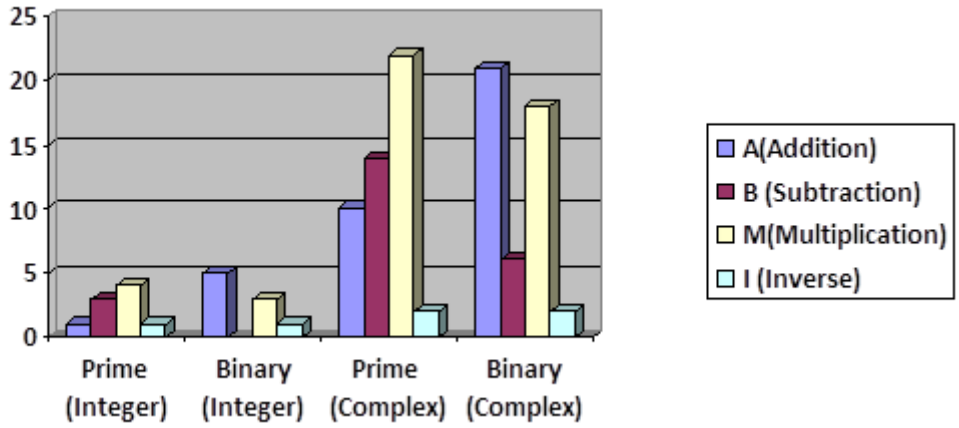

Figure 7.8 : Comparison on Computational Costs of Arithmetic Operations for Doubling of a Point

7.9 Summary

Matrix algebra and elliptic curve arithmetic over complex fields have mathematical characteristics that are consistent with those of finite fields. Like finite fields, complex fields have mathematical features as well. In comparison to elliptic curves over prime fields and binary fields, complex fields have a larger order of elliptic curves. But, the order of a base point on an elliptic curve in the complex field $Z\big(GF(2^m)\big)$ is smaller than that of a base point on the curve in the binary field $GF(2^m)$ because the sequence of the points cannot be continuously generated by the point on the curve in the complex field $Z\big(GF(2^m)\big)$ in case of the Boolean complexity on the basic arithmetic operations. The greatest of them is the order of a base point on an elliptic curve in the complex field- $Z\big(GF(P)\big)$. According to Hasses's theorem [114], the cyclic group order on the elliptic curves over $Z\big(GF(P)\big)$ may be between $+1-2\sqrt{q}$ and $q+1+2\sqrt{q}$, where $q=p^2$. The cyclic group order of the points on the elliptic curves over $Z\big(GF(P)\big)$ shown in section 7.7.3 is 47 which is between $49+1-2\sqrt{49}=36$ and $49+1+2\sqrt{49}=64$. Thus, the security level, in terms of the cyclic group order, is roughly squared [101]. The problem solving of ECDLP is computationally harder and more time–consuming in complex fields. The time complexity of solving on ECDLP also becomes squared in case of implementation on $Z\big(GF(P)\big)$. Therefore, the security of ECC is greatly improved on implementing an elliptic curve over complex field- $Z\big(GF(P)\big)$. In addition, the computational costs on the

fundamental arithmetic operations with elliptic curve over complex field- $Z\big(GF(P)\big)$ approve that the arithmetic of elliptic curves does not heavily rely on a single arithmetic operator and protects the mathematical attacks. The suggested method makes solving the ECDLP for elliptic curve attacks much more challenging and time-consuming. The proposed technique enables system developers to achieve a higher level of security with a small increase in storage.

CHAPTER 8
CRYPTOGRAPHIC APPLICATIONS

This chapter is to describe how to apply non-linear transformation techniques using complex numbers in traditional Hill cipher and elliptic curve cryptography for security improvement. The organization of this chapter is as follows. For improved security in the encryption and decryption of the Hill cipher, section 8.2 contains implementation strategies employing complex integers. In elliptic curve cryptographic protocols, such as elliptic curve encryption scheme and elliptic curve digital signature method, non-linear transformation techniques using complex numbers are implemented in section 8.3. For future work, how to use complex vector space in quantum cryptography is discussed in section 8.4. Section 8.5 describes a summary of the chapter.

Numerous application domains, such as coding theory and cryptography, usually depend heavily on finite fields [109]. Finite field cryptographic applications are used to be hacked by computer scientists and hackers who can explore the weak mathematical structure of finite fields [68, 76]. The diffusion and confusion features of the ciphers on complex fields are stronger than those of the standard ciphers on the real numbers because complex fields enable complicated mathematical transformations [113]. Therefore, matrix algebra and elliptic curve arithmetic over complex fields have stronger mathematical characteristics than those of finite fields [77, 78, 101].

8.1 Hill Ciphers

Hill cipher is a standard cipher based on matrix transformations [45]. The encryption scheme in Hill Cipher uses a linear matrix transformation to translate plaintext to ciphertext: $C = M \times K$ where M represents "*a plaintext matrix*", K represents "*a key matrix*" and C represents "*a ciphertext matrix*". "*A plaintext matrix*" is opened by transforming: $M = C \times K^{-1}$, satisfying $K \times K^{-1} = I$ [67, 90].

Affine Hill cipher extends the concept of Hill cipher by integrating it with a nonlinear affine transformation [4, 70]. The encryption scheme follows the nonlinear transformation of matrices: $C = (M \times K) + E$ where M represents "*a plaintext matrix*", K represents "*a key matrix*", E represents "*an embedded matrix*" and C represents "*a ciphertext matrix*". "*A plaintext matrix*" is opened by transforming: $M = (C - E) \times K^{-1}$, satisfying $K \times K^{-1} = I$ [77].

8.1.1 Hill Cipher on Complex Field

A new design of Hill cipher is developed on the complex field to construct non-linear cryptographic transformations [46]. The new implementation of Hill cipher over complex plane has diffusion and confusion properties stronger than the traditional Hill ciphers on the real numbers and it enhances the complexity of the ciphertext due to nonlinear transformations [66]. Consequently, the adversary cannot use the relationship between the plaintext and the ciphertext to find the key. Therefore, it is resistant against not only known-plaintext attack but also chosen-plaintext and ciphertext attacks [39]. The encryption and decryption schemes of the new Hill cipher on complex plane are followings. For implementation, the English letters 'A' to 'Z' are supposed to be set the numbers '0' to '25', space character (□) to '26', comma to '27', question mark to '28', apostrophe to '29' and full stop to '30'. The total number of elements in the finite field is 31. Therefore, the new design is implemented on the complex field Z(GF (31)). The implementation experiments on encryption and decryption schemes are demonstrated as follows [77].

Encryption Scheme. The plaintext characters are $m_1, m_2, m_3, \dots, m_i$ in which m_j is the j^{th} character and l is the number of characters acceptable to a package in data transmission. Each couple of characters of the plaintext is embedded in the complex number form: $m_j + m_{j+1}i$ on $Z(GF(p))$ where p is a prime. These complex numbers are set up in the matrix *M* on Z(GF(p)). The rank of the matrix *M* must be compatible with the ranks of the matrices *K and E* [28, 83].

The first key is $k_1, k_2, k_3, \dots, k_i$ over $Z(GF(p))$ which k_j is the j^{th} key character and l is the number of key characters. Each couple of key characters is embedded in the complex number form: $k_j + k_{j+1}i$ on $Z(GF(p))$ .These complex numbers are set up in the matrix K on $Z(GF(p))$. The rank of the matrix K must be compatible with the matrix M [50].

The second key is $e_1, e_2, e_3, \dots, e_i$ over $Z(GF(p))$ and each couple of key characters is embedded in the complex number form: $e_j + e_{j+1}i$ on $Z(GF(p))$ .These complex numbers are set up in the matrix E on $Z(GF((p))$. The rank of the matrix E must be compatible with the rank of the matrix M [107].

The matrix C is calculated through the nonlinear matrix transformations: $C = (e_{ij} \times M \times K) + E$ on $Z(GF(p))$. e_{ij} is a component of the matrix E. The cipher

text characters $c_1, c_2, c_3, \ldots, c_i$ are retrieved from the complex number $c_j + c_{j+1}i$ through the matrix C [52, 56].

Experiment on Encryption

Plaintext = SECURITY = $\{18, 4, 2, 20, 17, 8, 19, 24\}$

First Key = RAINFALL= $\{17, 0, 8, 13, 5, 0, 11, 11\}$

Second Key = UCSYUCSY = $\{20, 2, 18, 24, 20, 2, 18, 24\}$

$$\mathrm{M} = \{\{18 + 4i, 2 + 20i\}, \{17 + 8i, 19 + 24i\}\}$$

$$K = \{\{17, 8 + 13i\}, \{5, 11 + 11i\}\}$$

$$\mathrm{E} = \{\{20 + 2i, 18 + 24i\}, \{20 + 2i, 18 + 24i\}\}$$

$$\mathrm{M} = \begin{bmatrix} 18 + 4i & 2 + 20i \\ 17 + 8i & 19 + 24i \end{bmatrix}$$

$$\mathrm{K} = \begin{bmatrix} 17 & 8 + 13i \\ 5 & 11 + 11i \end{bmatrix}$$

$$\mathrm{E} = \begin{bmatrix} 20 + 2i & 18 + 24i \\ 20 + 2i & 18 + 24i \end{bmatrix}$$

$$\mathrm{C} = (e_{ij} \times M \times K) + E = \begin{bmatrix} 2 + 15i & 14 + 17i \\ 18 + 21i & 22 + 24i \end{bmatrix}$$

$$\mathrm{C} = \{\{2 + 15i, 14 + 17i\}, \{18 + 21i, 22 + 24i\}\}$$

Cipher text $= \{2, 15, 14, 17, 18, 21, 22, 24\} = \mathrm{CPORSVWY}$

Decryption Scheme. Each couple of characters of the cipher text: $c_1, c_2, c_3, \ldots, c_i$ is embedded in the complex number form: $c_j + c_{j+1}i$ on $Z(GF(p))$. These complex numbers are set up in matrix *C*. Like the matrix *M*, the rank of the matrix *C* must be compatible with the ranks of the matrices *K and E* [29].

The matrix K is created by the same way described in the encryption scheme and K^{-1} is calculated such that $K \times K^{-1} = I$.

The matrix E is created in the same way described in the encryption scheme.

The matrix M is calculated through the nonlinear matrix transformations: $M = (C - E) \times K^{-1} \times {e_{ij}}^{-1}$ on $Z(GF(p))$. ${e_{ij}}^{-1}$ is the reciprocal of the component of the matrix E. The plaintext character $m_1, m_2, m_3, \ldots, m_i$ are retrieved from the complex number $m_j + m_{j+1}i$ through the matrix M.

Experiment on Decryption

Cipher text = CPORSVWY = $\{2, 15, 14, 17, 18, 21, 22, 24\}$

First Key = RAINFALL = $\{17, 0, 8, 13, 5, 0, 11, 11\}$

Second Key = UCSYUCSY = $\{20, 2, 18, 24, 20, 2, 18, 24\}$

$$C = \{\{2 + 15i, 14 + 17i\}, \{18 + 21i, 22 + 24i\}\}$$

$$K = \{\{17, 8 + 13i\}, \{5, 11 + 11i\}\}$$

$$E = \{\{20 + 2i, 18 + 24i\}, \{20 + 2i, 18 + 24i\}\}$$

$$C = \begin{bmatrix} 2 + 15i & 14 + 17i \\ 18 + 21i & 22 + 24i \end{bmatrix}$$

$$K^{-1} = \begin{bmatrix} 23 + 20i & 15 + 25i \\ 17 + 19i & 29 + 16i \end{bmatrix}$$

$$E = \begin{bmatrix} 20 + 2i & 18 + 24i \\ 20 + 2i & 18 + 24i \end{bmatrix}$$

$$M = (C - E) \times K^{-1} \times {e_{ij}}^{-1} = \begin{bmatrix} 18 + 4i & 2 + 20i \\ 17 + 8i & 19 + 24i \end{bmatrix}$$

$$M = \{\{18 + 4i, 2 + 20i\}, \{17 + 8i, 19 + 24i\}\}$$

Plaintext = $\{18, 4, 2, 20, 17, 8, 19, 24\}$ = SECURITY

8.2 Elliptic Curve Cryptography

The safety measures of "*Elliptic Curve Cryptography*" (ECC) mostly rely on the quality of solving "*Elliptic Curve Discrete Logarithm Problem*" (ECDLP). Known *P* and *Q* on the curve satisfying *Q = kP*, the integer *k* is the discrete logarithm of *Q* to the base *P*. Although *P* and *Q* are known, the group order of the curve is so huge that it is unable to compute *k*. It was observed in section 1.6 that elliptic curve general attacks obviously would resolve ECDLP if the group order of the curve was small enough to compute *k* [75].

The implementation and the analysis on the mathematical properties of elliptic curve arithmetic based on the integration of complex number arithmetic with modular arithmetic are described in chapter 7. The group orders of the curves in the fields: $GF(p), GF\ (2^m), Z\big(GF(p)\big), and\ Z\big(GF(2^m)\big)$ were observed in which the group order

of the curve in $Z(GF(p))$ is 47 that exists between $49+1-2\sqrt{49}=36$ and $49+1+2\sqrt{49}=64$ in the case of $q+1-2\sqrt{q}$ and $q=p^2$ in accordance with Hasses's theorem [114]. Hence, the security rank is approximately squared in terms of the group order. The attempt to resolve ECDLP on complex plane is so more computationally difficult that time duration will be long. Thus, the safety rank of ECC is significantly enhanced by using the curve of $Z(GF(p))$.

8.2.1 Elliptic Curve ElGamal Encryption Scheme on Complex Field

The non-linear cryptographic transformations for encryption scheme and decryption scheme can be created by using elliptic curve arithmetic based on the integration of complex number arithmetic with modular arithmetic to improve the security of the typical elliptic curve ElGamal encryption scheme [101].

Let's consider to encrypt and decrypt the message using the elliptic curve $E: y^2 = x^3 + x + (1+5i)$ over $Z(GF(7))$ where $a=1$ and $b=1+5i$.The followings are the results of elliptic curve ElGamal encryption scheme implemented on the given curve through the developed system using java language [51].

Key Generation. Entity A and Entity B agree to choose the point $P=(5,3+2i)$ with prime order $n=47$ as a base point [89]. Entity B compute a private key and a public key as followings.

- Entity B chooses an integer $d=13$ as a private key.
- Entity B computes $Q=d\times P=13\times(5,3+2i)=(1+5i,4+5i)$ as a public key.

Encryption Scheme. Entity A chooses the point $M=(2+5i,3+6i)$ as a message to encrypt with the public key $Q=(1+5i,4+5i)$. The cipher text is computed as followings.

- Entity A chooses an integer $r=3$ as a random number.
- Entity A computes: $C_1=r\times P=3\times(5,3+2i)=(5+6i,5)$.
- Entity A computes: $C_2=M+(r\times Q)$
$$=(2+5i,3+6i)+3.(1+5i,4+5i)$$
$$=(1+1i,2+2i).$$
- Entity A sends the points C_1 and C_2 to Entity B as cipher texts.

Decryption Scheme. Entity B receives the points $C_1 = (5 + 6i, 5)$ and $C_2 = (1 + 1i, 2 + 2i)$ as cipher texts to decrypt with the private key $d = 13$. The message plain text is computed as following.

- Entity B computes the message $M = C_2 - (d \times C_1)$
$= (1 + 1i, 2 + 2i) - 13.(5 + 6i, 5)$
$= (2 + 5i, 3 + 6i).$

8.2.2 Elliptic Curve ElGamal Signature Scheme on Complex Field

The non-linear cryptographic transformations for signature signing scheme and signature verifying scheme can also be created by using elliptic curve arithmetic based on the integration of complex number arithmetic with modular arithmetic to improve the security of the typical elliptic curve ElGamal signature scheme [101].

Let's consider to sign the message and verify the signature using the elliptic curve$E: y^2 = x^3 + x + (1 + 5i)$ over $Z\big(GF(7)\big)$ where $a = 1$ and $b = 1 + 5i$ .The followings are the results of elliptic curve ElGamal signature scheme implemented on the given curve through the developed system using java language [112].

Key Generation. Entity A and Entity B agree to choose the point $P = (5, 3 + 2i)$ with prime order $n = 47$ as a base point. Entity A computes a private key and a public key as followings [106].

- Entity A chooses an integer $d = 13$ as a private key.
- Entity A computes $Q = d \times P = 13 \times (5, 3 + 2i) = (1 + 5i, 4 + 5i)$ as a public key.

Signing Scheme. Entity A computes the signature as the followings to sign the message with the private key $d = 13$ [36].

- Entity A chooses an integer $k = 3$ as a random number.
- Entity A computes: $R = k \times P = 3 \times (5, 3 + 2i) = (5 + 6i, 5)$.
- Entity A computes: $r = R.x.real \ mod \ n = 5 \ mod \ 47 = 5$.
- Entity A computes: $h = h(m) = 4$ as the hash value of the signing message.
- Entity A computes:$s = k^{-1}(h + rd) \ mod \ n = 16 \times (45 \times 13) \ mod \ 47$
$= 42$
- Entity A sends (R, s) to Entity B as the signature of the signing message.

Verifying Scheme. Entity B receives (R, s) as the signature of the signing message and computes the followings to verify the signature with the public key $Q = (1 + 5i, 4 + 5i)$.

- Entity B computes: $V_1 = s \times R = 42 \times (5 + 6i, 5) = (6 + 6i, 5 + 3i)$.
- Entity B computes: $r = R.x.real \; mod \; n = 5 \; mod \; 47 = 5$.
- Entity B computes: $h = h(m) = 4$ as the hash value of the same signing message.
- Entity B computes: $U_1 = h \times P = 4 \times (5, 3 + 2i) = (4 + 2i, 3 + 3i)$.
- Entity B computes: $U_2 = r \times Q = 5 \times (1 + 5i, 4 + 5i) = (2i, 4 + 6i)$.
- Entity B computes: $V_2 = U_1 + U_2 = (6 + 6i, 5 + 3i)$.
- Entity B accepts the signature, since $V_1 = V_2$.

8.3 Quantum Cryptography

Modern cryptography algorithms are based on the most important method of factorization against giant integers into their primes that is tough to solve. But modern cryptography is at risk of each technical growth of computing power in arithmetic to quickly reverse one-way functions like that of factorization massive numbers. So, quantum computing is introduced into cryptography that ends up in the analysis of quantum cryptography. Quantum cryptography is the science that applies quantum mechanics to carry out cryptographic tasks. Quantum cryptography is one of the emerging topics within the sector of computer industry. Quantum cryptography brings quantum key generation, quantum key distribution, quantum public key encryption and quantum random number generation. Quantum cryptography covers the weaknesses of recent digital cryptosystems, and eventually towards the longer-term direction [82].

A classical machine device for quantum computing has a two-state system: $0 \leftrightarrow 1$. The state of a quantum bit known as cubic is generally represented by the expression: $\alpha|0\rangle + \beta|1\rangle$, in which α and β are complex numbers which agree with $|\alpha|^2 + |\beta|^2 = 1$. Thus, a cubic state is represented as a unit vector in a set of complex numbers which is known as the 2-dimensional complex vector space shown in

Figure 8.1 [81, 118]. The vector $\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ may be expressed as $\alpha|0\rangle + \beta|1\rangle$ in quantum computing where $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ as in [118].

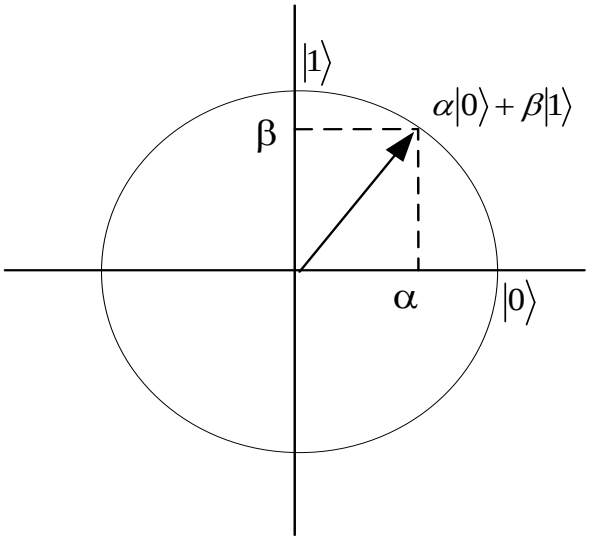

Figure 8.1: Visualization of a Qubit State

Quantum computing is totally different from most different branches of science therein it uses complex numbers in an elementary way. Quantum computing is driven through the language of complex vector space which is the set of vectors of a fixed length with complex entries. These vectors describe the states of quantum systems and quantum computers. The important role of complex vector spaces in quantum computing is described in the references [79, 80, 81].

8.4 Summary

The mathematical characteristics of complex fields are consistent with those of finite fields. Complex fields support complex mathematical transformations to improve the security features of both symmetric key cryptosystems and asymmetric key cryptosystems. Therefore, complex fields should be used in future sustainable development of cryptographic applications.

CHAPTER 9
CONCLUSION AND DISCUSSION

Residue matrices and elliptic curve arithmetic based on modular number arithmetic are often utilized by numerical calculations in traditional and modern cryptographic techniques. In recent years, traditional ciphers and modern ciphers applied non-linear cryptographic transformation methods using on residue matrices and elliptic curve arithmetic over finite fields of integer numbers and binary numbers. They are well known that traditional ciphers are vulnerable to known-plaintext attacks and chosen-ciphertext attacks while modern ciphers are subject to typical generic attacks like the Baby-Step, Giant-Step, Pollard's Rho, and Pohlig-Hellman methods.

The contributions of this dissertation includes the step-by-step implementation procedures for computing matrix algebra and elliptic curve arithmetic in complex field, the study on the mathematical characteristics of finite field integrating with complex numbers, the analysis on the mathematical characteristics of residue matrices integrating with complex numbers, the analysis on the mathematical characteristics of elliptic curve arithmetic integrating with complex numbers, the analysis on the curve order and on the cyclic group which are generated from the elliptic curves integrating with complex numbers. The computational costs on the arithmetic operations with real numbers, complex numbers and elliptic curves with real numbers and complex numbers are also studied. This dissertation supports non-linear cryptographic transformation techniques by using mathematical properties of residue matrices and elliptic curve arithmetic on complex plane which is made of complex numbers over finite fields based on modular arithmetic to improve the security.

9.1. Discussion

Matrix algebra and elliptic curve arithmetic over complex fields have mathematical characteristics that are consistent with those of finite fields. Like finite fields, complex fields have mathematical features as well.

The new implementation of Hill cipher over complex plane has the diffusion and confusion properties stronger than the traditional Hill ciphers on the real numbers and it enhances the complexity of the ciphertext due to nonlinear transformations. As a consequence, the adversary cannot use the relationship between the plaintext and the

ciphertext to find the key. Therefore, it is resistant against not only known-plaintext attack but also chosen-plaintext and ciphertext attacks.

In comparison to elliptic curves over prime fields and binary fields, complex fields have a larger order of elliptic curves. But, the order of a base point on an elliptic curve in the complex field $Z\big(GF(2^m)\big)$ is smaller than that of a base point on the curve in the binary field $GF(2^m)$ because the sequence of the points cannot be continuously generated by the point on the curve in the complex field $Z\big(GF(2^m)\big)$ in case of large amount of Boolean complexity on the basic arithmetic operations. The greatest of them is the order of a base point on an elliptic curve in the complex field *Z(GF(p)).* According to Hasses's theorem, the cyclic group order on the elliptic curves over $Z\big(GF(p)\big)$ may be between $q + 1 - 2\sqrt{q}$ and $q + 1 + 2\sqrt{q}$, where $q = p^2$. Thus, the security level, in terms of the cyclic group order, is roughly squared. The problem solving of ECDLP is computationally harder and more time–consuming in complex field. The time complexity of solving on ECDLP also becomes squared in case of implementation on $Z\big(GF(p)\big)$. Therefore, the security of ECC is greatly improved on implementing an elliptic curve over complex field-$Z\big(GF(p)\big)$. Additionally, the arithmetic of elliptic curves does not strongly rely on a single arithmetic operator, which defends against mathematical attacks, according to computation costs on the fundamental arithmetic operations with elliptic curve over complex field-$Z\big(GF(p)\big)$. The recommended approach significantly increases the difficulty and time required to solve the ECDLP for elliptic curve attacks. The proposed technique enables system developers to achieve a higher level of security with a small increase in storage.

9.2 Advantages and Limitations

Intelligent computing on the complex plane-based on the integration of complex number arithmetic with modular arithmetic is beneficial to the cryptographic applications. The proposed techniques need to double the memory areas to store the keys however their security levels are generally squared. The complex plane supports the non-linear cryptographic transformations not only for traditional ciphers but also for elliptic curve cryptography to get more secure for sustainable development. This research points to the importance of complex planes in modern cryptography. It is expected that the results of this research may become useful for the choice of the next-generation cryptographic applications. The research work supports the analysis of the

computational costs on the fundamental arithmetic operators used in cryptographic transformations of the proposed technique to protect the mathematical attacks. The followings are advantages and limitations on the complex fields: Z(GF(p)) and $Z\left(GF(2^m)\right)$.

Advantages

- Cyclic group order is bigger by using complex field Z(GF(p)).
- Time complexity is higher by using complex field Z(GF(p)).
- Support non-linear cryptographic transformations by using complex field Z(GF(p)).
- Develop traditional ciphers and modern ciphers by using complex field Z(GF(p)).
- Protect common attacks and mathematical attacks by using complex field Z(GF(p)).
- Improve the security of traditional ciphers and modern ciphers by using complex field Z(GF(p)).

Limitations

- The sequence of the points cannot be continuously generated by the point on the curve in the complex field $Z\left(GF(2^m)\right)$ in case of large amount of Boolean complexity on the basic arithmetic operations.
- The order of a base point on an elliptic curve in the complex field $Z\left(GF(2^m)\right)$ is smaller than that of a base point on the curve in the binary field $GF(2^m)$.
- Should not develop traditional ciphers and modern ciphers by using complex field $Z\left(GF(2^m)\right)$.

9.3 Future Work

The proposed techniques can be utilized to support non-linear mathematical transformations in various existing traditional ciphers and modern ciphers including quantum cryptography for more secure sustainable developments in future work.

9.4 Summary

A complex plane which is made of complex numbers over finite fields is becoming more valuable in computing science areas that deal with the applications in cryptography to make them more stable and more secure. As a result, the mathematical

properties of residue matrices and elliptic curve arithmetic on complex plane which is made of complex numbers over finite fields are observed to use them in the applications of cryptographic science. In cryptography, the security analysis based on the computational costs of fundamental arithmetic operations is a key component to protect the mathematical attacks. Therefore, this dissertation supports mathematical implementation, mathematical properties and features, and mathematical analysis to be applied in cryptography.

AUTHOR'S PUBLICATIONS

[$P1$] Ni Ni Hla, Tun Myat Aung, "Implementation of Finite Field Arithmetic Operations for Large Prime and Binary Fields Using java BigInteger class", In: International journal of Engineering Research and Technology, ISSN: 2278-0181, vol 6, no 8, pp. 450-453, 2017 August.

[$P2$] Ni Ni Hla, Tun Myat Aung, "Attack Experiments on Elliptic Curves of Prime and Binary Fields", In: IEMIS (2018), Advances in Intelligent Systems and Computing, Spriner, ISSN: 2194-5357, ISBN:978-981-13-1950-1, vol 755, pp.667-683, Indexing: Scopus, Web of Science, SJR, Conference Paper, 2018.

[$P3$] Ni Ni Hla, Tun Myat Aung, "Experiments on Implementation of Elliptic Curve Arithmetic over Complex Fields Using java BigInteger Class", In: Journal of Communications, ISSN 1796-2021, vol 14, no. 4, pp.293-300, United States, Indexing: Scopus, Scimago Journal Rank (SJR)(Q3)(2019) , Journal Paper.2019.

[$P4$] Ni Ni Hla, Tun Myat Aung, "A Complex Number Approach to Elliptic Curve Cryptosystems over Finite Fields:", In: IEEE Proceeding of ICCCI 2019, ISBN:978-1-5386-8259-3, ISSN:2329-7190, pp.221-228, Indexing: Scopus, Web of Science, SJR, Science of Security and Privacy, Conference Paper, 2019

[$P5$] Ni Ni Hla, Tun Myat Aung, "Computing and Analysis of Residue Matrices over Complex Plane for Cryptographic Applications", In: IEEE Proceeding of ICCIT-1441, ISBN: 978-1-7281-2680-7, pp.40-44, Saudi Arabia, Indexing: Scopus and Web of Science, Conference Paper, 2020.

Bibliography

[1]. A. Almarimi; U. Alsahdi, “Developing a cryptosystem for XML documents”, In: International Conference on Computer Technology and Development (ICCTD 2010), IEEE, DOI: 10.1109/ICCTD.2010.5645879.

[2]. A. B. Kinsman, N. Nicola, “Automated Range and Precision Bit-Width Allocation for Iterative Computations”, In: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, IEEE, 2011, DOI: 10.1109/TCAD.2011.2152840.

[3]. A. Jana, D. Maity, “Code-based Analysis Approach to Detect and Prevent SQL Injection Attacks”, In: International Conference on Computing, Communication and Networking Technologies (ICCCNT 2020), IEEE, DOI: 10.1109/ICCCNT49239.2020.9225575.

[4]. A. Ramesh, “Enhancing the Security of Hill Cipher using Columnar Transposition”, In: International journal of engineering research and technology, vol. 4, no. 7, 2015, DOI: 10.17577/IJERTV4IS070560.

[5]. Annabell Kuldmaa, “Efficient Multiplication in Binary Fields”, 2015.

[6]. Anoop, M.S.“Elliptic Curve Cryptography”.http://www.infosecwriters.com /Papers/Anoopms_ECC.pdf.

[7]. Ashraf A.M. Khalaf, Mona S. Abd El-karim and Hesham F.A. Hamed., “A triple hill cipher algorithm proposed to increase the security of encrypted binary data and its implementation using FPGA”, In: International Conference on Advanced Communication Technology (ICACT), IEEE, 2016. L4

[8]. B. A. Forouzan, “Elliptic Curve Cryptosystems”, In: Cryptography and Network Security, International Edition, Singapore, McGraw-Hill press, 2008, pp. 321-330.

[9]. B. A. Forouzan, “Introduction”, In: Cryptography and Network Security, International Edition, Singapore, McGraw-Hill press, 2008, pp. 1-14.

[10]. B. A. Forouzan. “Mathematics of Cryptography”-, In: Cryptography and Network Security-, International Edition, Singapore, McGraw-Hill press, 2008, pp 98-117. L1

[11]. B. A. Forouzan., “Traditional Symmetric Key Ciphers”, In: Cryptography and Network Security, McGraw-Hill Press, Int. Edition, 2008, pp.56-92. L2

[12]. Bhumika Dutta, “Cryptanalysis in Cryptography: Types and Applications”, https://www.analyticssteps.com/blogs/cryptanalysis-cryptography-types-and-applications

[13]. C. Li; S. Li; D. Zhang; G. Chen, “Cryptanalysis of a data security protection scheme for VoIP”, In: IEE Proceedings – Vision Image and Signal Processing, 2006, DOI: 10.1049/ip-vis:20045234.

[14]. D. Hankerson, A. Menezes, and S. Vanstone, “Elliptic Curve Arithmetic”, In: Guide to Elliptic Curve Cryptography, New York, USA, Springer Verlag, 2004, pp. 75-152.

[15]. D. Hankerson, S. Vanstone, A. Menezes, “Finite Field Arithmetic”, In: Guide to Elliptic Curve Cryptography, Springer Nature, pp.25-73, 2004.

[16]. D. Hankerson, S. Vanstone, A. Menezes, “Implementation Issues”, In: Guide to Elliptic Curve Cryptography, Springer Nature, pp.205-256, 2004.

[17]. D. Hankerson, S. Vanstone, A. Menezes, “Cryptographic Protocols”, In: Guide to Elliptic Curve Cryptography, Springer Nature, pp.153-204, 2004.

[18]. D. Sravana Kumar, Ch. Suneetha, A. Chandrasekhar, “Encryption of Data Using Elliptic Curve Over Finite Fields”, In: International Journal of Distributed and Parallel Systems (IJDPS), 3 (1) (2012). L19

[19]. Darrel Hankerson,Alfred Menezes and Scott Vanstone, Guide to Elliptic Curve Cryptography, Springer (2004). L13

[20]. Dave K. Kythe, Prem K. Kythe, “Galois Fields" In: Algebraic and Stochastic Coding Theory”, CRC Press, 2012.

[21]. Dave K. Kythe, Prem K. Kythe. “Cyclic Codes”, In: Algebraic and Stochastic Coding Theory, CRC Press, 2012.

[22]. Deeksha Priya Jha, Rashi Kohli and Archana Gupta, “Proposed encryption algorithm for data security using matrix properties”, In: International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH), IEEE, 2016. L5

[23]. E. Karthikeyan. “Survey of elliptic curve scalar multiplication algorithms”. Int. J. Adv. Netw. Appl. 04(02) (2012)

[24]. Elsayed Mohamed and Hassan Elkamchouchi, “Elliptic Curve Cryptography over Gaussian Integers”, International Journal of Computer Science and Network Security (IJCSNS), vol.9 no.1, 2009. pp. 413-416. L32

[25]. Erwin Kreyszig, “Complex Numbers and Their Geometric Representation”, In: Advanced Engineering Mathematics, 10th edition, USA, John Wiley & Son Inc., 2011, pp.608-612. L29

[26]. Erwin Kreyszig, “Linear Algebra. Vector Calculus”, In: Advanced Engineering Mathematics, 10th edition, USA, John Wiley & Son Inc., 2011, pp.255-399.

[27]. Evan Dummit, “Elliptic Curves in Cryptography”, In: Cryptography part 5, Course Note, 2016. L28

[28]. F. Amounas, EL Hassan EL Kinani, “Encryption of Data using Elliptic Curve over Circulant Matrices”, In: International Journal of Electronics Communication and Computer Engineering, vol.4, no.1, pp.61-65,2013.

[29]. F. Guo, W. Susilo, Y. Mu, “Introduction to Security Reduction”, In: Springer Science and Business Media LLC, Springer Cham, 2018, DOI: 10.1007/978-3-319-93049-7.

[30]. G. A. Jones, J.Mary Jones , “Information and Coding Theory”, In: Springer Undergraduate Mathematics Series, 2000.

[31]. Gary L. Mullen and D. Panario, Handbook of Finite Fields, In: Discrete Mathematics and Its Applications, Chapman and Hall/CRC, 2013

[32]. George Stergiopoulos, Miltiadis Kandias, and Dimitris Gritzalis, “Approaching encryption through complex number logarithms”, In: International Conference on Security and Cryptography (SECRYPT), IEEE, 2013. L35

[33]. Gopinath Ganapathy and K. Mani, Maximization of Speed in Elliptic Curve Cryptography Using Fuzzy Modular Arithmetic over a Micro-controller-based Environment, Proceedings of the World Congress on Engineering and Computer Science, vol. 1, (2009). L22

[34]. H. Malepati. “Data Security”, In: Digital Media Processing, ELSEVIER Press, 2010.

[35]. H. Modares, Y. Salem, R. Salleh and M. T. Shahgoli, “A Bit-Serial Multiplier Architecture for Finite Fields Over Galois Fields”, In: Journal of Computer Science, vol.6, no.11, 2010, DOI:10.3844/jcssp.2010.1237.1246.

[36]. H.Z. Liao, Y. Y. Shen. In: “On the elliptic curve digital signature algorithm”. Tunghai Sci. vol. 8 (2006).

[37]. Hamza Touil, Nabil EL Akkad and Khalid Satori, “Text Encryption: Hybrid cryptographic method using Vigenere and Hill Ciphers”, In: International Conference on Intelligent Systems and Computer Vision (ISCV), IEEE, 2020. L3

[38]. Igor E. Shparlinski, Computational and Algorithmic Problems in Finite Fields (Mathematics and its Applications, 88) 1992. L15

[39]. J. Li, Z. Liu, H. Peng, "Security and Privacy in New Computing Environments", In: International Conference on Security and Privacy in New Computing Environments, Springer LNICST, vol. 284, 2019.

[40]. Jessie R. Paragas, Ariel M. Sison, Ruji P. Medina, “A New Variant Of Hill Cipher Algorithm Using Modified S-Box”, In: International Journal of Scientific & Technology Research, vol. 8- no.10, 2019. L8.

[41]. Joan-Josep Climent, F. Ferrández, A. Zamora, “A nonlinear elliptic curve cryptosystem based on matrices”, In: Applied Mathematics and Computation, vol.174, 2006, pp. 150-164, DOI: 10.1016/j.amc.2005.03.032.

[42]. John J. D'Azzo, "Matrix Linear Algebra", In: Automation and Control Engineering, vol. 14, 2003.

[43]. Joppe W. Bos, Marcelo E. Kaihara, Thorsten Kleinjung, “On the Security of 1024-bit RSA and 160-bit Elliptic Curve Cryptography”. In: IACR Cryptol. ePrint Arch. 2009: 389 (2009)

[44]. Jorko Teeriaho, Cyclic Group Cryptography with Elliptic Curves, Brasov (2011). L17

[45]. K. Adinarayanareddy, B. Vishnuvardhan, “A Modified Hill Cipher Based on Circulant Matrices”, In: Procedia Technology, vol. 4, pp.114-118, ELSEVIER Press, 2012.

[46]. K. Agrawal, Anju Gera, "Elliptic Curve Cryptography with Hill Cipher Generation for Secure Text Cryptosystem", In: International Journal of Computer Applications, vol.106, no.1, pp. 18-24, 2014.

[47]. K. H. Rosen, “Number Theory and Cryptography”. In: Discrete Mathematics and its Applications, 7th ed, pp 237-294, McGraw-Hill press, USA, 2011.

[48]. K. J. Liew, H. Kamarulhaili, “Hasse's Theorem and the Statistical Properties of Points on Elliptic Curves Modulo Prime, p”, In: 2nd USM Fellowship Holders Symposium: Injecting Humanistic Values into Our Education, 2011.

[49]. K. Jarvinen, Helsinki and J. Skytta, On Parallelization of High-Speed Processors for Elliptic Curve Cryptography, VLSI Systems, In: IEEE Transaction, vol. 16, issue 9, pp. 1162–1175, August (2008). L20

[50]. K. Prasanthi, “A Detailed Analysis on Encryption of Messages Using Cryptographic Model with Matrices”, In: International Journal of Smart Home, vol. 14, no. 1, 2020, DOI:10.21742/ijsh.2020.14.1.04.

[51]. K. Rabah, “Elliptic Curve ElGamal Encryption and Signature Schemes”. In: Info. Tech. J.(ITJ), vol. 4 (3), pp 299-306, Pakistan, 2005.

[52]. K. Thiagarajan, P. Balasubramanian, “Encryption and decryption algorithm using algebraic matrix approach”, IOP Conf. Series: Journal of Physics: Conf. Series 1000 (2018) 012148, DOI:10.1088/1742-6596/1000/1/012148.

[53]. K.V. Pramod. “A Cryptosystem Using the Concepts of Algebraic Geometric Code”, In: Journal of Computer Science, 2010, vol. 6(3). pp. 244-249. DOI:10.3844/jcssp.2010.244.249.

[54]. Khalil Hariss, Maroun Chamoun and Abed Ellatif Samhat, “Fully Homomorphic Encryption Scheme Based On Complex Numbers”, In: Advances in Science, Technology and Engineering Systems Journal, Vol. 4, No. 5, pp. 30-38 (2019). L36

[55]. L. C. Washington, “Elliptic Curves”, In: Number Theory and Cryptography, Second Edition, Taylor & Francis Group, LLC, 2008.

[56]. L. D. Singh and K. M. Singh, “Implementation of Text Encryption using Elliptic Curve Cryptography”, In: International Multi-Conference on Information Processing (IMCIP-2015), Procedia Computer Science vol.54, pp.73-82, ELSEVIER Press, 2015.

[57]. L. Keliher. “Cryptanalysis of a Modified Hill Cipher”, International Journal of Computer and Network Security, Vol. 2, No. 7, July 2010, pp. 122-126.

[58]. Lawrence C. Washington, "The Basic Theory", In: Elliptic Curves: Number Theory and Cryptography, pp. 9-71, Chapman and Hall/CRC Press, 2008.

[59]. Lawrence C. Washington, “Elliptic Curve Cryptography”, In: Elliptic Curves: Number Theory and Cryptography, pp. 169-187, Chapman and Hall/CRC Press, 2008.

[60]. Lawrence C. Washington, “Elliptic Curves over Finite Fields”, In: Elliptic Curves: Number Theory and Cryptography, pp. 95-139, Chapman and Hall/CRC Press, 2008.

[61]. Lawrence C. Washington, “The Discrete Logarithm Problem”, In: Elliptic Curves: Number Theory and Cryptography, pp. 143-166, Chapman and Hall/CRC Press, 2008.

[62]. Lawrence C. Washington, Elliptic Curves: Number Theory and Cryptography, (Second Edition), Taylor & Francis Group (2008). L14

[63]. Little, “Appendix: Introduction to Matrices and Matrix Operations”, In: Environmental Fate and Transport Analysis with Compartment Modeling, CRC Press, 2012.

[64]. Lo’ai Tawalbeh, Moad Mowafi and Walid Aljoby, Use of Elliptic Curve Cryptography for Multimedia Encryption, IET Information Security, vol. 7, issue 2, pp. 67–74, (2012). L25

[65]. M. Amara and A. Siad, Elliptic Curve Cryptography and its Applications, In: 7th International Workshop on Systems, Signal Processing and their Applications, pp. 247–250, May (2011). L21

[66]. M. D. L. Siahaan, A. P. U. Siahaan, “Application of Hill Cipher Algorithm in Securing Text Messages”, In: International journal for innovative research in multidisciplinary field, vol. 4, n0. 10, 2018, DOI:10.31227/osf.io/n2kdb.

[67]. M. Eisenberg. “Hill Ciphers and Modular Linear Algebra”, Mimeographed notes, 1999, pp. 1-19.

[68]. M. Musson, “Attacking the elliptic curve discrete logarithm problem”, In: Master Thesis of Science (Mathematics and Statistics) Acadia University (2006)

[69]. M. Safieh, J. P. Thiers, J. Freudenberger, “A compact coprocessor for the elliptic curve point multiplication over Gaussian integers”, Electronics, vol. 9, no. 12, 2020. L33

[70]. M. Toorani, A. Falahati. “A secure cryptosystem based on affine transformation”, In: Security and Communication Networks, pp. 207-215, 2011.

[71]. Md. S. Rahman, Md. S. Hossain, En. H. Rahat, Deb. Roy Dipta et al. “Efficient Hardware Implementation of 256- bit ECC Processor Over Prime Field”, In: International Conference on Electrical, Computer and Communication Engineering (ECCE 2019), IEEE, DOI: 10.1109/ECACE.2019.8679184.

[72]. Megha Kolhekar and Anita Jadhav, Implementation of Elliptic Curve Cryptography on Text and Image, In: International Journal of Enterprise Computing and Business Systems, vol. 1, issue 2, July (2011). L27

[73]. Michael Oberguggenberger, Alexander Ostermann, Analysis for Computer Scientists, Springer, 2011. L30

[74]. Michal Rosing, Implementing Elliptic Curve Cryptography, Manning Publications, 1998. L16

[75]. Mohammad Al-Khatib, Wafaa Saif., "Improved Software Implementation for Montgomery Elliptic Curve Cryptosystem", In: Computers, Materials & Continua, vol. 70, no.3, pp. 4847-4865, Tech Science Press, 2022, DOI:10.32604/cmc.2022.021483.

[76]. N. N. Hla and T. M. Aung, "Attack Experiments on Elliptic Curves of Prime and Binary Fields," In: Emerging Technologies in Data Mining and Information Security (IEMIS 2018), Springer AISC vol.755, pp. 667-683 2017, DOI:10.1007/978-981-13-1951-8_60.

[77]. N. N. Hla and T. M. Aung, "Computing and Analysis of Residue Matrices over Complex Plane for Cryptographic Applications", In: International Conference on Computing and Information Technology (ICCIT-1441), IEEE, 10.1109/ICCIT-144147971.2020.9213722.

[78]. N. N. Hla and T. M. Aung, "Experiments on implementation of elliptic curve arithmetic over complex fields using java BigInteger Class", In: Journal of Communications, vol 14, no.4, pp. 293-300, 2019, DOI: 10.12720/jcm.14.4.293-300.

[79]. N. S. Yanofsky, M. A. Mannucci, "Complex Vector Spaces". In: Quantum Computing for Computer Scientists, pp 29-66, Cambridge University Press, 2008.

[80]. N. S. Yanofsky, M. A. Mannucci, "Basic Quantum Theory". In: Quantum Computing for Computer Scientists, pp 103-132, Cambridge University Press, 2008.

[81]. N. S. Yanofsky, M. A. Mannucci, "Complex Numbers". In: Quantum Computing for Computer Scientists, pp 7-15, Cambridge University Press, 2008.

[82]. N. S. Yanofsky, M. A. Mannucci, "Cryptography". In: Quantum Computing for Com-puter Scientists, pp 262-283, Cambridge University Press, 2008.

[83]. N. Sharma, S. Chirgaiya, "A Novel Approach to Hill Cipher", In: International Journal of Computer Applications, vol. 108. no. 11, 2014, DOI:10.5120/18958-0285.

[84]. N.N. Hla, T.M. Aung. “Implementation of finite field arithmetic operations for large prime and binary fields using java BigInteger class”. Int. J. Eng. Res. Technol. (IJERT), 6(08), 2017, DOI: 10.17577/IJERTV6IS080209.

[85]. Neal Koblitz, “Elliptic Curve Cryptosystems”, In: Mathematics of Computation, 48 (177) (1987), pp. 203-209. L11

[86]. Neal Koblitz, Alfred Menezes, Scott Vanstone, “The State of Elliptic Curve Cryptography”, In: Designs, Codes and Cryptography, 19 (2–3) (2000), pp. 173-193. L12

[87]. NIST, Recommended Elliptic Curves for Federal Government Use, 1999

[88]. Noson S. Yanofsky, Mirco A. Mannucci, Quantum computing for computer scientists, Cambridge University Press, 2008. L31

[89]. P. Bhattacharya, M. Debbabi, H. Otrok. “Improving the Diffie-Hellman Secure Key Exchange”, In: International Conference on Wireless Networks, Communications and Mobile Computing, 2005, IEEE, DOI: 10.1109/WIRLES.2005.1549408.

[90]. P. L. Sharma, M. Rehan, “On Security of Hill Cipher using Finite Fields”, In: International Journal of Computer Applications, vol. 71 no.4, 2013. DOI: 10.5120/12348-8637. L7

[91]. R. Balamurugan, V. Kamalakannan, D. Rahul Ganth and S. Tamilselvan, Enhancing Security in Text Messages Using Matrix based Mapping and ElGamal Method in Elliptic Curve Cryptography, International Conference on Contemporary Computing and Informatics, IEEE, pp. 103–106, (2014). L26

[92]. R. Lidl and H. Niederreiter, “Introduction to Finite Field Arithmetic and their Applications”, Cambridge University Press, 1986.

[93]. S. Maria Celestin Vigila, K. Muneeswaran, “Implementation of Text based Cryptosystem using Elliptic Curve Cryptography”, In: International Conference on Advanced Computing IEEE (2009), pp. 82-85. L18

[94]. S. Martín, P. Morillo, J. L. Villar. “Computing the order of points on an elliptic curve modulo N is as difficult as factoring N”, In: Applied Mathematics Letters, vol.14, no. 3, pp.341-346, ELSEVIER Press, 2001.

[95]. S. Pote, B. K. Lande, P. M. Mammen, “Elliptic Curve arithmetic over extension field to intensify security and privacy”, In: International Conference on Wireless

Communications, Signal Processing and Networking (WiSPNET 2016), IEEE, DOI: 10.1109/WiSPNET.2016.7566470.

[96]. Scott A. Vansfone, Elliptic Curve Cryptography-The Answer to Strong, Fast Public-Key Cryptography for Securing Constrained Environments, Information Security Technical Report, vol. 2, no. 2, pp. 78–87, (1997). L23

[97]. T. K. Moon. “Cyclic Codes, Rings, and Polynomial", In: Error Correction Coding, pp. 113-170, Wiley, 2005.

[98]. T. M. Aung, K. H. Myint, N. N. Hla “A Data Confidentiality Approach to SMS on Android”, In: International Conference on Intelligent Computing & Optimization (ICO 2018), Springer AISC vol. 866, pp. 505-514, 10.1007/978-3-030-00979-3_53.

[99]. T. M. Aung, N. N. Hla. “A Complex Number Approach to Elliptic Curve Cryptosystem over Finite Fields: Implementation and Experiments”, In: 2019 International Conference on Computer Communication and Informatics (ICCCI) , IEEE, pp. 221-228. 2019, DOI: 10.1109/ICCCI.2019.8821887.

[100]. T. M. Aung, N. N. Hla. “A Complex Polyalphabetic Cipher Technique: Myanmar Polyalphabetic Cipher”, In: International Conference on Computer Communication and Informatics (ICCCI), IEEE, pp. 229-237, 2019, DOI: 10.1109/ICCCI.2019.8821797.

[101]. T. M. Aung, N. N. Hla. “A New Technique to Improve the Security of Elliptic Curve Encryption and Signature Schemes”, In: Future Data and Security Engineering (FDSE 2019), Springer LNCS vol. 11814, pp.371-382, 2019, DOI: 10.1007/978-3-030-35653-8_25.

[102]. T.M. Aung, N.N. Hla. Implementation of elliptic curve arithmetic operations for prime field and binary field using java BigInteger class. Int. J. Eng. Res. Technol. (IJERT), 6(08), 2017, 10.17577/IJERTV6IS080211.

[103]. Tutorialspoint, “Cryptosystems”, https://www.tutorialspoint.com/ cryptography/origin_of_cryptography.htm.

[104]. Tutorialspoint, “Origin of Cryptography”, https://www.tutorialspoint.com/ cryptography/origin_of_cryptography.htm.

[105]. Victor S. Miller, “Use of Elliptic Curves in Cryptography”, In: Advances in Cryptology-CRYPTO’85 Proceedings Springer, 218 (2000), pp. 417-426. L10

[106]. W. J. Caelli, E. P. Dawson, and S. A. Rea. “PKI, elliptic curve cryptography, and digital signatures”.In: Computers and Security vol.18. no.1, pp. 47-66 (1999).

[107]. W. Stallings, "Classical Encryption Techniques", In: Cryptography and Network Security: Principles and Practice, 6th Edition, pp. 27-60, Pearson Press, 2014.

[108]. W. Stallings, "Cryptography and Network Security: Principles and Practice", Prentice Hall, 1999. L24

[109]. W. Stallings, “Basic Concepts in Number Theory and Finite Fields”, In: Cryptography and Network Security: Principles and Practice, 6th Edition, pp.85-129, Pearson Press, 2014.

[110]. W. Stallings, “More Number Theory”, In: Cryptography and Network Security: Principles and Practice, 6th Edition, pp. 231-252, Pearson Press, 2014.

[111]. W. Stallings, “Other Public Key Cryptosystems”, In: Cryptography and Network Security: Principles and Practice, 6th Edition, pp. 287-312, Pearson Press, 2014.

[112]. Walid W. Souror, Mohamed Fouad, Ali E. Takieldeen. "Hybrid Security Enhancement of ECC with Side Channel and Sign Fault Attack Countermeasures", In: 2022 International Telecommunications Conference (ITC-Egypt), IEEE, 2022, DOI: 10.1109/ITC-Egypt55520.2022.9855769.

[113]. Wanarat Juraphanthong and Suradet Jitprapaikulsarn, “An asymmetric cryptography using gaussian integers”, In: Engineering and Applied Science Research, vol.47, n0.02, 2000. L34

[114]. Wikipedia, Hasse's theorem on elliptic curves, https://en.wikipedia.org

[115]. William Easttom, “Modern Cryptography”, In: Applied Mathematics for Encryption and Information Security, Springer, 2021.

[116]. William P. Wardlaw, “The RSA Public Key Cryptosystem”, In: Coding Theory and Cryptography, pp.101-123, Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-59663-6_6. L9.

[117]. Y. S. Yeh, T. C. Wu, C. C. Chang and W. C. Yang., “A new cryptosystem using matrix transformation”, In: International Carnahan Conference on Security Technology, IEEE, 1991. L6

[118]. Yoshito Kanamori, Seong-Moo Yoo, “Quantum Computing: Principles and Applications”, In: Journal of International Technology and Information Management, vol.29, no.2,2020.

APPENDICES

APPENDIX (A). Experiment for Known-Plain text Attack on Hill Cipher

Ciphertext: FAGQQ ILABQ VLJCY QULAU STYTO JSYTO JSDJJ PODFS ZNLUH KMON

It is presumed that a 2×2 Hill cipher was used to encrypt this communication and that the transmission contains a plain text “crib”.The message starts with “a crib”.

ac	ri
[1, 3]	[18, 9]
[6, 1]	[7, 17]
FA	GQ

Find either solve for the key or the key inverse. To solve for the key,

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 6 \\ 1 \end{bmatrix}$$

and

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 18 \\ 9 \end{bmatrix} = \begin{bmatrix} 7 \\ 17 \end{bmatrix}$$

To solve for key inverse,

$$\begin{bmatrix} e & f \\ g & h \end{bmatrix} \begin{bmatrix} 6 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

and

$$\begin{bmatrix} e & f \\ g & h \end{bmatrix} \begin{bmatrix} 7 \\ 17 \end{bmatrix} = \begin{bmatrix} 18 \\ 9 \end{bmatrix}$$

Solve for the key:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 6 \\ 1 \end{bmatrix}$$

$$a + 3b = 6$$

$$c + 3d = 1$$

and $\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} 18 \\ 9 \end{bmatrix} = \begin{bmatrix} 7 \\ 17 \end{bmatrix}$ represents

$$18a + 9b = 7$$

$$18c + 9d = 17$$

Now, The following linear congruence mod 26.

$$\begin{cases} a + 3b = 6 \\ 18a + 9b = 7 \end{cases} \quad \text{and} \quad \begin{cases} c + 3d = 1 \\ 18c + 9d = 17 \end{cases}$$

Solve the pair of congruence: $\begin{cases} a + 3b = 6 \\ 18a + 9b = 7 \end{cases}$ first.

To eliminate an unknown, multiply congruence 1 by 3

$$\begin{cases} 3a + 9b = 18 \\ 18a + 9b = 7 \end{cases}$$

and subtract congruence 2 from congruence 1.

$$-15a = 11$$

Modulo 26, -15 is 11

$$11a = 11$$

Divide by 11 to obtain a

$$a = 1$$

Now substitute this in congruence 1

$$1 + 3b = 6$$

$$3b = 5$$

The multiplicative inverse of 3 is 9 modulo 26.

$$b = 3^{-1} \times 5 = 9 \times 5 = 45 = 19 \bmod 26$$

So, the key looks like

$$\begin{bmatrix} 1 & 19 \\ c & d \end{bmatrix}$$

Now solve the system $\begin{cases} c & +3d & = & 1 \\ 18c & +9d & = & 17 \end{cases}$

$$\begin{cases} 3c & +9d & = & 3 \\ 18c & +9d & = & 17 \end{cases}$$

$$15c = 14$$

$$c = (15)^{-1} \times 14 = 7 \times 14 = 98 = 20 \; mod \; 26$$

$$20 + 3d = 1$$

$$3d = -19 = 7 \bmod 26$$

$$d = 3^{-1} \times 7 = 9 \times 7 = 63 = 11 \bmod 26$$

The key is $\begin{bmatrix} 1 & 19 \\ 20 & 11 \end{bmatrix}$.

APPENDIX (B). Experiments for Baby-Step Giant-Step Attack on Elliptic Curve

Prime Field. Let an elliptic curve be $E: y^2 = x^3 + 5x + 4$ over $GF(13)$, $P = (0,2)$ and $Q = (6,4)$ .It is assumed that an integer scalar *k* is solved such that $Q = [k]P$ by using *Baby-Step, Giant-Step method*. *P* has order 17. First compute. The points iP for $1 \le i < 5$ are:

$(0,2), (4,6), (10,1), (6,9)$.

Then calculate $Q - jmP$ for $j = 0,1,2,3,\ldots$ and obtain:

$(6,4), (11,8), (10,1), (4,7), (1,6) \ldots$

at which point is stopped since this third point matches $3P$. Since $j = 2$ yielded the match, and finally:

$$(6, 4) = (3 + 2.5)P = 13P.$$

Therefore $k = 13$.

Binary Field. Let an elliptic curve be $E: y^2 + xy = x^3 + g^{11}x + g^{13}$ over GF(2^4), $P = (g^9, 1)$ and $Q = (g^6, g^6)$. It is assumed that an integer scalar *k* is solved such that $Q = [k]P$ by using *Baby-Step, Giant-Step method*. *P* has order 11.First compute: $m = \left| \sqrt{11} \right| = 4$. The points iP for $1 \le i < 4$ are:

$(g^9,1)$, (g^{12},g^4), $(g^6,0)$.

Then calculate $Q-jmP$ for $j=0,1,2,3,4,\ldots$ and obtain:

(g^6,g^6), $(g^{14},1)$, O, (g^{14},g^3), $(g^6,0)$.

at which point is stopped since this fifth point matches $3P$. Since $j=4$ yielded the match, and finally:

$(g^6,g^6)=((3+4.4)\bmod 11)P=8P$.

Therefore $k=8$.

APPENDIX (C). Experiments for Pollard's rho Attack on Elliptic Curve

Prime Field. Let an elliptic curve be $E: y^2=x^3+5x+4$ over $GF(13)$, $P=(0,2)$ and $Q=(6,4)$. It is assumed that an integer scalar k is solved such that $Q=[k]P$ by using *Pollard's rho method*. The point P has prime order 17.Choose $a,b\in[0,17]$ uniformly at random, compute $R=[a]P+[b]Q$ and keep the triple (a,b,R) in the memory until meet an another triple (a',b',R') such that $R=R'$ or $R=-R'$. Table C.1 shows computing data used for Pollard's rho attack on $E: y^2=x^3+5x+4$ over $GF(13)$. $[5]P+[12]Q=[2]P+[7]Q$ is solved. Then $k=(5-2)(7-12)^{-1}\bmod 17$; $k=3(-5)^{-1}\bmod 17$; $k=3.10\bmod 17$; Hence $k=13$.

Table C.1. Data Analysis for Pollard's Rho Attack on $E: y^2=x^3+5x+4$ over $GF(13)$.

[a]	[b]	R=[a]P + [b]Q
5	**12**	**(11,8)**
3	8	(8,6)
10	4	(2,3)
6	11	(6,4)
2	**7**	**(11,8)**
1	15	(11,5)

Binary Field. Let an elliptic curve be $E: y^2 + xy = x^3 + g^{11}x + g^{13}$ over $GF(2^4)$, $P = (g^9,1)$ and $Q = (g^6, g^6)$. It is assumed that an integer scalar *k* is solved such that $Q = [k]P$ by using *Pollard's rho method*. The point *P* has prime order 11.Choose $a,b \in [0,11]$ uniformly at random, compute $R = [a]P + [b]Q$ and keep the triple (a,b,R) in the memory until meet an another triple (a',b',R') such that $R = R'$ or $R = -R'$. Table C.2 shows computing data used for Pollard's rho attack on $E: y^2 + xy = x^3 + g^{11}x + g^{13}$ over $GF(2^4)$. $[10]P + [5]Q = [7]P + [4]Q$ is solved. Then $k = (10-7)(4-5)^{-1} \bmod 11$; $k = 3.10 \bmod 11$; Hence $k = 8$.

Table C.2. Data Analysis for Pollard's Rho Attack on $E: y^2 + xy = x^3 + g^{11}x + g^{13}$ over $GF(2^4)$

[a]	[b]	R=[a]P + [b]Q
10	**5**	**$(g^{13},1)$**
8	3	(g^9, g^7)
4	10	(g^{14}, g^3)
5	6	(g^{12}, g^6)
7	**4**	**$(g^{13},1)$**
2	7	$(g^6,0)$

APPENDIX (D). Experiments for Pohlig-Hellman Attack on Elliptic Curve

Prime Field. Let an elliptic curve be $E: y^2 = x^3 + 77x + 28$ over $GF(157)$, $P = (9,115)$ and $Q = (2,70)$. It is assumed that an integer scalar *k* is solved such that $Q = [k]P$ by using *Pohlig Hellman method*. The order *N* of point *P* is 162. The prime factorization of *N* is 2.3^4. Compute *k* mod 2, and mod 81, then recombine them to obtain *k* mod 162 using the Chinese Remainder Theorem.

k mod 2. Compute $T = \{(24,0)\}$.

Since $\frac{N}{2}.Q = (24,0) = 1.(\frac{N}{2}.P)$, and $k_0 = 1$.

Therefore $k \equiv 1 (\bmod 2)$.

k mod 81. Compute $T = \{(57,41), (5,99), (57,116), O\}$.

Since $\frac{N}{3}.Q = (57,41) = 1.(\frac{N}{3}.P)$, and $k_0 = 1$.

Therefore $Q_1 = Q - 1.P = (5,99)$.

Since $\frac{N}{9}.Q_1 = O = 0.(\frac{N}{3}.P)$, and $k_1 = 0$.

Therefore $Q_2 = Q_1 - 0.3.P = Q_1$.

Since $\frac{N}{27}.Q_2 = (57,116) = 2.(\frac{N}{3}.P)$, and $k_2 = 2$.

Therefore $Q_3 = Q_2 - 2.9.P = (57,41)$.

Since $\frac{N}{81}.Q_3 = (57,116) = 2.(\frac{N}{3}.P)$, and $k_3 = 2$.

Therefore $k = 1 + 0.3 + 2.9 + 2.27 \equiv 73 \pmod{81}$.

Now the simultaneous congruence have been obtained:

$k \equiv 1 \pmod 2$

$k \equiv 73 \pmod{81}$.

Then k = 73 is obtained using the Chinese Remainder theorem to recombine simultaneous congruence and they are as follows:

$M_1 = 162 / 2 = 81.$

$y_1 = M_1^{-1} \bmod 2 = 1.$

$M_2 = 162 / 81 = 2.$

$y_2 = M_2^{-1} \bmod 81 = 41.$

$k = 1.(81).1 + 73.(2).41 (\bmod 162) = 73.$

Binary Field. Let an elliptic curve be $E : y^2 + xy = x^3 + g^{11}x + g^{13}$ over $GF(2^4)$, P = (g^2, g^2) and Q = (g^6, g^6). It is assumed that an integer scalar k is solved such that $Q = [k]P$ by using *Pohlig Hellman method*. The order N of point P is 22. The prime factorization of N is 2.11. Compute k mod 2, and mod 11, then recombine them to obtain k mod 22 using the Chinese Remainder Theorem.

k mod 2.Compute $T = \{O\}$.

Since $\frac{N}{2}.Q = O = 0.(\frac{N}{2}.P)$, and $k_0 = 0$.

Therefore $k \equiv 0(\text{mod } 2)$.

k mod 11. Compute $T = \{(g^{13}, g^{6})\}$.

Since $\frac{N}{11}.Q = (g^{13}, g^{6}) = 4.(\frac{N}{11}.P)$, and $k_0 = 4$.

Therefore $k \equiv 4(\text{mod} 11)$.

Now the simultaneous congruence have been obtained:

$k \equiv 0 \text{ (mod } 2)$

$k \equiv 4 \text{ (mod } 11)$.

Then $k = 4$ is obtained using the Chinese Remainder theorem to recombine simultaneous congruence and they are as follows:

$M_1 = 22/2 = 11$.

$y_1 = M_1^{-1} \bmod 2 = 1$.

$M_2 = 22/11 = 2$.

$y_2 = M_2^{-1} \bmod 11 = 6$.

$k = 0.(11).1 + 4.(2).6(\text{mod } 22) = 4$.