Experiments on Implementation of Elliptic Curve Arithmetic over Complex Fields Using java BigInteger Class

Ni Ni Hla and Tun Myat Aung
University of Computer Studies, Yangon and 11411, Myanmar
Emails: ni2hla@ucsy.edu.mm, tma.mephi@gmail.com

Abstract —Elliptic curve cryptosystems are nowadays widely used in public communication channels for network security. Their security depends on the complexity of solving the elliptic curve discrete logarithm problem. But, there are several general attacks in them. Elliptic curve arithmetic is implemented over complex fields to improve the security of elliptic curve cryptosystems. This paper begins the characteristics of elliptic curve cryptosystems and their security services. Then we discuss finite field arithmetic and its properties, prime field arithmetic, binary field arithmetic and complex number arithmetic, and elliptic curve arithmetic over prime field and binary field. This paper proposes how to implement complex number arithmetic under prime field and binary field using java BigInteger class and we implement elliptic curve arithmetic and elliptic curve cryptosystems using complex numbers over prime field and binary field and discuss the experiments that got from our implementations.

Index Terms—binary field, complex number, elliptic curve, experiments, implementation, prime field

I. Introduction

Elliptic Curve Cryptosystem (ECC) is a public key cryptosystem. In ECC every entity connecting in the public communication channel generally has a pair of keys, a public key and a private key to perform cryptographic transformations such as encryption, decryption, signing, verification and authentication. The private key must be kept secret but the corresponding public key is distributed to all entities connecting in the public communication channel. ECC provides the security services such as data confidentiality, data integrity, message authentication, entity authentication, non-repudiation, and public key distribution.

Nowadays, ECC becomes a major role in the industry of information and network security technology. It becomes the industrial standard as a consequence of an increase in speed and a decrease in power consumption during implementation as a result of less memory usage and smaller key sizes. Its security depends on the complexity of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP). The ECDLP is considered as a complex problem, but attackers are still making an effort to achieve a difficult task on ECDLP until now. Several attacks have been created, experienced and analyzed by experts in mathematics and computer science over the years, to discover flaws in ECDLP. Some attacks have done successfully, but others have not.

This paper proposes how to implement complex number arithmetic under prime field and binary field using java BigInteger class and we implement elliptic curve arithmetic and elliptic curve cryptosystems using complex numbers over prime field and binary field. The structure of this paper is as follows. The section 2 includes finite field arithmetic and their properties, prime field arithmetic, binary field arithmetic and complex number arithmetic. In section 3, we discuss elliptic curve arithmetic over prime and binary fields, its geometric properties, the ECDLP and its properties. The section 4 describes how to implement arithmetic operations of complex numbers under prime field and binary field and how to implement ECC using complex numbers under prime field and binary field. Finally, the section 5 discusses the arithmetic properties of complex numbers and the security of ECC implemented over finite fields of complex numbers.

II. Finite Field Arithmetic

A. Introduction

A finite field, generally denoted by F, is a field that contains a finite number of elements. A finite field can be applied to the rational number system, the real number system and the complex number system. It contains a finite number of elements together with two arithmetic operations: addition denoted by the symbol + and multiplication denoted by the symbol $\cdot$ that satisfy the following arithmetic properties [5]:

- The Law of Commutativity: $x+y=y+x$; $x.y=y.x$, 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: $(x+y)z=x.z+y.z$, 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 $z \in F$.

Manuscript received August 20, 2018; revised March 2, 2019.
Corresponding author email: ni2hla@ucsy.edu.mm
doi:10.12720/jcm.14.4.293-300

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 $z.z^{-1} = 1$.

Galois open that the elements in the field to be finite and the number of elements should be p^m, where p is a prime number called the characteristic of the field and m is a positive integer. The finite fields are generally called Galois fields and also signified by $GF(p^m)$. When m = 1, then the field $GF(p)$ is called a prime field. When $m \geq 2$, then the field $GF(p^m)$ is called an extension field. The number of elements in a finite field is called the order of the field. Any two fields are isomorphic when their orders are the same [1].

Finite field F has additive group that performs on the arithmetic addition operation as well as multiplicative group that performs on the arithmetic multiplication operation. However, the subtraction of field elements is defined in the expressions of addition operation. For instance, let $x, y \in F$, $x - y$ is defined as $x + (-y)$, where $-y$ is the additive inverse of y. Correspondingly, the division of field elements is defined in the expression of multiplication operation. For instance, let $x, y \in F$ with $y \neq 0$, x / y is defined as $x.y^{-1}$, where y^{-1} is the multiplicative inverse of y [1].

B. Prime Field

A finite field of prime order p is called prime field denoted by $GF(p)$. It contains a set of integer elements modul p, $\{0,1,2,\ldots, p-1\}$ with additive and multiplicative groups that performed modulo p. For any integer x, $x \bmod p$ refers to the integer remainder r that obtained upon dividing x by p. This operation is called reduction modulo p. In this case, the remainder r is the unique integer element between 0 and $p-1$, i.e. $0 \leq r \leq p-1$. The arithmetic operations of elements over $GF(p)$ are performed as the following example (1) [1].

Example 1. (P*rime Field- $GF(p)$*) The elements of $GF(7)$ are {0,1,2,...,6}. The followings demonstrate arithmetic operations of elements in $GF(7)$.

- Addition: 3 + 5 = 1 since 8 mod 7 = 1.
- Subtraction: 3 − 5 = 5 since −2 mod 7 = 5.
- Multiplication: 3 × 5 = 1 since 15 mod 7 = 1.
- Inversion: $5^{-1} = 3$ since $5 \times 3 \bmod 7 = 1$.
- Division: $3 \div 5 = 2$ since $3 \times 5^{-1} = 9 \bmod 7 = 2$.

C. Binary Field

A finite field of order 2^m is called binary field denoted by $GF(2^m)$. It also refers to the finite field with characteristic-two. The elements of $GF(2^m)$ can be constructed by applying a polynomial basis representation defined by the equation (1). In this case, the elements of $GF(2^m)$ are the binary polynomials with degree at most m −1.

$$GF(2^m) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \ldots + a_2x^2 + a_1x + a_0, a_i \in \{0,1\} \quad (1)$$

f (x) is defined as an irreducible binary polynomial with degree *m* if *f(x)* cannot be factored as a product of binary polynomials with degree less than *m*. Let *a(x)* and *b(x)* be the elements of $GF(2^m)$. They are the binary polynomials with degree at most *m*−1. The addition of elements in $GF(2^m)$ refers to the addition of binary polynomials, that is, $a(x) \oplus b(x)$. The multiplication of elements in $GF(2^m)$ refers to the expression $a(x) \times b(x) \bmod f(x)$. Let $c(x) = a(x) \times b(x)$ and *c(x)* be a binary polynomial with degree more than *m*. The result of the expression *c(x)* mod *f(x)* refers to the unique remainder polynomial *r(x)* with degree less than *m* that obtained upon the division of *c(x)* by *f(x)*; this operation is called *reduction modulo f(x).* The division of elements in $GF(2^m)$ refers to the expression $a(x)/b(x) \bmod f(x)$. The division of elements in $GF(2^m)$ is calculated as the expression $a(x) \times b(x)^{-1} \bmod f(x)$. The arithmetic operations of elements in $GF(2^m)$ are performed as the following example (2) [1].

Example (2). (B*inary Field -* $GF(2^m)$) The elements of $GF(2^m)$ are generated by the reduction polynomial $f(x) = x^3 + x + 1$.The period of the reduction polynomial $f(x) = x^3 + x + 1$ is $2^3 - 1 = 7$. Therefore, there are 8 elements from 0 to 7 in $GF(f(x))$. The elements of $GF(f(x))$ are represented by 8 binary polynomials of degree at most 2 as shown in Table I.

TABLE I. BINARY AND POLYNOMIAL REPRESENTATIONS

Binary Polynomial	Binary Polynomial	Binary Polynomial	**Binary** Polynomial
000	001	010	011
0	1	x	$x+1$
100	101	110	111
x^2	x^2+1	x^2+x	x^2+x+1

The followings demonstrate arithmetic operations of elements in $GF(f(x))$.

- Addition: $3 + 5 = 6$ since $(x+1) \oplus (x^2+1) = x^2 + x$.
- Subtraction: $3 - 5 = 6$ since $(x+1) \oplus (x^2+1) = x^2 + x$.
- Multiplication: $3 \times 5 = 4$ since $(x+1) \times (x^2+1) = x^3 + x^2 + x + 1$ and $(x^3 + x^2 + x + 1) \bmod f(x) = x^2$.

- Inversion:
  $5^{-1} = 2.$
  Since
  $(x^2+1) \times x \bmod f(x) = 1.$
  Therefore $(x^2+1)^{-1} = x$.
- Division:
  $3/5 = 6.$
  Since
  $(x+1) \times (x^2+1)^{-1} = (x+1) \times x \bmod f(x) = x^2 + x$.

D. Complex Field

A finite field 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*. 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 [4]. It is demonstrated in Fig. (1).

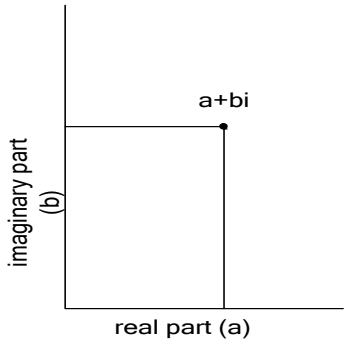


Fig. 1. Complex plane

The following rules are applied for addition, subtraction, multiplication, division, reciprocal and scalar multiplication that are the arithmetic operations of complex numbers over finite field.

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 (2) [4].

$$x + y = (a_1 + a_2) + (b_1 + b_2)i \,. \tag{2}$$

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 (3) [4].

$$x - y = (a_1 - a_2) + (b_1 - b_2)i \,. \tag{3}$$

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) [4].

$$x.y = (a_1 a_2 - b_1 b_2) + (a_1 b_2 + a_2 b_1)i \tag{4}$$

Reciprocal. The reciprocal of a nonzero complex number $z = a + bi$ is defined by the equation (5) [4].

$$\frac{1}{z} = z^{-1} = \frac{a}{a^2 + b^2} - \frac{b}{a^2 + b^2} i \tag{5}$$

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 (6) [4].

$$\frac{x}{y} = x.y^{-1} \tag{6}$$

Scalar Multiplication. The multiplication of a complex number $z = a + bi$ and the scalar integer k is defined by the equation (7) [4].

$$k.z = k.a + k.bi \tag{7}$$

Example (3). (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:
  $x + y = 3 + 3i$ since $(1+2) \bmod 7 + ((2+1) \bmod 7)i$.
- Subtraction:
  $x - y = 6 + 1i$ since $(1-2) \bmod 7 + ((2-1) \bmod 7)i$.
- Multiplication:
  $x.y = 5i$ since $(1.2 - 2.1) \bmod 7 + ((1.1 + 2.2) \bmod 7)i$
- Inversion:
  $y^{-1} = 6 + 4i$ since $(\frac{2}{4+1}) \bmod 7 + (-\frac{1}{4+1} \bmod 7)i$.
- Division: $\frac{x}{y} = 5 + 2i$ since $(1+2i) \times (6+4i)$.
- Scalar Multiplication: $5.x = 5 + 3i$ since $(5.1) \bmod 7 + ((5.2) \bmod 7)i$.

Table II 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 II. 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). (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 followings 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:
  $\frac{x}{y} = 1i$ since $(1 + gi) \times (g + 1i)^{-1} = 1i$.
- Scalar Multiplication: $5x = 5 + 1i$ since $5.x = g^6(1 + gi) = g^6 + 1i$.

III. Elliptic Curve Arithmetic

A. Introduction

The elliptic curve over finite field $E(GF)$ is a cubic curve defined by the general Weierstrass equation (8) over GF where $a_i \in GF$ and GF is a finite field [2].

$$y^2 + a_1 xy + a_3 y = x^3 + a_2 x^2 + a_4 x + a_6 \quad (8)$$

Elliptic curves are driven from the general Weierstrass equation (8). The elliptic curve $E(GF(p))$ is determined by the equation (9) [2]:

$$y^2 = x^3 + ax + b \quad (9)$$

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)

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 $E(GF(2^m))$ over $GF(2^m)$. The elliptic curve $E(GF(2^m))$ is determined by the equation (10) [1]:

$$y^2 + xy = x^3 + ax^2 + b \quad (10)$$

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 group of points on an elliptic curve that is used in the construction of elliptic curve cryptosystems. 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 Fig. (2). 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 Fig (3). 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.

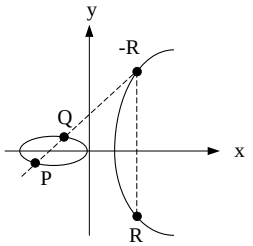


Fig. 2. Addition ($R = P + Q$)

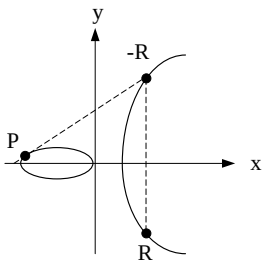


Fig. 3. Doubling ($R = P + P$)

B. Elliptic Curve Arithmetic Over $GF(p)$

The followings are algebraic methods for the addition of two points on $E(GF(p))$ and the doubling of a point on $E(GF(p))$ [2].

- $P + O = O + P = P$ and $P + (-P) = O$ for all $P \in E(GF(p))$. If $P = (x, y) \in E(GF(p))$, the point $(x, -y)$ is signified by (-P) that is called the inverse of P. O is the point at infinity serving as additive identity.

- (Point Addition). Let $P,Q \in E(GF(p))$, $P=(x_1,y_1)$ and $Q=(x_2,y_2)$ where $P \neq \pm Q$. 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 $\lambda=(y_2-y_1)/(x_2-x_1)$.
- (Point Doubling). 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 $\lambda=(3x_1^{\,2}+a)/2y_1$.

C. Elliptic Curve Arithmetic Over $GF(2^m)$

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

- $P+O=O+P=P$ and $P+(-P)=O$ for all $P \in E(GF(2^m))$. If $P=(x,y) \in E(GF(2^m))$, the point $(x,x+y)$ is signified by (-P) that is called the inverse of P. O is the point at infinity serving as additive identity.
- (Point Addition). Let $P,Q \in E(GF(2^m))$, $P=(x_1,y_1)$ and $Q=(x_2,y_2)$ where $P \neq \pm Q$. Then $P+Q=(x_3,y_3)$. In this case, $x_3=\lambda^2+\lambda-x_1+x_2+a$, $y_3=\lambda(x_1+x_3)+x_3+y_1$ where $\lambda=(y_2+y_1)/(x_2+x_1)$.
- (Point Doubling). Let $P=(x_1,y_1) \in E(GF(2^m))$ 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 $\lambda=x_1+(y_1/x_1)$.

D. Point Multiplication

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 [7].

Point Multiplication is a major operation usually used in ECC. The scalar multiplication operation of a integer scalar *k* with a point *P* on the elliptic curve creates another point *Q* on this curve. 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* is shown as the equation (11)

$$k=k_{n-1}2^{n-1}+k_{n-2}2^{n-2}+\cdots+k_1+k_0 \qquad (11)$$

where $k_{n-1}=1$ and $k_i \in \{0,1\}, i=0,1,2,\ldots,n-1$ [5]. 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 (1) illustrates the scalar multiplication operation of a integer scalar *k* with a point *P* on the elliptic curve using binary method [3]. This method can be applied for both elliptic curves over $GF(p)$ and $GF(2^m)$.

Algorithm (1). Scalar Multiplication of a Point
Input : point P and integer scalar k
Output : point Q such that Q = kP
Begin
$k_i \in \{0,1\}, i=0,1,2,\ldots,n-1$
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

IV. Implementation and Experiments

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. We have already described how to implement them in our paper [6]. 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, elliptic curve cryptosystems are implemented by using corresponding methods of ECCFpCx class and ECCF2mCx class. For

the implementation logic design of elliptic curve cryptosystems, the general hierarchy is shown in Fig (4).

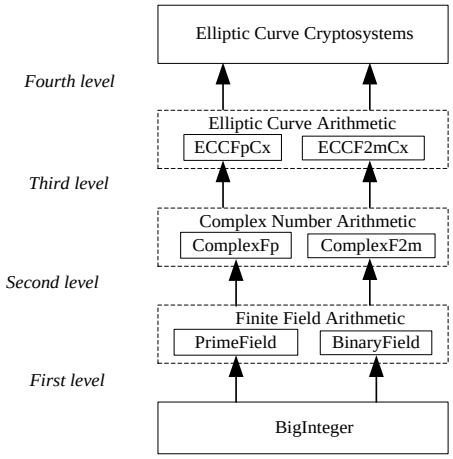


Fig. 4. Implementation logic design

A. Experiments on $Z(GF(p))$

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 methods of ComplexFp class. The followings are experiments on arithmetic operations of complex numbers in $Z(GF(p))$.

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

B. Experiments on $Z(GF(2^m))$

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 $f(x)=x^3+x+1$ as shown in Table (2), that is, $x,y,z\in Z(GF(f(x)))$. The arithmetic of these complex numbers is computed by using methods of ComplexF2m class. The followings are experiments on arithmetic operations of complex numbers in $Z(GF(2^m))$.

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

C. Elliptic Curve ElGamal Encryption Scheme

1) Experiments on $E(Z(GF(p)))$

Let's consider to encrypt and decrypt the message using the elliptic curve $E:y^2=x^3+x+1$ over $Z(GF(7))$ where $a=1$ and $b=1$. The followings are experiments on elliptic curve arithmetic operations using methods of ECCFpCx class.

a) Key generation

- Entity A and Entity B agree to choose the point $P=(3+3i,6+3i)$ as a base point.
- Entity B chooses an integer $d=15$ as a private key.
- Entity B computes $Q=d\times P=15\times(3+3i,6+3i)=(6,1i)$ as a public key.

b) Encryption

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

c) Decryption

- Entity B receives the points C_1 and C_2 as cipher texts.
- Entity B computes the message $M=C_2-(d\times C_1)$ = $(2+6i,3+5i)-15.(1+6i,4+4i)=(2+1i,3+2i)$.

2) Experiments on $E(Z(GF(2^m)))$

Let's consider to encrypt and decrypt the message using the elliptic curve $E:y^2+xy=x^3+x^2+1$ over $Z(GF(f(x)))$ where $a=1$ and $b=1$. The followings are experiments on elliptic curve arithmetic operations using methods of ECCF2mCx class.

a) Key generation

- Entity A and Entity B agree to choose the point $P = (1+3i,\ 1+4i)$ as a base point.
- Entity B chooses an integer $d = 5$ as a private key.
- Entity B computes $Q = d \times P = 5 \times (1+3i, 1+4i)$ $= (3+5i, 4+1i)$ as a public key.

b) Encryption

- Entity A chooses the point $M = (2+6i, 3+4i)$ as a message.
- Entity A chooses an integer $r = 3$ as a random number.
- Entity A computes $C_1 = r \times P = 3 \times (1+3i, 1+4i)$ $= (2+6i, 1+2i)$.
- Entity A computes $C_2 = M + (r \times Q)$. $= (2+6i, 3+4i) + 3.(3+5i, 4+1i) = (3,0)$.
- Entity A sends the points C_1 and C_2 to Entity B as cipher texts.

c) Decryption

- Entity B receives the points C_1 and C_2 as cipher texts.
- Entity B computes the message $M = C_2 - (d \times C_1)$ $= (3,0) - 5.(2+6i, 1+2i) = (2+6i, 3+4i)$.

D. Point Counting

The number of points on the elliptic curves over $GF(p)$ and $GF(2^m)$ is computed by our systems, ECCFp and ECCF2m, implemented in the reference [7]. 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 Appendix (A). 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 Appendix (B).

The number of points on the elliptic curves over $Z(GF(p))$ and $Z(GF(2^m))$ is computed by our systems, ECCFpCx and ECCF2mCx, implemented in this paper. Our systems compute and list all the point on the curve by substituting each of the values 0, 1, 2, 3,…,e.t.c in turn for real part and imaginary part of x in the curve elliptic equation and finding real part and imaginary part of y that satisfy the elliptic curve equation. The total number of points on the elliptic curve $E: y^2 = x^3 + x + 1$ over $Z(GF(7))$ is 55 and all the points are shown in Appendix (C). The total number of points on the elliptic curve $E: y^2 + xy = x^3 + x^2 + 1$ over $Z(GF(2^3))$ is 105 and all the points are shown in Appendix (D).

V. Conclusion

The sections (4.a) and (4.b) 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. The section (4.c) proves that complex numbers over finite fields can be used to construct the elliptic curve cryptosystems. The section (4.d) proves that the total number of points on the elliptic curve over finite field of complex numbers is much larger than the same curve over finite field of rational numbers, that is, the order of the elliptic curve over finite field of complex numbers is much greater than the order of the same curve over the same finite field of rational numbers. This effect increases the complexity of ECDLP. Therefore, the security of elliptic curve cryptosystems implemented over finite field of complex numbers is greatly improved. This approach makes general attacks [8] over elliptic curve more difficult.

Appendix A. All points on $E: y^2 = x^3 + x + 1$ over $GF(7)$.

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

Appendix B. 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*

Appendix C. 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*	

Appendix D. All points on $E: y^2 + xy = x^3 + x^2 + 1$ over $Z(GF(f(x)))$ 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*

REFERENCES

[1] B. A. Forouzan, *Mathematics of Cryptography, in Cryptography and Network Security*, International Edition, Singapore, McGraw-Hill press, 2008, pp. 98-117.

[2] B. A. Forouzan, *Elliptic Curve Cryptosystems, in Cryptography and Network Security*, International Edition, Singapore, McGraw-Hill press, 2008, pp. 321-330.

[3] 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.

[4] E. Kreyszig, *Complex Numbers and Their Geometric Representation, in Advanced Engineering Mathematics*, 10th edition, USA, John Wiley & Son Inc., 2011, pp. 608-612.

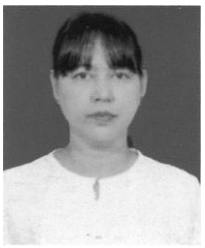
[5] K. H. Rosen, *Number Theory and Cryptography, in Discrete Mathematics and its Applications*, 7th ed, New York, USA, McGraw-Hill press, 2011, pp 237-294.

[6] N. N. Hla and T. M. Aung, "Implementation of finite field arithmetic operations for large prime and binary fields using java BigInteger class," *International Journal of Engineering Research and Technology*, vol. 6, no. 8, pp. 450-453, August 2017.

[7] T. M. Aung and N. N. Hla, "Implementation of elliptic curve arithmetic operations for prime field and binary field using java BigInteger class," *International Journal of Engineering Research and Technology*, vol. 6, no. 8, pp 454-459, August 2017.

[8] T. M. Aung and N. N. Hla. "A study of general attacks on elliptic curve discrete logarithm problem over prime field and binary field," *World Academy of Science, Engineering and Technology, International Journal of Computer and Information Engineering*, vol. 11, no. 11, pp. 1153-1160, 2017.

Tun Myat Aung was born in Yangon, Myanmar. He got M. Engnn & Tech (I.T) and Ph.D (I.T) from National Research Nuclear University MEPhI (Moscow Engineering Physics Institute). He is a professor from University of Computer Studies, Yangon. He is interested in Cryptography, Stenography and Network Security, Communication Technology, Software Computing Technology, Database Technology, Business and Economic Information Technology, and Mathematics.



Ni Ni Hla is a lecturer from University of Computer Studies, Yangon. She got M.Sc(Maths) from Yangon University and M.I.Sc from University of Computer Studies, Yangon. She is interested in Mathematics, Software Computing, Cryptography, Stenography and Network Security.