COMPLEX PUBLIC KEY CRYPTOSYSTEMS

Dr. Ali Makki Sagheer
University of Anbar

Naji Mottar
University of Diyala

Abstract:

After the 1973, there are several Public Key Cryptosystems are developments, all systems based on hard mathematical problems such as Discrete Logarithm, Integer Factorization, Subset, or Elliptic Curve Discrete Logarithm Problem. Which problems are defined over Finite Abelian Group. In this paper we proposes new concept in the public key system that is depend on complex numbers field. The complex numbers can be define over Finite Field to construct an Abelian Group under addition and multiplication operations, we call it Complex Finite Field. There are a hard mathematical problem is proposed in the constructed group we call it Complex Discrete Logarithm Problem. After that we design public key cryptosystems based on the suggested problem. Also it appears to offer equal security for a far smaller bit size, with problem harder than DLP.

1. INTRODUCTION

Mathematics is amazing not only in its power and beauty, but also in the way that it has applications in so many areas. Complex numbers comprise a computational system within which one may clarify and study many kinds of mathematical problems. In this brief essay, we will describe the complex number system carefully and pose several computational problems for discussion. Then we will apply the system to develop the mathematical problem such as Discrete Logarithm Problem (DLP) of construction with straightedge and compass, we call it Complex Discrete Logarithm Problem. Finally, we will apply the system to construct public key cryptosystems based on Complex Finite Field.

However, there is an even more essential reason why most practicing mathematicians can get by with a rather naive understanding of numbers and might be better off doing so. This is because of the extremely useful geometric picture of the real and complex numbers. Much of the time, it is perfectly reasonable to visualize the complex numbers as a geometric plane, and base all other constructions upon that basic picture, oblivious to the fine structure of our objects, pretty much as one can do plenty of classical physics without worrying about the fact that the macroscopic objects we are considering arise from the complicated interaction of elementary particles [1].

The introduction of complex numbers in the 16^{th} century was a natural step in a sequence of extensions of the real number, because the square of real number can not be negative, the equation $x^2=-1$, has no solution in the real number system. In the eighteenth century mathematicians remedied this problem by invented a new number, which they denoted by $i=\sqrt{-1}$ and which they defined to have the property $i^2=-1$,this, in turn, led to the development of the *complex number* which are number of the form $a+bi$ [2].

2. THE COMPLEX NUMBERS

Complex numbers are said to have real and imaginary parts. If $z=a+bi$, then the real part of z is denoted by $\Re e(z)=a$ and $\Im m(z)=b$ denotes the imaginary part of z. The complex numbers are defined to be the set:

$C=\{a+bi | a,b \in R$, and $i^2+1=0\}$

We add and subtract complex numbers by adding and subtracting their real and imaginary parts and multiply complex numbers the way we multiply other binomial, using the fact that $i^2=-1$ [3]:

$$(a + bi) + (c + di) = (a + c) + (b + d)i$$

$$(a + bi) \cdot (c + di) = (ac - bd) + (ad + bc)i$$

These definitions are natural enough. For example, the definition of multiplication comes from assuming the field properties and using the relation $i^2+1=0$. With these operations, it's possible to show that $(C,+,\cdot)$ is a field. The conjugate of z is $z=a-bi$. The norm of a complex number is defined as:

$$N(z)=\sqrt{z_1^{\,2}+z_2^{\,2}}$$

Proposition: The following are easy to show from the definitions above:

1. $\overline{\overline{z}}=z$.
2. $z+\overline{z}=2\cdot\Re(z)$.
3. $z-\overline{z}=2\cdot\Im(z)\cdot i$.
4. $z\cdot\overline{z}=N(z)^2$.
5. $N(z\cdot w)=N(z)N(w)$.

In particular, $a+bi=0$ if and only if $a=0$ and $b=0$ or equivalently, if and only if $a^2+b^2=0$ [3].

All the standard properties that apply to Real Numbers, like the Distributive, Commutative, and Associative Properties, also apply to Complex Numbers.

3. COMPLEX FINITE FIELD LAWS

The paper proposes a definition of a Complex Finite Field defined over finite field F_q denoted by $C(F_q)$. In this subsection, we shall study some of theoretic properties of the $C(F_q)$.

Defenition 1 (Complex Finite Field): **The $(C(F_q),+,\cdot)$ is a finite field with the following group theoretic properties:**

Let $a=a_1+a_2i, b=b_1+b_2i, c=c_1+c_2i, a,b,c\in C(F_q)$ and $a_1,a_2,b_1,b_2,c_1,c_2\in F_q$.

a:-Additive Properties

- ***(1) Closure:*** **$\forall a,b\in C(F_q)$, then $a+b\in C(F_q)$.**
- ***(2) Identity:*** **$\forall a\in C(F_q)$, then $a+I_A=I_A+a=a$, so that $I_A=(0+0i)$.**
- ***(3) Inverse:*** **$\forall a\in C(F_q)$, then $a+(-a)=I_A$.**
- ***(4) Associativity:*** **$\forall a,b,c\in C(F_q)$, then $(a+b)+c=a+(b+c)$.**
- ***(5) Communicative:*** **$\forall a,b\in C(F_q)$, then $a+b=b+a$.**

b:- Multiplicative Properties

- ***(1) Closure:*** **$\forall a,b\in C(F_q)$, then $a\cdot b\in C(F_q)$.**
- ***(2) Identity:*** **$\forall a\in C(F_q)$, then $a\cdot I_M=I_M\cdot a=a$, so that $I_M=(1+0i)$.**
- ***(3) Inverse:*** **$\forall a\in C(F_q)$, then $a\cdot a^{-1}=I_M$.**
- ***(4) Associativity:*** **$\forall a,b,c\in C(F_q)$, then $(a\cdot b)\cdot c=a\cdot(b\cdot c)$.**
- ***(5) Communicative:*** **$\forall a,b\in C(F_q)$, then $a\cdot b=b\cdot a$.**

c:*-Addition and Multiplicative Properties*

- ***(1) Distributive:*** **$\forall a,b,c\in C(F_q)$, then $a\cdot(b+c)=a\cdot b+a\cdot c$.**
- ***(2) No zero divisors:*** **if $a,b\in C(F_q)$ and $a\cdot b=0$, then either $a=0$ or $b=0$.**

(3) Elements of the field: $C(F_q)^* = \{\forall a \in C(F_q)$ and $a_1, a_2 \in F_q\}$ is a group of $C(F_q)$, that is generated from complex numbers.

Definition 2: Let a be an complex number element of the group $C(F_q)$ and $a \neq 0$. Then a is said to have order k if

$a^k = a \cdot a \ldots\ldots\ldots \cdot a = I_M$

k *product*

with $a^{k'} \neq I_M$ for all $1 \leq k' \leq k$ (that is, k *is* the smallest integer such that $a^k = I_M$). If such a k exists, then, the subgroup of $C(F_q)$ is said to have finite order k, otherwise, it has infinite order.

Definition 3: From here on, for the multiplication operations on a $C(F_q)$, for $k \in Z$, $a \in C(F_q)$, and $a \neq 0$, then:

$a^k = a \cdot a \ldots\ldots\ldots \cdot a$, ($k$ *times*), for $k > 0$,
$a^0 = I_M$, and
$a^k = (a^{-1})^{-k}$, for $k < 0$.

Definition 4: The order of a $C(F_q)$ is defined as the number of complex number element of $C(F_q)$ and denoted by $\#C$.

If $a \in C(F_q)$ is of order k, then

$$H = \{ a^i \mid 0 \leq i < k-1 \},$$

is a *subgroup* of $C(F_q)$ of order k.

Definition 5: Let $a \in C(F_q)$, and $a \neq 0$. Then a is said to generator element if

$$ord(a) = \#C$$

Then,

$$C(F_q) = \{ a^k \mid 0 \leq k < \#C - 1 \},$$

4. COMPLEX DISCRETE LOGARITHM PROBLEM (CDLP)

One of the most interesting open problem in cryptography is the realization of a trapdoor on the discrete logarithm, in which to solve the DLP is hard only if published parameters are used, while it is easy by using a secret key (trapdoor key) [4].

The DLP can be defined on various finite groups as well as multiplicative group over a finite filed F_q [5], this idea can be extended to arbitrary groups and, in particular, to Complex Group. A typical example except the multiplicative group is the discrete logarithm problem on Complex number over F_q, and many cryptographic schemes are constructed on the CDLP.

Definition 6 (CDLP): For a Complex Finite Field $C(F_q)$, let $a, b \in C(F_q)$, recall that in the CDLP to find an integer $k \in Z$, is such that $a^k = b$.

Since a Complex Finite Field $C(F_q)$ is made into Abelian group by a complex number multiplicative operation. The exponential of a complex element on $C(F_q)$ actually refers to the repeated multplications. Therefore, $b = a^i$ is the i^{th} power of $a \in C(F_q)$ is the i^{th} multiple of a. The logarithm of b to the base a would be i (i.e. the inverse of exponentiation). The CDLP is of interest because its apparent intractability forms the basis for the security of Complex cryptographic schemes.

5. COMPLEX CRYPTOSYSTEMS (CCS)

Unlike previous cryptosystems, complex number work as a finite Abelian group formed by the complex elements on C(F_q) group defined over a finite field. CCS include Key Distribution, Encryption/Decryption Schemes, and Digital Signature Algorithm (DSA). The key distribution algorithm is used to share a secret key, the encryption/decryption algorithm enables confidential communication, and the DSA is used to authenticate the signer and validate the integrity of the message.

This section proposes cryptosystems that employs Complex Finite Field. It does not invent new cryptographic algorithm, but it is the first to implement existing public-key cryptosystem using complex numbers. The proposal is an analogues to the Diffie-Hellman key exchange protocol, analogues to ElGamal, Massey-Omura schemes, and DSA. The modular complex number multiplication operation in CCS is the counterpart of modular multiplication in RSA and ElGamal, and exponentiation of complex in C(F_q) is the counterpart of the modular exponentiation. To form cryptographic system using complex number, we need to find a "hard problem" corresponding to the difficulty of factoring the product of two prime or taking the discrete logarithm or elliptic curve discrete logarithm.

Consider the equation $b=a^k$, where *a* and *b* are two complex number in the C(F_q) and *k* is an integer. It is relatively easy to calculate *b* given *a* and *k*, but determining the integer *k* from a multiple of a element a^k, even with the knowledge of *a, b* and C(F_q) is a very difficult problem, known as the Complex Discrete Logarithm Problem (CDLP).

5.1 Exponentiation over complex group C(F_q)

The fundamental operation in complex cryptographic schemes is that of complex element exponentiation of a complex element by an integer. If not the most confusing term, certainly the idea of multiplying matrix refers to computing $b=a^k$, where *a* and *b* are two complex numbers in the C(F_q), group and *k* is an integer. This really means that we multiply *a* to itself *k* times.

Definition 7 (Exponentiation of a complex number on complex group* C(F_q) *by an integer): **Given $k \in Z$, and *a* is a complex element on a C(F_q), *then***

$$a^k = a \cdot a \cdot \ldots\ldots \cdot a \quad (k \text{ times}) \ldots\ldots\ldots\ldots\ldots (1)$$

And it is so called complex number exponentiation, and it is the dominant cost operation in complex cryptographic schemed, and it dominates the execution time of complex cryptographic schemes, especially the representation of CDLP.

The algorithm that can be used to compute the complex exponentiation in the C(F_q), group is Repeated-Squaring and Multiplication or fast group operation Method.

5.2 Repeated-Squaring and Multiplication Method

The most fundamental computation on C(F_q) group is the multiplication operation as shown in equation(1) with k are very large positive integer, since the computation of a^k is so fundamental in all complex numbers related computations and applications, it is desirable that such computations are carried out as fast as possible.
Remarkably enough, the idea of repeated squaring for fast exponentiation can be used almost directly for fast group operation on C(F_q).
Let $e_{n-1}\ e_{n-2}\ \ldots\ e_1\ e_0$ be the binary representation of k. then for i starting from n-1 down to 0 (e_{n-1} almost 1 and used for initialization), check whether or not e_i =1. If e_i =1, then perform a squaring and a multiplication group operation; otherwise, just perform a squaring operation. For example: compute a^{67}, since 67= 1000011, we get the following table:

Table 1 Compute a^{67} using repeated Squaring and Multiplication

i	e_i	Value	*Operations*	Status
6	e_6	1	a	Initialization
5	e_5	0	$a^2=a^2$	Squaring
4	e_4	0	$(a^{2)2}=a^4$	Squaring
3	e_3	0	$((a^2)^2)^2=a^8$	Squaring
2	e_2	0	$(((a^2)^2)^2)^2=a^{16}$	Squaring
1	e_1	1	$((((a^2)^2)^2)^2)^2\cdot a= a^{33}$	Squaring and Multiplication
0	e_0	1	$(((((a^2)^2)^2)^2)^2\cdot a)^2\cdot a= a^{67}$	Squaring and Multiplication

We have the following algorithm which implements this idea of repeated squaring and multiplication (fast group operation) for computing a^k, that is, it reduces the complexity of the computation of a^k from k to $log\ k$.
Algorithm (Repeated-Squaring and Multiplication)
Input: **a complex number $a \in C(F_q)$ and positive integer k.**
Output: **complex number $b= a^k$.**
1. Write k in the binary expansion form $k= e_{n-1}\ e_{n-2}\ \ldots\ e_1\ e_0$ (Assume k has n bits)
2. Set $b = I_M$.
3. Compute a^k:
3.1 For i from n-1 down to 0 do
3.2 $b=b^2$.
3.3 *if* e_i=1, then $b=b\cdot a$.
4. Output b: (now $b=a^k$).

6. COMPLEX PUBLIC-KEY CRYPTOSYSTEMS

The section introduces design of public-key cryptography that employs the complex finite field. More specifically, it'll introduce complex cryptosystems analogues to several well known public-key cryptosystems including key exchange, encryption/decryption, and DSA schemes.

For any cryptographic system based on the DLP, there is an analogy to complex finite field. In what follows, it'll introduce complex cryptosystems analogues to three widely used public-key cryptosystems, namely Diffie-Hellman key exchange system, the Massey-Omura, the El-Gamal public-key cryptosystems.

6.1. Analogy of the Diffie-Hellman Key Exchange System

This system is merely a method for exchanging keys; no messages are involved. Alice and Bob first publicly choose a finite field F_q and a Complex group $C(F_q)$ defined over it. Then they publicly choose a complex number $b \in C(F_q)$ to serve as their "Base complex number". It is a generator of the key. To generate a key, Alice chooses random integer *e* between 1 and #C, and keeps it secret. She then computes $b^e \in C(F_q)$ and makes that public. Bob chooses his own secret random integer *d* between 1 and #C, and makes public $b^d \in C(F_q)$. The secret key is then $b^{ed} \in C(F_q)$. Both Alice and Bob can compute this key. For example, Alice knows b^d (public knowledge) and her own secret *e*. Charlie, on the other hand, only knows *b*, b^e and b^d. Without solving the CDLP, (finding *d* knowing *b* and b^d), there is no way for him to compute b^{ed} only knowing b^e and b^d. The following algorithm illustrates this manner.

Algorithm (Diffie-Hellman key exchange system with CDLP)

1. **Initialization**
   - **Alice and Bob publicly choose a complex finite field $C(F_q)$.**
   - **They publicly choose a random "Base complex number" $b \in C(F_q)$ such that *b* generates a large subgroup of $C(F_q)$.**
2. **Key generation**
   - **Alice chooses a secret random integer *e*. She then computes $b^e \in C(F_q)$.**
   - **Bob chooses a secret random integer *d*. He then computes $b^d \in C(F_q)$.**
   - **Make b^e and b^d public and keep *e* and *d* secret.**
3. **Calculation of the secret key b^{ed}**
   - **Alice computes the secret key $b^{ed}=(b^d)^e$.**
   - **Bob computes the secret key $b^{ed}=(b^e)^d$.**

There is no known fast way to compute b^{ed} if only knows *b*, b^e and b^d, which is CDLP.

6.2. Analogy of the Massey-Omura Cryptosystem

In this system the complex finite field $C(F_q)$ have been made publicly known. Alice and Bob both select a random integer e_1 and e_2 between 1 and *#C* respectively with $gcd(e_1, \#C)=1$ and $gcd(e_2, N)=1$. They also compute their inverses $d_1 = e_1^{-1} \bmod \#C$ (ie. $d_1 e_1 = 1 \bmod \#C$) and $d_2 = e_2^{-1} \bmod \#C$ (ie. $d_2 e_2 = 1 \bmod \#C$), then, keep everything secret. If Alice wants to send the message P_c (i.e. PlainText Complex, we represent the message as a pairs; each equivalent

to complex number denoted by P_c) to Bob, she first sends him the message P_c^{e1}. This means nothing to Bob, since he does not know d_1. He ever, he can exponentiate it by his e_2 and send the message P_c^{e1e2} back to Alice. Then Alice can help unravel the message by exponentiating this new message by d_1 which sends $P_c^{e1e2d1} = P_c^{e2}$ back to Bob. Then Bob can exponentiate this message by d_2 to get the original message ($P_c^{e2d2} = P_c$). During this process Charlie sees P_c^{e1}, P_c^{e2}, and P_c^{e1e2}.
Without solving the CDLP –finding e_2 and then its inverse knowing P_c^{e1} and P_c^{e1e2} - there is no way for him to find P_c. The following algorithm illustrates this manner.

Algorithm (Massey-Omura Cryptosystem with CDLP)

1. Initialization
   - Alice and Bob publicly choose a complex finite field C(F_q).
   - They publicly known the order number of the of C(F_q) denoted by *#C.*
2. Key generation
   - Alice chooses a secret random integer e_1 between *1* and *#C*, such that $gcd(e_1, \#C)=1$. She then computes its inverse $d_1=e_1^{-1} \bmod \#C$.
   - Bob chooses a secret random integer e_2 between *1* and *#C,* such that $gcd(e_2, \#C)=1$. He then computes its inverse $d_2=e_2^{-1} \bmod \#C$.
   - Keep e_1, d_1, e_2, and d_2 secret.
3. Transmission procedure

   Alice sends the message P_c to Bob as follows:
   - Alice computes P_c^{e1}, and sends it to Bob.
   - Bob computes P_c^{e1e2}, and sends it to Alice.
   - Alice computes $P_c^{e1e2d1} = P_c^{e2}$, and sends it to Bob.
   - Bob computes $P_c^{e2d2} = P_c$).

6.3. Analogy of the ElGamal Cryptosystem

In this system the complex finite field C(F_q), and the "Base complex number" $b \in C(F_q)$ are public information. Bob randomly chooses an secret integer d ($1<d<\#C$), and publishes the complex b^d. If Alice ants to send the message P_c (i.e. PlainText Complex, we represent the message as a pairs; each equivalent to complex number denoted by P_c) to Bob, she will choose a secret random integer e ($1<e<\#C$) and send ($P_c \cdot b^{ed}$, b^e) to Bob.
Bob will then exponentiate the second complex number in the pair by d to get b^{ed}, the compute the inverse of the key complex b^{ed} to get $(b^{ed})^{-1}$ and multiply by the first complex in the pair $P_c \cdot b^{ed}$ to find P_c. In the meantime, Charlie has only seen b^e and b^d. Without solving the CDLP (eg. finding d knowing b and b^e), there is no way for him to find P_c. The following algorithm illustrates this manner.

Algorithm (ElGamal Cryptosystem with CDLP)

1. Initialization
   - Alice and Bob publicly choose a complex finite field C(F_q).

- They publicly choose a random "Base complex number" $b \in C(F_q)$ such that b generates a large subgroup of $C(F_q)$.

2. Key generation
   - Bob chooses a secret random integer d in interval [2, #C].
   - He then computes $Q=b^d$.
   - Make Q public and keep d secret.
3. Encryption

Alice sends the message P_c to Bob as follows:
   - Select random integer e in interval [2, #C].
   - Compute b^e.
   - Compute $K=Q^e$, (i.e. $K=b^{de}$).
   - Compute ciphertext complex $C_c = P_c + K$.
   - Transmit the pair complexes (C_c, b^e).
4. Decryption

Bob retrieves the message as follows:
   - Compute $K= (b^e)^d$, (i.e. $K=b^{ed}$).
   - Compute $-K$, add $-K$ with the ciphertext complex C_c:
     $P_c=C_c-K$.

7. IMPLEMENTATION

The proposed system us programmed by MatLap Version 7 programming language on P4 PC with CPU of 3 G.B and RAM of 2 G.B. Then the methods is applied on different size messages, which takes plaintext and devided into pairs of blocks each pair corrispond to complex number containd in $C(F_q)$ and computes the running time of the encryption and decryption of each messages.

A. Diffie-Hellman key exchange system with CDLP

1. Initialization
   - Alice and Bob publicly choose a complex finite field $C(F_{4919})$.
   - They publicly choose a random "Base complex number" $b=2998+3213i \in C(F_{4919})$ such that $2998+3213i$ generates a large subgroup of $C(F_{4919})$.
2. Key generation
   - Alice chooses a secret random integer $e=4725$. She then computes $b^e=(2998+3213i)^{4725}=3546+4255i$.
   - Bob chooses a secret random integer $d=1234$. He then computes $b^d=(2998+3213i)^{1234}=1973+1049i$.
   - Make b^e and b^d public and keep e and d secret.
3. Calculation of the secret key b^{ed}

- Alice computes the secret key $b^{ed}=(b^{d})^{e}=(1973+1049i)^{4725}=2870+1364i$.
- Bob computes the secret key $b^{ed}=(b^{e})^{d}=(3546+4255i)^{1234}=2870+1364i$.

B. Massey-Omura Cryptosystem with CDLP

1. Initialization
   - Alice and Bob publicly choose a complex finite field $C(F_{4919})$.
   - They publicly known the order number of the of $C(F_{4919})$ denoted by $\#C=24196560$.
2. Key generation
   - Alice chooses a secret random integer $e_1=9321841$, such that gcd(9321841,24196560)=1. She then computes its inverse $d_1=e_1^{-1} \bmod \#C = 9321841^{-1} \bmod 24196560=623281$.
   - Bob chooses a secret random integer $e_2=5463829$, such that gcd(5463829,24196560)=1. He then computes its inverse $d_2=e_2^{-1} \bmod \#C = 5463829^{-1} \bmod 24196560=19433869$.
   - Keep e_1, d_1, e_2, and d_2 secret.
3. Transmission procedure

Alice sends the message $P_c=1000+2000i$ to Bob as follows:

- Alice computes $P_c^{e1}=(1000+2000i)^{9321841}=4495+1682i$, sends it to Bob.
- Bob computes $P_c^{e1e2}=(4495+1682i)^{5463829}=598+3852i$, sends it to Alice.
- Alice computes $P_c^{e1e2d1}=(598+3852i)^{623281}=3675+3069i= P_c^{e2}$, sends it to Bob.
- Bob computes $P_c^{e2d2}=(3675+3069i)^{19433869}=1000+2000i=P_c$).

C. ElGamal Cryptosystem with CDLP

1. Initialization
   - Alice and Bob publicly choose a complex finite field $C(F_{4919})$.
   - They publicly choose a random "Base complex number" $b=2998+3213i \in C(F_{4919})$ such that generates a large subgroup of $C(F_{4919})$.
2. Key generation
   - Bob chooses a secret random integer $d=1234$.
   - He then computes $Q=b^{d}=(2998+3213i)^{1234}=1973+1049i$.
   - Make Q public and keep d secret.
3. Encryption

Alice sends the message $P_c=1000+2000i$ to Bob as follows:

- Select random integer $e=4725$.
- Compute $b^{e}=(2998+3213i)^{4725}=3546+4255i$.
- Compute $K=Q^{e}=(b^{d})^{e}=(1973+1049i)^{4725}=2870+1364i$ (i.e. $K=b^{de}$).
- Compute ciphertext complex $C_c=P_c+K=1000+2000i+2870+1364i=3870+3364i$.
- Transmit the pair complexes (C_c, b^{e})=(3870+3364i, 3546+4255i).
4. Decryption

Bob retrieves the message as follows:

- Compute $K=(b^{e})d=(3546+4255i)^{1234}=2870+1364i$, (i.e. $K=b^{ed}$).
- Compute $-K=-2870-1364i=2049+3555i$ under $C(F_{4919})$.
- Add $-K$ with the ciphertext complex C_c:

$P_c=C_c+(-K)= 3870+3364i+2049+3555i=1000+2000i$.

8. THE COMPUTATIONAL COMPLEXITY

The Computational Complexity of the El-Gamal encryption/decryption algorithms using DLP compared to proposed problem CDLP is as follows:

1. **Encryption/Decryption using DLP:**
   - **Let the size of the input message block be *n.***
   - **The complexity of the computing $q=b^e$ is: $T(Q)=T(b^e)=O(log\ n)$ arithmetic (multiplication) operation, using Fast Exponential Algorithm, each multiplication operation has $O(log^2\ n)$ bit operation [6].**

     Then,

     $T(Q)=O(log^3\ n)$ bit operation.

     ***Also,* $T(k)=O(log^3\ n)$ bit operation.**

   - **The complexity of the computing ciphertext $c=m*k$ is: $T(c)=T(m*k)=O(log^2\ n)$ bit operation, for each encryption block, suppose there are 100 blocks of message; then, $100T(c)=100T(m*k)=O(100log^2\ n)$.**

     The overall Complexity is $O(100log^2\ n)+O(2log^3\ n)$.

2. **Encryption/Decryption using CDLP:**
   - **Let the size of the input message block be *n.***
   - **The complexity of the computing $q=b^e$ is: $T(Q)=T(b^e)=O(log\ n)$ complex multiplication operation, using Repeated Squaring and Multiplication method.**

     Then,

     $T(Q)=O(2log^3\ n)$ bit operation.

     ***Also,* $T(K)=O(2log^3\ n)$ bit operation.**

   - **The complexity of the computing ciphertext $C_c=P_c\cdot K$ is: $T(C_c)=T(P_c\cdot K)=O(2log^2\ n)$ bit operation, for each encryption block, suppose there are 100 blocks of message (are represented as 50 pairs of each pair correspond to complex number); then, $50T(C_c)=50T(P_c\cdot K)=O(100log^2\ n)$.**
     The overall Complexity is $O(100log^2\ n)+O(4log^3\ n)$.

Finally, the computational complexity of the implementation of encryption/decryption function are same approximately.

9. THE RUNNING TIME COMPARISON

The running time of the El-Gamal method with DLP over F_{4919} and the base number is 4567, the secret key is 2931 and the public key is 4066. The order of the multiplicative group over F_q is q-1, then, the order of the multiplicative group over F_{4919} is 4918. The DLP over F_{4919} is solved by 0 msec.
The running time of the El-Gamal method with CDLP over F_{4919} and the base complex number 2998+3213i, the secret integer key is 4725 and the public complex key is 3546+4255i. The order of the C(F_q) group of base complex is q^2-1, then, the order of the C(F_{4919}) group of base complex number 2998+3213i is q^2-1=24196560. The CDLP over F_{4919} is solved by 16 msec.
Therefore, we conclude the order of the C(F_q) group of base complex number is q^2-1 or its factors. The following table explain the running time of solving DLP and CDLP over F_{4919}, F_{59011}, F_{699367}, $F_{2099221}$, and $F_{9999991}$.

Table 1 Running Time of solving DLP and CDLP in *msec*

Finite Field F_q		DLP			CDLP	
	Base Number	Power	Running Time	Base Complex	Power	Running Time
F_{4919}	4567	2931	0	2998+3213i	2931	63
F_{59011}	54567	52931	0	2998+3213i	52931	1140
F_{699367}	664567	652931	125	2998+3213i	652931	14062
$F_{2099221}$	1774567	2077731	140	2998+3213i	2077731	154682
$F_{9999991}$	8999999	9888888	953	2998+3213i	9888888	1392138

There is a clear growth of the time execution when use the complex group C(F_q) and increase as long as the finite field size is increased. This increasing with small numbers, what is happen when a large number is applied, such as 100 digit number, 200 digit number or more, the complexity is increased rapidly, show Figure 1.

10. SECURITY OF COMPLEX CRYPTOSYSEMS

The complication associated with CCS comes from the wide variety of possible group structures of the complex element in the C(F_q) and from the fact that complex modular multiplication is somewhat more complicated than classical modular multiplication.

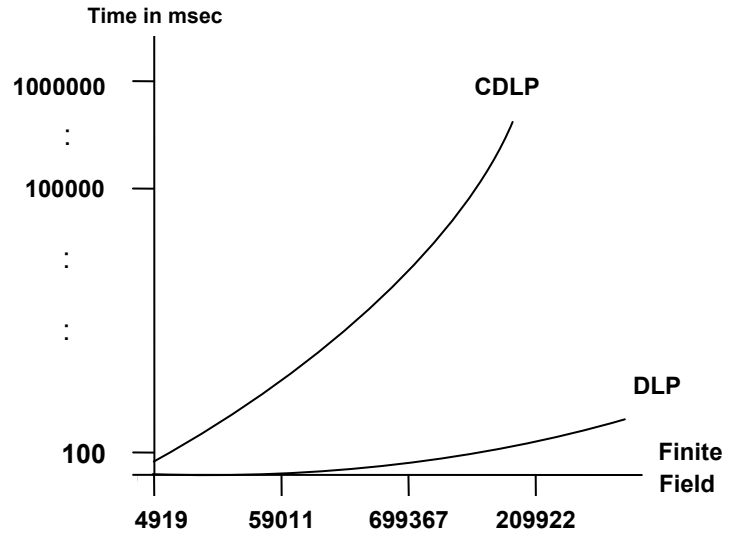

Figure 1: The complexity evaluation of DLP and CDLP

The security of CCS depends on how difficult it is to determine the integer *d*, given the complex number *b* and the complex a^d where $b=a^d \bmod q$. This is referred to as the CDLP. Also that it appears to offer equal security for a far smaller bit size, because the size (order) of the C(F_q) appears at most q^2-1, that means the calculation is applied with *q*-bit size, while to solve the CDLP needs q^2-1 operations.
Therefore, the cryptanalyzer need to analysis and solve the CDLP to cryptanalysis the public based on it.

11. CONCLUSION

The project defined the C(F_q) that proved as an Abelian group to use it in the proposed cryptosystems. Then, discover that the C(F_q) group has a one way function similar to DLP and ECDLP, which CDLP. The construction of cipher system is based on the difficulty of solution of the CDLP that is a clear change in the cryptography and opens new windows for treatment with special group and new operations. There is a computational advantage in using the CCS with the shorter key length that reduces the overall calculations with secure system. The CDLP appears more complicated than DLP, because the complex operations increase the complexity as long as the size is increased.
The CDLP over F_q is more intractable than the DLP in F_q. It is this feature that makes cryptographic system based on the CDLP even more secure than that based on the DLP, because the C(F_q) gives a large group over small field size. Since the group C(F_q) of order q^2-1 or its favtors, therefore, some of the strongest algorithms for solving DLP cannot be adaptive to the CDLP.

References:

[1] Minhyong Kim, Why everyone should know number theory, Department Of Mathematics, University Of Arizona, Tucson, AZ85721, April, 1998.

[2] H. Anton, I. Bivens and S. Davis, *Calculus*, John Wiley & Sons Inc., 2002.

[3] R. L. Finney and G. B. Thomas, *Calculus*, Addison Wesley Pub., 1990.

[4] M. B. Nathanson, *Elementary Methods in Number Theory*, Graduate Text in Mathematics 195, Springer-Verlag, 2000.

[5] R. A. Mollin, *Number Theory and Applications*, NATOASI series 1989.

[6] S.Y. Yan, *Number Theory for Computing* , Springer-Verlag, 2000.

[7] J. M. Kizza, Computer Network Security, Springer Inc., 2005.

[8] N. Koblitz, *A Course in Number Theory and Cryptography*, Springer-Verlag, 1987.

[9] G. Berkho_, Saunders Mac Lane, A Survey of Modern Algebra, ISBN:978-1-56881-454-4, AKP Classics, 2008.

[10] W. Stallings, *Cryptography and Network Security, Principle and Practice*, Addison Wesley, 1999.

[11] M. Stamp, *Information Security Principles and practice*, JohnWiley & Sons, Inc., 2006.

[12] G. Williams, *Linear Algebra with Applications*, 4th edition, Jones and Bartlett Publishers, Inc., 2001.

[13] Ali M. Sagheer, Design Of Public-Key Cryptosystems Based On Matrices Discrete Logarithm Problem, MASAUM Journal of Computing, Volume 1 Issue 2, September 2009.

انظمة التشفير ذات المفتاح المعلن المركب

د. علي مكي صغير
جامعة الانبار

وناجي مطر
جامعة ديالى

المستخلص:

بعد عام 1973 طورت العديد من انظمة التشفير ذات المفتاح المعلن، وهذه الانظمة جميعها تعتمد على مسائل رياضية صعبة مثل مسألة اللوغارتم المتقطع وتحلبل الارقام الى عوملها الاولية والمجموعة الجزئية ومسألة اللوغارتم المتقطع في المنحنيات الاهليلجية. هذه المسائل جميعها معرفة على زمرة ابيلية منتهية.

في هذا البحث اقترحنا مفهوم جديد في انظمة التشفير ذات المفتاح المعلن يعتمد على حقل الاعداد المركبة. الاعداد المركبة يمكن تعريفها على الحقل المنتهي لانشاء زمرة ابيلية من عمليات الجمع والضرب اطلقنا عليه الحقل المنتهي المركب. واوجدنا مسألة رياضية صعبة في الزمرة المكونة من الاعداد المركبة اطلقنا عليها مسألة اللوغارتم المتقطع المركب. بعد ذلك صممنا انظمة تشفير ذات مفتاح معلن تعتمد على المسألة المقترحة. والتي ظهرت بمستوى امنية عالي بمفتاح ذو حجم صغير واصعب في التحليل من مسألة اللوغارتم المتقطع.