Implementation of the Elliptic Curve Cryptography over Gaussian Integral Finite Group onto Excel

Kazuki Naganuma*, Takashi Suzuki*, Hiroyuki Tsuji*, and Tomoaki Kimura*
*Kanagawa Institute of Technology, Japan

***Abstract*— The Elliptic Curve Cryptography is known as cryptography safer than the RSA Cryptosystem. Defining the elliptic curve over a finite field operated with 64 bits of integral data type currently loaded on Excel is too dangerous to implement the Elliptic Curve Cryptography onto Excel. Therefore, we propose the method of defining the elliptic curve on Gaussian Integer to enhance the safety. In this paper, we confirm the method can enhance the safety and the cryptography can be operated accurately.**

I. INTRODUCTION

In 1985, Miller et al proposed the Elliptic Curve Cryptography (ECC), which is safer than the RSA Cryptosystem. It is known that 160 bits of the key length of ECC corresponds to 1024 bits of that of RSA Cryptosystem [1]. Therefore, ECC has recently spread from the viewpoints of calculation memory consumption and the amount of bits memorizing parameters. However, ECC is more complicated than the RSA Cryptosystem. The easier method of confirmation of ECC is preferable because of the difficulty. Hence, ECC can be confirmed easily by implementing ECC onto Microsoft Excel (Excel).

Implementing ECC onto Excel has three effects as follows. First, it can visualize the confirmation of ECC. Second, it can encrypt and decrypt files such as csv file including numerical value data. Third, it can be used wherever Excel can be operated.

On the other hands, there are problems in implementing ECC onto Excel. The biggest problem is the limitation of the bits length of calculations in integral type. Although current Excel has 64 bits of integral type, the finite groups constructed by the type are not enough to make ECC safe. Therefore the method to enhance the safety is required. A relative study has solved the limitation of integral type in C language by using CNP MP, which is an arbitrary precision arithmetic library [2]. However, the other method is required because arbitrary precision arithmetic on Excel seems to have high computational complexity. Hence, we propose the method to enhance ECC safety with a finite group constructed by Gaussian Integer. The possibility to make ECC safer by using Gaussian Integer is known in theory [3]. However there is nothing to mention the algorithm. Additionally, there is nothing to confirm operation in ECC accurately.

In this paper, we consider the algorithm of congruent operations in Gaussian Integer. Then, we define operations on elliptic curves over a finite group with not integer but Gaussian Integer and confirm the method will be able to enhance the safety. Moreover, we enable ECC to be used on Excel and confirm the accuracy of encryption and decryption.

II. ELLIPTIC CURVE CRYPTOGRAPHY

A. Elliptic Curve

Elliptic curves used in Cryptography include points satisfying the condition of the congruent formula as follows:

$$y^2 \equiv x^3 + ax + b \quad (1)$$

and an infinity point: $\mathcal{O} = (\infty, \infty)$ [4]. the a and b in formula E are values in a field K. Then, the K-rational-point sets of the elliptic curves are defined as follows [4].

$$E(K) = \{(x, y) \in K | y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\} \quad (2)$$

B. K-rational-point Addition and Multiplication in Elliptic Curves

Suppose $P = (x_1, y_1), Q = (x_2, y_2) \in E(K)$ which are not infinity points respectively. Addition $R = P + Q = (x_3, y_3)$ is defined as follows [4].

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2 \\ y_3 = \lambda(x_1 - x_3) - y_1 \end{cases} \quad (3)$$

So infinity point $\mathcal{O}$ is identity element in the addition, $P + \mathcal{O} = \mathcal{O} + P = P$ is satisfied.

Multiplication is power addition. Let n be in natural number and $P \in E(K)$. Multiplication *nP* is defined as follows [4].

$$nP = P + \cdots + P(n\ times\ addtion) \quad (4)$$

C. the Key Length of ECC

The safety of ECC is based on the difficulty of Discrete Logarithm Problem. Public key and secret key are defined as follows [4].

$$\begin{matrix} Public\ key = P, B \\ Secret\ key = s \end{matrix} \left(P, B \in E(\mathbb{F}_p), s \in \mathbb{N}, P = s * B\right) \quad (5)$$

$\mathbb{N}$ is a set of natural number, and $\mathbb{F}_p$ is a set of a finite field including p order of elements.

The p limits the length of secret key. $\#E(\mathbb{F}_p)$, which is order of $E(\mathbb{F}_p)$, satisfies the condition of following inequality because of Hasse's theorem [4].

$$p + 1 - 2\sqrt{p} \leq \#E(\mathbb{F}_p) \leq p + 1 + 2\sqrt{p} \quad (6)$$

The length of secret key does not exceed $\#E(\mathbb{F}_p)$ because of a property of cyclic group. Therefore, composing elliptic curves with higher order finite group is required to make the length longer.

III. Equivalence between Elliptic Curves over Gaussian Integer and That over Integer

In this chapter, the reason why elliptic curves corresponding 128 bits of integer can be composed just by using 64 bits of integral type is explained.

Suppose $\rho = p_1 + p_2 i \in \mathbb{Z}[i](p_1, p_2 \neq 0)$ and is Gaussian prime number. For a quotient group over ρ, the following relation is satisfied.

$$\mathbb{Z}/(p_1^2 + p_2^2)\mathbb{Z} = \mathbb{Z}[i]/\rho\mathbb{Z}[i] \quad (7)$$

$\mathbb{Z}[i]$ is a set of Gaussian Integer and $\mathbb{Z}$ is a set of integer. $\mathbb{Z}[i]/\rho\mathbb{Z}[i]$ and $\mathbb{Z}/(p_1^2 + p_2^2)\mathbb{Z}$ are a Gaussian Integral quotient group over ρ and an integral quotient group over $p_1^2 + p_2^2$ respectively.

The elements of $\mathbb{Z}[i]/\rho\mathbb{Z}[i]$ can be collected in integer as follows [5].

$$\{n | n \in \mathbb{N} \cup \{0\}, n < p_1^2 + p_2^2\}$$

The elements of $\mathbb{Z}/(p_1^2 + p_2^2)\mathbb{Z}$ can be collected as the above set [5]. Therefore, $\mathbb{Z}[i]/\rho\mathbb{Z}[i]$ is equal to $\mathbb{Z}/(p_1^2 + p_2^2)\mathbb{Z}$ and these sets are equal to $\mathbb{F}_{p_1^2+p_2^2}$. Hence, $\mathbb{Z}[i]/\rho\mathbb{Z}[i] = \mathbb{Z}/(p_1^2 + p_2^2)\mathbb{Z} = \mathbb{F}_{p_1^2+p_2^2}$ is satisfied.

Accordingly, elliptic curves composed by $\mathbb{Z}[i]/\rho\mathbb{Z}[i]$ and $\mathbb{Z}/(p_1^2 + p_2^2)\mathbb{Z}$ are both equal to $\mathrm{E}\left(\mathbb{F}_{p_1^2+p_2^2}\right)$. Thus they are equal to each other.

The length of digit of p_1 and that of p_2 are half or shorter than that of $p_1^2 + p_2^2$. Whereby, elliptic curves corresponding 128 bits of integer can be composed just by using 64 bits of integral type.

IV. Algorithms of Gaussian Integral Operation

In this chapter, we explain how to select elements of a quotient group over a Gaussian Integral finite group and how to do arithmetic operation in Gaussian Integer. To construct K-rational-point operations of Gaussian Integral elliptic curves, changing integral arithmetic and other operations in reference [2] into Gaussian Integral arithmetic and other operations below is required. It enables ECC to be constructed with Gaussian Integer.

A. Composition of Gaussian Integer

We use two integral types to compose Gaussian Integer. It means the types are used for real number and imaginary number.

B. Limitation of a Value of Integral Type

The limitation of a value of integral type is defined as "a max absolute integral value which can be used in Excel / 2" without overflow in one time addition. In this research, the limitation is defined as (2^63-1)/2 for 64-bit version Excel.

C. Division

Division in Gaussian Integer is calculated in algorithm as follows. It is described as α/β. This operation is required for selecting elements of a quotient group which is needed for congruent operations.

$\alpha = a_1 + a_2 i, \beta = b_1 + b_2 i$ are in Gaussian Integer

$$result \leftarrow round\left(\frac{a_1 b_1 + a_2 b_2}{b_1^2 + b_2^2}\right) + round\left(\frac{a_2 b_1 - a_1 b_2}{b_1^2 + b_2^2}\right) i$$

return result

$\frac{a_1 b_1 + a_2 b_2}{b_1^2 + b_2^2}$ and $\frac{a_2 b_1 - a_1 b_2}{b_1^2 + b_2^2}$ in *round* in line 2 are calculated in double type (64 bits). *Round* which rounds off an argument at first decimal place rounds down just 0.5 in this case.

D. Selection of Elements of Quotient Group

The selection of elements of quotient group is calculated in algorithm as follows. It is described as $\alpha(mod.\pi)$.

$\alpha = \alpha_1 + \alpha_2 i$ is in Gaussian Integer, $\pi = p_1 + p_2 i$ is a Gaussian prime number

$\kappa \leftarrow \alpha/\pi$

$result \leftarrow \alpha - \pi \times \kappa$

return result

$\pi \times \kappa$ in line 3 is multiplication in Gaussian Integer.

E. Congruent Operations in Gaussian Integer

In this section, we explain how to do congruent operations modulo Gaussian prime number. The congruent operations include congruent arithmetic operations and congruent exponentiation.

i. Congruent Addition

Congruent addition is calculated in algorithm as follows. It is described as $\alpha + \beta(mod.\pi)$.

$\alpha = a_1 + a_2 i,\ \beta = b_1 + b_2 i$ are elements of quotient group, π is a modulus

$result \leftarrow (a_1 + b_1) + (a_2 + b_2) i$

$result \leftarrow result(mod.\pi)$

return result

It can be calculated without overflow because of properties of the elements and limitation of a max value.

ii. Congruent Subtraction

Congruent subtraction is calculated in algorithm as follows. It is described as $\alpha - \beta(mod.\pi)$.

$\alpha = a_1 + a_2 i, \beta = b_1 + b_2 i$ are elements of quotient, π is a modulus

$result \leftarrow (a_1 - b_1) + (a_2 - b_2) i$

$result \leftarrow result(mod.\pi)$

return result

It can be calculated without overflow because of properties of the elements and limitation of a max value.

iii. Congruent Multiplication

Congruent multiplication is calculated in algorithm as follows. It is described as $\alpha * \beta(mod.\pi)$.

$\alpha,\ \beta$ are elements, π is a modulus

$result \leftarrow 0 + 0i$

while $\alpha \neq 0 + 0i$

do if $\alpha(mod.1+1i) \neq 0+0i$
then $tmp \leftarrow result + (1+1i) \times \alpha$
$result \leftarrow tmp(mod.\pi)$
$\beta \leftarrow (1+1i) \times \beta$
$\alpha \leftarrow \alpha/(1+1i)$
return result

Multiplication in line 5 can be operated because of the limitation of α and properties of multiplication by $1+1i$. Although it is required to add $result$ and $(1+1i) \times \alpha$ after calculating an element of $(1+1i) \times \alpha$ to add without overflow, the representation in line 5 is abbreviated. Hereafter abbreviation is often used. Multiplication in line 7 also can be operated because of the limitation of β and properties of multiplication by $1+1i$. These multiplications are not congruent. Therefore " $\times$ " is used to represent these multiplications.

iv. Congruent Division

Congruent division is calculated in algorithm as follows. It is described as $\alpha/\beta\ (mod.\pi)$.

α is dividend, β is divisor, π is a modulus
$\pi_{n-2} \leftarrow 0+0i$
$\pi_{n-1} \leftarrow 1+0i$
$result \leftarrow 0+0i$
$\xi \leftarrow \pi$
$\upsilon \leftarrow \pi$
$\zeta \leftarrow \xi(mod.\upsilon)$
$\kappa \leftarrow \xi/\upsilon$
while $\zeta \neq 0+0i$
do $\xi \leftarrow \upsilon$
$\upsilon \leftarrow \zeta$
$result \leftarrow \pi_{n-2} + \kappa * \pi_{n-1}(mod.\pi)$
$\pi_{n-2} \leftarrow \pi_{n-1}$
$\pi_{n-1} \leftarrow result$
$\zeta \leftarrow \xi(mod.\upsilon)$
$\kappa \leftarrow \xi/\upsilon$
$sign \leftarrow \beta * result(mod.\pi)$
if $sign = -1+0i$
then $result \leftarrow (-1+0i) * result(mod.\pi)$
if $sign = 0+1i$
then $result \leftarrow (0-1i) * result(mod.\pi)$
if $sign = 0-1i$
then $result \leftarrow (0+1i) * result(mod.\pi)$
$result \leftarrow \alpha * result(mod.\pi)$
return result

The algorithm from line 5 to line 16 is Euclidean Algorithm. The inverse of β is calculated from line1 to line 23. By multiplying it by α, $\alpha/\beta\ (mod.\pi)$ is calculated. The calculation in line 12 is abbreviated. Using congruent addition and congruent multiplication is required.

v. Congruent Exponentiation

Congruent exponentiation is calculated in algorithm as follows. It is described as $\alpha^n(mod.\pi)$. The n is integer.

α is a base, n is index, π is a modulus
$result \leftarrow 1+0i$
while $n > 2$
do if $n = 1$
then $result \leftarrow result * \alpha(mod.\pi)$
$\alpha \leftarrow \alpha * \alpha(mod.\pi)$
n ← n/2
return result

n/2 in line 7 is division of n by 2 in integer.

F. *Operation of* $\alpha^{a^2+b^2}(mod.\pi)$

For realizing implementing ECC onto Excel, congruent exponentiation $\alpha^{a^2+b^2}(mod.\pi)$ is required. a^2+b^2 in index can be with overflow if it is calculated directly. Therefore, it is calculated by using exponential law as follows.

$$\alpha^{a^2+b^2} = (\alpha^a)^a * (\alpha^b)^b$$

In congruent operation, right side is calculated as follows.

$$\left(\left(\alpha^a(mod.\pi)\right)^a(mod.\pi)\right) * \left(\left(\alpha^b(mod.\pi)\right)^b(mod.\pi)\right)(mod.\pi)$$

It enables $\alpha^{a^2+b^2}(mod.\pi)$ to be calculated without overflow.

V. Demonstrative Experiment

In this chapter, we conduct three experiments. First one is to confirm equivalence between a elliptic curve over Gaussian prime number $\rho = p_1 + p_2 i(\mathrm{p}_1, p_2 \neq 0)$ and that by prime number $p_1^2 + p_2^2$. The methods are to confirm the orders of cyclic groups with rational-point addition in Gaussian Integer and in integer are equal to each other and to confirm the orders of rational points of the elliptic curves are equal to each other. Second one is to confirm the safety of the ECC based on execution time required for brute force attack against secret keys generated in some max digit of Gaussian Integer and integer. Third one is to implement ElGamal Cryptosystem on Elliptic Curves with VBA of Excel and to confirm accurate encryption and decryption by it without overflow.

A. *Confirmation of Equivalence of Elliptic Curves*

This confirmation is conducted by programs in C language to calculate and output orders. In this experiment, Gaussian prime number $\rho = 4+5i$ and prime number $p = 41$ are selected as modulus because those satisfy $p_1, p_2 \neq 0$ and are not quite difficult to confirm it.

First, equivalence of two orders of cyclic groups is confirmed. Elliptic Curve's parameters a and b are selected at random. Coefficients of left side elliptic curve's rational points in Fig. 1 are the orders of cyclic groups with addition. To give an example, 4 in line 2: is the number. Fig. 1 shows the orders of cyclic groups with addition in Gaussian Integer are equal to that in integer.

Then, equivalence of two orders of elliptic curve rational point is confirmed. Elliptic curve's parameters a and b are changed from 0 to $p_1^2 + p_2^2 - 1$ respectively. The orders of elliptic curves over Gaussian Integer are shown in Fig. 2. The orders of elliptic curves over integer are shown in Fig. 3. Width,

depth and height in the figures are a, b and order respectively. The difference of Fig. 2 and Fig. 3 is shown in Fig.4 to confirm equivalence of Fig. 2 and Fig. 3. Fig. 4 shows the difference is 0. Moreover, a mean and a variance of Fig. 4 are both 0. These prove the equivalence of two orders of elliptic curve's rational points.

We can recognize that the equivalence of the elliptic curves by the above two confirmations. Therefore, we assume that it is possible to enhance safety without introduction of an arbitrary precision arithmetic.

```
a = -2-2i                                  a = 16
b =  0+0i                                  b = 0
p =  4+5i                                  p = 41
 0 : 2*( 0+0i,  0+0i) = (inf, inf)  0 : 2*( 0 ,  0) = (inf, inf)
 2 : 4*( 2+0i,  0+1i) = (inf, inf)  2 : 4*( 2 ,  9) = (inf, inf)
 4 : 4*( 4+0i, -1+3i) = (inf, inf)  4 : 4*( 4 , 13) = (inf, inf)
 5 : 2*( 0+4i,  0+0i) = (inf, inf)  5 : 2*( 5 ,  0) = (inf, inf)
 6 : 8*(-3-1i,  0-4i) = (inf, inf)  6 : 8*( 6 ,  5) = (inf, inf)
 7 : 8*(-2-1i,  2+0i) = (inf, inf)  7 : 8*( 7 ,  2) = (inf, inf)
 8 : 4*(-1-1i,  0-4i) = (inf, inf)  8 : 4*( 8 ,  5) = (inf, inf)
11 : 8*( 2-1i,  2-2i) = (inf, inf) 11 : 8*(11 , 20) = (inf, inf)
14 : 8*( 0+3i, -4+0i) = (inf, inf) 14 : 8*(14 ,  4) = (inf, inf)
27 : 8*( 0-3i,  0-4i) = (inf, inf) 27 : 8*(27 ,  5) = (inf, inf)
30 : 8*(-2+1i, -2-2i) = (inf, inf) 30 : 8*(30 , 16) = (inf, inf)
33 : 4*( 1+1i, -4+0i) = (inf, inf) 33 : 4*(33 ,  4) = (inf, inf)
34 : 8*( 2+1i,  0-2i) = (inf, inf) 34 : 8*(34 , 18) = (inf, inf)
35 : 8*( 3+1i, -4+0i) = (inf, inf) 35 : 8*(35 ,  4) = (inf, inf)
36 : 2*( 0-4i,  0+0i) = (inf, inf) 36 : 2*(36 ,  0) = (inf, inf)
37 : 4*(-4+0i,  3+1i) = (inf, inf) 37 : 4*(37 ,  6) = (inf, inf)
39 : 4*(-2+0i,  1+0i) = (inf, inf) 39 : 4*(39 ,  1) = (inf, inf)
```

Fig. 1 the Order of Cyclic Groups (left: Gaussian Integer, right: integer)

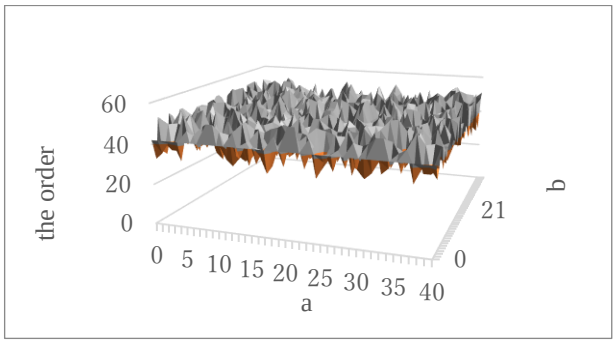

Fig. 2 the Orders of Elliptic Curve's Rational points over Gaussian Integer(p=4+5i)

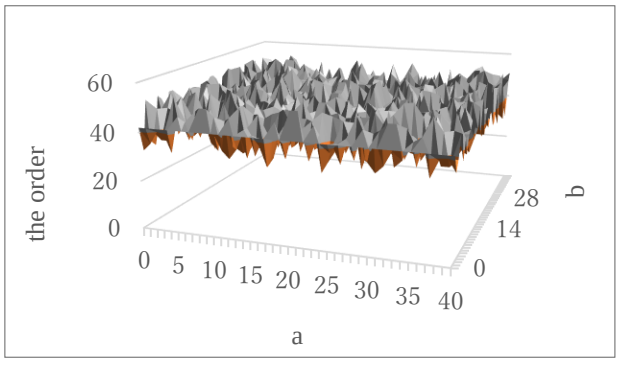

Fig. 3 the Orders of Elliptic Curve's Rational Points over Integer (p=41)

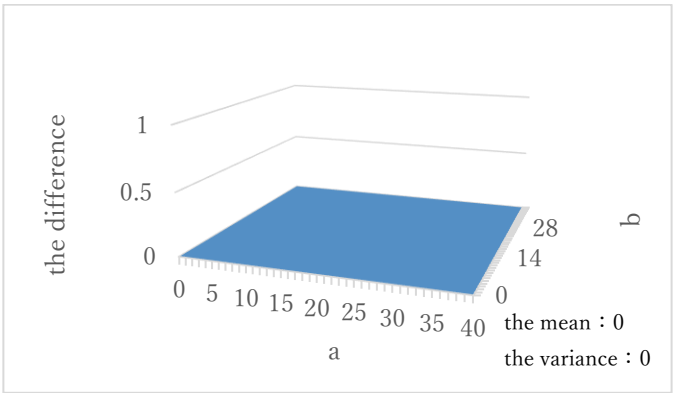

Fig. 4 the Difference of Two Orders of Elliptic Curve's Rational Points over Gaussian Integer and Integer

B. *Confirmation of the Safety*

The change of attack time in changing the digit of modulus is shown in Fig. 5. The vertical axis is the quotient of attack time by time required for one addition, or the number of times of addition required for attack. Reference [6] is referred to select parameters of elliptic curves. The values which make ECC safe in elliptic curves $y^2 \equiv x^3 - ax(mod.p)$ and of which p is the largest in a max digit are defined as parameters to calculate the order of elliptic curve's rational points easily. To give an example, if a max digit is two, 97 is the modulus in integer and 94+99i is the modulus in Gaussian Integer.

The solid line and the dashed line of Fig. 5 are attack time in Gaussian Integer and attack time in integer respectively. The value of two digits of solid line is approximately equal to that of four digits of dashed line. The value of three digits and the value of four digits of solid line are approximately equal to that of six digits and that of eight digits of dashed line respectively as well. These show the safety of Gaussian Integer approximately corresponds the twice digits of the safety of integer. Therefore this method can enhance the safety of ECC.

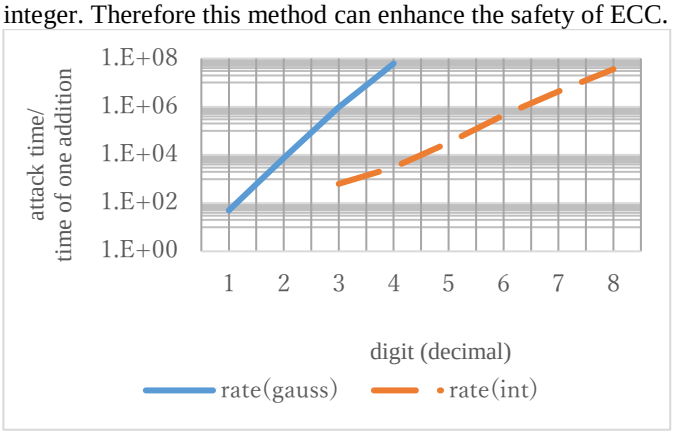

Fig. 5 Time of Attack for Some Digits of Modulus

C. *Confirmation of Operations on Excel*

The operations of functions implemented onto Excel are shown in Fig. 6, Fig. 7, Fig. 8 and Fig. 9. The value which is Gaussian prime number and of which real number or imaginary

number is close to the max value is defined as modulus. The other parameters a and b are defined at random.

The enc(plain text, public key, parameter) in line encryption in Fig. 6 is the function of encryption, and the decryp(cipher text, private key, parameter) in line decryption in Fig. 8 is the function of decryption. The functions of encryption and decryption are shown in Fig. 7 and Fig. 9 respectively.

In encryption, the first argument of enc is "1234567890" which is a plain text. The second argument is the value of "public key" and the third argument is the value of "parameter" respectively as well. The cipher text generated by this function is shown in line "cipher text" in Fig. 6.

In decryption, the first argument of decryp is the cipher text generated by the encryption. The second argument is value of line "private key" and the third argument is the value of "parameter". The decrypted text generated by this decryption is shown in line "decrypted text" in Fig. 8. This decrypted text is equal to the plain text in Fig. 6, and these functions are operated without any error. Therefore, ECC over Gaussian Integer can be operated without overflow.

encryption	enc(plain text, public key, parameter)
parameter	(10+0i,0+0i,2157241651175857360+2255773684465611297i)
public key	(-803719167948622100+-94838834634810116i,-1021919028780352562+681565794356246993i),,(-99542722922270479 0+820865727356792864i,-370817760315927499+-1195435618686159042i)
plain text	1234567890
cipher text	(451269122932874360+-296888417279811333i,-24951615675068071 5+441506572816209748i),,(-894076992567145704+125243099211790430 8i,57472032696026827 5+-1116265213372565357i)

Fig. 6 Encryption on Excel

encryption	enc(plain text, public key, parameter)
parameter	(10+0i,0+0i,2157241651175857360+2255773684465611297i)
public key	(-803719167948622100+-94838834634810116i,-1021919028780352562+681565794356246993i),,(-99542722922270479 0+820865727356792864i,-370817760315927499+-1195435618686159042i)
plain text	1234567890
cipher text	=enc(C5,C4,C3)

Fig. 7 Encryption on Excel (Appearance of Function)

decryption	decryp(cipher text, private key, parameter)
parameter	(10+0i,0+0i,2157241651175857360+2255773684465611297i)
private key	1111100100101010111010100001100011001100011111111011111110100110110011
cipher text	(451269122932874360+-296888417279811333i,-24951615675068071 5+441506572816209748i),,(-894076992567145704+125243099211790430 8i,57472032696026827 5+-1116265213372565357i)
decrypted text	1234567890

Fig. 8 Decryption on Excel

decryption	decryp(cipher text, private key, parameter)
parameter	(10+0i,0+0i,2157241651175857360+2255773684465611297i)
private key	1111100100101010111010100001100011001100011111111011111110100110110011
cipher text	(451269122932874360+-296888417279811333i,-24951615675068071 5+441506572816209748i),,(-894076992567145704+125243099211790430 8i,57472032696026827 5+-1116265213372565357i)
decrypted text	=decryp(C5,C4,C3)

Fig. 9 Decryption on Excel (Appearance of Function)

VI. CONCLUSION

In this paper, it is confirmed that the safety of ECC is enhanced with only integral type of Excel. Therefore the ECC safer than the ECC with only integral type is implemented onto

Excel without an arbitrary precision arithmetic library. In future, we would like to implement the ECC with an arbitrary precision arithmetic onto Excel and compare it with the ECC over Gaussian Integer.

REFERENCES

[1] S. Arita, R. Sakai, K. Tadagi, S. Cho, and K. Matsuo, "Angoriron to Daenkyokusen", Morikita Publishing Co., 2008 (in Japanese).
[2] S. Okuaki, "Implementation of ElGamal Cryptosystem on Elliptic Curves", Chiba Polytechnic College Bulletin, 2016.
[3] E. Mohamed, H. Elkamchouchi, "Elliptic Curve Cryptography over Gaussian Integers", International Journal of Computer Science and Network Security, Vol.9 No.1, pp413-416, 2009.
[4] A. Miyaji, "Cryptography in Algebraic Aspects", Nippon Hyoron Sha Co., 2012.
[5] T. Takagi, "Shoto Seisuron Kogi 2nd ed.", Kyoritsu Shuppan Co., 1971(in Japanese).
[6] Kenneth Irelad, Michael Rosen, "A Classical Introduction to Modern Number Theory 2nd ed.", ed. S. Axle, F. W. Gehring, and K. A. Ribet, Springer, 1990.