MESSAGE EMBEDDING USING COMPLEMENTARY ELLIPTIC CURVES FOR CRYPTOGRAPHIC USE

Hailiza Kamarulhaili, Putra Sumari & Teo Chun Yew

School of Mathematical Sciences, Universiti Sains Malaysia
11800 Penang, Malaysia

ABSTRACT. In this paper we introduced an elliptic curve which is called complementary elliptic curve, to embed messages and at the same time making use of all points on the curve as the embedding space. We showed how the embedding process takes place with the complementary curve and this includes point generation and message embedding, along with several modifications on El-Gamel algorithm. We have also made some modification in the encryption/decryption algorithm of El-Gamel to accommodate our embedding technique. As a result to this, by choosing relatively small prime number p compared to the existing technique, we managed to enlarge block of messages to be embedded and this modified embedding method has given a significant improvement in terms of processing and communication times. At the end of this paper we showed the simulation result.

KEYWORDS. Cryptographic, elliptic curves, message embedding

INTRODUCTION

The use of elliptic curves in cryptography has been known for years since 1985. It was first realized by a researcher named Neal Koblitz. Koblitz has found out that points generated by elliptic curves form a commutative group(Koblitz,1987 and Koblitz, 1988). Since then several algorithms on elliptic curves have been developed for cryptographic use as well as for other arithmetic purposes. One of the well known cryptographic algorithms is the El Gamel Elliptic curves cryptosystem (ElGamal, 1985). This algorithm was developed using the fact introduced by Neal Koblitz with El-Gamel method of encryption/decryption in which messages are embedded on elliptic curves before it was encrypted. However, this algorithm does not utilize all points on the respective curves. Thus, the existing system gives a limited hiding space. To overcome this problem we modified the existing system in a way that it could enlarged the hiding space as well as enlarging block of messages to be encrypted. In this paper we introduced a special type of curves called complementary curves to embed messages on.

EXISTING EMBEDDING PLAINTEXTS AND EL-GAMEL ELLIPTIC CURVE CRYPTOSYSTEM (ECC)

In this section we discussed the existing method of embedding plaintext (Silverman, 1986; Demyko, 1993 and Miller, 1986) and also the existing El-Gamel cryptosystem (Table 1) as well as the examples. The method for embedding plaintexts is described in Koblitz (1997) on an elliptic curve E defines over F_p where p is a prime greater then three. Let κ be a large enough integer so that we are satisfied with a failure probability of 1 out 2^{κ} of when we attempt to embed a plaintext message unit m; in practice $\kappa = 30$ or at worst $\kappa = 50$ should suffice.

Table 1. El-Gamal Elliptic cryptosystem

Public Information	**Encrypting Public Keys**	**Decrypting Secret Keys**
p is a prime number $E/F_p : y^2 = x^3 + ax + b$ with $a, b \in F_p$ $P = (x_1, y_1)$ base point of $E_p(a,b)$ $\#E_p(a,b) = n$	$Q = \alpha P = (x_2, y_2)$ is the encrypting public key of Alice	α is the decrypting secret keys of Alice

We suppose that our message units m are integers $0 \le m < M$. We also suppose that our finite field is chosen that $p > M\kappa$. We write the integers from 0 to $M\kappa - 1$ in the form $m\kappa + j$, where $1 \le j \le \kappa$, and we set up a 1-to-1 correspondence between such integers and a set elements of F_p. Thus, given m for each $j = 1,2,\mathrm{K}\,,\kappa$ we obtain an element x of F_q corresponding to $m\kappa + j$. For such an x, we compute the right side of the equation:

$$y^2 = f(x) = x^3 + ax + b,$$

and try to find a square root of $f(x)$. If we find y such that $y^2 = f(x)$, we take $P_m = (x, y)$. If it turns out that $f(x)$ is a non-square, then we increase x by 1 and try again. Provided that we find an x for which $f(x)$ is a square before j gets bigger than κ, we can recover m from the point

(x,y) by the formula $m = \left[\frac{\tilde{x}-1}{\kappa}\right]$, where $\tilde{x}$ is the integer corresponding to x under the 1-to-1 correspondence integers and elements of F_p. For $P = 271$, the point generation and message embedding is shown in Table 2.

Table 2. The point generation and message embedding for P = 271

M	j	X	$f(x) \equiv x^3 + x (\text{mod } 271)$	$\left(\frac{f(x)}{271}\right)$	$P_m = (x,y)$
0	1	1	2	1	(1,175)
1	1	31	12	-1	
	2	32	9	1	(32,268)
2	1	61	215	-1	
	2	62	181	-1	
	3	63	248	1	(63,178)
3	1	91	11	1	(91,163)
4	1	121	155	1	(121,56)
5	1	151	27	-1	
	2	152	71	-1	
	3	153	194	-1	
	4	154	151	1	(154,180)
6	1	181	171	-1	
	2	182	84	-1	
	3	183	5	1	(183,238)
7	1	211	198	1	(211,61)
8	1	241	70	1	(241,90)

Encryption

We claim that Bob is going to send the message M to Alice with $1 \le m \le p$:

1. Let $S = (x_3, y_3) \in E_p(a,b)$ as the point embedding by m. (m is embedded using existed method)
2. Bob chooses a random integer k with $2 \le k \le n-2$ where n is the number of points in $E_p(a,b)$ and computes $R = kP$ and $N = S + kQ$. So Bob have $(R,N) = ((x_4, y_4),(x_5, y_5))$
3. Bob sends the encrypted message $(R,N) = ((x_4, y_4),(x_5, y_5))$ to Alice.

Decryption

Alice receives the encrypted message that sends by Bob:

1. Alice receives $(R,N)=((x_4,y_4),(x_5,y_5))$
2. Computes $S=N-\alpha R$, Alice will get back $S=(x_3,y_3)$
3. Since m is the message embedded in point S, Alice can decrypt the message m.

EMBEDDING PLAINTEXTS USING COMPLEMENTARY CURVE

In this section we discussed the modified method of embedding plaintext and the modified El-Gamel algorithm. A method for embedding plaintexts on an elliptic curve is described in Koblitz (1987) and Koblitz (1988). However, only some of the points on an elliptic curve corresponds to the message that we want to embed. This subsection will provide a simple method to correspond to every point in $E_p(a,b)$ or $\overline{E_p(a,b)_v}$.

Now, we will describe how to imbed message on an elliptic curve with a curve and the corresponding complementary curve. But first we must know that the sum of the number of points on an elliptic curve and the number of points on the corresponding complementary curve is $\#E_p(a,b)+\#\overline{E_p(a,b)_v}=2p+2$.

Let m is correspond to the x-coordinate x_1 For every solution x_1 in Weierstrass equation $y^2=x^3+ax+b(\mathrm{mod}\,p)$, there are two possible values for y, namely y_1 (mod p) and p - y_1 (mod p). i.e. (x_1,y_1) and $(x_1,p-y_1)$. If the value of message $m<p$, then we pick the point (x_1,y_1) as the point correspond to message m. If $m\geq p$ and $m\equiv x_1 \bmod p$, then we pick the point $(x_1,p-y_1)$ as the point correspond to message m.

This means that each message $m\in\{0,1,\Lambda\ ,2p-1\}$ can be corresponds to x-coordinate equal to m of a point in one of the two curves (not to include the point at infinity, O in each curve). We have the following example depicted as follows to demonstrate the embedding technique for the $P=37$ (Table 3).

Let $E_{37}(2,9): y^2=x^3+2x+9$ and $\overline{E_{37}(2,9)_{19}}: 19y^2=x^3+2x+9$. Therefore, each message $m\in\{0,1,\Lambda\ ,73\}$ is corresponds to x-coordinate equal to m of a point in

$E_{37}(2,9)$ or $\overline{E_{37}(2,9)_{19}}$. Suppose we want to imbed a message $m = 43$, we know that there are two points where the x-coordinate equal to $43 \equiv 6 \bmod 37$ is in $\overline{E_{37}(2,9)_{19}}$, namely (6,17) and (6,20), since $43 > 37$, so the point (6, 20) is corresponds to message $m = 6$. Table 4 below shows the correspondence of every $m \in \{0,1,\Lambda\ ,73\}$ to a point in $E_{37}(2,9)$ or $\overline{E_{37}(2,9)_{19}}$.

Table 3. Points generation using Elliptic curve and the respective complementary curve

$E_{37}(2,9): y^2 \equiv x^3 + 2x + 9 (\bmod 37)$	$\overline{E_{37}(2,9)_{19}}: 19y^2 \equiv x^3 + 2x + 9 (\bmod 37)$
P = (0, 3),$2P$ = (33,23),$3P$ = (29,31) $4P$ = (11,17),$5P$ = (16,17),$6P$ = (5,25) $7P$ = (4, 9),8P = (26,32),$9P$ = (10,20) $10P$ = (31,22),$11P$ = (15,26),$12P$ = (1,30) $13P$ = (25,25),$14P$ = (2,13),$15P$ = (23,30) $16P$ = (13,30),$17P$ = (27,32),$18P$ = (21,32) $19P$ = (7,12),$20P$ = (9,33),$21P$ = (35,16) $22P$ = (35,21),$23P$ = (9, 4),$24P$ = (7,25) $25P$ = (21, 5),$26P$ = (27, 5),$27P$ = (13, 7) $28P$ = (23, 7),$29P$ = (2,24),$30P$ = (25,12) $31P$ = (1, 7),$32P$ = (15,11),$33P$ = (31,15) $34P$ = (10,17),$35P$ = (26, 5),$36P$ = (4,28) $37P$ = (5,12),$38P$ = (16,20),$39P$ = (11,20) $40P$ = (29, 6),$41P$ = (33,14),$42P$ = (0,34) $43P = O$	P = (3,11),$2P$ = (12, 9),$3P$ = (17,25) $4P$ = (36,30),$5P$ = (22,33),$6P$ = (18,32) $7P$ = (34,27),$8P$ = (20,22),$9P$ = (8, 1) $10P$ = (28, 2),$11P$ = (19,14),$12P$ = (32, 9) $13P$ = (24,30),$14P$ = (30,28),$15P$ = (6,20) $16P$ = (14,30),$17P$ = (14, 7),$18P$ = (6,17) $19P$ = (30, 9),$20P$ = (24, 7),$21P$ = (32,28) $22P$ = (19,23),$23P$ = (28,35),$24P$ = (8,36) $25P$ = (20,15),$26P$ = (34,10),$27P$ = (18, 5) $28P$ = (22, 4),$29P$ = (36, 7),$30P$ = (17,12) $31P$ = (12,28),$32P$ = (3,26),$33P = O$

Table 4. Plaintexts embedding using complementary curve

M	$P_m = (x,y)$	M	$P_m = (x,y)$	m	$P_m = (x,y)$	m	$P_m = (x,y)$
0	(0,3)	19	(19,14)	38	(1,30)	57	(20,22)
1	(1,7)	20	(20,15)	39	(2,24)	58	(21,32)
2	(2,13)	21	(21,5)	40	(3,26)	59	(22,33)
3	(3,11)	22	(22,4)	41	(4,28)	60	(23,30)
4	(4,9)	23	(23,7)	42	(5,25)	61	(24,30)
5	(5,12)	24	(24,7)	43	(6,20)	62	(25,25)
6	(6,17)	25	(25,12)	44	(7,25)	63	(26,32)
7	(7,12)	26	(26,5)	45	(8,30)	64	(27,32)
8	(8,1)	27	(27,5)	46	(9,33)	65	(28,35)
9	(9,4)	28	(28,2)	47	(10,20)	66	(29,31)
10	(10,17)	29	(29,6)	48	(11,20)	67	(30,28)
11	(11,17)	30	(30,9)	49	(12,28)	68	(31,22)
12	(12,9)	31	(31,15)	50	(13,30)	69	(32,28)
13	(13,7)	32	(32,9)	51	(14,30)	70	(33,23)
14	(14,7)	33	(33,14)	52	(15,26)	71	(34,27)
15	(15,11)	34	(34,10)	53	(16,20)	72	(35,21)
16	(16,17)	35	(35,16)	54	(17,25)	73	(36,30)
17	(17,12)	36	(36,7)	55	(18,32)		
18	(18,5)	37	(0,34)	56	(19,23)		

We can recover m from the point (x, y) by the value of y-coordinate, if y is less than prime p, the message is $m = x$, otherwise $m = x + p$. For example, we received a point (11,20) and we know that the point (11,20) have the other point (11,17) which the x-coordinate is equal to 11. Since the y-coordinate of (11,20) is greater than y-coordinate of (11,17) we know that the message $m = 11 + 37 = 48$.

Table 5. Table of keys for a cryptographic communication with modified ECC

Public Information	Encrypting Public Keys	Decrypting Secret Keys
p is a prime number $E/F_p : y^2 = x^3 + ax + b$ with $a,b \in F_p$ $\upsilon \in F_p, \upsilon \neq 0$ and $\left(\frac{\upsilon}{p}\right) = -1$ $\overline{E}/F_p : \upsilon y^2 = x^3 + ax + b$ $P = (x_1, y_1)$ base point of $E_p(a,b)$ $\overline{P} = (x_2, y_2)$ base point of $\overline{E_p(a,b)_\upsilon}$ $\#E_p(a,b) = n$ and $\#\overline{E_p(a,b)_\upsilon} = 2p + 2 - n$	$Q = \alpha P = (x_3, y_3)$ and $\overline{Q} = \overline{\alpha P} = (x_4, y_4)$ are encrypting public keys of Alice	α and $\overline{\alpha}$ are decrypting secret keys of Alice

Encryption

We claim that Bob is going to send the message M to Alice with $1 \le M \le p^2 - 2 + (p-1)^2$:

1. Bob computes $M = h(2p) + m$, with $0 \le h \le p-2$ and $0 \le m \le 2p-1$
2. Let $S = (x_5, y_5) \in E_p(a,b)$ or $\overline{E_p(a,b)_\upsilon}$ as the point embedding by m. (m is embedded using modified method)
3. $H \equiv \left(\left(x_5^2 - \upsilon\right)(h+1)\right) (\mathrm{mod}\, p)$
4. Bob chooses a random integer k with $2 \le k \le n-2$ (or $2 \le k \le 2p-n$) where n is the number of points in $E_p(a,b)$ (or $\overline{E_p(a,b)_\upsilon}$) and computes $R = kP (\text{or } R = k\overline{P})$ and $N = S + kQ (\text{or } N = S + k\overline{Q})$. So Bob have $(R,N) = ((x_6, y_6), (x_7, y_7))$
5. Bob sends the encrypted message $(R, N, H) = ((x_6, y_6), (x_7, y_7), H)$ to Alice.

Decryption

Alice receives the encrypted message that sends by Bob:

1. Alice receives $(R, N, H) = ((x_6, y_6), (x_7, y_7), H)$
2. If $\left(\frac{x_6^3 + ax_6 + b}{p}\right) = 1$, the message will decrypted using the curve $E_p(a,b)$ (Otherwise with $\overline{E_p(a,b)_\upsilon}$)
3. Computes $S = N - \alpha R$ (or $S = N - \overline{\alpha} R$), Alice will get back $S = (x_5, y_5)$
4. After obtained x_5 from S, computes $h = \left(\frac{H}{x_5^2 - \upsilon} (\mathrm{mod} p)\right) - 1$
5. Since m is the message embedded in x-coordinate of S which $m \equiv x_5 \bmod p$, then Alice can decrypt the message by compute $M = h(2p) + m$.

Following is the example of the modified ECC for the message "SECRET".

Bob send the first message block $M = 1804$ to Alice.

- $M = h(2p) + m \Rightarrow 1804 = 24(74) + 28$, there for $h = 24$ and $m = 28$.
- Since $m = 28$, we have point (28,2) and (28,35) to choose as the embedded point. But because of m is less than prime $p = 37$, so we choose (28,2) as the embedded point.
- Let $Q = (28,2) \in \overline{E_{37}(2,9)_{19}}$ the point of embedding message m.
- $H \equiv \left(\left(28^2 - 19\right)\left(24 + 1\right)\right) (\mathrm{mod}\, 37) \Rightarrow H \equiv 33 (\mathrm{mod}\, 37)$
- Because of $Q = (28,2) \in \overline{E_{37}(2,9)_{19}}$ so we only use the $\overline{E_{37}(2,9)_{19}}$ for the rest of the encrypting process.
- Choose randomly an integer $k = 17$ with $1 \le k \le 42$
- Computes $R = kP \Rightarrow R = 17(3,11) = (14,7)$, and
  $N = Q + k\alpha P \Rightarrow N = (28,2) + 17(6,20) = (28,2) + (8,36) = (3,11)$.
  So $(R, N) = ((14,7),(3,11))$.
- Bob transmit $(x_6, y_6, x_7, y_7, H) = (14,7,3,11,33)$.

Alice receives $(x_6, y_6, x_7, y_7, H) = (14,7,3,11,33)$

- Because of $\left(\frac{14^3+2(14)+9}{37}\right)=\left(\frac{2781}{37}\right)=-1$, so Alice decrypt the message with the curve $\overline{E_{37}(2,9)_{19}}$
- Alice build $(R, N) = ((14,7),(3,11))$
- Computes $Q = N - \alpha R = (3,11) - 15(14,7) = (28,2)$ and
- $m = 28$ is the message embedded in Q
- So Alice get the original message that sends by Bob $M = 24(74) + 28 = 1804$
- After decoding the integer 1804, Alice will get the original message 1804 = "SE".

Now, in the modified method (Table 5) we have managed to embed the word "SE" at a time for a small choice of prime P. For this particular message we can process two letters at a time with a small prime number P.

COMPARISON BETWEEN THE EXISTING AND THE MODIFIED EL-GAMAL (ECC)

In this section we discussed the comparison of the two, existing and the modified methods and also the simulation result. From the examples shown earlier, the existing method can only send a message from the range $0 \leq M \leq 8$ once at a time with the prime $p = 271$. But in the modified method, the message range is from $1 \leq M \leq 2663$ by only setting the prime $p = 37$. If we compare the points generated in Table 2 and Table 3, the difference is so obvious. In Table 2, the existing method manage to generate only nine points with relatively large prime p, and as we can see in the figure, not all M can generate point for all x, we have to keep choosing the x until the Legendre symbol is -1 and because of this reason we have to increase the value of x to less or equal to the prime 271. Unfortunately, in order to satisfy this, the existing method has only managed to generate nine points. This is way too small when compared to the modified method as it can be seen in the Table 3.

In the existing system, for a letter "S = 18" from the word "SECRET", we are not able to process it in one go, meaning 18 is out of the range, so we have to separate them two to "1" and "8". This means that we need to do it twice and this is time consuming. In the modified system we have managed to process two letters at a time due to bigger range of M. Therefore, the modified method has able us to generate far many points by only choosing a relatively small prime p. This shows that the modified method provides a bigger hiding space and at the same time the modified method of embedding technique managed to enlarge the size of plaintext with a smaller value of prime p to be processed.

In the modified method, we are using two groups of points as our embedding space and this gives us 74 points. When we compare this with the existing method that only have 9 points as embedding points, the existing method provide an insufficient embedding space for the system. Therefore our modified method provides ample hiding space and able to enlarge block of messages to be processed at a time and this contributes directly to the improvement of the system as a whole in terms of its processing time as well as the communication time.

SIMULATION RESULT

We have done a simulation on the existing and the modified methods. As it can be seen in the following diagrams, both encryption (Figure 1) and decryption (Figure 2) time has been improved when we enlarged the block of messages in the modified method.

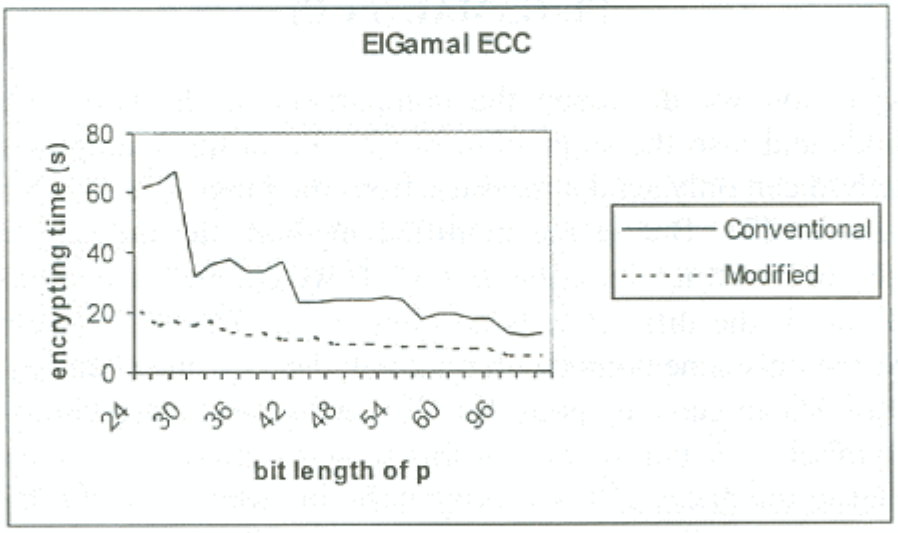

Figure 1. Encryption time comparison.

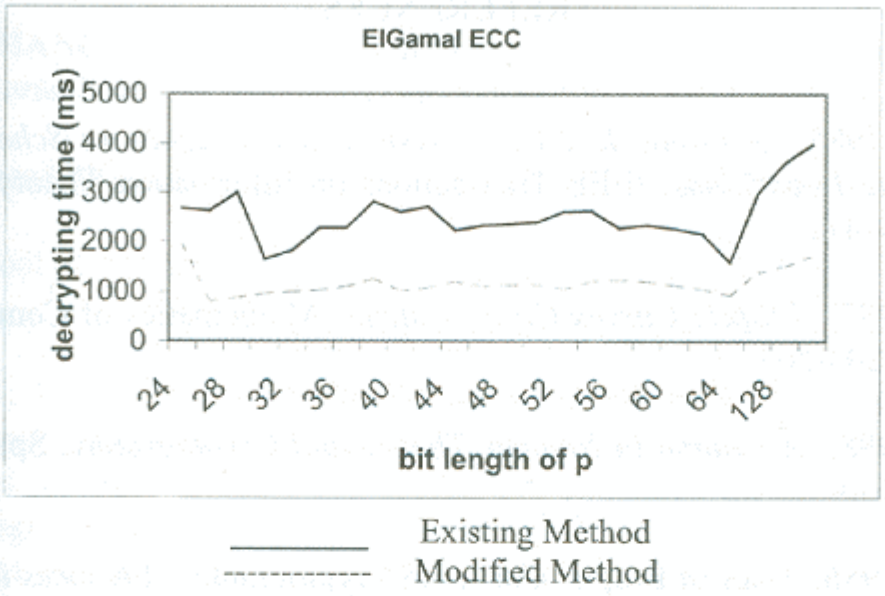

Figure 2. Decryption time comparison.

CONCLUSION

In this work we have modified the existing embedding method and modified the El-Gamel algorithm to manipulate these points, where we normally called this process as the encryption and decryption processes. In the modified technique, we only need to choose a small prime p to process messages and does not consume as much time as the existing technique with much bigger prime p. The modified method also gives much more hiding space compared to the existing method. This in return contributes to more reliable cryptosystem whereby it reduces probabilistic attack on the system as it has a bigger hiding space, besides from the improvement in the processing time.