REDUCING INVERSION PROCESSES OF POINT ADDITION TO SPEED UP ELLIPTIC CURVE CRYPTOGRAPHY

Kritsanapong Somsuk*

Department of Computer and Communication Engineering, Faculty of Technology,
Udon Thani Rajabhat University

บทคัดย่อ

งานวิจัยนี้นำเสนอการปรับปรุงขั้นตอนวิธีสำหรับระบบพิกัดสัมพรรคเพื่อเพิ่มความเร็วกระบวนการเข้ารหัสลับและถอดรหัสลับบนวิทยาการรหัสลับเส้นโค้งเชิงวงรี โดยการผกผัน กำลังสอง และการคูณคือกระบวนการหลักสำหรับทั้งการบวกของจุด และการเพิ่มเป็นสองเท่าของจุด จากทั้งสามวิธีพบว่าการผกผันใช้ทรัพยากรณ์การคำนวณสูงที่สุด จากงานวิจัยที่เกี่ยวข้องการคำนวณหา $8P$ เมื่อ P ถูกแทนจุดที่อยู่บนเส้นโค้ง จำเป็นต้องคำนวณการผกผันถึงสองครั้ง ในทางกลับกันขั้นตอนวิธีที่นำเสนอจำเป็นต้องคำนวณการผกผันเพียงหนึ่งครั้งเพื่อคำนวณหาค่า $8P$ จากผลการทดลองพบว่าขั้นตอนวิธีที่นำเสนอสามารถคำนวณหาค่า $8P$ ได้เร็วกว่าวิธีที่เปรียบเทียบ (วิธีที่เปรียบเทียบคือคำนวณหาค่า $2(4P)$) โดยขั้นตอนวิธีที่นำเสนอเหมาะสำหรับการคำนวณหาค่า $R = kP$ เมื่อ $k = 8^i + j$ และ j คือจำนวนเต็มที่มีขนาดเล็ก นอกเหนือจากนั้นสามารถนำขั้นตอนวิธีที่นำเสนอนี้ไปประยุกต์ใช้ร่วมกับขั้นตอนวิธีไตรภาค/ทวิภาคซึ่งเป็นขั้นตอนวิธีที่ปรับปรุงจากการคูณของจุดเพื่อเร่งความเร็ววิทยาการรหัสลับเส้นโค้งเชิงวงรี

คำสำคัญ: วิทยาการรหัสลับเส้นโค้งเชิงวงรี, การบวกของจุด, การเพิ่มเป็นสองเท่าของจุด การผกผัน, เวลาการคำนวณ

[*] ผู้ประสานงาน: กฤษณพงศ์ สมสุข

อีเมล์: Kritsanapong@udru.ac.th

Abstract

In this paper, a modified algorithm for an affine coordinate system is proposed to speed up encryption and decryption processes in elliptic curve cryptography (ECC). In fact, inversion, squaring and multiplication are the main processes for both point addition and point doubling. For three different processes, the inversion process has the highest cost. In the related works, to find $8P$ where P is represented as a point on curve, at least two inversion processes are needed to perform this task. On the other hand, the proposed method requires only one inversion process to compute $8P$. The experimental results show that the proposed method can perform the process faster than $2(4P)$ method. In general, the method is suitable for computing $R = kP$, where $k = 8^i + j$ and j is a small integer. Moreover, the proposed method also can be chosen to apply with Ternary/Binary Approach which is an improved point multiplication method to speed up ECC.

Keywords: Elliptic Curve Cryptography (ECC), point addition, point doubling, inversion, Computation Time.

Introduction

Nowadays, Communication technology is grown very fast and users can communicate each other very easy. Because, the communication channel is insecure, the private message may be trapped during transferring via the channel. Therefore, protecting the secret information becomes a major issue. Cryptography used to protect information by using encryption and decryption process is one of security techniques. In deep, there are two types of cryptography, symmetric key cryptography and asymmetric key cryptography. Symmetric key cryptography is the fast method when it is compared with the other. However, the problem is about the way to exchange the secret key between sender and receiver. In 1976, W. Diffie and M.E. Hellman (Diffie & Hellman, 1976) proposed the other cryptography, called asymmetric key cryptography (or public key cryptography). The main idea is about using two different keys. One which is disclosed to everyone is called public key and the other corresponding key which is kept secretly is called private key. With two different keys, public key cryptography can be chosen to solve the key exchanging problem. RSA is the best-well known public key algorithm proposed in 1978 (Rivest et al., 1978). This method is based on integer factorization. Although it is very difficult to break RSA, bits length for RSA should be at least 1024 bits. In 1980, Elliptic curve cryptography (ECC) (Koblitz, 1987) and (Miller, 1985) which is another public key cryptography was presented. The advantage of this method is security level which is very close to RSA, although bits length of ECC is very smaller than RSA. For example, (Prasana & Reddy, 2017) 160 bits length of ECC has the same security level with 1024 bit lengths of RSA. Therefore, ECC is well suitable with devices which have limited storages and low processing power such as smart phone. Furthermore, ECC can be applied with three different tasks, data encryption, digital signature and key agreement protocol (Elbirt, 2009).

Additionally, point addition and point doubling are the main processes to compute point multiplication: $Q = kP$, where P and Q are represented as the points on curve and k is integer. In fact, many techniques were proposed to speed up point multiplication such as binary method (Amara & Said, 2011) to compute $Q = kP$ by converting k as binary base for scanning bits of k from left to right or right to left and non-adjacent form (NAF) which is the signed binary representation having lower hamming weight in comparison to binary method (Karthikeyan, 2012).

Assuming, P is the generator point to find $Q = 8P$, usually three sequence point doublings are required, Q = (2(2(2P))). First, $2P = P + P$ is performed to find $4P = 2P + 2P$. Then, the last is the process to find $8P = 4P+4P$. That means, three inversion processes are included for the computation. In fact, inverse computing is the highest cost in comparison to squaring and multiplication computing. In 2006, the improved process for affine coordinate system to find $4P$ requiring only one inversion process was proposed (Ciet et al., 2006). Thus, the process to find $8P = 2(4P)$ requires two inversions, one is for $4P$ and the other is for point doubling.

In this paper, the new technique to reduce inversion processes of point addition to speed up $8P$ is introduced. The proposed algorithm requiring only one inversion process is modified from the process to find $4P$ in (Ciet et al., 2006). In fact, assuming j is represented as small integer, the proposed method suits for $Q = kP$ when $k = 8^i + j$. Furthermore, the method can be chosen to speed up point multiplication process because the inversion process is reduced.

Related Works

Elliptic Curve Cryptography

ECC is one of public key cryptography proposed by Miller and Koblitz. This method has very high security level although bits length is small. Therefore, it is suitable for communication devices which have limited storages and low processing power. In fact, ECC can be applied with several sets such as real number field, prime field ($E(Fp)$), p is prime number, and binary field. However, only prime field is focused in this paper. The equation of ECC over prime field is as following:

$$y^2 = x^3 + ax + b \bmod p \qquad (1)$$

Where a and b are the constants with

$$4a^3+27b^2 \neq 0 \bmod p \qquad (2)$$

In general, addition between two points on curve is the main core. Assuming $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ are represented as points on the curve. It is divided as two parts, point addition and point doubling as follow:

Point addition: ($P \neq Q$), R = $(x_3, y_3) = P + Q$

$$m = \frac{y_2 - y_1}{x_2 - x_1} \tag{3}$$

$$x_3 = m^2 - x_2 - x_1 \bmod p \tag{4}$$

$$y_3 = m(x_1 - x_3) - y_1 \bmod p \tag{5}$$

From (3) – (5), performing a point addition takes 1 inversion, 1 squaring and 1 multiplication.

Point doubling: ($P = Q$), $T = (x_4, y_4) = P + P$

$$m = \frac{3x_1^2 + a}{2y_1} \tag{6}$$

$$x_4 = m^2 - 2x_1 \bmod p \tag{7}$$

$$y_4 = m(x_1 - x_4) - y_1 \bmod p \tag{8}$$

From (6) – (8), performing a point doubling takes 1 Inversion, 2 squarings and 1 multiplication.

Improving point addition and point doubling

Recently, many researchers improved the equation in both of point addition and point doubling to reduce computation time. Assigning I, S and M are represented as inversion, squaring and multiplication, in order.

In 2003, the modified equation to compute $2P + Q$ (Eisentrager et al., 2003) was proposed by leaving some process to compute y. Generally, it usually takes 2I, 2S and 2M when the process is begun with $R = P + Q$ and is followed by $R + P = 2P + Q$, two point additions are included. Then, it becomes taking 2I, 3S and 2M whenever $2P$ is calculated before computing $2P + Q$. However, the modified equation takes only 2I, 3S and 1M.

Ciet et al. (2006) proposed the improved equations to reduce modular inversion. Although, processes of modular multiplication are increased, one inversion process is more expensive than many multiplications. In addition, the improved equation to find $2P + Q$, $3P$, $3P + Q$ and $4P$ were also introduced in that time. In fact, $4P$ takes 1I, 9S and 9M.

Moreover, the improved approach which is called Ternary/Binary Approach was proposed to speed up point multiplication. A recursive function is included in the process. However, k is divided by 6 whenever the function is recalled. Assuming $r = k \bmod 6$, the result is distinguished as 4 cases as follows:

Case 1: $r = 0$ or 3, Return $3(\frac{k}{3})P$

Case 2: $r = 2$ or 4, Return $2(\frac{k}{2})P$

Case 3: $r = 1$,

$$l = \frac{k-1}{6}$$

Return $2((3l)P) + P$

Case 4: $r = 5$

$$l = \frac{k+1}{6}$$

Return $2((3l)P) + P$

Algorithm: $4P$

Input: $P = (x_p, y_p)$

```
1. $A_1 = x_p$
2. $B_1 = 3x_p^2 + a$                                S
3. $C_1 = y_p$
4. $A_2 = 2B_1^2 - 8 A_1C_1^2$                       SSM
```

```
    5. $B_2 = 3xA_2^2 + 16aC_1^4$                    SSM
    6. $C_2 = B_1(4A_1C_1^2 - A_2) - 8C_1^4$         M
    7. $A_3 = B_2^2 - 8A_2C_1^2$                     SSM
    8. $C_3 = B_2(4A_2C_2^2 - A_3) - 8C_2^4$         SM
    9. IF $C_1C_2 = 0$ then
    10. Return $\varnothing$
    11. End IF
    12. $I = (4C_1C_2)^{-1}$                         MI
    13. $x_q = A_3I^2$                               SM
    14. $y_q = C_3I^3$                               MM
Output: $Q = (x_q, y_q)$
```

Furthermore, in 2011, the method (Li & Feng, 2011) to reduce inversion process to find 3P+Q was introduced. Usually to find $3P + Q$, three processes of point additions and point doublings are required, thus 3I must be also computed. For Li & Feng's method, although more multiplications and squarings are required, only one inversion is implemented, it takes only 1I, 3S and 16M to compute 3P+Q.

The Proposed Method

The aim of this research is to propose the new process to find $8P$ by reducing the inversion process. In fact, the proposed method is suitable for some point multiplications which have to compute continuously point doublings at least three times. In general, $4P$ using (Eisentrager et al., 2003) and point doubling are required, in order. Therefore, 2I, 11S and 11M must be executed. However, only 1 inversion will be required to calculate $8P$ for the proposed method. Assuming $Q = (x_q, y_q) = 4P$, $R = (x_r, y_r) = 8P$, it implies that

$$
\begin{aligned}
M &= \frac{3x_q^2+a}{2y_q} \\
&= \frac{3x_q^2+a}{2(C_3 I^3)} \\
&= \frac{3x_q^2+a}{(2C_3)I^3} \\
&= (3x_p^2 +a)(2C_3)^{-1}(I^3)^{-1} \\
&= (3x_p^2 +a)(2C_3)^{-1}(I^{-1})^3
\end{aligned}
$$

Because, $I = (4C_1C_2)^{-1}$, then $(4C_1C_2) = I^{-1}$. Thus, it implies that

$$m = (3x_p^2 +a)(2C_3)^{-1}(4C_1C_2)^3$$

Assigning the values of A_1, B_1, C_1, A_2, B_2, C_2, A_3 and C_3 for the proposed method are similar to all of them which are during step 1 to step 11 in algorithm 4*P*. Therefore, the remaining steps to find 8*P* are as following:

```
Algorithm: The proposed method (8P)
Input: $P = (x_p, y_p)$
    1. $H = 4C_1C_2$                                              M
    2. $J = 2C_3$
    3. $I = (HJ)^{-1}$                                            IM
    4. $I_1 = IJ$,     $(I_1 = (4C_1C_2)^{-1})$                   M
    5. $I_2 = IH^4$,    $(I_2 = (2C^3)^{-1}H^3)$                  MSS
    6. $x_q = A_3I_1^2$                                           SM
    7. $y_q = C_3I_1^3$                                           MM
    8. $K = (3x_p^2 +a)$                                          S
    9. $L = KI_2$                                                 M
    10. $x_r = L^2 - 2x_q$                                        S
    11. $y_r = L(x_q - x_r) - y_q$                                M
Output: $R = (x_r, y_r) = 8P$
```

In fact, the computation costs in the remaining steps are 1I, 5S and 9M. Therefore, the total costs are 1I, 13S and 14M when they are combined with the other parts during step 1 to step 11 in algorithm 4*P*.

Example: Assigning $P = (31, 5)$ is the point on curve: $y^2 = x^3 + 29x + 3 \bmod 41$, finding $R = 8P$ using the proposed method

Sol: From steps 1 – 11 of algorithm 4*P*, $A_1 = 31$, $B_1 = 1$, $C_1 = 5$, $A_2 = 33$, $B_2 = 35$, $C_2 = 35$, $A_3 = 3$ and $C_3 = 6$

Therefore, the process to find 8*P* using the proposed method can be performed as follows:

1. $H = 4*5*35 \bmod 41 = 3$
2. $J = 2*6 \bmod 41 = 12$
3. $I = (3*12)\text{-}1 \bmod 41 = 8$
4. $I_1 = 8*12 \bmod 41 = 14$
5. $I_2 = 8*34 \bmod 41 = 33$
6. $x_q = 3*\ 142 \bmod 41 = 14$
7. $y_q = 6*143 \bmod 41 = 23$
8. $K = (3*142 + 29) \bmod 41 = 2$
9. $L = 2*33 \bmod 41 = 25$
10. $x_r = 252 - 2(14) \bmod 41 = 23$
11. $y_r = 25(14 - 23) - 23 \bmod 41 = 39$

Therefore, $R = (23, 39)$ is the result point requiring only one inversion process.

In fact, assume $i, j \in Z$ and j is a small integer, the proposed method is suitable for the following equation: $Q = kP$, where $k = 8^i + j$, because,

$$8^i * P = (\underbrace{8(8(8(\cdots}_{i}\ (8P)))))$$

Experimental Result

In this section, the proposed method (8*P*) is chosen to compare about the speed to finish the process 8*P* with the compared method using 2(4*P*). 160 bits length of ECC is selected for the implementation. In addition, number of times to compute 8*P* are during 1000 -10000. However, Java Language and BigInteger Class (Deligiannidis, 2015) are selected to implement the proposed method and the compared one. In order to control the same settings, all experiments were conducted on 2.20 GHz an Intel® Core i7 with 4 GB memory.

The experimental results show that the proposed method can finish *k* loops of 8*P* faster than the compared method. Furthermore, the average computation time is decreased about 11%. In fact, if the compared method is chosen to find 8*P*, then the total costs which are for 4*P* and point doubling are 2I, 11S and 11M. However, the proposed method takes only 1I, 13S and 14M. Although, both of squaring and multiplication processes of the proposed method are larger than the compared technique. The costs are only a little in comparison to the cost of inversion process.

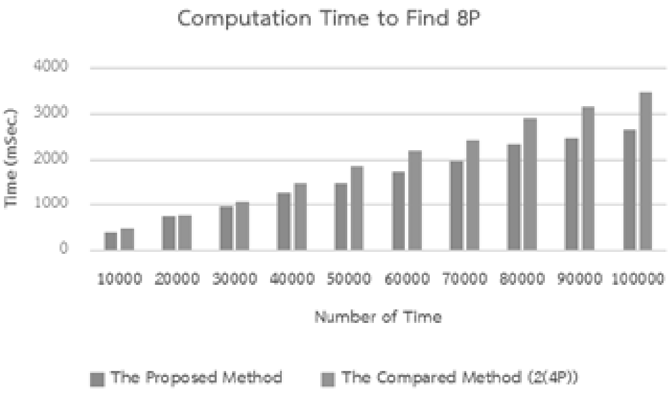

Fig. 1 Computation Time to find 8*P* by using The Proposed Method and Compared Method

Furthermore, the proposed method can be chosen to combine with Ternary/Binary Approach to speed up the process. In fact, some inversion processes can be removed whenever at least three points doubling are performed continuously.

For example, assuming *k* is equal to 52360, the beginning process to find *kP* by using Ternary/Binary Approach is as follows:

Step 1: (52360 mod 6 = 4), 52360 = 2(26180)*P*

Step 2: (26180 mod 6 = 2), 26180 = 2(13090)*P*

Step 3: (13090 mod 6 = 4), 13090 = 2(6545)*P*

In fact, for the first three steps to find 52360*P* by using Ternary/Binary Approach, three points doubling are required continuously. Therefore, if the proposed method is combined with Ternary/Binary Approach, then three points doubling can be replaced by the proposed method as follows:

$$52360 = 8(6545)P$$

Therefore, only one step is required instead of three points doubling.

Conclusion

In this research, the modified equation of affine coordinate system to speed up 8*P* is introduced. The key is to reduce the inversion process, although both of squaring and multiplication processes are a little increased. The reason is that inversion process takes very expensive cost when it is compared with squaring and multiplication processes. In fact, number of squarings and multiplications are larger than the compared method (2(4P)) about 2 and 3 processes, respectively. In addition, the experimental results show that the proposed method can finish process faster than the compared method. In deep, ECC is more suitable than RSA to be chosen to apply with low power and limited communication devices, because it uses only small key length with high security level.

References

Diffie, W., & Hellman, M. E. (1976). *New directions in cryptography*, IEEE Transactions on Information Theory, vol. 22, pp. 644–654.

Rivest, R. L., Shamir, A., & Adleman, L. (1978). *"A method for obtaining digital signatures and public key cryptosystems"*, Communications of ACM, vol. 21, pp. 120 – 126.

Koblitz, N. (1987). *Elliptic Curve Cryptosystems*. Mathematics of Computation, vol.48, pp. 203-209.

Miller, V. S. (1985). *Use of Elliptic Curves in Cryptography, Advances in Cryptology"*. Proceedings of CRYPTO85, LNCS-218, pp. 417-426.

Prasana, Y. L., & Reddy, E. M. (2017). A *Theoretical Study of Elliptic Curve Cryptography for Location Based Services*, Proceeding of 7th International Conference on Big Data Analytics and Computational Intelligenc, Chirala, India. pp.404-407.

Elbirt, A. J. (2009). *Under Standing and Applying Cryptography and Data Security*. USA: Auer-bach Publications.

Amara, M., & Said, A. (2011). *Elliptic Curve Cryptography and Its Applications,* Proceeding of 7th International Workshop on Systems, Signal Processing and their Applications, Tipaza, Algeria. pp. 247-250.

Karthikeyan, E. (2012). Survey of Elliptic Curve Scalar Multiplication Algorithms, *Int. J. Advanced Networking and Applications, 4*, 1581 – 1590.

Ciet, M., Joye, M., Lauter, K., & Montgomery, P. L. (2006). *Trading Inversions for Multiplications in Elliptic Curve Cryptography, Designs, Codes and Cryptography*, vol. 39, 189-206.

Eisentrager, K., Lauter, K., & Montgomery, P. L. (2003). *Fast Elliptic Curve Arithmetic and Improved Weil Pairing Evaluation*, Lecture Notes in Computer Science, vol.2612, pp. 343-354.

Li, Y., & Feng, L. (2011). *Overview of Scalar Multiplication in Elliptic Curve Cryptography*. Proceedings of International Conference on Computer Science and Network Technology, pp. 2670 – 2673, Harbin, China.

Deligiannidis, L. (2015). *Elliptic curve cryptography in Java*. Proceedings of International Conference on Intelligence and Security Informatics, pp. 193 – 193, MD, USA.