Send Orders for Reprints to reprints@benthamscience.net

Fast Elliptic Curve Algorithm of Embedded Mobile Equipment

Lihong Zhang*, Shuqian Chen and Yanglie Fu

School of Computer Engineer, Huaihai Institute of Technology, Lianyungang, Jiangsu, China

Abstract: Selection Algorithm and Generation Algorithm of elliptic curves have been the focus of research and hotspot of the Elliptic Curve Cryptosystem. This paper discusses a random elliptic curve realization method of Embedded Mobile Equipment, the SEA algorithm and its improved algorithm from Elliptic Curve's selection, Elliptic Curve's structure and Elliptic Curve's generation. Ensuring that the embedded system in the security situation goes through invariable situation causes the embedded system to realize a fast elliptic curve realization method, which enhances the efficiency of embedded system.

Keywords: Elliptic curve cryptosystem, SEA, table, fast algorithm.

1. INTRODUCTION

Embedded Mobile Equipment's processor mostly is ARM9, Strong ARM or Xscale and so on. The processor's processing speed and the operational capability are limited. Most embedded mobile equipments offer many kinds of data access methods, the use of Wi-Fi access methods being more popular. But at present, in the Wi-Fi encryption algorithm which is suited to the characteristics of embedded mobile equipment, computing secure access is less. The Elliptic Curve Cryptography public key system is a more secure cryptosystem. However, the selection and calculation of elliptic curves would be unbearable for embedded devices. The complex regular operation, combined with the characteristics of the embedded system, can use pre computing and establish form to reduce system's computation load. At the same time, a simplified algorithm is used to improve system's operating efficiency. But the processor speed of the textile machine's controller and image recognition speed and accuracy determine which of these scenarios in real-world applications are limited. In order to speed up the processing speed, the use of high speed ARM processor and the special-purpose DSP chip, simultaneously makes the algorithm achieve a better detection. This article discusses the weft detection method from the perspective of image recognition algorithms, and does not include the discussion on implementation in DSP processor, ARM and DSP control process.

2. SELECTION OF ELLIPTIC CURVES

Elliptic curve usually refers to a plane curve that satisfies the Weierstrass equation:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (1)$$

*Address correspondence to this author at the School of Computer Engineer, Huaihai Institute of Technology, Lianyungang, Jiangsu, China;
Tel: +8651885895390; Fax: +86051885895386; E-mail: zlh2341@126.com

In order to ensure the selected security of elliptic curve, we should pay attention to the following question:

(1) In order to avoid the MOV [1] algorithm attack, the selection must be non-super singular elliptic curve;

(2) In order to avoid the SSAS [2] algorithm attack, the selection cannot be the " anomalous " elliptic curve on the prime number field $GF(p)$;

(3) Based on the Weil [3] drop theory, the selection of the characteristic of 2's Complex Domain is avoided and should be selected p in domain $GF(2^p)$ of the elliptic curve as a prime number.

Based on the above secure request, there are two typical security elliptic curves. One is whose characteristic of finite field is prime, when q=p, $GF(q)=GF(p)$, defined in the prime domain $GF(p)$, its Weierstrass equation is

$$y^2 = x^3 + ax + b, a, b \in F \quad (2)$$

The other is an elliptic curve whose finite field characteristic is 2, when $q=2^n$, and elliptic curves in the Binary finite field $GF(q)$, its Weierstrass equation is

$$y^2 + xy = x^3 + a_2x^2 + a_6, \quad (3)$$

Note that here n is prime and 2n should be large enough to counter the Weil descent algorithm, But these two kinds of curves have the shortcoming of selection difficulty and running speed. At present, these problems are solved mainly through selection of the appropriate or a prime order elliptic curve of prime order parameter in a limited parameter group of elliptic curves.

3. CONSTRUCTION OF ELLIPTIC CURVES

Construction of an elliptic curve's method usually involves Random curve [4] and Complex Multiplication [5]

1874-1290/13

(CM). Because the CM method of constructing elliptic curves has some additional characteristic, from the security angle, these characteristics are considered a potential threat, but the random curve is obtained completely randomly generated, that is considered the safer elliptic curve. To simplify the calculation of random elliptic curve, an isomorphic curve is constructed in the same domain, and according to the characteristics of the domain. In the prime field *Fp*, *K* is defined in the domain, characteristic *p>3*, *Eq.1* can be simplified as $y^2 = x^3 + ax^2 + b, a, b \in F$, with $\Delta = 4a^3 + 27b^3 \bmod p \neq 0$ acquisition of the curve is mainly a finite group's Order calculation of an elliptic curve. In this aspect, there are many classical algorithms, such as the School algorithms, SEA algorithm, mold polynomial algorithm, Elkies algorithm, Atkin algorithm and so on. These algorithms in practical applications meeting the needs of the system bring lots of improvement.

4. ELLIPTIC CURVE GENERATING ALGORITHM

On the preceding analysis, the selection of a safe elliptic curve actually depends on how fast elliptic curve order is computed. Among all the algorithms, SEA algorithm is the most representative, combination of Elkies and Atkin brings improvements, and calculates the set of possible values, then the Chinese Remainder Theorem and Shank's Method(baby-step giant-step,BSGS) are used to calculate order. In the prime finite field *Fq*, the order is *#E(Fq)*, the process of the algorithm [6] is summarized as follows:

Input :GF(q) and E

Output: *#E (F(q))*

(1) $l \leftarrow 3$, $E \leftarrow \{(t \bmod 2,2)\}$, $A \leftarrow \{\}$

(2) for $M = 2; M < 4\sqrt{p}$

compute $f(q)=gcd(x^p - x, \phi_l(x, j(E)))$

if *f(q)≠1 Then Choose* Elkies

else *Choose* Atkin

$M \leftarrow M \cdot l$ *get Prime l*

(3) for *(j=; j<=i; j++)*

$L_j = \frac{M}{l_j}$; $y_j = L_j^{-1} \bmod l_j$

Compute $t = t_1 L_1 y_1 + t_2 L_2 y_2 + \cdots + t_{1i} L_i y_i$

(4) output #E (F(q)) =q+l-t

The main problem of the algorithm is the complexity in the calculation of the *l* polynomial model and the division polynomial, at the same time selection of the curve success ratio is lower [7].

5. FAST GENERATING ALGORITHM

In embedded systems, computing speed of add and subtract is faster than other operation. Multiplication mostly can complete in the Single-cycle, and supports 64-bit multiply. In order to adapt the characteristics of embedded systems, we need to improve the SEA algorithm. In above SEA algorithm description, we will improve the calculation of elliptic curves in the order of finite groups, thus will enhance the efficiency of the system

5.1. Improved Curve Selection Algorithm

After defining the finite domain $GF(q)$, according to the characteristics of embedded systems, it becomes appropriate to reduce the safety requirements, request parameter q becomes greater than 160bit, its password strength is equivalent to 1024bit RSA. Based on this, randomly generated parameter q, establishes the elliptic curve on this domain. According to the need of Generality of modular polynomials for solving *#E(F(q))* in the SEA algorithm, convert its definition [8], adopt the pretreatment method, and store the calculation results of polynomial model in the Flash or the main memory, and look-up table to obtain calculation results. Using these methods, the calculation time can be reduced. If the randomly generated curve parameter does not meet the requirements, curve parameter needs to re-select. So we need to set flag in the system, and alternately change the parameters a and b. For example: when the flag is 0, change the parameter a, then set flag to 1; at next selection, the flag is 1, change the parameter b, then set flag to 0, and so on , avoid select repeated parameters. Using these measures can improve the speed of the algorithm. The improved algorithm is as follows:

Input: *Random (a, b)*

Output: $y^2 = x^3 + ax + b, a, b \in F$

(1) *Random (a, b)*

(2) if *E=Special Curve* Then *return (1)step*

(3) if *Prime l* then *Elkies* else *Atkin*

(4) Compute *t*, *#E=q+l-t*

(5) if *non-Prime #E*

if flag =0 change a and flag←1

return *(1)step*

else

change b and flag←0 return *(1)step*

else

Output *Curve Equation*

5.2. Improved Modulus Algorithm

In the elliptic curve generating operation, the module computing spends longer time than any other computing. The module computing speed can effectively improve the overall computing efficiency of embedded systems, as mentioned above. The use of embedded memory look-up table method can save considerable computing time. In each encryption, decryption tasks, we establish a temporary table. Because of memory component cost's reduction, this plan's implementation possibly becomes gradual. Considering the Flash's reading speed, when forming initialization, we can map the form in the main memory (SDRAM or DDR) to speed up the data's reading speed.

When finite field Fq is determined, Pre-compute 2^l mod p, the result stores in system's form, and then modulus gets completed and demands the same order-mode. Modular multiplication can be divided into two steps: quadrature and modulo. The modulus operator will be decomposed into $\sum_{i=0}^{n-1} c_i 2^i$ by the modulus, just for each c_i which is not equal 0, the model results can be accumulated the pre-computed value 2^l mod P ,fill the pre-computed values in the memory table, using for reference table.

Suppose $X = \sum_{i=0}^{n-1} x_i Y^i, x_i \in [0, Y]$

$$X \equiv [\sum_{i=0}^{n-1} x_i Y^i + (x_{n-1} Y^{n-1} \bmod M)] \bmod M \quad (4)$$

$$M = \sum_{i=0}^{n-1} m_i Y^i, m_i \in [0, Y] \quad (5)$$

$$Z_1^j = j \cdot Y^a \bmod M \text{ or } \sum_{i=0}^{a-1} z_{1,j}^{(j)} Y^i \quad (6)$$

$$Z_0^j = j \cdot Y^{a-1} \bmod M \text{ or } \sum_{i=0}^{a-1} z_{0,j}^{(j)} Y^i \quad (7)$$

Where: $j = (1, 2, \cdots, Y-1)$; $Z_1^{(j)}$ is M 's $a+1$-order mode, recorded as Z_1 , $Z_0^{(j)}$ is M 's a -order mode, recorded as Z_0 。

According to the above definition of X and M

If $x_{a-1} < m_{a-1}$

$X \bmod M = X$

If $x_{a-1} = m_{a-1}$

$$X \bmod M = \begin{cases} X & M > X \\ 0 & M = X \\ X - M & M < X \end{cases}$$

If $x_{a-1} > m_{a-1}$

$$X \bmod M = \begin{cases} T & M > T \\ T - M & M < T \end{cases}$$

Where: $T = \sum_{i=0}^{a-2} x_i Y^i + Z_0^{(x_{a-1})}$

1. Fast modulus algorithm

First construct M 's $a+1$-order and a -order mold table, in each encryption and decryption of messages, built table should be completed before modulo operation, can build tables in the following order:

$$(X \times 2^n + Y) \bmod N = [(X \times 2^n) \bmod N + Y] \bmod N \quad (8)$$

$$\begin{aligned} &(X_1 \times 2^n + X_2) \bmod N \\ &= [(X_1 \times 2^n) \bmod N + (X_2 \bmod N)] \bmod N \end{aligned} \quad (9)$$

Then set X is N -bit Y -nary number, result of modulo is a A -bit Y - nary number, which may be greater than M , and may also be less than M , the algorithm is as follows:

Modular1 (X,M,Z)

for i=n-1 to a

$a_i = x_i$

$x_i = 0$

$X = X + Y^{n-1-a} Z_1^x$

The final result is A -bit Y - nary number, and is stored in the X .

Finally, take the same-modes to the results obtained in the previous step, the algorithm is as follows:

Modular2 (X,M,T)

for i=n-1 to a-1

if $x_{a-1} < m_{a-1}$

then $X \bmod M = X$

if $x_{a-1} = m_{a-1}$

then

if M>X

then X mod M=X

if M=X

then X mod M=0

if M<X

then X mod M=X-M

if $x_{a-1} > m_{a-1}$

then

if M>T

then X mod M=T

if M<T

then X mod M=T-M

2. Modular inversion

In the Modular inversion calculation, usually inverse operation is converted to multiplication operation, which reduces the inverse frequency to achieve the purpose of improving the efficiency. In the field Fq, for the given points p (x_1,y_1) and Q (x_2,y_2), the calculation of the size of 3P+Q computation is particularly important. In literature [9] pointed out: Assuming that $(x_3, y_3) = 2P$, $(x_4, y_4) = P + Q$. Algorithm is as follows:

$$\lambda_1 = (3x_1^2 + \alpha)/(2y_1) \; x_3 = \lambda_1^2 - x_1 x_2, y_3 = \lambda_1(x_1 - x_3) - y_1 \quad (10)$$

$$\begin{aligned} &\lambda_2 = (y_2 - y_1)/(x_2 - x_1) \\ &x_4 = \lambda_2^2 - x_1 - x_2, y_4 = \lambda_2(x_1 - x_4) - y_1 \end{aligned} \quad (11)$$

$$\begin{aligned} &\lambda_3 = (y_4 - y_3)/(x_4 - x_3) \\ &x_5 = \lambda_3^2 - x_3 - x_4, y_5 = \lambda_3(x_3 - x_5) - y_3 \end{aligned} \quad (12)$$

Introduction of common factor λ_c, λ_c is the $\lambda_1, \lambda_2, \lambda_3$ denominator least common multiple number, and is known as term simplest formula. For ease of calculation, suppose A_1, A_2, A_3 to separately express $\lambda_1, \lambda_2, \lambda_3$ denominator, B_1, B_2, B_3 to separately express $\lambda_1, \lambda_2, \lambda_3$ numerator, obviously $\lambda_1, \lambda_2, A_1, A_2, B_1, B_2$ is known, λ_3, A_3, B_3 need to calculate. Assuming that

$$\sigma = (A_1B_2 + A_2B_1) + (A_2B_1 - A_1B_2) - A_2(A_1A_2)^2 \quad (13)$$

$$A_3 = \sigma / (A_1A_2)^2 \ (12) \ \lambda_3 = B_3(A_1A_2)^2 / \sigma \quad (14)$$

Inverse operation to convert the multiplication, that is to strike λ_c, through-type (8) ~ (10) calculated (x_5, y_5), from the above steps, Inverse items are the only λ_c, which is inverse from the 2 times (λ_c and λ_3) and is reduced to 1 (λ_c) and then transformation and substitution B_3 which is in type (9), are obtained:

$$\lambda_3 = (A_1A_2)^3 \lambda_c (x_1 - x_3)(\lambda_1 - \lambda_2) \quad (15)$$

Transform equation (6), we can obtain x_5.

$$x_5 = (\lambda_3 + \lambda_2)(\lambda_3 - \lambda_2) - x_3 + x_2 + x_1 \quad (16)$$

By calculating $\lambda_c, \lambda_1, \lambda_2, \lambda_3$ and (x_3, y_3), we can obtain $(x_5, y_5) = 3P + Q$, Omit $(x_4, y_4) = P + Q$, calculated to further reduce, improved algorithms are as follows [10]:

Input : $P = (x_1, y_1) \neq o$ *and* $Q = (x_2, y_2) \neq o$

Output: $3P + Q = (x_5, y_5)$

Setp1: if (y_1=0) then return P+Q

if (x_1=x_2) then

if (y_1=y_2) then return 4P

else return 2P

step2: compute $\lambda_c, \lambda_1, \lambda_2, \lambda_3$

$A_1 \leftarrow 2y_1, B_1 \leftarrow 3x_1^2 + a$

$A_2 \leftarrow x_2 - x_1, B_2 \leftarrow y_2 - y_1$

$\sigma = (A_1B_2 + A_2B_1) + (A_2B_1 - A_1B_2) - A_2(A_1A_2)^2$

$\lambda_3 = (\sigma A_1A_2)^{-1}$

$\lambda_1 = \sigma \lambda_c A_2B_1$

$\lambda_2 = \sigma \lambda_c A_1B_2$

$\lambda_3 = \lambda_c (A_1B_2)^3 (x_1 - x_3)(\lambda_1 - \lambda_2) - \lambda_2$

Step3: compute (x_3, y_3)

$x_3 = \lambda_1^2 - 2x_1$, $y_3 = \lambda_1(x_1 - x_3) - y_1$

Step4: compute (x_5, y_5)

$x_5 = (\lambda_3 + \lambda_2)(\lambda_3 - \lambda_2) - x_3 + x_2 + x_1$,
$y_5 = \lambda_3(x_3 - x_5) - y_3$

Step5: result (x_5, y_5)

Known $x \in F_p$, $\gcd(x, p) = 1$, satisfies the equation $xA = 1 \mod p$'s $A \in F_p$ which is known as the inverse of the finite field, can be expressed as $x^{-1} \mod p$, the algorithm is as follows:

step1:

b:=P;u_0=0;u_1=1

while (b<>0)

step2:

s:=x div b;r:=x mod b;u: =u_0-s;u_1*

x:=b; u_0= u_1; u_1=u

step3:

if $u_0 \geqslant 0$

then return u_0

else return u_0+p

The algorithm description of the actual realized should put modulo division after each multiplication Operation is carried out. This may save more time, otherwise, the results of each Exponentiation will become more and more huge. We can use the following method to speed up the computation speed.

Input: m , p, x

Output: $x^m \mod p$

(1) $y=x$

(2) for $I=n$; $I>=0$; I—

$y=y^2 \cdot x^m \mod p$

(3) result y

5.3. Algorithm Efficiency Analysis

In the embedded system platform to achieve the elliptic curve calculation, we selected the ARM9 family of processors in the Samsung's S3C2410X, Xscale PXA 270, S3C6410 and S5PV310, in the Linux operating system, C programming and embedded assembly mix programming situation. We compared the computing time of Optimizing by this method and not by look-up table, the conclusion is shown in the Table **1**.

Table 1. Efficiency table

Algorithm	S3C2410	PXA 270	S3C6410	S5PV310
Look-up table optimization	687ms	272ms	102ms	72 ms
Not look-up table	956ms	482ms	243ms	161 ms

6. CONCLUSIONS

Implementation of elliptic curve cryptosystem has great difficulty. This paper has carried on a more thorough com-

parison to the elliptic curve generating algorithm, proposing the use of pre-computation. We use established in tabular form to simplify the system operation, and to improve the Modular Exponentiation Algorithm and reduce the amount of modular multiplication calculation. We have proposed for embedded systems the elliptic curve algorithm. We have carried on the movement confirmation on many kinds of embedded processors. When designing embedded system, if we choose with the DSP or FPGA co-processing system and Optimize procedure realization, the operations become more efficient.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

ACKNOWLEDGMENTS

Declared none.

REFERENCES

[1] Menezes, T. Okamoto, and S. Vanstone, "Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field," *IEEE Trans. Inform Theory*, vol. 39, no. 2, 1639-1646, 1993.

[2] N. Smart, "The Discrete Logarithm Problem on Elliptic Curves of Trace One," *J. Cryptol.*, vol. 12, no. 3, 193-196, 1999.

[3] I. Garcia-Selfa, A.O. Miguel, and J. M. Tomero, "Computing the rational torsion of an elliptic curve using tate normal form," *J. Number Theory*, vol. 96, no. l, pp. 76-88, 2002.

[4] N. Koblitz, "CM-Curve with Good Cryptographic Properties, Advances in Cryptology Crypto'91" LNCS576, 1991, Springer-Verlag, pp. 279-287.

[5] G. Lay, and H. Zimmer, "Constructing Elliptic Curves With Given Group Order Over Large Finite Field," Algorithmic Number Theory: First International Symposium, Lecture Notes in Computer Science (877), Springer-Verlag, pp. 250-263, 1994.

[6] X. Youan and L. Layuan, "A New Quick Public Key Crypto System Based on the Difficulty of Factoring Very Large Numbers," International Symposium on Distributed Computing and Applications to Business, Engineering and Science DCABES'2001, pp. 230-234, 2001.

[7] S. A. Vanstone and R. J. Zuccherato, "Elliptic curve cryptosystems using curves of smooth order the ringZn," *IEEE Trans. Inform Theory*, vol. 43, no. 7, pp. 1231-1237, 1997.

[8] X. Youan, "Researches Elliptic Curve Public Key Cryptosystems in Network Information Security", PhD thesis, WuHan University of Technology, 2003.

[9] M. Ciet, M. Joye, K. Lauter and L. M. Peter, "Trading inversions for multiplications in elliptic curve cryptography," *Des, Codes Cryptography*, vol. 39, no. 2, pp. 189-206, 2006.

[10] L. Lian-hao and S. Yong, "Fast algorithm for scalar multiplication in elliptic curves cryptography," *Appl Res Comput,* vol. 26, no. 3, pp. 1104-1108, 2009.

Received: July 02, 2013 Revised: July 07, 2013 Accepted: July 25, 2013