社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.
COMP93-25, ISEC93-9 (1993-07)

素体上の高速な楕円曲線暗号の構成

宮地 充子

松下電器産業株式会社 通信システム研究所

〒571 大阪府門真市門真 1006

あらまし　本論文では，素体上の高速な楕円曲線暗号の構成を提案し，さらに楕円曲線暗号の新しいメリットについて述べる．

和文キーワード　楕円曲線，暗号

Fast Elliptic Curve Cryptosystems

Atsuko Miyaji

Matsushita Electric Industrial Co., LTD.

1006, Kadoma, Kadoma-shi, Osaka, 571 Japan

Abstract　The security of elliptic curve cryptosystems does not depend on the definition field but on the group structure of an elliptic curve. So we can construct elliptic curve cryptosystems over a finite field in which we can compute modular multiplication fast. It is a great advantage over finite field discrete logarithm cryptosystems. In this paper, we study the feasibility of constructing an elliptic curve cryptosystem defined over such F_p. Another advantage of elliptic curve cryptosystems is also investigated.

英文 key words　elliptic curve, cryptology

1 Introduction

Koblitz ([6]) and Miller ([12]) proposed a method by which a public key cryptosystem can be constructed on the group of points on an elliptic curve over a finite field instead of a finite field. If elliptic curve cryptosystems avoid the Menezes-Okamoto-Vanstone reduction ([16]), then the only known attacks are the Pollard ρ−method ([18]) and the Pohlig-Hellman method ([17]). Then the condition for secure cryptosystems is only to construct E/F_q with $\#E(F_q)$ divisible by a large prime. In other words, the security does not depend on the group structure of the definition field F_q but only on the group structure of $E(F_q)$. Since the running time of elliptic curve cryptosystems depends deeply on the definition field, it is a benefit for us to be able to select a definition field that enables fast modular multiplication.

On the other hand, for a finite field discrete logarithm problem (DLP), there are some attacks ([2, 9]) which depend on the definition field. So if we construct cryptosystems based on DLP on such a field, we are compelled to enlarge the definition field.

Therefore elliptic curve cryptosystems have a great advantage over cryptosystems based on DLP, which are called finite field cryptosystems in this paper. Using the merit that a kind of definition field of E brings fast modular multiplication without causing a critical attack, some works on implementation of elliptic curve cryptosystems over F_{2^r} have been done ([5, 8]). On the other hand, for elliptic curve cryptosystems over F_p, there have been no such works. Only works on secure construction or devices have been done ([15, 13, 14]). The purpose of this paper is to investigate an elliptic curve over F_p from the point that we can select F_p which gives fast modular multiplication.

This paper is organized as follows. Section 2 summarizes elliptic curve cryptosystems. Section 3 investigates the advantage of elliptic curve cryptosystems, that is we can construct elliptic curves E over F_p in which we can compute modular multiplication fast, while maintaining the security. In Section 4, we show an algorithm to construct E over such F_p and discuss the expected running time of the algorithm. The examples constructed by the algorithm are shown in Section 5. Section 6 discusses another advantage of elliptic curve cryptosystems.

2 Elliptic curve cryptosystems

We will summarize cryptosystems using an elliptic curve over F_p, where $p \geq 5$. An elliptic curve over F_p is given as follows,

$$E : y^2 = x^3 + Ax + B \ \ (A, B \in F_p, 4A^3 + 27B^2 \neq 0).$$

Then the set of F_p-rational points on E (with a special element $\mathcal{O}$ at infinity), denoted $E(F_p)$, is a finite abelian group, where $E(F_p) = \{(x, y) \in F_p^2 | y^2 = x^3 + Ax + B\} \cup \{\mathcal{O}\}$.

The security of cryptosystems on E/F_p chosen appropriately depends on the size of a large prime l with $l \mid \#E(F_p)$. Therefore only the condition that l is 30 digits or more is required. Here we discuss briefly how to chose E/F_p appropriately. An elliptic curve E/F_p is supersingular if and only if $a \equiv 0 \pmod{p}$, where $a = p + 1 - \#E(F_p)$. On the other hand, from Hasse's theorem ([22]), the number of rational points $\#E(F_p)$ satisfies that

$$-2\sqrt{p} \leq a \leq 2\sqrt{p}, \ \ a = p + 1 - \#E(F_p).$$

Therefore we can say that, for a large prime p,

$$E/F_p \text{ is supersingular} \iff a = 0.$$

Cryptosystems on supersingular E/F_p is attacked by the method of [16]. Thus we choose E/F_p with $a \neq 0$. Then select E/F_p satisfying the condition that a large prime l with $l \mid \#E(F_p)$ does not divide $p^t - 1$ for all small t. After this, we use only such elliptic curves chosen appropriately.

The running time of cryptosystems on E/F_p depends on the computation of kP for $P \in E(F_p)$. It is accomplished by repeating doubling and adding. For the formulae, see [22]. The formulae say that one operation (i.e. addition and doubling) on elliptic curves requires only the arithmetic in definition field but so many (about more than 10 multiplications). Therefore, if we can select a definition field in which we can calculate modular multiplication fast, the running time for computing kP will be so reduced.

3 Definition Field Analysis

If we construct elliptic curve or finite field cryptosystems over F_{2^r}, we can select such definition field that there exists a basis which enables fast multiplication over F_{2^r}, for example optimal normal basis. To construct elliptic curve or finite

field cryptosystems over F_p, we had better select the definition field F_p with $p = 2^e - s$ (s is a small t-bit integer)([20]): Multiplications over F_p can be done by replacing $2^e \equiv s \pmod p$ without computing a residue modulo p. Thus we can compute multiplications over F_p by repeating the following (1).
Let $a, b \in F_p$.

$$\begin{aligned} a * b &= \sum_{i=0}^{2e-1} x_i 2^i \\ &\equiv \sum_{i=0}^{e-1} (x_i + s x_{i+e}) 2^i \pmod p = \sum_{i=0}^{e+t} y_i 2^i \\ &\vdots \end{aligned} \tag{1}$$

This means that the smaller s is, the faster modular multiplication is. Especially when s is enough small, modular multiplication (over F_p) can be accomplished by computation amount of only one multiplication of two e-bit integers.

If we construct a finite field cryptosystem over F_p with $p = 2^e - s$, there exists an attack of the number field sieve ([9]). The attack especially will be applied to primes $p = r^e - s$ for a small positive integer r and a nonzero integer s of small absolute value. Since the above $p = 2^e - s$ is the case of $r = 2$, we are forced to enlarge s or e in order to avoid the attack. To the contrary, the definition field F_p with $p = 2^e - s$ does not bring a critical attack for the elliptic curve cryptosystems. Since the attack is a generalization of the Gaussian integer method ([3]) to a general number field, the discussion of [12] that the index-calculus attacks do not extend to elliptic curve cryptosystems still holds.

We have seen that, for elliptic curve cryptosystems, there exists a definition field F_p that brings fast modular multiplication without causing a critical attack. Therefore if we can construct E/F_p ($p = 2^e - s$, s is small) easily, then it is a great advantage.

4 Construction of Elliptic Curves

An elliptic curve E/F_p, where p is a prime represented by $p = 2^e - s$ for a small s, can offer fast cryptosystems maintaining the security, as we have described above. Here we show an algorithm to construct an elliptic curve E over such F_p. We also discuss the expected running time of the algorithm.

4.1 Construction Method 1

Let p be a prime represented by $p = 2^e - s$ for a small integer s. For a given F_p, a natural algorithm to construct an elliptic curve whose the number of rational points is divisible by a large prime is given as follows.

Algorithm1

1. Choose $A, B \in F_p$ such that $4A^3 + 27B^2 \neq 0$.
2. Let $E : y^2 = x^3 + Ax + B$ and calculate $N = \#E(F_p)$ by Schoof's algorithm ([21]).
3. If N is divisible by a large prime then stop. If not, then goto 1.

The expected running time is given as a product of two factors, the expected time needed to test one element (i.e. A and B) and the expected number of repetition to find a good E/F_p. Obviously, the latter is almost equal to the expected number of repetition of step 3. It is determined by the probability that $\#E(F_p)$ of a randomly chosen E/F_p is divisible by a large prime. As for the probability, we have the next theorem.

Theorem 1 ([11]) *If S is a set of integers s with $|s-(p+1)| \leq \sqrt{p}$ then the probability of $\#E/F_p \in S$, $prob(\#E/F_p \in S)$ is*

$$prob(\#E/F_p \in S) \geq \frac{c(\#S - 2)}{\sqrt{p} \log p}.$$

From the theorem, the probability is roughly equal to the chance that a randomly chosen integer of size approximately p is divisible by a large prime. It occurs rather frequently. Actually, the probability itself is not so important for comparison between Algorithm 1 and Algorithm 2 shown in the next section, since both algorithms require this step.

On the other hand, the expected time needed to test A and B for each step is dominated by the running time of Schoof's algorithm, which is $O((\log p)^8)$. It seems to work rather slow. In the following section, we will show an algorithm which works faster.

4.2 Construction Method 2

We will show an algorithm using the fact that E/F_p can be described as the reduction modulo p of an elliptic curve with complex multiplication by an order of a quadratic field $Q(\sqrt{-D})$, where $D \equiv 0, 3 \pmod 4$ is a positive integer indivisible by the square of any odd prime ([4, 10]). Applying the fact, a primality proving algorithm is proposed ([1]). A problem in applying the fact is that the larger D becomes, the more difficult it is to construct E/F_p correspond to an order of $Q(\sqrt{-D})$. Therefore we set an upper bound of D to B. An algorithm to construct E/F_p whose $\#E(F_p)$ is divisible by a large prime is given as follows.

Algorithm2

1. Choose a prime p represented by $p = 2^e - s$.
2. Choose D with $\left(\frac{-D}{p}\right) = 1$. If such $D \leq B$ does not exist, then goto step 1.
3. If $D \equiv 3 \pmod 4$, then check $4p = a^2 + Db^2$ for an integer a, b. If $D \equiv 0 \pmod 4$, then set $D = 4D'$ and check $4p = a^2 + D'b^2$ for an integer a, b. If such integers a and b do not exist, then goto step 2.
4. Set $N = p + 1 - a$ and $\widetilde{N} = p + 1 + a$. Check either N or $\widetilde{N}$ is divided by a large prime. If it is not divided, then goto step 2.
5. Calculate a class polynomial $P_D(X)$. Take one solution j_0 of $P_D(X) \equiv 0 \pmod p$. Construct an elliptic curves E/F_p with j-invariant j_0 and $\#E(F_p)$ equal to the one divisible by a large prime, N or $\widetilde{N}$. Stop.

In step4, we check either N or $\widetilde{N}$ is divided by a large prime. The size of the large prime depends on a security level, which will be discussed at the end of this section.

The expected running time of Algorithm 2 is to be investigated. Each step 1 and 2 can be done easily. In step 3, we can also easily check by computing the expansion into continued fraction. The problem is the expected number of repetition of step 3, which is also relation to step 5. We will discuss it later. In step 4, which is also required in Algorithm1, we can easily check whether an integer is divisible by a large prime. As for the expected number of repetition of step 4, we have already discussed in Section 4.1.

There exists a barrier in step 5. The degree of $P_D(X)$, denoted $h(-D)$, is known to be $O(D^{1/2+\epsilon})$. For a large D, we can hardly construct $P_D(X)$. Generally, in proportion as p becomes large D becomes large. At first sight, the step 5 seems not to work well. Thus the bound B of D is important. If we can set B to be small, then there is no problem in step 5. But it will cause the decrease of probability to pass step 2 and 3. Next we will show the probability to pass step 2 and 3 is actually enough large for a small B.

As is proved later, the smaller $h(-D) \geq 1$ is, the larger the probability for D to pass step 3 is. We would use all D satisfying the condition that $h(-D)$ is smaller than or equal to at least three. The last D with $h(-D) \leq 3$ is equal to 907. Here we set $B = 1055\,(\geq 907)$. In fact, the probability to pass step 3 is almost equal with any $B \geq 907$.

Now we show the probability to pass step 2 and 3 is enough large, regardless of p. Since the upper bound B of D equals 1055, we get that

$$\#\{D \equiv 0, 3 \pmod 4 | D > 0 \text{ is indivisible by the square of any odd prime.}\} = 322.$$

Table 1 shows first D, last D and the number of D with $D \leq 1055$ and $h(-D) \leq 10$.

For a given p, the probability for D to pass step 2 is $\frac{1}{2}$. On the other hand, D passes step 3 if and only if p splits into two principal ideals in an imaginary quadratic field $Q(\sqrt{-D})$. So the probability for D to pass step 3 is $\frac{1}{h(-D)}$. To join these, we get the probability for D to pass step 2 and 3 is

$$\frac{1}{2h(-D)}.$$

Therefore the probability for at least one D to pass step 2 and step 3 is given as follows.

$$\begin{aligned} 1 - \prod_{D \leq B} \frac{2h(-D)-1}{2h(-D)} &= 1 - \prod_{h(-D)=1} \frac{1}{2} \prod_{h(-D)=2} \frac{3}{2^2} \prod_{h(-D)=3} \frac{5}{2 \cdot 3} \prod_{h(-D)=4} \frac{7}{2 \cdot 4} \prod_{h(-D)\geq 5} \frac{2h(-D)-1}{2h(-D)} \\ &\geq 1 - \left(\frac{1}{2}\right)^9 \left(\frac{3}{4}\right)^{18} \left(\frac{5}{6}\right)^{16} \left(\frac{7}{8}\right)^{46} \\ &\approx 1 \end{aligned} \tag{2}$$

We have got, regardless of p, at least one D goes to step 4 in probability almost equal to 1.

Thus we have got the next result. The running time of step 2, 3 and 5 is a low order term respectively. So the expected time needed to test D for each step is dominated by step 4. If we use a probabilistic primality test for step 4, the expected running time is $O((\log p)^3)$ ([19]). On the other hand, the expected number of repetition of Algorithm 2 to find a good E/F_p is almost equal to that of step 4. So it is almost equal to that of Algorithm 1. To sum up, we have seen that:

1. the expected time needed to test a candidate is $O((\log p)^8)$ for Algorithm 1;
2. the expected time needed to test a candidate is $O((\log p)^3)$ for Algorithm 2;
3. the expected number of repetition of Algorithm 1 is almost equal to that of Algorithm 2.

Therefore we have seen that Algorithm 2 works rather faster than Algorithm 1. We have also seen that, for a prime p represented by $p = 2^e - s$ (s is a small integer), the running time of Algorithm 2 is roughly equal to that of finding an integer, of size approximately p, divisible by a large prime. In fact, we were convinced experimentally of this. So we can easily construct secure elliptic curve cryptosystems over F_p by working Algorithm 2 (see Section 5).

$h(-D)$	First D	Last D	Number of D
1	3	163	9
2	15	427	18
3	23	907	16
4	39	1027	46
5	47	1051	18
6	87	1048	23
7	71	859	12
8	95	1043	41
9	199	823	6
10	119	923	20

表 1: First D, Last D with $h(-D) \leq 10 \text{and} D \leq 1055$

We will briefly discuss the size of "a large prime" in step 4 of Algorithm 2. If "a large prime" is more than 120-bit, then the known attacks on such an elliptic curve cryptosystems require at least 2^{60} elliptic curve operations. The amount of necessary operations is roughly equal to that of attacks on finite field cryptosystems on F_p (p is 512 bits). Sometimes lower security is required when fast implementation is required or memory storage is limited. In such a case, "a large prime" is replaced by a smaller prime like 97 bits. We will show examples for each case. Here we call the former case Higher Security Case and the latter case Lower Security Case.

We calculate roughly the expected number of repetition for step 4 of Algorithm 2 in the above two cases. We have seen that it is determined by the probability of the chance that a randomly chosen integer of size approximately p is divisible by a large prime more than L. This probability is roughly equal to $1 - u^{-u}$ where $u = \log p / \log L$ ([7]). In fact we were convinced experimentally that the probability is almost equal to the probability to pass step 4 for the next each case.

• **Higher security Case**

We set $p = 2^{127} - s$ ($s = 1, 25, 39, \cdots$) and $\log_2 L = 120$. As we know well, the prime of $s = 1$ is the 12th Mersenne prime. Then the probability for N ($\widetilde{N}$) to be divisible by a large prime more than L is

$$1 - (\log p / \log L)^{-\log p / \log L} = 1 - (127/120)^{-127/120} \approx 0.06$$

Since we have two elliptic curves for each D, the expected number of repetition for step 4 of Algorithm 2 is about 8.

• **Lower security Case**

We set $p = 2^{107} - s$ ($s = 1, 171, 321, \cdots$) and $\log_2 L = 97$. As we know well, the prime of $s = 1$ is the 11th Mersenne prime. Then the probability for N ($\widetilde{N}$) to be divisible by a large prime more than L is

$$1 - (\log p / \log L)^{-\log p / \log L} = 1 - (107/97)^{-107/97} \approx 0.10$$

Since we have two elliptic curves for each D, the expected number of repetition for step 4 of Algorithm 2 is about 5.

5 Examples

In this section, we show examples constructed by Algorithm 2 described in Section 4. First we show an example in the case that higher security is required.

• **Higher security Case**
Here we set $p = 2^{127} - 1$.

step 2 For $D = 24$, we get $\left(\frac{-24}{p}\right) = 1$.

step 3 Computing the expansion into continued fraction, we find that

$$p = a^2 + 6b^2,$$

with

$$a = 10671\ 93179\ 31455\ 45219,\quad b = 3061\ 89089\ 89631\ 04781$$

step 4 Set $N = p + 1 - a$ and $\widetilde{N} = p + 1 + a$. Then

$$N = 2 * 2 * 3 * 141\ 78431\ 95503\ 91026\ 40749\ 96471\ 19418\ 27071,$$

where the last prime is a 124-bit prime.

step 5 Calculate a class polynomial $P_{24}(X)$. Then we get

$$P_{24}(X) = X^2 - 4834944X + 14670139392.$$

Then $j = 31493462074257663932556096$ is one solution of $P_{24}(X) \equiv 0 \pmod{p}$. Construct an elliptic curve E/F_p with j-invariant j and $\#E(F_p) = N$. We get

$$E : y^2 = x^3 + Ax + B,$$

where

$$A = 915\ 03150\ 65123\ 53429\ 89289\ 36723\ 21130\ 54488$$

$$B = 1177\ 15828\ 25431\ 33059\ 03422\ 01272\ 67034\ 04901.$$

In the above example, $\#E(F_p)$ is divisible by a 124-bit prime. So $E/F_{2^{127}-1}$ can offer a fast cryptosystem keeping a desirable security.

Next we show an example in the case that lower security is allowed.

• **Lower security Case**
Here we set $p = 2^{107} - 1$.

step 2 For $D = 3$, we get $\left(\frac{-3}{p}\right) = 1$.

step 3 Computing the expansion into continued fraction, we find that

$$4p = a^2 + 3b^2,$$

with

$$a = 24\ 38789\ 23037\ 40815\ ;\ b = 4\ 25314\ 84925\ 08931.$$

step 4 Set $N = p + 1 - a$ and $\widetilde{N} = p + 1 + a$. Then

$$N = 2 * 89 * 91156\ 89709\ 50636\ 59896\ 51314\ 76441,$$

where the last prime is a 100-bit prime.

step 5 Calculate a class polynomial $P_3(X)$. Then we get $P_3(X) = X$. So $j = 0$ is one solution of $P_3(X) \equiv 0 \pmod{p}$. Construct an elliptic curve E/F_p with j-invariant 0 and $\#E(F_p) = N$. We get

$$E : y^2 = x^3 + 625.$$

In the above example, $\#E(F_p)$ is divisible by a 100-bit prime. So $E/F_{2^{107}-1}$ can offer a fast cryptosystem keeping a desirable security.

6 Other Aspects of Security

Here we investigate another advantage of elliptic curve cryptosystems. From a security point of view, we would prevent all cryptosystems from breaking when a cryptosystem happen to be broken. We would also reduce the probability that a system happens to be broken. This is why it is desirable that the system parameter of a cryptosystem is changed, in each system or periodically in the same system, to another non-isomorphic cryptosystem with the same security. On the other hand, the running time of a cryptosystem depends on that of the fundamental operation. Therefore it is also desirable that a different cryptosystem can be offered without changing the fundamental operation.

The system parameter of a cryptosystem on an elliptic curve E/F_p is the coefficient of E, p of the definition field F_p, a basepoint in $E(F_p)$ and $\#E(F_p)$ (or the order of the basepoint). As we have seen in Section 2, a parameter p of the definition field forms a part of the fundamental operation. In the case of signature or identification, $\#E(F_p)$ (or the order of basepoint) forms other part of fundamental operation. Applying the above desirable condition to elliptic curve cryptosystems, we had better construct non-isomorphic elliptic curves each other without changing these parameters, p and $\#E(F_p)$. Namely it is desirable that we can construct non-isomorphic elliptic curves over F_p each other with the same number of rational points.

By Hasse's theorem, we have $|a| \leq 2\sqrt{p}$ for $a = p+1-\#E(F_p)$. Conversely, for any integer $|a| \leq 2\sqrt{p}$, there exists E/F_p with $\#E(F_p) = p + 1 - a$ ([4]). On the other hand, there are at least $2p$ elliptic curves over F_p modulo F_p-isomorphism. Therefore there are some elliptic curves over F_p with the same $\#E(F_p)$ points modulo F_p-isomorphism. Two elliptic curves E and E_1 are called isogenous if $\#E(F_p) = \#E_1(F_p)$. From the above discussion, it is desirable to construct elliptic curves that are non-isomorphic and isogenous each other.

By Algorithm 1, we can hardly construct an elliptic curve E_1 isogenous to E since the probability to find such E_1 is too small, $O((\sqrt{p})^{-1})$. From the above point of view, Algorithm 1 is not so suitable. On the other hand, by Algorithm 2, we can construct such elliptic curves as follows.

For any $|a| \leq 2\sqrt{p}$, j-invariants of E/F_p with $p + 1 \pm a$ elements are represented as a solution of

$$\prod_{b'|b} P_{Db'}(X) \equiv 0 \pmod{p}, \ 4p = a^2 + Db^2. \tag{3}$$

So computing another solution j_1 (not equal to $j(E)$) of (3), we get an elliptic curve which is not isomorphic to E but has the same number of rational points.

We show one example. In the example of Higher Security Case (Section 5),

$$\begin{aligned} P_{24}(X) &= X^2 - 4834944X + 14670139392 \\ &\equiv (X - j)(X - j_1) \pmod{p}, \end{aligned}$$

where

$$\begin{aligned} j &= 3\ 14934\ 62074\ 25766\ 39325\ 56096, \\ j_1 &= 1701\ 41183\ 46043\ 77382\ 69613\ 04605\ 19563\ 84575. \end{aligned}$$

Then we construct an elliptic curve E_1/F_p with j-invariant j_1 and $\#E_1(F_p) = N$, where N is divisible by a 124-bit prime. We get

$$E_1 : y^2 = x^3 + A_1 x + B_1,$$

where

$$A_1 = 1400\ 68970\ 50479\ 08325\ 24538\ 34292\ 93927\ 22673,$$

$$B_1 = 366\ 65585\ 84970\ 41444\ 39129\ 79404\ 76337\ 79873.$$

As described above, two elliptic curves E, $E_1/F_{2^{127}-1}$ are not isomorphic each other but have the same N rational points. So we can construct two different cryptosystems, implemented by the same fundamental operations.

7 Conclusions

We have investigated an elliptic curve over F_p which gives fast and secure cryptosystems from the view point that we can select F_p with fast modular multiplication. We have shown an algorithm to construct such E/F_p where $p = 2^e - s$ (s is a small integer) and that the algorithm can work well. Especially we have shown examples of E/F_p constructed by the algorithm for two Mersenne primes $p = 2^{107} - 1$ and $p = 2^{127} - 1$. Furthermore we have discussed new advantage of elliptic curve cryptosystems, which are non-isomorphic each other and implemented by the same fundamental operations.