A Method for Generating Prime Order Elliptic Curves over $F_{q^{2^c}}$

Yasuyuki Nogami† Yoshitaka Morikawa†

Department of Communication Network Engineering
Okayama University
Okayama 700-8530 Japan

(Received November 24, 2004)

This paper proposes an algorithm for generating prime order elliptic curves over extension field whose extension degree is a power of 2. The proposed algorithm is based on the fact that the order of the twisted elliptic curve is able to be a prime number when the extension degree for the twist operation is a power of 2. When the definition field is $F_{(2^{40}-87)^4}$, the proposed algorithm can generate a prime order elliptic curve within 5 seconds on PentiumIII (800MHz) with C language.

1 Introduction

In the modern information-oriented society, various devices are connected via the Internet. Information security technology has played a key role in protecting these devices or important information from unauthorized invasion and evil Internet users. Especially, the public-key cryptosystem has many uses such as to sign digitally and to realize electronic commerce. The RSA cryptosystem, a public-key cryptosystem, has been the most widely used, but its key for ensuring security is approximately 2000 bits in length. Therefore, it is not efficient to implement the RSA cryptosystem on devices with scarce computation resources such as an IC card. On the other hand, since the elliptic curve cryptosystem(ECC) attains the same security level with an approximately 7-fold smaller key length as compared to the RSA, the ECC has received much attention and has been implemented on various processors.

This paper mainly deals with elliptic curves whose defining equation is written as follows ;

$$E(x, y) = y^2 - x^3 - ax - b = 0. \quad (1)$$

In general, the coefficients a, b are elements in a certain finite field, which is called the coefficient field in this paper, and the solutions (x, y) to Eq.(1) are called rational points. The rational points forms an additive Abelian group and the security attained by ECC relies upon the difficulty of a discrete logarithm problem on this additive Abelian group. This problem is called the elliptic curve discrete logarithm problem(ECDLP)[1]. Since this additive Abelian group plays a role of the key space, the order of the group that is the number of rational points must be a large prime number or have a large prime factor for security reasons of the ECC. In practice, the large prime number should be at least 160 bits[2]. Correspondingly, from Hasse's theorem[1], the order of the definition field, in which the coordinates of the rational points lie, has to be at least 160 bits[1].

The ECC has to avoid Anomalous attack[1], FR attack[3], and Weil Descent attack[4]. In other words, if these attacks can not reduce the ECDLP on the elliptic curve and of course the order has an enough large prime factor, it is said that the ECC is secure. From the viewpoint of implementation and security, it is said that a prime order elliptic curve is the best[5]. In order to check whether or not the ECC avoids these attacks, it is quite important to count the order of the elliptic curve. As the conventional order counting algorithm, Schoof's algorithm[6], Schoof Elkies Atkin (SEA) algorithm[1], and Satoh's algorithm[7] are well-known. In this pa-

†E-mail: {nogami,morikawa}@cne.okayama-u.ac.jp

per, we propose an algorithm for generating prime order elliptic curves in which we use these order counting algorithms.

Recently, several methods for generating prime order elliptic curves have been proposed. We can roughly classify them into two types, the one adapts some order counting algorithm[5],[8] and the other adapts CM method[9],[10]. The algorithm proposed in this paper belongs to the former. Horiuchi et al. algorithm[5] can generate prime order elliptic curves whose definition field is a prime field. They modified SEA algorithm. When the order of the definition field is about 160 bits, this algorithm, for one prime order elliptic curve generation, requires about 16 times as much computation time as the original SEA algorithm. Konstantinou et al. algorithm[9] is based on CM method[1]. In this algorithm, we cannot freely choose the definition field. In the research area of implementation of ECC, we usually fix the characteristic so as to fast carry out the arithmetic operations in the field[11],[12]. This algorithm is not suitable for the implementation of ECC.

The authors have already reported some properties [13], [14] and experimental results[8]. This paper is the extended version of these reports. In this paper, we propose an algorithm for generating prime order elliptic curves over the extension field whose extension degree is a power of 2. We particularly deal with the case that the characteristic of the definition field is larger than 3. In what follows, F_q and F_{q^m} mean a finite field and its m-th extension field, respectively. Without any additional explanation, p denotes the characteristic of these fields. $E(F_q)$ and $E(F_{q^m})$ denote elliptic curves defined over F_q and F_{q^m}, respectively. In addition, $\#E(F_q)$ and $\#E(F_{q^m})$ denote their orders.

First, we start from an elliptic curve whose coefficient field is a proper subfield of the definition field. According to Weil's theorem[1], when the coefficient field F_q is some proper subfield of the definition field F_{q^m} of an elliptic curve E, the order $\#E(F_{q^m})$ is easily calculated with the order $\#E(F_q)$. In general, the computation time for the order counting algorithm exponentially increases as the order of definition field becomes larger, therefore Weil's theorem achieves substantial savings of the order computation time. In this paper, we call $\#E(F_q)$ the *base order* of elliptic curve, and our objective is not for the base order.

Next, we introduce the fact that the base order $\#E(F_q)$ divides $\#E(F_{q^m})$. This fact indicates not only that $\#E(F_{q^m})$ is not a prime, but also that the largest prime factor of $\#E(F_{q^m})$ is considerably smaller than 160 bits even if the size of the definition field F_{q^m} is about 160 bits. In order to overcome such an undesirable property, we introduce *twist* technique. We can easily operate the twist by exchanging the coefficients of the defining equation of the curve, and the order of the twisted elliptic curve is easily calculated by using that of the original elliptic curve. Therefore, if we know the order of the original elliptic curve, we can obtain that of the twisted elliptic curve without any complicated calculations. In this paper, we denote the operation *twist* with $'$. For example, $\#E'(F_{q^m})$ denotes the order of the twisted elliptic curve $E'(F_{q^m})$.

After that, we show that $\#E'(F_q)$ divides $\#E'(F_{q^m})$ when m has an odd number factor. In this case, the previously mentioned undesirable property is not overcome. On the other hand, the authors found a fact that it is possible for $\#E'(F_{q^m})$ to be a prime number or have a large prime factor when the extension degree m is a power of 2. In this paper, based on this fact and using Weil's theorem, we especially propose an algorithm for generating prime order elliptic curves and then show some experimental results of the proposed algorithm.

In the last of this paper, using the proposed algorithm, we give some concrete examples of elliptic curve suitable for elliptic curve cryptosystem. In addition, since the proposed algorithm does not restrict the characteristic of the definition field to an odd prime number, we give some examples in the case that the characteristic is 2. The authors used Pentium III (800MHz) processor with C language. When the definition field is $F_{(2^{40}-87)^4}$, the proposed algorithm can generate one prime order elliptic curve within 5 seconds. We can say that the proposed algorithm is enough practical. Furthermore, since the proposed algorithm does not restrict the order counting algorithm, we can adopt another fast order counting algorithm, correspondingly the prime order elliptic curve generation will become faster.

2 Fundamentals

In this section, let us go over the fundamentals of elliptic curve, Anomalous attack[1], Frey-Rück (FR) attack[3], Weil Descent attack[4], other related works, and the conventional order counting algorithms.

2.1 Arithmetics on elliptic curve

2.1.1 Coefficient field and definition field

When the characteristic of F_q is not equal to 2 or 3, an elliptic curve over F_q is generally defined by

$$E(x, y) = y^2 - x^3 - ax - b = 0, \quad a, b \in F_q. \quad (2)$$

The solutions (x, y) to Eq.(2) are called F_q-rational points when the coordinates of x and y lie in F_q. This paper deals with elliptic curves whose coordinates lie in some extension field but coefficients a, b belong to its proper subfield, in order to distinguish these fields, we call the former *definition field* and the latter *coefficient field*.

2.1.2 Order and trace of elliptic curve

F_q-rational points on an elliptic curve form an additive Abelian group. In this paper, we denote this group and its order by $E(F_q)$ and $\#E(F_q)$, respectively. According to Hasse's theorem[19], the existing range of the order $\#E(F_q)$ becomes as follows ;

$$q + 1 + 2\sqrt{q} \geq \#E(F_q) \geq q + 1 - 2\sqrt{q}. \quad (3)$$

Therefore, the order of the definition field decides the size of the additive Abelian group. Since the order and trace of elliptic curve are closely related to the security of ECC, it is crucial important for cryptographic applications to count the order $\#E(F_q)$, where $t = q + 1 - \#E(F_q)$ is called the trace of $E(F_q)$.

2.2 Anomalous and FR attacks

Frey-Rück (FR) attack[3] reduces the ECDLP on $E(F_q)$ to the DLP in the multiplicative group of a certain extension field F_{q^k}. It is known that the FR attack can reduce the ECDLP which satisfies either of the following conditions, where t is the trace of $E(F_q)$, p is the characteristic of F_q. It is noted that $E(F_q)$ is called a *supersingular* elliptic curve if p divides its trace t.

- $E(F_q)$ is supersingular.
- The trace t is equal to 2.

If the order $\#E(F_q)$ is equal to the characteristic p, the elliptic curve is called *anomalous* curve and not secure for use of ECC[1]. We can easily check these weak elliptic curves from their orders.

2.3 Weil Descent attack

Weil Descent attack[4] reduces the ECDLP to the DLP in the Jacobian of a hyperelliptic curve[4]. The conditions for Anomalous and FR attacks are given on the order of elliptic curve, however, the condition for Weil Descent attack is given on the extension degree of the definition field[15]. In Chap.4, we deal with odd prime numbers and 2 as the characteristic. For example, in order to avoid Weil Descent attack, when the characteristic is 2, the following extension degrees are recommended[4],[16];

- prime numbers larger than 160
- 178, 226, 1018, 1186

When the characteristic is odd, the extension degrees must satisfy the following conditions[15],[17],[18] ;

- Every odd prime factor of the extension degree is larger than or equal to 11.
- The extension degree is not divisible by 8.

2.4 Other related works

Horiuchi et al.[5] have proposed an algorithm for generating prime order elliptic curves whose definition field is a prime field. They modified SEA algorithm. In this algorithm, the characteristic of the definition field must be larger than 160 bits from security reasons. When the order of the definition field is about 160 bits, for generating one prime order elliptic curve, this algorithm requires about 16 times as much computation time as the original SEA algorithm.

Konstantinou et al.[9] recently proposed a prime order elliptic curve generation algorithm. This algorithm uses Weber polynomials[1] and is based on CM method[1]. In this algorithm, we cannot freely choose the definition field and of course the characteristic. This algorithm is not suitable for the purpose of the implementation of elliptic curve cryptosystem because in the research area of implementation we usually fix the characteristic so as to fast carry out the arithmetic operations in the field[11],[12].

2.5 Order counting algorithm and Weil's theorem

As the conventional algorithm for counting the order of elliptic curve, Schoof's algorithm[6], Schoof Elkies Atkin (SEA) algorithm[1], and Satoh's algorithm[7] are

well known. In addition, when the coefficient and definition fields of an elliptic curve are F_q and its extension field F_{q^m}, respectively, we can obtain $\#E(F_{q^m})$ by the following steps, where the *base order* is the number of F_q-rational points on the elliptic curve.

1. Compute the *base order* $\#E(F_q)$.
2. Calculate the objective order $\#E(F_{q^m})$ by the next Weil's theorem[19] with the base order $\#E(F_q)$.

Theorem 1 *Let the coefficient and definition fields be a finite field F_q and its extension field F_{q^m}, respectively. Let $t_1 = q+1-\#E(F_q)$ be the trace of $E(F_q)$, then we have*

$$\#E(F_{q^m}) = q^m + 1 - t_m, \quad t_m = \alpha^m + \beta^m, \tag{4}$$

where α, β are complex numbers which satisfy $\alpha\beta = q$ and $\alpha + \beta = t_1$, and t_m is the trace of $E(F_{q^m})$.

In this paper, we call the above trace t_1 the *base trace* corresponding to the *base order*. Theorem 1 indicates that, when the coefficient field is a proper subfield of the definition field, we can obtain the order of elliptic curve by using the base trace. The detailed usage is shown in Eqs.(5).

3 Main result

In this section, we show a fact that it is possible that the order of the twisted elliptic curve is a prime number or has a large prime factor when the twist is operated over the extension field whose extension degree is a power of 2. Then, based on this fact and using two techniques of Weil's theorem and twist, we particularly propose an algorithm for generating prime order elliptic curves. After that, we show some experimental results of the proposed algorithm.

3.1 Prime order elliptic curve

When the coefficient and definition fields are a finite field F_q and its extension field F_{q^m}, respectively, the order $\#E(F_{q^m})$ is given by

$$\#E(F_{q^m}) = q^m + 1 - t_m. \tag{5a}$$

Let t_1 be the base race of the elliptic curve $E(F_q)$ that is $t_1 = q + 1 - \#E(F_q)$, then t_m is given as follows ;

$$t_m = \sum_{i=0}^{\lfloor m/2 \rfloor} \frac{m}{m-i}\binom{m-i}{i}(-q)^i {t_1}^{m-2i}, \tag{5b}$$

where $\lfloor m/2 \rfloor$ means the greatest integer less than or equal to $m/2$. In this case, the following relation holds for an arbitrary factor m' of the extension degree m[1];

$$\#E(F_{q^{m'}}) \mid \#E(F_{q^m}), \tag{6}$$

where $X \mid Y$ means that X divides Y. Eq.(6) indicates not only that $\#E(F_{q^m})$ is not a prime, but also that the largest prime factor of $\#E(F_{q^m})$ is considerably smaller than 160 bits even if the size of the definition field F_{q^m} is about 160 bits. Therefore, we must adopt a further larger extension field as the definition field for ensuring sufficient security. But it is not desirable from the viewpoint of compact implementation of the cryptosystem. This defect is due to the constrained setting that the coefficient field F_q is a proper subfield of the definition field F_{q^m}. In order to overcome this problem, we allow the coefficient field not to be a proper subfield of the definition field F_{q^m} and also we adopt a technique called *twist*[5].

For an original defining equation;

$$E(x, y) = y^2 - x^3 - ax - b = 0 \quad a, b \in F_q, \tag{7a}$$

The following $E'(x, y)$ is called the *twist* of $E(x, y)$.

$$E'(x, y) = y^2 - x^3 - aA^2x - bA^3 = 0, \tag{7b}$$

where A is a non-zero element in the definition field F_{q^m}. Corresponding to whether A is a quadratic residue (QR) or a quadratic non-residue (QNR), the order of $E'(x, y)$ over F_{q^m} becomes as follows ;

$$\#E'(F_{q^m}) = \begin{cases} q^m + 1 - t_m & \text{when } A \text{ is a QR} \quad \text{(8a)} \\ q^m + 1 + t_m & \text{when } A \text{ is a QNR} \quad \text{(8b)} \end{cases}.$$

By using the twist technique, we can extend the coefficient field F_q to the extension field F_{q^m} and easily obtain the twisted order $\#E'(F_{q^m})$ with t_m. As shown in Eq.(5a) and Eq.(8a), when A is a quadratic residue in the definition field F_{q^m}, the previously mentioned undesirable property Eq.(6) remains since Eq.(8a) is equal to Eq.(5a). In what follows, we consider that $E'(F_{q^m})$ is twisted with a QNR, accordingly the twisted order $\#E'(F_{q^m})$ is given by Eq.(8b).

Now, let us examine whether or not $\#E'(F_{q^m})$ is able to be a prime number (or have a large prime factor). When the extension degree m has an odd factor $m' \neq 1$, then $\#E'(F_{q^m})$ also has the same undesirable property as follows ;

$$\#E'(F_{q^{m/m'}}) \mid \#E'(F_{q^m}). \tag{9}$$

On the other hand, when the extension degree m is equal to 2^c with a positive integer c, we can calculate the twisted order $\#E'(F_{q^{2^c}})$ by

$$\#E'(F_{q^{2^c}}) = q^{2^c} + 1 + t_{2^c}, \quad (10a)$$

$$t_{2^c} = \sum_{i=0}^{2^{c-1}} \frac{2^c}{2^c - i} \binom{2^c - i}{i} (-q)^i {t_1}^{2^c - 2i}, \quad (10b)$$

and there exist a lot of t_1's such that the twisted order $\#E'(F_{q^{2^c}})$ is a prime (*see* A). The absolute values are tabulated in Table 1. The reason why we use the absolute values is that from Eq.(10b) $\#E'(F_{q^{2^c}})$ in either case of $\pm t_1$ are equal to each other. In the case of $(q, m) = (2^{28} + 3, 8)$, for example, $\#E'(F_{q^8})$ becomes a prime when $t_1 = \pm 59$. Concluding this section, we can generate prime order elliptic curves when the extension degree m for the twist Eqs.(7) is a power of 2.

Table 1: The absolute values of t_1's such that the twisted order $\#E'(F_{q^{2^c}})$ becomes a prime number

$q^\dagger$	2^c	absolute value of t_1
$2^{15} + 3$	16	$23, 39, 63, 103, \cdots$
$2^{24} - 3$	8	$39, 217, 261, 345, \cdots$
$2^{28} + 3$	8	$59, 79, 91, 111, \cdots$

† In this table, each q is a prime number.

3.2 Proposed algorithm

Prime order elliptic curve generation algorithm

Input: Coefficient field F_q and extension degree 2^c.
Output: A prime order elliptic curve $E'(F_{q^{2^c}})$.

Step1: Choose coefficients $a, b \in F_q$ of the defining equation $E(x, y)$ at random. Then, test the irreducibility of $E(x, 0)$. If $E(x, 0)$ is not irreducible, then choose different coefficients again. Otherwise, go to Step2.

Step2: Compute the base order $\#E(F_q)$ of the no two-torsion elliptic curve obtained in Step1. Then, determine $t_1 = q + 1 - \#E(F_q)$.

Step3: Calculate t_{2^c} by Eq.(10b), then test whether or not the twisted order $\#E'(F_{q^{2^c}})$ calculated by Eq.(10a) is a prime number . If it is not prime, then return to Step1. Otherwise, go to Step4.

Step4: Determine the twisted equation Eq.(7b) with a quadratic non-residue $A \in F_{q^{2^c}}$. Then, $E'(F_{q^{2^c}})$ is a prime order elliptic curve.

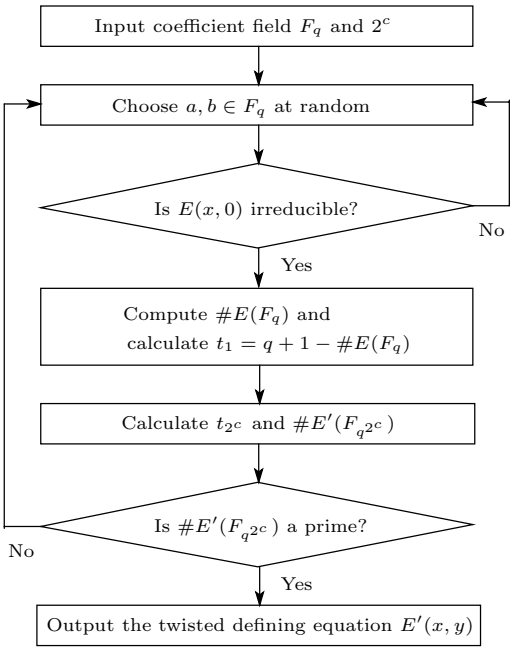

Figure 1: Calculation flow of the proposed algorithm

In Step1 and Step2, it is well-known that the sufficient and necessary condition for an elliptic curve to be a no two-torsion elliptic curve is that $E(x, 0)$ is irreducible over F_q[1]. The terminology "no two-torsion elliptic curve" means that the elliptic curve does not have any rational points of order 2, accordingly the order of no two-torsion elliptic curve is an odd number.

The computational complexity of the irreducibility test in Step1 is bounded at $\mathcal{O}(\log q)$ polynomial modulo operations over F_q by Hiramoto et al. algorithm[20]. On the other hand, that of the order computation in Step2 is bounded at $\mathcal{O}(\log^6 q)$ by SEA algorithm[1], or at $\mathcal{O}(\log^8 q)$ by Schoof's algorithm[6], respectively. In this paper, we do not deal with the primality test at Step3 into detail, however, its computational complexity is bounded at $\mathcal{O}(2^{3c} \log^3 q)$ bit-operations by Solovay-Strassen algorithm[21], for example. The major step in this procedure will be Step2.

3.3 Experimental results

In this section, we evaluate the performance of the proposed algorithm by using the computation time for generating one prime order elliptic curve. We adopted Hiramoto et al. algorithm[20] for the irreducibility test at

Step1, the Schoof's algorithm[6] for Step2, and a primality test library[22] for Step3, respectively. We implemented them on a PentiumIII (800MHz) processor with C language. In this section, we deal with a prime field F_p as the coefficient field and use prime numbers $2^{24}-3, 2^{28}+3$, and $2^{29}-3$ as the characteristic p.

Now, let us estimate the probability of generating a no two-torsion elliptic curve at random that is the probability of success in one iteration of Step1. The number of possible pairs of the coefficient a, b is equal to p^2 and the number of F_p-irreducible polynomials in the form of x^3+ax+b is $(p^2-1)/3$[23]. Therefore, we can estimate the probability at about $1/3$. Table 2 shows the average times for irreducibility test, Step1: no two-torsion elliptic curve generation, and Step2: order computation by the Schoof's algorithm, respectively.

Table 2: The average times for irreducibility test, Step1, and Step2

p	Irreducibility test $[\mu s]$	Step1 $[\mu s]$	Step2 $[ms]$
$2^{24}-3$	18.03	53.61	13.82
$2^{28}+3$	18.60	56.59	22.03
$2^{29}-3$	22.25	66.59	22.43

From Table 2, we can find that Step1 requires approximately three times as much computation time as irreducibility test and that Step1 is carried out much faster than Step2. In what follows, we do not take the computation time for Step1 into account.

Next, let us experimentally estimate the probability that $E'(F_{p^{2^c}})$ becomes a prime number at Step3. For the extension degrees $2^c = 2, 4, 8, 16$, Table 3 shows the number of t_1's such that the twisted order $\#E'(F_{p^{2^c}})$ is a prime number. It is noted that the base trace t_1 satisfies that $t_1 = p+1-\#E(F_p)$.

Table 3: The number of t_1's such that the twisted order $\#E'(F_{p^{2^c}})$ becomes a prime number

p	Extension degree $m=2^c$			
	2	4	8	16
$2^{24}-3$	418	478	328	178
$2^{28}+3$	1406	2290	1280	758
$2^{29}-3$	2662	752	1290	1028

The following facts have been already known ;

- It has been experimentally shown that the base orders are almost uniformly distributed in the range given by Eq.(3)[5].
- In the range given by Eq.(3), there are $\lfloor 2\sqrt{p} \rfloor$ distinct odd numbers.
- The base trace t_1 is odd if and only if the elliptic curve $E(F_p)$ is a no two-torsion elliptic curve.

Based on these facts, we can estimate the probability by dividing the numbers tabulated in Table 3 by $\lfloor 2\sqrt{p} \rfloor$. The results are shown in Table 4. For example, in the case of $(p, m) = (2^{24}-3, 2)$, since the probability is 0.051, we can generate a prime order elliptic curve with about 20 iterations from Step1 to Step3. In addition, from the table, we can find a tendency that the probability decreases as the extension degree increases. It seems that the distribution of prime numbers becomes sparse as the number increases in accordance with a heuristic reasoning, using the prime number theorem.

Table 4: Probability of $E'(F_{p^{2^c}})$ being prime in Step3

p	$\lfloor 2\sqrt{p} \rfloor$	Extension degree $m=2^c$			
		2	4	8	16
$2^{24}-3$	8191	0.051	0.058	0.040	0.021
$2^{28}+3$	32768	0.042	0.069	0.039	0.023
$2^{29}-3$	46340	0.057	0.016	0.027	0.022

Last, Table 5 exhibits the average time needed for generating one prime order elliptic curve $E'(F_{p^{2^c}})$.

Table 5: The average time for generating a prime order elliptic curve $[ms]$

p	Extension degree $m=2^c$			
	2	4	8	16
$2^{24}-3$	326.5	203.1	380.9	665.4
$2^{28}+3$	472.7	325.7	537.1	977.5
$2^{29}-3$	449.2	854.8	971.7	1256.5

For example, in the case of $(2^{28}+3, 8)$, the average time becomes $537.1ms$. In the case of $(2^{24}-3, 8)$, the average time becomes 380.9ms which is faster than that of the first case. This is because an order computation in the second case is faster than the first case's

as shown in Table 2. On the other hand, in the case of $(2^{28}+3, 16)$, the average time becomes 977.5 ms which is slower than that of the first case. This is because, as shown in Table 4, the probability that the twisted order $E'(F_{p^{2^c}})$ is a prime number in the third case is lower than that of the first case.

4 For elliptic curve cryptosystems

In this section, using the proposed algorithm, we concretely generate elliptic curves suitable for ECC. We do not deal with the case that the characteristic is 3. The authors have used Pentium III (800MHz) processor with C language for programming.

4.1 when the characteristic is larger than 3

First, in order to avoid FR attack, we modify Step3 in the proposed algorithm as follows ;

Step3': Calculate t_{2^c} by Eq.(10b), then test whether or not the characteristic p divides t_{2^c}. If p divides t_{2^c}, then return to Step1. If not, then test whether or not $\#E'(F_{q^{2^c}})$ is a prime number. If it is not a prime number, then return to Step1. Otherwise, go to Step4.

Next, in order to avoid Weil Descent attack, in this section we consider F_p as the coefficient field and restrict the definition field to extension fields F_{p^2} and F_{p^4}, where p is the characteristic. Correspondingly, we choose about 40 bits and 80 bits prime numbers as the characteristic p for ensuring sufficient security.

Table 6 and Table 7 show examples of the twisted elliptic curve whose order is a prime number. For the base order computation at Step2 in the proposed algorithm, we used Schoof's algorithm and SEA algorithm in Example(1) $\sim$ (3) and Example(4) $\sim$ (6), respectively. In the case of Example(3), for example, the order of the twisted elliptic curve, which is generated by the proposed algorithm, becomes a 188 bits prime number, where the detail is shown in Table 9. From Table 6 and Table 7, we find that when the definition field is F_{p^4} the proposed algorithm generates such an elliptic curve within one minute on PentiumIII (800MHz). On the other hand, when the definition field is F_{p^2} it takes several minutes.

4.2 when the characteristic is 2

Let us consider the case that the characteristic is 2, that is $q = 2^d$, where d is a number. In this case, the defining equation $E(x, y)$ is generally written as follows;

$$E(x,y) = y^2 + y - x^3 - ax^2 - b = 0, \quad a, b \in F_q, \tag{11a}$$

$$E(x,y) = y^2 + xy - x^3 - ax - b = 0, \quad a, b \in F_q. \tag{11b}$$

It is well-known that the base order $\#E(F_q)$ of the elliptic curve defined by Eq.(11a) is odd. On the other hand, the base order $\#E(F_q)$ of the elliptic curve defined by Eq.(11b) is even and correspondingly the base trace t_1 is odd. In what follows, we particularly consider the latter defining equation Eq.(11b). When the characteristic is 2, the defining equation of the twisted elliptic curve is given as follows[19] ;

$$E'(x,y) = y^2 + xy - x^3 - (a+a')x^2 - b, \quad a' \in F_{q^{2^c}}, \tag{12}$$

and the twisted order $\#E'(F_{q^{2^c}})$ is given by

$$\#E'(F_{q^{2^c}}) = \begin{cases} q^{2^c} + 1 - t_{2^c} & \text{when } \mathrm{Tr}(a') = 0 \quad (13a) \\ q^{2^c} + 1 + t_{2^c} & \text{when } \mathrm{Tr}(a') = 1 \quad (13b) \end{cases},$$

where t_{2^c} is given by Eq.(10b) and $\mathrm{Tr}(\cdot)$ means the following trace function over the prime field F_2 ;

$$\mathrm{Tr}(x) = \sum_{i=0}^{2^c d - 1} x^{2^i}. \tag{14}$$

From Eqs.(13) and Eq.(10b), t_{2^c} becomes odd, correspondingly $\#E'(F_{q^{2^c}})$ becomes even. Therefore, FR attack can not reduce the ECDLP on this twisted curve. In other words, when the characteristic is 2, in order to avoid FR attack the order of elliptic curve must be even, that is the reason why we consider the elliptic curve in the form of Eq.(11b).

From the above discussion, when the characteristic is 2, it is the best that the order of elliptic curve is a product of 2 and some large prime number. As shown in Sec.3.1 and A, it is possible that $\#E'(F_{q^{2^c}})$ is a product of 2 and a prime number when the order is given by Eq.(13b). We modify Step3 as follows ;

Step3': Calculate t_{2^c} by Eq.(10b), then test whether or not $\#E'(F_{q^{2^c}})$ is a product of 2 and a prime number. If it is not a prime number, then return to Step1. Otherwise, go to Step4.

Table 6: Examples of prime order elliptic curve over F_{p^4}

	Example(1)	Example(2)	Example(3)
characteristic p	$2^{40}-87$	$2^{44}+21$	$2^{47}+5$
coefficient field, definition field	F_p, F_{p^4}	F_p, F_{p^4}	F_p, F_{p^4}
modular polynomial	x^4-7	x^4-2	x^4-2
Schoof's algorithm over F_p $[ms]^\dagger$	138	263	361
irreducibility test over F_p $[ms]^{\dagger\dagger}$	1.6	1.7	1.7
average time $[s]^{\dagger\dagger\dagger}$	4.53	7.23	12.3
original defining equation $E(x,y)$	y^2-x^3-x-12	y^2-x^3-x-95	y^2-x^3-x-91
twisted defining equation $E'(x,y)^\ddagger$	$y^2-x^3-\omega^2x-12\omega^3$	$y^2-x^3-\omega^2x-95\omega^3$	$y^2-x^3-\omega^2x-91\omega^3$
order $\#E'(F_{p^4})$	159 bits prime number	176 bits prime number	188 bits prime number

† Computation time of Schoof's algorithm[22]. †† Computation time of Hiramoto et al. algorithm[20].

††† Average time for generating one prime order elliptic curve by the proposed algorithm.

‡ ω is a zero of the modular polynomial and a QNR in F_{p^4} (*see* C).

In addition, it should be noted that in this case we do not need irreducibility test at Step1.

Table 8 shows two examples of the twisted elliptic curve whose order is a product of 2 and a large prime number. The one is defined over $F_{2^{178}}$ and the other is defined over $F_{2^{226}}$. In order to avoid Weil Descent attack, we chose these extension degrees as introduced in Sec.2.3. We used Satoh's algorithm[7] for the base order computation at Step2 in the proposed algorithm. In the case that $F_{2^{226}}$ is the definition field, for example, we consider $F_{2^{113}}$ as the coefficient field and compute the base order $\#E(F_{2^{113}})$. The order of the twisted elliptic curve, which is generated by the proposed algorithm, is a product of 2 and 224 bits prime number, where the detail is shown in Table 9. We find that the proposed algorithm generates such an elliptic curve within a few seconds on PentiumIII (800MHz).

The hexadecimal representation $(\cdot)_{16}$ shown in Table 8 shows the vector representation of an element in the coefficient field with the polynomial basis. For example, $(55C)_{16}$ shown in Example(8) means that

$$\begin{aligned}(55C)_{16} &= (010101011100)_2 \\ &= \tau^{10}+\tau^8+\tau^6+\tau^4+\tau^3+\tau^2, \quad (15)\end{aligned}$$

where τ is a zero of $x^{113}+x^9+1$ over F_2.

5 Conclusion

In this paper, we have proposed an algorithm for generating prime order elliptic curves over the extension field whose extension degree is a power of 2.

First, we started from an elliptic curve whose coefficient field is a proper subfield of the definition field. According to Weil's theorem, when the coefficient field F_q is some proper subfield of the definition field F_{q^m} of an elliptic curve E, the order $\#E(F_{q^m})$ is easily calculated with the order $\#E(F_q)$, however, $\#E(F_q)$ divides $\#E(F_{q^m})$. In order to overcome such an undesirable property, we introduced *twist* technique. When m has an odd number factor, since $\#E'(F_q)$ divides $\#E'(F_{q^m})$, the previously mentioned undesirable property is not overcome, where $'$ means the twist operation. In this paper, we showed that it is possible that $\#E'(F_{q^m})$ is a prime number or has a large prime factor when the extension degree m is a power of 2. Based on this fact and using two techniques of Weil's theorem and twist, we proposed an algorithm for generating prime order elliptic curves over the extension field whose extension degree is a power of 2. Then, we showed some experimental results of the proposed algorithm.

In this paper, we gave some concrete examples of elliptic curve suitable for elliptic curve cryptosystem. Since the proposed algorithm does not restrict the characteristic of the definition field to an odd prime number, we also gave some examples in the case that the characteristic is 2. When the definition field is $F_{(2^{40}-87)^4}$, the proposed algorithm generates one prime order elliptic curve within 5 seconds on PentiumIII (800MHz), for example. From these results, we can

Table 7: Examples of prime order elliptic curve over F_{p^2}

	Example(4)	Example(5)	Example(6)
characteristic p	$2^{80}+13$	$2^{84}+45$	$2^{89}+29$
coefficient field, definition field	F_p, F_{p^2}	F_p, F_{p^2}	F_p, F_{p^2}
modular polynomial	x^2-2	x^2-2	x^2-2
SEA algorithm over F_p [s]†	5.34	7.43	6.02
irreducibility test over F_p [ms]††	3.3	3.4	3.6
average time [s]†††	457	434	402
original defining equation $E(x,y)$	$y^2=x^3+x-17$	$y^2=x^3+x-282$	$y^2=x^3+x-385$
twisted defining equation $E'(x,y)$‡	$y^2=x^3+\omega^2x-17\omega^3$	$y^2=x^3+\omega^2x-282\omega^3$	$y^2=x^3+\omega^2x-385\omega^3$
order $\#E'(F_{p^2})$	160 bits prime number	168 bits prime number	178 bits prime number

† Computation time of SEA algorithm[22]. †† Computation time of Hiramoto et al. algorithm[20].

††† Average time for generating one prime order elliptic curve by the proposed algorithm.

‡ ω is a zero of the modular polynomial and a QNR in F_{p^2} (*see* C).

conclude that the proposed algorithm is enough practical.

References

[1] I.Blake, G.Seroussi, and N.Smart: Elliptic Curves in Cryptography, LNS 265, Cambridge University Press, 1999.

[2] D.Hankerson, A.Menezes, and S.Vanstone: Guide to Elliptic Curve Cryptography, Springer, 2004.

[3] G.Frey and H.Rück: Math. Comp. **62**, 1994 , 865-874.

[4] P.Gaudry, F.Hess, and N.Smart: Hewlett Packard Tech. Report HPL-2000-10 , 2000.

[5] K.Horiuchi, Y.Futa, R.Sakai, M.Kaneko, and M.Kasahara: IEICE Trans., **J82-A**, no.8, 1999, 1269-1277.

[6] R.Schoof: Math. Comp. **44** no.170, 1985, 483-494.

[7] T.Satoh: Jour. of the Ramanujan Mathematical Society, vol.15, 2000, 247-270.

[8] Y.Nogami and Y.Morikawa: Proc. of Workshop on Coding and Cryptography (WCC2003), 2003, 347-3.

[9] E.Konstantinou, Y.Stamatiou, and C.Zaroliagis: Indocrypto2003, LNCS 2904, 2003, 309-322.

[10] G.Lay and H.Zimmer: Algorithmic Number Theory, ANTS-I, LNCS 877, 1994, 250-263.

[11] D.Bailey and C.Paar: Proc. Asiacrypt2000, **LNCS 1976**, 2000, 248-258.

[12] D.Han, K.Yoon, Y.Park, C.Kim, and J.Lim: Proc. of SAC2002, LNCS 2595, 2003, 369-384.

[13] Y.Nogami and Y.Morikawa: Tech. of IEICE, IT2001-44, 2001, 7-12.

[14] T.Danno, Y.Nogami, and Y.Morikawa: The 24th Symposium on Information Theory and its Applications (SITA2001), vol.1 of 2, 2001, 355-358.

[15] http://www.exp-math.uni-essen.de/~diem/english.html

[16] M.Ciet, J.Quisquater, and F.Sica: Indocrypto2001, LNCS 2247, 2001, 108-116.

[17] K.Nagao, S.Arita, K.Matsuo, and M.Shimura: The 2004 Symposium on Cryptography and Information Security (SCIS2004), 2004, 897-902.

[18] S.Arita: The 2004 Symposium on Cryptography and Information Security (SCIS2004), 2004, 903-908.

[19] A.Menezes: Elliptic Curve Public Key Cryptosystems, Kluwer Academic Publishers, 1993.

[20] T.Hiramoto, Y.Nogami, and Y.Morikawa, IEICE Trans. **J84-A** no.5, 2000, 633-641.

Table 8Examples of elliptic curve of order 2×(large prime number) over $F_{2^{178}}$ and $F_{2^{226}}$

	Example(7)	Example(8)
characteristic p	2	2
coefficient field, definition field	$F_{2^{89}}$, $F_{2^{178}}$	$F_{2^{113}}$, $F_{2^{226}}$
modular polynomials†	$x^{89}+x^{38}+1,\ x^2+x+1$	$x^{113}+x^9+1,\ x^2+x+1$
Satoh's algorithm over $F_{2^{89}}/F_{2^{113}}$ $[ms]$††	23	48
average time [s]†††	1.1	3.2
original defining equation $E(x,y)$	$y^2+xy+x^3+(9B)_{16}$	$y^2+xy+x^3+(55C)_{16}$
twisted defining equation $E'(x,y)$‡	$y^2+xy+x^3+\omega x+(9B)_{16}$	$y^2+xy+x^3+\omega x+(55C)_{16}$
order $\#E'(F_{2^{178}})/\#E'(F_{2^{226}})$	2× (176 bits prime number)	2× (224 bits prime number)

† The authors adopted "tower field" technique[24]. †† Computation time of Satoh's algorithm[25].

††† Average time for generating one prime order elliptic curve by the proposed algorithm.

‡ ω is a zero of the modular polynomial x^2+x+1 and satisfies $\mathrm{Tr}(\omega)=1$ (*see* B).

Table 9: The order $\#E'$ of each example

Example(1)	1461501636868331575725438632312124851656849706689
Example(2)	9578097130457539314853924949751110535452745423097924 9
Example(3)	392318858461723299602733168476816100884047506580631580769
Example(4)	146150163733090291820371396880191283504604784 3993
Example(5)	37414441915671114706014502201304509988866079778252 1
Example(6)	38312388521647221458958679158807248410418001516484048 9
Example(7)	38312388521647221458958675575869604690156212135087644 2
Example(8)	10783978666860255917866806034807850351367391064698932991536038851904 2

[21] R.Solovay and V.Strassen: SIAM Journal on Computing, 1977, 84-85.

[22] http://indigo.ie/˜mscott/

[23] R.Lidl and H.Niederreiter: Finite Fields, Encyclopedia of Mathematics and Its Applications, Cambridge University Press, 1984.

[24] B.Selcuk: thesis of Worcester Polytechnic Institute, 2003. http://www.wpi.edu/Pubs/ETD/Available/etd-0501103-132249/

[25] http://argote.ch/Demos.html

A It is possible that $\#E'(F_{q^{2^c}})$ is a prime number

For ease of explanation, let us consider the case that $c=2$. In this case, using the base trace t_1, the orders are given as follows ;

$$\#E(F_q) = q+1-t_1, \tag{16}$$
$$\#E'(F_q) = q+1+t_1, \tag{17}$$

$$\begin{aligned}\#E(F_{q^2}) &= q^2+1-t_2\\ &= (q+1-t_1)(q+1+t_1)\\ &= \#E(F_q)\#E'(F_q),\end{aligned} \tag{18}$$

$$\begin{aligned}\#E'(F_{q^2}) &= q^2+1+t_2\\ &= q^2+1-2q+t_1^2\\ &= (q-1)^2+t_1^2,\end{aligned} \tag{19}$$

$$\begin{aligned}\#E(F_{q^4}) &= q^4+1-t_4\\ &= (q^2+1-t_2)(q^2+1+t_2)\\ &= (q+1-t_1)(q+1+t_1)(q^2+1+t_2)\\ &= \#E(F_q)\#E'(F_q)\#E'(F_{q^2}),\end{aligned} \tag{20}$$

$$\begin{aligned}\#E'(F_{q^4}) &= q^4+1+t_4 \\ &= q^4+1-2q^2+t_2^2 \\ &= (q^2-1)^2+t_2^2. \end{aligned} \quad (21)$$

In general, we have

$$\#E(F_{q^{2^c}}) = \#E(F_q)\prod_{i=0}^{c-1}\#E'(F_{q^{2^i}}), \quad (22)$$

$$\#E'(F_{q^{2^c}}) = \left(q^{2^{c-1}}-1\right)^2+t_{2^{c-1}}^2. \quad (23)$$

Therefore, undesirable properties shown in Eq.(6) and Eq.(9) are all distilled to $\#E(F_{q^{2^c}})$. On the other hand, it is possible that the twisted order $\#E'(F_{q^{2^c}})$ is a prime number, and of course it is possible that $\#E'(F_{q^{2^c}})$ has a large prime factor.

B The proof of $\mathrm{Tr}(\omega)=1$

In these cases, since ω is a zero of the modular polynomial x^2+x+1 over F_2, ω satisfies

$$\omega^2+\omega=1. \quad (24)$$

In addition, since x^2+x+1 is irreducible over F_2, ω belongs to F_{2^2} and therefore satisfies

$$\omega^{2^2}=\omega. \quad (25)$$

On the other hand, when the definition field is $F_{2^{178}}$, $\mathrm{Tr}(\omega)$ is represented by

$$\begin{aligned}\mathrm{Tr}(\omega) &= \sum_{i=0}^{177}\omega^{2^i} \\ &= \sum_{i=0}^{88}\omega^{2^i}+\sum_{i=89}^{177}\omega^{2^i} \\ &= \sum_{i=0}^{88}\omega^{2^i}+\sum_{i=0}^{88}\omega^{2^{89+i}} \\ &= \sum_{i=0}^{88}\omega^{2^i}+\sum_{i=0}^{88}\left(\omega^{2^{89}}\right)^{2^i}, \end{aligned} \quad (26)$$

substituting Eq.(25) into the second term of the right-hand side of Eq.(26),

$$\begin{aligned} &= \sum_{i=0}^{88}\omega^{2^i}+\sum_{i=0}^{88}\left(\omega^2\right)^{2^i} \\ &= \sum_{i=0}^{88}(\omega+\omega^2)^{2^i}, \end{aligned} \quad (27)$$

and substituting Eq.(24), then we get $\mathrm{Tr}(\omega)=1$. In the same, we can prove the case of Example(8).

C ω is a QNR in F_{p^4}, F_{p^2}

In C, we only consider the case of Example(1) into detail. 7 is a QNR in F_p, where $p=2^{40}-87$, since the following relation holds ;

$$7^{(p-1)/2}=-1. \quad (28)$$

Then, the modular polynomial x^4-7 becomes irreducible over $F_{2^{40}-87}$[23],[11]. In the same way, we can check whether or not ω, that is a zero of x^4-7, is a QNR in F_{p^2} as follows ;

$$\omega^{(p^4-1)/2}=\begin{cases}1 & \text{when } \omega \text{ is a QR} \\ -1 & \text{when } \omega \text{ is a QNR}\end{cases}. \quad (29)$$

According to the relation between the coefficients and zeros of the modular polynomial, we have

$$\omega^{1+p+p^2+p^3}=-7, \quad (30)$$

and we can develop the left-hand side of Eq.(29) as

$$\begin{aligned}\omega^{(p^4-1)/2} &= (\omega^{1+p+p^2+p^3})^{(p-1)/2} \\ &= (-7)^{(p-1)/2}, \end{aligned} \quad (31)$$

noting that $4 \mid (p-1)$ in this case,

$$\begin{aligned} &= (-1)^{(p-1)/2}\cdot 7^{(p-1)/2} \\ &= 7^{(p-1)/2}, \end{aligned} \quad (32)$$

substituting Eq.(28), finally we get $\omega^{(p^4-1)/2}=-1$, therefore ω is a QNR in F_{p^4}. Since every characteristic p in Table 6 and Table 7 satisfies $4 \mid (p-1)$, in the same way we can prove for the other Examples(2)$\sim$(6).