Fast Generation of Elliptic Curves with Prime Order over $F_{p^{2^c}}$

Yasuyuki Nogami, Yoshitaka Morikawa*

1 Introduction

In the elliptic curve cryptosystem(ECC), the order of elliptic curve must be a large prime or divisible by a large prime in order to ensure its security. In practice, it is said that the order should be 160 bits long at least, in addition, it is preferred to be a prime number[1]. Correspondingly, if ECC is defined over an extension field F_{p^m}, the pair of p and m has to satisfy $m \log p \geq 160$[2]. For example, we may adopt a 30 bits long prime and 6 as p and m, respectively. As compared to the ECC defined over a prime field, the ECC defined over an extension field has some advantages, such that arithmetics operations in the definition field can be fast implemented. But, it is cautioned that there exist some insecure curves fragile to FR attack[4] and Weil descent attack[5]. For the latter attack, the following results have already been reported:

· When $p = 2$, most elliptic curves are irrelevant for ECC[5].
· When $p = 3$, some elliptic curves are irrelevant for ECC[6].
· When p is odd and $m = 3$ or 5 or 7, all elliptic curves are irrelevant for ECC[7].

For the other cases, such that the extension degree is a power of 2, Weil descent attack has not been developed sufficiently and there are many cases of curves and fields for which the attack techniques cannot apply[8]. Therefore, this paper takes only FR attack into account.

In this paper, however, we start from the elliptic curves whose coefficient and definition fields are a prime field F_p ($p \neq 2, 3$) and its extension field F_{p^m}, respectively. And we devise the elliptic curves so as to have all the following features by using some techniques.

· The elliptic curve can resist against FR attack.
· Encryption/decryption can be fast performed.
· The definition field can be compactly implemented.
· The elliptic curve has a prime order.

By discussing these features separately, some mathematical conditions for elliptic curve so as to have all features are deduced. The order of elliptic curve which we start from is unfortunately composite, in order to overcome this problem, the technique called *twist*[9] is introduced. After that, it is shown that the order of the *twisted* elliptic curve can be prime only when the extension degree m is a power of 2, in other words, an elliptic curve with a prime order can be generated by adopting $F_{p^{2^c}}$ as its definition field and using *twist* technique, where c is a positive integer. Finally, an efficient algorithm to generate an elliptic curve which satisfies all of the given conditions and has a prime order is proposed and two concrete

* The authors are with Communication Network Engineering, Okayama University, Okayama-shi, 700-8530, E-mail: nogami@cne.okayama-u.ac.jp .

examples are shown. By using this algorithm, such an elliptic curve with a prime order can be generated within 1 *second* on PentiumIII(800MHz) processor, where the characteristc p and the extension degree m are a 30 bits long prime and 8, respectively.

2 Fundamentals

This section deals with the fundamentals of arithmetics on an elliptic curve, FR attack, and efficient software implementation of the ECC.

2.1 Arithmetics on an elliptic curve

2.1.1 Coefficient field and definition field

An elliptic curve over finite field F_q is defined as the set of solutions to the equation

$$E(x, y) = y^2 - x^3 - ax - b = 0, \tag{1}$$

with $a, b \in F_q$ and the characteristic of F_q is not equal to 2 or 3. The solutions (x, y) to Eq.(1) are called F_q-rational points when the coordinates of x and y lie in F_q. This paper deals with elliptic curves that the coordinates lie in some extension field but coefficients a, b in its proper subfield. In order to describe the difference clearly, we call the field of a, b coefficient field and that of coordinates x, y definition field.

2.1.2 Order and trace of elliptic curve

F_q-rational points on Eq.(1) form an additive Abelian group. In this paper, this group and its order is denoted by $E(F_q)$ and $\#E(F_q)$, respectively. The existing range of $\#E(F_q)$ is given by Hasse's theorem[3]; let us consider $t = q + 1 - \#E(F_q)$, t satisfies

$$2\sqrt{q} \geq t \geq -2\sqrt{q}, \tag{2}$$

where t is called the trace of $E(F_q)$. The order or trace of elliptic curve is closely related to the security of the ECC[1], which will be discussed in Section2.2. Therefore, the problem of determing order $\#E(F_q)$ is of critical importance in cryptographic applications. As the efficient order computation algorithm, Schoof's algorithm[10] and SEA algorithm[1] are well known. If the coefficient field is F_p but the definition field is its extension field F_{p^m}, $\#E(F_{p^m})$ can be obtained by the following steps, where *base order* is the number of F_p-rational points on the curve.

1. Compute *base order* $\#E(F_p)$.
2. Determine the objective order by using the following Weil's theorem with the previously computed $\#E(F_p)$.

Theorem 1 *[3] Let coefficient and definition fields be a prime field F_p and its extension field F_{p^m}, respectively. And let $t_1 = p + 1 - \#E(F_p)$. Then,*

$$\#E(F_{p^m}) = p^m + 1 - (\alpha^m + \beta^m), \tag{3}$$

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

Therefore, it is enough to compute only base order by using SEA algorithm.

2.2 FR attack

By using FR attack[4], we can reduce the elliptic curve discrete logarithm problem (ECDLP) on $E(F_q)$ to the discrete logarithm problem on F_{q^k} of a certain extension degree k. It is known that only the ECDLP which satisfies any of the following conditions can be easily reduced by the attack, where t is the trace of $E(F_q)$ and l is a certain integer.

· $E(F_q)$ is super-singular. · Trace $t = 2$.

Let p be the characteristic of F_q, $E(F_q)$ is said to be *super-singular* if p divides its trace t.

2.3 Efficient software implementation

According to Eq.(2), $\#E(F_q)$ exists in the range between $q+1-2\sqrt{q}$ and $q+1+2\sqrt{q}$. This fact indicates that q, that is the order of definition field F_q, must be 160 bits at least. Therefore, we have to implement arithmetics in an extension field of such a large order. For software implementation, optimal extension field(OEF)[11] and all-one polynomial field(AOPF)[12] are known to realize fast implementation. In these fields, we restrict their characteristic and the modular polynomial for the extension as follows;

1. Characteristic p is a pseudo Mersenne prime of computer's *word size*, where we call a prime in the form of $2^n \pm c$ $(n/2 \geq \log_2 c)$ pseudo Mersenne prime.
2. The modular polynomial is an irreducible binomial(OEF) or all-one polynomial(AOPF).

3 Conditions

In this section, we start at first from the elliptic curves whose coefficient and definition fields are a prime field F_p $(p \neq 2, 3)$ and its extension field F_{p^m}, respectively, and show the sufficient conditions for the elliptic curve that has all of the features shown in Section1.

3.1 Conditions to resist against FR attack

From Theorem1 and the definition of the super-singular elliptic curve, if $E(F_{p^m})$ is super-singular, then its trace $\alpha^m + \beta^m$ must be divisible by p. And then, $\alpha^m + \beta^m$ can be written with $\alpha + \beta$, $\alpha\beta$, by using Dickson polynomials of the first kind[14],

$$\alpha^m + \beta^m = D_m(\alpha + \beta, \alpha\beta), \tag{4}$$

where Dickson polynomial $D_m(X, a)$ is defined by

$$D_m(X, a) = \sum_{i=0}^{\lfloor m/2 \rfloor} \frac{m}{m-i} \binom{m-i}{i} (-a)^i X^{m-2i}, \tag{5}$$

and m is referred to as the degree of Dickson polynomial. In Eq.(5), notation $\lfloor \cdot \rfloor$ shows the maximum integer less than or equal to $\cdot$. Substituting $\alpha + \beta$ and $\alpha\beta$ to t_1 and p, respectively, $\alpha^m + \beta^m$ is given with t_1 and p as follows;

$$\alpha^m + \beta^m = \sum_{i=0}^{\lfloor m/2 \rfloor} \frac{m}{m-i} \binom{m-i}{i} (-p)^i {t_1}^{m-2i}. \tag{6}$$

According to Eq.(6), $\alpha^m + \beta^m$ is divisible by p if and only if t_1 is divisible by p. On the other hand, the existing range of t_1 is given by considering $t = t_1$ in Eq.(2),

$$2\sqrt{p} \geq t_1 \geq -2\sqrt{p}. \tag{7}$$

In this range, the only $t_1 = 0$ is divisible by p. Therefore, $E(F_{p^m})$ is super-singular if and only if $t_1 = 0$, and accordingly the following condition is at first needed.
Condition(1) : $t_1 \neq 0$**.**

According to Section2.2, if the trace of an elliptic curve is equal to 2, it is also irrelevant for using in the ECC. To be more detailed, if the trace is equal to 2, the order of the elliptic curve becomes even, of course it is not prime number. In this paper, as a sufficient condition that trace is not equal to 2, we adopt the condition that the trace is not even.

At first, let us develop $\alpha^m + \beta^m$ as

$$\alpha^m + \beta^m = (\alpha + \beta)(\alpha^{m-1} + \beta^{m-1}) - \alpha\beta(\alpha^{m-2} + \beta^{m-2}). \tag{8}$$

Eq.(8) gives a successive expression in regard to the degree of Dickson polynomial;

$$D_m(t_1, p) = t_1 D_{m-1}(t_1, p) - pD_{m-2}(t_1, p), \tag{9}$$

where $D_0(t_1, p)$ and $D_1(t_1, p)$ are given by

$$D_0(t_1, p) = \alpha^0 + \beta^0 = 2, \quad D_1(t_1, p) = \alpha^1 + \beta^1 = t_1. \tag{10}$$

From the above relations, we can deduce the condition that $D_m(t_1, p)$ is not even, more accurately, a sufficient condition that trace is not equal to 2 as follows;
Condition(2) : $m \not\equiv 0 \pmod 3$ **and** t_1 **is odd.**

3.2 Requirements for fast implementation

In order to realize fast implementation, we should choose the definition field F_{p^m} from extension fields introduced in Section2.3, for example OEF. According to the discussion in Section2.3, Table 1 shows the rough settings of degree m and characteristic p versus to computer's *word size*, in which p and m satisfy $m \log p \geq 160$.

Table 1: Degree m and characteristic p versus to processor's *word size*

word size [bit]	p [bit]	m
16	$10 \sim 16$	$16 \sim 10$
32	$20 \sim 32$	$8 \sim 5$
64	$40 \sim 64$	$4, 3$

Since ECCs need addition between rational points, checking of quadratic power residue, and calculation of square root, the pair of p and m must be selected from Table 1 so as to perform these operations fast. Addition between rational points consists of additions, multiplications, and an inversion in F_{p^m}[1]. Therefore, fundamental arithmetics in F_{p^m} should be performed fast. For such a requirement, extension degree m is preferred to be a composite number since some of the arithmetics in F_{p^m} can be replaced by the corresponding arithmetics in its proper subfield[12],[13]. Especially, $m = 2^i (i \geq 1)$ is the most effective[2].

In calculating the square root of a quadratic power residue, the multiplicity d_2 defined as follows is preferred to be small[15].

$$p^m - 1 = 2^{d_2} T, \ T \text{ is an odd number.} \tag{11}$$

For example, comparing $(p, m) = (2^{31} - 1, 2), (2^{28} + 3, 2)$ which are the pairs of p and m, d_2 becomes 32 in the former case and 3 in the latter case. Therefore, the square root computation in the latter case may be about 10-fold faster than that in the former case.

Concluding this section, the following two requirements are also mentioned;
Requirement(3) : m is a power of 2.
Requirement(4) : Multiplicity d_2 is small.

3.3 Conditions for prime order

When the coefficient and definition fields are a prime field and its extension field, respectively, then $\#E(F_{p^m})$ is given by

$$\#E(F_{p^m}) = p^m + 1 - D_m(t_1, p), \tag{12}$$

as seen in Section3.1. In this case, the following equation always holds for an arbitrary factor m' of the extension degree m[1].

$$\#E(F_{p^{m'}}) \mid \#E(F_{p^m}), \tag{13}$$

where $X \mid Y$ means that X divides Y. Eq.(13) indicates not only that $\#E(F_{p^m})$ is not prime, but also that the largest prime factor of $\#E(F_{p^m})$ becomes considerably smaller than 160 bits long even if $m \log p$ is about 160. Therefore, we have to adopt a further larger extension field for secure ECC. But it is not desirable with respect to the third feature shown in Section1. This defect is due to the constrained setting that coefficient field F_p is proper subfield of definition field F_{p^m}. In order to overcome this undesirable property, we allow coefficient field not to be proper subfield of definition field F_{p^m} and then adopt a technique called *twist*[9].

For an original defining equation:

$$E(x, y) = y^2 - x^3 - ax - b = 0 \quad a, b \in F_p, \tag{14}$$

the new defining equation is introduced,

$$E'(x, y) = y^2 - x^3 - aA^2 x - bA^3 = 0, \tag{15}$$

where A is a non-zero element of F_{p^m}. The above $E'(x, y)$ is said to be a *twist* of $E(x, y)$. Corresponding to whether A is a quadratic power residue(QPR) or a quadratic power non residue(QPNR), the order of $E'(x, y)$ is given as follows;

$$\#E'(F_{p^m}) = \begin{cases} p^m + 1 - D_m(t_1, p) & ; \text{ if } A \text{ is QPR,} \quad (16a) \\ p^m + 1 + D_m(t_1, p) & ; \text{ if } A \text{ is QPNR.} \quad (16b) \end{cases}$$

By changing $E(x, y)$ to $E'(x, y)$, we can easily extend the coefficient field F_p to the extension field F_{p^m}, and moreover its order $\#E'(F_{p^m})$ can be determined with only t_1, where t_1 can be easily obtained by using Weil's Theorem with the base order $\#E(F_p)$. Comparing Eq.(16a) and Eq.(12), however, if A is a quadratic power residue, the preceding undesirable property remains since Eq.(16a) is not changed from Eq.(13). In the following, we consider that $E'(F_{p^m})$ is twisted with QPNR; accordingly the order $\#E'(F_{p^m})$ is given by Eq.(16b).

Now, let us examine whether or not $\#E'(F_{p^m})$ can be prime (or, divisible by a large prime or not). If we suppose that m has an odd factor $m' \neq 1$, then order $\#E'(F_{p^m})$ also has the same undesirable property as the same of Eq.(13). On the other hand, let extension degree m be 2^c, where c is an positive integer, if we calculate $\#E'(F_{p^{2^c}})$ by using

$$\#E'(F_{p^{2^c}}) = p^{2^c} + 1 + D_{2^c}(t_1, p), \tag{17a}$$

$$D_{2^c}(t_1, p) = \sum_{i=0}^{2^{c-1}} \frac{2^c}{2^c - i} \binom{2^c - i}{i} (-p)^i {t_1}^{2^c - 2i}, \tag{17b}$$

there exist many t_1's such that $\#E'(F_{p^{2^c}})$ becomes prime[16]. The absolute values of such t_1's are tabulated in Table 2. The reason for using the absolute values is that $\#E'(F_{p^{2^c}})$ in either case of $\pm t_1$ are equal to each other, which is easily understood from Eq.(17b). For example, in the case of $(p, m) = (2^{28} + 3, 8)$, $\#E'(F_{p^8})$ becomes prime when $t_1 = 59$. Concluding this section, elliptic curves with a prime order can be generated under the following condition;
Condition(5) : Use $E'(F_{p^{2^c}})$ with a prime order.

Table 2: t_1's such that $\#E'(F_{p^{2^c}})$ becomes prime

p	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	$59, 79, 91, 111, \cdots$

4 Algorithm and its performance

In this section, an efficient algorithm to generate elliptic curves which satisfies all of the conditions (1) $\sim$ (5) shown in Section3 is proposed. Then, we evaluate its performance in regard to computation time. Last, two concrete examples of such an elliptic curve are shown.

4.1 Algorithm

Since the conditions (3), and (4) are initial settings. In practice, calculations and condition checks are required only for (1),(2) and (5). For (1) and (2), if $\#E'(F_{p^{2^c}})$ is prime, then t_1 must at least be odd from Eq.(17). Conversely, if t_1 is odd, then (1) and (2) are always satisfied. However, these two conditions are given for $E(F_{p^{2^c}})$ to resist against FR attack but not for its twist $E'(F_{p^{2^c}})$. In order to reconsider these conditions (1) and (2) from the view point of $E'(F_{p^{2^c}})$, let us recall the originals of these conditions, that is,

(1) is the necessary and sufficient condition for elliptic curves not to be super-singular,
(2) is a sufficient condition for the trace not to be even.

Comparing Eq.(16a) and Eq.(16b), the only difference between $\#E(F_{p^{2^c}})$ and its twist version $\#E'(F_{p^{2^c}})$ is the sign of $D_{2^c}(t_1, p)$. Therefore, if t_1 is odd, $E'(F_{p^{2^c}})$ is also not super-singular and its trace is also not even. Consequently, in order to devise an efficient algorithm to generate $E'(F_{p^{2^c}})$ with a prime order, an elliptic curve $E(F_p)$ whose trace t_1 is odd must be efficiently generated. It is noted that $E(F_p)$ is an elliptic curve whose coefficient and definition fields are both prime field F_p.

We now introduce a class of elliptic curves whose defining equation $E(x, y)$ is given by using an irreducible polynomial $E(x, 0)$ over F_p. In the following, we call an elliptic curve in this class a *no two-torsion* elliptic curve. For example, the following elliptic curve is no two-torsion elliptic curve over F_5.

$$E(x, y) = y^2 - x^3 - x - 4 = 0 \tag{18}$$

Let $E(F_p)$ be a no two-torsion elliptic curve over definition field F_p, then, its trace t_1 is always odd[1]. Using this property, an efficient algorithm to generate an elliptic curve $E'(F_{p^{2^c}})$ satisfying all of the conditions becomes as follows, where it should be noted that the conditions (3),(4) are automatically satisfied as a result of extension degree restriction $m = 2^c$.

Algorithm: Generate an elliptic curve satisfying all of the conditions.

Input: Characteristic p satisfying condition (5), extension degree 2^c.
Output: An elliptic curve $E'(F_{p^{2^c}})$ with a prime order.

1. Choose coefficients $a, b \in F_p$ of 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.
2. Compute base order $\#E(F_p)$ of the no two-torsion elliptic curve $E(F_p)$ by Schoof's or SEA algorithm. Then, determine $t_1 = p + 1 - \#E(F_p)$.
3. Determine $D_{2^c}(t_1, p)$ by Eq.(17b), then test whether or not order $\#E'(F_{p^{2^c}})$ determined by Eq.(17a) is prime. If it is not prime, then return to Step1. Otherwise, go to Step4.
4. Determine twisted defining equation Eq.(15) with some quadratic power non residue $A \in F_{p^{2^c}}$. Then, $E'(F_{p^{2^c}})$ is an objective elliptic curve with a prime order.

4.2 Experimental results and concrete examples

In this section, we evaluate the performance of the proposed algorithm in regard to computation time. We adopted an irreducibility test algorithm[17] for Step1, Schoof's algorithm[10] for Step2, and a primary test algorithm[18] for Step4, respectively. Then, we implemented these algorithms on PentiumIII(800MHz) processor by programming in C language. Considering F_{p^8} as the definition field, we selected pseudo Mersenne primes $2^{24} - 3, 2^{28} + 3$, and $2^{29} - 3$ as its characteristic p. It is noted that each of these primes satisfies condition (5).

Now, let us estimate the probability of generating a no two-torsion elliptic curve at random, that is to say, the probability of success in one iteration of Step1. The number of possible pairs of the coefficient a, b is p^2. And also, the number of F_p-irreducible polynomials which takes the form of $x^3 + ax + b$ is $(p^2 - 1)/3$[19]. Therefore, the probability can be estimated by about $1/3$. Table 3 shows the average times of passing an irreducibility test, Step1: an irreducible elliptic curve generation, and Step2: an order computation by using Schoof's algorithm, respectively. From Table 3, we can see that Step1 requires approximately three times the duration required by the irreducibility test and also achieves much faster than Step2. In the following, we discuss without accounting for the computation time required by Step1.

Next, let us experimentally estimate the probability that $E'(F_{p^{2^c}})$ will be prime in Step3, in other words, that an elliptic curve with a prime order will be generated by only one order computation. For $2^c = 2, 4, 8, 16$, Table 4 shows the number of t_1's such that $\#E'(F_{p^{2^c}})$ becomes prime. In addition, the following facts have been already known:

Table 3: Average times of an 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

Table 4: The number of t_1's such that $\#E'(F_{p^{2^c}})$ becomes prime

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

· Trace t_1 is almost uniformly distributed in the range of Eq.(7)[9].
· In the range of Eq.(7), there are $\lfloor 2\sqrt{p} \rfloor$ distinct odd numbers.
· The trace t_1 is odd if and only if the elliptic curve is a no two-torsion elliptic curve[1].

Based on these facts, the probability can be estimated by dividing the numbers tabulated in Table 4 by $\lfloor 2\sqrt{p} \rfloor$, respectively. The results are shown in Table 5. For example, in the case of

Table 5: 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

$(p,m)=(2^{24}-3,2)$, since the probability is 0.051, an elliptic curve with a prime order will be generated by about 20 iterations from Step1 to Step3. And also, from the table, we can see 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 theory.

Last, Table 6 exhibits the average times of generating an objective good elliptic curve $E'(F_{p^{2^c}})$. For example, in the case of $(2^{28}+3,8)$, it takes 537.1ms on average. In the case of $(2^{24}-3,8)$, the average becomes 380.9ms, which is faster than the first case's. This is because an order computation in the second case is faster than the first case's as shown in Table 3. On the other hand, in the case of $(2^{28}+3,16)$, the average time becomes 977.5 ms, which is slower than the first case's. This is because the probability of $E'(F_{p^{2^c}})$ being prime in the third case is lower than the first case's as shown in Table 5.

Concluding this section, in the case of $2^c=8$ and any of the three primes, which will be one of the most practical cases, the time tabulated in Table 6 is less than 1 *second*. Therefore, we can conclude that the proposed algorithm is sufficiently practical as based on the experimental data. Finally, Table 7 shows two concrete examples of an elliptic curve

Table 6: Average times of generating an elliptic curve with a prime order $[ms]$

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

satisfying all of the conditions shown in Section3, in other words, these elliptic curves have all of the features described in Section1.

Table 7: Concrete examples of an elliptic curve satisfying all of the conditions

	Example 1	Example 2
field type	OEF	OEF
characteristic	$2^{24} - 3$	$2^{29} - 3$
extension degree	8	8
modular polynomial	$x^8 - 2$	$x^8 - 2$
QPNR A	ω †	τ †
defining Equation	$y^2 - x^3 - 3\omega^2 x - 10\omega^3$	$y^2 - x^3 - 3\tau^2 x - 195\tau^3$
order $\#E'$	191 bits prime	231 bits prime

† ω and tau are zeros of the modular polynomial.

5 Conclusion

In this paper, an efficient algorithm to generate an elliptic curve which satisfies all of the features shown in Section1 was proposed by using *no two-torsion* elliptic curves, and the concrete examples were shown.

By using this algorithm, such an elliptic curve with a prime order could be generated within 1 *second* on PentiumIII(800MHz) processor, where the characteristc p and the extension degree m are a 30 bits long prime and 8, respectively. From the experimental data shown in this paper, we can conclude that the proposed algorithm is sufficiently practical.

References

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

[2] T.Kobayashi, K.Aoki, and F.Hoshino, "OEF Using a Successive Extension," Proc. The 2000 Symposium on Cryptography and Information Security, no.B02, 2000.

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

[4] G.Frey and H.Rück,"A Remark Concerning m-Divisibility and the Discrete Logarithm in the Divisor Class Group of Curves," Math. Comp., vol.62, pp.865-874, 1994.

[5] P.Gaudry, F.Hess, and N.Smart,"Constructive and destructive facets of Weil descent on elliptic curves," Hewlett Packard Lab. Technical Report, HPL-2000-10, 2000.

[6] S.Arita,"Weil descent of Elliptic Curves over Finite Fields of Characteristic Three," Proc. Crypto'98, LNCS 1462, pp.472-485, 1998.

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

[8] S.D.Galbraith, "Weil Descent of Jacobians," http://www.cs.bris.ac.uk/~nigel/weil_descent.html.

[9] K.Horiuchi, Y.Futa, R.Sakai, M.Kaneko, and M.Kasahara, "Construction of Elliptic Curves with Prime Order and Estimation of Its Complexity," IEICE Trans., vol.J82-A, no.8, pp.1269-1277, 1999.

[10] R.Schoof,"Elliptic Curves over Finite Fields and the Computation of Square Roots Mod p," Math. Comp., vol.44, no.170, pp.483-494,1985.

[11] D.B.Bailey and C.Paar, "Optimal Extension Fields for Fast Arithmetic in Public-Key Algorithms," Proc. Asiacrypt2000, LNCS 1976, pp.248-258, 2000.

[12] A.Saito, T.Hiramoto, T.Danno, Y.Nogami, and Y.Morikawa, "Extension Fields by Using $(x^{m+1}-1)/(x-1)$ as the Modulus for High-Speed Arithmetic," IEICE Technical Report, ISEC2000-119, pp.129-134, 2000.

[13] M.Morii and M.Kasahara, "Efficient Construction of Gate Circuit for Computing Multiplicative Inverses over $GF(2^m)^n$," IEICE Trans. Fundamentals, vol.E72-A, pp.37-42, 1989.

[14] S.Abhyankar, S.Cohen, and M.Zieve, "Bivariate Factorizations Connecting Dickson Polynomials and Galois Theory," Trans. Amer. Math. Soc., vol.352, no.6, pp.2871-2887, 2000.

[15] T.Danno, A.Saito, Y,Nogami, and Y.Morikawa, "High-Speed Algorithm for Taking Square Root in an Extension Field by Using $(x^{m+1}-1)/(x-1)$ as the Modulus," IEICE Technical Report, IT2001-30, pp.31-36, 2001.

[16] T.Danno, Y.Nogami, and Y.Morikawa, "Conditions of Characteristic and Trace for Rank One Elliptic Curve Twisted over $F_{p^{2m}}$," Proc. The 24th Symposium on Information Theory and Its Applications, pp.355-358, 2001.

[17] T.Hiramoto, Y.Nogami, and Y.Morikawa, "A Fast Algorithm to Test Irreducibility of Cubic Polynomial over $GF(P)$," IEICE Trans., vol.J84-A, no.5, pp.633-641, 2000.

[18] http://indigo.ie/~mscott/#Elliptic

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