Memoirs of the Faculty of Engineering, Okayama University, Vol.41, pp.11-19, January, 2007

An Algorithm for Generating Irreducible Cubic Trinomials over Prime Field

Yasuyuki Nogami[†] Yoshitaka Morikawa[†]
The Graduate School of Natural Science and Technology
Okayama University
Okayama 700-8530 Japan

(Received November 14, 2006)

This paper proposes an algorithm for generating irreducible cubic trinomials in the form $x^3 + ax + b,\ b \in \mathbf{F}_p$, where a is a certain fixed non-zero element in the prime field $\mathbf{F}_p$. The proposed algorithm needs a certain irreducible cubic trinomial over $\mathbf{F}_p$ to be previously given as a generator; however, the proposed algorithm can generate irreducible cubic polynomials one after another by changing a certain parameter in $\mathbf{F}_p$. In this paper, we compare the calculation cost and the average computation time for generating an irreducible cubic polynomial, especially trinomial, among Hiramoto et al. irreducibility testing algorithm, Berlekamp-Massey minimal polynomial determining algorithm, and the proposed algorithm. From the experimental results, it is shown that the proposed algorithm is the fastest among the three algorithms for generating irreducible cubic trinomials.

Keywords: irreducible cubic polynomial, minimal polynomial

1 Introduction

The elliptic curve cryptosystem(ECC) is one of the public key cryptosystems[1]. It is said that a prime order elliptic curve is suitable for constructing an elliptic curve cryptosystem from the viewpoints of security and implementation. Therefore, several algorithms for generating prime order elliptic curves have been proposed[2]-[4]. In general, the defining equation is written as

$$E(x, y) = x^3 + ax + b - y^2,\ a, b \in \mathbf{F}_p, \tag{1}$$

where this paper deals with the characteristic $p > 3$. For the above defining equation, it is well known that $E(x, 0)$ is an irreducible cubic trinomial over $\mathbf{F}_p$ if the order of elliptic curve is a prime number[1]. To be more detailed, $E(x, 0)$ is an irreducible cubic trinomial over $\mathbf{F}_p$ if and only if the order of elliptic curve is an odd number. Therefore, for the purpose of generating a lot of secure prime order elliptic curves, it is helpful that we can fast generate irreducible cubic trinomials one after another[5]. Our motivation comes from this fact. As a related work, the authors have proposed an algorithm for generating prime order elliptic curves[4] in which the authors used an irreducibility testing algorithm in order to prepare irreducible cubic trinomial over $\mathbf{F}_p$.

An irreducibility testing algorithm for an arbitrary degree polynomial over prime field has been already known[6] and recently Hiramoto et al. proposed a fast irreducibility testing algorithm for a cubic polynomial over the prime field $\mathbf{F}_p$[7]. The related work[4] used Hiramoto el al. irreducibility testing algorithm. These irreducibility testing algorithms need polynomial multiplications and polynomial modulo operations over $\mathbf{F}_p$. On the other hand, Berlekamp-Massey algorithm[8] is well known as an algorithm for determining the minimal polynomial of a given pseudo-random sequence.

Berlekamp-Massey algorithm does not need polynomial modulo operations; however, it needs multiplications and inversions of non-zero elements in $\mathbf{F}_p$. Berlekamp-Massey algorithm can be applied for iteratively generating irreducible polynomials from the given pseudo-random sequence.

This paper proposes an algorithm for generating irreducible cubic trinomials in the form

$$x^3 + ax + b,\ b \in \mathbf{F}_p, \tag{2}$$

where a is a certain fixed non-zero element in $\mathbf{F}_p$. The proposed algorithm can generate irreducible cubic polynomials one after another by changing a certain parameter in $\mathbf{F}_p$. Since this algorithm is based on the minimal polynomial determination[8], it needs a certain irreducible cubic polynomial over $\mathbf{F}_p$ to be previously given as a generator. In this paper, we compare the calculation cost and the average computation time for generating an irreducible cubic polynomial, especially trinomnial, among Hiramoto et al. irreducibility testing algorithm, Berlekamp-Massey algorithm, and the proposed algorithm. From experimental results, it is shown that the proposed algorithm is the fastest among these three algorithms. To be more detailed, when the characteristic p is $2^{257} - 93$, the proposed algorithm is about 500 times and 22 times faster than Hiramoto et al. algorithm and Berlekamp-Massey algorithm, respectively, where the authors used PentiumIII (800MHz) and NTL[9] for programming.

Throughout this paper, $X \mid Y$ and $X \nmid Y$ mean that Y is divisible and not divisible by X, respectively. $\mathbf{F}_p$ and $\mathbf{F}_{p^3}$ mean the prime field of the characteristic p and its third-degree extension field, respectively. $\mathbf{F}_p^*$ and $\mathbf{F}_{p^3}^*$ mean the multiplicative groups of $\mathbf{F}_p$ and $\mathbf{F}_{p^3}$, respectively. p denotes a prime number larger than 3 and polynomials are monic.

2 Fundamentals

This section briefly introduces quadratic power residue, non residue, and trace in finite field.

2.1 Quadratic power residue and non residue

Let us consider a finite field $\mathbf{F}_{p^m}$ whose characteristic p is an odd prime number. For an arbitrary element $A \in \mathbf{F}_{p^m}$, we can test whether or not A is a quadratic power residue in $\mathbf{F}_{p^m}$ as follows;

$$A^{(p^m-1)/2} = \begin{cases} 0 \ \mathit{or}\ 1 & \text{quadratic power residue} \\ -1 & \text{quadratic power non residue} \end{cases}. \tag{3}$$

If A is a quadratic power residue, then A has its square roots in $\mathbf{F}_{p^m}$, otherwise, A does not have any square roots in $\mathbf{F}_{p^m}$. In this paper, we use **QR** and **QNR** as the abbreviations of quadratic power residue and non residue, respectively.

2.2 Trace of element in finite field

For an extension field $\mathbf{F}_{p^m}$ over a prime field $\mathbf{F}_p$, the trace T_A of an arbitrary element $A \in \mathbf{F}_{p^m}$ with respect to $\mathbf{F}_p$ is given as the sum of all conjugates as

$$\mathrm{T}_A = A + A^p + \cdots + A^{p^{m-1}}, \tag{4}$$

where T_A becomes a certain element in $\mathbf{F}_p$. Since this paper especially deals with irreducible cubic polynomials over $\mathbf{F}_p$, we use the following trace function $\mathrm{Tr}(x)$;

$$\mathrm{Tr}(x) = x + x^p + x^{p^2}, \tag{5}$$

accordingly the trace of $A \in \mathbf{F}_{p^3}$ with respect to $\mathbf{F}_p$ is given by $\mathrm{Tr}(A)$. The elements whose trace is $i \in \mathbf{F}_p$ are zeros of $\mathrm{Tr}(x) - i$. In addition, since Eq.(6) holds for $a, b \in \mathbf{F}_p$, we find that the trace function $\mathrm{Tr}(x)$ is a linear function over $\mathbf{F}_p$.

$$\mathrm{Tr}(ax + b) = a\mathrm{Tr}(x) + b\mathrm{Tr}(1). \tag{6}$$

Supposing that α is a zero of a monic irreducible cubic polynomial $h(x)$ over $\mathbf{F}_p$, the second-degree coefficient of $h(x)$ is given by $-\mathrm{Tr}(\alpha)$. Throughout this paper, the terminology **trace** means the trace given by the trace function Eq.(5).

3 Generation of irreducible cubic trinomials over prime field

This paper supposes that a monic irreducible cubic trinomial $f(x)$ over prime field $\mathbf{F}_p$ is previously given as

$$f(x) = x^3 + ax + b,\ a, b \in \mathbf{F}_p,\ a \neq 0. \tag{7}$$

Since this irreducible cubic trinomial $f(x)$ plays a role of the generator of other irreducible cubic trinomials over $\mathbf{F}_p$, we prepare $f(x)$ by using some irreducibility testing algorithm, such as Hiramoto et al. algorithm[7]. In this section, we propose a new algorithm for generating irreducible cubic trinomials in the following form;

$$g(x) = x^3 + ax + d,\ d \in \mathbf{F}_p, \tag{8}$$

where the constant term of $g(x)$ is only different from the given irreducible cubic trinomial $f(x)$. This algorithm is closely related to the minimal polynomial determination problem.

Let ω be a zero of $f(x)$, we have its conjugates as

$$\{\omega, \omega^p, \omega^{p^2}\}. \tag{9}$$

From the relations between zeros and coefficients of $f(x)$, we have

$$\begin{aligned}\mathrm{Tr}(\omega) = \omega + \omega^p + \omega^{p^2} &= 0, && (10a)\\ \omega\omega^p + \omega\omega^{p^2} + \omega^p\omega^{p^2} &= a, && (10b)\\ \omega\omega^p\omega^{p^2} &= -b. && (10c)\end{aligned}$$

Since arbitrary two conjugates among the three are independent[5], by using ω and ω^p, we can represent p^2 trace-zero elements in $\mathbf{F}_{p^3}$ as follows;

$$\tau = c_1\omega + c_p\omega^p,\ c_1, c_p \in \mathbf{F}_p. \qquad (11)$$

From Eq.(6) and $\mathrm{Tr}(\omega) = 0$, the trace of the above τ with respect to $\mathbf{F}_p$ becomes 0 as follows;

$$\begin{aligned}\mathrm{Tr}(\tau) &= \mathrm{Tr}(c_1\omega + c_p\omega^p)\\ &= c_1\mathrm{Tr}(\omega) + c_p\mathrm{Tr}(\omega^p) = 0. && (12)\end{aligned}$$

Since the degree of $\mathrm{Tr}(x)$ is p^2, in the form of Eq.(11) we can represent an arbitrary element in $\mathbf{F}_{p^3}$ whose trace is 0. Such an element τ becomes a zero of an irreducible cubic polynomial whose second-degree coefficient is 0.

Now, we consider the condition for the above τ to be a zero of $g(x)$ written as Eq.(8), where we note that $\mathrm{Tr}(x)/x$ does not have any zeros in $\mathbf{F}_p$. By using Eq.(10a), the conjugates of τ are represented as

$$\begin{aligned}\tau &= c_1\omega + c_p\omega^p, && (13a)\\ \tau^p &= -c_p\omega + (c_1 - c_p)\omega^p, && (13b)\\ \tau^{p^2} &= (c_p - c_1)\omega - c_1\omega^p. && (13c)\end{aligned}$$

So as to be a zero of $g(x)$, τ must satisfy

$$\begin{aligned}\mathrm{Tr}(\tau) = \tau + \tau^p + \tau^{p^2} &= 0, && (14a)\\ \tau\tau^p + \tau\tau^{p^2} + \tau^p\tau^{p^2} &= a, && (14b)\\ \tau\tau^p\tau^{p^2} &= -d. && (14c)\end{aligned}$$

By using Eqs.(10), Eqs.(13), and Eqs.(14), we have

$$\begin{aligned}(c_1 + c_p)b - c_1c_p(c_1X + c_pY) &= d, && (15a)\\ c_1^2 + c_p^2 - c_1c_p &= 1, && (15b)\end{aligned}$$

$$\begin{aligned}X &= 3\omega^2\omega^p + \omega^3 - \omega^{3p}, && (16a)\\ Y &= 3\omega\omega^{2p} - \omega^3 + \omega^{3p}, && (16b)\end{aligned}$$

where the details are shown in **A**. Then, as shown in **B** and from Eq.(16), we have

$$X = \frac{3b + \sqrt{D(f)}}{2},\ Y = \frac{3b - \sqrt{D(f)}}{2}, \quad (17a)$$
$$D(f) = -(4a^3 + 27b^2), \qquad (17b)$$

where $D(f)$ is the discriminant of $f(x)$ and it becomes a QR in $\mathbf{F}_p$ when $f(x)$ is irreducible[7]. In addition, as shown in **C**, we can represent c_1 and c_p satisfying Eq.(15b) with a parameter $t \in \mathbf{F}_p$ as follows;

$$c_1 = \frac{4t}{t^2+3},\quad c_p = \frac{3 + 2t - t^2}{t^2+3}. \qquad (18)$$

Consequently, by using Eq.(15a), Eqs.(17), and Eq.(18), we can determine the constant term d of $g(x)$. Moreover, by changing the parameter t, we can generate irreducible cubic trinomials in the form Eq.(8) one after another. The denominator t^2+3 in Eq.(18) must not be 0 and it is known that there exist t's $\in \mathbf{F}_p$ such that $t^2+3 = 0$ if and only if $3 \mid (p-1)$[7]. We can determine d also by using Eq.(13a), Eq.(14c), and Eq.(18); however, it needs arithmetic operations in $\mathbf{F}_{p^3}$.

We know that irreducible binomials over $\mathbf{F}_p$ exist if and only if $3 \mid (p-1)$[5]. Supposing that $g(x)$ is a binomial, its zero τ represented by Eq.(11) must satisfy

$$\begin{aligned}\mathrm{Tr}(\tau) = \tau + \tau^p + \tau^{p^2} &= 0, && (19a)\\ \tau\tau^p + \tau\tau^{p^2} + \tau^p\tau^{p^2} &= 0, && (19b)\\ \tau\tau^p\tau^{p^2} &= -d. && (19c)\end{aligned}$$

According to **A**, in this case we have

$$c_1^2 + c_p^2 - c_1c_p = 0. \qquad (20a)$$

As shown in **C**, this situation occurs only when $D = 0$, that is when $t = \pm\sqrt{-3} \in \mathbf{F}_p$. In addition, as previously described, there exist t's in $\mathbf{F}_p$ such that $t^2+3 = 0$ when $3 \mid (p-1)$. Therefore, when $3 \mid (p-1)$, we must restrict the parameter t such that $t^2 + 3 \neq 0$, in other words, we except $t = \pm\sqrt{-3}$.

Concludingly, we propose the following irreducible cubic trinomial generation algorithm;

<u>**Irreducible cubic trinomial generation**</u>

Input: An irreducible cubic trinomial $f(x)$ in the form of Eq.(7), where the coefficient a is not 0.

Output: Irreducible cubic trinomials $g(x)$ whose constant term d is only different from $f(x)$

Preparation: Calculate X and Y by Eq.(17).

Step1: Initialize the parameter $t \in \mathbf{F}_p$.

Step2: Calculate c_1 and c_p by Eq.(18).

Step3: Calculate d by Eq.(15a).

Step4: Increment t and then return to **Step2**.

(End of algorithm)

This algorithm requires an irreducible cubic trinomial $f(x)$ as the input; however, we can easily prepare $f(x)$ by using some irreducibility testing algorithm. We should note that this algorithm generates one irreducible cubic trinomial with two duplicates throughout the iterations, *see* **D**. The generator $f(x)$ itself is also contained of the generated irreducible cubic trinomials. In our previous work[4], we tested the irreducibility of a randomly generated cubic trinomial; however, we can use the proposed algorithm instead of the irreducibility test. In the above proposed algorithm, the parameter t is incremented by 1 in order to generate irreducible cubic trinomials iteratively; however, we can also iteratively generate irreducible cubic trinomials by randomly choosing the parameter t. In Sec.4.4, the difference of effectiveness between incrementing and randomly choosing is discussed.

4 Experimental results and comparison

For the experiment, the authors used NTL[9], PentiumIII (800MHz), and 39 prime numbers from 30 bits through 1024 bits as the characteristic p. Table 1 shows the experimental results, where the unit of the computation times in the table is ms.

In the column (A), the average computation time for generating an irreducible cubic trinomial over $\mathbf{F}_p$ by each algorithm is shown, in the column (B), the average computation times of a multiplication and an inversion in $\mathbf{F}_p$ are shown. For every prime number and algorithm, the authors generated 10000 irreducible cubic trinomials in order to measure the average computation time. As for the data in the column (B), the authors experimented with 1000000 multiplications and inversions with randomly chosen elements in $\mathbf{F}_p$.

Berlekamp-Massey algorithm generated irreducible cubic polynomials in which irreducible cubic trinomials will be included. Proposed(1) and (2) are based on the proposed algorithm; Proposed(1) generated irreducible cubic trinomials by incrementing the parameter t from 1 to 10000 and Proposed(2) generated irreducible cubic trinomials by randomly choosing the parameter t.

4.1 Arithmetic operations by NTL

Fig.1 shows the average computation times of a multiplication and an inversion in $\mathbf{F}_p$. As mentioned above, the authors used PentiumIII (800MHz) and NTL. From these graphs, we find that an inversion is more time-consuming as compared to a multiplication. In addition, as far as we can see from these graphs, it seems that the computation times of a multiplication and an inversion in $\mathbf{F}_p$ increase almost linearly with the size of the characteristic p. Such a behavior of NTL is quite important for the following considerations because Hiramoto et al. algorithm, Berlekamp-Massey algorithm, and the proposed algorithm require multiplications and inversions in $\mathbf{F}_p$.

In what follows, for the evaluation of the calculation cost, we do not consider additions and subtractions in $\mathbf{F}_p$ because these arithmetic operations are carried out much faster than a multiplication and an inversion in $\mathbf{F}_p$. Here, we note that the inversion of NTL is based on the extended Euclidean algorithm[10].

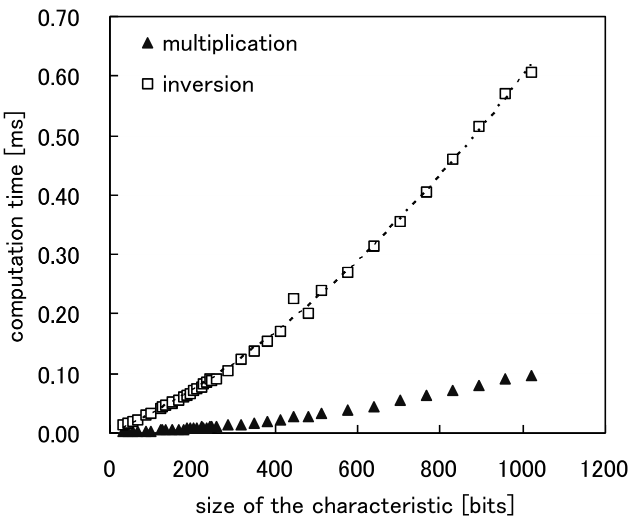

Figure 1: Average computation times of a multiplication and an inversion in $\mathbf{F}_p$

4.2 Hiramoto et al. algorithm

For Hiramoto et al. irreducibility testing algorithm[7], the authors inputted cubic trinomials in the form Eq.(8) with randomly choosing the constant term $d \in \mathbf{F}_p$. The first-degree coefficient a is fixed to a certain non-zero element in $\mathbf{F}_p$. Fig.2 shows the average computation time for generating an irreducible cubic trinomial. It is well known that $(p^3 - p)/3$ polynomials among p^3 distinct cubic polynomials over $\mathbf{F}_p$ are irreducible[5]. In other words, the ratio of irreducible cubic polynomials among cubic polynomials is about 1/3. Accordingly, the ratio of irreducible cubic trinomials among cubic trinomials is about 1/3[5]. Therefore, in order to generate an irreducible cubic trinomial, we need to try three cubic trinomials as the input of the irreducibility testing algorithm on average.

Hiramoto et al. algorithm requires $\mathcal{O}(\log_2 p)$ polynomial multiplications and modulo operations. In this experiment, the degree of the modular polynomial is 3, therefore the degrees of the multiplicand and multiplier polynomials of the polynomial multiplications are less than or equal to 2. Correspondingly, the degree of the objective of the polynomial modulo operation is less than or equal to 4. In addition, since the modular polynomial is monic as shown in Eq.(8), these polynomial operations can be implemented with several multiplications in $\mathbf{F}_p$ and this algorithm does not need any inversions.

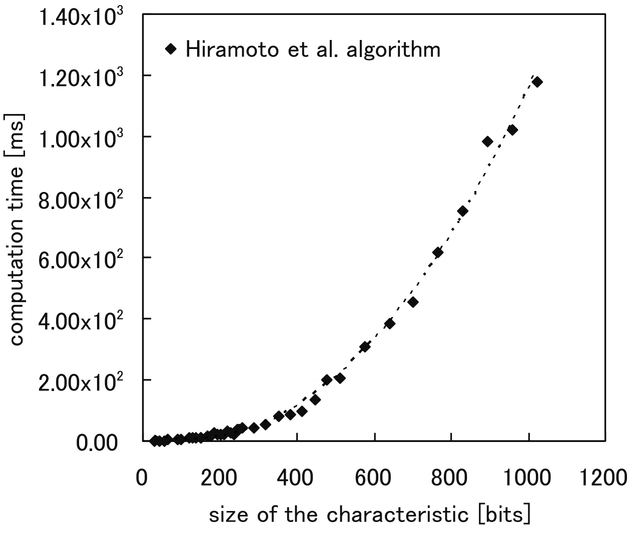

Figure 2: Average computation time for generating an irreducible cubic trinomial by Hiramoto et al. algorithm(HIT)

4.3 Berlekamp-Massey algorithm

Referring to [8], in order to generate irreducible cubic polynomials, Berlekamp-Massey algorithm, that is an algorithm to determine the minimal polynomial of a pseudo-random sequence, is applied as follows;

<u>**Berlekamp-Massey algorithm**</u>

Input: An irreducible cubic polynomial $f(x)$.

Output: Irreducible cubic polynomials.

Preparation: Let $f(x)$ be the generator, generate a pseudo-random sequence of elements of $\mathbf{F}_p$ as

$$s_0, s_1, s_2, \cdots, s_{n-2}, s_{n-1}, \tag{21}$$

where n is the length of the sequence.

Step1: $g_0(x) = 1, h_0(x) = x, m_0 = 0$ and $k = 2$.

Step2: Consider a polynomial $G(x)$ of degree 5;

$$G(x) = \sum_{i=0}^{5} s_{ik} x^i, \tag{22}$$

where each s_{ik} in Eq.(22) corresponds to the coefficient of the sequence Eq.(21). From $j = 0$ to 5, let b_j be the coefficient of x^j in $g_j(x)G(x)$, recursively compute the followings;

$$g_{j+1}(x) = g_j(x) - b_j h_j(x), \tag{23}$$

$$h_{j+1}(x) = \begin{cases} b_j^{-1} x g_j(x) & \text{if } b_j \neq 0 \text{ and } m_j \geq 0, \\ x h_j(x) & \text{otherwise,} \end{cases} \tag{24}$$

$$m_{j+1} = \begin{cases} -m_j & \text{if } b_j \neq 0 \text{ and } m_j \geq 0, \\ m_j + 1 & \text{otherwise.} \end{cases} \tag{25}$$

Step3: Output the $x^3 g_6(1/x)$.

Step4: $k = k + 1$, return to **Step2**.

(End of algorithm)

In the above algorithm, when $k = 1$, the generator $f(x)$ itself is generated. By incrementing the parameter k, the minimal polynomial of ω^k can be iteratively generated, where ω is a zero of $f(x)$. When we would like to generate 10000 irreducible cubic polynomials by the above algorithm, the length of the pseudo-random sequence, that is n as shown in Eq.(21), must be 50000. In this experiment, since the characteristic p is enough large, we can easily prepare such a long random sequence even if the generator $f(x)$ is not a primitive polynomial over $\mathbf{F}_p$, moreover, it is almost certain that the output $x^3 g_6(1/x)$ becomes an irreducible cubic polynomial for every k from 1 to 10000. In this experiment, there were no cases that the output was not an irreducible cubic polynomial.

The calculation cost of this algorithm becomes as follows; noting that the degree of $h_j(x)$ is less than or equal to 6, **Step2** requires a few dozens of multiplications and several inversions in $\mathbf{F}_p$. To be more detailed, since the output $x^3 g_6(1/x)$ becomes an irreducible cubic polynomial for every k as mentioned above, **Step2** constantly requires 5 inversions for $b_0^{-1}, b_1^{-1}, \cdots, b_4^{-1}$. The detailed number of multiplications depends on how to implement **Step2**. Fig.3 shows the graph of the average computation time for generating an irreducible cubic polynomial by this algorithm.

4.4 Proposed algorithm

From Eq.(18) and Eq.(15a), the proposed algorithm requires 10 multiplications and 1 inversion in $\mathbf{F}_p$ for generating an irreducible cubic trinomial over $\mathbf{F}_p$, where the authors considered that each $4t$ and $2t$ in Eq.(18) requires 1 multiplication such as $4 \times t$ and $2 \times t$.

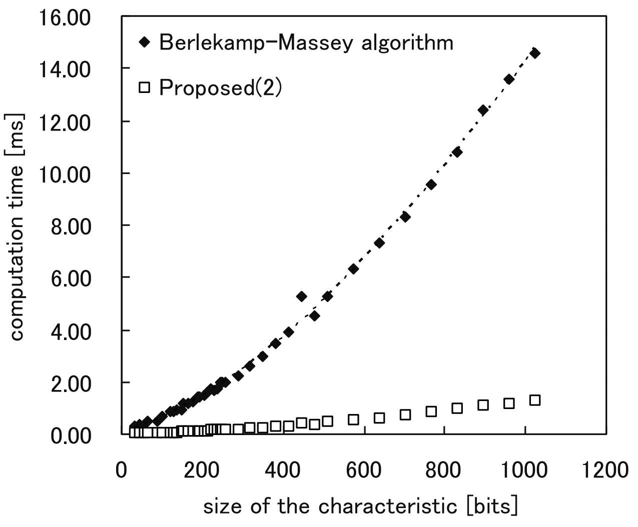

Figure 3: Average computation times for generating an irreducible cubic polynomial(trinomial) by Berlekamp-Massey(BM) algorithm and Proposed(2)

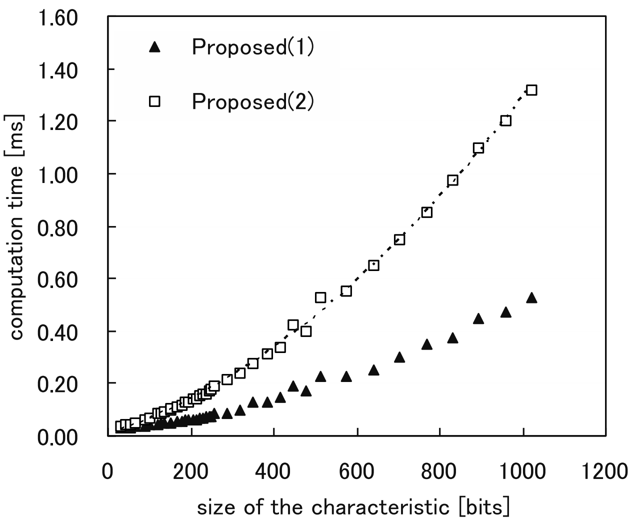

Figure 4: Average computation times for generating an irreducible cubic trinomial by Proposed(1) and Proposed(2)

Fig.4 shows the average computation times for generating an irreducible cubic trinomial by Proposed(1) and Proposed(2). As previously mentioned, the proposed algorithm can generate 10000 irreducible cubic trinomials by incrementing the parameter t from 1 to 10000 or randomly choosing the parameter t 10000 times. In the table and the graphs, the former and the latter cases are denoted by Proposed(1) and Proposed(2), respectively. We find that Proposed(1) is faster than Proposed(2). It is mainly due to that the inversion of NTL is based on the extended Euclidean algorithm. The calculation Eq.(18) needs the inverse of $t^2 + 3$. From $t = 1$ to 10000, $t^2 + 3$ increases from 4 to 100000003; however, these values are quite small as compared to the characteristic p used in this experiment. Therefore, the extended Euclidean algorithm can fast calculate these inverses.

4.5 Comparison

From the data, we can find that Berlekamp-Massey algorithm and the proposed algorithm can generate irreducible cubic polynomials quite faster than irreducibility testing algorithm. Though, they need a certain generator irreducible cubic polynomial. For example, when the characteristic p is $2^{257} - 93$, Berlekamp-Massey algorithm, Proposed(1), and Proposed(2) are about 22 times, 500 times, and 230 times faster than Hiramoto et al. irreducibility testing algorithm, respectively. Therefore, for the purpose of generating a lot of irreducible cubic polynomials, Berlekamp-Massey algorithm and the proposed algorithm are superior to such an irreducibility testing algorithm.

As previously mentioned, Berlekamp-Massey algorithm constantly needs 5 inversions and a few dozen of multiplications; however, the proposed algorithm needs only 1 inversion and 10 multiplications. Therefore, we can easily understand that the proposed algorithm is faster than Berlekamp-Massey algorithm. For example, when the characteristic p is $2^{257} - 93$, Proposed(1) and Proposed(2) are about 22 times and 10 times faster than Berlekamp-Massey algorithm, respectively. From the experimental results, we find that the proposed algorithm is the fastest among the three algorithms for generating irreducible cubic trinomials. In addition, since the proposed algorithm generates irreducible cubic trinomials in the form Eq.(8) whose constant term is only different from the generator Eq.(7), the proposed algorithm will be convenient for some prime order elliptic curve generation algorithms[5].

5 Conclusion

This paper has proposed an algorithm for generating irreducible cubic trinomials in the form

$$x^3 + ax + b,\ b \in \mathbf{F}_p, \tag{26}$$

where a is a certain fixed non-zero element in $\mathbf{F}_p$. This algorithm needs a certain irreducible cubic polynomial over $\mathbf{F}_p$ to be previously given as a generator; however, the proposed algorithm can generate irreducible cubic polynomials one after another by changing a certain parameter in $\mathbf{F}_p$. In this paper, we compared the calculation cost and the average computation time for generating an irreducible cubic polynomial, especially trinomial, among Hiramoto et al. irreducibility testing algorithm, Berlekamp-Massey minimal polynomial determining algorithm, and the proposed algorithm. When the characteristic p is $2^{257}-93$, the proposed algorithm is about 500 times and 22 times faster than Hiramoto et al. and Berlekamp-Massey algorithm, respectively, where the authors used PentiumIII (800MHz) and NTL. From the experimental results, it was shown that the proposed algorithm was the fastest among these three algorithms for generating irreducible cubic trinomials.

References

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

[2] 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.*, **J82-A**, no.8, pp.1269-1277, 1999.

[3] E.Konstantinou, Y.Stamatiou, and C.Zaroliagis, "On the construction of prime order elliptic curves," *Indocrypto2003*, LNCS 2904, pp.309-322, 2003.

[4] Y.Nogami and Y.Morikawa, "Fast Generation of Elliptic Curves with Prime Order over $\mathbf{F}_{p^{2^c}}$," *Proc. of Workshop on Coding and Cryptography (WCC2003)*, pp.347-3, 2003.

[5] Y.Nogami and Y.Morikawa, "A Classification of Irreducible Cubic Polynomials over Prime Field," *The 2004 International Technical Conference on Circuits/Systems, Computers and Communications*, CD-ROM, 2004.

[6] A.Menezes, *Handbook of Applied Cryptography*, CRC Press Inc., pp.154-159, 1997.

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

[8] R.Lidl and H.Niederreiter, *Finite Fields*, *Encyclopedia of Mathematics and Its Applications*, Cambridge University Press, pp.439-444, 1984.

[9] A Library for doing Number Theory, http://www.shoup.net/ntl/.

[10] D.Knuth, The Art of Computer Programming, vol.1 pp.13-14(extended Euclidean algorithm), Addison-Wesley, 1981.

A Proof of Eq.(15)

Substituting Eq.(10a) into Eq.(10b), Eq.(10c), we have

$$\begin{aligned} \omega^2+\omega\omega^p+\omega^{2p} &= -a, && \text{(A.1a)}\\ \omega^2\omega^p+\omega\omega^{2p} &= b. && \text{(A.1b)} \end{aligned}$$

Then, substituting Eq.(13) and Eq.(A.1a) into the left hand side of Eq.(14b), we have

$$\begin{aligned} &\tau\tau^p+\tau\tau^{p^2}+\tau^p\tau^{p^2}\\ =\ &\left(\omega^2+\omega\omega^p+\omega^{2p}\right)\left(c_1c_p-c_1^2-c_p^2\right)\\ =\ &-a\left(c_1c_p-c_1^2-c_p^2\right). \qquad \text{(A.2)} \end{aligned}$$

Therefore, we obtain Eq.(15b) by substituting Eq.(A.2) into Eq.(14b). On the other hand, substituting Eq.(13), Eq.(15b), Eq.(16), and Eq.(A.1b) into the left hand side of Eq.(14c), we have

$$\begin{aligned} \tau\tau^p\tau^{p^2} &= c_1^2c_p\left(3\omega^2\omega^p+\omega^3-\omega^{3p}\right)\\ &\quad +c_1c_p^2\left(3\omega\omega^{2p}-\omega^3+\omega^{3p}\right)\\ &\quad -\left(c_1^3+c_p^3\right)\left(\omega^2\omega^p+\omega\omega^{2p}\right)\\ &= c_1^2c_pX+c_1c_p^2Y-\left(c_1^3+c_p^3\right)b. \qquad \text{(A.3)} \end{aligned}$$

In addition, using Eq.(15b),

$$\begin{aligned} c_1^3+c_p^3 &= (c_1+c_p)(c_1^2+c_p^2-c_1c_p)\\ &= c_1+c_p \qquad \text{(A.4)} \end{aligned}$$

Therefore, we obtain Eq.(15a) by substituting Eq.(A.3) into Eq.(14c).

B Proof of Eq.(17)

Using Eq.(16), Eq.(A.1a), and Eq.(A.1b), we consider the sum and product of X and Y as

$$\begin{aligned} X+Y &= 3\left(\omega^2\omega^p+\omega\omega^{2p}\right)=3b, && \text{(A.5a)}\\ XY &= 9\left(\omega^2\omega^p+\omega\omega^{2p}\right)^2-\left(\omega^2+\omega\omega^p+\omega^{2p}\right)^3\\ &= 9b^2+a^3. && \text{(A.5b)} \end{aligned}$$

We find that X and Y are solutions of the following equation with regard to Z;

$$Z^2-3bZ+(a^3+9b^2)=0. \qquad \text{(A.6)}$$

Therefore, we obtain X and Y as Eq.(17a) with the discriminant Eq.(17b).

$$Z=\frac{3b\pm\sqrt{-(4a^3+27b^2)}}{2}. \qquad \text{(A.7)}$$

C Proof of Eq.(18)

We note that both c_1 and c_p are not 0 from $\tau \notin \mathbf{F}_p$. Let us suppose $c_1 \neq 0$, for example, the following relation is given from Eq.(15b) by multiplying c_1^{-2};

$$1+\left(\frac{c_p}{c_1}\right)^2-\left(\frac{c_p}{c_1}\right)=\left(\frac{1}{c_1}\right)^2. \qquad \text{(A.8)}$$

Substituting $C = c_p/c_1$ and $D = 1/c_1$ into Eq.(A.8) we have

$$D^2 = C^2 - C + 1. \qquad \text{(A.9)}$$

We can develop this relation as follows;

$$\begin{aligned} 3 &= 4D^2-(4C^2-4C+1) \\ &= (2D)^2-(2C-1)^2 \\ &= \{(2D)-(2C-1)\}\{(2D)+(2C-1)\} \qquad \text{(A.10)} \end{aligned}$$

By using the following parameter t;

$$t = 2D-(2C-1), \qquad \text{(A.11)}$$

from Eq.(A.10), we have

$$2D+(2C-1) = 3t^{-1}, \qquad \text{(A.12)}$$

where we should note that $t \in \mathbf{F}_p^*$ from Eq.(A.10) and Eq.(A.11). Then, we can solve C and D from simultaneous equations Eq.(A.11) and Eq.(A.12) as follows;

$$C=\frac{3+2t-t^2}{4t}, \quad D=\frac{t^2+3}{4t}. \qquad \text{(A.13)}$$

Finally, we get the representations for c_1 and c_p as

$$c_1=\frac{4t}{t^2+3}, \quad c_p=\frac{3+2t-t^2}{t^2+3}. \qquad \text{(A.14)}$$

D Two duplicates

As shown in Eqs.(13), an irreducible cubic polynomial has three conjugate zeros. This is the reason why the proposed algorithm generates two duplicates. Suppose that an irreducible cubic trinomial $g(x)$ is generated when the parameter $t = t_1$, its zero τ is given as

$$\tau = c_1\omega + c_p\omega^p = \frac{4t_1}{t_1^2+3}\omega + \frac{3+2t_1-t_1^2}{t_1^2+3}\omega^p, \qquad \text{(A.15a)}$$

and let us suppose that the other conjugates τ^p and τ^{p^2} are given with t_2 and t_3 as follows;

$$\begin{aligned} \tau^p &= \frac{4t_2}{t_2^2+3}\omega + \frac{3+2t_2-t_2^2}{t_2^2+3}\omega^p, \qquad \text{(A.15b)} \\ \tau^{p^2} &= \frac{4t_3}{t_3^2+3}\omega + \frac{3+2t_3-t_3^2}{t_3^2+3}\omega^p. \qquad \text{(A.15c)} \end{aligned}$$

If Eqs.(A.15) satisfy the conjugate relation Eqs.(19), then $g(x)$ is also generated in the both cases of the parameter $t = t_2$ and t_3. Therefore, we can consider

$$\begin{aligned} \frac{4t_2}{t_2^2+3} &= -\frac{3+2t_1-t_1^2}{t_1^2+3}, \qquad \text{(A.16a)} \\ \frac{3+2t_2-t_2^2}{t_2^2+3} &= \frac{4t_1}{t_1^2+3} - \frac{3+2t_1-t_1^2}{t_1^2+3}. \qquad \text{(A.16b)} \end{aligned}$$

$$\begin{aligned} \frac{4t_3}{t_3^2+3} &= \frac{3+2t_1-t_1^2}{t_1^2+3} - \frac{4t_1}{t_1^2+3}, \qquad \text{(A.17a)} \\ \frac{3+2t_3-t_3^2}{t_3^2+3} &= -\frac{4t_1}{t_1^2+3}. \qquad \text{(A.17b)} \end{aligned}$$

From Eqs.(A.16) and Eqs.(A.17), we have

$$t_2=\frac{t_1-3}{t_1+1}, \quad t_3=\frac{t_1+3}{1-t_1}, \qquad \text{(A.18)}$$

where t_1 must not be ± 1. When $t_1 = \pm 1$, we have

$$\tau = \omega + \omega^p, \quad \tau = -\omega, \qquad \text{(A.19)}$$

these are zeros of $f(-x)$.

Table 1: Computation times for generating an irreducible cubic trinomial over $\mathbf{F}_p$ and computation times for arithmetic operations in $\mathbf{F}_p$ by NTL (PentiumIII 800MHz)

unit : ms

	(A) Irreducible cubic trinomial (polynomial) generation				(B) Arithmetic operation	
p	HIT	BM	Proposed(1)	Proposed(2)	MUL	INV
$2^{31}-1$	1.04	2.96×10^{-1}	2.97×10^{-2}	3.59×10^{-2}	1.40×10^{-3}	1.25×10^{-2}
$2^{44}+7$	1.34	3.44×10^{-1}	3.12×10^{-2}	4.06×10^{-2}	1.87×10^{-3}	1.56×10^{-2}
$2^{55}+3$	1.84	3.75×10^{-1}	3.12×10^{-2}	4.53×10^{-2}	1.88×10^{-3}	1.97×10^{-2}
$2^{66}-5$	2.80	4.69×10^{-1}	3.44×10^{-2}	5.16×10^{-2}	2.50×10^{-3}	2.25×10^{-2}
$2^{89}-1$	4.09	5.15×10^{-1}	3.44×10^{-2}	5.93×10^{-2}	2.50×10^{-3}	2.94×10^{-2}
$2^{100}-15$	6.02	6.87×10^{-1}	4.06×10^{-2}	7.03×10^{-2}	3.43×10^{-3}	3.35×10^{-2}
$2^{122}-3$	9.01	8.43×10^{-1}	4.53×10^{-2}	8.44×10^{-2}	4.38×10^{-3}	4.10×10^{-2}
$2^{127}-1$	9.95	8.75×10^{-1}	4.69×10^{-2}	8.60×10^{-2}	4.37×10^{-3}	4.31×10^{-2}
$2^{137}-13$	1.14×10	9.38×10^{-1}	4.69×10^{-2}	9.22×10^{-2}	4.70×10^{-3}	4.63×10^{-2}
$2^{150}-3$	1.18×10	9.53×10^{-1}	4.84×10^{-2}	9.69×10^{-2}	4.53×10^{-3}	5.03×10^{-2}
$2^{152}-17$	1.27×10	1.16	5.00×10^{-2}	1.03×10^{-1}	5.64×10^{-3}	5.19×10^{-2}
$2^{166}-5$	1.57×10	1.19	5.32×10^{-2}	1.12×10^{-1}	5.62×10^{-3}	5.62×10^{-2}
$2^{178}+87$	1.54×10	1.22	5.30×10^{-2}	1.16×10^{-1}	5.62×10^{-3}	6.07×10^{-2}
$2^{187}-85$	2.53×10	1.44	6.24×10^{-2}	1.31×10^{-1}	7.50×10^{-3}	6.37×10^{-2}
$2^{194}+27$	1.90×10	1.44	5.94×10^{-2}	1.31×10^{-1}	7.80×10^{-3}	6.74×10^{-2}
$2^{204}+7$	1.97×10	1.50	6.28×10^{-2}	1.38×10^{-1}	7.52×10^{-3}	7.25×10^{-2}
$2^{212}+57$	2.27×10	1.62	6.24×10^{-2}	1.44×10^{-1}	8.76×10^{-3}	7.38×10^{-2}
$2^{221}-3$	3.35×10	1.75	6.88×10^{-2}	1.56×10^{-1}	1.00×10^{-2}	7.75×10^{-2}
$2^{228}+3$	2.58×10	1.69	6.88×10^{-2}	1.62×10^{-1}	9.36×10^{-3}	8.13×10^{-2}
$2^{236}+25$	2.44×10	1.75	7.48×10^{-2}	1.62×10^{-1}	8.76×10^{-3}	8.50×10^{-2}
$2^{244}+133$	3.58×10	2.00	7.52×10^{-2}	1.74×10^{-1}	9.92×10^{-3}	8.99×10^{-2}
$2^{248}+81$	4.03×10	2.00	7.52×10^{-2}	1.75×10^{-1}	1.12×10^{-2}	8.75×10^{-2}
$2^{257}-93$	4.40×10	2.00	8.80×10^{-2}	1.88×10^{-1}	1.13×10^{-2}	9.23×10^{-2}
$2^{287}+35$	4.45×10	2.25	8.72×10^{-2}	2.12×10^{-1}	1.26×10^{-2}	1.05×10^{-1}
$2^{319}+9$	5.69×10	2.62	1.00×10^{-1}	2.38×10^{-1}	1.50×10^{-2}	1.25×10^{-1}
$2^{351}+143$	8.17×10	3.00	1.26×10^{-1}	2.75×10^{-1}	1.62×10^{-2}	1.38×10^{-1}
$2^{383}+369$	8.66×10	3.50	1.26×10^{-1}	3.12×10^{-1}	2.00×10^{-2}	1.55×10^{-1}
$2^{415}+323$	1.00×10^{2}	3.88	1.50×10^{-1}	3.38×10^{-1}	2.24×10^{-2}	1.72×10^{-1}
$2^{447}+99$	1.33×10^{2}	5.26	1.87×10^{-1}	4.25×10^{-1}	2.87×10^{-2}	2.27×10^{-1}
$2^{479}+221$	2.02×10^{2}	4.53	1.74×10^{-1}	4.00×10^{-1}	2.75×10^{-2}	2.00×10^{-1}
$2^{511}+111$	2.06×10^{2}	5.29	2.24×10^{-1}	5.25×10^{-1}	3.25×10^{-2}	2.40×10^{-1}
$2^{575}+131$	3.07×10^{2}	6.31	2.24×10^{-1}	5.50×10^{-1}	3.76×10^{-2}	2.70×10^{-1}
$2^{639}+413$	3.84×10^{2}	7.31	2.51×10^{-1}	6.50×10^{-1}	4.50×10^{-2}	3.15×10^{-1}
$2^{703}+203$	4.54×10^{2}	8.31	2.99×10^{-1}	7.50×10^{-1}	5.49×10^{-2}	3.55×10^{-1}
$2^{767}+699$	6.19×10^{2}	9.56	3.50×10^{-1}	8.51×10^{-1}	6.24×10^{-2}	4.05×10^{-1}
$2^{831}+129$	7.56×10^{2}	1.08×10	3.74×10^{-1}	9.74×10^{-1}	7.26×10^{-2}	4.60×10^{-1}
$2^{895}+359$	9.80×10^{2}	1.24×10	4.50×10^{-1}	1.10	8.00×10^{-2}	5.15×10^{-1}
$2^{959}+233$	1.02×10^{3}	1.36×10	4.75×10^{-1}	1.20	8.99×10^{-2}	5.70×10^{-1}
$2^{1023}+1155$	1.18×10^{3}	1.46×10	5.26×10^{-1}	1.32	9.52×10^{-2}	6.05×10^{-1}

Remarks: HIT and BM mean Hiramoto et al. irreducibility testing algorithm and Berlekamp-Massey algorithm, respectively.

Proposed(1) denotes the case that the parameter t is incremented by 1 from 1 to 10000.

Proposed(2) denotes the case that the parameter t is randomly chosen.

Berlekamp-Massey algorithm(BM) generated irreducible cubic polynomials.

MUL and INV mean the multiplication and inversion in $\mathbf{F}_p$, respectively.