SCIS 2007 The 2007 Symposium on Cryptography and Information Security
Sasebo, Japan, Jan. 23-26, 2007
The Institute of Electronics, Information and Communication Engineers

Skew-Frobenius maps on hyperelliptic curves

Shunji Kozaki* Kazuto Matsuo*† Yasutomo Shimbara*

Abstract— The hyperelliptic curve cryptosystems take most of the time for computing a scalar multiplication kD of an element D in the Jacobian $\mathbb{J}_C$ of a hyperelliptic curve C for an integer k. Therefore its efficiency depends on the scalar multiplications. Among the fast scalar multiplication methods, there is a method using a Frobenius map. It uses a Jacobian defined over an extension field of the definition field of C, so that the Jacobian cannot be a 160 bit prime order. Therefore there is a loss of efficiency in that method. Iijima et al. proposed a method using a Frobeinus map on the quadratic twist of an elliptic curve, which is called a skew-Frobenius map in this paper. This paper shows constructions of the skew-Frobenius maps on hyperelliptic curves of genus 2 and 3.

Keywords: hyperelliptic curves, hyperelliptic curve cryptosystems, Frobenius maps, scalar multiplication, skew-Frobenius maps

1 Introduction

The hyperelliptic curve cryptosystems [11] have been studied as one of the public-key cryptosystems using a plain algebraic curve. They take most of the time for computing a scalar multiplication kD, where k is an integer and D is an element in the Jacobian $\mathbb{J}_C$ of a hyperelliptic curve C defined over $\mathbb{F}_q$. Therefore it requires a fast scalar multiplication to construct efficient hyperelliptic curve cryptosystems.

In order to achieve a fast scalar multiplication, one needs speeding up for two of operations i.e. additions of elements and addition chains. The sliding window method [3, Chapter IV.2] is one of the addition chain methods. Besides there is a method using a property of (hyper-)elliptic curves, i.e. the method using a Frobenius map [10, 12, 15].

The cryptosystems using a Frobenius map are constructed on $\mathbb{J}_C(\mathbb{F}_{q^n})$. However $\mathbb{J}_C(\mathbb{F}_{q^n})$ cannot be a prime order, so that one needs to choose C such that $\sharp\mathbb{J}_C(\mathbb{F}_{q^n})/\sharp\mathbb{J}_C(\mathbb{F}_q)$ is a prime number in such cryptosystems. Therefore there is a loss of efficiency in the method using a Frobenius map.

Iijima et al. proposed a method using a Frobeinus map on the quadratic twist of an elliptic curve [8]. We call this map the skew-Frobenius map in this paper. One can construct a prime order Jacobian $\mathbb{J}_{C_t}(\mathbb{F}_{q^n})$ by using the quadratic twist C_t over $\mathbb{F}_{q^n}$ of C if $n = 2^m$ for a natural number m [9]. For elliptic curves over finite fields of characteristic 2, Furihata et al.[4] and Furukawa et al.[5] proposed fast implementations of scalar multiplications using this method. This paper shows constructions of the skew-Frobenius maps on hyperelliptic curves of genus 2 and 3.

* Institute of Information Security, 2-14-1, Tsuruya-cho Kanagawa-ku, Yokohama 221-0835, Japan
† RID at Chuo Univ., 1-13-27, Kasuga Bunkyo-ku, Tokyo 112-8551, Japan

2 Preliminaries

This section briefly introduces the definitions and notations required in the following sections.

2.1 Hyperelliptic curves

Let C be a hyperelliptic curve of genus g defined over a finite field $\mathbb{F}_q$ which is of the form

$$\begin{aligned} C: \quad Y^2 &= F(X), \\ F(X) &= X^{2g+1} + a_{2g}X^{2g} + \cdots + a_0, \\ & \quad a_{2g}, \cdots, a_0 \in \mathbb{F}_q, \end{aligned} \tag{1}$$

where $\mathrm{char}(\mathbb{F}_q) \neq 2$. When $g = 1$, C is called an elliptic curve.

Let $\mathbb{J}_C$ be the Jacobian of C, and D is an element of $\mathbb{J}_C$. D can be represented as a finite formal sum of points $P_i = (x_i, y_i)$ on C as follows:

$$\begin{aligned} D &= \Sigma m_i P_i - (\Sigma m_i)\infty, \\ & \quad P_i, \infty \in C,\ m_i \in \mathbb{N}_{\geq 0},\ \Sigma m_i \leq g, \end{aligned} \tag{2}$$

where $P_i \neq (x_i, -y_i)$ for $i \neq j$. Moreover, D can be also represented by a couple of polynomials $u(X)$ and $v(X)$. The polynomials $u(X)$ and $v(X)$ satisfy

$$\begin{aligned} u(X) &:= \Pi_{i=1}^{g}(X - x_i) \\ &= X^{2g} + u_{2g-1}X^{2g-1} + \cdots + u_0, \quad (3) \\ v(x_i) &= y_i, \quad (4) \\ & \quad \deg v(X) < \deg u(X) \leq g, \\ & \quad u(X) | F(X) - v^2(X), \\ & \text{for} \quad P_i = (x_i, y_i) \in C,\ 0 \leq i \leq g \end{aligned}$$

[13]. We denote D as (u, v) by using the polynomials. The set of $D = (u, v)$ such that $u, v \in K[X]$ forms a subgroup $\mathbb{J}_C(K)$ of $\mathbb{J}_C$. When $g = 1$, $\mathbb{J}_C(\overline{\mathbb{F}_q}) \cong C(\overline{\mathbb{F}_q})$.

2.2 Scalar multiplications

Let k be an integer that satisfies

$$k \approx q^{gn} \geq 2^{160}.$$

The hyperelliptic curve cryptosystems take most of the time for computing a scalar multiplication kD for such k and $D \in \mathbb{J}_C(\mathbb{F}_{q^n})$. Therefore, speeding up a scalar multiplication induces speeding up hyperelliptic curve cryptosystems.

2.3 Frobenius maps

The q-th power Frobenius map ϕ on C is defined as

$$\begin{array}{rcl} \phi : C(\overline{\mathbb{F}}_q) & \longrightarrow & C(\overline{\mathbb{F}}_q) \\ (x, y) & \longmapsto & (x^q, y^q) \\ \mathcal{O} & \longmapsto & \mathcal{O}. \end{array}$$

Moreover, ϕ is an automorphism over $\mathbb{J}_C(\overline{\mathbb{F}_q})$, and its characteristic polynomial $\chi(X) \in \mathbb{Z}[X]$ is of the form

$$\begin{array}{rcl} \chi(X) & = & X^{2g} - s_1 X^{2g-1} + s_2 X^{2g-2} - \\ & & \cdots - s_1 q^{g-1} X + q^g, \\ & & s_1, \cdots, s_g \in \mathbb{Z}. \end{array}$$

Because

$$\chi(\phi) = 0,$$

the integer k can be expanded as

$$k = \Sigma_{i=0}^{s} r_i \phi^i,$$

where $r_i \in \{-\lceil q^g/2 \rceil + 1, \cdots, \lfloor q^g/2 \rfloor\}$, by using ϕ. Therefore, the scalar multiplication kD can be computed as

$$kD = \Sigma_{i=0}^{s} r_i \phi^i(D). \tag{5}$$

To apply the Frobenius map to the scalar multiplication, the cryptosystems should be constructed on $\mathbb{J}_C(\mathbb{F}_{q^n})/\mathbb{J}_C(\mathbb{F}_q)$, because the map is trivial on $\mathbb{J}_C(\mathbb{F}_q)$. However $\sharp\mathbb{J}_C(\mathbb{F}_{q^n}) \approx q^{gn}$ cannot be a prime number, so that one needs to choose C such that $\sharp\mathbb{J}_C(\mathbb{F}_{q^n})/\sharp\mathbb{J}_C(\mathbb{F}_q) \approx q^{g(n-1)}$ is a prime number in such cryptosystems. Therefore there is a loss of efficiency in the method using a Frobenius map.

3 Skew-Frobenius maps over elliptic curves

In order to construct more efficient cryptosystems using a Frobenius map, Iijima et al. proposed a method using a Frobeinus map on the quadratic twist of an elliptic curve [8].

This section shows the construction of skew-Frobenius maps on elliptic curves according to [8].

Let $c \in \mathbb{F}_{q^n}$ be a quadratic non-residue over $\mathbb{F}_{q^n}$. The quadratic twist C_t of an elliptic curve C of the form in (1) is defined as

$$C_t : Y^2 = X^3 + ca_2X^2 + c^2a_1X + c^3a_0.$$

It is known that

$$C_t(\mathbb{F}_{q^{2n}}) \cong C(\mathbb{F}_{q^{2n}}), \tag{6}$$

whereas

$$C_t(\mathbb{F}_{q^n}) \ncong C(\mathbb{F}_{q^n}).$$

Moreover,

$$End(C_t) \cong End(C). \tag{7}$$

Let σ_0 be the isomorphism of (6). The isomorphism

$$\begin{array}{rcl} \phi_t : C_t(\mathbb{F}_{q^{2n}}) \xrightarrow[\sigma_0^{-1}]{\sim} & C(\mathbb{F}_{q^{2n}}) & \xrightarrow[\phi]{\sim} \\ & C(\mathbb{F}_{q^{2n}}) & \xrightarrow[\sigma_0]{\sim} C_t(\mathbb{F}_{q^{2n}}) \end{array} \tag{8}$$

can be constructed by using such σ_0 and a q-th power Frobenius map ϕ over $C(\mathbb{F}_{q^{2n}})$. The isomorphism $\sigma_0 : C(\mathbb{F}_{q^{2n}}) \longrightarrow C_t(\mathbb{F}_{q^{2n}})$ is given as

$$\begin{array}{rcl} \sigma_0 : C(\mathbb{F}_{q^{2n}}) & \longrightarrow & C_t(\mathbb{F}_{q^{2n}}) \\ (x, y) & \longmapsto & (x_t, y_t) \\ (x, y) & \longmapsto & (cx, c^{3/2}y). \end{array}$$

Therefore the inversion σ_0^{-1} is given as

$$\begin{array}{rcl} \sigma_0^{-1} : C_t(\mathbb{F}_{q^{2n}}) & \longrightarrow & C(\mathbb{F}_{q^{2n}}) \\ (x_t, y_t) & \longmapsto & (x, y) \\ (x_t, y_t) & \longmapsto & (c^{-1}x_t, c^{-3/2}y_t). \end{array}$$

Consequently, the isomorphism ϕ_t is obtained as

$$\begin{array}{rcl} \phi_t : C_t(\mathbb{F}_{q^{2n}}) & \longrightarrow & C_t(\mathbb{F}_{q^{2n}}) \\ (x, y) & \longmapsto & (c^{1-q}x^q, c^{3(1-q)/2}y^q) \end{array}$$

from (8). For the elements in $C(\mathbb{F}_{q^n})$, ϕ_t is also an automorphism.

Moreover, the map ϕ_t has the same characteristic polynomial as ϕ from (7). Therefore ϕ_t can be used for a scalar multiplication as

$$kD = \Sigma_{i=0}^{s} r_i \phi^i(D)$$

by the same expansion of k in (5).

Furthermore, if $n = 2^m$ for $m \in \mathbb{N}$, one can choose C so that $C_t(\mathbb{F}_{q^n})$ is a prime order [8, 9].

The computation of ϕ_t needs 2 multiplications and 2 q-th power operations over $\mathbb{F}_{q^n}$. However, if $C_t(\mathbb{F}_{q^n})$ has a prime order, the size of $\mathbb{F}_{q^n}$ becomes smaller than the size of the definition field used by a naive method using a Frobenius map, so that we expect the scalar multiplication to be fast by using the skew-Frobenius map.

For elliptic curves over finite fields of characteristic 2, Furihata et al.[4] and Furukawa et al.[5] proposed fast implementations of scalar multiplications using the skew-Frobenius maps.

4 Skew-Frobenius maps on hyperelliptic curves

This section shows a construction of the skew-Frobenius maps on hyperelliptic curves of genus 2 and 3.

4.1 On hyperelliptic curves of $g = 2$

Let $c \in \mathbb{F}_{q^n}$ be a quadratic non-residue over $\mathbb{F}_{q^n}$. A quadratic twist C_t of an elliptic curve C of the form in (1) for $g = 2$ over $\mathbb{F}_{q^n}$ is defined as

$$\begin{aligned} C_t : Y^2 &= F_t(X), \\ F_t(X) &= X^5 + ca_4X^4 + c^2a_3X^3 \\ &\quad + c^3a_2X^2 + c^4a_1X + c^5a_0. \end{aligned}$$

Let $\mathbb{J}_{C_t}$ be the Jacobian of C_t, and let D_t be an element of $\mathbb{J}_{C_t}$. We denote $D_t = (u_t, v_t)$ by using the polynomials from (3) , (4).

An isomorphism of the Jacobians

$$\begin{aligned} \sigma : \mathbb{J}_C(\mathbb{F}_{q^{2n}}) &\longrightarrow \mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}}) \\ D &\longmapsto D_t \\ (u, v) &\longmapsto (u_t, v_t) \end{aligned}$$

can be obtained by applying the isomorphism

$$\begin{aligned} \sigma_0 : C(\mathbb{F}_{q^{2n}}) &\longrightarrow C_t(\mathbb{F}_{q^{2n}}) \\ (x_i, y_i) &\longmapsto (x_{ti}, y_{ti}), \end{aligned}$$

to each $P_i = (x_i, y_i)$ in (2). The isomorphism σ_0 is obtained as

$$\sigma_0 : (x, y) \longmapsto (cx, c^{5/2}y). \tag{9}$$

ϕ_t defined as (8) can be constructed by using σ and the q-th power Frobenius map ϕ over $\mathbb{J}_C(\mathbb{F}_{q^{2n}})$.

Now, we classify the elements in $\mathbb{J}_C(\mathbb{F}_{q^{2n}})$ into the following types,

$$\begin{aligned} \text{Type I} &: D = \{P\} \\ \text{Type II} &: D = \{P_1, P_2\}, P_1 \neq P_2 \\ \text{Type III} &: D = \{P, P\}, \end{aligned}$$

where $P = (x, y), P_1 = (x_1, y_1), P_2 = (x_2, y_2)$.

The following shows a construction of ϕ_t for each of the types.

Type I: The polynomials $u(X)$, $v(X)$, $u_t(X)$ and $v_t(X)$ are obtained as

$$\begin{aligned} u(X) &= X - x, & v(X) &= y \\ u_t(X) &= X - cx, & v_t(X) &= c^{5/2}y \end{aligned}$$

from (3) , (4) , (9).

Therefore, the isomorphism ϕ_t over $\mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}})$ is obtained as

$$\begin{aligned} \phi_t : \mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}}) &\longrightarrow \mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}}) \\ (u_0, v_0) &\longmapsto (c^{1-q}u_0^q, c^{5(1-q)/2}v_0^q) \end{aligned}$$

from (8).

Type II: The polynomial $u(X)$ is obtained as

$$u(X) = X^2 - (x_1 + x_2)X + x_1x_2 \tag{10}$$

from (3). The polynomial $v(X)$ is obtained as

$$\begin{aligned} v(X) &= ((y_1 - y_2)/(x_1 - x_2))X \\ &\quad + (x_1y_2 - x_2y_1)/(x_1 - x_2) \end{aligned} \tag{11}$$

by the Lagrange interpolation from (4).

The polynomials $u_t(X)$ and $v_t(X)$ are obtained as

$$\begin{aligned} u_t(X) &= X^2 - c(x_1 + x_2)X + c^2x_1x_2, & (12) \\ v_t(X) &= c^{3/2}(y_1 - y_2)/(x_1 - x_2)X \\ &\quad + c^{5/2}(x_1y_2 - x_2y_1)/(x_1 - x_2) & (13) \end{aligned}$$

from (3) , (4) , (9). The isomorphism σ is obtained as

$$\begin{aligned} \sigma : \mathbb{J}_C(\mathbb{F}_{q^{2n}}) &\longrightarrow \mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}}) \\ (u_1, u_0, v_1, v_0) &\longmapsto (cu_1, c^2u_0, c^{3/2}v_1, c^{5/2}v_0). \end{aligned}$$

from (10) , (11) , (12) , (13).

Therefore, the isomorphism ϕ_t over $\mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}})$ is obtained as

$$\begin{aligned} \phi_t : \mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}}) &\longrightarrow \mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}}) \\ (u_1, u_0, v_1, v_0) &\longmapsto (c^{1-q}u_1^q, c^{2(1-q)}u_0^q, \\ &\qquad c^{3(1-q)/2}v_1^q, c^{5(1-q)/2}v_0^q) \end{aligned}$$

from (8).

Type III: The polynomial $u(X)$ is obtained as

$$u(X) = X^2 - 2xX + x^2 \tag{14}$$

from (3). The polynomial $v(X)$ is obtained as

$$v(X) = (F'(x)/2y)X - (F'(x)/2y)x + y \tag{15}$$

by the Newton iteration from (4) , where F' denotes the derivative of F in (1).

The polynomials $u_t(X)$ and $v_t(X)$ are obtained as

$$\begin{aligned} u_t(X) &= X^2 - 2cxX + c^2x^2 & (16) \\ v_t(X) &= c^{-5/2}(F_t'(cx)/2y)X \\ &\quad - c^{-3/2}(F_t'(cx)/2y)x + c^{5/2}y & (17) \end{aligned}$$

from (3) , (4) , (9). The isomorphism σ is obtained as

$$\begin{aligned} \sigma : \mathbb{J}_C(\mathbb{F}_{q^{2n}}) &\longrightarrow \mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}}) \\ (u_1, u_0, v_1, v_0) &\longmapsto (cu_1, c^2u_0, c^{3/2}v_1, c^{5/2}v_0) \end{aligned}$$

from (14) , (15) , (16) , (17) , by $F_t'(cx) = c^4F'(x)$.

Therefore, the isomorphism ϕ_t over $\mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}})$ is obtained as

$$\begin{aligned} \phi_t : \mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}}) &\longrightarrow \mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}}) \\ (u_1, u_0, v_1, v_0) &\longmapsto (c^{1-q}u_1^q, c^{2(1-q)}u_0^q, \\ &\qquad c^{3(1-q)/2}v_1^q, c^{5(1-q)/2}v_0^q) \end{aligned}$$

from (8).

Consequently, we have the isomorphism ϕ_t over $\mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}})$ of a hyperelliptic curve of genus 2 as
Type I :

$$\begin{aligned}\phi_t : \mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}}) &\longrightarrow \mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}})\\ (u_0, v_0) &\longmapsto (c^{1-q}u_0^q, c^{5(1-q)/2}v_0^q)\end{aligned}$$

Type II , III :

$$\begin{aligned}\phi_t : \mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}}) &\longrightarrow \mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}})\\ (u_1, u_0, v_1, v_0) &\longmapsto (c^{1-q}u_1^q, c^{2(1-q)}u_0^q,\\ &\qquad c^{3(1-q)/2}v_1^q, c^{5(1-q)/2}v_0^q).\end{aligned}$$

ϕ_t is a non-trivial isomorphism even for the elements in $\mathbb{J}_{C_t}(\mathbb{F}_{q^n})$.

The isomorphism ϕ_t on the curves of genus 2 needs at most 4 multiplications and at most 4 q-th power operations over $\mathbb{F}_{q^n}$. However, if $\mathbb{J}_{C_t}(\mathbb{F}_{q^n})$ has a prime order, the size of $\mathbb{F}_{q^n}$ becomes smaller than the size of the definition field used by a naive method using Frobenius map, so that we expect the scalar multiplication to be fast by using the skew-Frobenius map on a curve of genus 2.

4.2 On hyperelliptic curves of $g = 3$

It is more important than genus 2 to construct skew-Frobenius maps on hyperelliptic curves of genus 3, because fast implementations [7, 14] of addition algorithms on a hyperelliptic curve of genus 3 is known, moreover, there exist efficient attacks against hyperelliptic curve cryptosystems of genus 2 if $n = 2^m$ [1, 2, 6].

This section shows the construction of skew-Frobenius maps on hyperelliptic curves of genus 3.

Let $c \in \mathbb{F}_{q^n}$ be a quadratic non-residue over $\mathbb{F}_{q^n}$. A quadratic twist C_t of an elliptic curve C of the form in (1) for $g = 3$ over $\mathbb{F}_{q^n}$ is defined as

$$\begin{aligned}C_t : Y^2 =& F_t(X),\\ F_t(X) =& X^7 + a_6cX^6 + a_5c^2X^5 + a_4c^3X^4\\ &+ a_3c^4X^3 + a_2c^5X^2 + a_1c^6X + a_0c^7.\end{aligned}$$

Let $\mathbb{J}_{C_t}$ be the Jacobian of C_t, and let D_t be an element of $\mathbb{J}_{C_t}$. Now we denote $D_t = (u_t, v_t)$ by using the polynomials from (3) , (4). An isomorphism over the Jacobians

$$\begin{aligned}\sigma : \mathbb{J}_C(\mathbb{F}_{q^{2n}}) &\longrightarrow \mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}})\\ D &\longmapsto D_t\\ (u, v) &\longmapsto (u_t, v_t),\end{aligned}$$

can be obtained by applying the isomorphism

$$\begin{aligned}\sigma_0 : C &\longrightarrow C_t\\ (x, y) &\longmapsto (cx, c^{7/2}y)\end{aligned} \tag{18}$$

to each point on C The isomorphism ϕ_t is obtained in the similar manner for genus 2 in the section 4.1.

Now, we classify the elements in $\mathbb{J}_C(\mathbb{F}_{q^{2n}})$ into the following types,

$$\begin{aligned}\text{Type I} : D &= \{P\}\\ \text{Type II} : D &= \{P_1, P_2\}, P_1 \neq P_2\\ \text{Type III} : D &= \{P, P\}\\ \text{Type IV} : D &= \{P_1, P_2, P_3\}, P_1 \neq P_2 \neq P_3 \neq P_1\\ \text{Type V} : D &= \{P_1, P_1, P_2\}, P_1 \neq P_2\\ \text{Type VI} : D &= \{P, P, P\}\end{aligned}$$

where $P = (x, y), P_1 = (x_1, y_1), P_2 = (x_2, y_2), P_3 = (x_3, y_3)$.

The following shows a construction of ϕ_t for each of the types.

Type I, II, III: For these types, the isomorphism ϕ_t over $\mathbb{J}_{C_t}(\mathbb{F}_{q^n})$ is obtained by the similar manner for 2 as follows:
Type I :

$$\begin{aligned}\phi_t : \mathbb{J}_{C_t}(\mathbb{F}_{q^n}) &\longrightarrow \mathbb{J}_{C_t}(\mathbb{F}_{q^n})\\ (u_0, v_0) &\longmapsto (c^{1-q}u_0^q, c^{7(1-q)/2}v_0^q)\end{aligned}$$

Type II , III :

$$\begin{aligned}\phi_t : \mathbb{J}_{C_t}(\mathbb{F}_{q^n}) &\longrightarrow \mathbb{J}_{C_t}(\mathbb{F}_{q^n})\\ (u_1, u_0, v_1, v_0) &\longmapsto (c^{1-q}u_1^q, c^{2(1-q)}u_0^q,\\ &\qquad c^{5(1-q)/2}v_1^q, c^{7(1-q)/2}v_0^q).\end{aligned}$$

Type IV: The polynomial $u(X)$ is obtained as

$$\begin{aligned}u(X) =& X^3 - (x_1 + x_2 + x_3)X^2\\ &+(x_1x_2 + x_2x_3 + x_3x_1)X - x_1x_2x_3\end{aligned} \tag{19}$$

from (3). The polynomial $v(X)$ is obtained as

$$\begin{aligned}v(X) =& X^2(y_1(x_3 - x_2) + y_2(x_1 - x_3) + y_3(x_2 - x_1))\\ &/((x_1 - x_2)(x_2 - x_3)(x_3 - x_1))\\ &-X(y_1(x_3^2 - x_2^2) + y_2(x_1^2 - x_3^2) + y_3(x_2^2 - x_1^2))\\ &/((x_1 - x_2)(x_2 - x_3)(x_3 - x_1))\\ &+(y_1x_2x_3(x_3 - x_2) + y_2x_1x_3(x_1 - x_3)\\ &+y_3x_1x_2(x_2 - x_1))/((x_1 - x_2)(x_2 - x_3)(x_3 - x_1))\end{aligned} \tag{20}$$

by the Lagrange interpolation from (4).

The polynomials $u_t(X)$ and $v_t(X)$ are obtained as

$$\begin{aligned}u_t(X) =& X^3 - c(x_1 + x_2 + x_3)X^2\\ &+c^2(x_1x_2 + x_2x_3 + x_3x_1)X - c^3x_1x_2x_3\end{aligned} \tag{21}$$

$$\begin{aligned}v_t(X) =& X^2c^{3/2}(y_1(x_3 - x_2) + y_2(x_1 - x_3) + y_3(x_2 - x_1))\\ &/((x_1 - x_2)(x_2 - x_3)(x_3 - x_1))\\ &-Xc^{5/2}(y_1(x_3^2 - x_2^2) + y_2(x_1^2 - x_3^2) + y_3(x_2^2 - x_1^2))\\ &/((x_1 - x_2)(x_2 - x_3)(x_3 - x_1))\\ &+c^{7/2}(y_1x_2x_3(x_3 - x_2) + y_2x_1x_3(x_1 - x_3)\\ &+y_3x_1x_2(x_2 - x_1))/((x_1 - x_2)(x_2 - x_3)(x_3 - x_1))\end{aligned} \tag{22}$$

from (3) , (4) , (18). The isomorphism σ is obtained as

$$\begin{aligned}\sigma : \mathbb{J}_C(\mathbb{F}_{q^{2n}}) &\longrightarrow \mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}})\\ (u_2, u_1, u_0, &\longmapsto (cu_2, c^2u_1, c^3u_0,\\ v_2, v_1, v_0) &\quad c^{3/2}v_2, c^{5/2}v_1, c^{7/2}v_0).\end{aligned}$$

from (19) , (20) , (21) , (22).

Therefore, the isomorphism ϕ_t over $\mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}})$ is obtained as

$$\begin{aligned}\phi_t : \mathbb{J}_{C_t}(\mathbb{F}_{q^n}) &\longrightarrow \mathbb{J}_{C_t}(\mathbb{F}_{q^n})\\ (u_2, u_1, u_0, &\longmapsto (c^{1-q}u_2^q, c^{2(1-q)}u_1^q, c^{3(1-q)}u_0^q,\\ v_2, v_1, v_0) &\quad c^{3(1-q)/2}v_2^q, c^{5(1-q)/2}v_1^q,\\ &\quad c^{7(1-q)/2}v_0^q)\end{aligned}$$

from (8).

Type V: The polynomial $u(X)$ is obtained as

$$u(X) = X^3 - (2x_1 + x_2)X^2 + (x_1^2 + 2x_1x_2)X - x_1^2x_2 \quad (23)$$

from (3). The polynomials $v(X)$ is obtained as

$$\begin{aligned}&= ((y_2 - y_1)/(x_1 - x_2)^2)X^2\\ &\quad +(2x_1(y_1 - y_2)/(x_1 - x_2)^2)X\\ &\quad +(x_1^2y_2 - (2x_1 - x_2)x_2y_1)/(x_1 - x_2)^2\end{aligned} \quad (24)$$

by the Chinese remainder algorithm from (4).

The polynomials $u_t(X)$ and $v_t(X)$ are obtained as

$$\begin{aligned}u_t(X) &= X^3 - c(2x_1 + x_2)X^2\\ &\quad +c^2(x_1^2 + 2x_1x_2)X - c^3x_1^2x_2 \quad (25)\\ v_t(X) &= c^{3/2}((y_2 - y_1)/(x_1 - x_2)^2)X^2\\ &\quad +c^{5/2}(2x_1(y_1 - y_2)/(x_1 - x_2)^2)X\\ &\quad +c^{7/2}(x_1^2y_2 - (2x_1 - x_2)x_2y_1)/(x_1 - x_2)^2 \quad (26)\end{aligned}$$

from (3) , (4) , (18). The isomorphism σ is obtained as

$$\begin{aligned}\sigma : \mathbb{J}_C(\mathbb{F}_{q^{2n}}) &\longrightarrow \mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}})\\ (u_2, u_1, u_0, &\longmapsto (cu_2, c^2u_1, c^3u_0,\\ v_2, v_1, v_0) &\quad c^{3/2}v_2, c^{5/2}v_1, c^{7/2}v_0).\end{aligned}$$

from (23) , (24) , (25) , (26).

Therefore, the isomorphism ϕ_t over $\mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}})$ is obtained as

$$\begin{aligned}\phi_t : \mathbb{J}_{C_t}(\mathbb{F}_{q^n}) &\longrightarrow \mathbb{J}_{C_t}(\mathbb{F}_{q^n})\\ (u_2, u_1, u_0, &\longmapsto (c^{1-q}u_2^q, c^{2(1-q)}u_1^q, c^{3(1-q)}u_0^q,\\ v_2, v_1, v_0) &\quad c^{3(1-q)/2}v_2^q, c^{5(1-q)/2}v_1^q,\\ &\quad c^{7(1-q)/2}v_0^q)\end{aligned}$$

from (8).

Type VI: The polynomial $u(X)$ is obtained as

$$u(X) = X^3 - 3xX^2 + 3x^2X - x^3 \quad (27)$$

from 3. The polynomial $v(X)$ is obtained as

$$\begin{aligned}v(X) &= X^2((F''(x)/4y) - (F'^2(x)/8y^3))\\ &\quad +X((F'(x)/2y)\\ &\quad -2x((F''(x)/4y) - (F'^2(x)/8y^3)))\\ &\quad +y - (xF'(x)/2y)\\ &\quad +x^2((F''(x)/4y) - (F'^2(x)/8y^3))\end{aligned} \quad (28)$$

by the Newton iteration from (4).

The polynomials $u_t(X)$ and $v_t(X)$ are obtained as

$$\begin{aligned}u_t(X) &= X^3 - 3cxX^2 + 3c^2x^2X - c^3x^3 \quad (29)\\ v_t(X) &= X^2((F_t''(cx)/4c^{7/2}y) - (F_t'^2(cx)/8c^{21/2}y^3))\\ &\quad +X((F_t'(cx)/2c^{7/2}y)\\ &\quad -2x((F_t''(cx)/4c^{7/2}y) - (F_t'^2(cx)/8c^{21/2}y^3)))\\ &\quad +c^{7/2}y - (cxF_t'(cx)/2c^{7/2}y)\\ &\quad +c^2x^2((F_t''(cx)/4c^{7/2}y) - (F_t'^2(cx)/8c^{21/2}y^3))\end{aligned} \quad (30)$$

from (3) , (4) , (18). The isomorphism σ is obtained as

$$\begin{aligned}\sigma : \mathbb{J}_C(\mathbb{F}_{q^{2n}}) &\longrightarrow \mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}})\\ (u_2, u_1, u_0, &\longmapsto (cu_2, c^2u_1, c^3u_0,\\ v_2, v_1, v_0) &\quad c^{3/2}v_2, c^{5/2}v_1, c^{7/2}v_0)\end{aligned}$$

from (27) , (28) , (29) , (30), where $F_t'(cx) = c^6F'(x), F_t''(cx) = c^5F''(x)$.

Therefore, the isomorphism ϕ_t over $\mathbb{J}_{C_t}(\mathbb{F}_{q^{2n}})$ is obtained as

$$\begin{aligned}\phi_t : \mathbb{J}_{C_t}(\mathbb{F}_{q^n}) &\longrightarrow \mathbb{J}_{C_t}(\mathbb{F}_{q^n})\\ (u_2, u_1, u_0, &\longmapsto (c^{1-q}u_2^q, c^{2(1-q)}u_1^q, c^{3(1-q)}u_0^q,\\ v_2, v_1, v_0) &\quad c^{3(1-q)/2}v_2^q, c^{5(1-q)/2}v_1^q,\\ &\quad c^{7(1-q)/2}v_0^q)\end{aligned}$$

from (8).

Consequently, we have the isomorphism ϕ_t over $\mathbb{J}_{C_t}(\mathbb{F}_{q^n})$ of a hyperelliptic curve of genus 3 as

Type I :

$$\begin{aligned}\phi_t : \mathbb{J}_{C_t}(\mathbb{F}_{q^n}) &\longrightarrow \mathbb{J}_{C_t}(\mathbb{F}_{q^n})\\ (u_0, v_0) &\longmapsto (c^{1-q}u_0^q, c^{7(1-q)/2}v_0^q)\end{aligned}$$

Type II , III :

$$\begin{aligned}\phi_t : \mathbb{J}_{C_t}(\mathbb{F}_{q^n}) &\longrightarrow \mathbb{J}_{C_t}(\mathbb{F}_{q^n})\\ (u_1, u_0, v_1, v_0) &\longmapsto (c^{1-q}u_1^q, c^{2(1-q)}u_0^q,\\ &\quad c^{5(1-q)/2}v_1^q, c^{7(1-q)/2}v_0^q)\end{aligned}$$

Type IV , V , VI:

$$\begin{aligned}\phi_t : \mathbb{J}_{C_t}(\mathbb{F}_{q^n}) &\longrightarrow \mathbb{J}_{C_t}(\mathbb{F}_{q^n})\\ (u_2, u_1, u_0, &\longmapsto (c^{1-q}u_2^q, c^{2(1-q)}u_1^q, c^{3(1-q)}u_0^q,\\ v_2, v_1, v_0) &\quad c^{3(1-q)/2}v_2^q, c^{5(1-q)/2}v_1^q,\\ &\quad c^{7(1-q)/2}v_0^q)\end{aligned}$$

ϕ_t is a non-trivial isomorphism even for the elements in $\mathbb{J}_{C_t}(\mathbb{F}_{q^n})$.

The isomorphism ϕ_t on the curves of genus 3 needs at most 6 multiplications and 6 q-th power operations over $\mathbb{F}_{q^n}$. However, if $\mathbb{J}_{C_t}(\mathbb{F}_{q^n})$ has a prime order, the size of $\mathbb{F}_{q^n}$ becomes, small so that we expect the scalar multiplication to be fast by using the skew-Frobenius map on a curve of genus 3.

Acknowledgements

This research was partially supported by "The Research on Security and Reliability in Electronic Society", Chuo University 21st Century COE Program.

References

[1] S. Arita. A Weil descent attack against genus two hyperelliptic curve cryptosystems over quadratic extention fields. Technical Report ISEC2002-62, IEICE, Japan, 2002. in Japanese.

[2] S. Arita, K. Matsuo, K. Nagao, and M. Shimura. A weil descent attack against elliptic curve cryptosystems over quartic extension fields. *IEICE Trans.*, E89-A(5), May 2006. 1246-1254.

[3] I. Blake, G. Seroussi, and N. Smart. *Elliptic Curves in Cryptography*. Number 265 in London Mathematical Society Lecture Note Series. Cambridge U. P., 1999.

[4] S. Furihata, T. Kobayashi, and T. Saito. Scalar multiplication on twist elliptic curves over $\mathbb{F}_{2^m}$. In *Proc. of SCIS2003*, pages 843–846, January 2003.

[5] K. Furukawa, T. Kobayashi, and K. Aoki. The cost of elliptic operations in oef using a successive extension. In *Proc. of SCIS2003*, pages 929–934, January 2003. in Japanese.

[6] P. Gaudry. Index calculus for abelian varieties and the elliptic curve discrete logarithm problem. Cryptology ePrint Archive, Report 2004/073, 2004. `http://eprint.iacr.org/`.

[7] M. Gonda, K. Matsuo, K. Aoki, J. Chao, and S. Tsujii. Improvements of addition algorithm on genus 3 hyperelliptic curves and their implementation. *IEICE Trans.*, E88-A(1), January 2005. 89-96.

[8] T. Iijima, K. Matsuo, J. Chao, and S. Tsujii. Cnstruction of Frobenius maps of twist elliptic curves and its application to elliptic scalar multiplication. In *Proc. of SCIS2002*, pages 699–702, January 2002.

[9] N. Kanayama, K. Nagao, and S. Uchiyama. Generating secure genus two hyperelliptic curves using Elkies' point counting algorithm. *IEICE Japan Trans. Fundamentals*, E86-A(4):908–918, 2003.

[10] T. Kobayashi, H. Morita, K. Kobayashi, and F. Hoshino. Fast elliptic curve algorithm combining Frobenius map and table reference to adapt to higher characteristic. In *Advances in Cryptology - EUROCRYPT '99*, number 1592 in Lecture Notes in Computer Science, pages 176–189. Springer-Verlag, 1999.

[11] N. Koblitz. Hyperelliptic curve cryptosystems. *J. Cryptology*, 1(3):139–150, 1989.

[12] N. Koblitz. CM-curves with good cryptographic properties. In *Advances in Cryptology - CRYPTO '91*, number 576 in Lecture Notes in Computer Science, pages 279–287, 1992.

[13] A. Menezes, Y. Wu, and R. Zuccherato. An elementary introduction to hyperelliptic curve. `http://cacr.uwaterloo.ca/techreports/1997/corr96-19.ps`, 1996.

[14] J. Nyukai, K. Matsuo, J. Chao, and S. Tsujii. On the resultant computation in the addition Harley algorithms on hyperelliptic cureves. (ISEC2006-5), July 2006. in Japanese.

[15] J. A. Solinas. An improved algorihtm for arithmetic on a family of elliptic curves. In *Advances in Cryptology - CRYPTO '97*, number 1294 in Lecture Notes in Computer Science, pages 357–371. Springer-Verlag, 1997.