Noname manuscript No.
(will be inserted by the editor)

Fast Endomorphism for any Genus 2 Hyperelliptic Curve over a Finite Field of Even Characteristic

Lei Li · Siman Yang

Received: 5 November 2010/ Revised: / Accepted:

Abstract In EUROCRYPT 2009, Galbraith, Lin and Scott constructed an efficiently computable endomorphism for a large family of elliptic curves defined over finite fields of large characteristic. They demonstrated that the endomorphism can be used to accelerate scalar multiplication in the elliptic curve cryptosystem based on these curves. In this paper we extend the method to any genus 2 hyperelliptic curve defined over a finite field of even characteristic. We propose an efficient algorithm to generate a random genus 2 hyperelliptic curve and its quadratic twist equipped with a fast endomorphism on the Jacobian. The analysis of the operation amount of the scalar multiplication is also given.

Keywords Hyperelliptic cryptosystem · Jacobian · efficiently computable endomorphism

Mathematics Subject Classifications (2010) 94A60 11G20

1 Introduction

In recent years elliptic curve cryptosystem and hyperelliptic cryptosystem have been extensively studied and deployed in the real world. The Jacobians of hyperelliptic curves provide an interesting alternative to elliptic curves for the design of discrete-log based cryptosystems due to a larger variety of the curves and smaller size of the underlying field for the same security level comparing with elliptic curves. Similar as in the elliptic curve cryptosystem, the most time-consuming operation in the hyperelliptic cryptosystem is the scalar multiplication (or point multiplication) $[k]D$ by a large integer k for a divisor D on the Jacobian of a hyperelliptic curve. The most conventional way is the *double-and-add* method, which requires in average l doubles and $l/2$ additions, where l is the length of binary representation of k. Many methods have been proposed to accelerate scalar multiplication. An improvement was carried out by Koblitz on certain characteristic 2 elliptic curves, now called the Koblitz curves [13] using the Frobenius map. The method was generalized by Gunther *et al.* [8] to hyperelliptic curves. Gallant, Lambert and Vanstone (GLV) [6] showed that certain efficiently computable endomorphisms ϕ can be used to accelerate scalar multiplication on certain elliptic curves with the decomposition $[k]D = [k_0]D + [k_1]\phi(D)$, where k_0, k_1 are almost half size of k. Park *et al.* [19] extended the GLV method to the hyperelliptic setting. The related arithmetic and algorithms have been studied thoroughly (cf. [14] and [22]).

However, GLV method is only applicable to very special elliptic curves (analyzed in Section 7 of [20]). Iijima *et al.* [11] first considered an endomorphism via composition of the Frobenius map defined over a quadratic twist curve of the underlying elliptic curve. Galbraith, Lin and Scott (GLS) [5] applied this endomorphism to a large class of elliptic curves to achieve a fast scalar multiplication method. They also noticed similar endomorphism can be constructed for genus 2 hyperelliptic curves defined over finite fields of odd characteristic. Hankerson *et al.* [10] analyzed the chance for GLS curves to be vulnerable to the generalized GHS Weil descent attack [7] is small.

L.Li · S.Yang (Corresponding author)
Department of Mathematics, East China Normal University, Shanghai, 200241, China
E-mail: smyang@math.ecnu.edu.cn (S.Yang)

L.Li
E-mail: lileiat163@163.com

The goal of this paper is to extend the GLS construction to all genus two hyperelliptic curves defined over finite fields of even characteristic. Furthermore we propose an efficient algorithm to implement it.

The rest of the paper is organized as follows. In Section 2, we briefly summarize some facts on the Jacobians of hyperelliptic curves. In Section 3, we construct an efficiently computable endomorphism on the Jacobian of the quadratic twist curve for each isomorphism class of genus 2 hyperelliptic curves defined over finite fields of even characteristic. In Section 4, We propose an efficient algorithm to generate a random genus 2 hyperelliptic curve and its quadratic twist equipped with a fast endomorphism on the Jacobian using Vercauteren's algorithm [25]. In Section 5 we show the speedup of the scalar multiplication in the hyperelliptic system with our endomorphism. Section 6 discusses known attacks and explains how to avoid them.

2 Preliminaries

In this section, we summary some basic definitions and facts that will be used in throughout this paper. For an extensive display, the readers are referred to [16].

2.1 Hyperelliptic Curves

Let $\mathbb{F}_q$ be a finite field and fix an algebraic closure $\overline{\mathbb{F}}_q$ of it. Throughout this paper we fix $q = 2^l$ for some prime l. Suppose that we are given a genus g hyperelliptic curve C with one point at infinity defined over $\mathbb{F}_q$ with an affine model

$$C: \quad Y^2 + H(X)Y = F(X), \tag{1}$$

where $H(X),\ F(X) \in \mathbb{F}_q[X],\ F(X)$ is monic, $degF(X) = 2g+1,\ degH(X) \leq g$. There are no singular points on $C(\overline{\mathbb{F}}_q)$, i.e., if $(x,y) \in \overline{\mathbb{F}}_q \times \overline{\mathbb{F}}_q$ is a solution of (1), then $2y + H(x)$ and $H'(x)y - F'(x)$ do not vanish simultaneously.

For a given integer $n > 1$ there is a unique degree n extension field of $\mathbb{F}_q$ in $\overline{\mathbb{F}}_q$, denoted by $\mathbb{F}_{q^n}$. The curve C defined in (1) can also be regarded as a curve defined over $\mathbb{F}_{q^n}$. The set of $\mathbb{F}_{q^n}$-rational points on C, denoted by $C(\mathbb{F}_{q^n})$, is the set of all points $P = (x,y) \in \mathbb{F}_{q^n} \times \mathbb{F}_{q^n}$ satisfying equation (1) together with a special point at infinity (denoted by ∞). The set of points $C(\overline{\mathbb{F}}_q)$ is simply denoted by C. The points in C other than ∞ are called finite points. The canonical involution of C is defined by $P = (x,y) \mapsto \tilde{P} = (x, -y - H(x))$. If $P = \infty$ then define $\tilde{P} = \infty$. As there is no group law of the points in a hyperelliptic curve, one uses the group of the Jacobian of a hyperelliptic elliptic curve in hyperelliptic cryptosystems, the main object considered in this paper.

The divisor class group of a hyperelliptic curve C denoted $\mathbb{D}_C$, is the free abelian group generated by the points of C. A divisor $D \in \mathbb{D}_C$ is a formal sum of points in C, i.e. $D = \sum_{P \in C} n_P P$ with $n_P \in \mathbb{Z}$ and $n_P = 0$ for almost all P's. The degree of D is defined as $deg(D) = \sum_{P \in C} n_P$. A divisor D is defined over $\mathbb{F}_{q^n}$, if $\sigma(D) = D$ for all $\sigma \in \mathrm{Gal}(\overline{\mathbb{F}}_q/\mathbb{F}_{q^n})$. The set $\mathbb{D}_C(\mathbb{F}_{q^n})$ of divisors defined over $\mathbb{F}_{q^n}$ forms a subgroup of $\mathbb{D}_C$. Divisors of degree zero in $\mathbb{D}_C(\mathbb{F}_{q^n})$ form a subgroup of $\mathbb{D}_C(\mathbb{F}_{q^n})$, denote by $\mathbb{D}^0_C(\mathbb{F}_{q^n})$. To every element f of the function field of C a divisor is associated via the valuations at every point of C, $div(f) = \sum_{P \in C} v_P(f)P$. Such divisors are called principal divisors and they form a subgroup of $\mathbb{D}^0_C(\mathbb{F}_{q^n})$, denoted by $\mathbb{P}_C(\mathbb{F}_{q^n})$. The quotient group $\mathbb{J}_C(\mathbb{F}_{q^n}) = \mathbb{D}^0_C(\mathbb{F}_{q^n})/\mathbb{P}_C(\mathbb{F}_{q^n})$ is called the Jacobian of $C/\mathbb{F}_{q^n}$, which is an abelian variety of dimension g over $\mathbb{F}_{q^n}$.

A reduced divisor of C is a divisor of the form $D = \sum m_i P_i - (\sum m_i)\infty$, where each $m_i \geqslant 0,\ \sum m_i \leqslant g$ (g is the genus of C) and the P_i's are finite points such that when $P_i \in supp(D)$ then $\tilde{P}_i \notin supp(D)$, unless $P_i = \tilde{P}_i$, in which case $m_i = 1$. It follows from the Riemann-Roch theorem that each divisor of a hyperelliptic curve is uniquely linear equivalent to a reduced divisor (see [21]). We handle the elements of the Jacobian through their Mumford representation [18]: a reduced divisor $D = \sum m_i P_i - (\sum m_i)\infty$, where $P_i = (x_i, y_i)$, is represented by a couple of polynomials $u(X)$ and $v(X)$, denoted by $D = (u(X),\ v(X))$, where $u(X) = \prod_{P_i \in supp(D)} (X - x_i)^{m_i}$, $deg(v) < deg(u) \leq g$, $v(X)$ is such that $v(x_i) = y_i$ for each i, and $u(X) | v^2(X) + v(X)H(X) - F(X)$. The zero of this group is represented by the pair $(1,0)$. The negative element of $D = (u(X),\ v(X))$ is $-D = (u(X), -H(X) - v(X) \mod u(X))$. If $D = (u(X), v(X)) \in \mathbb{J}_C(\mathbb{F}_q)$, then $u(X),\ v(X) \in \mathbb{F}_q[X]$. In this guise adding two elements of the Jacobian can be performed using *Cantor's algorithm* [1] for odd characteristic and Koblitz's algorithm for even characteristic [12].

It is shown in [2] that every genus 2 hyperelliptic curve over a finite field of even characteristic belongs to exactly one of the following type of isomorphism classes:

$$\begin{array}{lll} \text{Type 1:} & C: & Y^2 + (X^2 + a_3X + a_5)Y = X^5 + a_8X + a_{10},\ a_1 \neq 0 \\ \text{Type 2:} & C: & Y^2 + a_3XY = X^5 + a_4X^3 + a_6X^2 + a_{10}, \qquad a_3 \neq 0 \\ \text{Type 3:} & C: & Y^2 + a_5Y = X^5 + a_4X^3 + a_8X + a_{10}, \qquad a_5 \neq 0 \end{array}$$

The number of isomorphism classes of genus 2 hyperelliptic over $\mathbb{F}_q$ where $q = 2^l$ is $2q^3 + q^2 - q$ if l is not divided by 4, and $2q^3 + q^2 - q + 8$ otherwise [3]. In Section 3 we will construct an efficiently computable endomorphism on the Jacobian of the quadratic twist curve for every type of isomorphism classes, respectively.

2.2 Zeta-Functions

Let $\psi: C_1 \to C_2$ be a non-constant morphism between two curves. The induced pushforward map ψ_* on the Jacobian is defined by

$$\begin{array}{rccc} \psi_*: & \mathbb{J}_{C_1} & \longrightarrow & \mathbb{J}_{C_2} \\ & \sum m_iP_i - (\sum m_i)\infty & \mapsto & \sum m_i\psi(P_i) - (\sum m_i)\infty. \end{array} \tag{2}$$

Let C be a smooth projective curve of genus g defined over a finite field $\mathbb{F}_q$. The q-th power Frobenius automorphism on C defined by $(x,\ y) \mapsto (x^q,\ y^q)$ extends naturally to the Jacobian of C and its characteristic polynomial is of the form

$$P(T) = T^{2g} + c_1T^{2g-1} + \cdots + c_gT^g + qc_{g-1}T^{g-1} + \cdots + q^{g-1}c_1T + q^g, \tag{3}$$

where $c_0 = 1,\ ic_i = S_ic_0 + S_{i-1}c_1 + \cdots + S_1c_{i-1}$ for $S_i := N_i - (q^i + 1),\ 1 \leq i \leq g$ and $N_i = \#C(\mathbb{F}_{q^i})$. See [14] for details.

The zeta-function of $C/\mathbb{F}_q$ is of the form

$$Z(C/\mathbb{F}_q; T) = \frac{L(T)}{(1-T)(1-qT)},$$

where $L(T)$ is a degree $2g$ polynomial with integer coefficients, called L-polynomial. $L(T)$ is factorized in $\mathbb{C}$ as $L(T) = \prod_{i=1}^{2g}(1-\omega_iT)$ where $|\omega_i| = \sqrt{q}$ for all i and ω_i can be paired to be pairwise conjugate. It is well known (cf. [25]) that $L(1) = \#\mathbb{J}_C(\mathbb{F}_q)$ and $L(T) = T^{2g}P(1/T)$. The curve C can also be regarded as defined over an extension field $\mathbb{F}_{q^r}$ with the zeta-function (cf. [21])

$$Z(C/\mathbb{F}_{q^r}; T) = \frac{L_r(T)}{(1-T)(1-q^rT)},$$

where $L_r(T) = \prod_{i=1}^{2g}(1-\omega_i^rT)$. The $r = 2$ case will be applied in the later section.

Vercauteren [25] described an efficient algorithm to compute the zeta function of a hyperelliptic curve defined over a field $\mathbb{F}_q$ of even characteristic. His algorithm needs running time $O(2^{5+\epsilon}\log_q^{3+\epsilon})$ for a genus 2 hyperelliptic curve.

3 The Endomorphism on the Jacobian

In this section for each type of genus 2 hyperelliptic curve C defined over a finite field $\mathbb{F}_q$ of even characteristic, we give the explicit formula for the endomorphism on the Jacobian of the hyperelliptic curve C_t, where C_t is a quadratic twist of C defined over $\mathbb{F}_{q^2}$ (cf. [4]). The endomorphism on the Jacobian of $C_t/\mathbb{F}_{q^2}$ is defined as $\phi = \sigma_*\pi_*\sigma_*^{-1}$, where σ is an isomorphism between C and C_t defined over $\mathbb{F}_{q^4}$, π is the the q-th Frobenius automorphism on C, and $\_*$ is the reduced pushforward map on the Jacobian.

It is straightforward to check that the curve C_t given below is isomorphic to the quadratic twist curve listed in Galbraith's book [4].

3.1 Fast endomorphisms on Jacobians of type-1 hyperelliptic curves

Let C be a type-1 curve defined over $\mathbb{F}_q$ of the form

$$C: \quad Y^2 + (X^2 + a_3X + a_5)Y = X^5 + a_8X + a_{10},$$

and its quadratic twist over $\mathbb{F}_{q^2}$ C_t has the form

$$C_t: \quad Y^2 + (X^2 + a_3X + a_5)Y = X^5 + a_2'X^4 + a_8'X + a_{10}',$$

where a_2' is an element in $\mathbb{F}_{q^2}$ with $\mathrm{Tr}_{\mathbb{F}_{q^2}/\mathbb{F}_2}(a_2') = 1$, γ satisfies $\gamma^2 + \gamma = a_2'$,

$$\delta = \gamma a_3, \quad \varepsilon = \delta^2 + \delta a_3 + \gamma a_5, \tag{4}$$

$a_8' = \varepsilon a_3 + \delta a_5 + a_8$ and $a_{10}' = \varepsilon^2 + \varepsilon a_5 + a_{10}$.

C_t is defined over $\mathbb{F}_{q^2}$ as

$$a_8' = a_2'a_3^3 + a_8, \qquad a_{10}' = a_2'^2a_3^4 + a_2'a_5^2 + a_2'a_3^2a_5 + a_{10} \in \mathbb{F}_{q^2}. \tag{5}$$

The isomorphism $\sigma : C \to C_t$ is $(x, y) \mapsto (x, y + \gamma x^2 + \delta x + \varepsilon)$. As any reduced divisor of a genus 2 hyperelliptic curve has a support of at most two finite points, the reduced divisors D are classified into three cases:

$$\begin{array}{ll} \text{Case 1:} & D = P - \infty. \\ \text{Case 2:} & D = P_1 + P_2 - 2\infty, \quad P_1 \neq P_2. \\ \text{Case 3:} & D = 2P - 2\infty. \end{array}$$

We construct endomorphism $\phi = \sigma_*\pi_*\sigma_*^{-1}$ on Jacobian of C_t for each case, respectively.

In Case 1, $\sigma_* : P - \infty \mapsto \sigma(P) - \infty$. In Mumford representation, for $P = (x, y)$, $P - \infty = (u(X),\ v(X))$ with $u(X) = X - x,\ v(X) = y$, and $\sigma(P) - \infty = (u_t(X),\ v_t(X))$ with $u_t(X) = X - x,\ v_t(X) = y + \gamma x^2 + \delta x + \varepsilon$. Hence the expression of σ_* is $(u_0,\ v_0) \mapsto (u_0,\ v_0 + \gamma u_0^2 + \delta u_0 + \varepsilon)$. Note that σ_* is an involution. The endomorphisms ϕ on the Jacobian of C_t is of the form

$$\begin{array}{rl} \phi: \quad (u_0,\ v_0) & \overset{\sigma_*^{-1}}{\mapsto} (u_0,\ v_0 + \gamma u_0^2 + \delta u_0 + \varepsilon) \\ & \overset{\pi_*}{\mapsto} (u_0^q,\ v_0^q + \gamma^q u_0^{2q} + \delta^q u_0^q + \varepsilon^q) \\ & \overset{\sigma_*}{\mapsto} (u_0^q,\ v_0^q + (\gamma^q + \gamma)u_0^{2q} + (\delta^q + \delta)u_0^q + \varepsilon^q + \varepsilon). \end{array} \tag{6}$$

In fact, ϕ is defined over $\mathbb{F}_{q^2}$ as $\gamma^q + \gamma = \sum_{i=1}^{l}(\gamma^{2^i} + \gamma^{2^{i-1}}) = \sum_{i=1}^{l}(\gamma^2 + \gamma)^{2^{i-1}} = \sum_{i=1}^{l}(a_2')^{2^{i-1}} \in \mathbb{F}_{q^2}$, $\delta^q + \delta = (\gamma^q + \gamma)a_3 \in \mathbb{F}_{q^2}$ and $\varepsilon^q + \varepsilon = (\delta^q + \delta)^2 + (\delta^q + \delta)a_3 + (\gamma^q + \gamma)a_5 \in \mathbb{F}_{q^2}$. The cost of performing the map ϕ is roughly equal to two multiplications in $\mathbb{F}_{q^2}$.

In Case 2, $\sigma_* : P_1 + P_2 - 2\infty \mapsto \sigma(P_1) + \sigma(P_2) - 2\infty$ where $P_1 = (x_1, y_1),\ P_2 = (x_2, y_2),\ \sigma(P_1) = (x_1, y_1 + \gamma x_1^2 + \delta x_1 + \varepsilon),\ \sigma(P_2) = (x_2, y_2 + \gamma x_2^2 + \delta x_2 + \varepsilon)$ are represented by $P_1 + P_2 - 2\infty = (u(X),\ v(X)),\ \sigma(P_1) + \sigma(P_2) - 2\infty = (u_t(X),\ v_t(X))$ with $u(X) = X^2 + u_1X + u_0,\ v(X) = v_1X + v_0,\ u_t(X) = X^2 + u_{t1}X + u_{t0},\ v_t(X) = v_{t1}X + v_{t0}$. We have $u(X) = u_t(X) = X^2 + (x_1 + x_2)X + x_1x_2$, i.e., $u_{t1} = u_1,\ u_{t0} = u_0$, and $v(X) = ((y_1 + y_2)/(x_1 + x_2))X + (x_1y_2 + x_2y_1)/(x_1 + x_2)$ and also $v_t(X) = ((y_1 + y_2 + \gamma(x_1^2 + x_2^2) + \delta(x_1 + x_2))/(x_1 + x_2))X + (x_1y_2 + x_2y_1 + \gamma x_1x_2(x_1 + x_2) + \varepsilon(x_1 + x_2))/(x_1 + x_2)$, i.e., $v_{t1} = v_1 + \gamma u_1 + \delta,\ v_{t0} = v_0 + \gamma u_0 + \varepsilon$. Hence the expression of σ_* is $(u_1,\ u_0,\ v_1,\ v_0) \mapsto (u_1,\ u_0,\ v_1 + \gamma u_1 + \delta,\ v_0 + \gamma u_0 + \varepsilon)$, and thus ϕ is of the form

$$\begin{array}{rl} \phi: \ (u_1,\ u_0,\ v_1,\ v_0) & \overset{\sigma_*^{-1}}{\mapsto} (u_1,\ u_0,\ v_1 + \gamma u_1 + \delta,\ v_0 + \gamma u_0 + \varepsilon) \\ & \overset{\pi_*}{\mapsto} (u_1^q,\ u_0^q,\ v_1^q + \gamma^q u_1^q + \delta^q,\ v_0^q + \gamma^q u_0^q + \varepsilon^q) \\ & \overset{\sigma_*}{\mapsto} (u_1^q,\ u_0^q,\ v_1^q + (\gamma^q + \gamma)u_1^q + (\delta^q + \delta),\ v_0^q + (\gamma^q + \gamma)u_0^q + \varepsilon^q + \varepsilon). \end{array} \tag{7}$$

The map ϕ is also defined over $\mathbb{F}_{q^2}$ as $\gamma^q + \gamma,\ \delta^q + \delta,\ \varepsilon^q + \varepsilon \in \mathbb{F}_{q^2}$. The cost of performing the map ϕ is roughly equal to two multiplications in $\mathbb{F}_{q^2}$.

In Case 3, $\sigma_* : 2P - 2\infty \mapsto 2\sigma(P) - 2\infty$ where $P = (x, y),\ \sigma(P) = (x, y + \gamma x^2 + \delta x + \varepsilon)$. Let $2P - 2\infty = (u(X),\ v(X)),\ 2\sigma(P) - 2\infty = (u_t(X),\ v_t(X))$. In this case, $u(X) = u_t(X) = X^2 - x^2$. Let $v(X) = v_1X + v_0$, then $v^2(x) + v(x)H(x) + F(x) = 0$. Taking derivatives on both sides, obtain $v'(x)H(x) + v(x)H'(x) + F'(x) = 0$, i.e., $v_1H(x) + yH'(x) + F'(x) = 0$. Thus, $v_1 = (yH^{'}(x) + F^{'}(x))/H(x)$, and $v_0 = v_1x + y$. Similarly, put $v_t(X) = v_{t1}X + v_{t0}$, then $v_{t1} = ((y + \gamma x^2 + \delta x + \varepsilon)H_t'(x) + F_t'(x))/H_t(x) = v_1 + \delta,\ v_{t0} = v_{t1}x + y + \gamma x^2 + \delta x + \varepsilon = v_0 + \gamma x^2 + \varepsilon$, where $F'(X),\ F_t'(X),\ H'(X),\ H_t'(X)$ denote the derivatives of $F(X),\ F_t(X),\ H(X),\ H_t(X)$, respectively. Hence the expression of σ_* is $(u_1,\ u_0,\ v_1,\ v_0) \mapsto (u_1,\ u_0,\ v_1 + \delta,\ v_0 + \gamma u_0 + \varepsilon)$. The map $\phi = \sigma_*\pi_*\sigma_*^{-1}$ is of the form

$$(u_1,\ u_0,\ v_1,\ v_0) \mapsto (u_1^q,\ u_0^q,\ v_1^q + \delta^q + \delta,\ v_0^q + (\gamma^q + \gamma)u_0^q + \varepsilon^q + \varepsilon). \tag{8}$$

3.2 Fast endomorphisms on Jacobians of type-2 hyperelliptic curves

Suppose a hyperelliptic curve $C/\mathbb{F}_q$ is defined by $Y^2 + a_3XY = X^5 + a_4X^3 + a_6X^2 + a_{10}$ with $a_3 \neq 0$ and its quadratic twist C_t is given by

$$C_t: \quad Y^2 + a_3XY = X^5 + a_4X^3 + a_6'X^2 + a_{10},$$

where $a_6' \in \mathbb{F}_{q^2}$ satisfies $\mathrm{Tr}_{\mathbb{F}_{q^2}/\mathbb{F}_2}(a_3^{-2}a_6') = 1$. The isomorphism map $\sigma : C \to C_t$ is given by $(x, y) \mapsto (x, y + \delta x)$, where $\delta^2 + a_3\delta = a_6 + a_6'$. The fact $\delta \in \mathbb{F}_{q^4}\backslash\mathbb{F}_{q^2}$ follows from the next result.

Lemma 1 *For $a, b \in \mathbb{F}_q,\ a \neq 0$, the equation $x^2 + ax + b = 0$ has a solution in a characteristic 2 field $\mathbb{F}_q$ if and only if $\mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_2}(a^{-2}b) = 0$.*

The map σ_* is given by $(u_0,\ v_0) \mapsto (u_0,\ v_0 + \delta u_0),\ (u_1,\ u_0,\ v_1,\ v_0) \mapsto (u_1,\ u_0,\ v_1 + \delta,\ v_0)$, and ϕ is

$$\begin{array}{rll} \phi: & (u_0,\ v_0) & \mapsto (u_0^q,\ v_0^q + (\delta^q + \delta)u_0^q) \\ & (u_1,\ u_0,\ v_1,\ v_0) & \mapsto (u_1^q,\ u_0^q,\ v_1^q + \delta^q + \delta,\ v_0^q). \end{array} \tag{9}$$

The map ϕ is defined over $\mathbb{F}_{q^2}$ as $\delta^q + \delta = a_3((\delta/a_3)^q + \delta/a_3) = a_3\sum_{i=1}^{l}((\delta/a_3)^{2^i} + (\delta/a_3)^{2^{i-1}}) = a_3\sum_{i=1}^{l}((\delta/a_3)^2 + (\delta/a_3))^{2^{i-1}} = a_3\sum_{i=1}^{l}((a_6 + a_6')/a_3^2)^{2^{i-1}} \in \mathbb{F}_{q^2}$.

3.3 Fast endomorphisms on Jacobians of type-3 hyperelliptic curves

Suppose a hyperelliptic curve $C/\mathbb{F}_q$ is defined by $Y^2 + a_5Y = X^5 + a_4X^3 + a_8X + a_{10}$ with $a_5 \neq 0$ and its quadratic twist is given by

$$C_t: \quad Y^2 + a_5Y = X^5 + a_4X^3 + a_8X + a_{10}',$$

where $a_{10}' \in \mathbb{F}_{q^2}$ satisfies $\mathrm{Tr}_{\mathbb{F}_{q^2}/\mathbb{F}_2}(a_5^{-2}(a_{10}' + a_{10})) = \mathrm{Tr}_{\mathbb{F}_{q^2}/\mathbb{F}_2}(a_5^{-2}a_{10}') = 1$.

The isomorphism $\sigma: C \to C_t$ is given by $(x, y) \mapsto (x, y + \varepsilon)$ with $\varepsilon^2 + a_5\varepsilon = a_{10} + a_{10}'$. Then $\varepsilon \in \mathbb{F}_{q^4}\backslash\mathbb{F}_{q^2}$, C_t is defined over $\mathbb{F}_{q^2}$ and σ is defined over $\mathbb{F}_{q^4}$.

The pushforward map of σ is $\sigma_*: (u_0,\ v_0) \mapsto (u_0,\ v_0 + \varepsilon),\ (u_1,\ u_0,\ v_1,\ v_0) \mapsto (u_1,\ u_0,\ v_1,\ v_0 + \varepsilon)$, and thus $\phi = \sigma_*\pi_*\sigma_*^{-1}$ is

$$\begin{array}{rll} \phi: & (u_0,\ v_0) & \mapsto (u_0^q,\ v_0^q + \varepsilon^q + \varepsilon) \\ & (u_1,\ u_0,\ v_1,\ v_0) & \mapsto (u_1^q,\ u_0^q,\ v_1^q,\ v_0^q + \varepsilon^q + \varepsilon). \end{array} \tag{10}$$

The map ϕ is defined over $\mathbb{F}_{q^2}$ as $\varepsilon^q + \varepsilon = a_5((\varepsilon/a_5)^q + \varepsilon/a_5) = a_5\sum_{i=1}^{l}((\varepsilon/a_5)^{2^i} + (\varepsilon/a_5)^{2^{i-1}}) = a_5\sum_{i=1}^{l}((\varepsilon/a_5)^2 + (\varepsilon/a_5))^{2^{i-1}} = a_5\sum_{i=1}^{l}((a_{10} + a_{10}')/a_5^2)^{2^{i-1}} \in \mathbb{F}_{q^2}$.

4 Implement of the Endomorphism

In this section we give a fast scalar multiplication method on the Jacobian of C_t employing GLV technique. We have

Lemma 2 *Suppose that the characteristic polynomial of the Frobenius of C is $P(t) = t^4 + c_1t^3 + c_2t^2 + qc_1t + q^2$. Then the endomorphism $\phi = \sigma_*\pi_*\sigma_*^{-1}$ defined above satisfies $(\phi^4 + c_1\phi^3 + c_2\phi^2 + qc_1\phi + q^2)D = \mathcal{O}$, for any $D \in \mathbb{J}_{C_t}(\mathbb{F}_{q^2})$.*

The following result gives the value of the scalar when the endomorphism ϕ is regarded as a scalar multiplication restricted on a unique prime order subgroup of the Jacobian. Note that the same holds in both [5] and [10].

Theorem 1 *Let r is a prime number with $r > (q+1)^2, r \mid \#\mathbb{J}_{C_t}(\mathbb{F}_{q^2})$. Then $\phi(D) = [\lambda]D$ for any $D \in \mathbb{J}_{C_t}(\mathbb{F}_{q^2})[r]$ with $\lambda^2 + 1 \equiv 0 \pmod{r}$.*

Algorithm	Key generation
OUTPUT:	$l,\ C_t,\ \phi,\ \lambda$
1.	Choose a prime l and set $q = 2^l$
2.	**repeat**
3.	Choose random $a_3,\ a_5,\ a_8$ and $a_{10} \in \mathbb{F}_q$. Compute $c_1,\ c_2$ in (3) of $C/\mathbb{F}_q$ using Vercauteren's algorithm
4.	Choose random $a_2' \in \mathbb{F}_{2^{2l}}$ such that $\mathrm{Tr}_{\mathbb{F}_{2^{2l}}/\mathbb{F}_2}(a_2') = 1$
5.	Compute $\gamma,\ \delta,\ \varepsilon$ from (4)
6.	Compute $a_8',\ a_{10}'$ from (5)
7.	Set $C_t:\ Y^2 + (X^2 + a_3X + a_5)Y = X^5 + a_2'X^4 + a_8'X + a_{10}'$
8.	Compute $t = \#\mathbb{J}_{C_t}(\mathbb{F}_{q^2})$ by (11)
9.	**until** $t = hr$ where r is a prime larger than $(q+1)^2$
10.	Define ϕ by (6), (7), (8)
11.	Let $\lambda = (q-1)^{-1}c_1^{-1}(c_2 - q^2 - 1) \pmod r$
12.	**return** $l,\ C_t,\ \phi,\ \lambda$

Proof We prove the theorem for each type of hyperelliptic curves, respectively.

Suppose C is a type-1 hyperelliptic curve. Fix a divisor $D \in \mathbb{J}_{C_t}(\mathbb{F}_{q^2})$ (i.e., $u_i,\ v_i \in \mathbb{F}_{q^2}$ for each i), we have $\phi^2:\ (u_0,\ v_0) \mapsto (u_0,\ v_0 + (\gamma^{q^2} + \gamma)u_0^2 + (\delta^{q^2} + \delta)u_0 + \varepsilon^{q^2} + \varepsilon),\ (u_1,\ u_0,\ v_1,\ v_0) \mapsto (u_1,\ u_0,\ v_1 + (\gamma^{q^2} + \gamma)u_1 + (\delta^{q^2} + \delta),\ v_0 + (\gamma^{q^2} + \gamma)u_0 + (\varepsilon^{q^2} + \varepsilon))$ from (6), (7) and (8). Since $\gamma^{q^2} + \gamma = \mathrm{Tr}_{\mathbb{F}_{q^2}/\mathbb{F}_2}(a_2') = 1$, $\delta^{q^2} + \delta = (\gamma^{q^2} + \gamma)a_3 = a_3$ and $\varepsilon^{q^2} + \varepsilon = (\delta^{q^2} + \delta)^2 + (\delta^{q^2} + \delta)a_3 + (\gamma^{q^2} + \gamma)a_5 = a_5$, $\phi^2:\ (u_0,\ v_0) \mapsto (u_0,\ v_0 + u_0^2 + a_3u_0 + a_5),\ (u_1,\ u_0,\ v_1,\ v_0) \mapsto (u_1,\ u_0,\ v_1 + u_1 + a_3,\ v_0 + u_0 + a_5)$. If $D = (X + u_0,\ v_0)$, then $-D = (X + u_0,\ X^2 + a_3X + a_5 + v_0\ (mod\ X - u_0)) = (X + u_0,\ v_0 + u_0^2 + a_3u_0 + a_5) = \phi^2(D)$. If $D = (X^2 + u_1X + u_0,\ v_1X + v_0)$, then $-D = (X^2 + u_1X + u_0,\ X^2 + (a_3 + v_1)X + a_5 + v_0\ (mod\ X^2 + u_1X + u_0)) = (X^2 + u_1X + u_0,\ (v_1 + u_1 + a_3)X + v_0 + u_0 + a_5) = \phi^2(D)$.

If C is a type-2 hyperelliptic curve, then $\phi^2:\ (u_0,\ v_0) \mapsto (u_0,\ v_0 + a_3u_0),\ (u_1,\ u_0,\ v_1,\ v_0) \mapsto (u_1,\ u_0,\ v_1 + a_3,\ v_0)$. If C is a type-3 hyperelliptic curve, then $\phi^2: (u_0,\ v_0) \mapsto (u_0,\ v_0 + a_5),\ (u_1,\ u_0,\ v_1,\ v_0) \mapsto (u_1,\ u_0,\ v_1,\ v_0 + a_5)$. In both cases $\phi^2(D) = -D$ for any $D \in \mathbb{J}_{C_t}(\mathbb{F}_{q^2})$.

The Hasse-Weil bound gives that $(q-1)^4 \le \#\mathbb{J}_{C_t}(\mathbb{F}_{q^2}) \le (q+1)^4$. If prime $r > (q+1)^2$, then $r^2 \nmid \#\mathbb{J}_{C_t}(\mathbb{F}_{q^2})$ implying that there is a unique r-order subgroup of $\mathbb{J}_{C_t}(\mathbb{F}_{q^2})$. Thus there exists an integer $\lambda \in [0,\ r-1]$ such that $\phi(D) = [\lambda]D$ for all $D \in \mathbb{J}_{C_t}(\mathbb{F}_{q^2})[r]$. As $-D = \phi^2(D) = [\lambda^2]D$, $\lambda^2 + 1 \equiv 0 \pmod r$ which finishes the proof.

Corollary 1 *The value λ is $\lambda = (q-1)^{-1}c_1^{-1}(c_2 - 1 - q^2) \pmod r$.*

Proof From Theorem 1 and Lemma 2, we have $\lambda^4 + c_1\lambda^3 + c_2\lambda^2 + qc_1\lambda + q^2 \equiv 0 \pmod r,\ \lambda^2 \equiv -1 \pmod r$. Therefore $\lambda \equiv (q-1)^{-1}c_1^{-1}(c_2 - 1 - q^2) \pmod r$.

The following result is used to compute the size of $\mathbb{J}_{C_t}(\mathbb{F}_{q^2})$.

Proposition 1 *Let C be a hyperelliptic curve over $\mathbb{F}_q$, $M_1 = \#C(\mathbb{F}_q),\ M_2 = \#C(\mathbb{F}_{q^2})$. Then $\#\mathbb{J}_{C_t}(\mathbb{F}_{q^2}) = (1 + c_2 + q^2)^2 - (c_1 + c_1q)^2 - 2(2c_2 - c_1^2)(1 + q^2)$, where $c_1 = M_1 - 1 - q$, $c_2 = (M_2 - 1 - q^2 + c_1^2)/2$.*

Proof The L-polynomial of $C/\mathbb{F}_q$ is $L(T) = 1 + c_1T + c_2T^2 + c_1qT^3 + q^2T^4$, where $c_1 = M_1 - 1 - q$, $c_2 = (M_2 - 1 - q^2 + c_1^2)/2$. Suppose $L(T)$ factors as $(1 - \alpha_1T)(1 - \bar{\alpha}_1T)(1 - \alpha_2T)(1 - \bar{\alpha}_2T)$, then $\#\mathbb{J}_C(\mathbb{F}_q) = L(1) = 1 + c_1 + c_2 + c_1q + q^2$, $\#\mathbb{J}_C(\mathbb{F}_{q^2}) = L_2(1) = (1 - \alpha_1^2)(1 - \bar{\alpha}_1^2)(1 - \alpha_2^2)(1 - \bar{\alpha}_2^2) = L(1) \times L(-1) = (1 + c_2 + q^2)^2 - (c_1 + c_1q)^2$.

Suppose the characteristic polynomial of the q^2-th Frobenius on C is $P(T) = T^4 + b_1T^3 + b_2T^2 + q^2b_1T + q^4$. Then the q^2-th Frobenius on C_t is $P(-T) = T^4 - b_1T^3 + b_2T^2 - q^2b_1T + q^4$. Thus the L-polynomial of $C_t/\mathbb{F}_{q^2}$ is $1 - b_1T + b_2T^2 - b_1q^2T^3 + q^4T^4$, and $L_2(T) = 1 + b_1T + b_2T^2 + b_1q^2T^3 + q^4T^4$.

Now we have $\#\mathbb{J}_{C_t}(\mathbb{F}_{q^2}) = L(C_t/\mathbb{F}_{q^2}; 1) = 1 - b_1 + b_2 - b_1q^2 + q^4 = \#\mathbb{J}_C(\mathbb{F}_{q^2}) - 2(b_1 + b_1q^2)$, where $b_1 = 2c_2 - c_1^2$. Thus it holds

$$\#\mathbb{J}_{C_t}(\mathbb{F}_{q^2}) = (1 + c_2 + q^2)^2 - (c_1 + c_1q)^2 - 2(2c_2 - c_1^2)(1 + q^2). \tag{11}$$

We give a key generation algorithm to generate a random type-1 hyperelliptic curve and its quadratic twist equipped with an explicit fast endomorphism and *good* Jacobian (*good* means the size of Jacobian is the product of a large prime and a small integer). The algorithm also applies to other two types of hyperelliptic curves with little change.

In the next section we employ various known methods for simultaneous scalar multiplication to compare with the operation amount of scalar multiplication with our endomorphism. The analysis shows that a significant speedup can be expected because a great number of doublings are eliminated at the expense of a few additions on the Jacobian.

Table1: Expected number of Jacobian operations by single scalar multiplication methods

	Precomputation stage	Evaluation stage
2^w-ary method	$(2^w-2)A$	$\frac{m}{w}(1-\frac{1}{2^w})A+mD$
width-w NAF method	$1D+(2^{w-2}-1)A$	$\frac{m}{w+1}A+mD$
sliding window method over NAF	$1D+(\frac{2^w-(-1)^w}{3}-1)A$	$\frac{m}{w+v(w)}A+mD$ [1]

Table 2: Expected number of Jacobian operation for multiexponentition

	Precomputation stage	Evaluation stage
Simultaneous 2^w-ary method	$(2^{2w}-3)A$	$\frac{m}{2w}(1-\frac{1}{2^{2w}})A+\frac{m}{2}D$
Simultaneous sliding window method	$1A \quad (w=1)$ $(2^{2w}-2^{2(w-1)}-2)A+2D \quad (w>1)$	$\frac{m}{2}\cdot\frac{1}{w+1/3}A+\frac{m}{2}D$
width-w NAF-based interleaving method	$0 \quad (w=2)$ $(2\cdot 2^{w-2}-2)A+2D \quad (w>2)$	$\frac{m}{w+1}A+\frac{m}{2}D$
JSF method	$2A$	$\frac{m}{4}A+\frac{m}{2}D$

5 Performance Comparisons

In this section, we compare the costs of computing a large scalar multiplication $[k]D$ for general curves and our hyperelliptic curves.

The basic idea used to speed up single scalar multiplication for general hyperelliptic curves is to use the signed binary expansions of k (e.g. non-adjacent forms) with precomputations. A natural extension of the NAF form is width-w NAF. The average density (i.e. proportion of non-zero coefficients) of a width-w NAF is $1/(w+1)$ and the precomputation costs 2^{w-2} Jacobian operations (cf. p.99 of [9]). Another method is the sliding windows over NAF expansions (cf. p.101 of [9]) which is slightly cheaper than a width-w NAF. Table 1 summarizes the average numbers of additions and doublings of different single scalar multiplication, where A and D denote cost of addition and doubling on Jacobian, m is the bitlength of k, w is the size of the window.

As analyzed in [20] the identity $\phi^2(D)=-D$ for $D\in\mathbb{J}_{C_t}(\mathbb{F}_{q^2})[r]$ leads to the decomposition $[k]D=[k_0]D+[k_1]\phi(D)$ with $k_0,\ k_1\leq\sqrt{2r}$. To use GLV decomposition to compute $k_0,\ k_1$, one firstly produces a sequence of relations $s_ir+t_i\lambda=r_i,$ for $i=0,1,2,\cdots$ by making use of the extended Euclidean algorithm applied to $r,\ \lambda$. The length of the sequence is less than $2\log_2 r$ (cf. p.226 of [24]). Hence, the cost of finding linearly independent short vectors $v_1,\ v_2$ is less than $\lceil 2\log_2 r\rceil$ integer divisions. Note that $\phi(D)$ and $v_1,\ v_2$ (which do not depend on k) can be precomputed if $\lambda,\ r$, and D are known. When k is given, one needs to take a vector close to $(k,0)$ which was generated by $v_1,\ v_2$ with integer coefficients. This is equal to solve a system of two linear equations in two unknowns over rational number field, then round the solutions to the nearest integers. They are much cheaper than computing a scalar multiplication.

Allowing precomputation, one can use the fixed window method, simultaneous sliding window method (cf. [9], Section 3.3.1 and 3.3.3) and interleaving method [17] to accelerate simultaneous scalar multiplication. Other approaches include using the joint sparse form (JSF) [23] to save additions. In Table 2 the average number of Jacobian operation of different methods for multiexponentation $[k]D=[k_0]D+[k_1]\phi(D)$ is listed, where the bitlength of components $k_0,\ k_1$ are assumed to be half that of k.

Lange [15] analyzed the cost of Jacobian operations over three different coordinate systems: affine coordinates $\mathcal{A}$, projective coordinates $\mathcal{P}$ and a new coordinates $\mathcal{N}$, as listed in Table 3 where the parameter a_1 is the coefficient in $H(X)$ in (1). The addition $\mathcal{C}_1+\mathcal{C}_2=\mathcal{C}_3$ denotes the computation of an addition, where the first input is in coordinate system $\mathcal{C}_1$, the second in $\mathcal{C}_2$ and the output is in $\mathcal{C}_3$. Same for doubling $2\mathcal{C}_1=\mathcal{C}_2$.

[1] $v(w)=4/3-(-1)^w/(3\cdot 2^{w-2})$.

Table 3: Jacobian operation cost for various coordinates

Doubling			Addition		
operation	costs		operation	costs	
	$a_1 \neq 0$	$a_1 = 0$		$a_1 \neq 0$	$a_1 = 0$
$2\mathcal{P} = \mathcal{P}$	$7S,\ 38M$	$7S,\ 36M$	$\mathcal{A} + \mathcal{P} = \mathcal{P}$	$4S,\ 39M$	$4S,\ 39M$
$2\mathcal{N} = \mathcal{N}$	$6S,\ 37M$	$6S,\ 35M$	$\mathcal{A} + \mathcal{N} = \mathcal{N}$	$5S,\ 37M$	$6S,\ 36M$

Multiplication cost in $\mathbb{F}_{2^l}$ is denoted by M, Squaring cost in $\mathbb{F}_{2^l}$ is denoted by S.

Table 4: Comparisons of cost by the number of field squaring and multiplication

Cost of sliding window method over NAF for general curves over $\mathbb{F}_{2^{127}}$		
$m = 160,\ w = 2$	$m = 160,\ w = 3$	$m = 160,\ w = 4$
$1232.7S + 7943.3M$	$1153.8S + 7385.5M$	$1138.4S + 7297.6M$
$m = 256,\ w = 2$	$m = 256,\ w = 3$	$m = 256,\ w = 4$
$1968.7S + 12679.3M$	$1836.4S + 11726.8M$	$1805.8S + 11526.2M$
Cost of wNAF-based interleaving method for our hyperelliptic curves over $\mathbb{F}_{2^{67\times 2}}$		
$m = 160,\ w = 2$	$m = 160,\ w = 3$	$m = 160,\ w = 4$
$746.7S + 4933.3M$	$702S + 4640M$	$682S + 4544M$
$m = 256,\ w = 2$	$m = 256,\ w = 3$	$m = 256,\ w = 4$
$1194.7S + 7893.3M$	$1110S + 7304M$	$1066S + 7030.4M$

The cost of our curves is calculated for the case when $a_1 \neq 0$.

In Table 4, we list the field squaring and multiplication cost to compute $[k]D$ for curves that we constructed comparing with that the best known method for general random curves, where the cost of squaring is denoted by S, and the cost of multiplication is denoted by M. We use the Jacobian doubling formulae $2\mathcal{N} = \mathcal{N}$ and addition $\mathcal{A}+\mathcal{N} = \mathcal{N}$ in Table 3 to avoid field inversions in the evaluation stage with a "penalty" which takes one field inversion and seven multiplications for each precomputed entry as to change coordinates $\mathcal{N}$ to coordinates $\mathcal{A}$. In practice, this penalty is small unless the space constraints for precomputation is very limited. We assume that the cost of one inversion equals six field multiplications as in [10]. We use the sliding window method over NAF to compute scalar multiplication $[k]D$ for general random curves and width-w NAF-based interleaving method to compute multiexponentiation $[k]D = [k_0]D + [k_1]\phi(D)$ for our curves. From the theoretical comparison of the cost listed in Table 4, it is clear that a great number of multiplications and squarings are eliminated for our curves. We stress that a strict comparison of the cost is impossible mainly due to the security requirement that our curves should be defined over $\mathbb{F}_{2^{2l}}$ with l being a prime, while the general curves are defined over $\mathbb{F}_{2^k}$ for some prime k.

6 Security Considerations

For cryptographic purposes, it is essentially necessary to have the Jacobian of a hyperelliptic curve with a large group order in the hyperelliptic cryptosystem based on the DLP on the Jacobian. In general it is computationally hard to compute the order of the Jacobian. Here we present an algorithm to test whether the twist of a random hyperelliptic curve is suitable for cryptography. We employ Vercautern's algorithm to compute the zeta function of a genus 2 hyperelliptic curve defined over a field that is half the size of the underlying field of our curve, which makes the key generation comparatively fast. We require the group order of the Jacobians of our hyperelliptic curves to be divisible by a large prime at least 160-bit to protect against Pollard-rho and BSGS attacks. To avoid the variant of Weil descent attack, we can increase the field size so that our curves are intractable to Gaudry's index-calculus attack.

7 Conclusion

We construct an efficiently computable endomorphism on the Jacobian of a quadratic twist curve of any genus 2 hyperelliptic curves defined over a finite field of even characteristic. Our construction is valid for every genus hyperellptic curve over a finite field of even characteristic. This broaden the range for selecting suitable curves in hyperelliptic cryptosystem. The performance comparison shows that our construction offers significant point multiplication acceleration via GLV decomposition technique.

Acknowledgements The second author was supported by the National Natural Science Foundation of China under Grant 10801050 and Shanghai Leading Academic Discipline Project, Project Number: B407.

References

1. Cantor D. G.: Computing in the Jacobian of A Hyperelliptic Curves. Math. Comp. **48**, 95-101 (1987).
2. Choie Y., Yun D.: Isomorphism classes of Hyperelliptic curves of genus 2 over $\mathbb{F}_q$. In: L. Batten, J. Seberry (eds.) ACISP 2002, LNCS **2384**, 190-202 (2002).
3. Deng Y., Liu M.: Isomorphism classes of hyperelliptic curves of genus 2 over finite fields with characteristic 2. Sci. China Ser. A., **49**, 173-184 (2006).
4. Galbraith S. D.: *Mathematics of Public Key Cryptography*. Available at http://www.isg.rhul.ac.uk/ s̃dg/crypto-book/.
5. Galbraith S. D., Lin X., Scott M.: Endomorphisms for Faster Elliptic Curve Cryptography on a Large Class of Curves. In: A. Joux (ed.) EUROCRYPT 2009, LNCS **5479**, 518-535 (2009).
6. Gallant R.P., Lambert R.J., Vanstone S.A.: Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms. In: J. Kilian (ed.) CRYPTO 2001, LNCS **2139**, 190-200 (2001).
7. Gaudry P., Hess F., Smart N.P.: Constructive and destructive facets of Weil Descent on elliptic curves. J. Cryptology, vol. 15, no.1, 19-46 (2002).
8. Günther C., Lange T., Stein A.: Speeding up the Arithmetic on Koblitz Curves of Genus Two. In: D.R. Stinson, S. Tavares (eds.) SAC 2000, LNCS **2012**, 106-117 (2001).
9. Hankerson D., Menezes A.J., Vanstone S.: *Guide to Elliptic Curve Cryptography*. Springer (2004).
10. Hankerson D., Karabina K., Menezes A.J.: Analyzing the Galbraith-Lin-Scott Point Multiplication Method for Elliptic Curves over Binary Fields. IEEE Transactions on Computers, 1411-1420 (2009).
11. Iijima T., Matsuo K., Chao J., Tsujii S.: Construction of Frobenius Maps of Twist Elliptic Curves and its Application to Elliptic Scalar Multiplication. SCIS 2002, IEICE, 699-702 (2002).
12. Koblitz N.: Hyperelliptic cryptosystems. J. Cryptology, vol.1, no.3, 139-150 (1989).
13. Koblitz N.: CM-curves with good cryptographic properties. In: J. Feigenbaum (ed.) CRYPTO 1991, LNCS **576**, 279-287 (1992).
14. Lange T.: *Efficient arithmetic on hyperelliptic Koblitz curves*. Ph.D. Thesis, University of Essen (2001).
15. Lange T.: Formulae for arithmetic on genus 2 hyperelliptic curves. Applicable Algebra in Engineering, Communication and Computing, **15**, 295-328 (2005).
16. Menezes A., Wu Y.-H., Zuccherato R.: An elementary introduction to hyperelliptic curves. In: *Algebraic aspects of cryptography*, by N. Koblitz, 155-178, Springer-Verlag (1997).
17. Möller B.: Algorithms for Multi-Exponentiation. In: S. Vaudenay, A.M. Youssef (eds.) SAC 2001, LNCS **2259**, 165-180 (2001).
18. Mumford D.: *Tata Lectures on Theta I*. Birkhauser (1983).
19. Park Y-H., Jeong S., Lim J.: Speeding Up Point Multiplication on Hyperelliptic Curves with Efficiently-Computable Endomorphisms. In: L.R. Knudsen (ed.) EUROCRYPT 2002, LNCS **2332**, 197-208 (2002).
20. Sica F., Ciet M., Quisquater J.-J.: Analysis of the Gallant-Lambert-Vanstone Method Based on Efficient Endomorphisms: Elliptic and Hyperelliptic Curves. In: K. Nyberg, H.M. Heys (eds.) SAC 2002, LNCS **2595**, 21-36 (2003).
21. Silverman J.H.: *The Arithmetic of Elliptic Curves*. Graduate Texts in Mathematics **106**, Springer-Verlag (1986).
22. Solinas J.A.: Efficient arithmetic on Koblitz curves. Designs, Codes and Cryptography, **19**, 195-249 (2000).
23. Solinas J.A.: Low-Weight Binary Representations for Pairs of Integers. Technical Report CORR 2001-41, University of Waterloo (2001).
24. Stinson D.R.: *Cryptography: Theory and Practice*. Discrete mathematics and its applications, Chapman & Hall/CRC (2006).
25. Vercautern F.: Computing zeta functions of hyperelliptic curves over finite fields of characteristic 2. In: M. Yung (ed.) CRYPTO 2002, LNCS **2442**, 369-384 (2002).