Implementing 4-Dimensional GLV Method on GLS Elliptic Curves with j-Invariant 0

Zhi Hu[1], Patrick Longa[2⋆], and Maozhi Xu[1]

[1] School of Mathematical Sciences,
Peking University, Beijing, 100871, P.R.China
{huzhi,mzxu}@math.pku.edu.cn
[2] Microsoft Research,
One Microsoft Way, Redmond, WA 98052, USA
plonga@microsoft.com

Abstract. The Gallant-Lambert-Vanstone (GLV) method is a very efficient technique for accelerating point multiplication on elliptic curves with efficiently computable endomorphisms. Galbraith, Lin and Scott (J. Cryptol. 24(3), 446-469 (2011)) showed that point multiplication exploiting the 2-dimensional GLV method on a large class of curves over $\mathbb{F}_{p^2}$ was faster than the standard method on general elliptic curves over $\mathbb{F}_p$, and left as an open problem to study the case of 4-dimensional GLV on special curves (e.g., $j(E) = 0$) over $\mathbb{F}_{p^2}$. We study the above problem in this paper. We show how to get the 4-dimensional GLV decomposition with proper decomposed coefficients, and thus reduce the number of doublings for point multiplication on these curves to only a quarter. The resulting implementation shows that the 4-dimensional GLV method on a GLS curve runs in about 0.78 the time of the 2-dimensional GLV method on the same curve and in between 0.78-0.87 the time of the 2-dimensional GLV method using the standard method over $\mathbb{F}_p$. In particular, our implementation reduces by up to 27% the time of the previously fastest implementation of point multiplication on x86-64 processors due to Longa and Gebotys (CHES2010).

Key words: Elliptic curves, point multiplication, GLV method, GLS curves.

1 Introduction

The fundamental operation in elliptic curve cryptography is point multiplication. In 2001, Gallant, Lambert, and Vanstone [10] described a new method (a.k.a. GLV method) for accelerating point multiplication on certain classes of elliptic curves with efficiently computable endomorphisms. Let E be an elliptic curve over a finite field $\mathbb{F}_q$ and let $P \in E(\mathbb{F}_q)$ have prime order r. Given an efficiently computable endomorphism ψ for E s.t. $\psi(P) = [\lambda]P \in \langle P \rangle$, the GLV method

[⋆] This work was partially carried out while the author was at the University of Waterloo.

consists in replacing the computation $[k]P$ by a multi-scalar multiplication with the form $[k_1]P + [k_2]\psi(P)$, where the decomposition coefficients $|k_1|, |k_2| \approx r^{1/2}$. Since the number of doublings is halved, this method potentially injects a significant speedup in the point multiplication computation on these elliptic curves. This approach might be generalized to m-dimensional case, which can achieve further speedups, if one could get higher degree decompositions with the form $[k_1]P + [k_2]\psi(P) + \ldots + [k_m]\psi(P)^{m-1}$ where $|k_i| \approx r^{1/m}$.

Constructing efficiently computable endomorphisms is one of the key problems in the GLV method. Gallant, Lambert and Vanstone gave some special examples in [10]. In 2002, Iijima, Matsuo, Chao and Tsujii [15] constructed an efficient computable homomorphism on elliptic curves $E(\mathbb{F}_{p^2})$ with $j(E) \in \mathbb{F}_p$ arising from the Frobenius map on a twist of E. Galbraith, Lin, and Scott [7,8] generalized their construction for a large class of elliptic curves over $\mathbb{F}_{p^2}$ (referred to as GLS curves) and applied the GLV method. They gave detailed implementations on these curves, showing that their method ran in between 0.70 and 0.84 the time of the best methods for elliptic curve point multiplication on general curves at that time. A detailed analysis on various x86-64 processors was carried out in [20] and later extended by Longa in [18]. Longa showed that GLS curves over $\mathbb{F}_{p^2}$ reduce costs in the range 9%-45% in comparison with general curves over $\mathbb{F}_p$, implying that the boost in performance obtained with GLS tightly depends on the particular platform.

Since previous applications of the GLV method have been limited to dimension 2, scalar decomposition with coefficients of size $O(r^{1/2})$ has been extensively studied [17,23,25,9,7,8]. In an earlier work [27], Zhou et al. tried the LLL algorithm [6, Alg. 2.6.3] to get a reduced basis for computing Babai rounding in the 3-dimensional case. Galbraith, Lin and Scott also showed how to get a 4-dimensional construction on certain GLS curves with j-invariant 0. Simultaneously to the present work, Birkner and Sica [4] have proposed a 4-dimensional decomposition method and provided a general bound for the size of the coefficients given by $O(C \cdot r^{1/4})$ for some explicit $C > 100$.

Our contributions can be summarized as follows:

- We describe an explicit lattice basis with good properties for 4-dimensional GLV decomposition on GLS curves with j-invariant 0. The decomposed coefficients are shown to have size $O(2\sqrt{2}r^{1/4}) \leq 2\sqrt{2p}$, thus enable the reduction of the number of doublings on these curves to a quarter.
- We extend Brown, Myers and Solinas's decomposition method for "compact curves" to support 2-dimensional GLV decomposition on ordinary curves with j-invariant 0 [5]. According to Sica et al.'s analysis [25], our upper bound for the size of decomposition coefficients is optimal.
- We realize high-speed implementations of point multiplication at the 128-bit security level on x86-64 processors using j-invariant 0 curves: i) over $\mathbb{F}_p$ using the 2-dimensional GLV method; ii) over $\mathbb{F}_{p^2}$ using the 2-dimensional GLV method; and iii) over $\mathbb{F}_{p^2}$ using the 4-dimensional GLV method.

The resulting implementations show that the 4-dimensional GLV method on a j-invariant 0 GLS curve runs in 0.78 the time of the 2-dimensional GLV method on the same curve and in between 0.78-0.87 the time of the 2-dimensional GLV method using the standard method over $\mathbb{F}_p$. In comparison with the previously fastest implementation due to Longa and Gebotys [20] (see also [18, Ch. 5]), which uses a 2-dimensional GLV-based GLS curve with Twisted Edwards coordinates and runs in 166,000 cycles on a 2.7GHz Intel Core i7-2620M processor, the presented GLV4-based implementation reduces the time by up to 27%, running in only 122,000 cycles on the same platform. It is also much faster than the very recent implementation by Bernstein, Duif, Lange, Schwabe and Yang [2], which runs in 194,000 cycles on the same platform (i.e., our result reduces their time by 37% in this case).

The rest of the paper is organized as follows. Section 2 presents the definition of twist maps on elliptic curves and their constructions. Section 3 describes Galbraith-Lin-Scott elliptic curves and some efficiently computable endomorphisms on them. In Section 4 we describe our decomposition method for supporting 4-dimensional GLV. In Section 5 we present a 2-dimensional GLV decomposition method for ordinary curves over $\mathbb{F}_p$ with j-invariant 0. Our efficient implementations and the corresponding benchmark results are described in Section 6. We end this paper with some conclusions in Section 7.

2 Twists on Elliptic Curves

Let E and E' be two elliptic curves over $\mathbb{F}_q$ where q is a power of some prime p. E' is called a twist of degree d of E if there exists an isomorphism $\phi_d : E' \to E$ defined over $\mathbb{F}_{q^d}$ and d is minimal. If E' is a degree d twist of E, then the automorphism group $Aut(E)$ must contain an element of order d [12]. Moreover, if $p \geq 5$, we have $\#Aut(E)|6$ according to [26, Th. III.10.1].

All twists can be described explicitly as in [26, Prop. X.5.4]. Suppose $p \geq 5$, the set of twists of E is canonically isomorphic to ${\mathbb{F}_q}^*/({\mathbb{F}_q}^*)^d$ with $d = 2$ if $j(E) \neq 0, 1728$, $d = 4$ if $j(E) = 1728$ and $d = 6$ if $j(E) = 0$. Let E be given by a short Weierstrass equation $y^2 = x^3 + ax + b$ with $a, b \in \mathbb{F}_q$ and $D \in \mathbb{F}_q^*$. The twists corresponding to D mod $(\mathbb{F}_q^*)^d$ are given by

$$\begin{aligned}
d = 2 &: y^2 = x^3 + a/D^2x + b/D^3, \\
&\quad \phi_d : E' \to E : (x, y) \mapsto (Dx, D^{3/2}y) \\
d = 4 &: y^2 = x^3 + a/Dx, \\
&\quad \phi_d : E' \to E : (x, y) \mapsto (D^{1/2}x, D^{3/4}y) \\
d = 6 &: y^2 = x^3 + b/D, \\
&\quad \phi_d : E' \to E : (x, y) \mapsto (D^{1/3}x, D^{1/2}y).
\end{aligned}$$

Iijima, Matsuo, Chao and Tsujii [15] constructed an efficient computable homomorphism on elliptic curves $E(\mathbb{F}_{p^k})$ arising from the Frobenius map π on a twist of E:

$$\psi_d : E'(\mathbb{F}_{p^k}) \xrightarrow{\phi_d} E(\mathbb{F}_{p^{dk}}) \xrightarrow{\pi} E(\mathbb{F}_{p^{dk}}) \xrightarrow{\phi_d^{-1}} E'(\mathbb{F}_{p^k}).$$

3 The GLS Elliptic Curves

Galbraith, Lin and Scott [7,8] implemented the 2-dimensional GLV method by using an efficiently computable homomorphism on elliptic curves over $\mathbb{F}_{p^2}$. They generalized Iijima et al.'s construction as follows:

Theorem 1. *[7,8] Let E be an elliptic curve defined over $\mathbb{F}_q$ such that $\#E(\mathbb{F}_q) = q+1-t$ and let $\phi : E \to E'$ be a separable isogeny of degree i defined over $\mathbb{F}_{q^k}$ where E' is an elliptic curve defined over $\mathbb{F}_{q^m}$ with $m|k$. Let $r|\#E'(\mathbb{F}_{q^m})$ be a prime such that $r > i$ and $r|\#E'(\mathbb{F}_{q^k})$. Let π be the q-power Frobenius map on E and let $\hat{\phi} : E' \to E$ be the dual isogeny of ϕ. Define $\psi = \phi\pi\hat{\phi}$, then $\psi \in End_{\mathbb{F}_{q^k}}(E')$, and for $P \in E'(\mathbb{F}_{q^k})$ we have $\psi^k(P) - [i^k]P = \mathcal{O}_{E'}$ and $\psi^2(P) - [it]\psi(P) + [i^2q]P = \mathcal{O}_{E'}$.*

Corollary 1. *[7,8] Let $p > 3$ be a prime. Let u be a non-square in $\mathbb{F}_{p^2}$. Define $a' = u^2a$ and $b' = u^3b$, then $E'(\mathbb{F}_{p^2}) : y^2 = x^3 + a'x + b'$ is the quadratic twist of $E(\mathbb{F}_{p^2})$ and $\#E'(\mathbb{F}_{p^2}) = (p-1)^2 + t^2$. Define $\phi(x, y) = (ux, u^{3/2}y)$ and $\psi = \phi^{-1}\pi\phi$. For $P \in E'(\mathbb{F}_{q^2})[r]$, we have $\psi(P)^2 + P = \mathcal{O}_{E'}$, and $\psi(P) = [\lambda]P$ where $\lambda \equiv t^{-1}(p-1) \bmod r$.*

Consider the lattice $L = \{(x, y) \in \mathbb{Z}^2 : x + y\lambda \equiv 0 \bmod r\}$. Galbraith et al. used the basis $\{(t, p-1), (1-p, t)\}$ of some lattice $L' \subset L$ to get the 2-dimensional decomposition, with each coefficient bounded by $(p+1)/\sqrt{2}$.

Galbraith et al. also described how to get higher dimensional expansions by using elliptic curves E over $\mathbb{F}_{p^2}$ with $\#Aut(E) > 2$. The basic principle is to use a twist $\phi : E \to E'$ where E' is defined over $\mathbb{F}_{p^2}$ and ϕ is defined over $\mathbb{F}_{p^{2d}}$, and not defined over any subfield of $\mathbb{F}_{p^{2d}}$, for some even integer $d \geq 4$. A natural example is to use twist of degree 6 on elliptic curves with j-invariant 0.

Corollary 2. *[7,8] Let $p \equiv 1 \bmod 6$ and let $B \in \mathbb{F}_p^*$. Define $E : y^2 = x^3 + B$. Choose $u \in \mathbb{F}_{p^{12}}^*$ such that $u^6 \in \mathbb{F}_{p^2}$ and define $E' : y^2 = x^3 + u^6B$ over $\mathbb{F}_{p^2}$. The isomorphism $\phi : E' \to E$ is given by $\phi(x, y) = (u^2x, u^3y)$ and is defined over $\mathbb{F}_{p^{12}}$. Let $\psi = \phi^{-1}\pi\phi$. For $P \in E'(\mathbb{F}_{p^2})$, we have $\psi^4(P) - \psi^2(P) + P = \mathcal{O}_{E'}$.*

Hence the 4-dimensional GLV method can be efficiently applied to these curves. Note that $-\psi^2$ satisfies the characteristic equation $x^2 + x + 1 = 0$ and so acts as the standard automorphism $(x, y) \mapsto (\zeta_3 x, y)$ on E'; ψ^3 satisfies the characteristic equation $x^2 + 1 = 0$ and so acts as the homomorphism in Corollary 1.

4 4-Dimensional GLV Method on GLS Curves with j-Invariant 0

Let E', ψ be defined as in Corollary 2, and suppose $r|\#E'(\mathbb{F}_{p^2})$ is prime. We assume that $\#E'(\mathbb{F}_{p^2})$ is prime or nearly prime, i.e., $\log_2 r \approx 2\log_2 p$. In the following we show how to decompose the scalar k in suitable coefficients of size $O(r^{1/4})$.

Let $P \in E'(\mathbb{F}_{p^2})$ be a point of order r, and suppose $\psi(P) = [\lambda]P$ where $\lambda \in \mathbb{Z}/r\mathbb{Z}$. Denote $\langle \cdot, \cdot \rangle$ as the inner product of two vectors. Define vectors $\mathbf{\Psi} = (1, \psi, \psi^2, \psi^3)$, $\mathbf{\Lambda} = (1, \lambda, \lambda^2, \lambda^3)$, and lattice $L = \{(k_0, k_1, k_2, k_3) \in \mathbb{Z}^4 : \langle (k_0, k_1, k_2, k_3), \mathbf{\Lambda} \rangle \equiv 0 \bmod r\}$. Let $\{\mathbf{v_0}, \mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}\}$ be a basis for lattice L', where $0 \subset L' \subseteq L$. By Babai rounding method, we can decompose the scalar k as $k \equiv \langle (k_0, k_1, k_2, k_3), \mathbf{\Lambda} \rangle \bmod r$, where $\max_i\{|k_i|\} \leq 2 \max_i\{\|\mathbf{v_i}\|\}$.

In order to find a proper lattice basis $\{\mathbf{v_0}, \mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}\}$ that satisfies the upper bound $\max_i\{\|\mathbf{v_i}\|\} = O(r^{1/4})$, we first study the relations among the characteristic p, the group order r and the Frobenius trace t on our targeted elliptic curves.

Lemma 1. *Let n be an integer, then the binary quadratic form $x^2 + xy + y^2 = n$ has $6M(n)$ integral solutions, where $M(n) = \#\{k : k|n, k \equiv 1 \bmod 3\} - \#\{k : k|n, k \equiv 2 \bmod 3\}$.*

Proof. Since the discriminant of the binary quadratic form is -3 and the number of classes $h(-3) = 1$, from Theorem 1 in [14, Ch. 12.4] we can obtain that the number of integral solutions is $6\Sigma_{k|n}(\frac{-3}{k})$, where $(\cdot)$ is the Kronecker symbol. It follows that $6\Sigma_{k|n}(\frac{-3}{k}) = 6M(n)$ by the definition of Kronecker symbol. □

Lemma 2. *Let p be prime. If the equation $x^2 + xy + y^2 = p$ has one integral solution, then there exist exactly 12 integral solutions.*

Proof. Let $T = \{(x, y) \in \mathbb{Z}^2 : x^2 + xy + y^2 = p\}$. If $(a, b) \in T$, then T is not an empty set, which implies that $p \equiv 1 \bmod 3$. By Lemma 1, $\#T = 12$. We can easily verify that

$$\begin{aligned} T' = \{&(a, b), (-a, -b), (b, a), (-b, -a), \\ &(a + b, -a), (-a - b, a), (-a, a + b), (a, -a - b), \\ &(a + b, -b), (-a - b, b), (-b, a + b), (b, -a - b)\} \end{aligned}$$

is also a set of solutions for $x^2 + xy + y^2 = p$. Since p is prime, we must have $gcd(a, b) = 1$. Then the vectors in T' are pairwise different, and thus $T = T'$. □

For counting the rational points on our targeted curves, we have the next theorem.

Theorem 2. *[16, Ch. 18.3, Th. 4] Let $p \geq 5$ be prime and $p \nmid B$. Consider the elliptic curve $E : y^2 = x^3 + B$ over $\mathbb{F}_p$. If $p \equiv 2 \bmod 3$ then $\#E(\mathbb{F}_p) = p + 1$. If $p \equiv 1 \bmod 3$ let $p = \pi\overline{\pi}$ with $\pi \in \mathbb{Z}[\omega]$ and $\pi \equiv 2 \bmod 3$. Then*

$$\#E(\mathbb{F}_p) = p + 1 + \overline{\left(\frac{4B}{\pi}\right)_6}\pi + \left(\frac{4B}{\pi}\right)_6\overline{\pi}.$$

If $p \equiv 1 \bmod 3$, by the construction in Lemma 2, we can find one integral solution $(a,b) \in T$ such that $a \equiv 2 \bmod 3, b \equiv 0 \bmod 3$. So we can choose $\pi = a - b\omega$ in Theorem 2, and then $p = \pi\overline{\pi} = a^2 + ab + b^2$. Hence there are six cases of Frobenius trace $t = -\overline{(\frac{4B}{\pi})_6}\pi - (\frac{4B}{\pi})_6\overline{\pi}$ given by

$$t = \pm(a+2b), \pm(2a+b), \pm(a-b).$$

$\#E'(\mathbb{F}_{p^2})$ can be computed through [12, Prop. 8]. For example, if $t = p + 1 - \#E(\mathbb{F}_p) = a + 2b$, we can obtain that $\#E'(\mathbb{F}_{p^2}) = (p-1)^2 + (2a+b)^2$ or $\#E'(\mathbb{F}_{p^2}) = (p-1)^2 + (a-b)^2$. In fact, if we assume that $\#E'(\mathbb{F}_{p^2})$ is prime or nearly prime, there are only three cases of $\#E'(\mathbb{F}_{p^2})$ given by

$$\begin{aligned} r_1(a,b) &= (p-1)^2 + (a+2b)^2, \\ r_2(a,b) &= (p-1)^2 + (2a+b)^2, \\ r_3(a,b) &= (p-1)^2 + (a-b)^2. \end{aligned}$$

By Theorem 1 we have $\psi^4 - \psi^2 + 1 = 0$ and $\psi^2 - t\psi + p = 0$, and the following proposition can be defined.

Proposition 1. *Let notation be as above. Suppose that* $p = a^2 + ab + b^2$, *where* $a, b \in \mathbb{Z}, a \equiv 2 \bmod 3, b \equiv 0 \bmod 3$. *For*
(1) $\#E'(\mathbb{F}_{p^2}) = r_1(a,b)$, *define* $\lambda' \equiv a(1-p)(b^2 + 2ab + 1)^{-1} \bmod r$;
(2) $\#E'(\mathbb{F}_{p^2}) = r_2(a,b)$, *define* $\lambda' \equiv b(1-p)(a^2 + 2ab + 1)^{-1} \bmod r$;
(3) $\#E'(\mathbb{F}_{p^2}) = r_3(a,b)$, *define* $\lambda' \equiv (bp - a)(b^2 + 2ab - 1)^{-1} \bmod r$.
Then λ' *satisfies* $\lambda'^4 - \lambda'^2 + 1 \equiv 0 \bmod r$. *Moreover, there exists some* $i \in (\mathbb{Z}/12\mathbb{Z})^*$ *such that* $\psi^i(P) = [\lambda']P$.

Proof. We only give the detailed proof for case (1), the other two cases are analogous. Let $x = a(1-p), y = b^2 + 2ab + 1, f = x^4 - x^2y^2 + y^4$. Since $r | \#E'(\mathbb{F}_{p^2})$, $\gcd(y, r) = 1$ and

$$\begin{aligned} f = r_1(a,b) \cdot (&a^8 + 2a^7b + 3a^6b^2 - 3a^6 + 2a^5b^3 - 6a^5b + a^4b^4 + 2a^4 \\ &- 10a^4b^2 - 4a^3b^3 + 2a^3b - a^2b^4 + 11a^2b^2 + 6ab^3 + 6ab + b^4 + 2b^2 + 1). \end{aligned}$$

We obtain that $\lambda' \equiv y^{-1}x(\mathrm{mod} r)$ satisfies $\lambda'^4 - \lambda'^2 + 1 \equiv 0 \bmod r$. Moreover, since $\psi^4(P) - \psi^2(P) + P = \mathcal{O}_{E'}$, there must exist some $i \in (\mathbb{Z}/12\mathbb{Z})^*$ such that $\psi^i(P) = [\lambda']P$. □

For convenience, we assume that $\lambda = \lambda'$ in the following. Otherwise we can replace ψ with some ψ^i, where $i \in (\mathbb{Z}/12\mathbb{Z})^*$.

Proposition 2. *Let notation be as above. For*
(1) $\#E'(\mathbb{F}_{p^2}) = r_1(a,b)$, *define* $\mathbf{v} = (1, -a, 0, -b)$;
(2) $\#E'(\mathbb{F}_{p^2}) = r_2(a,b)$, *define* $\mathbf{v} = (1, -b, 0, -a)$;
(3) $\#E'(\mathbb{F}_{p^2}) = r_3(a,b)$, *define* $\mathbf{v} = (1, -a-b, 0, a)$.
Then $\langle \mathbf{v}, \mathbf{\Lambda} \rangle = 0$.

Proof. We only give proof for case (1); the other two cases are analogous. Let x, y be defined as in Proposition 1. Since

$$y^3\langle \mathbf{v}, \mathbf{\Lambda}\rangle = r_1(a,b)\cdot(a^5b+a^4b^2-2a^3b+a^3b^3+a^2b^2+4ab+b^2+1),$$

it follows that $\langle \mathbf{v}, \mathbf{\Lambda}\rangle \equiv 0 \bmod r$. □

Proposition 3. *Let notation be as above. Define the matrix*

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 1 & 0 \end{bmatrix},$$

and vectors $\mathbf{v_i} = \mathbf{v}A^i$, $i = 0, \ldots, 3$. *Then* $\{\mathbf{v_0}, \mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}\}$ *is a basis for a sublattice of* L.

Proof. We only give proof for the case $\#E'(\mathbb{F}_{p^2}) = r_1(a,b)$; the other two cases are analogous. Because $\langle \mathbf{v_i}, \mathbf{\Lambda}\rangle = 0$, we have $\mathbf{v_i} \in L$. Since $\det(\mathbf{v_0}, \mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}) \neq 0$, we define lattice $L' = \mathbf{v_0}\mathbb{Z} + \mathbf{v_1}\mathbb{Z} + \mathbf{v_2}\mathbb{Z} + \mathbf{v_3}\mathbb{Z}$, and then $L' \subseteq L$. □

Note that $\det(\mathbf{v_0}, \mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}) = \#E'(\mathbb{F}_{p^2})$, thus L' is a sublattice of $\mathbb{Z}^4$ of index $r_1(a,b)$. If $r = \#E'(\mathbb{F}_{p^2})$, we have $L' = L$.

Theorem 3. *For any integer* $k \in [1, r)$, $\{\mathbf{v_0}, \mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}\}$ *induces a* 4*-dimensional GLV decomposition* $k \equiv k_0 + k_1\lambda + k_2\lambda^2 + k_3\lambda^3 \bmod r$, *with* $\max_i\{|k_i|\} \leq 2\sqrt{2p} = O(2\sqrt{2}r^{1/4})$.

Proof. We only give proof for the case $\#E'(\mathbb{F}_{p^2}) = r_1(a,b)$; the other two cases are analogous. By Proposition 3, we have

$$\begin{aligned}\mathbf{v_0} &= \mathbf{v} = (1, -a, 0, -b), \mathbf{v_1} = \mathbf{v}A = (b, 1, -a-b, 0),\\ \mathbf{v_2} &= \mathbf{v}A^2 = (0, b, 1, -a-b), \mathbf{v_3} = \mathbf{v}A^3 = (a+b, 0, -a, 1).\end{aligned}$$

Since $|a-b| \geq 1$, we have that $|ab| \leq (a^2+b^2-1)/2$ and

$$p = a^2+b^2+ab \geq a^2+b^2-(a^2+b^2-1)/2 = \|\mathbf{v_0}\|^2/2.$$

Thus $\|\mathbf{v_0}\| \leq \sqrt{2p}$.

Note that $(-a,-b), (b, -a-b), (a+b, -a)$ all satisfy the quadratic form $x^2 + xy + y^2 = p$ by Lemma 2. We have $\|\mathbf{v_i}\| \leq \sqrt{2p}$, $i = 0, ..., 3$. Thus

$$\max_i\{|k_i|\} \leq 2\max_i\{\|\mathbf{v_i}\|\} \leq 2\sqrt{2p} = O(2\sqrt{2}r^{1/4}).$$

□

Remark 1. The 4 dimensional decomposition for GLS curves with j-invariant 1728 can be done in a similar way. In this case we find that the integral solutions of quadratic form $t^2 + s^2 = p$ induce a natural decomposition with coefficients of size $O(2r^{1/4}) < 2\sqrt{p+1}$. Here we omit the details.

5 2-Dimensional GLV Method on Ordinary Curves with j-Invariant 0

For comparison, we also consider ordinary elliptic curves over $\mathbb{F}_p$ with j-invariant 0. Let $p \equiv 1 \bmod 3$ be prime and $E(\mathbb{F}_p) : y^2 = x^3 + B$ be such an ordinary elliptic curve. Also, let $r | \#E(\mathbb{F}_p)$ and $r > 2\sqrt{\#E(\mathbb{F}_p)}$ be prime.

There is a standard automorphism $\psi(x, y) = (\zeta_3 x, y) = [\lambda](x, y)$ on this elliptic curve. For $P \in E(\mathbb{F}_p)$, $\psi^2(P) + \psi(P) + P = \mathcal{O}_E$. Thus $\lambda^2 + \lambda + 1 \equiv 0 \bmod r$, which implies that $r \equiv 1 \bmod 3$.

Two-dimensional GLV decomposition methods for this case have been proposed in [17,23,25]. Here we extend Brown, Myers and Solinas's decomposition method for "compact curves" to decompose the scalar. They regarded ψ as the third root of unity $\omega = (-1 + \sqrt{-3})/2 \in \mathbb{C}$, and considered decomposing k in the algebraic integer ring $\mathbb{Z}[\omega]$.

The **Round** operation defined in [5] as

$$\begin{aligned}\mathbf{Round}[s + z\omega] := &\lfloor (\lfloor s + z \rfloor + \lfloor 2s - z \rfloor + 2)/3 \rfloor \\ &+ \lfloor (\lfloor s + z \rfloor + \lfloor 2z - s \rfloor + 2)/3 \rfloor \cdot \omega,\end{aligned}$$

rounds an element $s + z\omega \in \mathbb{Q}[\omega]$ to the closest element of $\mathbb{Z}[\omega]$ with Euclidean distance $\leq 1/\sqrt{3}$, which, essentially, is a tool to solve the closest vector problem in some lattice of $\mathbb{Z}[\omega]$.

Lemma 3. *Let notation be as above. If there exist $a, b \in \mathbb{Z}$ such that $a^2 + ab + b^2 = r$, then for $P \in E(\mathbb{F}_p)[r]$, either $(a - b\psi)(P) = \mathcal{O}_E$ or $(b - a\psi)(P) = \mathcal{O}_E$.*

Proof. It easily follows since $a - b\lambda \equiv 0 \bmod r$ or $b - a\lambda \equiv 0 \bmod r$.

Since $r \equiv 1 \bmod 3$, by Cornacchia algorithm [6, Alg. 1.5.3] we can get an integral solution (x, y) to the equation $x^2 + 3y^2 = 4r$ such that $a = (x+y)/2$ and $b = -y$ $(a, b \in \mathbb{Z})$ satisfy $a^2 + ab + b^2 = r$. In practice, we usually choose elliptic curves generated by the complex multiplication method such that $\#E(\mathbb{F}_p) = r$. In the parameter generation stage we have $4p = 3s^2 + t^2$ for some $s, t \in \mathbb{Z}$. Then we obtain $a = (t + s - 2)/2$ and $b = -s$, which satisfy $a^2 + ab + b^2 = r$.

By Lemma 3 and the property of the **Round** operation, we obtain the next result.

Theorem 4. *Let notation be as above and $P \in E(\mathbb{F}_p)[r]$. If $(a - b\psi)(P) = \mathcal{O}_E$ and $k - (a - b\omega)$* **Round** *$[k/(a - b\omega)] = k_1 + k_2\omega$, then $[k]P = [k_1]P + [k_2]\psi(P)$, where $\max\{|k_1|, |k_2|\} \leq 2\sqrt{r}/3$.*

Compared with the previous results in [17,23,25], we observe that our upper bound in Theorem 4 is better. Actually, this bound is optimal for these curves and endomorphism, according to the special case of equilateral triangle in Sica, Ciet and Quisquater's work [25, Lemma 2].

6 Performance Evaluation

In this section, we evaluate the performance of different variants of the GLV method in practice. The main objective is to determine the gain obtained with the use of GLS curves with j-invariant 0 exploiting the 4-dimensional GLV method. We describe an efficient parameter selection, carry out the corresponding operation count and present timings when computing a variable-scalar variable-point scalar multiplication at the 128-bit security level on representative x86-64 processors. We compare results of *four* efficient alternatives based on: a GLS curve with j-invariant 0 using 4-dimensional GLV; a GLS curve with j-invariant 0 using 2-dimensional GLV; a GLS curve with Twisted Edwards coordinates using 2-dimensional GLV; and an ordinary j-invariant 0 curve using 2-dimensional GLV.

6.1 Curves

GLS curve with j-invariant 0 using 4-dimensional GLV. In this case, we use the elliptic curve given by the equation

$$E_1' : y^2 = x^3 + u \cdot 7 \tag{1}$$

defined over $\mathbb{F}_{p_1^2}$, where $p_1 = 2^{128} - 40557$, $u \in \mathbb{F}_{p_1^2}$ and $\#E_1'(\mathbb{F}_{p_1^2}) =$ 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFEC327FF5BF96F8A8A7FFFE37C5E4F5FA9A8CD is a 256-bit prime. We have that $p_1 = a^2 + ab + b^2$, $a = -532813233214206943$ and $b = 18707378648059847118$, where $a \equiv 2 \bmod 3$, $b \equiv 0 \bmod 3$. Note that $p_1 \equiv 3 \bmod 8$ and hence $\mathbb{F}_{p_1^2} = \mathbb{F}_{p_1}[i]$, where $i^2 = -1$. Let $u = 1 + i$ and $\xi = 45401944847829159044964786890828045383$ ($\xi^3 \equiv 1 \bmod p_1$). The homomorphism on affine points of E_1' is given by

$$\psi(x, y) = (w^2 x^p, w^3 y^p) = [\lambda](x, y),$$

where

$$\begin{aligned} w &= \xi^2 u^{(1-p)/6} = 27424789720863306522509119738078285705 6 + 2742478972086330652250911973807828570 56i, \\ \lambda &= a(1-p)(b^2 + 2ab + 1)^{-1} \bmod r \\ &= 4264084180618062230861895375307695559083203536590755086416491912614330096539 0. \end{aligned}$$

The 4-dimensional GLV decomposition is described in Section 4.

GLS curve with j-invariant 0 using 2-dimensional GLV. In this case, we also use curve equation (1) defined over $\mathbb{F}_{p_1^2}$ with $\psi_1 = \psi^3$ to get a 2-dimensional GLV expansion using Galbraith et al.'s techniques [7,8].

GLS curve with Twisted Edwards coordinates using 2-dimensional GLV. For this case, we use the elliptic curve given by the equation

$$E_2' : -ux^2 + y^2 = 1 + 109ux^2y^2 \tag{2}$$

defined over $\mathbb{F}_{p_2^2}$ with the Mersenne prime $p_2 = 2^{127} - 1$; $u = 2 + i \in \mathbb{F}_{p_2^2}$ is non-square. Also, $\#E_2'(\mathbb{F}_{p_2}) =$ 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFA626 1414C0DC87D3CE9B68E3B09E01A5 $= 4r$, where r is a 252-bit prime. Note that the same curve is used in [20].

Ordinary curve with j-invariant 0 using 2-dimensional GLV. For this case, we use the elliptic curve given by the equation

$$E_3 : y^2 = x^3 + 5 \tag{3}$$

defined over $\mathbb{F}_{p_3}$ with the pseudo-Mersenne prime $p_3 = 2^{256} - 1539$. Also, $\#E_3(\mathbb{F}_{p_3}) =$ 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEF95AE576CE7C 6CCA38E2B32E6FB6214B, which is a 256-bit prime. The 2-dimensional GLV decomposition method is described in Section 5.

6.2 Operation Count and Timings

For all implementations using the curves above we apply the interleaving method (INT) [11, Alg. 3.51] to perform the multi-scalar multiplication with dimension m, where $m = 2, 4$. To compute $[k]P = \sum_i [k_i]\psi^i(P)$ where $i = \{0, 1, ..., m-1\}$ and P is a variable point, we use the (fractional) width-w non-adjacent form (denoted by $(f)w$NAF) representation of each k_i, where $\log_2 |k_i| \approx \log_2 |k|/m$. In this case we need to precompute $P_j = [j]P$ and $[j]\psi^i(P) = \psi^i(P_j)$ for $j = \{1, 3, 5, ..., t\}$ and $i = \{1, 2, ..., m-1\}$. Denote by DBL, mADD and ADD the cost of performing point doubling, mixed addition and addition on a given curve, respectively. Denote by *Precomp* the cost of precomputing $P_j = [j]P$ and $[j]\psi^i(P) = \psi^i(P_j)$. The expected cost of computing $[k]P$ is approximately

$$\log_2 |k| \cdot ((\frac{2}{t+1}) \cdot \delta_{fw\mathrm{NAF}} \cdot \mathrm{mADD} + (\frac{t-1}{t+1}) \cdot \delta_{fw\mathrm{NAF}} \cdot \mathrm{ADD} + \frac{1}{m} \cdot \mathrm{DBL}) + \mathit{Precomp},$$

where $\delta_{fw\mathrm{NAF}} = \left(\lfloor \log_2 t \rfloor \ + \ (t+1)/(2^{\lfloor \log_2 t \rfloor}) \ + \ 1\right)^{-1}$.

The costs of mADD, ADD and DBL depend on the chosen coordinate system. In the following, we assume that the costs of multiplication by 2, division by 2 and subtraction are roughly equivalent to addition for simplification purposes. For the case of elliptic curves with j-invariant 0 we use Jacobian coordinates. A state-of-the-art doubling formula can be found in Longa [18, formula (6.7)], which involves 3 multiplications, 4 squarings and 7 additions (note that line evaluation operations are not considered). Addition formulas are taken from [11]. Point addition involves 11 multiplications, 3 squarings and 7 additions (Z_j^2 and

Z_j^3 are precomputed), and mixed addition involves 8 multiplications, 3 squarings and 7 additions. For the case of Twisted Edwards curves we use mixed homogeneous/extended homogeneous coordinates [13]. Formulas considered in this work can be found in [18, Appendix B1]. Point doubling involves 4 multiplications, 3 squarings and 7 additions; point addition involves 9 multiplications and 9 additions; and mixed addition involves 8 multiplications and 9 additions (considering that the cost of a multiplication with the curve parameter u is approx. equivalent to 2 additions).

Interestingly enough the LM precomputation scheme [21] also applies to curves with j-invariant 0. Since inversion in the quadratic extension field is not so expensive, we use the scheme that converts precomputation points to affine (cost given by [18, formula (3.6)]) for the GLS curves over $\mathbb{F}_{p_1^2}$. In the case of the ordinary curve over $\mathbb{F}_{p_3}$ precomputed points are left in Jacobian coordinates (cost given by [18, formula (3.4)]). For the 2- and 4-dimensional GLV method on GLS curves we need to compute 1 and 3 tables with the form $[j]\psi^i(P) = \psi^i(P_j)$, respectively. In these cases each multiplication by ψ costs approx. 2 multiplications and 1 addition. Since we set $t = 13$ (optimal), the cost is given by 14 multiplications and 7 additions in the 2-dimensional case and 42 multiplications and 21 additions in the 4-dimensional case. The total cost of *Precomp* is then 69 multiplications, 17 squarings, 56 additions and 1 inversion for dimension 2 and 97 multiplications, 17 squarings, 70 additions and 1 inversion for dimension 4. For the ordinary curve using 2-dimensional GLV, the cost of multiplying by ψ is approx. one multiplication. Since in this case $t = 15$ (optimal), the cost of computing $[j]\psi^i(P) = \psi^i(P_j)$ is given by 8 multiplications. Then the total cost of *Precomp* is 52 multiplications, 25 squarings and 56 additions. For Twisted Edwards we follow a traditional precomputation scheme (see details in [18, Sec. 5.6.2]). To compute the table $[j]\psi^i(P) = \psi^i(P_j)$ with $t = 15$ (optimal) one needs 7 multiplications with ψ on projective points, each costing approx. 4 multiplications, 1 squaring and 1.5 additions; and one multiplication with ψ on an affine point, costing 2 multiplications and 1 addition. Thus the cost is 30 multiplications, 7 squarings and 11.5 additions; and the total cost of *Precomp* is 97 multiplications, 9 squarings and 80.5 additions.

Finally, at the end of point multiplication the result should be converted to affine. This cost is given by 1 inversion, 3 multiplications and 1 squaring in those cases using Jacobian coordinates; and 1 inversion and 2 multiplications in the case using Twisted Edwards coordinates.

Table 1 summarizes operation counts of the *four* implementations using the curves described in Section 6.1. The notation mGLV+INT means interleaving using $(f)w$NAF with $(t+1)/2$ precomputed points and the m-dimensional GLV method. M, S, A and I denote multiplication, squaring, addition and inversion over $\mathbb{F}_p$, and m, s, a and i denote the same operations over $\mathbb{F}_{p^2}$.

Table 1. Point multiplication operation count, 128-bit security level

Curve	Method	Operation Count
$E'_1(\mathbb{F}_{p_1^2})$ 128-bit p	4GLV+INT, 7pts.	648.0m+407.5s+829.5a+2i
$E'_1(\mathbb{F}_{p_1^2})$ 128-bit p	2GLV+INT, 7pts.	812.0m+663.5s+1263.5a+2i
$E'_2(\mathbb{F}_{p_2^2})$ 127-bit p [20,18]	2GLV+INT, 8pts.	994.5m+393.0s+1360.8a+1i
$E_3(\mathbb{F}_{p_3})$ 256-bit p	2GLV+INT, 8pts.	892.8M+666.1S+1250.9A+1I

As can be seen, extending GLV from *two* to *four* dimensions introduces a significant reduction in the number of operations on the j-invariant 0 GLS curve E'_1. The GLS curve using 4-dimensional GLV is also significantly more efficient in terms of operation counts than a state-of-the-art implementation based on the GLS-based Twisted Edwards curve E'_2 using 2-dimensional GLV [20][18]. The comparison with the ordinary curve E_3 with j-invariant 0 using 2-dimensional GLV is more complicated. Although quadratic extension field operations are defined on a smaller field each one requires a few field operations internally (e.g., a multiplication over $\mathbb{F}_{p^2}$ involves 3 field multiplications and 5 field additions when using Karatsuba). Actual implementations are ultimately required to determine efficiency.

We have implemented the different variants mostly in C language using the MIRACL library [24]. We have written most relevant field operations in assembly and applied aggressive optimizations at the different arithmetic levels closely following the techniques discussed in [20][18]. Cycle counts obtained when running the implementations on a 2.80GHz Intel Xeon W3530 (Westmere architecture), a 3.0GHz AMD Phenom II X4 940 (Phenom II architecture) and a 2.7GHz Intel Core i7-2620M (Sandy Bridge architecture) are detailed in Table 2. In our tests we averaged the cost of 10^5 variable-scalar variable-point scalar multiplications and approximated the results to the nearest 1000 cycles. All experiments were performed on one core of the targeted processors using the GNU Compiler Compilation tool (a.k.a. GCC) for compilation.

Table 2. Point multiplication timings (in clock cycles), 64-bit processors

Curve	Method	Intel Xeon	AMD Phenom	Core i7
$E'_1(\mathbb{F}_{p_1^2})$ 128-bit p	4GLV+INT, 7pts.	162,000	150,000	122,000
$E'_1(\mathbb{F}_{p_1^2})$ 128-bit p	2GLV+INT, 7pts.	211,000	193,000	157,000
$E'_2(\mathbb{F}_{p_2^2})$ 127-bit p [20,18]	2GLV+INT, 8pts.	191,000	181,000	166,000
$E_3(\mathbb{F}_{p_3})$ 256-bit p	2GLV+INT, 8pts.	193,000	173,000	156,000
`curve25519`, 255-bit p [2]	Montgomery ladder	229,000	212,000	194,000

Closely following results from the operation count analysis, our GLS-based implementation using 4-dimensional GLV is faster than the state-of-the-art GLS-based implementation using Twisted Edwards with 2-dimensional GLV. In fact, these timings set a new speed record for point multiplication on x86-64 processors. For instance, on an AMD Phenom II X4 940 processor the new implementation reduces the best numbers presented by Longa [18, Ch. 5] in 17%. This translates to a latency of only 50us per point multiplication running on one core and a throughput of 80,000 point multiplications/second running on the four cores of the targeted AMD processor. Moreover, on an Intel Core i7-2620M processor the same implementation runs in an unprecedented mark of 122,000 cycles, enabling the computation of point multiplication in only 45us (with 2.7GHz as working frequency). The later implies a reduction of 27% in comparison with the 2-dimensional GLV implementation using Twisted Edwards [20,18].

In comparison with an ordinary curve using 2-dimensional GLV, the use of the GLS method for enabling a 4-dimensional GLV injects cost reductions in between 13% and 22%. Our results also show that the use of dimension 2 with GLS-based j-invariant 0 curves is insufficient to get competitive with the standard approach.

In Table 2, we also include cycle counts for the recent implementation of curve25519 by Bernstein et al. [2]. The results were obtained by running their software on exactly the same platforms, with the same compilation tool and under the same test conditions (e.g., using only one core with Intel's Turbo Boost and Hyper-Threading disabled). Also note that these results closely follow those available online on eBATS [3] for similar processors (see hydra2 for the case of Intel Xeon, hydra1 for the case of AMD Phenom and sandy0 for the case of Intel Core i7). In general all our implementations are significantly faster than curve25519 (independently of the method). In particular, our GLV4-based implementation runs in 0.63 the time of curve25519 on a Core i7 based on the Sandy Bridge architecture, and on a Westmere machine (i.e., the architecture targeted by [2]) our implementation runs in 0.71 the time of curve25519. However, there are important factors that should be noted. On one side, curve25519 is additionally protected against certain side-channel attacks. On the other hand, our implementations are immediately useful for any elliptic-curve protocol whereas the use of a Montgomery curve limits the applicability of curve25519 to Diffie-Hellman (DH)-like schemes only.

More results. In the full version of [2] (version of 27-09-2011), there appear comparisons with our software on an Intel Core 2 Duo E6750 and an Intel Core i5-540M. However, the cited timings correspond to measurements of preliminary versions of our code. We update the numbers here and also include results for curve25519 obtained under the same test conditions (again, using only one core with Intel's Turbo Boost and Hyper-Threading disabled). The results are as follows: the GLS-based implementation using 4-GLV runs in 189,000 and 180,000 cycles on the above processors, respect., and the GLS-based implementation using 2-GLV with Twisted Edwards coordinates runs in 210,000 and 216,000 cycles, respect. On the other hand, curve25519 runs in 308,000 and 236,000 cycles on the same platforms, respectively. Thus, it can be seen that our 4-GLV implementation runs 1.63 and 1.31 faster in these two cases.

Finally, we remark the importance of comparing results obtained on the same processor to avoid any bias in code performance comparisons.

For extended benchmark results and comparisons on different 64-bit platforms, the reader is referred to our online database [19].

7 Conclusion

We studied the performance of the 4-dimensional GLV method for faster point multiplication on some GLS curves with j-invariant 0. We showed how to get the 4-dimensional GLV decomposition with suitable coefficients bounded by $O(2\sqrt{2}r^{1/4})$, thus enabling the reduction in the number of doublings to only a quarter for point multiplication on these curves. Our high-speed implementations showed that the 4-dimensional GLV method using a j-invariant 0 curve over $\mathbb{F}_{p^2}$ runs in about 0.78 the time of the 2-dimensional GLV method on the same curve and in about 0.78-0.87 the time of the 2-dimensional GLV method on an ordinary curve over $\mathbb{F}_p$, enabling the currently fastest computation of elliptic curve point multiplication at the 128-bit security level in the literature.

8 Acknowledgments

We would like to thank Steven Galbraith and the reviewers for their very useful comments and suggestions. We also thank Diego F. Aranha for carrying out the performance tests on an Intel Core i5-540M processor. The authors Hu and Xu were supported by the Natural Science Foundation of China (Grant No. 10990011) and Open Foundation of the Key Lab for Information Assurance Technology (Grant No. KJ-11-02).

References

1. Avanzi R., Cohen H., Doche C., Frey G., Lange T., Nguyen K., Vercauteren F.: Handbook of Elliptic and Hyperelliptic Cryptography. Chapman and Hall/CRC (2006)
2. Bernstein D.J., Duif N., Lange T., Schwabe P., Yang B.-Y.: High-speed high-security signatures. In: Preneel B., Takagi T. (eds.) CHES 2011, LNCS, vol. 6917. Springer, Heidelberg (2011). Full version: eprint 2011/368, `http://eprint.iacr.org/2011/368`
3. Bernstein D.J., Lange T.: eBATS: ECRYPT Benchmarking of Asymmetric Systems (eBATS), accessed on August 5, 2011. `http://bench.cr.yp.to/ebats.html`
4. Birkner P., Sica F.: Four-dimensional Gallant-Lambert-Vanstone scalar multiplication. ArXiv:1106.5149 (2011). http://arxiv.org/abs/1106.5149
5. Brown E., Myers B.T., Solinas J.A.: Elliptic curves with compact parameters. Tech. Report, Centre for Applied Cryptographic Research (2001). `http://www.cacr.math.uwaterloo.ca/techreports/2001/corr2001-68.ps`

6. Cohen H.: A Course in Computational Algebraic Number Theory. Springer, Berlin (1996)
7. Galbraith S.D., Lin X.B., Scott M.: Endomorphisms for faster elliptic curve cryptography on a large class of curves. In: Joux A. (ed.) EUROCRYPT 2009, LNCS, vol. 5479, pp. 518-535. Springer, Heidelberg (2009)
8. Galbraith S.D., Lin X.B., Scott M.: Endomorphisms for faster elliptic curve cryptography on a Large class of curves. J. Cryptol. 24(3), 446-469 (2011)
9. Galbraith S.D., Scott M.: Exponentiation in pairing friendly groups using homomorphisms. In: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5209, pp. 211-224. Springer, Heidelberg (2008)
10. Gallant R.P., Lambert R.J., Vanstone S.A.: Faster point multiplication on elliptic curves with efficient endomorphisms. In: Kilian J.(ed.) CRYPTO 2001, LNCS, vol. 2139, pp.190-200. Springer, Heidelberg (2001)
11. Hankerson D., Menezes A.J.,Vanstone S.: Guide to Elliptic Curve Cryptography. Springer, Heidelberg (2004)
12. Hess F., Smart N., Vercauteren F.: The eta-pairing revisited. IEEE Trans. Inf. Theory. 52(10), 4595-4602 (2006)
13. Hisil H., Wong K., Carter G., Dawson E.: Twisted edwards curves revisited. In: Pieprzyk J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 326-343. Springer, Heidelberg (2008)
14. Hua L.K.: Introduction to Number Theory, translated from the Chinese by Peter Shiu. Springer, Berlin (1982)
15. Iijima T., Matsuo K., Chao J., Tsujii S.: Construction of Frobenius maps of twist elliptic curves and its application to elliptic scalar multiplication. In: SCIS 2002, IEICE Japan, 2002, pp. 699-702.
16. Ireland K., Rosen M.: A Classical Introduction to Modern Number Theory, Second Edition. GTM, vol. 84, Springer, New York (1990)
17. Kim D., Lim S.: Integer decomposition for fast scalar multiplication on elliptic curves. In: Nyberg K., Heys H.M. (eds.) SAC 2002, LNCS, vol. 2595, pp. 13-20. Springer, Heidelberg (2003)
18. Longa P.: High-speed elliptic curve and pairing-based cryptography. Ph.D Thesis, University of Waterloo (2011). http://hdl.handle.net/10012/5857
19. Longa P.: Speed Benchmarks for Elliptic Curve Scalar Multiplication (2010-2011). http://www.patricklonga.bravehost.com/speed_ecc.html#speed
20. Longa P., Gebotys C.: Efficient techniques for high-speed elliptic curve cryptography. In: Mangard S., Standacrt F.-X (eds.) CHES 2010, LNCS, vol. 6225, 80-94. Springer, Heidelberg (2010)
21. Longa P., Miri A.: New composite operations and precomputation scheme for elliptic curve cryptosystems over prime fields. In: Cramer R. (ed.) PKC 2008. LNCS, vol. 4939, pp. 229-247. Springer, Heidelberg (2008)
22. Menezes A., Van Oorschot P., Vanstone S.: Handbook of Applied Cryptography. CRC Press (1996)
23. Park Y.H., Jeong S., Kim C.H., Lim J.: An alternate decomposition of an integer for faster point multiplication on certain elliptic curves. In: Naccache D., Paillier P.(eds.) PKC 2002, LNCS, vol. 2274, pp. 323-334. Springer, Heidelberg (2002)
24. Scott M.: MIRACL-Multiprecision Integer and Rational Arithmetic C/C++ Library, updated 31/12/10, http://www.shamus.ie/index.php?page=Downloads
25. Sica F., Ciet M., Quisquater J.J.: Analysis of Gallant-Lambert-Vanstone method based on efficient endomophisms: elliptic and hyperelliptic curves. In: Nyberg K., Heys H.M. (eds.) SAC 2002, LNCS, vol. 2595, pp. 21-36. Springer, Heidelberg (2003)

26. Silverman, J.: The Arithmetic of Elliptic Curves. Springer, New York (1986)
27. Zhou Z., Hu Z., Xu M.Z., Song W.G.: Efficient 3-dimensional GLV method for faster point multiplication on some GLS elliptic curves. Information Processing Letters. 110, 1003-1006 (2010)