

Refinement of the Four-Dimensional GLV Method on Elliptic Curves

Hairong Yi^{1,2(✉)}, Yuqing Zhu^{1,2(✉)}, and Dongdai Lin^{1(✉)}

¹ State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
{yihairong,zhuyuqing,ddlin}@iie.ac.cn

² School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract. In this paper we refine the four-dimensional GLV method on elliptic curves presented by Longa and Sica (ASIACRYPT 2012). First we improve the twofold Cornacchia-type algorithm, and show that the improved algorithm possesses a better theoretic upper bound of decomposition coefficients. In particular, our proof is much simpler than Longa and Sica's. We also apply the twofold Cornacchia-type algorithm to GLS curves over \mathbb{F}_{p^4} . Second in the case of curves with j -invariant 0, we compare this improved version with the almost optimal algorithm proposed by Hu, Longa and Xu in 2012 (Designs, Codes and Cryptography). Computational implementations show that they have almost the same performance, which provide further evidence that the improved version is a sufficiently good scalar decomposition approach.

Keywords: GLV method · Elliptic curves
Four-dimensional scalar decomposition

1 Introduction

Scalar multiplication is the fundamental operation in elliptic curve cryptography. It is of vital importance to accelerate this operation and numerous methods have been extensively discussed in the literature; for a good survey, see [3]. The Gallant-Lambert-Vanstone (GLV) method [5] proposed in 2001 is one of the most important techniques that can speed up scalar multiplication on certain kinds of elliptic curves over fields of large characteristic. The underlying idea, which was originally exploited by Koblitz [10] when dealing with subfield elliptic curves of characteristic 2, is to replace certain large scalar multiplication with a relatively fast endomorphism, so that any single large scalar multiplication can be separated into two scalar multiplications with only about half bit length. If scalar multiplication can be parallelized, this two-dimensional GLV will result in a twofold performance speedup. Specifically, let E be an elliptic curve, P be a point of prime order n on it and ρ be an efficiently computable endomorphism of E satisfying $\rho(P) = \lambda P$. The GLV method consists in replacing kP with

multi-scalar multiplication of the form $k_1 + k_2\rho(P)$, where the decomposition coefficients $|k_1|, |k_2| = O(n^{1/2})$.

Higher dimensional GLV method has also been intensively studied, because m -dimensional GLV would probably achieve m -fold performance acceleration using parallel computation. In 2009, Galbraith et al. [4] proposed a new family of GLS curves on which the GLV method can be implemented. On restricted GLS curves with j -invariant 0 or 1728 they considered four dimensional GLV. Later in 2010, Zhou et al. [18] introduced a three-dimensional variant of GLV by combining the two approaches of [5] and [4]. But soon Longa and Sica [11] indicated that the more natural understanding of Zhou et al. idea is in four dimensions. Moreover they extended this idea and realized four-dimensional GLV method on quadratic twists of all previous GLV curves appeared in [5].

Apart from constructing curves and efficient endomorphisms, scalar decomposition is also a crucial step to realize the GLV method. Two approaches are often used. One uses Babai rounding with respect to a reduced lattice basis, since the problem of scalar decomposition can be reduced to solving the closest vector problem (CVP). The other uses division with remainder in some order of a number field after finding a short divisor. In two-dimensional case, these two methods have been fully analyzed, including the theoretically optimal upper bound of decomposition coefficients [16] and comparison of the two methods [13]. In four-dimensional case, Longa and Sica [11, 12] used the first approach. Instead of LLL algorithm, they introduced a specific and more efficient reduction algorithm, the twofold Cornacchia-type algorithm, to get a short basis. More importantly, they showed this new algorithm gained an improved and uniform theoretic upper bound of coefficients $C \cdot n^{1/4}$ where $C = 103\sqrt{1 + |r| + s}$ with small values r, s given by the curve, which guaranteed a relative speedup when moving from a two-dimensional to a four-dimensional GLV method over the same underlying field. As for the restricted case of GLS curves with j -invariant 0 in [4], Hu, Longa and Xu [7] essentially exploited the second approach, whereas the short divisor was found by a specific way, which led to an almost optimal upper bound of coefficients $2\sqrt{2p} = O(2\sqrt{2}n^{1/4})$.

From the analysis it seems that in j -invariant 0 case Hu et al.'s decomposition method is better than Longa et al. On the other hand, practical implementations show that Longa et al. analysis of the upper bound $C = 103\sqrt{1 + |r| + s}$ is far from compact, hence it is expected to be optimized. In this paper, we improve the original twofold Cornacchia-type algorithm described in [11, 12]. And we showed that this improved version possesses a better theoretic upper bound of decomposition coefficients $C \cdot n^{1/4}$ with $C = 6.82\sqrt{1 + |r| + s}$, which is very close to Hu et al.'s. In particular, our proof is much simpler than Longa and Sica's [12]. Finally we also make experiments to compare the improved version with the original one, which shows the former outputs a shorter basis in most cases. Moreover, we also indicate that the twofold Cornacchia-type algorithm can also be applied to the four-dimensional GLV method on GLS curves over \mathbb{F}_{p^4} [4].

It is also necessary to compare the two different four-dimensional decomposition methods (the twofold Cornacchia-type algorithm and the algorithm in [7]) just as [13] did for the two-dimensional case. To this end, we first show that a j -invariant 0 curve which is suitable for one of the four-dimensional GLV method will be applicable for the other, and by this we provide a unified way to construct a j -invariant 0 curve equipped with both endomorphisms required in [11, 12] and the endomorphism required in [4, 7]. In addition, we discover the explicit relation of the two 4-GLV methods. Next we can make comparison by computational implementation. Implementations show that our improved Cornacchia-type algorithm behaves almost the same as Hu et al. algorithm, which provide further evidence that the improved version is a sufficiently good scalar decomposition approach.

Paper Organization. The rest of the paper is organized as follows. In Sect. 2 we recall some basic facts on GLV method and GLS curves, and the main idea of Longa and Sica's to realize four-dimensional GLV. In Sect. 3 we improve the twofold Cornacchia-type algorithm and give a better upper bound, and extend this algorithm to four-dimensional GLS curves over \mathbb{F}_{p^4} . Section 4 explores the uniformity of the two four-dimensional GLV methods on j -invariant 0 curves. In Sect. 5 we compare our modified algorithm with the original one and compare the two four-dimensional decomposition methods on j -invariant 0 curves using computational implementations. Finally, in Sect. 6 we draw our conclusions.

2 A Brief Recall of GLV and GLS

2.1 The GLV Method

In this part, we briefly summarize the GLV method following [5]. Let E be an elliptic curve defined over a finite field \mathbb{F}_q . Assume that $\#E(\mathbb{F}_q)$ is almost prime (that is hn with large prime n and cofactor $h \leq 4$) and $\langle P \rangle$ is the subgroup of $E(\mathbb{F}_q)$ with order n . Let us consider a non-trivial and efficiently computable endomorphism ρ defined over \mathbb{F}_q with characteristic polynomial $X^2 + rX + s$. We call a curve satisfying the above properties a GLV curve. Then $\rho(P) = \lambda P$ for some $\lambda \in [0, n)$ where λ is a root of $X^2 + rX + s \pmod n$.

Define the group homomorphism (the GLV reduction map w.r.t. $\{1, \rho\}$)

$$\begin{aligned} f : \mathbb{Z} \times \mathbb{Z} &\rightarrow \mathbb{Z}/n \\ (i, j) &\mapsto i + \lambda j \pmod n. \end{aligned}$$

Then $\mathcal{K} = \ker f$ is a sublattice of $\mathbb{Z} \times \mathbb{Z}$ with full rank. Assume v_1, v_2 are two linearly independent vectors of \mathcal{K} satisfying $\max\{|v_1|, |v_2|\} < c\sqrt{n}$ for some positive constant c , where $|\cdot|$ denotes the maximum norm. Expressing $(k, 0)$ as the \mathbb{Q} -linear combination of v_1, v_2 and rounding coefficients to the nearest integers, we can obtain

$$kP = k_1P + k_2\rho(P), \quad |(k_1, k_2)| < c\sqrt{n}.$$

For scalar decomposition in this way, it is essential to choose a basis $\{v_1, v_2\}$ of \mathcal{K} as short as possible. To this end, Gallant et al. [5] exploited a specific algorithm, the Cornacchia's algorithm. Complete analysis of the output of this algorithm was given in [16], which showed the constant c in upper bound can be chosen as $\sqrt{1 + |r| + s}$.

2.2 The GLS Curves

In 2009, Galbraith et al. [4] extended the work of Gallant et al. and implemented this method on a wider class of elliptic curves by generalizing Iijima et al. construction [8]. For an elliptic curve E defined over \mathbb{F}_p , the latter considered its quadratic twist E' defined over \mathbb{F}_{p^k} , and constructed an efficient endomorphism on $E'(\mathbb{F}_{p^k})$ by composition of the quadratic twist map (denoted by t_2) and its inverse, and the Frobenius map π of E :

$$\psi : E'(\mathbb{F}_{p^k}) \xrightarrow{t_2^{-1}} E(\mathbb{F}_{p^{2k}}) \xrightarrow{\pi} E(\mathbb{F}_{p^{2k}}) \xrightarrow{t_2} E'(\mathbb{F}_{p^k}). \quad (1)$$

Galbraith et al. replaced t_2 with a general separable isogeny (t_2^{-1} with the dual isogeny) or particularly a twist map of higher degree¹. Instead of considering the characteristic polynomial of ψ on $E'(\mathbb{F}_{p^k})$, they use the polynomial of ψ on $E'(\mathbb{F}_{p^k})$. For example, in (1) ψ satisfies $\psi^k(P) + P = \mathcal{O}_{E'}$ for any $P \in E'(\mathbb{F}_{p^k})$. Moreover, Galbraith et al. also described how to obtain higher dimensional GLV method by using elliptic curves E over \mathbb{F}_{p^2} with $\#\text{Aut}(E) > 2$ [4, Sect. 4.1].

Theorem 1 ([4]). *Let $p \equiv 1 \pmod{6}$ and let E defined by $y^2 = x^3 + B$ be a j -invariant 0 elliptic curve over \mathbb{F}_p . Choose $u \in \mathbb{F}_{p^{12}}$ such that $u^6 \in \mathbb{F}_{p^2}$ and define $E' : y^2 = x^3 + u^6 B$ over \mathbb{F}_{p^2} . The isomorphism $t_6 : E \rightarrow E'$ is given by $t_6(x, y) = (u^2 x, u^3 y)$ and is defined over $\mathbb{F}_{p^{12}}$. Let $\Psi = t_6 \pi t_6^{-1}$. For $P \in E'(\mathbb{F}_{p^2})$ we have $\Psi^4(P) - \Psi^2(P) + P = \mathcal{O}_{E'}$.*

For this case, Hu et al. [7] described the complete implementation of 4-dimensional GLV method on such kind of GLS elliptic curves. For scalar decomposition, first they found a short vector v_1 in $\ker f$ through analyzing properties of p and $\#E'(\mathbb{F}_{p^2})$. Since \mathbb{Z}^4 is isomorphic to the order $\mathbb{Z}[\Psi]$ and $\ker f$ is isomorphic to some prime ideal \mathfrak{n} of $\mathbb{Z}[\Psi]$ (which will be explained in Sect. 2.3), this amounts to having found a short element in \mathfrak{n} , still denoted by v_1 . $\{v_1, v_1\Psi, v_1\Psi^2, v_1\Psi^3\}$ forms a sublattice of $\ker f$. Then to decompose an arbitrary scalar k under this basis is equivalent to divide k by v_1 in $\mathbb{Z}[\Psi]$ with remainder that is the decomposition of k .

We present here the pseudo-algorithm of their method. Note that p is a prime with $p \equiv 1 \pmod{6}$ and we choose u such that $\#E'(\mathbb{F}_{p^2})$ is prime or almost prime. The matrix A appeared in the algorithm is given in [7].

¹ Assume E and E' are defined over \mathbb{F}_q . E' is called a twist of degree d of E if there exists an isomorphism $t_d : E \rightarrow E'$ defined over \mathbb{F}_{q^d} and d is minimal.

Algorithm 1. (Finding a short basis)**Input:** $p, N = \#E'(\mathbb{F}_{p^2}), A$.**Output:** Four linearly independent vectors in $\ker f$: v_1, v_2, v_3, v_4 .

-
- 1) Find integers a, b such that $a^2 + ab + b^2 = p$
and $a \equiv 2 \pmod{3}, b \equiv 0 \pmod{3}$.
 - 2) Let $r_1 \leftarrow (p-1)^2 + (a+2b)^2$,
 $r_2 \leftarrow (p-1)^2 + (2a+b)^2$,
 $r_3 \leftarrow (p-1)^2 + (a-b)^2$.
 - 3) If $N = r_1$, then $v_1 \leftarrow (1, -a, 0, -b)$,
else if $N = r_2$, then $v_1 \leftarrow (1, -b, 0, -a)$,
else if $N = r_3$, then $v_1 \leftarrow (1, -a-b, 0, a)$.
 - 3) Return: $v_1, v_2 = v_1 A, v_3 = v_2 A, v_4 = v_3 A$.
-

2.3 Combination of GLS and GLV and the Twofold Cornacchia-Type Algorithm

In [11, 12], Longa and Sica put forward that choosing a GLV curve E/\mathbb{F}_p , we may obtain four-dimensional scalar multiplication on a quadratic twist of E as in Sect. 2.2.

Let E'/\mathbb{F}_{p^2} be a quadratic twist of E via the twist map $t_2 : E \rightarrow E'$. Let ρ be the non-trivial \mathbb{F}_p -endomorphism on E with $\rho^2 + r\rho + s = 0$. Suppose that $\#E'(\mathbb{F}_{p^2}) = nh$ is almost prime and $\langle P \rangle \subset E'(\mathbb{F}_{p^2})$ is the large prime subgroup. Let $\psi = t_2\pi t_2^{-1}$ and $\phi = t_2\rho t_2^{-1}$. They are defined over \mathbb{F}_{p^2} on E' . ψ, ϕ satisfy $\psi^2(P) + P = \mathcal{O}_E, \phi^2(P) + r\phi(P) + sP = \mathcal{O}_E$ with $\psi(P) = \mu P, \phi(P) = \lambda P$ respectively. Hence for any scalar $k \in [1, n-1]$ we can obtain a four dimensional decomposition

$$kP = k_1P + k_2\phi(P) + k_3\psi(P) + k_4\psi\phi(P), \quad \text{with } \max_i(|k_i|) < 2Cn^{1/4}$$

for some constant C . As in 2-dimensional GLV case, first we consider the 4-GLV reduction map w.r.t. $\{1, \phi, \psi, \phi\psi\}$

$$\begin{aligned} \mathfrak{f} : \mathbb{Z}^4 &\rightarrow \mathbb{Z}/n \\ (x_1, x_2, x_3, x_4) &\mapsto x_1 + x_2\lambda + x_3\mu + x_4\lambda\mu \pmod{n}. \end{aligned}$$

Second, find a short basis of the lattice $\ker \mathfrak{f}$: $\{v_1, v_2, v_3, v_4\}$ with $\max_i |v_i| \leq Cn^{1/4}$. Obviously, we can use LLL algorithm [2] to find a reduced basis, but the theoretic constant C is not desired [11, 16]. Then Longa and Sica proposed the twofold Cornacchia-type algorithm to find such a short basis $\{v_1, v_2, v_3, v_4\}$. It consists of the Cornacchia's algorithm in \mathbb{Z} and the Cornacchia's algorithm in $\mathbb{Z}[i]$. It is efficient but more importantly, it gives a better and uniform upper bound with constant $C = 51.5(\sqrt{1+|r|}+s)$.

View ϕ, ψ as algebraic integers satisfying $X^2 + rX + s = 0, X^2 + 1 = 0$ respectively. Assume that they generate disjoint quadratic extension of \mathbb{Q} and denote this biquadratic extension $\mathbb{Q}(\phi, \psi)$ by K . Let \mathfrak{o}_K be its ring of integers. Since the prime n is large and integer solutions λ, μ of the two polynomials

with coefficients modulo n exist, we always have that n splits completely in K [9, Theorem 7.4]. Hence there are four prime ideals of \mathfrak{o}_K lying over n . And there is only one that contains $\phi - \lambda, \psi - \mu$. Denote it by \mathfrak{n} . We have $\phi \equiv \lambda \pmod{\mathfrak{n}}$ and $\psi \equiv \mu \pmod{\mathfrak{n}}$.

The order $\mathbb{Z}[\phi, \psi] \subseteq \mathfrak{o}_K$ is a \mathbb{Z} -module of rank 4. Under the basis $\{1, \phi, \psi, \phi\psi\}$ there is a canonical isomorphism φ from \mathbb{Z}^4 to $\mathbb{Z}[\phi, \psi]$, and we can show that $\varphi(\ker f)$ is the submodule $\mathfrak{n} \cap \mathbb{Z}[\phi, \psi]$. Denote $\mathfrak{n} \cap \mathbb{Z}[\phi, \psi]$ by \mathfrak{n}' and $\mathbb{Z}[\phi, \psi]$ by \mathfrak{o} . The following composition of two maps is just the GLV reduction map f w.r.t. $\{1, \phi, \psi, \phi\psi\}$.

$$\begin{array}{ccccc} \mathbb{Z}^4 & \xrightarrow[\varphi]{\simeq} & \mathbb{Z}[\phi, \psi] & \xrightarrow{\text{mod } \mathfrak{n} \cap \mathbb{Z}[\phi, \psi]} & \mathbb{Z}/n \\ (x_1, x_2, x_3, x_4) & \longmapsto & x_1 + x_2\phi + x_3\psi + x_4\phi\psi & \longmapsto & x_1 + x_2\lambda + x_3\mu \\ & & & & + x_4\lambda\mu \pmod{n} \end{array}$$

Note that \mathfrak{o} contains the Gaussian domain $\mathbb{Z}[\psi] = \mathbb{Z}[i]$. To find a short \mathbb{Z} -basis of \mathfrak{n}' , first we find out the generator ω of the prime ideal $\mathfrak{n}' \cap \mathbb{Z}[i]$ (Gaussian domain is a PID) using the original Cornacchia's algorithm. Then $\mathfrak{n}' = \omega\mathfrak{o} + (\phi - \lambda)\mathfrak{o}$. Note that $\mathfrak{o} = \mathbb{Z}[i] + \phi \cdot \mathbb{Z}[i]$. We can deduce

$$\begin{aligned} \mathfrak{n}' &= \omega \cdot \mathbb{Z}[i] + \omega\phi \cdot \mathbb{Z}[i] + (\phi - \lambda) \cdot \mathbb{Z}[i] + \phi(\phi - \lambda) \cdot \mathbb{Z}[i] \\ &= \omega \cdot \mathbb{Z}[i] + (\phi - \lambda) \cdot \mathbb{Z}[i]. \end{aligned}$$

We can equate \mathfrak{o} with $\mathbb{Z}[i] \times \mathbb{Z}[i]$ naturally under the basis $\{1, \phi\}$. Then \mathfrak{n}' is a $\mathbb{Z}[i]$ -submodule generated by $(\omega, 0)$ and $(-\lambda, 1)$. It is essential to view \mathfrak{n}' in this way, since we may recall that in [5] Cornacchia's algorithm is just used to find a short basis of the \mathbb{Z} -submodule of \mathbb{Z}^2 generated by $(n, 0)$ and $(-\lambda, 1)$. Replacing \mathbb{Z} with $\mathbb{Z}[i]$, we can generalize the algorithm in \mathbb{Z} to the variant in $\mathbb{Z}[i]$ (Cornacchia's algorithm in $\mathbb{Z}[i]$) to obtain a short basis of \mathfrak{n}' .

$$\begin{array}{ccccc} \mathbb{Q}(\phi) & \mathbb{Z}[\phi] & (n, \phi - \lambda) & \rightsquigarrow & \mathbb{Q}(i, \phi) & \mathbb{Z}[i, \phi] & \mathfrak{n}' \\ \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow \\ \mathbb{Q} & \mathbb{Z} & n\mathbb{Z} & & \mathbb{Q}(i) & \mathbb{Z}[i] & \omega\mathbb{Z}[i] \end{array}$$

Finally, once we find a short² $\mathbb{Z}[i]$ -basis $\{v_1, v_2\}$ of \mathfrak{n}' , then $\{v_1, v_1 \cdot i, v_2, v_2 \cdot i\}$ is also a short \mathbb{Z} -basis of \mathfrak{n}' . More specifically, let $v_1 = (a_1 + b_1i, c_1 + d_1i)$, $v_2 = (a_2 + b_2i, c_2 + d_2i)$, then

$$\mathfrak{n}' = (a_1 + b_1i + (c_1 + d_1i)\phi)\mathbb{Z}[i] + (a_2 + b_2i + (c_2 + d_2i)\phi)\mathbb{Z}[i].$$

Furthermore, $\ker f = \varphi^{-1}(\mathfrak{n}')$ is generated by rows of the matrix

$$\begin{pmatrix} a_1 & c_1 & b_1 & d_1 \\ -b_1 & -d_1 & a_1 & c_1 \\ a_2 & c_2 & b_2 & d_2 \\ -b_2 & -d_2 & a_2 & c_2 \end{pmatrix}.$$

² For a vector $v = (\alpha, \beta) \in \mathbb{Z}[i] \times \mathbb{Z}[i]$, we denote by $|v|_\infty$ the maximal norm, that is $|v|_\infty = \max\{|\alpha|, |\beta|\}$ where $|\alpha|$ is the absolute value as a complex number.

3 Improvement and Extension of the Twofold Cornacchia-Type Algorithm

In this section, we give our improvement of the twofold Cornacchia-type algorithm and analyze it. We will show that the output of this improved algorithm has a much lower (better) upper bound compared with that of the original one. For the full description and analysis of the original twofold Cornacchia-type algorithm, one can refer to [12].

3.1 The Improved Twofold Cornacchia-Type Algorithm

The first part of the improved twofold Cornacchia-type algorithm is also to find out the Gaussian integer ω lying over n , which exploits the Cornacchia's algorithm in \mathbb{Z} as described in [12]. Here we briefly describe and analyze this algorithm. Note that it is the following analysis of this algorithm that inspires us to give the proof of Theorem 2.

Algorithm 2. (Cornacchia's algorithm in \mathbb{Z})

Input: Two integers: n, μ .

Output: The Gaussian integer lying over n : ω .

- 1) Let $r_0 \leftarrow n, r_1 \leftarrow \mu, t_0 \leftarrow 0, t_1 \leftarrow 1$
 - 2) While $|r_1| \geq \sqrt{n}$ do
 - $q \leftarrow \lfloor \frac{r_0}{r_1} \rfloor,$
 - $r \leftarrow r_0 - qr_1, r_0 \leftarrow r_1, r_1 \leftarrow r,$
 - $t \leftarrow t_0 - qt_1, t_0 \leftarrow t_1, t_1 \leftarrow t.$
 - 3) Return: $\omega = r_1 - it_1$.
-

This is actually the procedure to compute the gcd of n and μ using the extended Euclidean algorithm. It is well known that it produces three sequences $(r_j)_{j \geq 0}, (s_j)_{j \geq 0}$ and $(t_j)_{j \geq 0}$ satisfying

$$\begin{pmatrix} r_{j+1} & s_{j+1} & t_{j+1} \\ r_{j+2} & s_{j+2} & t_{j+2} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & -q_{j+1} \end{pmatrix} \begin{pmatrix} r_j & s_j & t_j \\ r_{j+1} & s_{j+1} & t_{j+1} \end{pmatrix}, \quad j \geq 0$$

where $q_{j+1} = \lfloor r_j / r_{j+1} \rfloor$ and the initial data

$$\begin{pmatrix} r_0 & s_0 & t_0 \\ r_1 & s_1 & t_1 \end{pmatrix} = \begin{pmatrix} n & 1 & 0 \\ \mu & 0 & 1 \end{pmatrix}.$$

These sequences also satisfy the following important properties for all $j \geq 0$:

1. $r_j > r_{j+1} \geq 0$ and $q_{j+1} \geq 1$,
2. $(-1)^j s_j \geq 0$ and $|s_j| < |s_{j+1}|$ (this holds for $j > 0$),
3. $(-1)^{j+1} t_j \geq 0$ and $|t_j| < |t_{j+1}|$,
4. $s_{j+1} r_j - s_j r_{j+1} = (-1)^{j+1} \mu$,
5. $t_{j+1} r_j - t_j r_{j+1} = (-1)^j n$,
6. $s_j n + t_j \mu = r_j$.

The former three properties make sure that

$$|t_{j+1}r_j| + |t_jr_{j+1}| = n \text{ and } |s_{j+1}r_j| + |s_jr_{j+1}| = \mu, \quad (2)$$

the former of which implies $|t_{j+1}r_j| < n$. If Algorithm 2 stops at the m -th step such that $r_m \geq \sqrt{n}$ and $r_{m+1} < \sqrt{n}$, then $|t_{m+1}| < \sqrt{n}$. Then $|\omega|^2 = |r_{m+1} - it_{m+1}|^2 = r_{m+1}^2 + t_{m+1}^2 < 2n$. Together with $n|N_{\mathbb{Z}[i]}(\omega)| = |\omega|^2$ we have $|\omega| = \sqrt{n}$.

For the (original) Cornacchia's algorithm in $\mathbb{Z}[i]$, we also have three such sequences. But just as mentioned in [12], in the j -th step with $r_j = q_{j+1}r_{j+1} + r_{j+2}$, positive quotient q_{j+1} and nonnegative remainder r_{j+2} are not available in $\mathbb{Z}[i]$. If we choose q_{j+1} as the closest Gaussian integer to r_j/r_{j+1} denoted by $\lfloor r_j/r_{j+1} \rfloor$, the former three properties will not hold any more, which makes it more difficult to analyze the behaviour of $\{|s_j|\}$ and $\{|t_j|\}$. Hence the Eq. (2), which plays a crucial role in the analysis of Cornacchia's algorithm in \mathbb{Z} , becomes invalid in $\mathbb{Z}[i]$.

For controlling $\{|s_j|\}$, Longa and Sica [12] use the notation of “good” (“bad”) index. When j is good, they obtain an upper bound of $|s_{j+1}r_j|$ (also of $|s_jr_{j+1}|$ since they are bounded each other by (2)) [12, Lemma 4]. When j is bad, they transfer the upper bound of $|s_{j+1}|$ (or $|s_j|$) to that of $|s_{j-1}|$ [12, Lemma 5]. They take $1/\sqrt{1+|r|+s}$ as the terminal condition of the main loop of the algorithm, which is indeed determined by the ability of analyzing the upper bound of $|s_j|$ and $|r_j|$.

In this paper, we give up the notation of “good” index, and replace it by something easier to work with (the following Lemma 1). This appears to be the “expected behavior” for the $\{|s_j|\}$, which leads to a neater and shorter argument. And during this improved analysis, by some calculation we obtain an optimized terminal condition of the sequence $\{r_j\}$, which is an absolute constant independent of the curve. In addition, we make a subtle modification of the second output. We describe the second part of our improved twofold Cornacchia-type algorithm in the following Algorithm 3. Note that about the running time of Algorithm 3, it is completely the same as that of the original algorithm, that is $O(\log^2 n)$. One may refer to [12].

Algorithm 3. (Improved Cornacchia's algorithm in $\mathbb{Z}[i]$)

Input: Two Gaussian integers: ω, λ .

Output: Two vectors in $\mathbb{Z}[i]^2$: v_1, v_2 .

- 1) Let $r_0 \leftarrow \lambda, r_1 \leftarrow \omega, s_0 \leftarrow 1, s_1 \leftarrow 0$
 - 2) While $|r_1| \geq \sqrt{2 + \sqrt{2}n^{1/4}}$ do
 - $q \leftarrow \lfloor \frac{r_0}{r_1} \rfloor$,
 - $r \leftarrow r_0 - qr_1, r_0 \leftarrow r_1, r_1 \leftarrow r$,
 - $s \leftarrow s_0 - qs_1, s_0 \leftarrow s_1, s_1 \leftarrow s$.
 - 3) Compute $r_2 \leftarrow r_0 - \lfloor \frac{r_0}{r_1} \rfloor r_1, s_2 \leftarrow s_0 - \lfloor \frac{r_0}{r_1} \rfloor s_1$
 - 4) Return: $v_1 = (r_1, -s_1)$,
 $v_2 = (r_0, -s_0)$ if $\max\{|r_0|, |s_0|\} \leq \max\{|r_2|, |s_2|\}$,
 $= (r_2, -s_2)$ otherwise.
-

3.2 A Better Upper Bound

Theorem 2. *The two vectors v_1, v_2 output by Algorithm 3 are $\mathbb{Z}[i]$ -linearly independent. They belong to \mathfrak{n}' and satisfy $|v_1|_\infty \leq \sqrt{2 + \sqrt{2}}n^{1/4}$, $|v_2|_\infty \leq (2 + \sqrt{2})(\sqrt{1 + |r| + s})n^{1/4}$.*

Before proving the theorem, we need the following two lemmas. Lemma 1 replaces Longa and Sica's Lemma 4 in [12], and is crucial to our proof of Theorem 2.

Lemma 1. *If $|\frac{s_j}{s_{j+1}}| < 1$, then we have*

$$|s_{j+1}r_j| \leq (2 + \sqrt{2})|\omega|, \quad |s_jr_{j+1}| \leq (3 + \sqrt{2})|\omega|.$$

Proof. First we have $s_{j+1}r_j - s_jr_{j+1} = (-1)^{j+1}\omega$. If the condition $|\frac{s_j}{s_{j+1}}| < 1$ holds, and noticing that $|\frac{r_{j+1}}{r_j}| \leq \frac{1}{\sqrt{2}}$, from $|\frac{s_j}{s_{j+1}} \cdot \frac{r_{j+1}}{r_j}| < \frac{1}{\sqrt{2}}$ we can deduce

$$\left|1 - \frac{s_jr_{j+1}}{s_{j+1}r_j}\right| \geq 1 - \left|\frac{s_jr_{j+1}}{s_{j+1}r_j}\right| > 1 - \frac{1}{\sqrt{2}}.$$

Together with $s_{j+1}r_j - s_jr_{j+1} = (-1)^{j+1}\omega$ we have

$$|\omega| = |s_{j+1}r_j - s_jr_{j+1}| > (1 - \frac{1}{\sqrt{2}})|s_{j+1}r_j|,$$

which implies

$$|s_{j+1}r_j| \leq \frac{1}{1 - \frac{1}{\sqrt{2}}}|\omega| = (2 + \sqrt{2})|\omega|,$$

and

$$|s_jr_{j+1}| \leq (3 + \sqrt{2})|\omega|.$$

□

Lemma 2. *For any nonzero vector $(\alpha, \beta) \in \mathfrak{n}' \subset \mathbb{Z}[i]^2$ we have*

$$\max\{|\alpha|, |\beta|\} \geq \frac{\sqrt{|\omega|}}{\sqrt{1 + |r| + s}}.$$

Proof. The key point is that \mathfrak{n}' is an ideal in \mathfrak{o} with norm n , then the norm of any nonzero element in \mathfrak{n}' is divisible by n , hence no less than n . Note that here the norm is from $\mathbb{Z}[i, \phi]$ to $\mathbb{Z}[i]$. Complete proof can be found in [16]. □

Proof (Proof of Theorem 2). The vectors v_1, v_2 are $\mathbb{Z}[i]$ -linearly independent according to the fourth property, and they belong to \mathfrak{n}' because $(r_j, -s_j) = t_j(\omega, 0) + (-s_j)(-\lambda, 1)$ deduced from the sixth property.

We denote the output $\{r, s\}$ of the j -th step in the loop of Algorithm 3 by $\{r_{j+1}, s_{j+1}\}$, and assume Algorithm 3 stops at the m -th step. Then $v_1 = (r_{m+1}, -s_{m+1})$ and $|r_m| \geq \sqrt{2 + \sqrt{2}}n^{1/4}$ and $|r_{m+1}| < \sqrt{2 + \sqrt{2}}n^{1/4}$.

We need to consider two cases. For brevity, we denote two constants $\sqrt{1+|r|+s}$, $\sqrt{2+\sqrt{2}}$ by c_1, c_2 respectively.

For the case $|\frac{s_m}{s_{m+1}}| < 1$, using Lemma 1 we have $|s_{m+1}| \leq c_2\sqrt{|\omega|}$. Together with $|r_{m+1}| < c_2\sqrt{|\omega|}$ we can easily deduce

$$|v_1|_\infty \leq c_2 n^{1/4}.$$

Moreover, if $|r_{m+1}| < \frac{\sqrt{|\omega|}}{c_1}$, by Lemma 2 we have a lower bound $|s_{m+1}| \geq \frac{\sqrt{|\omega|}}{c_1}$, which implies $|r_m| \leq c_1(2+\sqrt{2})\sqrt{|\omega|}$ using again Lemma 1. Together with the restricted condition $|s_m| < |s_{m+1}| \leq c_1(2+\sqrt{2})\sqrt{|\omega|}$ we can obtain

$$|(r_m, -s_m)|_\infty \leq c_1(2+\sqrt{2})n^{1/4}.$$

If $|r_{m+1}| \geq \frac{\sqrt{|\omega|}}{c_1}$, when $|s_{m+1}| \geq |s_{m+2}|$ we have

$$|s_{m+2}| < c_2\sqrt{|\omega|}, \quad |r_{m+2}| \leq |r_{m+1}| < c_2\sqrt{|\omega|}.$$

When $|s_{m+1}| < |s_{m+2}|$ we can use Lemma 1 to deduce $|s_{m+2}| \leq c_2(2+\sqrt{2})\sqrt{|\omega|}$. Hence in both cases we have

$$|(r_{m+2}, -s_{m+2})|_\infty \leq c_1(2+\sqrt{2})n^{1/4}.$$

Finally by the definition of v_2 we always have

$$|v_2|_\infty \leq c_1(2+\sqrt{2})n^{1/4}.$$

For the case $|\frac{s_m}{s_{m+1}}| \geq 1$, let $k \leq m$ be the index satisfying

$$|s_k| \geq |s_{k+1}| \geq \dots \geq |s_m| \geq |s_{m+1}| \quad \text{and} \quad |s_{k-1}| < |s_k|.$$

Applying Lemma 1 to the $(k-1)$ -th step we have $|s_k r_{k-1}| \leq (2+\sqrt{2})|\omega|$. Since $|r_{k-1}| > |r_k| > \dots > |r_m| \geq c_2\sqrt{|\omega|}$ we can easily deduce $|s_k| \leq c_2\sqrt{|\omega|}$ and then $|s_{m+1}| \leq |s_k| \leq c_2\sqrt{|\omega|}$. Together with $|r_{m+1}| < c_2\sqrt{|\omega|}$ we obtain

$$|v_1|_\infty \leq c_2 n^{1/4}.$$

Similarly, if $|r_{m+1}| < \frac{\sqrt{|\omega|}}{c_1}$ we have $|s_{m+1}| \geq \frac{\sqrt{|\omega|}}{c_1}$ by Lemma 2, which implies $|s_k| \geq \frac{\sqrt{|\omega|}}{c_1}$ and then $|r_{k-1}| \leq c_1(2+\sqrt{2})\sqrt{|\omega|}$ by Lemma 1. Hence $|r_m| \leq c_1(2+\sqrt{2})\sqrt{|\omega|}$. Together with $|s_m| \leq |s_k| \leq c_2\sqrt{|\omega|}$ we have

$$|(r_m, -s_m)|_\infty \leq c_1(2+\sqrt{2})n^{1/4}.$$

On the other hand, if $|r_{m+1}| \geq \frac{\sqrt{|\omega|}}{c_1}$, following the same argument described in the case $|s_m| < |s_{m+1}|$ we also have

$$|(r_{m+2}, -s_{m+2})|_\infty \leq c_1(2+\sqrt{2})n^{1/4}.$$

Therefore,

$$|v_2|_\infty \leq c_1(2+\sqrt{2})n^{1/4}.$$

□

Following Theorem 2 and the argument in Sect. 2.3, we can easily obtain the conclusion.

Theorem 3. *In the 4-dimensional GLV scalar multiplication using the combination of GLV and GLS, the improved twofold Cornacchia-type algorithm will result in a decomposition of any scalar $k \in [1, n)$ into integers k_1, k_2, k_3, k_4 such that*

$$kP = k_1P + k_2\phi(P) + k_3\psi(P) + k_4\psi\phi(P)$$

with

$$\max_i (|k_i|) < 6.82 \left(\sqrt{1 + |r| + s} \right) n^{\frac{1}{4}}.$$

Remark 1. Our proof technique is general and by some modification it can also be applied to improve the upper bound of coefficients given by the original twofold Cornacchia-type algorithm in [12].

3.3 Extension to 4-Dimensional GLS Curves over \mathbb{F}_{p^4}

The twofold Cornacchia-type algorithm can be extended naturally to the 4-dimensional GLV method on GLS curves over \mathbb{F}_{p^4} , which is just the case $k = 4$ in Eq. (1). Let E be an elliptic curve over \mathbb{F}_p , E'' be a quadratic twist of $E(\mathbb{F}_{p^4})$ over \mathbb{F}_{p^4} . Then as described in Eq. (1), the efficient \mathbb{F}_{p^4} -endomorphism φ on E'' satisfying $\varphi^4 + 1 = 0$ on the large prime subgroup $\langle P \rangle$ of $E''(\mathbb{F}_{p^4})$. Hence 4-dimensional GLV method can be implemented on E'' . Moreover, in this case, the twofold Cornacchia-type algorithm can be used for scalar decomposition as well. Let's explain it more specifically.

View φ as an algebraic integer satisfying $X^4 + 1 = 0$. Let $K = \mathbb{Q}(\varphi)$ be the quartic extension over \mathbb{Q} , \mathfrak{o}_K be the ring of integers of K . Since φ is a 8-th root of unity, then $\mathfrak{o}_K = \mathbb{Z}[\varphi]$. Note that φ^2 satisfies $X^2 + 1 = 0$. Write φ^2 as i , then $\mathbb{Z}[\varphi^2] = \mathbb{Z}[i] \subset \mathfrak{o}_K$. We assume that P is of prime order n and $\varphi(P) = \nu P$, then ν is a root of $X^4 + 1 \equiv 0 \pmod{n}$. Denote by \mathfrak{n} the prime ideal lying over n which contains n and $\varphi - \nu$.

First, find out the Gaussian integer $\omega \in \mathbb{Z}[i]$ lying over n with $\omega P = 0$ using Algorithm 2 on the input $(n, \nu^2 \pmod{n})$. Then invoke Algorithm 3 on the input (ω, ν) . Denote the output by (u_1, u_2) where $u_i \in \mathbb{Z}[i] \times \mathbb{Z}[i]$. Following the same argument of Theorem 2 we can obtain that u_1 and u_2 are $\mathbb{Z}[i]$ -linearly independent and

$$|u_1|_\infty \leq \sqrt{2 + \sqrt{2}} n^{1/4}, \quad |u_2|_\infty \leq \sqrt{3}(2 + \sqrt{2}) n^{1/4}.$$

If we assume $u_k = (\alpha_k, \beta_k)$ with $\alpha_k = a_k + ib_k$ and $\beta_k = c_k + id_k$ for $k = 1, 2$, then a short basis of the kernel of the GLV reduction map with respect to $\{1, \varphi, \varphi^2, \varphi^3\}$ is generated by rows of the following matrix

$$\begin{pmatrix} a_1 & c_1 & b_1 & d_1 \\ -b_1 & -d_1 & a_1 & c_1 \\ a_2 & c_2 & b_2 & d_2 \\ -b_2 & -d_2 & a_2 & c_2 \end{pmatrix}.$$

4 Relations of the Two 4-Dimensional GLV Methods on j -invariant 0 Elliptic Curves over \mathbb{F}_{p^2}

In this section, we focus on the elliptic curves with j -invariant 0. We want to explore the relations of the two 4-dimensional GLV methods on this kind of elliptic curves. The first one is put forward in [4] and described in Sect. 2.2, and the second one is put forward by Long and Sica [12] and described in Sect. 2.3.

Note that both two methods create their target curves and endomorphisms by using twists of original curves (especially twists of higher degree). For the general theory of twists, one may refer to [6] or [17, Chap. X]. And twists used to be employed to find pairing-friendly elliptic curves with prime order [1, 14]. By carefully choosing and balancing some parameters of twists, we can obtain the following theorem.

Theorem 4. *For any j -invariant 0 curve E' over \mathbb{F}_{p^2} , if one of the two 4-dimensional GLV methods can be implemented, then the other can be used as well.*

Let \mathbb{F}_p be a prime field with $p \equiv 1 \pmod{3}$, E' be an elliptic curve over \mathbb{F}_{p^2} with j -invariant 0. Fix a primitive element α of the field \mathbb{F}_{p^2} . Up to a \mathbb{F}_{p^2} -isomorphism, E' can be written as

$$E' : y^2 = x^3 + \alpha^l, \text{ for some } l \in \{0, \dots, 5\}.$$

Let $\zeta_3 = (\alpha^{(p+1)})^{\frac{p-1}{3}}$ be a 3-th root of unity in \mathbb{F}_p , then $\rho : (x, y) \mapsto (\zeta_3 x, y)$ is an efficient endomorphism of E' . It is not hard to discover the following two lemmas.

Lemma 3. *If and only if $l = 1, 3$ or 5 , we can find an $A \in \mathbb{F}_p$ and a non-quadratic residue $v \in \mathbb{F}_{p^2}$, such that $\alpha^l = Av^3$.*

Proof. Since $\mathbb{F}_{p^2}^* = \langle \alpha \rangle$, we can write $v = \alpha^m$ for some odd integer m if it exists. Then the existence of such an A and v is equivalent to the existence of an odd integer $m \in [1, p^2 - 1)$ satisfying

$$\frac{\alpha^l}{\alpha^{3m}} \in \mathbb{F}_p.$$

This condition is equivalent to $p^2 - 1 \mid (p - 1)(3m - l)$, namely $p + 1 \mid 3m - l$, since the order of α is $p^2 - 1$. Because $p + 1$ is even and m needs to be odd, it is necessary that l is odd.

Since $p \equiv 1 \pmod{3}$, when $l = 1$, we can take $m = \frac{p+2}{3}$; when $l = 3$, take $m = 1$ and when $l = 5$, take $m = \frac{2(p+1)+5}{3}$. \square

Lemma 4. *If and only if $l = 1, 3$ or 5 , we can find a $B \in \mathbb{F}_p$ and a $u \in \mathbb{F}_{p^2}$ which is neither a quadratic residue nor a cubic residue, such that $\alpha^l = Bu$.*

Proof. The argument is similar to that of Lemma 3. If such a u exists, we can let $u = \alpha^k$ for some integer k with $2 \nmid k$ and $3 \nmid k$. Then the existence of such B and u is equivalent to the existence of an integer $m \in [1, p^2 - 1)$ satisfying $2 \nmid k, 3 \nmid k$ and

$$\frac{\alpha^l}{\alpha^k} \in \mathbb{F}_p.$$

This condition is equivalent to $p+1 \mid k-l$ since the order of α is $p^2 - 1$. Because $p+1$ is even and k needs to be odd, it is necessary that l is odd.

Note that $p \equiv 1 \pmod{3}$. When $l = 1$, we can take $k = 3(p+1) + 1$; when $l = 3$, take $k = 2(p+1) + 3$ and when $l = 5$, take $k = 4(p+1) + 5$. \square

Remark 2. Note that m and k appeared in the proofs are not unique. We evaluate them in this way because we should choose v and u carefully to obtain the equality of endomorphisms explaining the relation of the two 4-GLV methods, which is described in the following Theorem 5.

Assume that we have found an E' as above with almost prime group $E'(\mathbb{F}_{p^2})$ and $l = 1, 3$ or 5 . According to Lemma 3, we can find an $A \in \mathbb{F}_p$ and a non-quadratic residue $v \in \mathbb{F}_{p^2}$ such that $\alpha^l = Av^3$. Let E_1 be the curve over \mathbb{F}_p defined by

$$E_1 : y^2 = x^3 + A.$$

Then obviously E' is a quadratic twist of $E_1(\mathbb{F}_{p^2})$. Denote the twist map $(x, y) \mapsto (vx, v^{3/2}y)$ from E_1 to E' by t_2 , the Frobenius endomorphism of E_1 by π_1 . Now, Long and Sica's 4-dimensional GLV method described in Sect. 2.3 can be applied on E' . Take $\psi = t_2\pi_1 t_2^{-1}$ and $\phi = t_2\pi t_2^{-1}$. Then on the large prime subgroup of $E'(\mathbb{F}_{p^2})$ they satisfy $\psi^2 + 1 = 0$ and $\phi^2 + \phi + 1 = 0$ respectively. Following the twofold Cornacchia-type algorithm we will accomplish the 4-dimensional scalar decomposition.

Let E_2 be the curve over \mathbb{F}_p defined by

$$E_2 : y^2 = x^3 + B.$$

Obviously, E' is a twist of degree 6 of $E_2(\mathbb{F}_{p^2})$. Denote this twist map $(x, y) \mapsto (u^{1/3}x, u^{1/2}y)$ from E_2 to E' by t_6 , the Frobenius endomorphism of E_2 by π_2 . Let $\Psi = t_6\pi_2 t_6^{-1}$. On the large prime subgroup of $E'(\mathbb{F}_{p^2})$ it satisfies $\Psi^4 - \Psi^2 + 1 = 0$. Therefore, we can implement the 4-dimensional GLV scalar multiplication on E' as described in Sect. 2.2 and [7].

Proof (of Theorem 4). This theorem is almost trivial following Lemma 3 and Lemma 4, because they conclude that the condition of choosing E' that is suitable for the two GLV methods are the same, i.e. $l = 1, 3$, or 5 .

Moreover, from the above we see that there is a unified and easy way to construct a j -invariant 0 curve over \mathbb{F}_{p^2} suitable for both 4-dimensional GLV methods, that is, we only need to try α, α^3 and α^5 when given p and α , until the group order is almost prime. This is very helpful for our implementation in Sect. 5. \square

For the first comparison, two GLV curves are chosen from [11], which are $E_1 : y^2 = 4x^3 - 30x - 28$ over \mathbb{F}_p with $\rho^2 + 2 = 0$ and $E_2 : y^2 = x^3 + b$ over \mathbb{F}_p with $p \equiv 1 \pmod{3}$ and $\rho^2 + \rho + 1 = 0$. For some prime p , choose a primitive element α of $\mathbb{F}_{p^2}^*$. For E_1 , we use its twist w.r.t. $\sqrt{\alpha}$ as our target curve, denoted by E'_1 . For E_2 we exploit the way as in Sect. 4. Choosing

curves E_2 (or parameters b) and their twists amount to choosing target curves E'_2 of the form $y^2 = x^3 + \alpha^l$ with $l = 1, 3$ or 5 . We use SEA algorithm [15] to compute $\#E'_i(\mathbb{F}_{p^2})$ and enumerate p within certain range until the group order is prime. We choose three about 127-bit primes for each E_i to implement the original and improved twofold Cornacchia-type algorithm. We care about the ratio of \max_o (resp. \max_m) to $n^{1/4}$ where \max_o (resp. \max_m) denotes the maximum value of the maximum norm of four vectors output by the original (resp. improved) twofold Cornacchia-type algorithm. First, from tables it is certain that the improved decomposition algorithm performs better than the original one in most cases. Second, this performance seems to depend on the GLV model that we choose, since the improvement showed in Table 2 is more evident and consistent than that in Table 1. Finally, we should also recognize that in practice this improvement is rather limited and only by a few bits, so its general practical effect is no more than a couple percentage points.

Table 1. Decomposition on E_1

p	128-bit	127-bit	126-bit
n	254-bit	252-bit	250-bit
$\max_o / n^{1/4}$	3.67	0.98	3.00
$\max_m / n^{1/4}$	0.68	0.98	0.67

Table 2. Decomposition on E_2

p	127-bit	128-bit	129-bit
n	254-bit	255-bit	257-bit
$\max_o / n^{1/4}$	4.64	8.56	4.61
$\max_m / n^{1/4}$	1.08	1.05	1.09

For the second comparison, we find ‘cryptographically good’ j -invariant 0 curves by the way described in Sect. 4. That is for any prime p , we consider $y^2 = x^3 + \alpha^l$ with $l = 1, 3$ or 5 where $\langle \alpha \rangle = \mathbb{F}_{p^2}^*$. We also enumerate p with $p \equiv 1 \pmod{3}$ within certain range until the group order is prime. As showed in Sect. 4, we implement Algorithm 1 and the improved twofold Cornacchia-type algorithm to find a short basis of the kernel of the GLV reduction map w.r.t. $\{1, \phi, \psi, \phi\psi\}$. We choose 15 different curves with prime order. For 11 of them the output of the two decomposition algorithms are identically same. In the remaining 4 cases the length differences of components of four vectors are within 1 bits since the ratios of maximum length are less than 2. In a word, the two decomposition algorithms are same for more than 70% of all cases we have investigated, and in remaining cases the length differences are almost negligible.

6 Conclusion

We refined Longa and Sica's four-dimensional GLV method and analyzed it from two aspects. First we improve the original twofold Cornacchia-type algorithm and show that it possesses a better theoretic upper bound of decomposition coefficients through a neater and shorter proof. Comparison implementations show our improved version performs better in most cases. Second we present relations of the two four-dimensional GLV methods in j -invariant 0 case, and compare our improved twofold Cornacchia-type algorithm with the almost optimal scalar decomposition method using computational implementation. Implementations show that they have almost the same performance, which provide further evidence that the improved version is a sufficiently good scalar decomposition method.

Acknowledgements. We would like to thank Jincheng Zhuang and Chun Guo for their advice on a first version of this work. And we would like to thank the anonymous reviewers for their detailed comments and suggestions. This work is supported by National Natural Science Foundation of China (61379139) and the Strategic Priority Research Program of the Chinese Academy of Sciences, Grant No. XDA06010701.

A Implementation I

We list up tables in this part showing comparable data of the original twofold Cornacchia-type algorithm and the improved one. We chose two GLV curves and considered 3 different primes p for each curve. In the tables, R_1 represents $\max_o/n^{1/4}$ while R_2 represents $\max_m/n^{1/4}$.

$$E_1 : y^2 = 4x^3 - 30x - 28 \text{ with } \rho^2 + 2 = 0$$

p	255211775190703847597530955573826073969
n	16283262548997589981439669766846726243580995059600230271972911887471787246897
Original twofold Cornacchia outputs:	
$v1$	[7673580244184025940, -1568296852280298804, -7673580244184025939, 1568296852280298804]
$v2$	[41504494925480727303, -167904017217468081, 41504494925480727308, -167904017217468080]
$v3$	[7673580244184025939, -1568296852280298804, 7673580244184025940, -1568296852280298804]
$v4$	[-41504494925480727308, 167904017217468080, 41504494925480727303, -167904017217468081]
R_1	3.6741744846002408025240887433477717824
Improved twofold Cornacchia outputs:	
$v1$	[7673580244184025940, -1568296852280298804, -7673580244184025939, 1568296852280298804]
$v2$	[3136593704560597608, 7673580244184025939, 3136593704560597608, 7673580244184025940]
$v3$	[7673580244184025939, -1568296852280298804, 7673580244184025940, -1568296852280298804]
$v4$	[-3136593704560597608, -7673580244184025940, 3136593704560597608, 7673580244184025939]

(continued)

R_2	0.67930167056205598699343592919037215116
p	170141183460469231731687303715884047161
n	7237005577332262213973186563042989258422395530349600540822048403344118204929
Original twofold Cornacchia outputs:	
R_1	0.97789758543585283902428717528465557293
Improved twofold Cornacchia outputs:	
R_2	0.97789758543585283902428717528465557293
p	85070591730234615865843651857942020329
n	180925139433306553493296640760747176355670214223299403819059252318350656377
Original twofold Cornacchia outputs:	
R_1	2.9993369087131711807648390857675716140
Improved twofold Cornacchia outputs:	
R_2	0.67147473658255740348192194949887835750
$E_2: y^2 = x^3 + \alpha^l$ with $\rho^2 + \rho + 1 = 0$	
p	170141183460469231731687303715884022771
n	28948022309329048855892746252171948734834290114750903245851799285340816353501
Original twofold Cornacchia outputs:	
$v1$	[-1, 0, -11594629644441225966, 2528224560705443369]
$v2$	[-5, -1, -60501372782911573199, -1481731401619452490]
$v3$	[11594629644441225966, -2528224560705443369, -1, 0]
$v4$	[60501372782911573199, 1481731401619452490, -5, -1]
R_1	4.6383178294172491273770196206212208904
Improved twofold Cornacchia outputs:	
$v1$	[1, 0, -14122854205146669335, -2528224560705443369]
$v2$	[0, 1, 2528224560705443369, -11594629644441225966]
$v3$	[14122854205146669335, 2528224560705443369, 1, 0]
$v4$	[-2528224560705443369, 11594629644441225966, 0, 1]
R_2	1.0827239688765246962710751584135402862
p	212676479325586539664609129644855136153
n	4523128485832663883733241601901871570337988259090681324905724939638218907073
Original twofold Cornacchia outputs:	
R_1	8.5642929985382088374000139113179346885
Improved twofold Cornacchia outputs:	
R_2	1.0514088258644207810221621225523176688
p	340282366920938463463374607431768216949
n	115792089237316195423570985008687911591087162208992817759013437099099781551273
Original twofold Cornacchia outputs:	
R_2	4.6127000361231510412490970651777643836
Improved twofold Cornacchia outputs:	
R_2	1.0866705232352987405800189002480493540

B Implementation II

The table in this part shows comparable data of the two 4-dimensional scalar decomposition methods on j -invariant 0 curves, the Improved twofold Cornacchia-type algorithm and Algorithm 1 in Sect. 2.2. We considered 15 such curves. In this table, R_1 represents $\max_1/n^{1/4}$ where \max_1 denotes the maximum value of the maximum norm of four vectors output by Algorithm 1, while R_2 represents $\max_m/n^{1/4}$.

p_1	170141183460469231731687303715884008641
n_1	28948022309329048855892746252171943926515682497197240131140526345303172118961
R_1	1.1538418893890212054803449849102612298
R_2	1.6922054648996739026934892690950051293
p_2	170141183460469231731687303715884022771
n_2	28948022309329048855892746252171948734834290114750903245851799285340816353501
R_1	1.0827239688765246962710751584135402862
R_2	1.0827239688765246962710751584135402862
p_3	170141183460469231731687303715884023107
n_3	28948022309329048855892746252171948849171146919057501863641160451816855376557
R_1	1.0790558850578940013923115927424455442
R_2	1.0790558850578940013923115927424455442
p_4	170141183460469231731687303715884025321
n_4	28948022309329048855892746252171949602569744119638481820693392786551599853609
R_1	1.0401348050858175786751875288307348229
R_2	1.0401348050858175786751875288307348229
p_5	170141183460469231731687303715884032929
n_5	28948022309329048855892746252171952191369913468171321827487260495155499120273
R_1	1.1333269230515924468020204150516792233
R_2	1.1333269230515924468020204150516792233
p_6	212676479325586539664609129644855136153
n_6	45231284858326638837332416019018715703337988259090681324905724939638218907073
R_1	1.0514088258644207810221621225523176688
R_2	1.0514088258644207810221621225523176688
p_7	212676479325586539664609129644855146767
n_7	45231284858326638837332416019018720218018417802573857331203271572310044141717
R_1	1.0631742993045731231299194386209802906
R_2	1.0631742993045731231299194386209802906
p_8	212676479325586539664609129644855147811

(continued)

n_8	45231284858326638837332416019018720662601939572979434825619748814731700156669
R_1	1.1546548547005925646516200767383098250
R_2	1.7408774991513931654681153269611596499
p_9	212676479325586539664609129644855149071
n_9	45231284858326638837332416019018721198744499491278384489588201195670328020421
R_1	1.0522088043252422449082166720537976875
R_2	1.0522088043252422449082166720537976875
p_{10}	212676479325586539664609129644855151543
n_{10}	45231284858326638837332416019018722249701332999103696126702882015428082644173
R_1	1.0728298189131269695962120071877785004
R_2	1.0728298189131269695962120071877785004
p_{11}	340282366920938463463374607431768214633
n_{11}	115792089237316195423570985008687910015705821725059268401541167257952106734113
R_1	1.1533461523122356182763890986020527353
R_2	1.7784391593844653239311556780499756686
p_{12}	340282366920938463463374607431768216949
n_{12}	115792089237316195423570985008687911591087162208992817759013437099099781551273
R_1	1.0866705232352987405800189002480493540
R_2	1.0866705232352987405800189002480493540
p_{13}	340282366920938463463374607431768218167
n_{13}	115792089237316195423570985008687912421194511547120121287422632647148162076797
R_1	1.0701593715410208710572988188847176510
R_2	1.0701593715410208710572988188847176510
p_{14}	340282366920938463463374607431768225079
n_{14}	7115792089237316195423570985008687917124536804592137431048915777342117090497813
R_1	1.0993884293669724136434037720100320491
R_2	1.0993884293669724136434037720100320491
p_{15}	340282366920938463463374607431768229507
n_{15}	115792089237316195423570985008687920137464469908874606049864275583449996674837
R_1	1.1542595730165208435416372700910158905
R_2	1.7590231897838901268908695652015694112

References

1. Barreto, P.S.L.M., Naehrig, M.: Pairing-friendly elliptic curves of prime order. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 319–331. Springer, Heidelberg (2006). https://doi.org/10.1007/11693383_22
2. Cohen, H.: A Course in Computational Algebraic Number Theory, vol. 138. Springer Science & Business Media, Heidelberg (2000). <https://doi.org/10.1007/978-3-662-02945-9>
3. Cohen, H., Frey, G., Avanzi, R., Doche, C., Lange, T., Nguyen, K., Vercauteren, F.: Handbook of Elliptic and Hyperelliptic Curve Cryptography. CRC Press, Boca Raton (2005)

4. Galbraith, S.D., Lin, X., 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). https://doi.org/10.1007/978-3-642-01001-9_30
5. 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). https://doi.org/10.1007/3-540-44647-8_11
6. Hess, F., Smart, N.P., Vercauteren, F.: The eta pairing revisited. *IEEE Trans. Inf. Theory* **52**(10), 4595–4602 (2006)
7. Zhi, H., Longa, P., Maozhi, X.: Implementing the 4-dimensional GLV method on GLS elliptic curves with j -invariant 0. *Des. Codes Crypt.* **63**(3), 331–343 (2012)
8. Iijima, T., Matsuo, K., Chao, J., Tsujii, S.: Construction of Frobenius maps of twists elliptic curves and its application to elliptic scalar multiplication. In: Proceedings of SCIS 2002, pp. 699–702. IEICE, Japan (2002)
9. Janusz, G.J.: *Algebraic Number Fields*, vol. 7. American Mathematical Society (1996)
10. Koblitz, N.: CM-curves with good cryptographic properties. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 279–287. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_22
11. Longa, P., Sica, F.: Four-Dimensional Gallant-Lambert-Vanstone scalar multiplication. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 718–739. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34961-4_43
12. Longa, P., Sica, F.: Four-dimensional Gallant-Lambert-Vanstone scalar multiplication. *J. Cryptol.* **27**(2), 248–283 (2014)
13. 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). https://doi.org/10.1007/3-540-45664-3_23
14. Pereira, G.C.C.F., Simpício, M.A., Naehrig, M., Barreto, P.S.L.M.: A family of implementation-friendly BN elliptic curves. *J. Syst. Softw.* **84**(8), 1319–1326 (2011)
15. Schoof, R.: Counting points on elliptic curves over finite fields. *J. Théor. Nombres Bordeaux* **7**(1), 219–254 (1995)
16. Sica, F., Ciet, M., Quisquater, J.-J.: Analysis of the Gallant-Lambert-Vanstone method based on efficient endomorphisms: elliptic and hyperelliptic curves. In: Nyberg, K., Heys, H. (eds.) SAC 2002. LNCS, vol. 2595, pp. 21–36. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36492-7_3
17. Silverman, J.H.: *The Arithmetic of Elliptic Curves*, vol. 106. Springer, New York (2009). <https://doi.org/10.1007/978-0-387-09494-6>
18. Zhou, Z., Zhi, H., Maozhi, X., Song, W.: Efficient 3-dimensional GLV method for faster point multiplication on some GLS elliptic curves. *Inf. Process. Lett.* **110**(22), 1003–1006 (2010)

Selected Areas in Cryptography – SAC 2017
24th International Conference, Ottawa, ON, Canada,
August 16-18, 2017, Revised Selected Papers
Adams, C.; Camenisch, J. (Eds.)
2018, XI, 459 p. 70 illus., Softcover
ISBN: 978-3-319-72564-2