ANALYSIS ON THE ELLIPTIC SCALAR MULTIPLICATION
USING INTEGER SUB-DECOMPOSITION METHOD

Ruma Kareem K. Ajeena$^1$, Hailiza Kamarulhaili$^2$

$^{1,2}$School of Mathematical Sciences
University Sains Malaysia
11800 USM, Penang, MALAYSIA

Abstract: This study proposes a new approach called, integer sub-decomposition (ISD), to compute any multiple $kP$ of a point $P$ of order $n$ lying on an elliptic curve. Our method depends, in computations, on fast endomorphisms $\psi_1$ and $\psi_2$ of elliptic curve over prime fields. The integer sub-decomposition to multiple $kP$, when the value of $k$ is decomposed into two values $k_1$ and $k_2$, where both values or one of them is not bounded by $\pm C \sqrt{n}$, is illustrated in the following formula:

$$kP = k_{11}P + k_{12}[\lambda_1]P + k_{21}P + k_{22}[\lambda_2]P = k_{11}P + k_{12}\psi_1(P) + k_{21}P + k_{22}\psi_2(P).$$

where $-C\sqrt{n} < k_{11}, k_{12}, k_{21}, k_{22} < C\sqrt{n}$. The integers $k_{11}, k_{12}, k_{21}$ and $k_{22}$ are computed by solving a closest vector problem in lattice. Consequently, as for this sub-decomposition, we have managed to increase the percentage of a successful computation of $kP$. Moreover, the gap in the proof of the bound of kernel $K$ vectors of the reduction map $T: (a,b) \rightarrow a + \lambda b (mod \ n)$ on ISD method will be filled through the analysis of the multiplier $k$, using two fast endomorphisms with minimal polynomials $X^2 + rX_i + s_i$ for $i = 1, 2, 3$. In particular, we prove an integer sub-decomposition (ISD) with explicit constant

$$kP = k_{11}P + k_{12}\psi_1(P) + k_{21}P + k_{22}\psi_2(P),$$

with

$max\{|k_{11}|, |k_{12}|\}$ and $max\{|k_{21}|, |k_{22}|\} < \sqrt{1 + |r_i| + s_i \sqrt{n}}$, for $i = 1, 2, 3$. 

Received: May 3, 2013

$^8$Correspondence author
AMS Subject Classification: 06-xx, 18B35, 06Bxx, 03G10

Key Words: elliptic curves, fast performance, efficiently-computable endomorphisms, integer sub-decomposition

1. Introduction

The attractive features of elliptic curves history awarded it studying by mathematicians over a hundred of years to solve a variety of problems. The entry of these curves into cryptography independently by Neal Koblitz [1] and Victor Miller [2] in 1985 who suggested elliptic curve public key cryptosystems. The elliptic curves performance has active importance in the security level as a traditional asymmetric cryptosystem, such as RSA [3],[4]. The fundamental step of elliptic curve cryptosystems is to compute elliptic curve scalar multiplication $kP$ for a point $P$ which has a large prime order $n$. To accomplish this end, various methods have been innovated, adopting on elliptic curves $E$ over finite fields[5],[6],[7] and [8]. A group of methods cleverly employs a distinguished endomorphism $\psi \in \text{End}(E)$ to split a large computation into a sequence of cheaper ones, so that the overall computational cost will be lowered [3].

Recently, Gallant, Lambert and Vanstone [9],[10],[11] used such a technique that, contrary to the previous ones, also applied to curves defined over large prime fields. Their method uses an efficiently computable endomorphism $\psi \in \text{End}(E)$ to rewrite $kP$ as

$$kP = k_1 P + k_2 \psi(P), \text{ with } \max\{|k_1|, |k_2|\} = O(\sqrt{n}). \quad (1.1)$$

Their key point is an algorithm, that will be called the GLV method, which inputs integers $n$ and $\lambda \in [1, n-1]$ and produces for any $k \ (mod \ n)$, two residues $k_1$ and $k_2 \ (mod \ n)$ such that

$$k = k_1 + \lambda k_2 \ (mod \ n). \quad (1.2)$$

On the other hand, they do not succeed to give an upper bound on $\max\{|k_1|, |k_2|\}$ and they give a guided estimation shows that this must be $O(\sqrt{n})$, but it does not demonstrate any estimation of the concerned constant in their study too. The first appearance for an upper bound was in [12] where a different method was used. Moreover, we were perceived of another usage to the GLV method [11] where a necessary condition is innovated to be sure that the constant in $O(\sqrt{n})$ is 1 in equation (1.1). This algorithm was the alternative to the presented GLV method.
Improving the GLV algorithm would be to find the decomposition
\[ kP = k_1 P + k_2 \psi(P) + \ldots + k_d \psi^{d-1}(P), \]
with \[ \max\{|k_i|\} = O(n^{\frac{1}{d}}). \] (1.3)
In general using the GLV paradigm in equation (1.3) is not possible, since the powers \( \psi^i \) are independent over \( \mathbb{Z} \) only when \( i < 2 \). However, a class of \( \psi^i's \) for which such a decomposition exists is found as in [13].

Starting with analyzing the GLV method of Gallant, Lambert and Vanstone, our study uses two fast endomorphisms with minimal polynomials \( X^2 + r_i X + s_i \), for \( i = 1, 2, 3 \) to compute any multiple \( kP \) of a point \( P \) of order \( n \) lying on an elliptic curve. When both values or one of them is not bounded by \( \pm \sqrt{1 + |r_i| + s_i \sqrt{n}}, i = 1, 2, 3 \), the value \( k \) is then decomposed into the values \( k_1 \) and \( k_2 \). The sub-decomposition from \( k = k_1 + k_2 \lambda (mod n) \) is shown clearly as follows:

\[ k = k_1 + k_{12} \lambda_1 (mod n) \quad \text{and} \quad k = k_{21} + k_{22} \lambda_2 (mod n). \] (1.4)

We calculate, in particular, the integer sub-decomposition (ISD) as follows:

\[ kP = k_{11} P + k_{12} [\lambda_1] P + k_{21} P + k_{22} [\lambda_2] P \]
\[ = k_{11} P + k_{12} \psi_1(P) + k_{21} P + k_{22} \psi_2(P). \] (1.5)

where \( -\sqrt{1 + |r_i| + s_i \sqrt{n}} < k_{11}, k_{12}, k_{21}, k_{22} < \sqrt{1 + |r_i| + s_i \sqrt{n}}, i = 1, 2, 3. \)

A proof is supplied, in this paper, that the ISD algorithm works by producing a required upper bound of the kernel \( K \) vectors of the reduction map \( T : (a, b) \rightarrow a + \lambda b (mod n). \) We prove, in particular, an integer sub-decomposition with explicit constant

\[ kP = k_{11} P + k_{12} \psi_1(P) + k_{21} P + k_{22} \psi_2(P), \quad \text{with} \]
\[ \max \left\{ \frac{|k_{11}|, |k_{12}|}{|k_{21}|, |k_{22}|} \right\} < \sqrt{1 + |r_i| + s_i \sqrt{n}}, \quad \text{for} \; i = 1, 2, 3. \] (1.6)

The outline of this paper shows: Section 2 gives a summary of the Mathematical background to clarify elliptic curve \( E \) over prime field and endomorphisms on it. Section 3 reviews the procedure of scalar multiplication using a GLV method and fills the logical gap of this method. Section 4 shows the value of the bound \( C \) of kernel vectors of the reduction map \( T \) in GLV method. Section 5 presents a new method called, integer sub-decomposition (ISD), to compute scalar multiplication depending on the sub-decomposition and demonstrates the filling up of the logical gap of the ISD method. Section 6 displays the Mathematical proofs which help us find the value of the bound \( C \) of kernel vectors of the reduction map \( T \) on ISD method. Finally, Section 7 draws the concluding remarks.
2. Mathematical Background

2.1. Elliptic Curves over Prime Fields

**Definition 2.1.** Let \( p \neq 2, 3 \). An elliptic curve \( E(F_p) \) over \( F_p \), is defined by an equation of the form [14]:

\[
E : Y^2 = X^3 + AX + B \pmod{p},
\]

where \( A, B \in F_p \). The curve \( E \) is said to be non-singular if it has no double zeroes, that means the discriminant \( D_E = 4A^3 + 27B^2 \neq 0 \pmod{p} \).

**Definition 2.2.** Let \( E(F_p) \) be an elliptic curve defined in equation (2.1) over the field \( F_p \), \( P = (x_P, y_P) \) and \( Q = (x_Q, y_Q) \) two points on \( E \) such that \( P, Q \neq \infty \). We define \( P + Q = R = (x_R, y_R) \) as follows [14] and [15]:

\[
\mu \equiv \begin{cases} 
\frac{y_Q - y_P}{x_Q - x_P} \pmod{p}, & \text{if } P \neq Q \\
\frac{3x_P^2 + A}{2y_P} \pmod{p}, & \text{if } P = Q
\end{cases}
\]

\[
x_R \equiv \lambda^2 - x_P - x_Q \pmod{p}
\]

\[
y_R \equiv \lambda(x_P - x_R) - y_P \pmod{p}.
\]

(2.2)

A special case when \( P = -Q \) then \( P + Q = \infty \).

2.2. Endomorphisms of Elliptic Curve over Prime Fields

Assume that \( E \) is an elliptic curve defined over the finite field \( F_p \). The point at infinity is denoted by \( O_E \). The set of \( F_p \)-rational points on \( E \) forms the group \( E(F_p) \). A rational map \( \psi : E \to E \) satisfies \( \psi(O_E) = O_E \) dubbed an endomorphism of \( E \). The endomorphism \( \psi \) will be defined over \( F_q \) where \( q = p^n \), if the rational map is defined over \( F_q \). Therefore, clearly, for any \( n \geq 1 \), \( \psi \) is a group homomorphism of \( E(F_p) \) and also of \( E(F_q) \) [3] and [15].

**Definition 2.3.** The endomorphism of elliptic curve \( E \) defined over \( F_q \) is the \( m \)-multiplication map \([m] : E \to E \) defined by

\[
P \to mP
\]

(2.3)

for each \( m \in \mathbb{Z} \). The negation map \([-1] : E \to E \) defined by \( P \to -P \) is a special case from \( m \)-multiplication map [3].
Theorem 2.4. (Hasse Theorem). Let $E$ be an elliptic curve over a finite field $F_p$ [3]. Then, the order of $E(F_p)$ satisfies

$$|p + 1 - \#E(F_p)| \leq 2\sqrt{p}.$$ (2.4)

Definition 2.5. The rectangle norm [4] of $(x, y)$ is defined by $\max\{|x|, |y|\}$. We denote it by $|(x, y)|$.

3. Bridging the Logical Gaps of the GLV Algorithm

The Gallant-Lambert-Vanstone’s computation method [9] will be briefly summarized in this part. Assume that $F_q$ is a finite field. The point $P = (x, y)$ is a point on an elliptic curve $E$ defined over a field $F_q$, with order $n$ such that the cofactor $h = \#E(F_q)/n$ is small, say $h \leq 4$. The characteristic polynomial of a non trivial endomorphism $\psi$ defined over $F_q$ takes the form $X^2 + rX + s$, where $r$ and $s$ are actually small fixed integers. By the Hasse bound, since $n$ is large, then $\psi(P) = \lambda P$ for some $\lambda \in [1, n - 1]$. As a matter of fact, there is only one copy of $\mathbb{Z}/n\mathbb{Z}$ inside $E(F_p)$ and $\psi(P)$ has also an order dividing $n$. Moreover, the parameter $\lambda$ is a root of $X^2 + rX + s$ modulo $n$, where the case $\lambda = 0$ is excluded from all cases.

The definition of the group homomorphism $T$ as follows:

$$T : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}/n$$

$$(i, j) \to i + \lambda j \ (mod \ n)$$ (3.1)

represents a pivotal point in GLV method. Let $K = \ker T$. Obviously, $K$ is a sublattice of $\mathbb{Z} \times \mathbb{Z}$. And let $v_1$ and $v_2$ be two linearly independent vectors of $K$ satisfying $\max\{|v_1|, |v_2|\} < M$ for some $M > 0$, where $|\cdot|$ indicates to any metric norm. Consider

$$(k, 0) = \beta_1 v_1 + \beta_2 v_2,$$ (3.2)

where $\beta_i \in \mathbb{Q}$. Then the rounding of $\beta_i$ to the nearest integer is $b_i = [\beta_i] = [\beta_i + 1/2]$ and suppose that $v = b_1 v_1 + b_2 v_2$. Observe that $v \in K$ and that $u = (k, 0) - v$ is short. The triangle inequality gives us the following fact

$$|u_0| \leq \left| \frac{v_1 + v_2}{2} \right| < M.$$ (3.3)

If one puts

$$(k_1, k_2) = u_0,$$ (3.4)
then from equation (1.2), one can have

\[ kP = k_1P + k_2\psi(P), \text{ with } |(k_1, k_2)| < M. \] (3.5)

In this way, it is fundamental in the GLV method that \( M \) should be as small as possible, taking into consideration that by a simple counting argument we must have \( M \geq \sqrt{n}/2 \). Gallant et. al, then, claim without proof the fact that

\[ M \leq C\sqrt{n}, \] (3.6)

for some constant \( C \) [4].

4. A Value for \( C \) in the GLV Algorithm

Remember that the extended Euclidean algorithm applied to \( n \) and \( \lambda \) is used by the GLV algorithm to generate a sequence of relations

\[ s_ln + t_l\lambda = r_l, \text{ for } l = 0, 1, 2, \ldots, \] (4.1)

where \(|s_l| < |s_{l+1}|\) for \( l \geq 1 \), \(|t_l| < |t_{l+1}|\) and \( r_l > r_{l+1} \geq 0 \) for \( l \geq 0 \). Also, we have from Lemma (1-iv) in [9]:

\[ r_l|t_{l+1}| + r_{l+1}|t_l| = n \text{ for all } l \geq 0. \] (4.2)

The index \( m \) of the GLV algorithm defines as the largest integer for which \( r_m > \sqrt{n} \). Then (4.2) with \( l = m \) gives that \(|t_{m+1}| < \sqrt{n}\), so that the kernel vector \( v = (r_{m+1}, -t_{m+1}) \) has rectangle norm bounded by \( \sqrt{n} \). The GLV algorithm then sets \( v_2 \) to be the shorter between \((r_m, -t_m)\) and \((r_{m+2}, -t_{m+2})\), but does not give any estimate on the size of \( v_2 \). In reality, Gallant et al. claimed that

\[ \min(|(r_m, -t_m)|, |(r_{m+2}, -t_{m+2})|) \leq C\sqrt{n}. \] (4.3)

This will be explained with an explicit value of \( C \) [4]. Let \( \lambda \) and \( \mu \) be the zeros of \( X^2 + rX + s \pmod{n} \). For any \((x, y) \in \mathcal{K} - \{(0, 0)\}\), one can have \( 0 \equiv (x + \lambda y)(x + \mu y) \equiv x^2 - rxy + sy^2 \pmod{n} \), hence, since \( X^2 + rX + s \) is irreducible in \( \mathbb{Z}[X] \), one must have \( x^2 - rxy + sy^2 \geq n \). Certainly, this leads to

\[ \max(|x|, |y|) \geq \sqrt{n / (1 + |r| + s)}. \] (4.4)

In particular,

\[ |(r_{m+1}, -t_{m+1})| \geq \sqrt{n} / \sqrt{1 + |r| + s}. \] (4.5)
There are two cases of the components of the vector $v$:

**Case 1.** If $|t_{m+1}| \geq \sqrt{n}/\sqrt{1+|r|+s}$. Then, the equation (4.2) with $l = m$ produces that $r_m < \sqrt{1+|r|+s\sqrt{n}}$, hence

$$|(r_m,-t_m)| < \sqrt{1+|r|+s\sqrt{n}}. \quad (4.6)$$

**Case 2.** If $r_{m+1} \geq \sqrt{n}/\sqrt{1+|r|+s}$. The same equation (4.2) with $l = m+1$ implies that $|t_{m+2}| < \sqrt{1+|r|+s\sqrt{n}}$, hence

$$|(r_{m+2},-t_{m+2})| < \sqrt{1+|r|+s\sqrt{n}}. \quad (4.7)$$

**Theorem 4.1.** An admissible value [4] for $C$ is

$$C = \sqrt{1+|r|+s}. \quad (4.8)$$

In particular, the decomposition of any multiple $kP$ can take the form

$$kP = k_1P + k_2\psi(P), \text{ with } \max\{|k_1|,|k_2|\} < \sqrt{1+|r|+s\sqrt{n}}.$$

5. Bridging the Logical Gaps of the (ISDA) Integer Sub-Decomposition Algorithm

The integer sub-decomposition computation method can be interpreted through this section as follows. Assume that $F_q$ is a finite field. The point $P = (x,y)$ is a point on an elliptic curve $E$ defined over a field $F_q$, with order $n$ such that the cofactor $h = \#E(F_q)/n$ is small, say $h \leq 4$. The characteristic polynomials of non trivial endomorphisms $\psi_1$ and $\psi_2$ defined over $F_q$ take the form $X^2 + r_iX + s_i$, where $r_i$ and $s_i$ are actually small fixed integers and $i = 1,2,3$. By the Hasse bound, since $n$ is large, then, $\psi_1(P) = \lambda_1 P$ and $\psi_2(P) = \lambda_2 P$ for some $\lambda_1$ and $\lambda_2 \in [1,n-1]$. Actually, there is only one copy of $Z/n$ inside $E(F_q)$ and $\psi_1(P)$ and $\psi_2(P)$ have also an order dividing $n$. Furthermore, the parameters $\lambda_j$, $j = 0,1,2$, are roots of $X^2 + r_iX + s_i$ modulo $n$, $i = 1,2,3$ and the cases $\lambda_1$ and $\lambda_2 = 0$ are excluded from all cases.

A fundamental role of the ISD method lies in the definition of the group homomorphism

$$T : Z \times Z \to Z/n$$

$$(a,b) \to a + \lambda_j b \ (mod \ n) \quad (5.1)$$
where \( j = 0, 1, 2 \). Let \( \mathcal{K} = \text{ker}T \). Clearly, the \( \mathcal{K} \) is a sublattice \( Z \times Z \). Let \( v_1, v_2, v_3, v_4, v_5 \) and \( v_6 \) be linearly independent vectors of \( \mathcal{K} \) and integer lattice points that satisfy

\[
\max \left\{ \frac{|v_1|}{|v_2|}, \frac{|v_3|}{|v_4|}, \frac{|v_5|}{|v_6|} \right\} < M
\]

for some \( M > 0 \), where \( |\cdot| \) denotes to any metric norm. These points can be computed by solving the closest vector problem in a lattice which is embodied in using a GLV generator algorithm in [3] to compute \( \{v_1, v_2\} \) and our modified ISD generators algorithm (1) in Appendix (A) to compute \( \{v_3, v_4\} \) and \( \{v_5, v_6\} \).

Express

\[
\begin{align*}
(k, 0) &= \beta_1 v_1 + \beta_2 v_2, \\
(k_1, 0) &= \beta_3 v_3 + \beta_4 v_4, \\
(k_2, 0) &= \beta_5 v_5 + \beta_6 v_6,
\end{align*}
\]

where \( \beta_i \in Q, \ i = 1, 2, 3, 4, 5, 6 \). Then the rounding of \( \beta_i \) to the nearest integer \( b_i = [\beta_i] = [\beta_i + 1/2] \) and let

\[
\begin{align*}
v &= b_1 v_1 + b_2 v_2, \\
v' &= b_3 v_3 + b_4 v_4, \\
v'' &= b_5 v_5 + b_6 v_6.
\end{align*}
\]

Observe that \( v, v', v'' \in \mathcal{K} \) and these

\[
\begin{align*}
u_0 &= (k, 0) - v, \\
u_1 &= (k_1, 0) - v', \\
u_2 &= (k_2, 0) - v''.
\end{align*}
\]

are short. By the triangle inequality, one can obtain

\[
\begin{align*}
|u_0| &\leq \frac{v_1 + v_2}{2} \\
|u_1| &\leq \frac{v_3 + v_4}{2} \\
|u_2| &\leq \frac{v_5 + v_6}{2}
\end{align*}
\]

\[
\left\{ \begin{array}{l}
|u_0| \\
|u_1| \\
|u_2|
\end{array} \right\} < M. \tag{5.2}
\]

If one sets

\[
(k_1, k_2) = u_0, \tag{5.3}
\]

then

\[
k = k_1 + (k_2 \lambda) \pmod{n} \tag{5.4}
\]
where $k_1$ and $k_2$ are integers resulting from the decomposition of the multiplier $k$ by using the balanced length-two representation of a multiplier algorithm [3]. The formula in the equation (5.4) is equivalent to

$$k = k_1 + k_2 \pmod n,$$

with $|(k_1, k_2)| > M$. \hspace{1cm} (5.5)

Thus, the main idea of ISD method is to sub-decompose the values $k_1$ and $k_2$ when both values or one of them is not bounded by $\pm M$. Therefore, we decompose $k_1$ and $k_2'$ again into integers $k_{11}, k_{12}, k_{21}$ and $k_{22}$ which means that the sub-decomposition of $k$ by applying the modified balanced length-two representation of a sub-decomposition multiplier algorithm (2), in Appendix (B), as follows:

$$k = k_{11} + k_{12}\lambda_1 + k_{21} + k_{22}\lambda_2 \pmod n \hspace{1cm} (5.6)$$

with $-M < k_{11}, k_{12}, k_{21}, k_{22} < M$ from any ISD generators $\{v_3, v_4\}$ and $\{v_5, v_6\}$. Assume that one puts

$$u_1 = (k_{11}, k_{12}) \quad \text{and} \quad u_2 = (k_{21}, k_{22}), \hspace{1cm} (5.7)$$

then

$$k_1 = k_{11} + k_{12}\lambda_1 \pmod n \quad \text{and} \quad k_2 = k_{21} + k_{22}\lambda_2 \pmod n \hspace{1cm} (5.8)$$

which are equivalent to

$$k_1P = k_{11}P + k_{12}\psi_1(P) \quad \text{and} \quad k_2P = k_{21}P + k_{22}\psi_2(P). \hspace{1cm} (5.9)$$

That means

$$kP = k_{11}P + k_{12}\psi_1(P) + k_{21}P + k_{22}\psi_2(P), \hspace{1cm} (5.10)$$

with

$$|(k_{11}, k_{12})| \quad \text{and} \quad |(k_{21}, k_{22})| < M. \hspace{1cm} (5.11)$$

The fast performance of scalar multiplication $kP$ in equation (5.11) determines our modification, in algorithm (3), in Appendix (C), that uses in computations two endomorphisms $\psi_1(P) = [\lambda_1]P$ and $\psi_2(P) = [\lambda_2]P$, where $P \in E(F_p)$, $\lambda_1, \lambda_2 \in [1, n-1]$ and $\lambda_1 \neq \pm \lambda_2$. Basically, $M$ is as small as possible in the ISD method and we must have $M \geq \sqrt{n}/2$. The integer sub-decomposition method, ISD will help increase 50% more successful rate as compared to the GLV method in the computation of the $kP$. See algorithm (4) in Appendix (D).
6. A Value for $C$ in an Integer Subdecomposition Method (ISDM)

In this section, we overcome on the omission which applied to ISD method that focuses on the sub-decomposition of integer $k$ when the values were decomposed $k_1$ and $k_2$ are not bounded by $\pm M$. The using of the extended Euclidean algorithm in the ISD algorithm utilized to $n$ and $\lambda_0$ firstly to generate a sequence of relations in the equation (4.1). Also, we had the condition in equation (4.2) from Lemma (1-iv) in [9]. The GLV algorithm used in ISD method defines the index $m$ as the largest integer for which $r_m > \sqrt{n}$. Then, the equation (4.2) with $l = m$ gives that $|t_{m+1}| < \sqrt{n}$, so that the vector $v_1 = (r_{m+1}, -t_{m+1})$ in $\mathcal{K}$, has a rectangle norm bounded by $M$. The modified GLV algorithm, then, sets $v_2$ to be the shorter between $(r_m, -t_m)$ and $(r_{m+2}, -t_{m+2})$ and satisfies the conditions in Lemmas (1) and (2) in [11] such that

$$\min(|(r_m, -t_m)|, |(r_{m+2}, -t_{m+2})|) \leq C\sqrt{n},$$

where $gcd(r_m, -t_m)=1$ and $gcd(r_{m+2}, -t_{m+2})=1$, with an explicit value of $C = 1$.

In similar way, we can set the vectors $v_4$ and $v_6$ by depending on $v_3$ and $v_5$ as follows

$$\min \left\{ |(\bar{r}_m, -\bar{t}_m)|, |(\bar{r}_{m+2}, -\bar{t}_{m+2})| \right\} \leq C\sqrt{n},$$

(6.1)

where

$$gcd \left\{ \begin{array}{c}
(\bar{r}_m, -\bar{t}_m) \\
(\bar{r}_{m+2}, -\bar{t}_{m+2}) \\
(\hat{r}_m, -\hat{t}_m) \\
(\hat{r}_{m+2}, -\hat{t}_{m+2})
\end{array} \right\} = 1,$$

with an explicit value $C = 1$.

Now, one can show the explicit value of $C$ when this value greater than 1 as follows. Let $\lambda_j$ and $\mu_j \in [1, n - 1]$, $j = 0, 1, 2$, be the zeros of $X^2 + r_iX + s_i \pmod n$, $i = 1, 2, 3$. For any $(x, y) \in \mathcal{K} - \{(0, 0)\}$, then

$$0 \equiv (x + \lambda_jy)(x + \mu_jy) \equiv x^2 - r_ixy + s_iy^2 \pmod n,$$

(6.2)

hence, since $X^2 + r_iX + s_i$ is irreducible in $\mathbb{Z}[X]$, one must have

$$x^2 - r_ixy + s_iy^2 \geq n.$$  \hspace{1cm} (6.3)

This certainly leads to

$$\max(|x|, |y|) \geq \sqrt{\frac{n}{1 + |r_i| + s_i}}, \quad i = 1, 2, 3.$$  \hspace{1cm} (6.4)
In particular,
\[
\begin{align*}
&\left\{ \begin{array}{c}
| (r_{m+1}, -t_{m+1}) | \\
| (\hat{r}_{m+1}, -\hat{t}_{m+1}) | \\
| (\check{r}_{m+1}, -\check{t}_{m+1}) | \\
\end{array} \right\} \geq \sqrt{n} / \sqrt{1 + |r_i| + s_i}, \text{ where } i = 1, 2, 3. \\
(6.5)
\end{align*}
\]

**Theorem 6.1.** Suppose that
\[
\left\{ \begin{array}{c}
| t_{m+1} | \\
| \check{t}_{m+1} | \\
| \hat{t}_{m+1} | \\
\end{array} \right\} \geq \sqrt{n} / \sqrt{1 + |r_i| + s_i}, \text{ where } i = 1, 2, 3.
\]

Then, the equation (4.2) with \( l = m \) implies that
\[
\left\{ \begin{array}{c}
r_{m} \\
\check{r}_{m} \\
\hat{r}_{m} \\
\end{array} \right\} \geq \sqrt{n} / \sqrt{1 + |r_i| + s_i}, \text{ where } i = 1, 2, 3.
\]

hence,
\[
\left\{ \begin{array}{c}
| (r_m, -t_m) | \\
| (\check{r}_m, -\check{t}_m) | \\
| (\hat{r}_m, -\hat{t}_m) | \\
\end{array} \right\} \geq \sqrt{n} / \sqrt{1 + |r_i| + s_i}, \text{ where } i = 1, 2, 3. \\
(6.6)
\]

**Proof.** From the conditions in equation (4.1) \(|t_l| < |t_{l+1}|, r_l > r_{l+1} \geq 0\) and in equation (4.2), \( r_l |t_{l+1}| + r_{l+1} |t_l| = n \) for all \( l \geq 0 \).

\( l = t_l |t_{l+1}| + r_{l+1} |t_l| > n \) \( r_l |t_{l+1}| + r_{l+1} |t_l| = r_l (|t_{l+1}| + |t_l|) \).

That is, \( n > r_l (|t_{l+1}| + |t_l|) \). Since \( |t_{l+1}| > |t_l| \)

\( n = r_l (|t_{l+1}| + |t_l|) = 2 r_l |t_{l+1}| \)

\( \frac{n}{2} > r_l (|t_{l+1}|) \). From the hypothesis \( |t_{m+1}| \geq \sqrt{n} / \sqrt{1 + |r_i| + s_i}, i = 1, 2, 3, \)

\( \frac{n}{2} > r_i \sqrt{\frac{n}{2} \sqrt{1 + |r_i| + s_i}} \)

\( \frac{n}{2} \sqrt{\frac{n}{2} \sqrt{1 + |r_i| + s_i}} > r_i \)

\( r_i < \sqrt{\frac{n}{2} \sqrt{1 + |r_i| + s_i}} \) \(< \sqrt{n} \sqrt{1 + |r_i| + s_i}, \text{ hence,} \)

\( |(r_m, -t_m)| < \sqrt{1 + |r_i| + s_i} \sqrt{n}, \text{ when } i = 1. \)

In the same way, we can find
\[
\left\{ \begin{array}{c}
| (\check{r}_m, -\check{t}_m) | \\
| (\hat{r}_m, -\hat{t}_m) | \\
\end{array} \right\} < \sqrt{1 + |r_i| + s_i} \sqrt{n}, \text{ where } i = 2, 3.
\]

\[\square\]
Theorem 6.2. Assume that
\[
\begin{cases}
    r_{m+1} \\
    \tilde{r}_{m+1} \\
\end{cases} \geq \sqrt{n}/\sqrt{1 + |r_i| + s_i}, \; i = 1, 2, 3.
\]

The same equation (4.2) with \( l = m + 1 \) implies that
\[
\begin{cases}
    |t_{m+1}| \\
    |\tilde{t}_{m+1}| \\
\end{cases} < \sqrt{1 + |r_i| + s_i} \sqrt{n}, \; i = 1, 2, 3.
\]

hence,
\[
\begin{cases}
    |(r_{m+2}, -t_{m+2})| \\
    |(\tilde{r}_{m+2}, -\tilde{t}_{m+2})| \\
\end{cases} < \sqrt{1 + |r_i| + s_i} \sqrt{n}, \; i = 1, 2, 3. \tag{6.7}
\]

Proof. From the conditions in equation (4.1) \(|t_l| < |t_{l+1}|, r_l > r_{l+1} \geq 0 \) and in equation (4.2), \( r_l|t_{l+1}| + r_{l+1}|t_l| = n \) for all \( l \geq 0 \).
\[
\Rightarrow n = r_l|t_{l+1}| + r_{l+1}|t_l| > r_l|t_{l+1}| + r_{l+1}|t_{l+1}| = |t_{l+1}|(r_l + r_{l+1}).
\]
That is, \( n > |t_{l+1}|(r_l + r_{l+1}) \). Since \( r_l > r_{l+1} \geq 0 \),
\[
\Rightarrow n > |t_{l+1}|(r_l + r_{l+1}) = 2r_{l+1}|t_{l+1}|.
\]
\[
\Rightarrow \frac{n}{2} > r_{l+1}|t_{l+1}|. \; \text{From the hypothesis} \; r_{m+1} \geq \sqrt{n}/\sqrt{1 + |r_i| + s_i}, \; i = 1, 2, 3.
\]
\[
\Rightarrow \frac{n}{2} > \frac{\sqrt{n}}{\sqrt{1 + |r_i| + s_i}} |t_{l+1}|,
\]
\[
\Rightarrow \frac{\sqrt{n}}{\sqrt{1 + |r_i| + s_i}} > |t_{l+1}|,
\]
\[
\Rightarrow |t_{l+1}| < \frac{\sqrt{n}}{\sqrt{1 + |r_i| + s_i}} < \sqrt{1 + |r_i| + s_i} \sqrt{n}. \; \text{Since} \; l = m + 1,
\]
\[
\Rightarrow |t_{l+2}| < \sqrt{1 + |r_i| + s_i} \sqrt{n}, \; i = 1.
\]

In similar way, we can prove
\[
\begin{cases}
    |(\tilde{r}_{m+2}, -\tilde{t}_{m+2})| \\
    |(\tilde{r}_{m+2}, -\tilde{t}_{m+2})| \\
\end{cases} < \sqrt{1 + |r_i| + s_i} \sqrt{n}, \; i = 2, 3.
\]

Hence,
\[
\begin{cases}
    |(r_{m+2}, -t_{m+2})| \\
    |(\tilde{r}_{m+2}, -\tilde{t}_{m+2})| \\
\end{cases} < \sqrt{1 + |r_i| + s_i} \sqrt{n}, \; i = 1, 2, 3.
\]

\[\square\]

Theorem 6.3. An admissible value for \( C \) is
\[
C = \sqrt{1 + |r_i| + s_i}, \; i = 1, 2, 3. \tag{6.8}
\]
In particular, any multiple \( kP \) can be decomposed as in equation (5.10) with

\[
\max \left\{ \begin{array}{l}
\{|k_1|, |k_2|\} < \sqrt{1 + |r_1| + s_1 \sqrt{n}}, \\
\{|k_{11}|, |k_{12}|\} < \sqrt{1 + |r_2| + s_2 \sqrt{n}}, \\
\{|k_{21}|, |k_{22}|\} < \sqrt{1 + |r_3| + s_3 \sqrt{n}}.
\end{array} \right. 
\]  
(6.9)

\[
\text{Proof. First, we want to prove } C = \sqrt{1 + |r_i| + s_i}, \text{ for } i = 1, 2, 3.
\]

From Theorem (6.1), we can obtain

\[
\left\{ \begin{array}{l}
|\tilde{r}_m, -\tilde{t}_m| \\
|\tilde{r}_m, -\tilde{t}_m| \\
|\tilde{r}_m, -\tilde{t}_m|
\end{array} \right\} < \sqrt{1 + |r_i| + s_i \sqrt{n}}, \text{ for } i = 1, 2, 3.
\]

And from Theorem(6.2), we can get

\[
\left\{ \begin{array}{l}
|\tilde{r}_{m+2}, -\tilde{t}_{m+2}| \\
|\tilde{r}_{m+2}, -\tilde{t}_{m+2}| \\
|\tilde{r}_{m+2}, -\tilde{t}_{m+2}|
\end{array} \right\} < \sqrt{1 + |r_i| + s_i \sqrt{n}}, \text{ } i = 1, 2, 3,
\]

then

\[
\min \left\{ \begin{array}{l}
|\tilde{r}_m, -\tilde{t}_{m+1}| \\
|\tilde{r}_m, -\tilde{t}_{m+1}| \\
|\tilde{r}_m, -\tilde{t}_{m+1}|
\end{array} \right\} < \sqrt{1 + |r_i| + s_i \sqrt{n}}, \text{ } i = 1, 2, 3. \quad (6.10)
\]

By comparison between two equations (6.1) and (6.10), we can find the value of \( C \) as in equation (6.8).

Now to prove any multiple \( kP \) can be decomposed as in equation (5.10) with the conditions in equation (6.9). Since \( X^2 + r_iX + s_i \) are irreducible in \( Z[X] \), we must have the inequality in equation (6.3). This implies that the inequality in equation (6.4). In particular,

\[
\left\{ \begin{array}{l}
|\tilde{r}_{m+1}, -\tilde{t}_{m+1}| \\
|\tilde{r}_{m+1}, -\tilde{t}_{m+1}| \\
|\tilde{r}_{m+1}, -\tilde{t}_{m+1}|
\end{array} \right\} \geq \sqrt{n} / \sqrt{1 + |r_i| + s_i}, \text{ for } i = 1, 2, 3,
\]

and \( |(r_{m+1}, -t_{m+1})| = |v_1|, |(\tilde{r}_{m+1}, -\tilde{t}_{m+1})| = |v_2| \) and \( |(\hat{r}_{m+1}, -\hat{t}_{m+1})| = |v_3| \).

Since \( u_1 = (k_{11}, k_{12}) \) and \( u_2 = (k_{21}, k_{22}) \) from equation (5.7) and from equation (5.8), respectively, we can get \( k_1 = k_{11} + k_{12} \lambda_1 \text{ (mod } n) \) and \( k_2 = k_{21} + k_{22} \lambda_2 \text{ (mod } n) \) which are equivalent to \( k_1P = k_{11}P + k_{12}\psi_1(P) \) and \( k_2 = k_{21}P + k_{22}\psi_2(P) \) as shown in equation(5.9).
From inequalities in equation (5.2) as

\[ |u_1| \leq \left| \frac{v_3 + v_4}{2} \right| < M \quad \text{and} \quad |u_2| \leq \left| \frac{v_5 + v_6}{2} \right| < M, \]

then

\[ |(k_{11}, k_{12})| < M \quad \text{and} \quad |(k_{21}, k_{22})| < M. \]

Since \( M \leq C\sqrt{n} \), then \(|(k_{11}, k_{12})| < C\sqrt{n} \) and \(|(k_{21}, k_{22})| < C\sqrt{n} \). Now, from definition (2.5) of rectangle norm

\[ |(k_{11}, k_{12})| = \max(|k_{11}|, |k_{12}|) \quad \text{and} \quad |(k_{21}, k_{22})| = \max(|k_{21}|, |k_{22}|). \]

This means that \( \max(|k_{11}|, |k_{12}|) < C\sqrt{n} \) and \( \max(|k_{21}|, |k_{22}|) < C\sqrt{n} \).

Finally, from equation (6.8) to compute \( C \), we can find

\[
\max \left\{ \begin{array}{c} |k_{11}|, |k_{12}| \\ |k_{21}|, |k_{22}| \end{array} \right\} < \sqrt{1 + |r_i| + s_i} \sqrt{n} \quad \text{for} \quad i = 2, 3.
\]

7. Conclusion

The present work proposes a new method which help facilitate the use of Gallant et al.'s (GLV) integers are not bounded by ±\( \sqrt{n} \). This new method, namely, the integer sub-decomposition method, ISD will help increase 50% more successful rate as compared to the GLV method in the computation of the \( kP \).

This study also, focuses on presenting an accurate analysis of the ISD method that optimizes and proves on existing bound. This bound determines value \( C \) which is greater than 1, say \( C = \sqrt{1 + |r_i| + s_i}, \quad i = 1, 2, 3 \) in case in which the endomorphism rings \( \text{End}[\psi] \) over \( \mathbb{Z} \). This analysis can be applied when embedding endomorphism rings \( \text{End}[\psi] \) into complex number field \( \mathbb{C} \), one can further notice that dealing with similar case where \( C > 1 \) is more complicated than in case in which the endomorphism rings \( \text{End}[\psi] \) over \( \mathbb{Z} \). Moreover, the generalization can include the hyperelliptic curves of the ISD method.

References


Appendix A. ISD Generators Algorithm

Algorithm 1 (Find ISD generators $v_1 = (a, b)$, $v_2 = (c, d)$, $v_3 = (g, j)$ and $v_4 = (e, f)$ for given $n$ and $\lambda_1, \lambda_2 \in \mathbb{Z}$, where $\lambda_1 \neq \pm \lambda_2$).

**Input.** Integers $n, \lambda_1, \lambda_2$.

**Output.** The vectors $v_1, v_2, v_3$ and $v_4$.

1. **Step 1.** Compute $v_1 = (a_{m+1}, -b_{m+1})$ and $v_3 = (g_{m+1}, -j_{m+1})$ such that $s_{m+1}n + b_{m+1}\lambda_1 = a_{m+1}$ and $u_{m+1}n + j_{m+1}\lambda_1 = g_{m+1}$ where $|a_{m+1}|, |b_{m+1}|, |g_{m+1}|$ and $|j_{m+1}| < C\sqrt{n}$ by using the extended Euclidean algorithm to find firstly the greatest common divisor of $n$ and $\lambda_1$ and secondly of the same $n$ and $\lambda_2$. (This is the extension of Gallant et al.’s algorithm for two vectors $v_1$ and $v_3$).

2. **Step 2.** Check if each component of $v_2$ either $(a_m, -b_m)$ or $(a_{m+2}, -b_{m+2})$ and $(g_m, -j_m)$ or $(g_{m+2}, -j_{m+2})$ is bounded by $C\sqrt{n}$, stop and set the shorter of $(a_m, -b_m)$ and $(a_{m+2}, -b_{m+2})$ as the second vector $v_2$, also set the shorter of $(g_m, -j_m)$ and $(g_{m+2}, -j_{m+2})$ as the fourth vector $v_4$. Otherwise, go to step 3.

3. **Step 3.** Find any $d', w', f'$ and $v'$ such that $s_{m+1}d' - b_{m+1}w' = 1$ and $u_{m+1}f' - j_{m+1}v' = 1$.

For example, $d'$ and $w'$ are obtained from the extended Euclidean algorithm, since $s_{m+1}$ is relatively prime to $-b_{m+1}$, and the same thing with $f'$ and $v'$ are obtained from the extended Euclidean algorithm, since $u_{m+1}$ is relatively prime to $-j_{m+1}$.

4. **Step 4.** Compute

$$I_{11} = -\frac{d'}{b} - \frac{\sqrt{n}}{b}, \quad I_{12} = -\frac{d'}{b} + \frac{\sqrt{n}}{b}$$
and
\[ I_{11}' = -\frac{f'}{j} - \sqrt{n} \quad \text{and} \quad I_{12}' = -\frac{f'}{j} + \sqrt{n} \].

**Step 5.** Let
\[ I_1 = [I_{11}, I_{12}], \quad I_1' = [I_{11}', I_{12}'], \quad \text{if} \ b > 0, \]
and
\[ I_1 = [I_{12}, I_{11}], \quad I_1' = [I_{12}', I_{11}'], \quad \text{if} \ b < 0. \]

**Step 6.** Compute
\[ I_{21} = -\frac{d'\lambda_1 - w'n}{a} - \frac{\sqrt{n}}{a}, \quad I_{22} = -\frac{d'\lambda_1 - w'n}{a} + \frac{\sqrt{n}}{a}. \]

Also,
\[ I_{21}' = -\frac{f'\lambda_2 - v'n}{g} - \frac{\sqrt{n}}{g}, \quad I_{22}' = -\frac{f'\lambda_2 - v'n}{g} + \frac{\sqrt{n}}{g}. \]

**Step 7.** Let \( I_2 = [I_{21}, I_{22}] \) and \( I_2' = [I_{21}', I_{22}'] \).

**Step 8.** Find all integers in the intersection of \( I_1 \) and \( I_2 \) and define them by \( \alpha_1 \), also all integers in the intersection of \( I_1' \) and \( I_2' \) and define them by \( \alpha_2 \). Note that the numbers of \( \alpha_1's \) and \( \alpha_2's \) are at most 4. If there is not any of such integers exist, stop.

**Step 9.** Set \( v_2 = (c, d) \) and \( v_4 = (e, f) \), where
\[ c = w'n - d'\lambda_1 + \alpha_1 a, \quad d = d' + \alpha_1 b \]
and
\[ e = v'n - f'\lambda_2 + \alpha_2 g, \quad f = f' + \alpha_2 j. \]

One can easily verify that \( v_2 = (c, d) \) and \( v_4 = (e, f) \) are in the \( \mathcal{K} \) and \( |c|, |d|, |e| \) and \( |f| < C\sqrt{n} \), therefore, \( \{v_1, v_2\} \) and \( \{v_3, v_4\} \) are ISD generators.

**Appendix B. Balanced Length-Two Representation of a Sub-Decomposition Multiplier Algorithm**

**Algorithm 2** (Balanced length-two representation of a sub-decomposition multiplier algorithm).

**Input.** Integers \( n, \lambda_1, \lambda_2 \in [1, n - 1] \), where \( \lambda_1 \neq \pm \lambda_2 \) and \( k_1, k_2 \in [1, n - 1] \).

**Output.** Integers \( k_{11}, k_{12}, k_{21} \) and \( k_{22} \) such that \( k = k_{11} + k_{12}\lambda_1 + k_{21} + k_{22}\lambda_2 \mod n \) and \( |k_{11}|, |k_{12}|, |k_{21}|, |k_{22}| < C\sqrt{n} \).
Step 1. Run ISD generators algorithm (1) with inputs $n, \lambda_1$ and $\lambda_2$. The algorithm produces the ISD generators $\{v_3, v_4\}$ and $\{v_5, v_6\}$.

Step 2. Set $v_3 = (\bar{r}_{m+1}, -\bar{t}_{m+1}) = (\bar{r}, -\bar{t})$ and $v_5 = (\hat{r}_{m+1}, -\hat{t}_{m+1}) = (\hat{r}, -\hat{t})$.

Step 3. If $(\bar{r}_m^2 + \bar{t}_m^2) \leq (\bar{r}_{m+2}^2 + \bar{t}_{m+2}^2)$ then set

$$ v_4 = (\bar{u}, \bar{v}) \leftarrow (\bar{r}_m, -\bar{t}_m) \quad \text{and} \quad v_6 = (\hat{u}, \hat{v}) \leftarrow (\hat{r}_m, -\hat{t}_m). $$

Else

$$ v_4 = (\bar{u}, \bar{v}) \leftarrow (\bar{r}_{m+2}, -\bar{t}_{m+2}) \quad \text{and} \quad v_6 = (\hat{u}, \hat{v}) \leftarrow (\hat{r}_{m+2}, -\hat{t}_{m+2}). $$

Step 4. Compute $c_3 = \lceil \bar{k}_1/n \rceil$, $c_4 = \lceil -\bar{k}_1/n \rceil$ and $c_5 = \lceil \hat{k}_2/n \rceil$, $c_6 = \lceil -\hat{k}_2/n \rceil$.

Step 5. Compute $k_{11} = k_1 - c_3 \bar{r} - c_4 \bar{u}$, $k_{12} = -c_3 \bar{t} - c_4 \bar{v}$ and $k_{21} = k_2 - c_5 \hat{r} - c_6 \hat{u}$, $k_{22} = -c_5 \hat{t} - c_6 \hat{v}$.

Step 6. Return $k_{11}, k_{12}, k_{21}$ and $k_{22}$.

**Appendix C. Modification of Point Multiplication with Two Efficiently Computable Endomorphisms Algorithm**

**Algorithm 3** (Modification of point multiplication with two efficiently computable endomorphisms algorithm).

**Input.** Integer $n$, $k_1, k_2 \in [1, n-1]$, $P \in E(F_p)$, window widths $w_1, w_2, w_3$ and $w_4$, $\lambda_1, \lambda_2 \in \mathbb{Z}$, where $\lambda_1 \neq \pm \lambda_2$.

**Output.** $kP$.

**Step 1.** Use balanced length-two representation a sub-decomposing of a multiplier algorithm to find $k_{11}, k_{12}, k_{21}$ and $k_{22}$ such that

$$ k = k_{11} + k_{12} \lambda_1 + k_{21} + k_{22} \lambda_2 \pmod{n}. $$

**Step 2.** Calculate $P_2 = \psi_1(P)$, $P_3 = \psi_2(P)$ and let $P_1 = P$.

**Step 3.** Use computing width-$w$ NAF of positive integer algorithm to compute

$$ NAF_{w_j}(|k_{z,j}|) = \Sigma_{i=1}^{l_j-1} k_{z,j,i} 2^i \quad \text{for} \quad j = 1, 2 \quad \text{and} \quad z = 1, 2. $$

**Step 4.** Let $l_z = \max\{l_{z,1}, l_{z,2}\} \quad \text{if} \quad z = 1, 2$.

**Step 5.** If $k_{z,j} < 0$, then set $G_{z,j,i} \leftarrow -G_{z,j,i}$ for $i = 0 : l_z$, $j = 1, 2$ and $z = 1, 2.$
Step 6. Compute \( iP_j \) and \( iP_s \) for \( i \in \{1, 3, ..., 2^{w_j} - 1\} \) and \( i \in \{1, 3, ..., 2^{w_s} - 1\} \), where \( j = 1, 2 \) and \( s = 1, 3 \).

Step 7. \( Q \leftarrow \infty \).

Step 8. For \( i = l_z - 1 : 0 \) do

8.1 \( Q \leftarrow 2Q \).

8.2 For \( j = 1, 2, z = 1 \) do

- If \( G_{z,j,i} \neq 0 \) then:
  - If \( G_{z,j,i} > 0 \) then \( Q \leftarrow Q + k_{z,j,i}P_j \);
  - Else \( Q \leftarrow Q - |k_{z,j,i}|P_j \).

Step 9. For \( j = 1, 2, z = 2 \) do

- If \( G_{z,j,i} \neq 0 \) and \( s = 1, 3 \) then
- If \( G_{z,j,i} > 0 \) then \( Q \leftarrow Q + k_{z,j,i}P_s \);
- Else \( Q \leftarrow Q - |k_{z,j,i}|P_s \).

Step 10. Return \( Q \).

Appendix D. ISD Method to Compute Point Multiplication Elliptic Curve \( kP \)

Algorithm 4 (ISD Method to Compute Point Multiplication Elliptic Curve \( kP \)). This algorithm consists of the following steps:

Step 1. Apply GLV generator algorithm in [11] to find the generator \( \{v_1, v_2\} \) for the given \( n \) and \( \lambda \) such that \( v_1 \leftarrow (r, t) \) and \( v_2 \leftarrow (u, v) \).

Step 2. Use balanced length-two representation of a multiplier algorithm in [3] to decompose \( k \) to find \( k_1 \) and \( k_2 \) for a given \( n \), \( \lambda \) and \( k \in [1, n - 1] \).

As for the proposed steps for modification, they include the following:

Step 3. Use algorithm (2) to find

3.1 For \( n \) and \( \lambda_1 \), generate the ISD generator \( \{v_3, v_4\} \) such that \( v_3 \leftarrow (\bar{r}, \bar{t}) \) and \( v_4 \leftarrow (\bar{u}, \bar{v}) \).

3.2 For \( n \) and \( \lambda_2 \), generate the ISD generator \( \{v_5, v_6\} \) such that \( v_5 \leftarrow (\hat{r}, \hat{t}) \) and \( v_6 \leftarrow (\hat{u}, \hat{v}) \).

Step 4. Use algorithm (3) to decompose \( k_1 \) and \( k_2 \) such that \( k_1 = k_{11} + k_{12}\lambda_1 \ (mod \ n) \) and \( k_2 = k_{21} + k_{22}\lambda_2 \ (mod \ n) \). That is, one can get \( k = k_{11} + k_{12}\lambda_1 + k_{21} + k_{22}\lambda_2 \ (mod \ n) \).
Step 5. Use algorithm (4) to compute $kP$ defined as

$$kP = k_{11}P + k_{12}[\lambda_1]P + k_{21}P + k_{22}[\lambda_2]P$$

$$= k_{11}P + k_{12}\psi_1(P) + k_{21}P + k_{22}\psi_2(P).$$

such that $\psi_1(P) \leftarrow [\lambda_1]P$ and $\psi_2(P) \leftarrow [\lambda_2]P$, where $\lambda_1, \lambda_2 \in \mathbb{Z}$ and $\lambda_1 \neq \pm \lambda_2$. 