Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080209

Point Multiplication using Integer Sub-Decomposition for Elliptic Curve Cryptography

Ruma Kareem K. Ajeena[*] *and Hailiza Kamarulhaili*

School of Mathematical Sciences, Universiti Sains Malaysia, 11800, Penang, Malaysia

Received: 13 Mar. 2013, Revised: 13 Jul. 2013, Accepted: 15 Jul. 2013
Published online: 1 Mar. 2014

Abstract: In this work, we proposed a new approach called integer sub-decomposition (ISD) based on the GLV idea to compute any multiple kP of a point P of order n lying on an elliptic curve E. This approach uses two fast endomorphisms ψ_1 and ψ_2 of E over prime field F_p to calculate kP. The basic idea of ISD method is to sub-decompose the returned values k_1 and k_2 lying outside the range $\sqrt{n}$ from the GLV decomposition of a multiplier k into integers k_{11}, k_{12}, k_{21} and k_{22} with $-\sqrt{n} < k_{11}, k_{12}, k_{21}, k_{22} < \sqrt{n}$. These integers are computed by solving a closest vector problem in lattice. The new proposed algorithms and implementation results are shown and discussed in this study.

Keywords: Elliptic Curve Cryptosystem, Scalar Multiplication, Integer Sub-Decomposition, Efficiently Computable Endomorphisms.

1 Introduction

In 1985, Miller [1] and Koblitz [2] introduced the elliptic curve cryptosystem. Since then, the elliptic curve cryptosystem has been invested in the field of public key cryptography because of its low bandwidth and small space storage requirements [3]-[6]. In the main operation of public key schemes, scalar multiplication can be accomplished using elliptic curves [7]. The latter can be dealt with by successively doubling and adding the points. Inversions and multiplications over the underlying finite field are required. Despite the small size of the key, the required complexity may still be relatively heavy. Accordingly, many studies have been conducted and several other approaches have been proposed to improve the computational efficiency of the elliptic curve cryptography [8]-[13]. For instance, an approach was set to analyze the algebraic structure of elliptic curves. This approach was further used to classify a class of special curves with better efficiency in the scalar multiplication. The use of Koblitz curves can increase efficiency. In Koblitz curves, scalar multiplication requires no point to be doubled by exploiting a feature of the Frobenious endomorphism [14]. The Frobenious endomorphism can be efficiently computed when the underlying finite field is of characteristic 2 because the squaring operation is much faster than the multiplication. The same idea can be invested to elliptic curves having arbitrary characteristics; however, any improvement in the efficiency is not guaranteed . Recently, a scalar multiplication kP has been suggested by Gallant et al. (2001) in [15] using an efficiently computable endomorphism of an elliptic curve E over a prime field F_p for a point $P \in E$ of prime order n. They introduced an idea using the decomposition of $k = k_1 + k_2\lambda \ (mod\ n)$, where λ is an integer that satisfies $\psi(P) = \lambda P$ and ψ is an endomorphism on E. They stated that if endomorphism is efficiently computable and if each component of k_1, k_2 in the decomposition is short enough, then their method can improve the computational efficiency up to 50%. The following equation is the decomposition process:

$$k = k_1 + k_2\lambda \ (mod\ n) \qquad (1)$$

with $-\sqrt{n} < k_1, k_2 < \sqrt{n}$.
Gallant et al. introduced a method in which they used two linearly independent short vectors v_1 and v_2 in the kernel of the homomorphism

$$T : Z \times Z \to Z/n, \qquad (2)$$

defined by

$$T(i, j) = i + j\lambda (mod\ n). \qquad (3)$$

To make the notation simple, a set of such vectors denoted as v_1 and v_2 which are called a GLV (R. Gallant,

[*] Corresponding author e-mail: ruma.usm@gmail.com

R. Lambert and S. Vanstone) generator, will be defined later. The original GLV was not perfect because of its certain gaps that were left unproven. Hence, the GLV method offered in another model that assists in finding a GLV generator. The existence of a GLV generator and the success of finding this generator cannot be guaranteed. Therefore, Kim and Lim (2003) in [17] proposed a necessary condition for the existence of the GLV generator and a method of finding it when such generator exists.

The GLV method and subsequent improvements on it rely on the decomposition values of k_1 and k_2 for values that fall within the range $-\sqrt{n} < k_1, k_2 < \sqrt{n}$. For k_1 and k_2 values not within this range, the GLV method will not work. A new generator should be obtained to generate the next values of k_1 and k_2 that fall within the given range. In this paper, we propose a new method called integer sub-decomposition (ISD) to overcome this problem. This method allows us to work with k_1 and k_2 values that fall outside the given range. ISD method has improved the computational efficiency compared with the general method of computing scalar multiplication in elliptic curves over the prime field. In the present paper, we introduce a sub-decomposition process and present three main problems that are aimed to be investigated in ordinary elliptic curves E that are defined over F_p.

Problem 1: Let E be ordinary elliptic curve over F_p, $P \in E(F_p)$ has a large prime order n, and $\lambda_1, \lambda_2 \in [1, n-1]$, where $\lambda_1 \neq \pm\lambda_2$. Construct the linearly independent integer vectors v_1, v_2, v_3 and v_4 which are lattice integer points computed by solving the closest vector problem in lattice.

Problem 2: Let E be ordinary elliptic curve over F_p, $P \in E(F_p)$ has a large prime order n, and $\lambda_1, \lambda_2 \in [1, n-1]$, where $\lambda_1 \neq \pm\lambda_2$ and the linearly independent integer vectors v_1, v_2, v_3 and v_4 are originated. Find two ISD generators $\{v_1, v_2\}$ and $\{v_3, v_4\}$ that satisfy the necessary condition that includes the relation between components for any vector v_i, for $i = 1, 2, 3, 4$ is relatively prime.

Problem 3: Let E be ordinary elliptic curve over F_p such that $\#E(F_p) = p + 1 - t$, P is a point lying on E has a large prime order n, and $k \in [1, n-1]$. Assume that $\{v_1, v_2\}$ and $\{v_3, v_4\}$ are ISD generators. Compute the point multiplication elliptic curve kP when the values k_1 and k_2 are not bounded by $\pm\sqrt{n}$ in the ISD method.

This paper is organized as follows. Section 2 presents a synopsis of the mathematical background to explain elliptic curve E over prime finite field and its endomorphism ψ . Section 3 briefly reviews the mechanisms of the scalar multiplication using a GLV generator proposed in [15],[17]. Section 4 presents the extension of the necessary condition for the existence of two ISD generators. In addition, we demonstrate a new algorithm that helps find ISD generators. Section 5 displays a new method for computing scalar multiplication depending on the sub-decomposition of k_1 and k_2 when both or one of them is not bounded by $\pm\sqrt{n}$. Section 6 shows the implementation results. Finally, Section 7 is the concluding remarks.

2 Preliminaries

Most applications of elliptic curves theory in cryptography deal with elliptic curves defined over a finite field, F_p, where p is a prime number. This curve is called prime curve.

2.1 Elliptic Curve over F_p

Definition 1. *[19] Let $p \neq 2, 3$. An elliptic curve $E(F_p)$ over F_p, be defined by an equation of the form:*

$$E : Y^2 = X^3 + AX + B \,(mod\ n), \tag{4}$$

where $A, B \in F_p$. The curve E is non-singular if it has no double zeroes, that means the discriminant $D_E = 4A^3 + 27B^2 \neq 0 \,(mod\ n)$.

Definition 2. *[19],[20] Let $E(F_p)$ be an elliptic curve defined in equation (4) 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:*

$$\mu \equiv \left(\frac{y_Q - y_P}{x_Q - x_P}\right) (mod\ p)\ \ if\ \ P \neq Q \tag{5}$$

or

$$\mu \equiv \left(\frac{3x_P^2 + A}{2y_p}\right) (mod\ p)\ \ if\ \ P = Q \tag{6}$$

$$x_R \equiv \lambda^2 - x_P - x_Q \,(mod\ p)$$
$$y_R \equiv \lambda(x_P - x_R) - y_P \,(mod\ p).$$

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

2.2 Endomorphisms ψ of Elliptic curve E over F_p

Assume that E is an elliptic curve defined over the finite field F_p. The point at infinity is denoted as 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 ψ will be defined over F_q where $q = p^n$ if the rational map is defined over F_q. Therefore clearly, for any $n \geq 1$, ψ is a group homomorphism of $E(F_p)$ and $E(F_q)$ [11],[20].

Definition 3. *The endomorphism of elliptic curve E defined over F_q is the $m-$ multiplication map $[m]: E \to E$ defined by*

$$P \to mP \tag{7}$$

for each $m \in Z$. The negation map $[-1]: E \to E$ defined by $P \to -P$ is a special case from $m-$multiplication map [11].

3 GLV generator

Definition 4. *[17] A GLV generator is a set $\{v_1, v_2\}$ of two linearly independent vectors v_1 and v_2 in the kernel of the homomorphism T in an equation (2) defined in the equation (3). It is called so if each component of v_1 and v_2 is bounded by $\sqrt{n}$.*

Lemma 1. *[17] Let n be prime and $\lambda \in [1, n-1]$. In addition, assumes that $v_1 = (r,t)$, $v_2 = (u,v) \in kerT$ and $-\sqrt{n} < r,t,u,v < \sqrt{n}$. If v_1 and v_2 are linearly independent, then r is relatively prime to t and u is relatively prime to v.*

Lemma 2. *[17] Let n be prime and $\lambda \in [1, n-1]$. If there is a vector $v = (r,t)$ in the kernel of T such as $gcd(r,t) \neq 1$ and $-\sqrt{n} < r,t < \sqrt{n}$, then there will be no GLV generator.*

4 Original 2-GLV Method by Gallant et al.

4.1 Domain Parameters of The Original 2-GLV Method

A set of parameters should be followed in the original method [15]-[18]. These involve the following:

(i) F_p is a finite field of p elements, p is the prime number;
(ii) E is an elliptic curve defined over F_p with the point at infinity O_E;
(iii) $P \in E(F_p)$ is a rational point of a large prime order n. That is, the cofactor $h = \#E(F_p)/n$ is small, and $h \leq 4$;
(iv) An endomorphism ψ of E is a rational map $\psi: E \to E$ with $\psi(O_E) = O_E$, is an efficiently computable endomorphism of E over F_p, and it acts on the subgroup $\langle P \rangle$ as a multiplication by λ such that

$$\psi(P) = [\lambda]P. \tag{8}$$

(v) λ is a root of the characteristic polynomial

$$Charpoly(X) = X^2 + rX + s, \tag{9}$$

of ψ, where r, s represent small fixed integers in F_p and $\lambda \in [1, n-1]$;
(vi) k is an integer that is selected uniformly at random from the interval $[1, n-1]$;
(vii) The group homomorphism (the GLV reduction map) $T: Z \times Z \to Z/n$ is defined in equation (3).

4.2 How to use a GLV generator to calculate kP

The following section illustrates how Gallant et al. used a GLV generator to accelerate the computation of kP. Suppose that $\{v_1, v_2\}$ is a GLV generator. In view of the fact that v_1 and v_2 are linearly independent over $Q \times Q$, the latter will span $Q \times Q$. Consequently, one will have

$$(k, 0) = \beta_1 v_1 + \beta_2 v_2, \tag{10}$$

for some $\beta_1, \beta_2 \in Q$. Let b_1, b_2 be the nearest integers to β_1, β_2 respectively. Finally, set

$$\begin{aligned} x &= (k_1, k_2) = (k, 0) - (b_1 v_1, b_2 v_2) \\ &= (k, 0) - (\beta_1 v_1 + \beta_2 v_2) + (\beta_1 v_1 + \beta_2 v_2) - (b_1 v_1, b_2 v_2) \\ &= (\beta_1 - b_1) v_1 + (\beta_2 - b_2) v_2). \end{aligned}$$

Then, $T(x) = k$ can be obtained from equation (2) and $\|x\| \leq \frac{1}{2}(\| v_1 \| + \| v_2 \|)$ can be obtained from lemma 2 in [15], where $\| \cdot \|$ is an Euclidean norm. Since $\{v_1, v_2\}$ represents a GLV generator, each component of v_1 and v_2 is bounded by $\sqrt{n}$ and the result will be

$$-\sqrt{n} < k_1, k_2 < \sqrt{n}. \tag{11}$$

Thus, equation (1) can always be decompose with the condition in equation (11) from any GLV generator $\{v_1, v_2\}$. Hence, kP can be calculated by

$$kP = k_1 P +_E k_2 \psi(P) \tag{12}$$

using the window simultaneous multiple point multiplication method for P and $\psi(P)$. In addition, the efficiency improvement should roughly be 50% over the general scalar multiplication method for the currently recommended key sizes.

Remark. In decomposing the integer k into k_1 and k_2, we can sometimes get to one of the values of k_1 or k_2 equal to zero. This case is not admissible in decomposition because it cannot satisfy the equation (1).

5 Elliptic Scalar Multiplication using Integer Sub-Decomposition (ISD) Method

5.1 A condition for the new ISD generators

In this study, we state and prove a necessary condition for the existence of ISD generators based on the idea of necessary condition of GLV generator [17].

Assume that $v_1 = (a,b)$, $v_2 = (c,d)$, $v_3 = (g,j)$ and $v_4 = (e,f)$ are linearly independent integer vectors in the kernel of T such that $-\sqrt{n} < a,b,c,d,g,j,e,f < \sqrt{n}$. Then we have

$$\begin{cases} a+b\lambda_1 = sn, \\ c+d\lambda_1 = wn, \\ g+j\lambda_2 = un, \\ e+f\lambda_2 = vn, \end{cases} \tag{13}$$

for some $s,w,u,v \in Z$. By multiplying the first and the second equations in (13) by c and a, respectively, we obtain

$$(bc-ad)\lambda_1 = (sc-aw)n, \tag{14}$$

By multiplying the third and the fourth equations in (13) by e and g, respectively, we obtain

$$(je-gf)\lambda_2 = (ue-gv)n. \tag{15}$$

Similarly, we have

$$(bc-ad) = (sd-bw)n, \tag{16}$$

and

$$(gf-je) = (uf-jv)n. \tag{17}$$

Note that $|bc-ad| < 2n$ and $|je-gf| < 2n$. If $bc-ad = 0$ then (a,b) and (c,d) are linearly dependent. And, if $je-gf = 0$ then (g,j) and (e,f) are linearly dependent.Thus $bc-ad = -n,n$ and $je-gf = -n,n$ because n divides $bc-ad$ and $je-gf$. From equations (16) and (17), we have

$$\begin{cases} sd-bw = -1,1 \\ uf-jv = -1,1 \end{cases} \tag{18}$$

Therefore, we conclude that b,d,j, and f are relatively prime to s,w,u, v, relatively. We shall state and prove the following Lemmas.

Lemma 3. *Let n be prime, λ_1 and $\lambda_2 \in [1,n-1]$, where $\lambda_1 \neq \pm\lambda_2$. Assume that $v_1 = (a,b)$, $v_2 = (c,d)$,$v_3 = (g,j)$ and $v_4 = (e,f) \in kerT$ such that $-\sqrt{n} < a,b,c,d,g,j,e,f < \sqrt{n}$. If v_1,v_2,v_3 and v_4 are linearly independent, then a,c,g, and e are relatively prime to b,d,j, and f, respectively.*

Proof. Since v_1,v_2,v_3 and $v_4 \in kerT$, we have $s,w,u,v \in Z$ which satisfy equation (13). Assume that the greatest common divisor of a and b is $\alpha > 1$. Then α becomes a common divisor of s and b from equation (13) since n is prime and this contradicts to (18) □.

Remark. Lemma (3) shows a necessary condition for the existence of ISD generators $\{v_1,v_2\}$ and $\{v_3,v_4\}$. If $v_1 = (a,b) \in kerT$, $gcd(a,b) \neq 1$ and $|a| < \sqrt{n}, |b| < \sqrt{n}$, then the second vector $v_2 = (c,d)$, $|c| < \sqrt{n}, |d| < \sqrt{n}$ never existed. The same thing will happen with v_3 and v_4. In fact, Lemma (3) itself shows that if $gcd(a,b) \neq 1$ and $gcd(g,j) \neq 1$, there are no ISD generators that contain the vectors (a,b) and (g,j), respectively.

Lemma 4. *Let n be prime, λ_1 and $\lambda_2 \in [1,n-1]$, where $\lambda_1 \neq \pm\lambda_2$. If there are vectors $v = (a,b)$ and $u = (g,j)$ in the kernel of T such that $gcd(a,b) \neq 1$, $gcd(g,j) \neq 1$ and $-\sqrt{n} < a,b,g,j < \sqrt{n}$, then there exist no ISD generators.*

Proof. Suppose that $\{v_1,v_2\}$ and $\{v_3,v_4\}$ are ISD generators. Thus, either $\{v,v_1\}$ or $\{v,v_2\}$ is an ISD generator containing v, also $\{u,v_3\}$ or $\{u,v_4\}$ is an ISD generator containing u. Thus, contradicts Lemma (3). Therefore, there exist no ISD generators from Lemma (3) □.

5.2 The proposed algorithm to find ISD generators

5.2.1 Finding v_2 and v_4

Using the method proposed by Gallant et al. described in section 4, one can always get the vectors v_1 and v_3, where each component of v_1 and v_3 is bounded by $\sqrt{n}$. Now we present an algorithm to find the second and the fourth short vectors v_2 and v_4 after obtaining the vectors v_1 and v_3. Suppose we have the vectors $v_1 = (a_{m+1}, -b_{m+1})$ and $v_3 = (g_{m+1}, -j_{m+1})$ in the kernel of T as in Gallant et al.'s algorithm. We know that $|a_{m+1}|$, $|b_{m+1}|$, $|g_{m+1}|$ and $|j_{m+1}|$ are already less than $\sqrt{n}$. Let $v_2 = (c,d)$ and $v_4 = (e,f)$ be the vectors so that $\{v_1,v_2\}$ and $\{v_3,v_4\}$ are ISD generators. Suppose $v_1 = (a,b)$, $v_2 = (c,d)$,$v_3 = (g,j)$, and $v_4 = (e,f)$ satisfy the equation (13) for some s,w,u and $v \in Z$. From the equation (18), we know that s and u are relatively prime to $-b$ and $-j$, respectively. We apply the extended Euclidean algorithm to find the greatest common divisor of s and $-b$, and also to find the greatest common divisor of u and $-j$. Then the algorithm returns d',w',u' and v' which satisfy

$$\begin{cases} sd'-bw' = 1 \\ uf'-jv' = 1 \end{cases} \tag{19}$$

In general, every integer vector (d,w) and (f,v) which satisfy $sd-bw = 1$ and $uf-jv = 1$ can be represented by $(d'+\alpha_1 b, w'+\alpha_1 s)$, $(f'+\alpha_2 j, v'+\alpha_2 u)$ where $\alpha_1,\alpha_2 \in Z$. Our purpose is to find a suitable α_1 and α_2. Set $d = d'+\alpha_1 b, w = w'+\alpha_1 s$ and $f = f'+\alpha_2 j$, $v = v'+\alpha_2 u$. Since $|d| < \sqrt{n}$, $b = -b_{m+1} \neq 0$ and $|f| < \sqrt{n}$, $j = -j_{m+1} \neq 0$, we have

$$-\frac{d'}{b} - \frac{\sqrt{n}}{b} < \alpha_1 < -\frac{d'}{b} + \frac{\sqrt{n}}{b} \tag{20}$$

and

$$-\frac{f'}{j} - \frac{\sqrt{n}}{j} < \alpha_2 < -\frac{f'}{j} + \frac{\sqrt{n}}{j}, \tag{21}$$

where $b,j > 0$.

Also,

$$-\frac{d'}{b} + \frac{\sqrt{n}}{b} < \alpha_1 < -\frac{d'}{b} - \frac{\sqrt{n}}{b} \tag{22}$$

and

$$-\frac{f'}{j} + \frac{\sqrt{n}}{j} < \alpha_2 < -\frac{f'}{j} - \frac{\sqrt{n}}{j}, \tag{23}$$

where $b, j < 0$.

Note that $c = wn - d\lambda_1$ and $a = a_{m+1} > 0$, then we have

$$\frac{d'\lambda_1 - w'n}{a} - \frac{\sqrt{n}}{a} < \alpha_1 < \frac{d'\lambda_1 - w'n}{a} + \frac{\sqrt{n}}{a}, \qquad (24)$$

also, $e = vn - f\lambda_2$ and $g = g_{m+1} > 0$, then we have

$$\frac{f'\lambda_2 - v'n}{g} - \frac{\sqrt{n}}{g} < \alpha_2 < \frac{f'\lambda_2 - v'n}{g} + \frac{\sqrt{n}}{g}. \qquad (25)$$

Hence, α_1 has to be an integer in the intersection of equations (20), (22) and (24). Also, α_2 has to be an integer in the intersection of equations (21), (23) and (25). From Lemma (4), in order to seek α_1 and α_2 for the second and fourth vectors v_2 and v_4 of ISD generators, it is sufficient to test only eight integers at most since one of $|a|, |b|$ and $|g|, |j|$ is greater than $\frac{1}{2}\sqrt{n}$. Now, we present our algorithm to find the second and the fourth vector v_2 and v_4 respectively.

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 $\lambda_1, \lambda_2 \in Z$, where $\lambda_1 \neq \pm\lambda_2$.

Input: Integers n, λ_1, λ_2.

Output: The vectors v_1, v_2, v_3 and v_4.

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 $|j_{m+1}| < \sqrt{n}$ by using the extended Euclidean algorithm to find firstly the greatest common divisor of n and λ_1 and secondly of the same n and λ_2. (This is the extension of Gallant et al.'s algorithm for two vectors v_1 and v_3).

Step 2. Check if each components of 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 $\sqrt{n}$, 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.

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 $-j_{m+1}$.

Step 4. Compute

$$I_{11} = -\frac{d'}{b} - \frac{\sqrt{n}}{b}, \; I_{12} = -\frac{d'}{b} + \frac{\sqrt{n}}{b}$$

and

$$I'_{11} = -\frac{f'}{j} - \frac{\sqrt{n}}{j}, \; I'_{12} = -\frac{f'}{j} + \frac{\sqrt{n}}{j}.$$

Step 5. Let

$$I_1 = [I_{11}, I_{12}], \;\; I'_1 = [I'_{11}, I'_{12}], \; if \; b > 0,$$

and

$$I_1 = [I_{12}, I_{11}], \;\; I'_1 = [I'_{12}, I'_{11}], \; if \; b < 0.$$

Step 6. Compute

$$I_{21} = -\frac{d'\lambda_1 - w'n}{a} - \frac{\sqrt{n}}{a}, \; I_{22} = -\frac{d'\lambda_1 - w'n}{a} + \frac{\sqrt{n}}{a}.$$

Also,

$$I'_{21} = -\frac{f'\lambda_2 - v'n}{g} - \frac{\sqrt{n}}{g}, \; I'_{22} = -\frac{f'\lambda_2 - v'n}{g} + \frac{\sqrt{n}}{g}.$$

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 α_1, also all integers in the intersection of I'_1 and I'_2 and define them by α_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

$$c = w'n - d'\lambda_1 + \alpha_1 a, \;\; d = d' + \alpha_1 b$$

and

$$e = v'n - f'\lambda_2 + \alpha_2 g, \;\; f = f' + \alpha_2 j.$$

The vectors $v_2 = (c,d)$ and $v_4 = (e, f)$ are easily verify to be in the $kerT$, and $|c|, |d|, |e|$ and $|f| < \sqrt{n}$; therefore, $\{v_1, v_2\}$ and $\{v_3, v_4\}$ are ISD generators.

5.3 The proposed integer sub-decomposition method (ISDM) to Compute kP

The proposed method modified the Gallant, Lambert, Vanstone GLV method (Gallant et al., 2001) to have faster point multiplication on an elliptic curve E over a prime finite field F_p. This modification embeds that the second decomposition of the values k_1 and k_2 when one or both values is not bounded by $\pm\sqrt{n}$. The sub-decomposition from $k = k_1 + k_2\lambda_2 \; (mod \; n)$ is explained in the following:

$$k_1 = k_{11} +_E k_{12}\lambda_1 \; (mod \; n) \qquad (26)$$

and

$$k_2 = k_{21} +_E k_{22}\lambda_2 \; (mod \; n). \qquad (27)$$

One has to find ISD generators $\{v_1, v_2\}$ and $\{v_3, v_4\}$ based on the algorithm (1) that depends on the same way followed by a GLV generator algorithm [17], so that each component of v_1, v_2, v_3 and v_4 is bounded by $\sqrt{n}$. Accordingly, the result will be integers k_{11}, k_{12}, k_{21} and k_{22} are computed by solving the closest vector problem in lattice which is embodied in using an extended Euclidean algorithm. That is, one can decompose k through applying the balanced length-two representation of a sub-decomposition multiplier algorithm (2) as follows:

$$k = k_{11} + k_{12}\lambda_1 + k_{21} + k_{22}\lambda_2 \ (mod\ n) \qquad (28)$$

with $-\sqrt{n} < k_{11}, k_{12}, k_{21}, k_{22} < \sqrt{n}$ from any ISD generators $\{v_1, v_2\}$ and $\{v_3, v_4\}$. The Fig (1) shows that clearly.

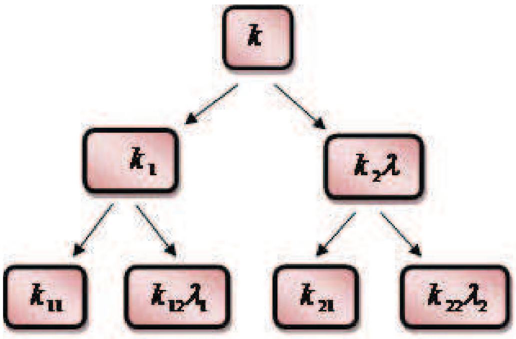

Fig. 1 Shows the Subdecomposition of the integer k

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 $|k_{11}|, |k_{12}|, |k_{21}|, |k_{22}| < \sqrt{n}$.

Step 1: Run ISD generators algorithm (1) with inputs n, λ_1 and λ_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

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

Else

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

Step 4: Compute $c_3 = \lfloor \bar{v}k_1/n \rceil$, $c_4 = \lfloor -\bar{t}k_1/n \rceil$ and $c_5 = \lfloor \hat{v}k_2/n \rceil$, $c_6 = \lfloor -\hat{t}k_2/n \rceil$.

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}.

Hence, kP can be calculated by using the following formula:

$$\begin{aligned} kP &= k_{11}P + k_{12}[\lambda_1]P + k_{21}P + k[\lambda_2]P \\ &= k_{11}P + k_{12}\psi_1(P) + k_{21}P + k_{22}\psi 2(P). \end{aligned} \qquad (29)$$

The computation in equation (29) can be achieved using the window simultaneous multiple point multiplication method which has been computed in an algorithm (3) for P, $\psi_1(P)$ and $\psi_2(P)$.One can see that in the Fig (2).

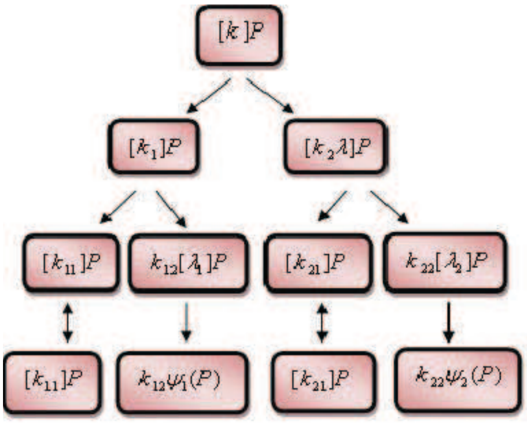

Fig. 2 Shows the Subdecomposition of the elliptic curve point multiplication kP

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, $\lambda_1, \lambda_2 \in Z$, 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

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

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

Step 4: Let $l_z = max\{l_{z,1}, l_{z,2}\}$, $z = 1,2$.

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 $i \in \{1,3,\ldots,2^{w_j-1}-1\}$ and $i \in \{1,3,\ldots,2^{w_s-1}-1\}$, 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.

To summarize, the ISD method involves applying the same method as in the original GLV for finding the GLV generator $\{v_1, v_2\}$ for the given n and λ by using the GLV generator algorithm in [17]. Accordingly, the result will decompose k into k_1 and k_2 for n, λ and $k \in [1, n-1]$. This step can be done using the balanced length-two representation of a multiplier algorithm in [11]. Depending on the algorithm (1), find ISD generators $\{v_3, v_4\}$ and $\{v_5, v_6\}$ such that each component of v_3, v_4, v_5 and v_6 is bounded by $\sqrt{n}$. The result will sub-decompose k_1 and k_2 into the integers k_{11}, k_{12}, k_{21} and k_{22} which are computed by solving the closest vector problem in lattice that is embodied in using an extended Euclidean algorithm. k is sub-decomposed by applying the algorithm (2) to find the equation (28) with $-\sqrt{n} < k_{11}, k_{12}, k_{21}, k_{22} < \sqrt{n}$ from any ISD generators $\{v_3, v_4\}$ and $\{v_5, v_6\}$. Eventually, kP can be calculated by using the formula in the equation (29), See algorithm (4).

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 [17] to find the generator $\{v_1,\ v_2\}$ for the given n and λ 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 [11] to decompose k to find k_1 and k_2 for a given n, λ and $k \in [1, n-1]$.

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

Step 3: Use algorithm (2) to find

3.1. For n and λ_1, generate the ISD generator $\{v_3, v_4\}$ such that $v_3 \leftarrow (\overline{r}, \overline{t})$ and $v_4 \leftarrow (\overline{u}, \overline{v})$.
3.2. For n and λ_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

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

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

6 Results

The sub-decomposition method proposed in this paper, known as the ISD method, is a modification of the GLV method introduced by Gallant et al. to compute scalar multiplication kP. In the original GLV method, the decomposition of the integer k into k_1 and k_2 assumes only values lying in the range $-\sqrt{n} < k1, k2 < \sqrt{n}$. For those values fall outside the range, that is, the values of k_1 and k_2 which lie outside the range $\sqrt{n}$, the GLV method will not work. This has resulted in a low percentage of successful computation of the multiplication operation of kP; hence GLV method is limited by the value of k for kP computation. To solve this problem, we have proposed new algorithms to anticipate those values of k_1 and k_2 that lie outside the range. In the ISD method, we have sub-decomposed the values of k_1 and k_2 into k_{11}, k_{12}, k_{21} and k_{22} which lie within the range of $\sqrt{n}$. The proposed ISD method aimed to increase the percentage of successful computation of kP.

The experimental results of the proposed method is indicated in Table 1. In our experiment, we have implemented the ISD method on four sets of sample data. We considered three parameters in the experiment: the parameter n, the parameter λ, and the prime p with 100-bit length. The results indicate clearly that the ISD method significantly increases the percentage of successfully computed kP. The ISD method has helped increase the successful computation of kP by 50%

comparison with the original GLV method, resulting in a more reliable method for kP computation. This improvement has a direct impact on the scalar multiplication techniques in elliptic curve cryptography, and will promote the application of the elliptic curve cryptosystem in the today's modern world. Below are the experimental results on the percentage of successful computation of kP for both the GLV and the ISD methods.

Table 1 Percentage of successful computation of kP for both the original GLV and the proposed ISD method

$\%of\ GLV\ Method$	$\%of\ proposed\ ISD\ Method$
0.0455	0.3939
0.0269	0.4462
0.0116	0.4701
0.0065	0.4879

7 Conclusion

The present paper was concerned with presenting new algorithms to facilitate the use of Gallant et al.'s idea for speeding up the scalar multiplication of elliptic curves over a prime field in a more concrete way when the two values or one of the decomposing integers is not bounded by $\sqrt{n}$. This new method, namely, the integer sub-decomposition method, ISD will help increase the success rate of kP computation by 50% compared with the GLV method.

References

[1] V. Miller, ”Use of elliptic curves in cryptography,” in *Advances in CryptologyCRYPTO85 Proceedings*, 417-426 (1986).

[2] N. Koblitz, ”Elliptic curve cryptosystems,” *Mathematics of computation*, **48**, 203-209 (1987).

[3] Y. Hitchcock and P. Montague, ”A new elliptic curve scalar multiplication algorithm to resist simple power analysis,” in Information Security and Privacy, 214-225 (2002).

[4] Y. Hao, S. Ma, G. Chen, X. Zhang, H. Chen, and W. Zeng, ”Optimization Algorithm for Scalar Multiplication in the Elliptic Curve Cryptography over Prime Field,” in *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues*, ed: Springer, 904-911 (2008).

[5] P. Longa and C. Gebotys, ”Efficient techniques for high-speed elliptic curve cryptography,” in *Cryptographic Hardware and Embedded Systems, CHES 2010*, ed: Springer, 80-94 (2010).

[6] R. R. Farashahi, H. Wu, and C.-A. Zhao, ”Efficient arithmetic on elliptic curves over fields of characteristic three,” in Selected Areas in Cryptography, 135-148 (2013).

[7] P. Longa and A. Miri, ”Accelerating scalar multiplication on elliptic curve cryptosystems over prime fields,” ed: Google Patents, (2011).

[8] H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, et al., Handbook of elliptic and hyperelliptic curve cryptography: *Chapman & Hall/CRC, London*, (2005).

[9] R. Shi and J. Cheng, ”Two new fast methods for simultaneous scalar multiplication in elliptic curve cryptosystems,” in Networking and Mobile Computing, ed: Springer, 462-470 (2005).

[10] C. Negre, ”Scalar multiplication on elliptic curves defined over fields of small odd characteristic,” in Progress in Cryptology-INDOCRYPT 2005, ed: Springer, 389-402 (2005).

[11] D. Hankerson, A. J. Menezes, and S. Vanstone, Guide to elliptic curve cryptography: *Springer*, (2004).

[12] R. Barua, S. K. Pandey, and R. Pankaj, ”Efficient window-based scalar multiplication on elliptic curves using double-base number system,” in *Progress in CryptologyINDOCRYPT 2007*, ed: Springer, 351-360 (2007).

[13] D. Liu, Z. Tan, and Y. Dai, ”New elliptic curve multi-scalar multiplication algorithm for a pair of integers to resist SPA,” in *Information Security and Cryptology*, 253-264 (2009).

[14] J. A. Solinas, ”Efficient arithmetic on Koblitz curves,” *Design, Codes and Cryptography*, **19**, 195-249 (2000).

[15] R. Gallant, R. Lambert, and L. Vanstone, ”Faster point multiplication on elliptic curves with efficient endomorphisms”, *in Advances in Cryptology-CRYPTO 2001*, 190-201 (2001).

[16] M. Ciet, J.-J. Quisquater, and F. Sica, ”Preventing differential analysis in GLV elliptic curve scalar multiplication,” *Cryptographic Hardware and Embedded Systems-CHES 2002*, 1-13 (2003).

[17] D. Kim, S. Lim, ”Integer decomposition for fast scalar multiplication on elliptic curves”, *in Selected Areas in Cryptography, Springer, Berlin, **2003***, 13-20 (2003).

[18] P. Longa and F. Sica, ”Four-Dimensional gallant-lambert-vanstone scalar multiplication,” *in Advances in CryptologyASIACRYPT 2012*, Springer, 718-739 (2012).

[19] D. Venturi, ”Lecture Notes on Algorithmic Number Theory,” (2000).

[20] L. C. Washington, Elliptic curves: *number theory and cryptography*, Chapman & Hall/CRC, **50**, (2008).

Ruma Ajeena PhD Scholar in Mathematical Sciences School at University Sciences Malaysia. She received the M.Sc. degree in Cryptography and number theory from Babylon University, Iraq in 2006. Her research interests are in the areas of Applied Mathematics, Cryptography, Coding Theory, Number Theory and Abstract Algebra. She has presented her research findings in many international conferences. She has published research articles in international journals of cryptology research and Mathematical Sciences journals.

Hailiza Kamarulhaili Associate Professor Hailiza Kamarulhaili received her PhD from the University of Liverpool, England in 1999. Her research interest is in Number Theory, Cryptography and Analysis. She has published more than 100 articles in refereed journals, conference proceedings and a chapter in book. She has presented her research findings in many international conferences and workshops. She is a committee member of the Malaysian Society for Cryptology Research, a reciprocity Member of The London Mathematical Society, a life member of the International Society for Analysis, its Applications and Computation (ISAAC). She is also a member of the International Association for Cryptologic Research(IACR) and the Malaysian Society for Mathematical Sciences (PERSAMA).