On Efficient Computations of Koblitz Curves over Prime Fields

Guangwu Xu[*], Ke Han[†], Yunxiao Tian[‡]

Abstract

The family of Koblitz curves $E_b : y^2 = x^3 + b/\mathbb{F}_p$ over primes fields has notable applications and is closely related to the ring $\mathbb{Z}[\omega]$ of Eisenstein integers. Utilizing nice facts from the theory of cubic residues, this paper derives an efficient formula for a (complex) scalar multiplication by $\tau = 1-\omega$. This enables us to develop a window τ-NAF method for Koblitz curves over prime fields. This probably is the first window τ-NAF method to be designed for curves over fields with large characteristic. Besides its theoretical interest, a higher performance is also achieved due to the facts that (1) the operation τ^2 can be done more efficiently that makes the average cost of τ to be close to $2.5\mathbf{S} + 3\mathbf{M}$ ($\mathbf{S}$ and $\mathbf{M}$ denote the costs for field squaring and multiplication, respectively); (2) the pre-computation for the window τ-NAF method is surprisingly simple in that only about one-sixth of the coefficients need to be processed. The overall improvement over the best current method is more than 11%. The paper also suggests a simplified modular reduction for Eisenstein integers where the division operations are eliminated. The efficient formula of τP can be further used to speed up the computation of $3P$, compared to $10\mathbf{S} + 5\mathbf{M}$, our new formula just costs $4\mathbf{S}+6\mathbf{M}$. As a main ingredient for double base chain method for scalar multiplication, the $3P$ formula will contribute to a greater efficiency.

1 Introduction

Some families of elliptic curves were proposed by Koblitz for cryptography because of their computational efficiency [13, 14]. These curves are defined over $\mathbb{F}_q$ for q

[*]SCST, Shandong University, China, e-mail: gxu4sdq@sdu.edu.cn(Corresponding author)
[†]SCST, Shandong University, China, e-mail: 202237084@mail.sdu.edu.cn
[‡]SCST, Shandong University, China, e-mail: 202337040@mail.sdu.edu.cn

relatively small, and a subgroup of the set of rational points over $\mathbb{F}_{q^n}$ is of interest. The well-known family of Koblitz curve is given by

$$E_a : y^2 + xy = x^3 + ax^2 + 1/F_{2^m},\ a \in \mathbb{F}_2. \tag{1}$$

There are four specific Koblitz curves were recommended to be used Elliptic curve cryptography (ECC) by NIST [1].

The Frobenius map τ_0 of $E_a(\mathbb{F}_{2^m})$ defined by $\tau_0(x, y) = (x^2, y^2)$ (for the point at infinity, $\tau_0(\mathcal{O})$ is set to be $\mathcal{O}$) is an endomorphism that only requires $2\mathbf{S}$[1]. The use of this efficiently computable endomorphism was initiated in [13] for fast scalar multiplication. Write $\mu = (-1)^{1-a}$, then for each point $P \in E_a(\mathbb{F}_{2^m})$,

$$\tau_0^2(P) + 2P = \mu\tau_0(P).$$

This means that τ_0 can be identified as the complex number satisfying $\tau_0^2 - \mu\tau_0 + 2 = 0$. Working with subgroup M of $E_a(F_{2^m})$ that is annihilated by $\delta = \frac{\tau_0^m - 1}{\tau_0 - 1}$ (i.e., $\delta(P) = \mathcal{O}$ for every $P \in M$), then a rational integer k can be written as $k \equiv \sum_{i=0}^{l-1} \epsilon_i \tau_0^i \pmod{\delta}$ in $\mathbb{Z}[\tau_0]$ with $\epsilon_i \in \{0, 1\}$, so the scalar multiplication kP can be computed as $\sum_{i=0}^{l-1} \epsilon_i \tau_0^i(P)$ for $P \in M$.

This idea was substantially developed to the celebrated width-w TNAF method (window τ_0-adic non-adjacent form of size w) by Solinas in [18]. The main ingredients of width-w TNAF are reduction, sparse τ_0-expansion and pre-computation. Solinas also devises a reduction procedure that converts an integer k to an element $k_1 + k_2\tau_0 \in \mathbb{Z}[\tau_0]$ such that $k \equiv k_1 + k_2\tau_0 \pmod{\delta}$ and the sizes of k_1 and k_2 are about $\frac{m}{2}$. The sparse τ_0-expansion relies on a pre-selected set $C = \{c_1, c_3, \ldots, c_{2^{w-1}-1}\}$ of coefficients with $c_i \equiv i \pmod{\tau_0^w}$, so that a reduction result has the following sparse form

$$k_1 + k_2\tau_0 = \sum_{i=0}^{l-1} \varepsilon_i u_i \tau_0^i,$$

where $\varepsilon_i \in \{-1, 1\}$ and $u_i \in C \cup \{0\}$ with the property that any set $\{u_k, u_{k+1}, \ldots, u_{k+w-1}\}$ contains at most one nonzero element. This implies that for $P \in M$,

$$kP = k_1P + k_2\tau_0(P) = \sum_{i=0}^{l-1} \epsilon_i u_i \tau_0^i(P) = \sum_{i=0}^{l-1} \epsilon_i \tau_0^i(u_i P).$$

[1]We will use $\mathbf{S}$ and $\mathbf{M}$ to denote the costs for field squaring and multiplication, respectively.

The pre-computation produces $c_1P, c_3P, \cdots, c_{2^{w-1}-1}P$ once the set C is given, so u_iP's in the above computation are already available. Several coefficient sets for pre-computation were reported in [18, 10].

A series further study of the width-w TNAF method are given in [6, 19, 20, 22] where a theoretical framework for (sparse) τ_0-adic representation of integers in $\mathbb{Q}(\tau_0)$ was developed. The framework directs selection of coefficient sets for a width-w T-NAF, and the termination problem for the algorithm to produce a width-w TNAF is resolved. Greater flexibility for choosing coefficient sets is provided in this framework, several efficient pre-computations are proposed. A simplified reduction is also reported.

In [14], Koblitz considered the following supersingular elliptic curve over finite fields of characteristic 3,

$$E_{3,a} : y^2 = x^3 - x - (-1)^a / \mathbb{F}_{3^m},\ a \in \{0, 1\}. \tag{2}$$

The Frobenius map for this case is given as $\tau_1(x, y) = (x^3, y^3)$ (for the point at infinity, $\tau_1(\mathcal{O})$ is set to be $\mathcal{O}$) is also efficiently computable, with $2\mathbf{S} + 2\mathbf{M}$. Similar to the case for binary Koblitz curves, this τ_1 is identified to be a complex number satisfying

$$\tau_1^2 - 3\mu\tau_1 + 3 = 0.$$

The nonadjacent form of τ_1-adic expansion for an integer in $\mathbb{Z}[\tau_1]$ is proved to exist and be unique in [14]. The scalar multiplication using width-w TNAF method for $E_{3,a}$ is described in [5] where a new design of reduction and a termination proof of the width-w TNAF algorithm are obtained.

The mathematical essence of a complete treatment of both width-w TNAF methods for Koblitz curves E_a over $\mathbb{F}_{2^m}$ and $E_{3,a}$ over $\mathbb{F}_{3^m}$ is the representation of algebraic integers in Euclidean imaginary quadratic number fields, with radix being some algebraic integers of norm bigger than 1. This was discussed in [7] as a generalization of the ideas in [18, 6, 5].

The curves over prime fields $\mathbb{F}_p$ with large p and with a restricted set of coefficients are now of practical interest. This paper will discuss the family of curves over a prime field $\mathbb{F}_p$ that take the form of

$$E_b : y^2 = x^3 + b / \mathbb{F}_p, \tag{3}$$

where the prime $p \equiv 1 \pmod 3$ [2] and $b \in \mathbb{F}_p^*$. This family of curves is referred to as Koblitz curves because it is a special case of CM curves with simple expression. One of such Koblitz curves described in the Standards for Efficient Cryptography Group (SECG)[3] is

$$\text{secp256k1:} \quad y^2 = x^3 + 7/\mathbb{F}_p$$

where $p = 2^{256} - 2^{32} - 977$ is a prime of 256 bits. This curve has been chosen by some applications (e.g., digital signatures for blockchain platforms such as Bitcoin) and is an allowed curve by the NIST Recommendations for Discrete Logarithm-based Cryptography [2].

It seems that the Koblitz curves E_b over prime fields and the Koblitz curves E_a (or $E_{3,a}$) over extension fields with characteristic 2 (or 3) are quite different in nature, but they appear to have some similarities. Of course, they are all CM curves. Recently, another interesting observation on the similarities is made in [21]. For a rational prime $p \equiv 1 \pmod 3$, there are integers c, d such that $p = c^2 - cd + d^2$. In other words, $p = N(\pi)$, the norm of prime $\pi = c + d\omega$ in the ring $\mathbb{Z}[\omega]$ of Eisenstein integers (we may further require π to be primary in the sense that $c \equiv 2 \pmod 3$ and $d \equiv 0 \pmod 3$). Based on a point counting formula of Rajwade for $E_b/\mathbb{F}_p$ [17] and some calculations of cubic residues, the following form is given in [21]

$$\#E_b(\mathbb{F}_p) = N(\pi - u), \ \#\mathbb{F}_p = N(\pi). \tag{4}$$

where u is a unit in $\mathbb{Z}[\omega]$. We put the trivial fact $\#\mathbb{F}_p = N(\pi)$ here is for relating the number of points and the cardinality of the underlying field, as well as for comparing with the binary case below. The Koblitz curve $E_a/\mathbb{F}_{2^m}$ has coefficients in the subfield $\mathbb{F}_2$. This allows efficient point counting via the zeta function (e.g. [15]). More precisely, write $q = 2^m$, $\alpha = \tau_0^m$, then we see that

$$\#E_a(\mathbb{F}_q) = N(\alpha - 1), \ \#\mathbb{F}_q = N(\alpha). \tag{5}$$

It is remarked that the forms (4) and (5) are similar, and both of them indicate a nice relation between the cardinalities of the rational points of the curve and the underlying field. However, (4) and (5) are obtained with completely unrelated process of derivations.

[2] We note that when $p \equiv 2 \pmod 3$, $E_b/\mathbb{F}_p$ is known to be a supersingular curve and the group $E_b(\mathbb{F}_p)$ is a cyclic group of order $p + 1$.

For arithmetic of curves over prime fields, we do not know whether powerful methods like width-w TNAF are available. But for the Koblitz curve E_b over a prime field $\mathbb{F}_p$ with $p \equiv 1 \pmod 3$, GLV method [12] can be used to speed up its scalar multiplication. To give an account of this, let us write $n = \#E_b(\mathbb{F}_p)$ and assume that n is prime (by choosing b and p properly if necessary). In this case, $n \equiv 1 \pmod 3$ holds. There exist cubic roots of unity $\beta \in \mathbb{F}_p$ and $\lambda \in \mathbb{F}_n$ such that for each $P = (x, y) \in E_b(\mathbb{F}_p)$

$$\lambda P = (\beta x, y). \tag{6}$$

This means that λP (with λ being treated as a scalar) can be computed efficiently with a cost of field multiplication. Utilizing (6), the GLV method works as follows. To compute kP, a reduction is performed to get integers k_1, k_2 of size about $\sqrt{n}$, such that

$$k \equiv k_1 + k_2\lambda \pmod n. \tag{7}$$

Then, kP can be obtained by computing $k_1P + k_2Q$ by using simultaneous multiple scalar multiplication or interleaving [10], where $Q = \lambda P$.

This paper makes a further study of efficient computations for Koblitz curves E_b over prime fields. Identifying the scalar λ with complex number $\omega = \frac{-1+\sqrt{-3}}{2}$, we may work with the ring of Eisenstein integers $\mathbb{Z}[\omega]$ and utilize many nice mathematical results of the subject. Among the results obtained in this paper, we are able to create a width-w TNAF method for Koblitz curves E_b over prime fields where the operation τ is given by $\tau = 1 - \omega$. Even though using different approaches, it does share similarities with the cases for Koblitz curves E_a (or $E_{3,a}$) over extension fields with characteristic 2 (or 3). To be more specific, our main results include

1. Creating an simple reduction procedure for (7).

2. Designing an efficient formula for τP, it costs $3\mathbf{M} + 3\mathbf{S}$ under the Jacobian projective coordinate. Based on the formula for τP, a more efficient way of computing $\tau^2 P$ is suggested, and its cost is $6\mathbf{M} + 5\mathbf{S}$. An efficient formula for $3P$ is also obtained with a cost of $6\mathbf{M} + 4\mathbf{S}$ under the Jacobian projective coordinate.

3. Developing a width-w window τ-NAF (width-w TNAF) method for Koblitz curves E_b over prime field $\mathbb{F}_p$ with $p \equiv 1 \pmod 3$. A pre-computation is carefully chosen and six units of $\mathbb{Z}[\omega]$ are better used to significantly reduce the cost.

We would like to make some remarks.

- Our simple reduction procedure is based on an idea we used in 2002 for implementing binary Koblitz curves $E_b(\mathbb{F}_{2^m})$ that simplifies Solinas' reduction [18] whose efficiency has also been validated in [19]. For using the GLV method in the Koblitz curves over prime field, Brown, Myers and Solinas suggested a reduction [8] with an approach simiar to that in [18] (see also [11]). Our simple reduction procedure is applicable with GLV as well.

- The fact that operation τP has a low cost implies that an efficient window τ-NAF can be created. For the case of window τ-NAF, we can frequently use τ^2 that reduces the average cost of τ to close to $3\mathbf{M} + 2.5\mathbf{S}$. This is comparable to the Frobenius map for subfield curves over fields of characteristic 5, even though τ does not have a meaningful algebraic property as that of the Frobenius maps τ_0, τ_1.

- Pre-computation is surprisingly efficient because multiplications by units of $\mathbb{Z}[\omega]$ saves about five-sixth of the cost.

- The overall improvement over the best current method is more than 11%.

The rest of our paper is arranged into four sections. Section 2 provides some preliminaries and develops some tools. The main Algorithms and some discussions are given in the section 3. We conclude the paper in section 4.

2 Preliminaries and Tools

The curves we discuss in the paper are of the form (3):

$$E_b : y^2 = x^3 + b/\mathbb{F}_p,$$

where the prime $p \equiv 1 \pmod 3$ and $b \in \mathbb{F}_p^*$. Modulo such a prime p, there is a primitive cubic unity β, i.e., $\beta^3 \equiv 1 \pmod p$.

Denote $n = \#E_b(\mathbb{F}_p)$. In this paper, we just consider the case that n is prime. It can be seen that $n \equiv 1 \pmod 3$ (e.g., from (4)), so there is a primitive cubic unity λ modulo n. We can choose λ such that

$$\lambda(x, y) = (\beta x, y)$$

for all points $(x, y) \in E_b(\mathbb{F}_p) \setminus \{\mathcal{O}\}$. From this, it is immediate that

$$\lambda^2(x, y) = (\beta^2 x, y).$$

For such a prime n, and a primitive cubic unity λ, we form the lattice

$$\Lambda(\lambda, n) := \{(x, y) \in \mathbb{Z} \times \mathbb{Z} \mid \lambda x + y \equiv 0 \pmod{n}\},$$

and find a non-zero shortest vector $\mathbf{v} = (c, d) \in \Lambda(\lambda, n)$ using the Lagrange-Gauss algorithm for two dimensional lattice [9], one gets

$$n = c^2 - cd + d^2, \tag{8}$$

in other words, $n = N(c + d\omega)$.

Jacobian Projective Coordinates: For every $(X, Y, Z) \in \mathbb{F}_p^3$, Jacobian projective coordinate $(X : Y : Z)$ is the following equivalent class defined over $\mathbb{F}_p^3$

$$(X : Y : Z) = \{(\lambda^2 X, \lambda^3 Y, \lambda Z) : \lambda \in \mathbb{F}_p^*\}.$$

If $Z \neq 0$, Jacobian projective coordinate $(X : Y : Z)$ corresponds to the affine coordinate $(\frac{X}{Z^2}, \frac{Y}{Z^3})$. The equation of $E_b/\mathbb{F}_p$ under the Jacobian projective coordinate becomes

$$E_b : Y^2 = X^3 + bZ^6, \tag{9}$$

with $(1 : 1 : 0)$ being the point of infinity.

Under projective coordinate, if two non-trivial point of $E(\mathbb{F}_p)$

$$P = (X_P : Y_P : Z_P), Q = (X_Q : Y_q : Z_Q)$$

satisfy $P \neq -Q$ and if their sum is $R = (X_R : Y_R : Z_R)$, then putting

$$(\frac{X_P}{Z_P^2}, \frac{Y_P}{Z_P^3}), (\frac{X_Q}{Z_Q^2}, \frac{Y_Q}{Z_Q^3})$$

in the addition formula in affine coordinate,

1. if $P \neq \pm Q$, we have $\begin{cases} X_R = (Y_Q Z_P^3 - Y_P Z_Q^3)^2 \\ \qquad -(X_Q Z_P^2 - X_P Z_Q^2)^2 (X_P Z_Q^2 + X_Q Z_P^2), \\ Y_R = (Y_Q Z_P^3 - Y_P Z_Q^3)\big(X_P Z_Q^2 (X_Q Z_P^2 - X_P Z_Q^2)^2 - X_R\big) \\ \qquad -Y_P Z_Q^3 (X_Q Z_P^2 - X_P Z_Q^2)^3, \\ Z_R = (X_Q Z_P^2 - X_P Z_Q^2) Z_P Z_Q. \end{cases}$

2. if $P = Q$, we have $\begin{cases} X_R = 9X_P^4 - 8X_P Y_P^2, \\ Y_R = 3X_P^2(4X_P Y_P^2 - X_R) - 8Y_P^4, \\ Z_R = 2Y_P Z_P. \end{cases}$

In Jacobian projective coordinates, point addition and doubling for Koblitz curves over prime fields can be performed by using the procedures in [10]. So the cost for addition in Jacobian coordinates is $4\mathbf{S} + 12\mathbf{M}$, and the cost for doubling is $4\mathbf{S} + 3\mathbf{M}$. In [16], point addition is improved by trading a multiplication with a squaring. The costs for point addition and point doubling for Koblitz curves over prime fields, as described in table 1, are

$$1\mathbf{ADD} = 5\mathbf{S} + 11\mathbf{M},\ 1\mathbf{DBL} = 4\mathbf{S} + 3\mathbf{M}.$$

Table 1: Point Addition and Doubling ($y^2 = x^3 + b$, Jacobian coordinates)

Addition Procedure of E	**Doubling Procedure of** E
$R = P + Q\ (P \neq \pm Q)$	$R = 2P$
$A = X_P Z_Q^2 - X_Q Z_P^2$;	$A = 3X_P^2$;
$B = Y_P Z_Q^3 - Y_Q Z_Q^3$;	$B = 2Y_P$;
$C = 2X_Q Z_P^2 A^2$;	$C = B^2$;
$D = 4Y_Q Z_P^3 A^3$	$D = CX_P$;
$X_R = 4(B^2 - A^3 - C)$;	$X_R = A^2 - 2D$;
$Y_R = 2(B(2C - X_R) - D)$;	$Y_R = (D - X_R)A - \frac{C^2}{2}$;
$Z_R = ((A+Z_Q)^2 - A^2 - Z_Q^2)Z_P$;	$Z_R = BZ_P$;

In this paper, we shall adopt the conversion $1\mathbf{S} = 0.8\mathbf{M}$ suggested in [4][3]. This means that

$$1\mathbf{ADD} \approx 15\mathbf{M},\ 1\mathbf{DBL} \approx 6.2\mathbf{M}.$$

[3]It is remarked that for the purpose of countering side-channel attack, $1\mathbf{S} = 1\mathbf{M}$ should be enforced in the implementation. In this case, our method could show more improvement.

2.1 Reduction for Decomposing a Scalar

Recall that we use n to denote the number of points of $E_b/\mathbb{F}_p$. We know that λ is a primitive cubic root of unity of n. In this subsection, we explain a procedure that, for each integer $k < n$, generates integers k_1, k_2 about size $\sqrt{n}$ such that

$$k \equiv k_1 + \lambda k_2 \pmod{n}.$$

We have from (8) that

$$n = c^2 - cd + d^2$$

Note that $1 + \lambda + \lambda^2 = 0 \pmod{n}$, we see that

$$\begin{aligned}(c + d\lambda)(c + \lambda^2 d) &= c^2 + (\lambda + \lambda^2)cd + \lambda^3 d^2 \\ &\equiv c^2 - cd + d^2 \pmod{n} \equiv 0 \pmod{n}.\end{aligned}$$

We may assume that

$$c + d\lambda \equiv 0 \pmod{n}.$$

Otherwise if $c + \lambda^2 d \equiv 0 \pmod{n}$, then $c - d - d\lambda \equiv 0 \pmod{n}$. We may substitute $(c - d, -d)$ by (c, d).

Let q_1, q_2 be integers and set

$$\alpha_1 = \frac{k(c-d)}{n} - q_1,\ \alpha_2 = \frac{-kd}{n} - q_2.$$

Write

$$k_1 + k_2\omega = (\alpha_1 + \alpha_2\omega)(c + d\omega),$$

then we have

$$\begin{aligned}k &= \frac{k}{c + d\omega}(c + d\omega) = \frac{k(c + d\overline{\omega})}{n}(c + d\omega) \\ &= \frac{k(c-d) - kd\omega}{n}(c + d\omega) = ((q_1 + q_2\omega) + (\alpha_1 + \alpha_2\omega))\,(c + d\omega) \\ &= (q_1 + q_2\omega)(c + d\omega) + (k_1 + k_2\omega).\end{aligned}$$

This implies that k_1, k_2 are integers and

$$k \equiv k_1 + \lambda k_2 \pmod{n}.$$

It can be seen that

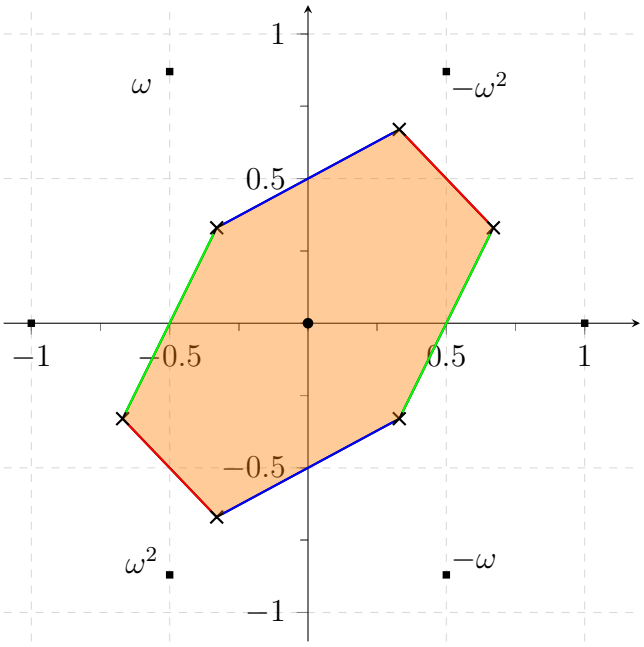
$$\begin{aligned} k_1 &= k - q_1 c + q_2 d, \\ k_2 &= q_2 d - q_1 d - q_2 c \end{aligned} \tag{10}$$

Since $k_1 + k_2\omega = (\alpha_1 + \alpha_2\omega)(c + d\omega)$, to make $k_1 + k_2\omega$ small, the Eisenstein integer quotient $(q_1 + q_2\omega)$ should be obtained by rounding the rational coefficients $\frac{k(c-d)}{n}$ and $\frac{-kd}{n}$ to the nearest integers with respect to some appropriate distance. For Koblitz curves over prime fields, a method proposed in [8] (see also [11]) is similar to that in [18] by using Voronoi cell, but the latter is for the case of Koblitz curves over binary fields.

It should be noted that this Voronoi cell is with respect to norm N in the sense that for an point (x, y) in the interior of a Voronoi cell $\mathcal{U}$,

$N(x + y\omega) < N((x + s) + (y + t)\omega)$

for all pairs of integers $(s, t) \neq (0, 0)$. The Voronoi cell for the origin is given on the right. It is bounded by lines $x + y = \pm 1$ (in red), $2y - x = \pm 1$ (in blue) and $2x - y = \pm 1$ (in green).



Now, denoting $x = \frac{k(c-d)}{n}$ and $y = \frac{-kd}{n}$, then the precise values of q_1, q_2 given in [8] are

$$q_1 = \left\lfloor \frac{\lfloor x + y \rfloor + \lfloor 2x - y \rfloor + 2}{3} \right\rfloor, \quad q_2 = \left\lfloor \frac{\lfloor x + y \rfloor + \lfloor 2y - x \rfloor + 2}{3} \right\rfloor. \tag{11}$$

Since c, d and n are fixed, the main calculation of q_1 and q_2 requires 2 integer multiplication and 2 divisions, and the integers involved in the division are large.

We propose a simplification of the calculation by using a one-time pre-computations

(once for the curve) so that the divisions can be removed. To this end, we first let

$$s = \lceil \log_2 n \rceil,\ t = \lceil \log_2 2(|c| + |d|) \rceil,$$

and pre-compute

$$\eta_1 = \left\lfloor \frac{(c-2d)2^{s+t}}{n} \right\rfloor,\ \eta_2 = \left\lfloor \frac{(2c-d)2^{s+t}}{n} \right\rfloor,\ \eta_3 = \left\lfloor \frac{(-c-d)2^{s+t}}{n} \right\rfloor.$$

Then for a given $0 \leq k < n$, we compute

$$f_j = \left\lfloor \frac{k\eta_j}{2^{s+t}} \right\rfloor,\ j = 1, 2, 3.$$

and finally get q_1, q_2.

$$q_1 = \left\lfloor \frac{f_1 + f_2 + 2}{3} \right\rfloor,\ q_2 = \left\lfloor \frac{f_1 + f_3 + 2}{3} \right\rfloor. \tag{12}$$

We make some remarks on the simplified procedure.

Remarks

1. It is seen that we need to compute two 3 multiplications $k\eta_1, k\eta_2, k\eta_3$. The division by 2^{s+t} is trivial.

2. The integers q_1, q_2 are close to x, y respectively. For example, it is easy to verify that
$$3x - \frac{1}{2^{t-1}} \leq f_1 + f_2 + 2 < 3x + 2.$$

3. For binary Koblitz curves, it has been pointed out in [19] that in practice one can simply using the floors of the rational coefficients in the reduction, without affecting efficiency. In our case, we can take the following
$$q_1 = \left\lfloor \frac{k(c-d)}{n} \right\rfloor,\ q_2 = \left\lfloor \frac{-kd}{n} \right\rfloor. \tag{13}$$
Our experiment shows that the performance is the same as that of using (11) or (12). It is remarked that (13) can be also refined by using pre-computation approach .

2.2 Efficient Formulas for $\tau P, \tau^2 P$ and $3P$

In the ring of Eisenstein integers, the number 3 is of special interest. It is associated to the square of the prime $\tau = 1-\omega$, i.e., $3 = -\omega^2\tau^2$. As mentioned earlier, for scalar multiplication ωP of a point P by ω is meaningful which is identified as λP. In this subsection, we derive three efficient formulas for scalar multiplication by τ, τ^2 and by 3. To be more specific, we have

Proposition 2.1. *Let $P = (x, y) \in E_b(\mathbb{F}_p)$.*

1. *The affine coordinates of $\tau P = (1 - \lambda)P$ are*

$$\begin{cases} x' = \frac{x^3+4b}{(1-\beta)^2x^2} \\ y' = y\frac{x^3-8b}{(1-\beta)^3x^3}. \end{cases}$$

2. *In Jacobian projective coordinates, τP can be computed using $3\mathbf{M} + 3\mathbf{S}$;*

3. *In Jacobian projective coordinates, $\tau^2 P$ can be computed using $6\mathbf{M} + 5\mathbf{S}$;*

4. *In Jacobian projective coordinates, $3P$ can be computed using $6\mathbf{M} + 4\mathbf{S}$.*

Proof. 1. For $P = (x, y)$, we have $\lambda P = (\beta x, y)$. Write $P_\tau = P - \lambda P = (x', y')$. To compute (x', y'), we first see that the slope is

$$\ell = \frac{2y}{(1-\beta)x}.$$

Now notice that $(1 - \beta)^2 = -3\beta$, we have

$$\begin{aligned} x' &= \ell^2 - (1+\beta)x = -\frac{4y^2}{3\beta x^2} - (1+\beta)x \\ &= -\frac{\big(4 + 3\beta(1+\beta)\big)x^3 + 4b}{3\beta x^2} = \frac{x^3+4b}{(1-\beta)^2x^2}. \end{aligned}$$

$$\begin{aligned} y' &= \ell(x-x') - y = \frac{2y}{(1-\beta)x}\left(\frac{3\beta x^3}{3\beta x^2} + \frac{x^3+4b}{3\beta x^2}\right) - y \\ &= y\frac{2(3\beta+1)x^3 - 3\beta(1-\beta)x^3 + 8b}{3\beta(1-\beta)x^3} \\ &= -y\frac{x^3-8b}{3\beta(1-\beta)x^3} = y\frac{x^3-8b}{(1-\beta)^3x^3}. \end{aligned}$$

2. To derive a formula in Jacobian coordinates, write $P = (\frac{X}{Z^2} : \frac{Y}{Z^3} : 1)$. Then

$$X' = \frac{(\frac{X}{Z^2})^3 + 4b}{(1-\beta)^2(\frac{X}{Z^2})^2} = \frac{X^3+4bZ^6}{((1-\beta)XZ)^2},$$

$$Y' = \frac{Y}{Z^3}\frac{(\frac{X}{Z^2})^3 - 8b}{(1-\beta)^3(\frac{X}{Z^2})^3} = \frac{Y(X^3-8bZ^6)}{((1-\beta)XZ)^3},$$

So a Jacobian coordinates (X_τ, Y_τ, Z_τ) for $(1-\lambda)P$ can be

$$\begin{aligned} X_\tau &= X^3 + 4bZ^6 \\ Y_\tau &= Y(X^3 - 8bZ^6) \\ Z_\tau &= (1-\beta)XZ \end{aligned}$$

Using $bZ^6 = Y^2 - X^3$ and $(1-\beta)^2 = -3\beta$, this is further reduced to

$$\begin{aligned} X_\tau &= 4Y^2 - 3X^3 \\ Y_\tau &= Y(9X^3 - 8Y^2) \\ Z_\tau &= (1-\beta)XZ = Z\frac{(1-\beta+X)^2 - X^2 + 3\beta}{2} \end{aligned} \tag{14}$$

Computing X_τ requires $2\mathbf{S}$+$1\mathbf{M}$, computing Y_τ needs $1\mathbf{M}$, and Z_τ needs $1\mathbf{S}$+$1\mathbf{M}$ with the square computation of $(1-\beta+X)$.

3. Let $(X_{\tau^2} : Y_{\tau^2} : Z_{\tau^2})$ be the Jacobian projective coordinates of $\tau^2 P = \tau(\tau P)$. Similar to the above, we have

$$\begin{aligned} X_{\tau^2} &= 4Y_\tau^2 - 3X_\tau^3 \\ Y_{\tau^2} &= Y_\tau(9X_\tau^3 - 8Y_\tau^2) \\ Z_{\tau^2} &= (1-\beta)X_\tau Z_\tau \end{aligned}$$

We first compute X_τ, Y_τ as in (14), $2\mathbf{S} + 2\mathbf{M}$ is needed. Then the computation of X_{τ^2}, Y_{τ^2}, in the same manner as (14), also requires $2\mathbf{S} + 2\mathbf{M}$.

For Z_{τ^2}, we note that

$$\begin{aligned} Z_{\tau^2} &= (1-\beta)X_\tau Z_\tau = (1-\beta)^2 X_\tau XZ = -3\beta X_\tau XZ \\ &= 3\frac{(X_\tau - \beta)^2 - X_\tau^2 + \beta + 1}{2} \cdot X \cdot Z. \end{aligned}$$

This means that, without computing Z_τ, we can get Z_{τ^2} by using $1\mathbf{S}+2\mathbf{M}$ (note that the element X_τ^2 is already calculated).

Therefore, the cost for $\tau^2 P$ is $5\mathbf{S} + 6\mathbf{M}$

4. By the fact that
$$3 \equiv (1-\lambda)(1-\lambda^2) \pmod{n},$$
$3P = (1-\lambda^2)P_\tau$. Let $(X_3 : Y_3 : Z_3)$ be the Jacobian projective coordinates of $3P$ computed from $(1-\lambda^2)P_\tau$, similar to the above, we have
$$\begin{aligned} X_3 &= 4Y_\tau^2 - 3X_\tau^3 \\ Y_3 &= Y_\tau(9X_\tau^3 - 8Y_\tau^2) \\ Z_3 &= (1-\beta^2)X_\tau Z_\tau \end{aligned}$$
We assume the result of X_τ, Y_τ. Then the computation of X_3, Y_3, as in (14), requires $2\mathbf{S} + 2\mathbf{M}$. We can do better for Z_3 here. Notice that
$$Z_3 = (1-\beta^2)X_\tau Z_\tau = (1-\beta^2)X_\tau(1-\beta)XZ = 3X_\tau(XZ),$$
so without actually computing Z_τ, the calculation of Z_3 costs $2\mathbf{M}$.

Plus the computation of X_τ, Y_τ, we need $4\mathbf{S} + 6\mathbf{M}$ to get $3P$ from P.

□

Remarks

1. In Jacobian projective coordinates for a general Weierstrass form, a previously known cost for calculating $3P$ for general curves over prime is given in [16]. When it is restricted to the case of Koblitz curves, the cost is $10\mathbf{S} + 5\mathbf{M}$. Our specific formula for Koblitz curves over prime fields has achieved a lower cost of $4\mathbf{S} + 6\mathbf{M}$.

2. The efficient formula for $3P$ can be used to do scalar multiplication where double-base chain method is applicable.

3. It is also interesting to note that the computation of $2P = (X_2 : Y_2 : Z_2)$ is very close to that of τP in (14), in fact, one has

$$\begin{aligned} X_2 &= X(9X^3 - 8Y^2) \\ Y_2 &= 9X^3(4Y^2 - 3X^3) - 8Y^4 \\ Z_2 &= 2YZ. \end{aligned}$$

   It is noted that the optimal cost for $2P$ is $4\mathbf{S} + 3\mathbf{M}$ [10].

4. Several facts about τ suggest a possibility of developing window τ-NAF method, which has been very successful for the family of Koblitz curves over binary fields, for the scalar multiplication of Koblitz curves over prime fields. These facts include (1). $N(\tau) > 1$; (2). τP can be done efficiently; and (3). any integer $k < n$ has an efficient reduction of a compact form of $a + b\tau$.

   Previous work of window τ-NAF is mainly with respect to Frobenius maps of certain curves over extension fields of small characteristics, see [13, 14, 18, 5, 6, 7]. For these cases, Frobenius maps can be implemented efficiently. If the characteristic is 2, the Frobenius map requires $2\mathbf{S}$; for characteristics being 3 and 5, corresponding Frobenius maps require $2\mathbf{S} + 2\mathbf{M}$ and $4\mathbf{S} + 2\mathbf{M}$ respectively. In our case, τP is of a comparable cost of $3\mathbf{S} + 3\mathbf{M}$. When it is used in window τ-NAF, we can see a better result as τ^2 can be used for most cases to replace two applications of τ. The average cost of τ in this case is close to $\frac{1}{2}(5\mathbf{S} + 6\mathbf{M}) = 2.5\mathbf{S} + 3\mathbf{M}$.

   We shall propose a window τ-NAF method for the scalar multiplication of Koblitz curves over prime fields in the next section. We will see that due to some nice properties of the Eisenstein integers, a very efficient pre-computation can be constructed.

3 Window Base-τ NAF Method

First we note that since $\tau = 1 - \omega$, the ring of Eisenstein ring $\mathbb{Z}[\omega]$ can be also written as

$$\mathbb{Z}[\tau] = \{x + y\tau : x, y \in \mathbb{Z}\}.$$

Given a natural number w, the idea of the width w window τ NAF method is to seek the following expansion (*window τ-NAF*) of an element $a + b\tau \in \mathbb{Z}[\omega]$:

$$a + b\tau = \sum_{j=0}^{N} \varepsilon_j u_j \tau^j \tag{15}$$

with the property that each nonzero u_j is taken from a suitable set called the *pre-computation set* and each segment $\{u_j, u_{j+1}, \ldots, u_{j+w-1}\}$ contains at most one nonzero element, $\varepsilon_j \in \{-1, 1\}$.

To compute $(a + b\tau)P$, one first sets up a pre-computation: perform uP for each u in some pre-computation set, and compute $\sum_{j=0}^{N} \tau^j(\varepsilon_j u_j P)$.

3.1 Criterion of Divisibility by τ^w

To derive (15), a criterion of the divisibility of elements in $\mathbb{Z}[\tau]$ by τ^w is useful. The next lemma is from [5, 7] where a proof was outlined. Since it is crucial in our discussion, we provide a more detailed proof.

Lemma 3.1. *Let k be a positive integer, then*

1. $$\tau^k = (-3\omega)^{\lfloor \frac{k}{2} \rfloor} \tau^{\lceil \frac{k}{2} \rceil - \lfloor \frac{k}{2} \rfloor}$$

2. *For $x + y\tau \in \mathbb{Z}[\tau]$,*

$$\tau^k | x + y\tau \iff 3^{\lceil \frac{k}{2} \rceil} | x \text{ and } 3^{\lfloor \frac{k}{2} \rfloor} | y.$$

Proof. 1. Notice that $\tau^2 = -3\omega$. The argument follows from induction.

2. Let $\delta = \lceil \frac{k}{2} \rceil - \lfloor \frac{k}{2} \rfloor$, i.e. $\delta = \begin{cases} 0 & \text{if } k \text{ is even} \\ 1 & \text{if } k \text{ is odd .} \end{cases}$

   Now
   $$\tau^k = 3^{\lfloor \frac{k}{2} \rfloor} (-\omega)^{\lfloor \frac{k}{2} \rfloor} \tau^\delta$$

   If $\tau^k | x + y\tau$, since $(-\omega)^{\lfloor \frac{k}{2} \rfloor}$ is a unit, $3^{\lfloor \frac{k}{2} \rfloor} \tau^\delta | x + y\tau$. Thus, there exists $\alpha + \beta\tau \in \mathbb{Z}[\tau]$, such that
   $$x + y\tau = 3^{\lfloor \frac{k}{2} \rfloor} (\alpha\tau^\delta + \beta\tau^{1+\delta}).$$

If k is even, then $\delta = 0$. Hence $x = 3^{\lfloor \frac{k}{2} \rfloor}\alpha = 3^{\lceil \frac{k}{2} \rceil}\alpha,\ y = 3^{\lfloor \frac{k}{2} \rfloor}\beta$.

If k is odd, then $\delta = 1$. Since $\tau^2 = -3 + 3\tau$, we get

$$x = -3^{\lfloor \frac{k}{2} \rfloor + 1}\beta = -3^{\lceil \frac{k}{2} \rceil}\beta,\ y = 3^{\lfloor \frac{k}{2} \rfloor}(\alpha + 3\beta).$$

Therefore, in either case

$$3^{\lceil \frac{k}{2} \rceil} | x \text{ and } 3^{\lfloor \frac{k}{2} \rfloor} | y.$$

Conversely, if $x = 3^{\lceil \frac{k}{2} \rceil} f, y = 3^{\lfloor \frac{k}{2} \rfloor} g$ in $\mathbb{Z}$, then

$$\begin{aligned} x + y\tau &= 3^{\lfloor \frac{k}{2} \rfloor}(3^\delta f + g\tau) = 3^{\lfloor \frac{k}{2} \rfloor}(-\omega)^{\lfloor \frac{k}{2} \rfloor}\tau^\delta \frac{3^\delta f + g\tau}{\tau^\delta}(-\omega)^{-\lfloor \frac{k}{2} \rfloor} \\ &= \tau^k \frac{3^\delta f + g\tau}{\tau^\delta}(-\omega)^{-\lfloor \frac{k}{2} \rfloor}. \end{aligned}$$

The result is proved as $\frac{3^\delta f + g\tau}{\tau^\delta}(-\omega)^{-\lfloor \frac{k}{2} \rfloor} \in \mathbb{Z}[\tau]$ due to the fact that $\tau | 3$.

□

3.2 Pre-computation

We need to decide a set for the coefficients of the representation (15).

Consider the set of all elements of $\mathbb{Z}[\tau]$ which are not divisible by τ. By lemma 3.1, a set of representatives of congruence classes of such elements modulo τ^w is

$$R = \{x + y\tau : 0 \le x \le 3^{\lceil \frac{w}{2} \rceil} - 1, 0 \le y \le 3^{\lfloor \frac{w}{2} \rfloor} - 1, \text{ and } 3 \nmid x\}.$$

For each $x + y\tau \in R$, let

$$C_{x,y} = \{g + h\tau \in \mathbb{Z}[\tau] : g \equiv x \pmod{3^{\lceil \frac{w}{2} \rceil}}, h \equiv y \pmod{3^{\lfloor \frac{w}{2} \rfloor}}, N(g + h\tau) < 3^w\}.$$

Since $\mathbb{Z}[\tau]$ is a Euclidean ring, $C_{x,y}$ is nonempty. In fact, $C_{x,y}$ usually contains several elements. This is a useful property as one has a flexibility in selecting an element in $C_{x,y}$ that contributes to a better pre-computation to serve as a coefficient of expansion (15).

We choose one element $\tilde{x} + \tilde{y}\tau$ from each $C_{x,y}$, then a pre-computation set (nonzero coefficients of expression (15)) can be formed as follows:

$$\text{Pre}_w = \{\tilde{x} + \tilde{y}\tau : 0 \le x \le 3^{\lceil \frac{w}{2} \rceil} - 1, 0 \le y \le 3^{\lfloor \frac{w}{2} \rfloor} - 1, \text{ and } 3 \nmid x\}. \tag{16}$$

In [5], Blake, Murty and Xu design a window base-τ_1 NAF method for Koblitz curves over fields of characteristic three, where τ_1 is the Frobenius map. Even though as endomorphisms of elliptic curves, τ_1 and τ are different, but they are the same as complex numbers. So the window NAF expansion based on them are the same as well. We can use Algorithm 3.1 of [5] to produce (15). We include this algorithm below.

Algorithm 1 Width w window τ-NAF Method

Require: an element $\rho = a + b\tau$ of $\mathbb{Z}[\tau]$

Ensure: S, the array of coefficients of window τ-NAF of ρ

function GEN-NAF(a, b)
  $S \leftarrow <>$
  while $a \neq 0$ or $b \neq 0$ **do**
    if $3 \nmid a$ **then**
      $x \leftarrow a \pmod{3^{\lceil \frac{w}{2} \rceil}}$
      $y \leftarrow b \pmod{3^{\lfloor \frac{w}{2} \rfloor}}$
      $a \leftarrow a - \tilde{x}$
      $b \leftarrow b - \tilde{y}$
      prepend $\tilde{x} + \tilde{y}\tau$ to S
    else
      prepend 0 to S
    end if
    $t \leftarrow a$
    $a \leftarrow a + b$
    $b \leftarrow \frac{-t}{3}$
  end while
  return S
end function

The correctness of this algorithm is carefully discussed in [5] (Theorem3.1): because $\mathbb{Z}[\omega]$ is Euclidean and imaginary quadratic, the width w window τ-NAF method terminates.

3.2.1 A pre-computation for $w = 4$

We now describe an explicit efficient pre-computation for $w = 4$.

Let $w = 4$. In this case, we need to find easily computable coefficients $\tilde{x} + \tilde{y}\tau$ from each $C_{x,y}$ for $x = 1, 2, 4, 5, 7, 8$ and $y \in \{0, 1, 2, \cdots, 8\}$. It is noted that only

$x = 1, 2, 4$ should be considered, as

$$-(x + y\tau) \equiv (9 - x) + (9 - y)\tau \pmod{\tau^4}.$$

The part for $x = 5, 7, 8$ is obtained by negation. This means that we only need to determine 27 coefficients and a half of the pre-computations is saved. As it can be seen later, there are actually 9 pre-computations need to be taken care of, the rest 18 can be obtained easily.

First, there are three trivial coefficients in terms of pre-computations: $1, 1-\tau, 2-\tau$, since for point $P = (r, s)$,

$$(1 - \tau)P = (\beta r, s), \text{ and } (2 - \tau)P = (\beta^2 r, -s).$$

For each of the rest 24 representatives $x + y\tau$, we choose a suitable element $\tilde{x} + \tilde{y}\tau$ from $C_{x,y}$ in a manner to reduce the cost of pre-computation. In table 2, the first, third and fifth row are all $x + y\tau$'s for $x = 1, 2, 4$ and $0 \leq y < 9$, except for the trivial ones $1, 1 - \tau$ and $2 - \tau$. These $x + y\tau$'s serve as indexes for pre-computation. The second row lists the selected $\tilde{x} + \tilde{y}\tau$ from $C_{x,y}$ corresponding to the (index) $x + y\tau$ in the first row. The pre-computation is then $Q_{x+y\tau} = (\tilde{x} + \tilde{y}\tau)P$.

The third to sixth rows are carefully arranged in the way for better efficiency. We start with the fourth row: an element $\tilde{x} + \tilde{y}\tau$ is obtained by multiplying ω to the corresponding (column) element in the second row if $\tilde{x} + \tilde{y}\tau \in C_{x,y}$ with $1 \leq x \leq 4$, otherwise, $\tilde{x} + \tilde{y}\tau$ is obtained by multiplying $-\omega$ to the corresponding (column) element in the second row. The corresponding representative $x+y\tau$ is put in the third row as its index. We form the sixth row and fifth row in a similar way by multiplying ω^2 or $-\omega^2$ to the corresponding (column) element in the second row.

Table 2: Pre-Computation for $w = 4$

$x + y\tau$	2	4	$1 + \tau$	$2 + 2\tau$	$1 + 2\tau$	$2 + 4\tau$	$2 + \tau$	$1 + 7\tau$
$\tilde{x} + \tilde{y}\tau$	2	4	$1 + \tau$	$2 + 2\tau$	$1 + 2\tau$	$2 + 4\tau$	$2 + \tau$	$1 - 2\tau$
$x + y\tau$	$2 + 7\tau$	$4 + 5\tau$	$4 + 6\tau$	$1 + 6\tau$	$2 + 5\tau$	$4 + \tau$	$4 + 4\tau$	$4 + 3\tau$
$\tilde{x} + \tilde{y}\tau$	$2(1 - \tau)$	$4(1 - \tau)$	$4 - 3\tau$	$-8 + 6\tau$	$-7 + 5\tau$	$-14 + 10\tau$	$-5 + 4\tau$	$-5 + 3\tau$
$x + y\tau$	$4 + 7\tau$	$1 + 4\tau$	$4 + 2\tau$	$1 + 5\tau$	$1 + 3\tau$	$2 + 6\tau$	$2 + 3\tau$	$4 + 8\tau$
$\tilde{x} + \tilde{y}\tau$	$4 - 2\tau$	$-8 + 4\tau$	$-5 + 2\tau$	$10 - 4\tau$	$-8 + 3\tau$	$-16 + 6\tau$	$-7 + 3\tau$	$4 - \tau$

We explain why this set of coefficients achieves efficiency. It can be checked that

$\tilde{x} + \tilde{y}\tau$'s in the fourth row in table 2 are obtained as

$$\begin{array}{lll} 2(1-\tau) = 2\omega, & 4(1-\tau) = 4\omega, & 4 - 3\tau = \omega(1+\tau) \\ -8 + 6\tau = -\omega(2+2\tau), & -7 + 5\tau = \omega(1+2\tau), & -14 + 10\tau = -\omega(2+4\tau) \\ -5 + 4\tau = -\omega(2+\tau) & -5 + 3\tau = \omega(1-2\tau) & \end{array}$$

Similarly, $\tilde{x} + \tilde{y}\tau$'s in the sixth row in table 2 are obtained as

$$\begin{array}{lll} 4 - 2\tau = 2(-\omega^2), & -8 + 4\tau = 4\omega^2, & -5 + 2\tau = \omega^2(1+\tau) \\ 10 - 4\tau = -\omega^2(2+2\tau), & -8 + 3\tau = \omega^2(1+2\tau), & -16 + 6\tau = \omega^2(2+4\tau) \\ -7 + 3\tau = \omega^2(2+\tau), & 4 - \tau = \omega^2(1-2\tau). & \end{array}$$

This implies that if we have computed a point $Q = (\tilde{x} + \tilde{y}\tau)P$ for the second row, then the corresponding pre-computation in the fourth row and sixth row can be simply computed as λQ (or $-\lambda Q$) and $\lambda^2 Q$ (or $-\lambda^2 Q$) , respectively. Note that computing λQ is easy: if $Q = (r, s)$, then $\lambda Q = (\beta r, s)$; computing $\lambda^2 Q$ can even be neglected in the sense that

$$\lambda^2 Q = (\beta^2 r, s) = (-\beta r - r, s),$$

since the multiplication βr has been computed in λQ.

We compute all $Q_{x+y\tau} = (\tilde{x} + \tilde{y}\tau)P$ for the second row one by one in the following order:

$$\begin{array}{lll} Q = \tau P, & Q_2 = 2P, & Q_4 = 4P = 2Q_2 \\ Q_{1+\tau} = (1+\tau)P = P + Q, & Q_{2+2\tau} = (2+2\tau)P = 2Q_{1+\tau}, & Q_{2+\tau} = (2+\tau)P = Q_2 + Q \\ Q_{1+2\tau} = (1+2\tau)P = Q_{2+2\tau} - P & Q_{1+7\tau} = (1-2\tau)P = Q_2 - Q_{1+2\tau} & Q_{2+4\tau} = (2+4\tau)P = 2Q_{1+2\tau} \end{array}$$

This requires 1**operation** $\tau + 4$**DBL** $+ 4$**ADD**. The computation for the fourth row needs 8 field multiplications. Plus the scalar multiplications by $1-\tau = \omega, 2-\tau = -\omega^2$, which costs $2\mathbf{M}$, the pre-computation cost is

$$5.4\mathbf{M} + 8\mathbf{M} + 4 \cdot 6.2\mathbf{M} + 4 \cdot 15\mathbf{M} + 2\mathbf{M} = 100.2\mathbf{M}.$$

3.3 Width-4 window τ-NAF Method for Scalar Multiplication

In coordination with the pre-computation designed above, we choose window size $w = 4$ for scalar multiplication. This is an appropriate size one because it also helps to maximize the benefit brought by the more efficient $\tau^2 P$ formula. We have

determined the coefficients in section 3.2.1 for pre-computation. The selected set of coefficients is $C \cup (-C) \cup \{0\}$ if we write

$$\begin{aligned} C &= \{1, 1-\tau, 2-\tau, 2, 4, 1+\tau, 2+2\tau, 1+2\tau, 2+4\tau, 2+\tau, 1-2\tau, 2(1-\tau)\} \cup \\ &\quad \{4(1-\tau), 4-3\tau, -8+6\tau, -7+5\tau, -14+10\tau, -5+4\tau, -5+3\tau, \} \cup \\ &\quad \{4-2\tau, -8+4\tau, -5+2\tau, 10-4\tau, -8+3\tau, -16+6\tau, -7+3\tau, 4-\tau\} \end{aligned} \tag{17}$$

The pre-computation with respect to a point $P \in E_b(\mathbb{F}_p)$ is to compute uP for every $u \in C$. The scalar multiplication with negative part of C is trivial.

The following is our scalar multiplication algorithm for Koblitz curves over prime fields. We rewrite the expression (15) for $a + b\tau$ to explicitly mark it lowest term:

$$a + b\tau = \sum_{j=\ell}^{N} \varepsilon_j u_j \tau^j \tag{18}$$

where $u_j \in C \cup \{0\}, \varepsilon_j \in \{-1, 1\}, u_\ell \neq 0, u_N \neq 0$. Note that every non-zero term (except for the last one) is followed by at least 4 terms with zero coefficient, so we can perform at least two τ^2-operations by the formula in proposition 2.1.

Algorithm 2 Window τ-NAF Method for Scalar Multiplication

Require: a positive integer $k < n$ and a point $P \in E_b(\mathbb{F}_p)$
Ensure: the scalar multiplication kP

function SCAL-MUL(k, P)
  Pre-Computation: uP for each $u \in C$.
  Use Section 2.1 to get reduction $k = k_1 + k_2\lambda \pmod{n}$
  $a \leftarrow k_1 + k_2,\ b \leftarrow -k_2$
  Use Algorithm 1 to get the expression (18)
  $Q \leftarrow \mathcal{O}$
  $j \leftarrow N$
  while $j \geq \ell$ **do**
    if $u_j \neq 0$ **then**
      $Q \leftarrow Q + \varepsilon_j P_{u_j}$
      if $j = \ell$ **then**
        for i from 0 to $\lfloor \frac{\ell}{2} \rfloor$ **do**
          $Q \leftarrow \tau^2 Q$
          if ℓ is odd **then**
            $Q \leftarrow \tau Q$
          end if

```
            end for
            return Q
        end if
        Q ← τ²Q, Q ← τ²Q
        j ← j − 4
        while u_j = 0 do
            Q ← τQ
            j ← j − 1
        end while
    end if
end while
return Q
end function
```

3.3.1 Estimation of the cost for width-4 window τ-NAF

In the process of calculating kP for $k < n$, one reduces k to $k_1 + k_2\lambda$ such that $k \equiv k_1 + k_2\lambda \pmod{n}$ with $|k_1|, |k_2| < \sqrt{n}$. Set $a = k_1 + k_2, b = -k_2$, we have

$$k \equiv a + b\tau \pmod{n}.$$

Note that $N(\tau) = 3$, the window τ-NAF for $a + b\tau$ should be of length $\log_3 n$. In the expansion (15) for $a+b\tau$, after each nonzero coefficient, there will be w consecutive zeros, beyond that, asymptotic expectation of zeros followed is $\frac{1}{3} + \frac{1}{3^2} + \cdots = \frac{1}{2}$. So the average number of nonzero terms is

$$\frac{\log_3 n}{w + \frac{1}{2}}.$$

With $w = 4$, this means that, given a pre-computation, algorithm 2 requires $2\frac{\log_3 n}{4.5}$**operation** τ^2, $(\log_3 n - 4\frac{\log_3 n}{4.5})$ **operation** τ, and $\frac{\log_3 n}{4.5}$**ADD**.

Recall that **operation** $\tau \approx 5.4$**M**, **operation** $\tau^2 \approx 10$**M**, and **ADD** ≈ 15**M**, so the overall cost for the computation of kP in terms of the number of field multiplications is

$$\begin{aligned}\left(\frac{2}{4.5} \cdot 10 \quad + (1 - \tfrac{4}{4.5}) \cdot 5.4 + \tfrac{1}{4.5} \cdot 15\right) \log_3 n\mathbf{M} + 100.2\mathbf{M} \\ \approx (5.29 \log_2 n + 100.2)\mathbf{M}. \qquad (19)\end{aligned}$$

Currently, the best method for scalar multiplication for Koblitz curves over primes fields is the GLV method. According to our experiments, the version of interleaving with 4-NAF performs the best for Koblitz curve SECP256K1. For Koblitz curves over prime fields, the interleaving with w-NAF is to compute $k_1P + k_2Q$ where $Q = \lambda P$ using w-NAF to k_1 and k_2. The pre-computation can be done essentially just for P: pre-compute jP for $j = 3, 5, \cdots, 2^{w-1}-1$. The Q part of pre-computation is obtained simply by $jQ = \lambda(jP)$ by using 4 field multiplications for $j = 1, 3, 5, 7$. The rest of the cost of interleaving with 4-NAF is $\frac{\log_2 n}{2}\mathbf{DBL} + \frac{\log_2 n}{5}\mathbf{ADD}$. Converting to field multiplications, the cost becomes

$$\left(\frac{\log_2 n}{2} + 1\right) \cdot 6.2\mathbf{M} + \left(\frac{\log_2 n}{5} + 3\right) \cdot 15\mathbf{M} + 4\mathbf{M} \approx (6.1 \log_2 n + 55.2)\mathbf{M}. \quad (20)$$

Comparing (19) with (20), we see that the width-4 window τ-NAF method achieves more than 11% and 12.5% of improvements over the GLV method, when $\log_2 n = 256$ and 384 respectively.

4 Conclusion

This paper discussed Koblitz curves over primes fields by utilizing several nice properties of the Eisenstein integers. An efficient formula for a (complex) scalar multiplication by $\tau = 1 - \omega$ is derived. Based on it, further optimzed fast formulas for $\tau^2 P$ and $3P$ are obtained. The cost of τP becomes $3\mathbf{S} + 3\mathbf{M}$ in Jacobian coordinates (for the situations where $\tau^2 P$ can be used, the average cost can be reduced to close to $2.5\mathbf{S} + 3\mathbf{M}$). This cost is comparable to that for the Frobenius map over extension fields of characteristic 5 and suggests a possibility of creating a window τ-NAF method for the family of Koblitz curves over prime fields. This is achieved in the paper by developing some mathematical tools and by designing a very efficient pre-computation. This method gains more than 11% of improvement over the GLV method. The paper also proposed a simplified modular reduction for Eisenstein integers where the division operations are eliminated. This modular reduction can be used in the GLV method for this class of curves as well. Another efficient formula developed in this paper is the computation of $3P$, compared to $10\mathbf{S} + 5\mathbf{M}$, our new formula just costs $4\mathbf{S} + 6\mathbf{M}$. As a main ingredient for double base chain method for scalar multiplication, the formula will contribute to a greater efficiency.

References

[1] National Institute of Standards and Technology(NIST).: Digital signature standard(DSS). FIPS PUB 186-5(Draft). 2019 October. https://doi.org/10.6028/NIST.FIPS.186-5-draft

[2] SP 800-186 Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters, *https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-186.pdf*,2023.

[3] SEC 2: Recommended Elliptic Curve Domain Parameters, *https://www.secg.org/sec2-v2.pdf*, 2010.

[4] Bernstein D. J., Chuengsatiansup C., and Lange T. : Double-base scalar multiplication revisited. https://eprint.iacr.org/2017/037.

[5] Blake I. F., Murty K. V., Xu G.: Efficient algorithms for Koblitz curves over fields of characteristic three, *Journal of Discrete Algorithms*, 3(2005)113-124.

[6] Blake I. F., Murty K. V., Xu G.: A note on window τ-NAF algorithm, *Information Processing Letters*, 95(2005), no. 5, 496-502.

[7] Blake I. F., Murty K. V., Xu G.: Nonadjacent radix-τ expansions of integers in Euclidean imaginary quadratic number fields, *Canadian Journal of Mathematics*, 60(2008), 1267-1282.

[8] Brown E., Myers B.T., Solinas J.A.: Elliptic curves with compact parameters. Tech. Report, Centre for Applied Cryptographic Research (2001). http://www.cacr.math.uwaterloo.ca/techreports/2001/ corr2001-68.ps.

[9] Cohen H.: A Course in Computational Algebraic Number Theory. Springer, 2000.

[10] Hankerson D., Menezes A., Vanstone S.: Guide to elliptic curve cryptography. New York, NY, USA: Springer-Verlag, 2004.

[11] Hu Z., Longa P., Xu M.: Implementing the 4-dimensional GLV method on GLS elliptic curves with j-invariant 0. Des. Codes Cryptogr. (2012) 63:331343.

[12] Gallant R., Lambert R., Vanstone S.: Fast point multiplication on elliptic curves with efficient endomorphisms. *Crypto 2001*, LNCS **2139**, 190-200.

[13] Koblitz N.: CM-curves with good cryptographic properties, in Proc. 11th Annu. Int. Cryptol. Conf. Adv. Cryptol., pp. 279-287, 1992.

[14] Koblitz N.: An elliptic curves implementation of the finite field digital signature algorithm, *Advances in Cryptology-CRYPTO '98*, LNCS **1462**, 1998, 327-337.

[15] Koblitz N.: Algebraic Aspects of Cryptography, Springer, 1999.

[16] Longa P. and Miri A.: Fast and flexible elliptic curve point arithmetic over prime fields, *IEEE Transactions on Computers*, 57(2008) , 289-302.

[17] Rajwade A. R.: On rational primes p congruent to 1 (mod 3 or 5), Proc. Cambridge Philos. Sot. 66 (1969), 61-70.

[18] Solinas J.: Efficient arithmetic on Koblitz curves, Des., Codes Cryptography, vol. 19, pp. 195-249, 2000.

[19] Trost W.: Pre-computation in Width-w τ-adic NAF Implementations on Koblitz Cruves, MS Thesis, University of Wisconsin-Milwaukee, May 2014.

[20] Trost W. and Xu G.: On the optimal pre-computation of window τNAF for Koblitz curves. IEEE Transactions on Computers. Vol. 65, No. 9. pp. 2918-2924, September 2016.

[21] Wu H. and Xu G.: On Koblitz curves over prime fields. Journal of Cryptologic Research, 20XX, 0(0): 18. (in Chinese) [DOI: 10.13868/j.cnki.jcr.000735]

[22] Yu W. and Xu G.: Pre-Computation Scheme of Window τNAF for Koblitz Curves Revisited. In: Advances in CryptologyEUROCRYPT 2021, Part II. Springer Cham, 2021: 187218. [DOI: 10.1007/978-3-030-77886-6_7]