Speeding up Pairing Computation Using Non-adjacent Form and ELM Method

Chao-Liang Liu[1], Gwoboa Horng[2], and Du-Shiau Tsai[3]
(Corresponding author: Chao-Liang Liu)

Department of Applied Informatics and Multimedia, Asia University[1]
500, Lioufeng Rd., Wufeng, Taichung 41354, Taiwan R.O.C.
Department of Computer Science and Engineering, National Chung-Hsing University[2]
250 Kuo Kuang Rd., Taichung, Taiwan R.O.C.
Department of Information and Networking Technology, Hsiuping University of Science and Technology[3]
No.11 Gongye Rd, Dali Dist., Taichung City 412-80,Taiwan, R.O.C
(Email: jliu@asia.edu.tw)
(Received Oct. 3, 2013; revised and accepted Jan. 15 & Mar. 26, 2014)

Abstract

The bilinear pairings such as Weil pairing and Tate pairing on elliptic curves have recently found many applications in cryptography. The first efficient algorithm for computing pairing was originally proposed by Miller and much subsequent research has been directed at many different aspects in order to improve efficiency. In 2003, Eisenträger, Lauter and Montgomery proposed a new point-double-addition method to speed up elliptic curve arithmetic computation and obtained a 7.8% performance improvement of the Miller algorithm of a general elliptic curve. In 2006, Blake et al. proposed a new concept based on the conjugate of a line to reduce the total number of lines in the Miller algorithm. In this paper we propose an enhancement of Eisenträger et al.'s algorithm for computing pairings. Our enhancement can further speed up the pairing computation by 5.9%.

Keywords: Elliptic curve cryptosystem, pairing-based cryptosystem, pairing computation

1 Introduction

Elliptic curve cryptograph, introduced by Miller [21, 22] and Koblitz [13] independently around 1985, provides the same level of security as the conventional public-key cryptography but with shorter keys. Numerous research efforts have been devoted to elliptic curve cryptography and a lot of cryptosystems have been proposed. By using Weil pairing, Menezes, Okamoto and Vanstone found some weak curves which contain cyclic groups that can be transformed into a finite field with small extension degree (MOV degree) [19]. Frey and Ruck extended their attack and found more weak curves with the Tate pairing [10]. Basically, the Weil/Tate pairing is a mapping with non-degenerate and bilinear properties, which will map a special pair of points on an elliptic curve to a certain multiplicative subgroup of a finite field. In recent years, bilinear pairings especially, Weil/Tate pairings, have found positive applications in cryptography. Indeed, many cryptographic applications based on pairings have been proposed, such as identity-based encryption systems [4], digital signatures [5, 6, 25, 26], signcryption [16, 24] and key agreement [12, 29]. As a result, the application of pairings plays an important role in modern cryptography. Therefore, efficiently implementation of pairing computation is an important issue due to being the most costly operation in these cryptosystems. The first efficient algorithm for computing pairing was proposed by Miller [21, 22]. The main idea of the Miller algorithm is to use lines to integrate the divisors, which the algorithm has processed (see Section 2, for details). A lot of research has been aimed in many different directions in order to improve efficiency [1, 2, 3, 7, 8, 10, 15, 17, 23, 27, 28, 33]. The research of Barreto, Kim, Lynn and Scott [1], and Galbraith, Harrison and Soldera [10] focuses particularly on the Tate pairing over some special curves. The research in [3, 8] can improve the performance of Weil/Tate pairing computation in general elliptic curves. We will continue in this direction.

It is well known that point subtraction and point addition on an elliptic curve have the same cost. Non-adjacent form (NAF for short) has been widely used for the scalar multiplication of nP for some point P on an elliptic curve [11]. Through this property, the efficiency of pairing computation can also be improved. For example, Eisenträger, Lauter and Montgomery gave a new point-addition/subtraction method (ELM method for short) to speed up scalar multiplication and pairing computation [8]. The majority of research in [8] literacy has fo-

cused on the double-addition/subtraction step when the bit of NAF representation of n is 1/-1. It is noticeable that the number of double step is twice the number of double-addition/subtraction step on average. With a parabola substitution, they get a 7.8% performance improvement of the Miller algorithm for a general elliptic curve (see Section 2, for details).

In 2006, Blake et al. proposed a new concept based on the conjugate of a line to reduce the total number of lines in the Miller algorithm [3]. Three different algorithms are proposed for three cases namely, BMX-1, BMX-2 and BMX-3. The first algorithm, $\log_2^n$ field multiplications are eliminated when there are relatively more zero bits (or average cases) of the binary representation of integer n. The second case is when there are relatively more one bits and $2H(n)$ field multiplications are removed where $H(n)$ is the number of bit 1. The third case saves $\log_3^n$ field multiplications when the characteristic of the field is three. Some successive works further improving Blake et al.'s algorithms [14, 18, 31, 32].

In this paper, we propose an algorithm to eliminate one more field multiplication in a double step, which the ELM method can not apply. Our new reduction method can reduce the number of lines, and hence improve the efficiency of pairing computation even further. The result can speed the computation of the Weil and Tate pairing by up to 5.9%, that is, combined with the ELM method, we can obtain a 13.3% performance improvement.

The rest of the paper is organized as follows. We briefly review some mathematical preliminaries, the Miller algorithm, the ELM method and Blake et al.'s formulae in Section 2. In Section 3 we describe our proposed algorithm. Its analysis is given in Section 4. Finally, some concluding remarks are given in Section 5.

2 Background

2.1 Weil/Tate Pairing and Miller Algorithm

Let E be an elliptic curve over a finite field F_q where q is a power of a prime p. We can express E as the Weierstrass equation

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$

where a_1, a_2, a_3, a_4, a_6 are all in F_q. If $\gcd(q, 6) = 1$, a nonsingular elliptic curve over the field F_q is given by an equation of the form

$$E_s : y^2 = x^3 + ax + b$$

with $a, b \in F_q$ and $4a^3 + 27b^2 \neq 0$. Let $E(F_q)$ denotes the set of points $(x, y) \in F_q^2$, satisfying E_s together with the point at infinity denoted as ∞. Then $E(F_q)$ together with point addition has a structure of an abelian group which is denoted as E. Explicit formulas for computing the coordinates of a point $R = P+Q$ from the coordinates of P and Q are well known [24, 26]. We give the formulae relative to $R = P + Q$ when $P \neq \pm Q$. That is, $R = (\lambda^2 - x_1 - x_2, \lambda x_1 - \lambda x_3 - y_1)$, where $P = (x_1, y_1)$, $Q = (x_2, y_2)$ and $\lambda = \frac{(y_2-y_1)}{(x_2-x_1)}$.

A divisor D is a formal sum of symbols from the set $\{(P)\colon\ P \in E\}$ with integer coefficients. That is $D = \sum_{P\in K} n_P(P)$. The set of all divisors, denoted by $Div(E)$, is a free abelian group generated by E. Define the degree of a divisor D, $\deg(D)$, to be $deg(D) = \sum_{P\in E} n_P$. We can define an important subgroup of $Div(E)$, denoted as $Div^0(E) = \{D \in Div(E)\colon\ deg(D) = 0\}$. The divisor of a nonzero rational function f is $div(f) = \sum_{P\in E} ord_P(f)(P)$, where $ord_P(f)$ is the order of f at P. It is well known that $div(f) \in Div^0(E)$ is called a principle divisor. If there exists a nonzero rational function f such that $D_1 = D_2 + div(f)$ then D_1 and D_2 are said to be equivalent, denoted as $D_1 \sim D_2$. The support of a divisor D is the set of points with nonzero coefficients, that is, $supp(D) = \{P \in E\colon\ n_P \neq 0\}$. If $div(f)$ and D have disjoint support, then we can evaluate $f(D) = \prod_{P\in E} f(P)^{n_P}$.

Let n be an integer relatively prime to q and $P, Q \in E[n]$, where $E[n]$ is the n-torsion subgroup of E. Then there exist divisors D_P, D_Q such that $D_P \sim (P) - (\infty)$ and $D_Q \sim (Q) - (\infty)$. Further, there exist functions f_P, f_Q such that $div(f_P) = nD_P$ and $div(f_Q) = nD_Q$. If D_P and D_Q have disjoint supports, then the Weil pairing is $e_n(P, Q) = \frac{f_P(D_Q)}{f_Q(D_P)}$. And the Tate pairing of order n is the map $\tau_n\colon\ E(F_q)[n]E(F_{q^k})/nE(F_{q^k}) \to F_{q^k}$, with $\tau_n(P, Q) = f_n(D_Q)^{(q^k-1)/n}$, where $div(f_n) = n(P) - n(\infty)$. Hence, computing the Weil/Tate pairing can be reduced to the evaluation of $f_P(S)$, where S is in the support of D_Q.

We briefly describe the main idea of the Miller algorithm as follows: Let $D_P = (P+R)-(R)$ with an auxiliary point R and $D_P^j = j(P+R)-j(R)-(jP)+(\infty)$, and then there is a rational function f_j such that $div(f_j) = D_P^j$, for each integer j, in particular, $f_n = f_P$. Hence

$$\begin{aligned}
div(f_{j+k}) &= (j+k)(P+R) - (j+k)(R) \\
&\quad -((j+k)P) + (\infty) \\
&= [(j(P+R) - j(R) - (jP) + (\infty)] \\
&\quad +[k(P+R) - k(R) - (kP) + (\infty)] \\
&\quad +(jP) + (kP) - ((j+k)P) - (\infty) \\
&= div(f_j) + div(f_k) \\
&\quad +[(jP) + (kP) + (-(j+k)P - 3(\infty)] \\
&\quad -[((j+k)P) + (-(j+k)P) - 2(\infty)] \\
&= div(f_j) + div(f_k) + div(L_{jP,kP}) \\
&\quad -div(L_{(j+k)P}),
\end{aligned}$$

where $L_{jP,kP}$ be a line through the points jP, kP and $-(j+k)P$. $L_{(j+k)P}$ be a vertical line through the points $(j+k)P$ and $-(j+k)P$. Then, $f_{j+k} = f_j f_k (\frac{L_{jP,kP}}{L_{(j+k)P}}$. As a result, we can obtain f_{j+k} from f_j and f_k with some "glue": the appropriate lines, $L_{jP,kP}$ and $L_{(j+k)P}$.

We can compute $f_n(S)$ recursively with initial values $f_0 = 1$ and $f_1 = \frac{L_{P+R}}{L_{R,R}}$. We describe the following algorithm, which is similar to the algorithm proposed in [1, 3]. Note that we can perform the Miller algorithm to compute Tate pairing by changing the initial value $f_1 = 1$, see [1] for details.

Algorithm 1 Miller algorithm

1: INPUT: Elliptic curve E, integer $n = \sum_{i=0}^{t} b_i 2^i$ with $b_i \in \{0, 1\}$ and $b_t = 1$, and points P, $S \in E$ where P has order n.
2: OUTPUT: $f = f_n(S)$.
3: $f \leftarrow f_1$; $Z \leftarrow P$;
4: for $j \leftarrow t-1$ down to 0 do
5: $f \leftarrow f^2 \frac{L_{Z,Z}(S)}{L_{2Z}(S)}$; $Z \leftarrow 2Z$;
6: for $j \leftarrow t-1$ down to 0 do
7: $f \leftarrow f^2 \frac{L_{Z,Z}(S)}{L_{2Z}(S)}$; $Z \leftarrow 2Z$;
8: if $b_j = 1$ then
9: $f \leftarrow f_1 f \frac{L_{Z,P}(S)}{L_{Z+P}(S)}$; $Z \leftarrow Z + P$;
10: return f;
11: End

2.2 ELM Method

In Algorithm 1, the cost of pairing computation consists of two main parts. One is a scalar multiplication of nP. The other is an exponential computation and multiplication with the glue. To decrease the cost of point's double-addition/subtraction of scalar multiplication, Eisenträger et al. eliminate two field multiplications through a new method to compute $2P + Q$ by computing $P + Q$ and $2P + Q = (P + Q) + P$. Note that, we do not care about the intermediate result $P + Q$. The explicit formulae are described as follows:

$$\begin{aligned}
\lambda_1 &= \frac{y_2 - y_1}{x_2 - x_1}, x_3 = \lambda_1^2 - x_1 - x_2 \\
\lambda_2 &= -\lambda_1 - \frac{2y_1}{x_3 - x_1}, x_4 = \lambda_2^2 - x_1 - x_3 \\
y_4 &= (x_1 - x_4)\lambda_2 - y_1,
\end{aligned}$$

where $P = (x_1, y_1)$, $Q = (x_2, y_2)$, $x_1 \neq x_2$, $P + Q = (x_3, y_3)$, and $2P + Q = (x_4, y_4)$ on an elliptic curve E_S. Moreover, λ_1 is the slope of $L_{P,Q}$ and λ_2 is the slope of $L_{P+Q,P}$.

To apply this point double-addition/subtraction method for the Miller algorithm, they construct a parabola to glue the Miller's divisors, whenever the corresponding bit is one, see [8] for detail.

Suppose we use the binary method in [11] to form nP, where n has t bits. There are $2t/3$ doubles and $t/3$ double-additions/subtractions. By way of estimating a division as 5.18 field multiplications, they compute the average cost of the standard algorithm as

$$\frac{(16.18 \times 2t/3) + (31.36 \times t/3)}{t} = 21.24$$

field multiplications per bit, and the average cost of their new method is

$$\frac{(16.18 \times 2t/3) + (26.36 \times t/3)}{t} = 19.57$$

field multiplications per bit. The performance improvement ratio for their new method is 7.8%. It is noticeable that these estimations are based on the computation of Tate pairing for which $f_n(Q_1)$ and $f_n(Q_2)$ are computed at the same time. Please see [8] or Section 4 for details.

We also need a divisor subtraction formula to use the NAF method to form $f_n(S)$ with respect to the Miller algorithm. Therefore, they proposed the first divisor subtraction formula:

$$\begin{aligned}
div(f_{j-k}) &= (j-k)(P+R) - (j-k)(R) \\
&\quad -((j-k)P) + (\infty) \\
&= [(j(P+R) - j(R) - (jP) + (\infty))] \\
&\quad -[k(P+R) - k(R) - (kP) + (\infty)] \\
&\quad +(jP) - (kP) - ((j-k)P) + (\infty) \\
&= div(f_j) - div(f_k) \\
&\quad +[(jP) + (-jP) - 2(\infty)] \\
&\quad -[(-jP) + (kP) + ((j-k)P) - 3(\infty)] \\
&= div(f_j) - div(f_k) + div(L_{jP}) \\
&\quad -div(L_{-jP,kP}),
\end{aligned}$$

Therefore,

$$f_{j-k} = \frac{f_j}{f_k} \cdot \frac{L_{jP}}{L_{-jP,kP}}. \quad (1)$$

2.3 Blake et al's Lemmas

From the analysis in [8], we know that if we can reduce one line then at least one field multiplication is saved in the Miller algorithm. For this reason, Blake et al. proposed three algorithms to reduce the number of lines. The first algorithm is suitable for every case. The second algorithm can work well if the Hamming weight of n is high. The third algorithm is proposed for fields of characteristic 3. These algorithms are based on the following two lemmas which were proved in [3].

Lemma 1. *If the line $L(x, y)$ intersects with E at points $P = (a, b)$, $Q = (c, d)$ and $-(P + Q) = (\alpha, \beta)$, then $L(x, y)\bar{L}(x, y) = -(x - a)(x - c)(x - \alpha)$, where $\bar{L}(x, y)$ is the conjugate of L with $L(R) = \bar{L}(-R)$ for $R \in E$.*

Lemma 2. *Let $Q \in E[n]$, $S \in E$ and $S \neq Q$, $2Q$, $\cdots$, nQ, then*

1) $\frac{L_{Q,Q}(S)}{L_Q^2(S)L_{2Q}(S)} = \frac{-1}{L_{Q,Q}(-S)}$.

2) For all integer k, we have $\frac{L_{(k+1)Q,kQ}(S)}{L_{(k+1)Q}(S)L_{(2k+1)Q}(S)} = -\frac{L_{kQ}(S)}{L_{(k+1)Q,kQ}(-S)}$.

3) $\frac{L_{Q,Q}(S)L_{2Q,Q}(S)}{L_{2Q}(S)L_{3Q}(S)} = -\frac{L_{Q,Q}(S)L_Q(S)}{L_{2Q,Q}(-S)}$.

They also remark that [3]:

1) Since $div(f) = fiv(cf)$ for any nonzero constant $c \in K$, the sign does not affect the pairing computation and therefore, minus signs will be omitted in the use of the above lemma.

2) The point $P \in E[n]$ will be fixed and Q is taken to be some multiple of P. In order to satisfy the condition of the lemma, it is sufficient to let $S \neq P$, $2P$, $\cdots$, nP. This is also the requirement of the original Miller algorithm.

3 A New Method for Computing Pairings

In Section 2, the ELM method concentrates on the double-addition/subtraction step in the point's scalar multiplication, however the number of double steps is twice the number of double-addition/subtraction steps. Therefore, we suggest a new algorithm to reduce one field multiplication when the corresponding bit of n is 0 in the Miller algorithm. Before expressing this new algorithm, we briefly describe the limitations of their method to compute the pairing.

To compute $2P+Q$ via $P+Q$, where P, Q, $P+Q$ and $2P+Q$ on an elliptic curve E_s but $P \neq \pm Q$. Then the capability and the limitations of their method are:

1) We have the x-coordinators of the points P, Q, $P+Q$ and $2P+Q$. But we do not have the x-coordinator of the point $2P$.

2) We have the y-coordinators of the points P, Q and $2P+Q$. But we do not have the y-coordinators of $P+Q$ and $2P$.

3) We have the slopes for the lines $L_{P,Q}$ and $L_{P+Q,P}$.

4) We can construct the linear functions $L_{P,Q}$, $L_{P+Q,P}$, L_P, L_Q, L_{P+Q}, and L_{2P+Q}. But we cannot construct $L_{2P,Q}$ and L_{2P}.

The detail description of the divisor subtraction formula (Equation (1)) with their point double-subtraction method in the Miller algorithm is

$$f \leftarrow \frac{f^2}{f_1} \cdot \frac{L_Z(S)L_{Z-P,Z}(S)}{L_{-Z,P}(S)L_{2Z-P}(S)}; Z \leftarrow 2Z - P;$$

when the bit of n is -1. Although the linear functions L_Z, $L_{Z-P,Z}$, L_{2Z-P}, and $L_{-Z,P}$ can be constructed, no parabola was revealed in [8]. It is well-known that there are no consecutive nonzero bits in the NAF representation such that there is always a zero bit before -1. In [3], they have examined the reduction formulae of bit 0 and bit 1, however, there were few studies of the relationship between bit 0 and bit -1. Therefore, we have to extend Lemma 2.2 to Lemma 3.1 in order to establish a reduction formula for this case.

Lemma 3. *Let $Q \in E[n]$, $S \in E$ and $S \neq Q, 2Q, \cdots, nQ$, then $\frac{L_(k-1)Q,kQ(S)}{L_{kQ}(S)L_{(2k-1)Q}(S)} = \frac{L_{(k-1)Q}(S)}{L_{(k-1)Q,kQ}(-S)}$.*

Proof. For a point $S \in E$, we write $S = (x_S, y_S)$, that is, x_S is the x-coordinate of S and y_S is the y-coordinate of S. By Lemma 2, we have:

$$\begin{aligned} & \frac{L_(k-1)Q, kQ(S)}{L_{kQ}(S)L_{(2k-1)Q}(S)} \\ = & \frac{L_(k-1)Q, kQ(S)\bar{L}_{(k-1)Q,kQ}(S)}{L_{kQ}(S)L_{(2k-1)Q}(S)\bar{L}_{(k-1)Q,kQ}(S)} \\ = & \frac{-(x_S - x_{(k-1)Q})(x_S - x_{kQ})(x_S - x_{(2k-1)Q}}{(x_S - x_{kQ})(x_S - x_{(2k-1)Q})L_{(k-1)Q,kQ}(-S)} \\ = & \frac{L_{(k-1)Q}(S)}{L_{(k-1)Q,kQ}(-S)}. \end{aligned}$$

□

Consider the NAF representation of $n = \sum_{i=0}^{t} b_i 2^i$ with $b_i \in \{0, 1\}$, $b_t = 1$ and $b_{i+1} \cdot b_i = 0$ for $0 \leq i < t$. We give the detail descriptions of the following three reduction formulae by applying Lemma 2.2 and Lemma 3.1. These formulae play a key role in our algorithm. Suppose that the Miller algorithm is performed by an addition/subtraction chain and glues the divisors in the trace of the point addition $(Z \pm P) + Z$ after $Z \pm P$ in the three cases. Note that, $L_Q(S)$ and $L_{-Q}(S)$ have the same value for points Q and S on an elliptic curve, and we can omit the minus signs as remarked in Section 2.3.

1) Case (0, 0) performs:

$$f \leftarrow f^2 \frac{L_{Z,Z}(S)}{L_{2Z}(S)}; Z \leftarrow 2Z;$$
$$f \leftarrow f^2 \frac{L_{2Z,2Z}(S)}{L_{4Z}(S)}; Z \leftarrow 2Z;$$

Putting together, we have:

$$f \leftarrow (f^2 \frac{L_{Z,Z}(S)}{L_{2Z}(S)})^2 \frac{L_{2Z,2Z}(S)}{L_{4Z}(S)} = f^4 \frac{-L^2_{Z,Z}(S)}{L_{2Z,2Z}(-S)}$$

Omitting the minus sign, we have:

$$f \leftarrow f^4 \frac{L^2_{Z,Z}(S)}{L_{2Z,2Z}(-S)}; Z \leftarrow 4Z.$$

2) Case (0,1) performs:

$$f \leftarrow f^2 \frac{L_{Z,Z}(S)}{L_{2Z}(S)}; Z \leftarrow 2Z;$$
$$f \leftarrow f_1 f^2 \frac{L_{Z,P}(S)L_{Z+P,Z}(S)}{L_{Z+P}(S)L_{2Z+P}(S)}; Z \leftarrow 2Z + P;$$

Putting together, we have:

$$\begin{aligned} f &\leftarrow (f^2\frac{L_{Z,Z}(S)}{L_{2Z}(S)})^2 \cdot f_1 \frac{L_{2Z,P}(S)}{L_{2Z+P}(S)} \frac{L_{2Z+P,2Z}(S)}{L_{4Z+P}(S)} \\ &= f_1 f^4 \frac{L^2_{Z,Z}(S)}{L^2_{2Z}(S)} \frac{L_{2Z,P}(S)}{L_{2Z+P}(S)} \\ &\quad \cdot \frac{L_{2Z+P,2Z}(S)L_{2Z+P,2Z}(-S)}{L_{4Z+P}(S)L_{2Z+P,2Z}(-S)} \\ &= f_1 f^4 \frac{L^2_{Z,Z}(S)}{L^2_{2Z}(S)} \frac{L_{2Z,P}(S)}{L_{2Z+P}(S)} \\ &\quad \cdot [-\frac{L_{2Z+P}(S)L_{2Z}(S)L_{4Z+P}(S)}{L_{4Z+P}(S)L_{2Z+P,2Z}(-S)} \\ &= f_1 f^4 \frac{-L^2_{Z,Z}(S)L_{2Z,P}(S)}{L_{2Z}(S)L_{2Z+P,2Z}(-S)}. \end{aligned}$$

Omitting the minus sign, we have:

$$f \leftarrow f_1 f^4 \frac{L^2_{Z,Z}(S)L_{2Z,P}(S)}{L_{2Z}L_{2Z+P,2Z}(-S)}; Z \leftarrow 4Z + P.$$

3) Case (0, -1) performs:

$$f \leftarrow f^2 \frac{L_{Z,Z}(S)}{L_{2Z}(S)}; Z \leftarrow 2Z;$$
$$f \leftarrow \frac{f^2}{f_1} \frac{L_Z(S)}{L_{-Z,P}(S)} \frac{L_{Z-P,Z}(S)}{L_{2Z-P}(S)}; Z \leftarrow 2Z - P;$$

Putting together, we have:

$$\begin{aligned} f &\leftarrow (f^2\frac{L_{Z,Z}(S)}{L_{2Z}(S)})^2 \cdot \frac{1}{f_1} \frac{L_{2Z}(S)}{L_{2Z,P}(S)} \frac{L_{2Z-P,2Z}(S)}{L_{4Z-P}(S)} \\ &= \frac{f^4}{f_1} \frac{L^2_{Z,Z}(S)}{L^2_{2Z}(S)} \frac{L_{2Z}(S)}{L_{-2Z,P}(S)} \\ &\quad \cdot \frac{L_{2Z-P,2Z}(S)L_{2Z-P,2Z}(-S)}{L_{4Z-P}(S)L_{2Z-P,2Z}(-S)} \\ &= \frac{f^4}{f_1} \frac{L^2_{Z,Z}(S)}{L^2_{2Z}(S)L_{-2Z,P}(S)} \\ &\quad \cdot [-\frac{L_{2Z-P}(S)L_{2Z}(S)L_{4Z-P}(S)}{L_{4Z-P}(S)L_{2Z-P,2Z}(-S)} \\ &= \frac{f^4}{f_1} \frac{-L^2_{Z,Z}(S)L_{2Z-P}(S)}{L_{-2Z,P}(S)L_{2Z-P,2Z}(-S)}. \end{aligned}$$

Omitting the minus sign, we have:

$$f \leftarrow \frac{f^4}{f_1} \frac{L^2_{Z,Z}(S)L_{2Z-P}(S)}{L_{-2Z,P}(S)L_{2Z-P,2Z}(-S)}; Z \leftarrow 4Z - P.$$

From these formulae, there is only one line $L_{Z,Z}$ (or $L_{2Z,2Z}$) which needs to be evaluated at point S whence the relative bit of n is zero. That is, we can eliminate one field multiplication when we glue the divisors in the three cases (0, 0), (0, 1) and (0, -1). These detail descriptions of the three cases also provide the correctness of an improved Miller algorithm which we will describe in Algorithm 2.

Algorithm 2 The improved Miller algorithm

1: INPUT: Elliptic curve E, integer $n = \sum_{i=0}^{t} b_i 2^i$ with $b_i \in \{0, 1\}$, $b_t = 1$, $b_{i+1} \cdot b_i = 0$ for $0 \le i < t$, and points P, $S \in E$ where P has order n.
2: OUTPUT: $f = f_n(S)$.
3: $f \leftarrow f_1$; $Z \leftarrow P$; $i \leftarrow t - 1$;
4: while $i > 0$ do
5: if $(b_i, b_{i-1}) = (0, 0)$ then
6: $f \leftarrow f^4 \frac{L^2_{Z,Z}(S)}{L_{2Z,2Z}(-S)}$; $Z \leftarrow 4Z$; $i \leftarrow i - 2$; {Case 0}
7: if $(b_i, b_{i-1}) = (0, 1)$ then
8: $f \leftarrow f_1 f^4 \frac{L^2_{Z,Z}(S)L_{2Z,P}(S)}{L_{2Z}(S)L_{2Z+P,2Z}(-S)}$; $Z \leftarrow 4Z + P$; $i \leftarrow i - 2$; {Case 1}
9: if $(b_i, b_{i-1}) = (0, -1)$ then
10: $f \leftarrow \frac{f^4}{f_1} \frac{L^2_{Z,Z}(S)L_{2Z-P}(S)}{L_{-2Z,P}(S)L_{2Z-P,2Z}(-S)}$; $Z \leftarrow 4Z - P$; $i \leftarrow i - 2$; {Case 2}
11: if $(b_i, b_{i-1}) = (1, 0)$ then
12: $f \leftarrow f_1 f^2 \frac{L_{Z,P}(S)L_Z(S)}{L_{Z+P,Z}(-S)}$; $Z \leftarrow 2Z + P$; $i \leftarrow i - 1$; {Case 3}
13: if $(b_i, b_{i-1}) = (-1, 0)$ then
14: $f \leftarrow \frac{f^2}{f_1} \frac{L_Z(S)L_{Z-P,Z}(S)}{L_{-Z,P}(S)L_{2Z-P}(S)}$; $Z \leftarrow 2Z - P$; $i \leftarrow i - 1$; {Case 4}
15: end-while
16: if $i = 0$ then
17: if $b_i = 1$ then
18: $f \leftarrow f^2 L_{Z,Z}(S)$; $Z \leftarrow 2Z$;
19: if $b_i = 1$ then
20: $f \leftarrow f_1 f^2 \frac{L_{Z,P}(S)L_Z(S)}{L_{Z+P,Z}(-S)}$; $Z \leftarrow 2Z + P$;
21: if $b_i = -1$ then
22: $f \leftarrow \frac{f^2}{f_1} \frac{L_Z(S)L_{Z_P,Z}(S)}{L_{-Z,P}(S)}$; $Z \leftarrow 2Z - P$;
23: return f;
24: End

4 Analysis

In this section, detailed analysis of the improvement is given. Additionally, the estimation of the cost of the improvement is in accordance with the rules which were discussed in [8]. The basic concept of the improvement is that it tries to find the maximum number of the pattern (0, 0) and only processes the first bit of the pattern (1, 0) in Case 3 and the pattern (-1, 0) in Case 4. It is noticeable that the methods of Case 3 and Case 4 can be replaced with the parabola substitution method which was described in [8]. As a result, only one line has to be evaluated for each zero bit of n in our improvement.

As indicated in [1, 3, 8], in the actual implementation of pairing computation, the operations in the numerator and denominator in each step are separated and perform one division at the very end. In [8], they estimate the total cost of pairing computation with the following specifications:

1) The pairing evaluates a quotient of the form $\frac{f_n(Q_1)}{f_n(Q_2)}$ for two points Q_1, Q_2 on E, where n is a t bits integer which consists of $2t/3$ zero bits and $t/3$ nonzero bits.

2) The cost of each bit is counted as the total number of field multiplications, but the cost of all field additions/subtractions are omitted.

3) The cost of a division is estimated as 5.18 field multiplications.

4) The cost of the standard algorithm is 16.18 field multiplications for each zero bit and 31.36 field multiplications for each nonzero bit.

5) The cost of the ELM method is 16.18 field multiplications for each zero bit and 26.36 field multiplications for each nonzero bit.

For simplicity, our estimation follows the analysis in [8] which counts the cost in each case separately. Plus, only three different cases need to be analyzed between our improvement and the ELM method. These are the cases of the cost of bit 0 and the cost of bit ±1 in (0, 1) and (0, -1):

1) The cost of bit 0 which appears in the patterns (0, 0), (0, 1) and (0, -1): In these cases, we must perform $f \leftarrow f^2 L_{Z,Z}(S)$ and $Z \leftarrow 2Z$ for each bit 0.

   a. A point doubling operation costs 3 field multiplications and a division.

   b. Evaluating $L_{Z,Z}$ at points Q_1 and Q_2 costs 2 field multiplications.

   c. Multiplying 4 fractions f_{nu}, f_{de}, $L_{Z,Z}(Q_1)$, and $L_{Z,Z}(Q_2)$ costs 4 field multiplications. Where f_{nu} is the numerator of f and f_{de} is the denominator of f. That is, we must compute $\frac{f_{nu} \cdot f_{nu} \cdot L_{Z,Z}(Q_1)}{f_{de} \cdot f_{de} \cdot L_{Z,Z}(Q_2)}$ in the improvement whence the relative bit is 0.

   The total cost of this case is $3 + 5.18 + 2 + 4 = 14.18$ field multiplications.

2) The cost of bit 1 of (0, 1): In this case, we must perform $f \leftarrow f_1 f^4 \frac{L^2_{Z,Z}(S) L_{2Z,P}(S)}{L_{2Z}(S) L_{2Z+P,2Z}(-S)}$ and $Z \leftarrow 4Z + P$. Then they can be separated as:

$$f \leftarrow [f^2 L^2_{Z,Z}(S)]^2 f_1 \frac{L_{2Z,P}(S)}{L_{2Z}(S) L_{2Z+P,2Z}(-S)}$$

   and $Z \leftarrow (2Z+P)+2Z$. The cost of the first component is estimated in A. We estimate the cost of the second component as follows:

   a. A point double-addition costs 3 field multiplications and 2 divisions.

   b. Evaluating $L_{2Z,P}$ at points Q_1 and Q_2 costs 2 field multiplications. Evaluating $L_{2Z+P,2Z}$ at points $-Q_1$ and $-Q_2$ costs 2 field multiplications.

   c. Multiplying 10 fractions costs 10 field multiplications.

   The total cost of this case is $3+10.36+4+10 = 27.36$ field multiplications.

3) The cost of bit -1 which appear in the pattern (0, -1): In this case, we must perform $f \leftarrow \frac{f^4}{f_1} \frac{L^2_{Z,Z}(S) L_{2Z-P}(S)}{L_{-2Z,P}(S) L_{2Z-P,2Z}(-S)}$ AND $Z \leftarrow 4Z - P$. Then they can be separated as:

$$f \leftarrow [F^2 l_{z,z}(s)]^2 \frac{L_{2Z-P}(S)}{f_1 \cdot L_{-2Z,P}(S) L_{2Z-P,2Z}(-S)}$$

   and $Z \leftarrow (2Z-P)+2Z$. The cost of the first component is estimated in A. We estimate the cost of the second component as follows:

   a. A point double-subtraction costs 3 field multiplications and 2 divisions.

   b. Evaluating $L_{-2Z,P}$ at points Q_1 and Q_2 costs 2 field multiplications. Evaluating $L_{2Z-P,2Z}$ at points $-Q_1$ and $-Q_2$ costs another 2 field multiplications.

   c. Multiplying 10 fractions costs 10 field multiplications.

   The total cost of this case is $3 + 10.36 + 2 + 2 + 10 = 27.36$ field multiplications.

Before we compute the average cost of our refinement, we define two sets, ODD and $EVEN$, for the pattern w, which appears in the NAF representation of n, where $n = \sum_{i=0}^{t} b_i 2^i$ with $b_i \in \{0, 1\}$, $b_t = 1$ and $b_{i+1} \cdot b_i = 0$ for $0 \le i < t$. That is, $ODD = \{w = b_{i+r+1}(0, 0, \cdots, 0)b_i$: r is odd, $b_{i+r+1} \cdot b_i \neq 0$, $0 \le i < i+r+1 \le t\}$ and $EVEN = \{w = b_{i+r+1}(0, 0, \cdots, 0)b_i$: r is even, $b_{i+r+1} \cdot b_i \neq 0$, $0 \le i < i + r + 1 \le t\}$.

Without lost of generality, assume $|ODD| = |EVEN|$. Accordingly, the total number of Case 1 and Case 2 is estimated as the same as the total number of Case 3 and Case 4 in Algorithm 2. That is, in half of all nonzero bits, each bit costs 27.36 field multiplications and each bit of the rest costs 26.36 field multiplications. Therefore, the average cost of our improvement is $\frac{14.18 \times 2t/3 + 27.36 \times t/6 + 26.36 \times t/6}{t} = 18.41$ field multiplications per bit. Compared to the standard algorithm, the improvement is $\frac{21.24 - 18.41}{21.24} = 13.3\%$. In other words, we enhance the ELM method to obtain a $\frac{19.57 - 18.41}{19.57} = 5.9\%$ improvement in performance.

5 Concluding Remarks

An improvement in the computation of the pairings has been given and the corresponding performance has been analyzed. It is noticeable that this algorithm can be more efficient if more lines belonging to the nonzero bits are reduced. We can achieve this purpose by recoding the NAF representation of n into many patterns, such as (0^r), $(0, 1, 0)$ and $(0, -1, 0)$. But this is getting half the result with twice the effort. Therefore, we propose a concise

algorithm which focuses on performance improvement of the zero bits and gives a simplified performance analysis. As a result, the proposed algorithm gains an improvement of 5.9% in performance when compared to the ELM method.

Acknowledgments

This study was supported by the Asia University, under grant No. 98-NSC-02 and 101-ASIA-34. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] P. Barreto, H. Kim, B. Lynn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems," in *Advance in Cryptography (Crypt'02)*, LNCS 2442, pp. 354–368, Springer-Verlag, 2002.

[2] S. Basu, "A new parallel window-based implementation of the elliptic curve point multiplication in multi-core architectures," *International Journal of Network Security*, vol. 14, no. 2, pp. 101–108, 2012.

[3] I. Blake, V. Murty, and G. Xu, "Refinement of Miller's algorithm for computing the Weil/Tate pairing," *Journal of Algorithms*, vol. 58, pp. 134–149, 2006.

[4] D. Boneh, and M. Franklin, "Identity-base encryption from the Weil pairing," in *Advance in Cryptography (Crypto'01)*, LNCS 2139, pp. 213–239, Springer-Verlag, 2001.

[5] D. Boneh, B. Lynn, and H. Shacham, "Short signature from the weil pairing," in *Advance in Cryptography (Asiacrypt'01)*, LNCS 2248, pp. 514–532, 2Springer-Verlag, 001.

[6] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Advance in Cryptography (Crypto'04)*, LNCS 3152, pp. 41–55, Springer-Verlag, 2004.

[7] Y. Ding, K. W. Wong, and Y. M. Wang, "Joint sparse form of window three for Koblitz curve," *International Journal of Network Security*, vol. 2, no. 2, pp. 126–130, 2006.

[8] K. Eisentr?ger, K. Lauter, and P. L. Montgomery, "Fast elliptic curve arithmetic and improved Weil pairing evaluation," in *Topics in Cryptology (CT-RSA'03)*, LNCS 2612, pp. 343–354, Springer-Verlag, 2003.

[9] G. Frey and H. Ruck, "A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves," *Mathematics of Computation*, vol. 62, pp. 865–874, 1994.

[10] S. Galbraith, K. Harrison, and D. Soldera, "Implementing the tate pairing," in *Algorithm Number Theory Symposium*, LNCS 2369, pp. 324–337, Springer-Verlag, 2002.

[11] IEEE Computer Society, *IEEE Standard Specifications for Public-key Cryptography*, IEEE Standard 1363–2000, 2000.

[12] A. Joux, "A one round protocol for tripartite Diffie-Hellman," in *Algorithmic Number Theory*, LNCS 1838, pp. 385–393, Springer-Verlag, 2000.

[13] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, pp. 203–209, 1987.

[14] D. P. Le and C. L. Liu, "Refinements of Miller's algorithm over weierstrass curves revisited," *The Computer Journal*, vol. 54, no. 10, pp. 1582–1591, 2011.

[15] J. Lee, H. Kim, Y. Lee, S. M. Hong, and H. Yoon, "Parallelized scalar multiplication on elliptic curves defined over optimal extension field," *International Journal of Network Security*, vol. 4, no. 1, pp. 99–106, 2007.

[16] X. Li and K. Chen, "Identity based proxy-signcryption scheme from pairings," in *Proceedings of 2004 IEEE International Conference on Services Computing (IEEE-SCC'04)*, pp. 494–497, 2004.

[17] T. C. Lin, "Algorithms on elliptic curves over fields of characteristic two with non-adjacent forms," *International Journal of Network Security*, vol. 9, no. 2, pp. 117–120, 2009.

[18] C. Liu, G. Horng, and T. Chen. "Further refinement of pairing computation based on Miller's algorithm," *Applied Mathematics and Computation*, vol. 189, no. 1, pp. 395–409, 2007.

[19] A. Menezes, T. Okamoto, and S. A. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field," *IEEE Transaction on Information Theory*, vol. 39, pp. 1639–1646, 1993.

[20] A. Menezes, *Elliptic Curve Cryptosystems*, Kluwer Academic Publishers, 1993.

[21] V. Miller, "Use of elliptic curves in cryptosystems," in *Advance in Cryptography (Crypto'85)*, LNCS 218, pp. 417–426, Springer-Verlag, 1986.

[22] V. Miller, *Short Programs for Functions on Curve*, Sept. 2002. (http://www.researchgate.net/profile/Victor_Miller/publication/2551688_Short_Programs_for_functions_on_Curves/links/ 0c96052e065ca0bdbf000000.pdf)

[23] S. Moon, "A binary redundant scalar point multiplication in secure elliptic curve cryptosystems," *International Journal of Network Security*, vol. 3, no. 2, pp. 132–137, 2006.

[24] D. Nalla and K. Reddy, "Signcryption scheme for identity-based cryptosystems," *Cryptology ePrint Archive*, Report, 2003/066, 2003.

[25] H. Sahu and B. K. Sharma, "An MSS based on the elliptic curve cryptosystem," *International Journal of Network Security*, vol.11, no. 2, pp. 118, 2010.

[26] R. Sakai, K. Ohgishi, and M. Kasahara, "Cryptosystems based on pairing," in *SCIS'00*, pp. 26–28, 2000.

[27] Z. J. Shi and H. Yun, "Software implementations of elliptic curve cryptography," *International Journal of Network Security*, vol. 7, no. 1, pp. 141–150, 2008.

[28] S. M. Shohdy, A. B. El-Sisi, and N. Ismail, "Hardware implementation of efficient modified karatsuba multiplier used in elliptic curves," *International Journal of Network Security*, vol. 11, no. 3, pp. 155–162, 2010.

[29] N. P. Smart, "An identity based authenticated key agreement protocol based on weil pairing," *Electronics Letters*, vol. 38, pp. 630–632, 2002.

[30] J. H. Silverman, *The Arithmetic of Elliptic Curves*, Graduate Texts in Mathematics, vol. 106, Springer, New York, 1986.

[31] T. Wu, H. Du, M. Zhang, and R. Wang, "Improved algorithm of the tate pairing in characteristic three," in *The First International Symposium on Data, Privacy, and E-Commerce*, pp. 453–455, 2007.

[32] T. Wu, M. Zhang, X. Xu, and R. Wang, "Improved algorithm for tate pairing computation," in *Proceedings of The IEEE International Symposium on Electronic Commerce and Security (ISECS'08)*, pp. 41–45, 2008.

[33] D. Yong, Y. F. Hong, W. T. Wang, Y. Y. Zhou, and X. Y. Zhao, "Speeding Scalar Multiplication of Elliptic Curve over $GF(2^{mn})$," *International Journal of Network Security*, vol. 11, no. 2, pp. 70–77, 2010.

Chao-Liang Liu was born in Hsinchu, Taiwan, in 1966. He completed his B.Sc. degree in mathematics from National Cheng-Kung University, Taiwan, in 1989, and the Ph.D. degree in Institute of Computer Science from National Chung Hsing University, Taiwan, in 2007. He is currently an assistant professor at the Department of Applied Informatics and Multimedia, Asia University, Taiwan. His research interests include information security, cryptography, elliptic curve cryptosystem and pairing computation.

Gwoboa Horng received the B.S. degree in Electrical Engineering from National Taiwan University in 1981 and the M.S. and Ph.D. degrees from University of Southern California in 1987 and 1992 respectively, all in Computer Science. Since 1992, he has been on the faculty of the Department of Computer Science and Engineering at National Chung-Hsing University, Taichung, Taiwan, R.O.C. His current research interests include artificial intelligence, cryptography and information security.

Du-Shiau Tsai received the B.S. degree in Department of Computer Science and Information Management from Providence University, Taiwan, in 1996 and the M.S. degree in Institute of Computer Science, National Chung Hsing University, Taiwan, in 2003. Since 2005, he has been on the faculty at Hsiuping University of Science and Technology, Taiwan, ROC. He received the Ph.D. degree in the Institute of Computer Science, National Chung Hsing University, Taiwan, in 2007. His research interests include cryptography, information security, secret image sharing, image processing and digital watermarking.