

Further Refinements of Miller Algorithm on Edwards curves

Duc-Phong Le*, Chik How Tan

*Temasek Laboratories, National University of Singapore
5A Engineering Drive 1, #09-02, Singapore 117411.*

Abstract

Recently, Edwards curves have received a lot of attention in the cryptographic community due to their fast scalar multiplication algorithms. Then, many works on the application of these curves to pairing-based cryptography have been introduced. Xu and Lin (CT-RSA, 2010) presented refinements to improve the Miller algorithm that is central role compute pairings on Edwards curves. In this paper, we study further refinements to Miller algorithm. Our approach is *generic*, hence it allow to compute *both* Weil and Tate pairings on pairing-friendly Edwards curves of *any* embedding degree. We analyze and show that our algorithm is faster than the original Miller algorithm and the Xu-Lin's refinements.

Keywords: Miller algorithm, Pairing computation, Edwards curves, Tate/Weil pairings

1. Introduction

In 2007, Bernstein and Lange [8] introduced Edwards curves to cryptography. Their study and subsequent works in [9, 2, 18, 10] showed that the addition law on such curves is more efficient than all previously known formulas. Edwards curves have thus attracted great interest in applications that require elliptic curve operations to achieve faster arithmetic. Then, the application of Edwards curves to pairing-based cryptography has been studied in several research papers [15, 19, 1, 28, 23]. Although, pairing computation on Edwards curves is slightly slower than on Weierstrass curves so far. However, in many pairing-based cryptosystems, the most time-consuming operation is still to compute scalar multiples aP of a point P .

Pairing (or *bilinear map*) is probably the most useful cryptographic tool in the 2000s. It was first introduced to cryptography in Joux's seminal paper [20] in 2001 that describes a tripartite (bilinear) Diffie-Hellman key exchange. Then, the use of cryptosystems based on pairings has had a huge success with some notable breakthroughs such as the first identity-based encryption scheme [4], the short signature scheme [11].

Ever since it was first described, Miller's algorithm [24, 25] has been the heart of the computation of pairings on elliptic curves. Many papers are devoted to improvements in its efficiency. For example, it can run faster on pairing-friendly elliptic curves that belong to specific families [5, 12, 13, 6]. Another approach of improving the Miller's algorithm is to reduce the Miller-loop length by introducing variants of Tate pairings, for example Eta pairing [7], Ate pairing [17], and particular *optimal pairings* [26, 16]. For a more generic approach, studies

*Corresponding author

Email addresses: tslld@nus.edu.sg (Duc-Phong Le), tsltch@nus.edu.sg (Chik How Tan)

in [14, 3, 22] improved the performance for computing pairings of any type (*i.e.*, Weil, Tate, optimal pairings), and on *generic* pairing-friendly elliptic curves.

Basically, Miller’s algorithm is based on a rational function g of three points P_1, P_2, P_3 . This function is called Miller function and has its divisor $\text{div}(g) = (P_1) + (P_2) - (P_3) - (\mathcal{O})$, where \mathcal{O} is a distinguished rational point. For curves of Weierstrass form, this function is defined to be the line passing through points P_1 and P_2 divided by the vertical line passing through the point P_3 , where $P_3 = (P_1 + P_2)$. On Edwards curves, finding such a point P_3 is not straightforward as in Weierstrass curves because Edwards equation has degree 4, *i.e.* any line has 4 intersections with the curves instead of 3 on Weierstrass curves.

In [1], Arene *et al.* presented the first geometric interpretation of the group law on Edwards curves and showed how to compute Tate pairing on twisted Edwards curves by using a conic \mathcal{C} of degree 2. They also introduced explicit formulas with a focus on curves having an even *embedding degree*¹. In order to speed up the pairing computation on *generic* Edwards curves, Xu and Lin [28] proposed refinements to Miller’s algorithm. Their refinements are inspired from Blake-Murty-Xu’s refinements on Weierstrass curves [14]. Their refinements are *generally* faster than the original Miller’s algorithm on Edwards curves described in [1].

In this paper, we further extend the Blake-Murty-Xu’s method on Edwards curves and propose new refinements of Miller’s algorithm. Similarly Xu-Lin’s refinements, our approach is *generic*. Although it did not bring a dramatic efficiency as that of Arene *et al.* for computing Tate pairing over Edwards curves with *even* embedded degree, but the proposed refinements can be used to compute pairing of *any* type over pairing-friendly Edwards curves with *any* embedding degree. This approach is of particular interest to compute optimal pairings [26, 16], and in situations where the denominator elimination technique using a twist is not possible (e.g., Edwards curves with *odd* embedding degrees).² We also analyze and show that our new algorithm is faster than the original Miller’s algorithm and its refinements presented in [28].

The paper is organized as follows. In Section 2 we briefly recall some background on pairing computation on Edwards curves, and then the Xu-Lin’s refinements. In Section 3 we present our refinements of Miller’s algorithm. Section 4 gives some discussion on performance of the proposed algorithms. Section 5 is our conclusion.

2. Preliminaries

2.1. Pairings on Edwards Curves

Let \mathbb{F}_p be a prime finite field, where p is a prime different from 2. A *twisted Edwards curve* $E_{a,d}$ defined over \mathbb{F}_p is the set of solutions (x, y) of the following *affine equation*:

$$E_{a,d} : ax^2 + y^2 = 1 + dx^2y^2, \tag{1}$$

where $a, d \in \mathbb{F}_p^*$, and $a \neq d$. Edwards curves are special case of twisted Edwards curves where a can be rescaled to 1. Twisted Edwards curves have the fastest doubling and addition operations in elliptic curve cryptography (see [2]).

¹Let E be an elliptic curve defined over a prime finite field \mathbb{F}_p , and r be a prime dividing $\#E(\mathbb{F}_p)$. The embedding degree of E with respect to r is the smallest positive integer k such that $r \mid p^k - 1$. In other words, k is the smallest integer such that $\mathbb{F}_{p^k}^*$ contains r -roots of unity.

²Note that by definition optimal pairings only require about $\log_2(r)/\varphi(k)$ iterations of the basic loop, where r is the group order, φ is Euler’s totient function, and k is the embedding degree. For example, when k is prime, then $\varphi(k) = k - 1$. If we choose a curve having embedding degree $k \pm 1$, then $\varphi(k \pm 1) \leq \frac{k \pm 1}{2}$ which is roughly $\frac{\varphi(k)}{2} = \frac{k-1}{2}$, so that at least twice as many iterations are necessary if curves with embedding degrees $k \pm 1$ are used instead of curves of embedding degree k .

Cryptographic pairing is a bilinear map that takes as input two points on elliptic curves defined over finite fields and returns a value in the extension finite field. The key to the definition of pairings is the evaluation of rational functions in divisors. The pairings over (hyper-)elliptic curves are computed using the algorithm proposed by Miller [25]. The main part of Miller's algorithm is to construct the rational function $f_{r,P}$ and evaluating $f_{r,P}(Q)$ with $\text{div}(f_{r,P}) = r(P) - (rP) - [r-1](\mathcal{O})$ for divisors P and Q . In this section, we just recall the Miller algorithm that is so far the best known method to compute pairings. Readers who want to study more about pairings can take a look at papers [25, 26].

Let m and n be two integers, and $g_{mP,nP}$ be a rational function whose divisor $\text{div}(g_{mP,nP}) = (mP) + (nP) - ([m+n]P) - (\mathcal{O})$. We call the function $g_{mP,nP}$ *Miller function*. Miller's algorithm is based on the following lemma.

Lemma 2.1 (Lemma 2, [25]). *For n and m two integers, up to a multiplicative constant, we have*

$$f_{m+n,P} = f_{m,P} f_{n,P} g_{mP,nP}. \quad (2)$$

Equation (2) is called *Miller relation*, which is proved by considering divisors. The Miller algorithm makes use of Lemma 2.1 with $m = n$ in a doubling step and $n = 1$ in an addition step. For Edwards curves, Arene *et al.* [1] defined Miller's function in the following theorem.

Theorem 2.2 (Theorem 2, [1]). *Let $a, d \in \mathbb{F}_p^*$, $a \neq d$ and $E_{a,d}$ be a twisted Edwards curve over \mathbb{F}_p . Let $P_1, P_2 \in E_{a,d}(\mathbb{F}_p)$. Define $P_3 = P_1 + P_2$. Let ϕ is the equation of the conic \mathcal{C} passing through $P_1, P_2, -P_3, \Omega_1, \Omega_2, \mathcal{O}'$ whose divisor is $(P_1) + (P_2) + (-P_3) + (\mathcal{O}') - 2(\Omega_1) - 2(\Omega_2)$. Let ℓ_{1,P_3} be the horizontal line going through P_3 whose divisor is $\text{div}(\ell_{1,P_3}) = (P_3) + (-P_3) - 2(\Omega_2)$, and $\ell_{2,\mathcal{O}}$ is the vertical line going through \mathcal{O} and \mathcal{O}' whose divisor is $(\mathcal{O}) + (\mathcal{O}') - 2(\Omega_1)$. Then we have*

$$\text{div} \left(\frac{\phi_{P_1, P_2}}{\ell_{1, P_3} \ell_{2, \mathcal{O}}} \right) \sim (P_1) + (P_2) - (P_3) - (\mathcal{O}). \quad (3)$$

The rational function $g_{P_1, P_2} = \frac{\phi_{P_1, P_2}}{\ell_{1, P_3} \ell_{2, \mathcal{O}}}$ consisting of three terms, can be thus considered as Miller function on Edwards curves. Miller's algorithm for Edwards curves using this function works as in Algorithm 1.

Input: $r = \sum_{i=0}^t r_i 2^i$ with $r_i \in \{0, 1\}$, $P, Q \in E[r]$;

Output: $f = f_r(Q)$;

$R \leftarrow P, f \leftarrow 1, g \leftarrow 1$

for $i = t - 1$ **to** 0 **do**

$f \leftarrow f^2 \cdot \phi_{R,R}(Q)$

$g \leftarrow g^2 \cdot \ell_{1,\mathcal{O}}(Q) \cdot \ell_{2,2R}(Q)$

$R \leftarrow 2R$

if $(r_i = 1)$ **then**

$f \leftarrow f \cdot \phi_{R,P}(Q)$

$g \leftarrow g \cdot \ell_{1,\mathcal{O}}(Q) \cdot \ell_{2,R+P}(Q)$

$R \leftarrow R + P$

end

end

return f/g

Algorithm 1: Miller's Algorithm for twisted Edwards curves [28]

2.2. Xu-Lin Refinements

For simplicity, in what follows, we make use of the notation $\phi_{P,P}$ (resp. $\ell_{2,[2]P}$, and $\ell_{1,\mathcal{O}}$) replacing for $\phi_{P,P}(Q)$ (resp. $\ell_{2,[2]P}(Q)$, and $\ell_{1,\mathcal{O}}(Q)$). By extending the Blake et al.'s method [14] to Edwards curves, Xu and Lin [28] presented a refinement to Miller algorithm. Their algorithm was achieved owing to the following theorem.

Theorem 2.3 (Theorem 1 in [28]). *Let $E_{a,d}$ be a twisted Edwards curve over \mathbb{F}_p and $P, R \in E_{a,d}$ be a point of order r . Then³*

1.

$$\left(\frac{\phi_{R,R}}{\ell_{2,[2]R}\ell_{1,\mathcal{O}}}\right)^2 \frac{\phi_{[2]R,[2]R}}{\ell_{2,[4]R}\ell_{1,\mathcal{O}}} = \frac{\phi_{R,R}^2}{\phi_{[-2]R,[-2]R}\phi_{\mathcal{O},\mathcal{O}}};$$

2.

$$\frac{\phi_{R,R}}{\ell_{2,[2]R}\ell_{1,\mathcal{O}}} \frac{\phi_{[2]R,P}}{\ell_{2,[2]R+P}\ell_{1,\mathcal{O}}} = \frac{\phi_{R,R}\ell_{2,P}}{\phi_{[2]R+P,-P}\ell_{1,\mathcal{O}}}.$$

The above theorem was proven by calculating divisors (see [28] for more details). From this theorem, they introduced refinements and an improved Miller algorithm in radix-4 representation [28, Algorithm 3]. They also claimed that the total cost of the proposed algorithm is about 76.8% of that of the original Miller's algorithm.

3. Our Improvements on Miller's Algorithm

3.1. First Improvement

We first present a new rational function whose divisor is equivalent to Miller function (Eq 3) presented in [1].

Definition 3.1. *Let $E_{a,d}$ be a twisted Edwards curve and $R, P \in E_{a,d}$. Let $\phi_{R,P}$ be a conic passing through R and P , $\phi_{-R,-P}$ be a conic passing through $-R$ and $-P$, and let $\phi_{R+P,-[R+P]}$ a conic passing through $R+P$ and $-[R+P]$. Then we define*

$$g_{R,P} = \frac{\phi_{R,P}}{\phi_{R+P,-[R+P]}}. \quad (4)$$

Lemma 3.1. *We have*

$$\text{div}(g_{R,P}) = (R) + (P) - ([R+P]) - \mathcal{O}.$$

Proof. By calculating divisors, it is straightforward to see that

$$\begin{aligned} \text{div}(g_{R,P}) &= (R) + (P) + (-[R+P]) + (\mathcal{O}') - 2(\Omega_1) - 2(\Omega_2) \\ &\quad - ([R+P]) - (-[R+P]) - (\mathcal{O}) - (\mathcal{O}') + 2(\Omega_1) + 2(\Omega_2) \\ &= (R) + (P) - ([R+P]) - (\mathcal{O}). \end{aligned}$$

which concludes the proof. □

Recall that, on Weierstrass curves, the Miller function $g_{R,P} = \frac{\ell_{R,P}}{v_{R+P}}$, where $\ell_{R,P}, v_{R+P}$ are lines passing through R, P and $R+P, -[R+P]$, respectively. The Eq. (4) thus looks similar to the Miller function on Weierstrass curves if a conic plays role as a line function. A variant of Miller algorithm by using Eq. 4 is described in Algorithm 2.

³There were typos in the first formula of Theorem 1 in [28]. It should be $\ell_{2,[2]R}\ell_{1,\mathcal{O}}$ instead of $\ell_{1,[2]R}\ell_{2,\mathcal{O}}$.

Input: $r = \sum_{i=0}^t r_i 2^i$ with $r_i \in \{0, 1\}$, $P, Q \in E[r]$;

Output: $f = f_r(Q)$;

$R \leftarrow P, f \leftarrow 1, g \leftarrow 1$

for $i = t - 1$ **to** 0 **do**

$f \leftarrow f^2 \cdot \phi_{R,R}(Q)$

$g \leftarrow g^2 \cdot \phi_{2R,-2R}(Q)$

$R \leftarrow 2R$

if $(r_i = 1)$ **then**

$f \leftarrow f \cdot \phi_{R,P}(Q)$

$g \leftarrow g \cdot \phi_{R+P,-(R+P)}(Q)$

$R \leftarrow R + P$

end

end

return f/g

Algorithm 2: First improvement of Miller's Algorithm for twisted Edwards curves

Remark : As the original Miller's algorithm, our algorithm cannot avoid divisions needed to update f . But we can reduce them easily to one inversion at the end of the addition chain (for the cost of one squaring in addition at the each step of the algorithm).

At first glance, Algorithm 2 requires only one multiplication for updating the function g instead of two in Algorithm 1. Note that this operation is costly because it is performed in the full extension finite field. In Section 4, we will provide a detailed analysis on the performance of these algorithms. In the following section, we introduce a further refinement that even offers a better performance in comparison to Algorithm 2.

3.2. Refinement

The new refinement is inspired from the following lemmas.

Lemma 3.2. *We have*

$$g_{R,P} = \frac{\phi_{R,-R} \cdot \phi_{P,-P}}{\phi_{-R,-P} \cdot \phi_{\mathcal{O},\mathcal{O}}}. \quad (5)$$

Proof. This lemma is again proved by considering divisors. Indeed,

$$\begin{aligned} \operatorname{div}\left(\frac{\phi_{R,-R} \cdot \phi_{P,-P}}{\phi_{-R,-P} \cdot \phi_{\mathcal{O},\mathcal{O}}}\right) &= (R) + (-R) + (\mathcal{O}) + (\mathcal{O}') - 2(\Omega_1) - 2(\Omega_2) \\ &\quad + (P) + (-P) + (\mathcal{O}) + (\mathcal{O}') - 2(\Omega_1) - 2(\Omega_2) \\ &\quad - (-R) - (-P) - ([R+P]) - (\mathcal{O}') + 2(\Omega_1) + 2(\Omega_2) \\ &\quad - 3(\mathcal{O}) - (\mathcal{O}') + 2(\Omega_1) + 2(\Omega_2) \\ &= (R) + (P) - ([R+P]) - (\mathcal{O}) \\ &= \operatorname{div}(g_{R,P}). \end{aligned}$$

which concludes the proof. □

Lemma 3.3. *Let $P, R \in E_{a,d}$ be points of order r , we have*

$$\frac{\phi_{R,R}}{\phi_{R,-R}^2 \cdot \phi_{2R,-2R}} = \frac{1}{\phi_{-R,-R} \cdot \phi_{\mathcal{O},\mathcal{O}}}. \quad (6)$$

This lemma is easy to be proven using Definition 3.1 and Lemma 3.2. The factor $\phi_{\mathcal{O},\mathcal{O}}$ can be precomputed and integrated into the factor $\phi_{-R,-R}$ as follows:

$$\begin{aligned}\phi'_{-R,-R}(Q) &= \phi_{-R,-R}(Q) \cdot \phi_{\mathcal{O},\mathcal{O}}(Q) \\ &= (c_{Z^2}(Z_Q^2 + Y_Q Z_Q) + c_{XY} X_Q Y_Q + c_{XZ} X_Q Z_Q)(X_Q(Z_Q - Y_Q)) \\ &= c_{Z^2} \eta_1 + c_{XY} \eta_2 + c_{XZ} \eta_3,\end{aligned}$$

where $\eta_1 = (Z_Q^2 + Y_Q Z_Q)(X_Q(Z_Q - Y_Q))$, $\eta_2 = X_Q^2 Y_Q(Z_Q - Y_Q)$, $\eta_3 = X_Q^2 Z_Q(Z_Q - Y_Q)$ are fixed for whole computation, thus they can be precomputed and stored. The Eq. 6 can thus be rewritten as follows:

$$\frac{\phi_{R,R}}{\phi_{R,-R}^2 \cdot \phi_{2R,-2R}} = \frac{1}{\phi'_{-R,-R}}. \quad (7)$$

Our algorithm is described by the pseudo-code in Algorithm 3 by applying Eq 7. To do so, the proposed algorithm tries to delay the factor $\phi_{R,-R}$ for the next step. We make use of a memory variable m to imply whether a delayed factor in the current step or not. If $m = 1$, there is a delayed factor for the next step and otherwise. There will be no delayed factor for the next step (*i.e.*, m will be assigned to 0) if the current bit $b_i = 0$ and there exists a delayed factor for the current step (line 2 in the Algorithm 3). This will lead to the most expensive case (line 4) of the proposed algorithm if the next bit $b_{i-1} = 1$ (counting from the most significant bit).

Input: $r = \sum_{i=0}^t b_i 2^i$, $b_i \in \{0, 1\}$.
Output: f

$T \leftarrow P$, $f \leftarrow 1$, $g \leftarrow 1$, $m \leftarrow 0$;

for $i = t - 1$ **to** 0 **do**

1 **if** $(b_i = 0) \wedge (m = 0)$ **then**
 | $f \leftarrow f^2 \cdot \phi_{R,R}$; $g \leftarrow g^2$; $R \leftarrow 2R$;
 end

2 **if** $(b_i = 0) \wedge (m = 1)$ **then**
 | $f \leftarrow f^2$; $g \leftarrow g^2 \cdot \phi'_{-R,-R}$; $R \leftarrow 2R$;
 end

3 **if** $(b_i = 1) \wedge (m = 1)$ **then**
 | $f \leftarrow f^2 \cdot \phi_{2R,P}$; $g \leftarrow g^2 \cdot \phi'_{-R,-R}$; $R \leftarrow 2R + P$;
 end

4 **if** $(b_i = 1) \wedge (m = 0)$ **then**
 | $f \leftarrow f^2 \cdot \phi_{R,R} \cdot \phi_{2R,P}$; $g \leftarrow g^2 \cdot \phi'_{2R,-2R}$; $T \leftarrow 2R + P$;
 end
 $m \leftarrow \neg m \vee b_i$

end

return $\frac{f}{g}$

Algorithm 3: Improved Refinement of Miller's Algorithm for any Pairing-Friendly Edwards Curves

	Doubling	Doubling and Addition
Algorithm 1 (Miller’s algorithm [1, 28])	$2\mathbf{S}_{p^k} + 3\mathbf{M}_{p^k}$ $= 4.6\mathbf{M}_{p^k}$	$2\mathbf{S}_{p^k} + 5\mathbf{M}_{p^k}$ $= 6.6\mathbf{M}_{p^k}$
Algorithm in [1]	$1\mathbf{S}_{p^k} + 1\mathbf{M}_{p^k}$ $= 1.8\mathbf{M}_{p^k}$	$1\mathbf{S}_{p^k} + 2\mathbf{M}_{p^k}$ $= 2.8\mathbf{M}_{p^k}$
Algorithm 2	$2\mathbf{S}_{p^k} + 2\mathbf{M}_{p^k}$ $= 3.6\mathbf{M}_{p^k}$	$2\mathbf{S}_{p^k} + 4\mathbf{M}_{p^k}$ $= 5.6\mathbf{M}_{p^k}$
Algorithm 3	$2\mathbf{S}_{p^k} + 1\mathbf{M}_{p^k}$ $= 2.6\mathbf{M}_{p^k}$	$2\mathbf{S}_{p^k} + 2\mathbf{M}_{p^k} = 3.6\mathbf{M}_{p^k}$ (line 3) $2\mathbf{S}_{p^k} + 3\mathbf{M}_{p^k} = 4.6\mathbf{M}_{p^k}$ (line 4)

Table 1: Comparison of the cost of updating f, g of Algorithms. “Doubling” is when algorithms deal with the bit “ $b_i = 0$ ” and “Doubling and Addition” is when algorithms deal with the bit “ $b_i = 1$ ”.

4. Discussion

In this section, we first compare the proposed algorithm with the original Miller’s algorithm over Edwards curves [1, 28], and the Xu-Lin refinements [28]. Before analyzing the costs of algorithm, we introduce notations for field arithmetic costs. Let \mathbb{F}_{p^m} be an extension of degree m of \mathbb{F}_p for $m \geq 1$ and let \mathbf{I}_{p^m} , \mathbf{M}_{p^m} , \mathbf{S}_{p^m} , and \mathbf{add}_{p^m} the costs for inversion, multiplication, squaring, and addition in the field \mathbb{F}_{p^m} , respectively. We denote by \mathbf{m}_a the multiplication by the curve coefficient a .

The cost of the algorithms for pairing computation consists of three parts: the cost of updating the functions f, g , the cost of updating the point R and the cost of evaluating rational functions at some point Q . Note that during Ate pairing computation, coordinates of the point R that is on the twisted curve, . The analysis in [1] showed that the total cost of updating the point R and coefficients c_{Z^2}, c_{XY} , and c_{ZZ} of the conic is $6\mathbf{M}_{p^e} + 5\mathbf{S}_{p^e} + 2\mathbf{m}_a$ for each doubling step and $14\mathbf{M}_{p^e} + 1\mathbf{m}_a$ for each addition step (see [1, §5] for more details), where $e = k/d$. Without special treatment, this cost is the same for all algorithms.

The most costly operations in pairing computations are operations in the full extension field \mathbb{F}_{p^k} . At high levels of security (i.e. k large), the complexity of operations in \mathbb{F}_{p^k} dominates the complexity of the operations that occur in the lower degree subfields. In this subsection, we only analyze the cost of updating the functions f, g which are generally executed on the full extension field \mathbb{F}_{p^k} . Assume that the ratio of one full extension field squaring to one full extension field multiplication is set to $\mathbf{S}_{p^k} = 0.8\mathbf{M}_{p^k}$, a commonly used value in the literature. It is clear to see that to update functions f and g , the proposed algorithm requires $1\mathbf{M}_{p^k} + 2\mathbf{S}_{p^k}$ for a doubling step (lines 1, 3), and $1\mathbf{M}_{p^k}$ for an addition step (lines 2, 4). TABLE 1 shows the number of operations needed in \mathbb{F}_{p^k} for updating f, g in different algorithms.

From Table 1, for the *generic* case we can see that **Algorithm 3** saves two full extension field multiplication when the bit $b_i = 0$ compared with **Algorithm 1**. When the bit $b_i = 1$, **Algorithm 3** saves two or three full extension field multiplications in comparison to **Algorithm 1**, depending on which case Algorithm 3 executes.

In comparison to Arene *et al.*’s algorithm [1], Algorithm 3 requires one more squaring in the full extension field for each doubling step. However, as already mentioned, Arene *et al.* can only be applied on Edwards curves with an even embedding degree k for Tate pairing computation, while our approach is generic. It can be applied to any (pairing-friendly) Edwards curve and for both the Weil and the Tate pairing.

The refinements in [28] are described in radix 4. Their algorithm allows to eliminate some rational functions from Eq (3) during pairing computation. Let $r = \sum_{i=0}^{l'-1} q_i 4^i$, with $q_i \in \{0, 1, 2, 3\}$. Table 2 compares our algorithm and their algorithm. From Table 2, it clearly see

that **Algorithm 3** is generally faster than the refinements of Miller’s algorithm in [28] for all four cases.

	Algorithm in [28]	Algorithm 3
$q = 0$	$5\mathbf{S}_{p^k} + 3\mathbf{M}_{p^k}$	$4\mathbf{S}_{p^k} + 2\mathbf{M}_{p^k}$
$q = 1$	$4\mathbf{S}_{p^k} + 7\mathbf{M}_{p^k}$	$4\mathbf{S}_{p^k} + 3\mathbf{M}_{p^k}$ (line 3) $4\mathbf{S}_{p^k} + 4\mathbf{M}_{p^k}$ (line 4)
$q = 2$	$4\mathbf{S}_{p^k} + 7\mathbf{M}_{p^k}$	$4\mathbf{S}_{p^k} + 3\mathbf{M}_{p^k}$ (line 3) $4\mathbf{S}_{p^k} + 4\mathbf{M}_{p^k}$ (line 4)
$q = 3$	$4\mathbf{S}_{p^k} + 10\mathbf{M}_{p^k}$	$4\mathbf{S}_{p^k} + 4\mathbf{M}_{p^k}$ (line 3) $4\mathbf{S}_{p^k} + 5\mathbf{M}_{p^k}$ (line 4)

Table 2: Comparison of our algorithm with the refinements in [28].

5. Conclusion

In this paper, we extended the Blake-Murty-Xu’s method to propose further refinements to Miller’s algorithm over Edwards curves. Our algorithm is *generically* more efficient than the original Miller’s algorithm (Algorithm 1) the Xu-Lin’s refinements in [28]. Especially, the proposed algorithm can be applied for computing pairings of any type over any pairing-friendly Edwards curve. This allows the use of Edwards curves with embedding degree not of the form $2^i 3^j$, and is suitable for the computation of *optimal pairings* [26].

References

- [1] Christophe Arène, Tanja Lange, Michael Naehrig, and Christophe Ritzenthaler. Faster computation of the Tate pairing. *Journal of Number Theory*, 131(5):842–857, 2011.
- [2] Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. Twisted Edwards curves. In *Proceedings of the Cryptology in Africa 1st international conference on Progress in cryptology, AFRICACRYPT’08*, pages 389–405. Springer Berlin/Heidelberg, 2008.
- [3] J. Boxall, N. El Mrabet, F. Laguillaumie, D.-P. Le, A variant of miller’s formula and algorithm, in: Proceedings of the 4th international conference on Pairing-based cryptography, Pairing’10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 417–434.
- [4] Dan Boneh and Matthew K. Franklin. Identity-Based Encryption from the Weil Pairing. In *CRYPTO ’01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, pages 213–229. Springer-Verlag, 2001.
- [5] P. S. L. M. Barreto, H. Y. Kim, B. Lynn, M. Scott, Efficient algorithms for pairing-based cryptosystems, in: CRYPTO ’02: Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology, Springer-Verlag, London, UK, 2002, pp. 354–368.
- [6] C. Costello, T. Lange, M. Naehrig, Faster Pairing Computations on Curves with High-Degree Twists, in: P. Nguyen, D. Pointcheval (Eds.), *Public Key Cryptography – PKC 2010*, Vol. 6056 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2010, pp. 224–242.

- [7] P. S. Barreto, S. D. Galbraith, C. Ó. Héigearthaigh, M. Scott, Efficient pairing computation on supersingular abelian varieties, *Des. Codes Cryptography* 42 (3) (2007) 239–271.
- [8] Daniel J. Bernstein and Tanja Lange. Faster addition and doubling on elliptic curves. In *Proceedings of the Advances in Cryptology 13th international conference on Theory and application of cryptology and information security*, ASIACRYPT’07, pages 29–50, Berlin, Heidelberg, 2007. Springer-Verlag.
- [9] D. J. Bernstein, T. Lange, Inverted edwards coordinates, in: *Proceedings of the 17th International Conference on Applied Algebra, Algebraic Algorithms and Error-correcting Codes*, AAECC’07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 20–27.
- [10] D. J. Bernstein, T. Lange, A complete set of addition laws for incomplete edwards curves, *Journal of Number Theory* 131 (5) (2011) 858 – 872, *elliptic Curve Cryptography*.
- [11] Dan Boneh, Ben Lynn, and Hovav Shacham. Short Signatures from the Weil Pairing. In *ASIACRYPT ’01: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security*, pages 514–532. Springer-Verlag, 2001.
- [12] P. S. L. M. Barreto, B. Lynn, M. Scott, On the selection of pairing-friendly groups, in: *Selected Areas in Cryptography*, 2003, pp. 17–25.
- [13] P. S. L. M. Barreto, M. Naehrig, Pairing-friendly elliptic curves of prime order, in: *Proceedings of SAC 2005*, volume 3897 of LNCS, Springer-Verlag, 2005, pp. 319–331.
- [14] Ian F. Blake, V. Kumar Murty, and Guangwu Xu. Refinements of Miller’s algorithm for computing the Weil/Tate pairing. *J. Algorithms*, 58(2):134–149, 2006.
- [15] M. P. Das and P. Sarkar, “Pairing Computation on Twisted Edwards Form Elliptic Curves,” in *Proceedings of the 2nd international conference on Pairing-Based Cryptography*, ser. Pairing ’08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 192–210.
- [16] Florian Hess. Pairing lattices. In *Proceedings of the 2nd international conference on Pairing-Based Cryptography*, Pairing ’08, pages 18–38, Berlin, Heidelberg, 2008. Springer-Verlag.
- [17] Florian Hess, Nigel P. Smart, and Frederik Vercauteren. The eta pairing revisited. *IEEE Transactions on Information Theory*, 52:4595–4602, 2006.
- [18] H. Hisil, K. K.-H. Wong, G. Carter, E. Dawson, Twisted edwards curves revisited, in: *Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, ASIACRYPT ’08, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 326–343.
- [19] S. Ionica and A. Joux, “Another Approach to Pairing Computation in Edwards Coordinates,” in *Progress in Cryptology - INDOCRYPT 2008*, ser. Lecture Notes in Computer Science, D. Chowdhury, V. Rijmen, and A. Das, Eds. Springer Berlin / Heidelberg, 2008, vol. 5365, pp. 400–413.
- [20] Antoine Joux. A One Round Protocol for Tripartite Diffie-Hellman. In *ANTS-IV: Proceedings of the 4th International Symposium on Algorithmic Number Theory*, pages 385–394. Springer-Verlag, 2000.
- [21] Neal Koblitz. *Algebraic aspects of cryptography*. Springer-Verlag New York, Inc., New York, NY, USA, 1998.

- [22] Duc-Phong Le and Chao-Liang Liu. Refinements of Miller’s Algorithm over Weierstrass Curves Revisited. *The Computer Journal*, 54(10):1582–1591, 2011.
- [23] Duc-Phong Le and Chik How Tan. Improved Miller’s Algorithm for Computing Pairings on Edwards Curves. *IEEE Transactions on Computers*, 99(PrePrints):1–8, 2013.
- [24] V. M. Exploratory, V. S. Miller, Short programs for functions on curves, in: IBM Thomas J. Watson Research Center, 1986.
- [25] Victor S. Miller. The Weil Pairing, and Its Efficient Calculation. *Journal of Cryptology*, 17(4):235–261, 2004.
- [26] Frederik Vercauteren. Optimal pairings. *IEEE Transactions on Information Theory*, 56(1):455–461, 2010.
- [27] Brent Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT ’05*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer, 2005.
- [28] Lei Xu and Dongdai Lin. Refinement of Miller’s Algorithm Over Edwards Curves. In Josef Pieprzyk, editor, *CT-RSA*, volume 5985 of *Lecture Notes in Computer Science*, pages 106–118. Springer, 2010.