Improved Miller's Algorithm for Computing Pairings on Edwards Curves

Article *in* IEEE Transactions on Computers · October 2014

DOI: 10.1109/TC.2013.125

2 authors, including:

Lê Đức Phong
Institute for Infocomm Research
26 PUBLICATIONS **65** CITATIONS

SEE PROFILE

Improved Miller's Algorithm for Computing Pairings on Edwards Curves

Duc-Phong Le, and Chik How Tan

Abstract—Since Edwards curves were introduced to elliptic curve cryptography by Bernstein and Lange in 2007, they have received a lot of attention due to their very fast group law operation. Pairing computation on such curves is slightly slower than on Weierstrass curves. However, in some pairing-based cryptosystems, they might require a number of scalar multiplications which is time-consuming operation and this can be advantageous to use Edwards in this scenario. In this paper, we present a variant of Miller's algorithm for pairing computation on Edwards curves. Our approach is *generic*, it is able to compute *both* Weil and Tate pairings on pairing-friendly Edwards curves of *any* embedding degree. Our analysis shows that the new algorithm is faster than the previous algorithms for odd embedding degree and as fast as for even embedding degree. Hence, the new algorithm is suitable for computing *optimal pairings* and in situations where the denominators elimination technique is not possible.

Index Terms—Edwards curves, Pairing-friendly elliptic curves, Miller's algorithm, Pairing computation, Weil/Tate pairings, Pairing-based cryptography.

1 Introduction

Edwards curves and the Edwards group law were first introduced in [1]. Bernstein and Lange [2] then introduced Edwards curves to cryptography and showed that the addition law on Edwards curves is more efficient than all previously known formulas. Edwards curves were then generalized to the twisted Edwards curves [3] that cover considerably more elliptic curves over a finite field than the original ones.

Pairing-based cryptography has received a lot of attention over the past more than ten years. The first notable application of pairings to cryptology was the work of Menezes, Okamato and Vanstone [4]. They showed that the discrete logarithm problem on an elliptic curve can be reduced to the discrete logarithm problem in a finite field in 1991 through the Weil pairing. Then, Frey and Rück [5] also considered this situation using the Tate pairing. Pairings were thus used as a means of attacking cryptosystems.

Nevertheless, pairings on elliptic curves only become a great interest since their first application in constructing cryptographic protocols [6], which describes an one-round 3-party Diffie-Hellman key exchange protocol. Since then, the use of cryptographic protocols based on pairings has had a huge success with some notable breakthroughs such as the first practical Identity-based Encryption (IBE) scheme [7], the short signature scheme [8], and many other new cryptographic primitives [9], [10], [11].

Efficient algorithms for pairing computation play a very important role in pairing-based cryptography. The best known method for computing the Weil and the Tate pairing is based on Miller's algorithm [12] for rational functions from scalar multiplications of divisors. The Weil pairing requires two *Miller loops*, while the Tate pairing requires only one *Miller loop* and a *final exponentiation*; and about two times faster than the Weil pairing.

In comparison to Weierstrass curves, twisted Edwards curves introduce a faster addition law. However, pairing computation over Edwards curves is more complicated than over Weierstrass ones. The following question is important for computing the Weil/Tate pairings on elliptic curves when using Miller's algorithm: given points P_1 and P_2 on an elliptic curve, find a point P_3 $(= P_1 + P_2)$ and a rational function g, called Miller's function such that $div(g) = (P_1) + (P_2) - (P_3) - (\mathcal{O})$, where $\mathcal{O}$ is a distinguished rational point. For curves of Weierstrass form, this function is easy to obtain due to the *chord-and-tangent rule* for addition. While Edwards equation has degree 4, *i.e.* any line has 4 intersections with the curves instead of 3 as in Weierstrass curves. Hence it is not easy to find such a function.

In [13] and [14] computing a pairing uses a bi-rational equivalence that maps an Edwards curve to a curve of degree 3 and then express the Miller's function g by line functions. Arene *et al.* [15] 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

- *Manuscript received June 23, 2012; revised May 16, 2013; accepted May 29, 2013.*

- *D.-P. Le and C.H. Tan are with Temasek Laboratories, National University of Singapore, 5A Engineering Drive 1, 117411, Singapore.*
- *Emails: tslld@nus.edu.sg, tsltch@nus.edu.sg*

on curves having an even *embedding degree*[1]. Although pairing computation on Weierstrass curves is slightly faster than on Edwards curves, in some pairing-based cryptosystems, there required a number of scalar multiplication that can benefit the fastest group law operation on Edwards curves.

Based on Arene *et al.*'s algorithm and inspired from refinements to Miller's algorithm on Weierstrass curves [16], [17], Xu and Lin [18] proposed refinements to Miller's algorithm on Edwards curves. Although this approach did not bring a significant improvement as Arene *et al.*'s, it can be applied for computing both Weil and Tate pairing on pairing-friendly Edwards elliptic curve with *any* embedding degree. For example, Edwards curves with odd embedding degree don't provide a denominator elimination technique, but it may allow a shorter Miller loop.

In this paper, we study a variant of Miller's algorithm for Edwards curves. Similar to Xu and Lin's approach, our new algorithm can also be applied on *any* pairing-friendly Edwards curves and for computing any cryptographic pairing. We analyze and show that our new algorithm is *generally* faster than the original Miller's algorithm on Edwards curves and its refinements [18]. Our variant of Miller's algorithm is particularly interesting to compute optimal pairings [19], [20], and in situations where the denominator elimination technique using a twist is not possible (e.g., Edwards curves with odd embedding degree). Note that optimal pairings only require $\log_2(r)/\varphi(k)$ 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 $\varphi(k \pm 1) \leq \frac{k+1}{2}$ 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.

In this paper, we also show that our algorithm can eliminate denominators when computing Tate pairing on Edwards curves with even embedding degree. The efficiency of this modification can be thus comparable to that of Arene *et al.* [15].

The rest of paper is organized as follows. Section 2 briefly recalls some definitions of Edwards curves, the Weil/Tate pairings, and Miller's algorithm. Section 3 presents our improvements to the original Miller's algorithm for generic pairing-friendly Edwards curves. Section 4 analyzes theoretically the efficiency of our algorithm and compares with previous works. Section 5 is our conclusion.

1. Let 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 $r|p^k - 1$. In other words, k is the smallest integer such that $\mathbb{F}_{p^k}^*$ contains r-roots of unity.

2 Preliminaries

2.1 Edwards curves and Addition law

Let $\mathbb{F}_p$ be a 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 a 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. Let $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$, and let $P_3 = P_1 + P_2 = (x_3, y_3)$. The addition law on points of the twisted Edwards curve $E_{a,d}$ is given by the following formulas

$$(x_3, y_3) = \left(\frac{x_1y_2 + x_2y_1}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - ax_1x_2}{1 - dx_1x_2y_1y_2} \right).$$

The neutral element is $\mathcal{O} = (0, 1)$, and the negative of P_1 is $-P_1 = (-x_1, y_1)$. The point $\mathcal{O}' = (0, -1)$ has order 2. Two points at infinity Ω_1, Ω_2 are singular and blow up to two points each. Bernstein *et al.* [3] showed that this addition law is *complete*[2] when a is a square and d is not a square.

2.2 Background on Pairings

The key to the definition of pairings is the evaluation of rational functions in divisors (see [21], [12]). Let E be an elliptic curve defined over the prime field $\mathbb{F}_p$, let r be a prime number different from p and $r|\#E(\mathbb{F}_p)$, where $\#E(\mathbb{F}_p)$ denotes the number of points on the elliptic curve E. Let k be the embedding degree of the elliptic curve E with respect to r. By this setting, we can define subgroups of points of prime order r on $E(\mathbb{F}_{p^k})$, denoted by $E[r]$ and a multiplicative group of order r in the extension field $\mathbb{F}_{p^k}^*$, i.e., $\mathbb{F}_{p^k}^*$ contains the group μ_r of r-roots of unity. Let $P, Q \in E[r]$, let D_P, D_Q be degree zero divisors with $D_P \sim (P) - (\mathcal{O})$ and $D_Q \sim (Q) - (\mathcal{O})$, and let f_P, f_Q be functions such that $div(f_P) = rD_P$ and $div(f_Q) = rD_Q$. The Weil pairing $\omega : E[r] \times E[r] \to \mu_r$ is defined as

$$\omega(P, Q) = \frac{f_P(D_Q)}{f_Q(D_P)}$$

The *reduced* Tate pairing $\tau : E(\mathbb{F}_{p^k})[r] \times E(\mathbb{F}_{p^k})/rE(\mathbb{F}_{p^k}) \to \mu_r$ is defined as

$$\tau(P, Q) = f_P(Q)^{\frac{p^k-1}{r}}$$

The Ate pairing is an optimized version of the Tate pairing when restricted to Frobenius eigenspaces. Let $\mathbb{G}_1 = E[r] \cap Ker(\pi_p - [1]) = E(\mathbb{F}_p)[r]$, $\mathbb{G}_2 = E[r] \cap$

2. *Complete* means that the addition formulas work for all pairs of input points. There are no troublesome points at infinity as in Weierstrass curves.

$Ker(\pi_p - [p]) \subseteq E(\mathbb{F}_{p^k})[r]$. For $Q \in \mathbb{G}_2$ and $P \in \mathbb{G}_1$, the Ate pairing is defined in [22] as (the arguments are swapped in comparison to Tate pairing):

$$a_T = \mathbb{G}_2 \times \mathbb{G}_1 \to \mu_r, \quad (Q, P) \mapsto f_{T,Q}(P)^{(p^k-1)/r}$$

The length of Miller loop (see the following section in Ate pairing computation) is determined by the trace of Frobenius t. Thus, the Ate pairing is particularly suitable for pairing-friendly elliptic curves with small values of t. When computing Tate pairing and its variants, instead of taking the point Q on the curve $\mathbb{G}_2 \subseteq E(\mathbb{F}_{p^k})[r]$, one can take $Q' \in \mathbb{G}_2' \subseteq E'(\mathbb{F}_{p^e})[r]$, where E' is a twist of E, $d|k$ is the degree of the twist, and $e = k/d$ as points on the twisted curve are defined over a smaller field, and hence obviously faster in computation.

2.3 Pairing Computation on Edwards Curves

The pairings over (hyper-)elliptic curves are computed using the algorithm proposed by Miller [12]. The main part of Miller's algorithm is to construct the rational function $f_{r,P}$ and evaluating $f_{r,P}(Q)$ with $div(f_{r,P}) = r(P) - (rP) - [r-1](\mathcal{O})$ for divisors P and Q.

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

Lemma 2.1 (Lemma 2, [12]): For n and m two integers, up to a multiplicative constant, we have

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

Equation (2) is called *Miller relation*, which is proved by considering divisors. For Edwards curves, Arene *et al.* [15] defined Miller's function in the following theorem.

Theorem 2.2 (Theorem 2, [15]): 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 be 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} is the horizontal line going through P_3 whose divisor is $div(\ell_{1,P_3}) = (P_3) + (-P_3) - 2(\Omega_2)$, 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

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

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.

Algorithm 1: Miller's Algorithm for twisted Edwards curves [18]

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

$R \leftarrow P$, $f \leftarrow 1$;
for $i = t-1$ **to** 0 **do**
1 $\quad f \leftarrow f^2 \frac{\phi_{R,R}(Q)}{\ell_{1,\mathcal{O}}(Q)\ell_{2,2R}(Q)}$, $R \leftarrow 2R$;
$\quad$ **if** $r_i = 1$ **then**
2 $\quad\quad f \leftarrow f \frac{\phi_{R,P}(Q)}{\ell_{1,\mathcal{O}}(Q)\ell_{2,R+P}(Q)}$, $R \leftarrow R + P$;
$\quad$ **end**
end
return f

3 Our Variant of Miller's Algorithm on Edwards curves

In this section, we first introduce a variant of Miller's function. Then, we describe a variant of Miller's algorithm that is generally more efficient than Algorithm 1 for pairing computation over any Edwards curves (*i.e.*, without twists). Finally, we discuss our variant with denominator elimination for even embedding degree.

3.1 Variant of Miller function

Similar to the method in [23], our algorithm requires a rational function h whose divisor is $(P_1) + (P_2) + (-P_3) - 3(\mathcal{O})$ instead of Miller's function whose divisor is $(P_1) + (P_2) - (P_3) - (\mathcal{O})$. For Weierstrass curves, such a function h is given by the line function passing through P_1 and P_2. On Edwards curves, we define the function h as follows:

Definition 3.1: If $P_1, P_2 \in E_{a,d}(\mathbb{F}_p)$, then define

$$h_{P_1,P_2} = \frac{\phi_{P_1,P_2}}{\phi_{\mathcal{O},\mathcal{O}}},$$

where ϕ is the equation of the conic $\mathcal{C}$ defined as in Theorem 2.2.

Lemma 3.1: Let $P_3 = P_1 + P_2$. The divisor of the function h_{P_1,P_2} is equal to $(P_1) + (P_2) + (-P_3) - 3(\mathcal{O})$.

Proof: By calculating divisors, we have:

$$\begin{aligned} div(\frac{\phi_{P_1,P_2}}{\phi_{\mathcal{O},\mathcal{O}}}) &= div(\phi_{P_1,P_2}) - div(\phi_{\mathcal{O},\mathcal{O}}) \\ &= (P_1) + (P_2) + (-P_3) + (\mathcal{O}') - 2(\Omega_1) \\ &\quad - 2(\Omega_2) - 3(\mathcal{O}) - (\mathcal{O}') + 2(\Omega_1) + 2(\Omega_2) \\ &= (P_1) + (P_2) + (-P_3) - 3(\mathcal{O}), \end{aligned}$$

which concludes the proof. □

In comparison to Eq. (3), our equivalent function $h_{P_1,P_2} = \frac{\phi_{P_1,P_2}}{\phi_{\mathcal{O},\mathcal{O}}}$ consists of only two factors. Furthermore, the factor $\phi_{\mathcal{O},\mathcal{O}}$ whose divisor is $3(\mathcal{O}) + (\mathcal{O}') - 2(\Omega_1) - 2(\Omega_2)$ is fixed during pairing computation. Let $P_1 = (X_1, Y_1, Z_1)$, $P_2 = (X_2, Y_2, Z_2)$ and $Q = (X_Q, Y_Q, Z_Q) \in E(\mathbb{F}_{p^k})$. The factor $\phi_{\mathcal{O},\mathcal{O}}$ can be precomputed and integrated into the factor ϕ_{P_1,P_2} as follows:

$$h_{P_1,P_2}(Q) = \frac{\phi_{P_1,P_2}(Q)}{\phi_{\mathcal{O},\mathcal{O}}(Q)} = \frac{c_{Z^2}(Z_Q^2 + Y_Q Z_Q) + c_{XY} X_Q Y_Q + c_{XZ} X_Q Z_Q}{X_Q(Z_Q - Y_Q)} = c_{Z^2}\eta_1 + c_{XY}\eta_2 + c_{XZ}\eta_3, \quad (4)$$

where $\eta_1 = \frac{Z_Q^2 + Y_Q Z_Q}{X_Q(Z_Q - Y_Q)}$, $\eta_2 = \frac{Y_Q}{Z_Q - Y_Q}$, $\eta_3 = \frac{Z_Q}{Z_Q - Y_Q}$ are fixed for whole computation, thus they can be precomputed and stored. Coefficients c_{Z^2}, c_{XY}, c_{XZ} are defined in [15, Section 4] as follows:

If $P_1 \neq P_2$, then

$$\begin{aligned} c_{Z^2} &= X_1 X_2 (Y_1 Z_2 - Y_2 Z_1), \\ c_{XY} &= Z_1 Z_2 (X_1 Z_2 - X_2 Z_1 + X_1 Y_2 - X_2 Y_1), \\ c_{XZ} &= X_2 Y_2 Z_1^2 - X_1 Y_1 Z_2^2 + Y_1 Y_2 (X_2 Z_1 - X_1 Z_2). \end{aligned} \quad (5)$$

If $P_1 = P_2$, then

$$\begin{aligned} c_{Z^2} &= X_1 Z_1 (Z_1 - Y_1), \\ c_{XY} &= d X_1^2 Y_1 - Z_1^3, \\ c_{XZ} &= Z_1 (Z_1 Y_1 - a X_1^2). \end{aligned} \quad (6)$$

We can see that Eq (4) has no more denominator factor. Assume that we compute pairings on Edwards curves with odd embedding degree, Eq (4) saves two multiplications and one inversion in comparison to the original Miller function (lines 1, 2 in Algorithm 1). Furthermore, Eq (4) becomes simpler when computing Ate pairing. The following lemma shows that the factor $\phi_{\mathcal{O},\mathcal{O}}(P)$ can be ignored without changing the final result.

Lemma 3.2: Let $P \in E(\mathbb{F}_p)[r]$ and $Q \in E(\mathbb{F}_{p^k})[r]$. In computing Ate pairing $a_T(Q,P)$, the factor $\phi_{\mathcal{O},\mathcal{O}}(P)$ can be ignored without changing the value of $a_T(Q,P)$.

Proof: By definition in [15, Theorem 1], the conic $\phi_{\mathcal{O},\mathcal{O}}$ evaluated at P has the form $\phi_{\mathcal{O},\mathcal{O}}(P) = X_P(Z_P - Y_P)$, where $X_P, Y_P, Z_P \in \mathbb{F}_p$ are the abscissas of P. Hence, $\phi_{\mathcal{O},\mathcal{O}}(P) \in \mathbb{F}_p$. This factor will become 1 after being raised to the exponent $(p^k - 1)/r$. □

The main improvement is from the following lemma.

Lemma 3.3: For i and j two integers, up to a multiplicative constant, we have

$$f_{n+m,P} = \frac{1}{f_{-n,P} f_{-m,P} h_{-nP,-mP}}.$$

Proof: This lemma and its proof is very similar to Lemma 2 in [23]. The proof can be achieved by considering divisors. Reader can see [23, Lemma 2] for more details.

□

From Lemma 3.3, we can see that the function $f_{n+m,P}$ is computed from $f_{-n,P}$ and $f_{-m,P}$ instead of $f_{n,P}$ and $f_{m,P}$ on which Miller's algorithm is based. Relation between $f_{n,P}$ and $f_{-n,P}$ as follows:

$$f_{-n,P} = \frac{1}{f_{n,P} h_{nP,-nP}}.$$

However, in order to avoid this expensive operation, algorithm in [23] used an *expansion to the base of* -2 instead of the base of 2. The following section will describe our variant of Miller's algorithm over pairing-friendly Edwards curves.

3.2 Algorithm

Our variant of Miller's algorithm over Edwards curves is described by the pseudo-code in **Algorithm 2**. It was inspired by the idea of applying Lemma 3.3 with $m = n$ or $n \in \{\pm 1\}$. Since our algorithm computes $f_{n+m,P}$ from $f_{-n,P}$ and $f_{-m,P}$, the scalar input will be given by -2-*adic expansion*. Let r be the prime order of subgroup of points on the twisted Edwards curve $E_{a,d}$, and let l_r and h_r be the length and the Hamming weight of r in binary representation. The algorithm updates numerators and denominators separately, so that only one final inversion appears at the end of the algorithm. If the value of $(l_r + h_r)$ is even, the value of f will be initialized to $f_{1,P}$. Otherwise, the value of g will be initialized to $f_{-1,P}$. Note that $f_{1,P} = 1$, and $f_{-1,P} = \frac{1}{f_{1,P} h_{P,-P}} = \frac{\phi_{\mathcal{O},\mathcal{O}}}{\phi_{P,-P}}$ which is fixed and can be precomputed.

We use the notation $h'_{-T,-P}$ for the function $f_{-1,P} h_{-T,-P}$. In many situations, this can be computed faster by computing $f_{-1,P}$ and $h_{-T,-P}$ separately and taking the product. For Edwards curves, we have

$$h'_{-T,-P} = f_{-1,P} h_{-T,-P} = \frac{\phi_{-T,-P}}{\phi_{P,-P}}, \quad (7)$$

where $\phi_{P,-P}$ depends only on fixed arguments P, Q. By Eq. 4, we have:

$$h'_{P_1,P_2}(Q) = c_{Z^2}\gamma_1 + c_{XY}\gamma_2 + c_{XZ}\gamma_3, \quad (8)$$

where $\gamma_1 = \frac{Z_Q^2 + Y_Q Z_Q}{\phi_{P,-P}(Q)}$, $\gamma_2 = \frac{X_Q Y_Q}{\phi_{P,-P}(Q)}$, $\gamma_3 = \frac{X_Q Z_Q}{\phi_{P,-P}(Q)}$, and all these factors are fixed for whole computation, so they can be precomputed and cached.

Remark : Note that in **Algorithm 2**, the value of R is always a *positive* multiple of P. Although this approach does not eliminate denominators, but it improves the computational performance of Miller algorithm when computing *any* pairing on pairing-friendly Edwards curves having *any* small embedding degree (*i.e.*, without twists).

3.3 Edwards curves with even embedding degrees

For twisted Edwards curves having an even embedding degree (i.e., $2|k$), Miller's algorithm can be implemented more efficiently. As pointed out in [15] such curves admit an even twist which eliminates denominators and all irrelevant terms in the subfield of $\mathbb{F}_{p^k}$ in Tate pairing computation. Another advantage of embedding degrees of the form $2^i 3^j$, where $i \geq 1$, $j \geq 0$ is that the corresponding extensions of $\mathbb{F}_p$ can be written as composite extensions of degree 2 or 3, which enables faster basic arithmetic operations [24].

Algorithm 2: Variant of Miller's Algorithm on Edwards curves

Data: $r = \sum_{i=0}^{l_r-1} r_i 2^i$, $r_i \in \{0,1\}$, $r_{l_r-1} = 1$, h_r is the Hamming weight of r
Result: $f_{r,P}(Q)$;
$f \leftarrow 1$, $R \leftarrow P$;
if $l_r + h_r$ *is odd* **then**
  $\delta \leftarrow 1$, $g \leftarrow f_{-1,P}$;
end
else
  $\delta \leftarrow 0$, $g \leftarrow 1$;
end
for $i = l_r - 2$ **to** 0 **do**
  if $\delta = 0$ **then**
1 $f \leftarrow f^2 \cdot h_{R,R}(Q)$, $g \leftarrow g^2$;
    $R \leftarrow 2R$, $\delta \leftarrow 1$;
    if $r_i = 1$ **then**
2 $g \leftarrow g \cdot h'_{-R,-P}(Q)$, $R \leftarrow R + P$, $\delta \leftarrow 0$;
    end
  end
  else
3 $g \leftarrow g^2 \cdot h_{-R,-R}(Q)$, $f \leftarrow f^2$;
    $R \leftarrow 2R$, $\delta \leftarrow 0$;
    if $r_i = 1$ **then**
4 $f \leftarrow f \cdot h_{R,P}(Q)$, $R \leftarrow R + P$, $\delta \leftarrow 0$;
    end
  end
end
5 **return** $\frac{f}{g}$

Similarly as in [23], [17], by using conjugates of elements in $\mathbb{F}_{p^k}$ when k is even, we don't need to update the numerators and denominators separately (two functions f and g). This will save one squaring in full extension field for each bit and the division (line 5 in Algorithm 2).

Let $v = (a + ib)$ be a representation of an element of $\mathbb{F}_{p^k}$, where $a, b \in \mathbb{F}_{q^{k/2}}$, and i is a quadratic non-residue and $\delta = i^2$. The conjugate of v over $\mathbb{F}_{q^{k/2}}$ is given by $\bar{v} = \overline{(a+ib)} = a - ib$. It follows that, if $v \neq 0$, then

$$\frac{1}{v} = \frac{\bar{v}}{a^2 - \delta b^2}$$

where $a^2 - \delta b^2 \in \mathbb{F}_{q^{k/2}}$. Thus, in a situation where elements of $\mathbb{F}_{q^{k/2}}$ can be ignored, $\frac{1}{v}$ can be replaced by $\bar{v}$, thereby saving an inversion in $\mathbb{F}_{p^k}$.

By using this fact, the updating of the function g in lines 2, 3 in Algorithm 2) can be performed as follows:

$$f \leftarrow f \cdot \overline{h'_{-R,-P}(Q)}, \text{ and } f \leftarrow f^2 \cdot \overline{h_{-R,-R}(Q)}$$

where f is the same function in lines 1, 4 and $\overline{h'_{-R,-P}}, \overline{h_{-R,-R}}$ are conjugates of $h'_{-R,-P}$ and $h_{-R,-R}$, respectively.

4 Performance Analysis

In this section, we first compare the proposed algorithm with the original Miller's algorithm over Edwards curves [15], [18], and the Xu-Lin refinements [18]. We also compare our algorithms with the Arene et al.'s algorithm [15] when computing the Tate pairing on even twisted curves. Then, we will give a performance analysis for Ate pairing computation over different choices of Edwards curves at 128-bit security level.

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 $\mathbf{add}_{p^m}$ be the costs for inversion, multiplication, squaring, and addition in the field $\mathbb{F}_{p^m}$ respectively. Denote $\mathbf{m}_a$ be 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 [15] showed that the total cost of updating the point R and coefficients c_{Z^2}, c_{XY}, and c_{ZZ} (Eqs. 5-6) 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 [15, §5] for more details), where $e = k/d$ as denoted in § 2.2. Without special treatment, this cost is the same for all algorithms.

4.1 Updating Miller function

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 $\mathbb{F}_{p^k}$.

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.

TABLE 1
Comparison of Algorithm 2 with the previous algorithms

	Doubling	Addition
Algorithm 1 [15], [18]	$2\mathbf{S}_{p^k} + 3\mathbf{M}_{p^k}$	$2\mathbf{M}_{p^k}$
Algorithm in [15]	$1\mathbf{S}_{p^k} + 1\mathbf{M}_{p^k}$	$1\mathbf{M}_{p^k}$
Algorithm 2	$2\mathbf{S}_{p^k} + 1\mathbf{M}_{p^k}$	$1\mathbf{M}_{p^k}$
Modified Algorithm (§ 3.3)	$1\mathbf{S}_{p^k} + 1\mathbf{M}_{p^k}$	$1\mathbf{M}_{p^k}$

From TABLE 1, it can be seen that Algorithm 2 is *generally* faster than the general results in [15] (Algorithm 1). In comparison to Algorithm 1, the proposed algorithm saves 2 multiplications in the full extension field in doubling steps and one multiplication in the full extension field in addition steps when updating the Miller function.

In comparison to Arene *et al.*'s algorithm [15], Algorithm 2 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. In the same setting of curves, our modification (Section 3.3) needs no extra effort to update f than the Arene *et al.*'s algorithm.

The refinements in [18] 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.

TABLE 2
Comparison of our algorithm with the refinements in [18].

	Algorithm in [18]	**Algorithm 2**
$q=0$	$5\mathbf{S}_{p^k} + 3\mathbf{M}_{p^k}$	$4\mathbf{S}_{p^k} + 2\mathbf{M}_{p^k}$
$q=1$	$4\mathbf{S}_{p^k} + 7\mathbf{M}_{p^k}$	$4\mathbf{S}_{p^k} + 3\mathbf{M}_{p^k}$
$q=2$	$4\mathbf{S}_{p^k} + 7\mathbf{M}_{p^k}$	$4\mathbf{S}_{p^k} + 3\mathbf{M}_{p^k}$
$q=3$	$4\mathbf{S}_{p^k} + 10\mathbf{M}_{p^k}$	$4\mathbf{S}_{p^k} + 4\mathbf{M}_{p^k}$

From TABLE 2, it clearly see that **Algorithm 2** is generally faster than the refinements of Miller's algorithm in [18].

4.2 Analysis at the 128-bit Security Level

In this subsection, we give an analysis about the efficiency of Miller algorithm for Ate pairing computation over three families of pairing-friendly Edwards curves with embedding degrees $k = 8, 9, 10$ at 128-bit security level. The constructions of these families of curves were presented in [25], [26]. Recall that the length of Miller loop equals to $\log(r)/\varphi(k)$ for Ate pairing computation. Then, the respective lengths of Miller loop are 64, 43, and 64 for curves with $k = 8, 9$, and 10.

For a pairing-based cryptosystem to be secured, the discrete logarithm problems in the largest subgroup of points on $E/\mathbb{F}_p$ and in the multiplicative group $\mathbb{F}^{\times}_{p^k}$ must both be computational infeasible. At 128-bit security level, the subgroup size r must be equal to 256 and $p^k \geq 3072$ (both in bits, see [27]). TABLE 3 shows sizes in bits of r, p and p^k corresponding different k.

TABLE 3
Security Matching for the 128-bits security level

k	r (in bits)	p (in bits)	p^k (in bits)
8	256	384	3072
9	256	341	3072
10	256	384	3840

In this analysis, we apply a twist of degree 4, 3, and 2 for curves with $k = 8, 9$, and 10, respectively. By carefully choosing parameters, one can get a value of T such that its Hamming weight is very low, where $|T| = \log(r)/\varphi(k)$ as denoted in Section 2.2. Thus, one can only focus on the cost of doubling steps. Let C denote the cost of Miller loop in Ate pairing computation. Using Algorithm 2, Arene et al.'s algorithm, and analysis in [15, §6], we have

$$C_{k=8} = 64(1\mathbf{S}_{p_1^8} + 1\mathbf{M}_{p_1^8} + 6\mathbf{M}_{p_1^2} + 5\mathbf{S}_{p_1^2} + \frac{k}{2}\mathbf{M}_{p_1} + 2\mathbf{m}_a),$$
$$C_{k=9} = 43(2\mathbf{S}_{p_2^9} + 1\mathbf{M}_{p_2^9} + 6\mathbf{M}_{p_2^3} + 5\mathbf{S}_{p_2^3} + k\mathbf{M}_{p_2} + 2\mathbf{m}_a),$$
$$C_{k=10} = 64(1\mathbf{S}_{p_3^{10}} + 1\mathbf{M}_{p_3^{10}} + 6\mathbf{M}_{p_3^5} + 5\mathbf{S}_{p_3^5} + k\mathbf{M}_{p_3} + 2\mathbf{m}_a),$$

where size in bits of p_i, $i = 1, 2, 3$ corresponding with curves having $k = 8, 9, 10$ are described as in TABLE 3. Using Toom-Cook and Karatsuba algorithms, we assume $\mathbf{M}_{p_i^{2m}} \approx 3\mathbf{M}_{p_i^m}$, $\mathbf{S}_{p_i^{2m}} \approx 3\mathbf{S}_{p_i^m}$, and $\mathbf{M}_{p_i^{3m}} \approx 5\mathbf{M}_{p_i^m}$, $\mathbf{S}_{p_i^{3m}} \approx 5\mathbf{S}_{p_i^m}$, for $i = 1, 2, 3$, and $m \geq 1$. For field operations in $\mathbb{F}_{p_3^5}$, Montgomery [28] presented an efficient formulas, for which $\mathbf{M}_{p_3^5} \approx 13\mathbf{M}_{p_3}$, $\mathbf{S}_{p_3^5} \approx 13\mathbf{S}_{p_3}$. The following table shows the theoretical analysis on Ate pairing computation over the above curves at 128-bit security level.

TABLE 4
Comparison of operation counts for different curves at the 128-bit security level

k	p (in bits)	Number of operations
8	384	$3136\mathbf{M}_{p_1} + 2688\mathbf{S}_{p_1} + 128\mathbf{m}_a$
9	341	$2752\mathbf{M}_{p_2} + 3225\mathbf{S}_{p_2} + 86\mathbf{m}_a$
10	384	$8128\mathbf{M}_{p_3} + 6656\mathbf{S}_{p_3} + 128\mathbf{m}_a$

From TABLE 4, it is easy to see that pairing-friendly Edwards curves with $k = 9$ offer a better performance than that with even $k = 10$. There is no big difference on the number of operations between curves with $k = 8$ and $k = 9$. But, it is worth to note that the size of the base field $\mathbb{F}_{p_2}$ for curves with $k = 9$ is smaller than that of the field $\mathbb{F}_{p_1}$ for curves with $k = 8$ (see TABLE 3).

At the 128-bit security level, Barreto-Naehrig (BN for short) curves [29] achieved the most efficient implementations. There were many benchmarks reported in papers [30], [31], [32], [33]. So far the fastest software implementation presented in [33] allows us to compute a pairing under 2 million cycles on 64-bit computing platforms. Although Table 4 and Table 2 in [33] show that the number of operations in Miller loop for curves of embedding degrees $k = 8, 9$ is fewer than that for BN curves, with many optimizations in both Miller loop [29], [19], [34] and the final exponentiation [35], [36], [37], BN curves are still suited for implementing a single pairing at the 128-bit security level.

However, a BN curve cannot transform to an Edwards curve whose order is a multiple of 4. Thus, public-key cryptosystems implemented on BN curves don't benefit the fast group law operation as on Edwards curves. In some pairing-based cryptosystems, they might

require a number of scalar multiplications which is time-consuming operation and this can be advantageous to use Edwards. Furthermore, when computing several pairings in parallel with only one final exponentiation, such curves with shorter Miller loop (e.g., $k = 9$) may be a good choice (see more discussion in [38]).

5 Examples of Pairing-Friendly Edwards Curves

Edwards curves have a cofactor 4. Generation of pairing-friendly Edwards curves are discussed in [15, §7]. In this section, we present some examples of pairing-friendly Edwards curves. Let $\rho = \log(p)/\log(r)$, where p is the size of the finite field $\mathbb{F}_p$, r is the size of subgroup of points; D denote CM discriminant; the number of points $\#E(\mathbb{F}_p) = 4hr$. The value of $\log(r)/\varphi(k)$ implies the number of iterations that Miller algorithm needs for optimal pairing computation.

5.1 At the 112 bits security level

$k = 7, \rho = 1.33$ following Construction 6.20 in [26]: $D = 3$, $\log(p) = 309$, $\log(r) = 223$, $\log(r)/\varphi(k) = 39$.

$$\begin{aligned} r =& 11792486460390409119540171794482663984948784753 \\ & 601049200190136218459 \\ p =& 57360932776319280207874727702805176779012756816 \\ & 024141289570110872838431452628194715137685826 9 \\ h =& 7^7 \cdot 31^2 \cdot 127 \cdot 277^2 \cdot 39709^2 \end{aligned}$$

$k = 8, \rho = 1.50$ following Example 6.10 [26]: $D = 1$, $\log(p) = 343$, $\log(r) = 224$, $\log(r)/\varphi(k) = 56$.

$$\begin{aligned} r =& 17422989430463274386677569399619042078142 88 \\ & 17396985867102205828986 97 \\ p =& 10813434369352814954576413407087650262884981 \\ & 25793222946509169874343448157047865842177638 \\ & 7235773533870249 \\ h =& 4723 \cdot 5077^2 \end{aligned}$$

5.2 At the 128 bits security level

$k = 8, \rho = 1.50$ following Example 6.10 in [26]: $D = 1$, $\log(p) = 401$, $\log(r) = 258$, $\log(r)/\varphi(k) = 65$.

$$\begin{aligned} r =& 39582340297147856121327521222320014067371613303 \\ & 48360433238015221101525969267 37 \\ p =& 32681600019537443452669476460183149414570521319 \\ & 67115728952206651872840362786176908634547273737 \\ & 38090269200001361478677586 9 \\ h =& 37 \cdot 337 \cdot 1453 \cdot 113931708944716295372312953567053661 \end{aligned}$$

$k = 9, \rho = 4/3$ following [25]: $D = 3$, $\log(p) = 340$, $\log(r) = 254$, $\log(r)/\varphi(k) = 43$.

$$\begin{aligned} r =& 17585923602443760494233455400229627974592727 \\ & 364023471411932687465045674845344 17 \\ p =& 11958793459230290820953097887678427630245690 \\ & 91004011123144944964165877854119429728328541 \\ & 600674561175289 \\ h =& 2^4 \cdot 3^2 \cdot 11^2 \cdot 179^2 \cdot 139602797^2 \end{aligned}$$

$k = 10, \rho = 1.50$ following Example 6.5 in [26]: $D = 1$, $\log(p) = 395$, $\log(r) = 257$, $\log(r)/\varphi(k) = 65$.

$$\begin{aligned} r =& 16409247407405131786536680753483726935465346 5 \\ & 21754526500010380741412334216572 1 \\ p =& 40466054822298248247061490544673939436019447 1 \\ & 81730438452050710645480451559089995870532033 9 \\ & 60965948006332423977863263969 \\ h =& 181 \cdot 5023^4 \cdot 855269^4 \end{aligned}$$

6 Conclusion

In this paper, we proposed a variant of Miller's algorithm on Edwards curves. The proposed algorithm improves the computational performance of all pairings on *generic* pairing-based Edwards curves. Our analysis showed that the new algorithm is faster than previous methods for curves with odd embedding degree and as fast as those curves with even embedding degree.

Our algorithm is particularly interest to compute the Ate pairings on Edwards curves having small embedding degrees k, in the cases where denominators elimination technique is not possible (for example on Edwards curves with odd embedding degrees). We believe that there will be applications in pairing-based cryptography using elliptic curves with embedding degree not of the form $2^i 3^j$. Further work is needed to clarify such a question.

References

[1] H. M. Edwards, "A Normal Form for Elliptic Curves," *Bulletin of the American Mathematical Society*, vol. 44, no. 3, pp. 393–422, Jul. 2007.

[2] D. J. Bernstein and T. Lange, "Faster addition and doubling on elliptic curves," in *Proceedings of the Advances in Crypotology 13th international conference on Theory and application of cryptology and information security*, ser. ASIACRYPT'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 29–50.

[3] D. J. Bernstein, P. Birkner, M. Joye, T. Lange, and C. Peters, "Twisted Edwards curves," in *Proceedings of the Cryptology in Africa 1st international conference on Progress in cryptology*, ser. AFRICACRYPT'08. Springer Berlin/Heidelberg, 2008, pp. 389–405.

[4] A. Menezes, S. Vanstone, and T. Okamoto, "Reducing elliptic curve logarithms to logarithms in a finite field," in *STOC '91: Proceedings of the twenty-third annual ACM symposium on Theory of computing*. New York, NY, USA: ACM, 1991, pp. 80–89.

[5] G. Frey and H.-G. Rück, "A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves," *Math. Comput.*, vol. 62, no. 206, pp. 865–874, 1994.

[6] A. Joux, "A One Round Protocol for Tripartite Diffie-Hellman," in *ANTS-IV: Proceedings of the 4th International Symposium on Algorithmic Number Theory*. Springer-Verlag, 2000, pp. 385–394.

[7] D. Boneh and M. K. Franklin, "Identity-Based Encryption from the Weil Pairing," in *CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*. Springer-Verlag, 2001, pp. 213–229.

[8] D. Boneh, B. Lynn, and H. 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*. Springer-Verlag, 2001, pp. 514–532.

[9] D. Boneh and X. Boyen, "Secure identity based encryption without random oracles," in *CRYPTO*, ser. Lecture Notes in Computer Science, M. K. Franklin, Ed., vol. 3152. Springer, 2004, pp. 443–459.

[10] B. Waters, "Efficient identity-based encryption without random oracles," in *EUROCRYPT '05*, ser. Lecture Notes in Computer Science, R. Cramer, Ed., vol. 3494. Springer, 2005, pp. 114–127.

[11] D.-P. Le, A. Bonnecaze, and A. Gabillon, "Multisignatures as Secure as the Diffie-Hellman Problem in the Plain Public-Key Model," in *Proceedings of the 3rd International Conference Palo Alto on Pairing-Based Cryptography*, ser. Pairing '09. Springer Berlin/Heidelberg, 2009, pp. 35–51.

[12] V. S. Miller, "The Weil Pairing, and Its Efficient Calculation," *Journal of Cryptology*, vol. 17, no. 4, pp. 235–261, 2004.

[13] 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.

[14] 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.

[15] C. Arène, T. Lange, M. Naehrig, and C. Ritzenthaler, "Faster computation of the Tate pairing," *Journal of Number Theory*, vol. 131, no. 5, pp. 842–857, 2011.

[16] I. F. Blake, V. K. Murty, and G. Xu, "Refinements of Miller's algorithm for computing the Weil/Tate pairing," *J. Algorithms*, vol. 58, no. 2, pp. 134–149, 2006.

[17] D.-P. Le and C.-L. Liu, "Refinements of miller's algorithm over weierstrass curves revisited," *Comput. J.*, vol. 54, no. 10, pp. 1582–1591, Oct. 2011.

[18] L. Xu and D. Lin, "Refinement of Miller's Algorithm Over Edwards Curves," in *CT-RSA*, ser. Lecture Notes in Computer Science, J. Pieprzyk, Ed., vol. 5985. Springer, 2010, pp. 106–118.

[19] F. Vercauteren, "Optimal pairings," *IEEE Transactions on Information Theory*, vol. 56, no. 1, p. 7, 2010.

[20] F. Hess, "Pairing lattices," in *Proceedings of the 2nd international conference on Pairing-Based Cryptography*, ser. Pairing '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 18–38.

[21] N. Koblitz, *Algebraic aspects of cryptography*. New York, NY, USA: Springer-Verlag New York, Inc., 1998.

[22] F. Hess, N. P. Smart, and F. Vercauteren, "The eta pairing revisited," *IEEE Transactions on Information Theory*, vol. 52, pp. 4595–4602, 2006.

[23] J. Boxall, N. E. Mrabet, F. Laguillaumie, and D.-P. Le, "A Variant of Miller's Formula and Algorithm," in *Pairing*, 2010, pp. 417–434.

[24] N. Koblitz and A. Menezes, "Pairing-based cryptography at high security levels," in *Proceedings of Cryptography and Coding 2005, volume 3796 of LNCS*. Springer-Verlag, 2005, pp. 13–36.

[25] X. Lin, C.-A. Zhao, F. Zhang, and Y. Wang, "Computing the ate pairing on elliptic curves with embedding degree k = 9," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E91-A, no. 9, pp. 2387–2393, 2008.

[26] D. Freeman, M. Scott, and E. Teske, "A Taxonomy of Pairing-Friendly Elliptic Curves," *J. Cryptol.*, vol. 23, pp. 224–280, April 2010.

[27] A. K. Lenstra, "Unbelievable security. matching aes security using public key systems," in *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, ser. ASIACRYPT '01. London, UK, UK: Springer-Verlag, 2001, pp. 67–86.

[28] P. L. Montgomery, "Five, six, and seven-term karatsuba-like formulae," *IEEE Trans. Comput.*, vol. 54, no. 3, pp. 362–369, Mar. 2005.

[29] P. S. L. M. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime order," in *Proceedings of SAC 2005, volume 3897 of LNCS*. Springer-Verlag, 2005, pp. 319–331.

[30] J.-L. Beuchat, J. E. González-Díaz, S. Mitsunari, E. Okamoto, F. Rodríguez-Henríquez, and T. Teruya, "High-speed software implementation of the optimal ate pairing over barreto-naehrig curves," in *Pairing*, ser. Lecture Notes in Computer Science, M. Joye, A. Miyaji, and A. Otsuka, Eds., vol. 6487. Springer, 2010, pp. 21–39.

[31] G. C. C. F. Pereira, J. M. A. Simplício, M. Naehrig, and P. S. L. M. Barreto, "A family of implementation-friendly bn elliptic curves," *J. Syst. Softw.*, vol. 84, pp. 1319–1326, August 2011.

[32] M. Naehrig, R. Niederhagen, and P. Schwabe, "New software speed records for cryptographic pairings," in *Proceedings of the First international conference on Progress in cryptology: cryptology and information security in Latin America*, ser. LATINCRYPT'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 109–123.

[33] D. Aranha, K. Karabina, P. Longa, C. Gebotys, and J. López, "Faster explicit formulas for computing pairings over ordinary curves," in *Advances in Cryptology – EUROCRYPT 2011*, ser. Lecture Notes in Computer Science, K. Paterson, Ed. Springer Berlin / Heidelberg, 2011, vol. 6632, pp. 48–68.

[34] C. Costello, T. Lange, and M. Naehrig, "Faster Pairing Computations on Curves with High-Degree Twists," in *Public Key Cryptography – PKC 2010*, ser. Lecture Notes in Computer Science, P. Nguyen and D. Pointcheval, Eds. Springer Berlin / Heidelberg, 2010, vol. 6056, pp. 224–242.

[35] M. Scott, N. Benger, M. Charlemagne, L. J. D. Perez, and E. J. Kachisa, "On the final exponentiation for calculating pairings on ordinary elliptic curves," in *Pairing*, ser. Lecture Notes in Computer Science, H. Shacham and B. Waters, Eds., vol. 5671. Springer, 2009, pp. 78–88.

[36] R. Granger and M. Scott, "Faster squaring in the cyclotomic subgroup of sixth degree extensions," in *Public Key Cryptography*, ser. Lecture Notes in Computer Science, P. Q. Nguyen and D. Pointcheval, Eds., vol. 6056. Springer, 2010, pp. 209–223.

[37] K. Karabina, "Squaring in cyclotomic subgroups," *Math. Comput.*, vol. 82, no. 281, 2013.

[38] D.-P. Le and C. H. Tan, "Speeding up ate pairing computation in affine coordinates," in *ICISC*, 2012, pp. 262–277.