Further Refinements of Miller Algorithm on Edwards curves

Duc-Phong Le
Temasek Laboratories. National University of Singapore.

Abstract
Recently, Edwards curves have received a lot of attention in the cryptographic community due to their fastest curve scalar multiplication. In this paper, we study further refinements to Miller algorithm used to compute pairings 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. We analyze and show that our algorithm is faster than the original Miller algorithm and the recent refinements of Xu and Lin (2010).

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

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

Pairing computation on Edwards curves is slightly slower than on Weierstrass curves. However, in some pairing-based cryptosystems, scalar multiplication is the most time-consuming operation and it can be advantageous to use Edwards in such a scenario. Recall that pairing-based cryptography has received a lot of attention over the past more than ten years. Pairings were first introduced into cryptography in Joux’ seminal paper describing a tripartite (bilinear) Diffie-Hellman key exchange [10]. Since then, the use of cryptosystems based on bilinear maps has had a huge success with some notable breakthroughs such as the first identity-based encryption scheme [6], the short signature scheme [7] and many other new cryptographic primitives [5,16,12].

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 [14] 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, making it 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 that over Weierstrass ones. The following question is important to 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 \( \text{div}(g) = (P_1) + (P_2) - (P_3) - (\mathcal{O}) \), where \( \mathcal{O} \) is a distinguished rational point. For curves of Weierstrass form, this function is easy to obtain due to the chord-and-tangent rule for addition. 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.

Arene et al. [1] 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 \( C \) of degree 2. They also introduced explicit formulas with a focus on curves having an even embedding degree.

To speed up the pairing computation on generic Edwards curves, Xu and Lin [17] proposed refinements to Miller’s algorithm. Their refinements are inspired from Blake-Murty-Xu’s work on Weierstrass curves [4]. Though this approach did not bring a dramatic efficiency as that of Barreto et al. for Tate pairing computation, but it can be applied for computing both Weil and Tate pairings on any pairing-friendly Edwards curve. This approach is of particular interest to compute optimal pairings [15, 8], and in situations where the denominator elimination technique using a twist is not possible (e.g., Edwards curves with odd embedding degree). Note that by definition optimal pairings only require about \( \log_2(r)/\phi(k) \) iterations of the basic loop, where \( r \) is the group order, \( \phi \) is Euler’s totient function, and \( k \) is the embedding degree. For example, when \( k \) is prime, then \( \phi(k) = k - 1 \). If we choose a curve having embedding degree \( k \pm 1 \), then \( \phi(k \pm 1) \leq \frac{k+1}{2} \) which is roughly \( \frac{e(k)}{2} = \frac{k-1}{2} \), 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 continue their work and suggest a generalized algorithm. Our algorithm is an extension of refinements in [13] to Edwards curves. We analyze and show that our new algorithm is generally faster than the original Miller’s algorithm on Edwards curves and its refinements [17].

The paper is organized as follows. In Section 2 we briefly recall some background on Edwards curves, pairings, the Miller’s algorithm, and the Xu-Lin’s refinement. In Section 3 we explain our refinement of Miller algorithm. Section 4 gives some discussion on performance of our algorithm. Section 5 is our conclusion.

---

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 \( r | p^k - 1 \). 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 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:

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

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. 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

$$(x_3, y_3) = \left( \frac{x_1 y_2 + x_2 y_1}{1 + dx_1 x_2 y_1 y_2}, \frac{y_1 y_2 - ax_1 x_2}{1 - dx_1 x_2 y_1 y_2} \right).$$

The neutral element is $O = (0, 1)$, and the negative of $P_1$ is $-P_1 = (-x_1, y_1)$. The point $O' = (0, -1)$ has order 2. Two points at infinity $\Omega_1, \Omega_2$ are singular and blow up to two points each. Bernstein et al. [2] showed that this addition law is complete 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 [11, 14]). 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 $\#E(\mathbb{F}_p)$ is denoted 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 $\mu_r$ of $r$-roots of unity. Let $P, Q \in E[r]$, let $D_P, D_Q$ be degree zero divisors with $D_P \sim (P) - (O)$ and $D_Q \sim (Q) - (O)$, and let $f_P, f_Q$ be functions such that $\text{div}(f_P) = rD_P$ and $\text{div}(f_Q) = rD_Q$. The Weil pairing $\omega : E[r] \times E[r] \rightarrow \mu_r$ is defined as

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

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

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

The Ate pairing is an optimized version of the Tate pairing when restricted to Frobenius eigenspaces. Let $G_1 = E[r] \cap \text{Ker}(\pi_p - [1]) = E(\mathbb{F}_p)[r]$, $G_2 = \text{Complete}$ means that the addition formulas work for all pairs of input points. There are no troublesome points at infinity as in Weierstrass curves.
Lemma 2.1 \([\text{Lemma 2, [14]}]\)

Miller function \(f\) is a constant, we have:

\[
f_r(P) = (Q, P) \mapsto f_r(Q) = (P^s - 1)/r
\]

The length of Miller loop (see the following section in Ate pairing computation) is determined by the trace of Frobenius \(t\). The Ate pairing is thus particularly suitable for pairing-friendly elliptic curves with small values of \(t\).

For Edwards curves, Arene et al. [1] defined Miller’s function in the following theorem.

\[
\text{Let } m \text{ and } n \text{ be two integers, and } g_{m,n,P} \text{ be a rational function whose divisor } \text{div}(g_{m,n,P}) = (mP) + (nP) - (m + n)[P] - (\mathcal{O}). \text{ We call the function } g_{m,n,P} \text{ Miller function.} \]

Miller’s algorithm is based on the following lemma.

Lemma 2.1 (Lemma 2, [14]). For \(n\) and \(m\) two integers, up to a multiplicative constant, we have

\[
f_{m+n,P} = f_{m,P}f_{n,P}g_{m,n,P}
\]

Equation (2) is called Miller relation, which is proved by considering divisors.

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 \(\phi\) is the equation of the conic \(C\) passing through \(P_1, P_2, -P_3, \Omega_1, \Omega_2, \Omega'\) whose divisor is \((P_1) + (P_2) + (-P_3) + (\Omega') - 2(\Omega_1) - 2(\Omega_2)\). Let \(\ell_{1,P_2}\) be the horizontal line going through \(P_3\) whose divisor is \(\text{div}(\ell_{1,P_2}) = (P_3) + (-P_3) - 2(\Omega_2)\), and \(\ell_{2,\mathcal{O}}\) is the vertical line going through \(\mathcal{O}\) and \(\Omega'\) whose divisor is \((\mathcal{O}) + (\Omega') - 2(\Omega_1)\).

Then we have

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

The rational function \(g_{P_1,P_2} = \frac{\phi_{P_1,P_2}}{\ell_{1,P_2} \ell_{2,\mathcal{O}}}\) 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].

2.3 Pairing Computation on Edwards Curves

The pairings over (hyper-)elliptic curves are computed using the algorithm proposed by Miller [14]. 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 \(Q\).

Let \(m\) and \(n\) be two integers, and \(g_{m,n,P}\) be a rational function whose divisor \(\text{div}(g_{m,n,P}) = (mP) + (nP) - (m + n)[P] - (\mathcal{O}).\)

We call the function \(g_{m,n,P}\) Miller function. Miller’s algorithm is based on the following lemma.

Lemma 2.1 (Lemma 2, [14]). For \(n\) and \(m\) two integers, up to a multiplicative constant, we have

\[
f_{m+n,P} = f_{m,P}f_{n,P}g_{m,n,P}.
\]
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
  \( f \leftarrow f^2 \cdot \phi_{R,R}(Q) \)
  \( g \leftarrow g^2 \cdot \ell_{1,\mathcal{O}}(Q) \cdot \ell_{2,R}(Q) \)
  \( R \leftarrow 2R \)
if \( r_i = 1 \) then
  \( f \leftarrow f \cdot \phi_{R,P}(Q) \)
  \( g \leftarrow g \cdot \ell_{1,\mathcal{O}}(Q) \cdot \ell_{2,R+P}(Q) \)
  \( R \leftarrow R + P \)
return \( f/g \)

Algorithm 1: Miller’s Algorithm for twisted Edwards curves [17]

2.4 Xu-Lin Refinements

By extending the Blake et al.’s method [4] to Edwards curves, Xu and Lin [17] presented a refinement to Miller algorithm. Their algorithm was achieved owing to the following theorem.

Theorem 2.3 (Theorem 1 in [17]). Let \( E_{a,d} \) be a twisted Edwards curve over \( \mathbb{F}_p \) and \( Q \in E_{a,d} \) be a point of order \( r \). Then

1. \[
\left( \frac{\phi_{Q,Q}}{\ell_{1,\mathcal{O}} \ell_{2,\mathcal{O}}} \right)^2 \frac{\phi_{2|Q|2|Q}}{\ell_{2,|Q|} \ell_{1,\mathcal{O}}} = \frac{\phi_{Q,Q}^2}{\phi_{-2|Q|,-2|Q|} \phi_{\mathcal{O},\mathcal{O}}}.
\]

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

The above theorem was proven by calculating divisors (see [17] for more details). From this theorem, they introduced refinements and an improved Miller algorithm in radix-4 representation [17, Algorithm 3].

3 Our Improvements on Miller’s Algorithm

3.1 Refinement

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 \( P, Q \in E_{a,d} \). Let \( \phi_{P,Q} \) be a conic passing through \( P \) and \( Q \), \( \phi_{-P,-Q} \) be a conic passing through \(-P\) and \(-Q\), and let \( \phi_{P+Q,-[P+Q]} \) a conic passing through \( P + Q \) and \(- (P + Q)\). Then we define
\[ g_{P,Q} = \frac{\phi_{P,Q}}{\phi_{P+Q} - (P+Q)}. \] (4)

**Lemma 3.1.** We have \[ \text{div}(g_{P,Q}) = (P) + (Q) - ([P + Q]) - \mathcal{O}. \]

**Proof.** By calculating divisors, it is straightforward to see that
\[
\text{div}(g_{P,Q}) = (P) + (Q) + ([P + Q]) + (\mathcal{O}') - 2(\Omega_1) - 2(\Omega_2)
- ([P + Q]) - ([P + Q]) - (\mathcal{O}) - (\mathcal{O}') + 2(\Omega_1) + 2(\Omega_2)
= (P) + (Q) - ([P + Q]) - (\mathcal{O}),
\]
which concludes the proof. \(\square\)

Recall that, on Weierstrass curves, the Miller function \(g_{P,Q} = \ell_{P,Q} \upsilon_{P+Q}\), where \(\ell_{P,Q}, \upsilon_{P+Q}\) are lines passing through \(P, Q\) and \(P + Q, -(P + Q)\), respectively. Thus, one can see that this function looks similar to the Miller function on Weierstrass curves if we replace the conics by line functions. A variant of Miller algorithm by using Eq. 4 is described in Algorithm 2.

**Algorithm 2:** Miller’s Algorithm for twisted Edwards curves

```
Input: \( r = \sum_{i=0}^{t} r_i 2^i \) with \( r_i \in \{0, 1\} \), \( P, Q \in \mathcal{E}[r] \);
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
        \( f \leftarrow f \cdot \phi_{R,P}(Q) \)
        \( g \leftarrow g \cdot \phi_{R+P,-(R+P)}(Q) \)
        \( R \leftarrow R + P \)
return \( f/g \)
```

**Lemma 3.2.** We have
\[ g_{P,Q} = \frac{\phi_{P,-P} \cdot \phi_{Q,-Q}}{\phi_{-P,-Q} \cdot \phi_{\mathcal{O},\mathcal{O}}} \] (5)
Proof. This lemma is again proved by considering divisors. Indeed,

\[ \text{div} \left( \frac{\phi_{P - R}}{-R} \cdot \frac{-Q}{\phi_{O, O}} \right) = (P) + (-P) + (O) + (O') - 2(\Omega_1) - 2(\Omega_2) + (Q) + (-Q) + (O) + (O') - 2(\Omega_1) - 2(\Omega_2) - (P) - (-Q) - ([P + Q]) - (O') + 2(\Omega_1) + 2(\Omega_2) - 3(O) - (O') + 2(\Omega_1) + 2(\Omega_2) = (P) + (Q) - ([P + Q]) - (O) = \text{div}(g_{P, Q}), \]

which concludes the proof. \(\square\)

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

\[ \frac{\phi_{R, R}}{\phi_{R - R} \cdot \phi_{2R - 2R}} = 1 \]

This lemma is easy to be proven using Lemma 3.2.

3.2 Algorithm

Our algorithm is described by the pseudo-code in Algorithm 3. It was inspired by idea of applying Lemma 3.3. Similar to Algorithm 3 in [13], our algorithm make uses a memory variable \( m \) to note that whether there is still a conic function \( \phi_{R - R} \) delayed in the current step or not. At each step, we will apply Eq. (6) if \( m = 1 \).

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 each step of the algorithm).

Remark: When computing Ate pairing, the value of the factor \( \phi_{O, O}(P) = X_P(Z_P - Y_P) \) that is in a proper subfield of \( \mathbb{F}_{p^k} \). After being raised to the exponent \((p^k - 1)/r\), this factor will become 1, hence one can ignore it without changing the final result.

4 Discussion

In this section, we first compare the proposed algorithm with the original Miller’s algorithm over Edwards curves [11, 17], and the Xu-Lin refinements [17].

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 \( I_{p^m}, M_{p^m}, S_{p^m}, \) and \( \text{add}_{p^m} \) the costs for inversion, multiplication, squaring, and addition in the field \( \mathbb{F}_{p^m} \), respectively. We denote by \( m_a \) 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 §[2.3] showed that the total cost of updating the point $R$ and coefficients $c_{XZ}, c_{XY}$, and $c_{ZZ}$ of the conic is $6M_{p^e} + 5S_{p^e} + 2m_a$ for each doubling step and $14M_{p^e} + 1m_a$ for each addition step (see §[2.3] for more details), where $c = k/d$ as denoted in §[2.3]. 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 $S_{p^k} = 0.8M_{p^k}$, a commonly used value in the literature.

It is clear to see that to update functions $f$ and $g$, the proposed algorithm requires $1M_{p^k} + 2S_{p^k}$ for a doubling step (lines 1, 3), and $1M_{p^k}$ 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.
Doubling and Addition

Algorithm 1

\[ 2S_{p^k} + 3M_{p^k} = 4.6M_{p^k} \]

Algorithm in [1]

\[ 1S_{p^k} + 1M_{p^k} = 1.8M_{p^k} \]

Algorithm 2

\[ 2S_{p^k} + 2M_{p^k} = 3.6M_{p^k} \]

Algorithm 3

\[ 2S_{p^k} + 1M_{p^k} \]

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 \( b_i = 1 \).

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 [17] are described in radix 4. Their algorithm allows to eliminate some rational functions from Eq (3) during pairing computation. Let \( r = \sum_{i=0}^{t-1} q_i 4^i \), with \( q_i \in \{0, 1, 2, 3\} \). Table 2 compares our algorithm and their algorithm.

| Algorithm in [17] | Algorithm 3 |
|-------------------|-------------|
| \( q = 0 \)       | 5S_{p^k} + 3M_{p^k} | 4S_{p^k} + 2M_{p^k} |
| \( q = 1 \)       | 4S_{p^k} + 7M_{p^k} | 4S_{p^k} + 3M_{p^k} (line 3) |
| \( q = 2 \)       | 4S_{p^k} + 7M_{p^k} | 4S_{p^k} + 3M_{p^k} (line 3) |
| \( q = 3 \)       | 4S_{p^k} + 10M_{p^k} | 4S_{p^k} + 4M_{p^k} (line 3) |

Table 2: Comparison of our algorithm with the refinements in [17].

From Table 2 it clearly see that Algorithm 3 is generally faster than the refinements of Miller’s algorithm in [17] for all four cases.
5 Conclusion

In this paper, we extended the Le-Liu’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 [17]. Our improvement works perfectly well for computing both of Weil and Tate pairings over any pairing-friendly Edwards curve. It allows the use of Edwards curve with embedding degree not of the form \(2^i3^j\), and is suitable for the computation of optimal pairings [15].

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] 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 cryptography and information security, ASIACRYPT’07, pages 29–50, Berlin, Heidelberg, 2007. Springer-Verlag.

[4] 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.

[5] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In Matthew K. Franklin, editor, CRYPTO, volume 3152 of Lecture Notes in Computer Science, pages 443–459. Springer, 2004.

[6] 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.

[7] 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.

[8] 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.
[9] Florian Hess, Nigel P. Smart, and Frederik Vercauteren. The eta pairing revisited. *IEEE Transactions on Information Theory*, 52:4595–4602, 2006.

[10] 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.

[11] Neal Koblitz. *Algebraic aspects of cryptography*. Springer-Verlag New York, Inc., New York, NY, USA, 1998.

[12] Duc-Phong Le, Alexis Bonnecaze, and Alban 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*, Pairing '09, pages 35–51. Springer Berlin/Heidelberg, 2009.

[13] Duc-Phong Le and Chao-Liang Liu. Refinements of Miller’s Algorithm over Weierstrass Curves Revisited. *The Computer Journal*, 54(10):1582–1591, 2011.

[14] Victor S. Miller. The Weil Pairing, and Its Efficient Calculation. *Journal of Cryptology*, 17(4):235–261, 2004.

[15] Frederik Vercauteren. Optimal pairings. *IEEE Transactions on Information Theory*, 56(1):455–461, 2010.

[16] 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.

[17] 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.