Refinement of Miller's Algorithm Over Edwards Curves

Lei Xu[1,2] and Dongdai Lin[1]

[1] State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing, China

[2] Graduate University of Chinese Academy of Sciences, Beijing, China

Abstract. Edwards gave a new form of elliptic curves in [1], and these curves were introduced to cryptography by Bernstein and Lange in [2]. The Edwards curves enjoy faster addition and doubling operations, so they are very attractive for elliptic curve cryptography.

In 2006, Blake, Murty and Xu proposed three refinements to Millers algorithm for computing Weil/Tate pairings over Weierstraß curves. In this paper we extend their method to Edwards curve and propose a faster algorithm for computing pairings with Edwards coordinates, which comes from the analysis of divisors of rational functions.

Keywords: Cryptography, bilinear pairing, Miller algorithm, twisted Edwards curve.

1 Introduction

Bilinear pairing on elliptic curves are of great interests due to their application in cryptography. It was first introduced by Alfred J.Menezes, Tatsuaki Okamoto and Scott A.Vanstone to reduce the discrete logarithms of elliptic curves to fintie fields([3]), which is known as the MOV attack. Frey and Rück([4]) also consider this situation using Tate pairing instead of Weil pairing.

Recent work on bilinear pairing has considered their positive applications. Dan Boneh and Matt Franklin proposed the first practical identity based encryption scheme([5]), which was first described by Shamir([6]). And many interesting applications of bilinear pairing are developed. Such as a one round protocol for tripartite Diffie-Hellman key exchange by Antoine Joux([7]), a short signature from Weil pairing by D.Boneh, B.Lynn and H.Shacham([8]) and so on.

Due to their various applications, a lot of effort has gone into efficient computing of bilinear pairing. Miller propose the first effective algorithm to calculate the bilinear pairing, which works in double-and-add manner([9]). And many improvements had been done to accelerate Miller's algorithm, see [10] for a survey.

In 2007, Edwards generalized an example from Euler and Gauss and introduced a new form of elliptic curve([1]). He showed that all elliptic curves over number fields could be transformed to the shape $x^2 + y^2 = c^2(1 + x^2y^2)$, with $(0, c)$ as neutral element and with simple and symmetric addition law. Bernstein and Lange in [2] presented fast explicit formulas for addition and doubling in

J. Pieprzyk (Ed.): CT-RSA 2010, LNCS 5985, pp. 106–118, 2010.

projective coordinates on an Edwards curve. They also generalize the addition law to the curve $x^2+y^2 = c^2(1+dx^2y^2)$, which covers more elliptic curves over finite field. In [11], Bernstein, Birkner, Joye, Lange, and Peters further generalized the Edwards curve to cover all curves $ax^2 + y^2 = 1 + dx^2y^2$.

Compared with Weierstraß curves, Edwards curves enjoy more efficient addition and double operation. So discrete logarithm based systems such as Diffie-Hellman key exchange or digital signatures that require efficient computation of scalar multiples benefit from Edwards curve.

However, the situation becomes complicated when Edwards curve is in the world of pairing based cryptography, where Miller's algorithm needs a function whose divisor is $(P) + (Q) - (P + Q) - (\mathcal{O})$, for input points P, Q and their sum $P + Q$.

Until recently, little work was dedicated to the improvement of pairing computation over Edwards curves. M. Prem Laxman Das and Palash Sarkar([12]) used birational equivalence to Weierstraß curves to calculate the bilinear pairing, Sorina Ionica and Antoine Joux([13]) used a different map to curve of degree 3 and compute the 4-th power of the Tate pairing, which is faster than Das and Sarkar's algorithm. Christophe Aréne, Tanja Lange, Michael Naehrig, and Christophe Ritzenthaler([14]) improved the computation of bilinear pairing on Edwards curve and twisted Edwards curve in a way that is similar to [15]. [14] also gives a geometric interpretation of the group law on Edwards curves and concise formulas for the coefficients of the conic.

In this paper, we propose an efficient algorithm to compute bilinear pairing over twisted Edwards curves. Our improvement comes from the consideration of the different combinations of the divisors and is different from the previous effort on pairing computation on Edwards curves.

The remainder of this paper is organized as follows: Section 2 recalls basic properties of bilinear pairing and Edwards curves. Section 3 presents our improvements to the original Miller's algorithm for twisted Edwards curves. In Section 4 we compare the improved algorithms with the original algorithm and give some detailed analysis. Section 5 gives the conclusion and some comments.

2 Background on Pairing and Twisted Edwards Curves

2.1 Bilinear Pairing and Miller's Algorithm

Let E/K be an elliptic curve. Weil pairing and Tate pairing are the two most important bilinear pairings.

Definition 1 (Divisor). *A divisor is an element of the free abelian group (Denoted by $Div(E)$) generated by the set of points of $E(\overline{K})$.*

Given a divisor $D = \sum_{P\in E} n_P(P)$, the degree of D is defined by $deg(D) = \sum_{P\in E} n_P$. The sum of divisor D is defined by $sum(D) = \sum_{P\in E} n_P P$.

The divisor of degree 0 is a subgroup of $Div(E)$ and is denoted by $Div^0(E)$.

The support of divisor D is the set of points P with $n_P \neq 0$.

For a nonzero rational function f over E, the corresponding divisor is defined to be $div(f) = ord_P(f)(P)$. It can be proved that $div(f) \in Div^0(E)$, and $div(f)$ is called principal divisor. A characterization of principal divisors is : $D = \sum_{P \in E} n_P(P)$ is principal iff $deg(D) = 0$ and $sum(D) = \mathcal{O}$, where $\mathcal{O}$ is the neutral element of the points group. The relation $\sim$ on $Div^0(E)$ is defined to be $D_1 \sim D_2$ iff $D_1 - D_2$ is principal.

If f is a nonzero rational function such that $div(f)$ and D have disjoint supports, then the evaluation of f at D is defined by $f(D) = \prod_{P \in E} f(P)^{n_p}$.

For more information on rational functions, divisors, and their relations, we refer the readers to [16].

Let n be an integer which is prime to $p = char(K)$ and $E[n] = \{P \in E(\overline{K}) | nP = \mathcal{O}\}$. Take $P, Q \in E[n]$, there exist $D_P, D_Q \in Div^0(E)$ s.t. $D_P \sim (P) - (\mathcal{O}), D_Q \sim (Q) - (\mathcal{O})$. Let $div(f_P) = nD_P$, $div(f_Q) = nD_Q$, and suppose that D_P, D_Q have disjoint supports, the *Weil pairing* is defined to be:

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

Take a point $S \in E$ s.t. $D_Q = (Q + S) - (S)$ and $div(f_P)$ have disjoint supports. Then the *Tate pairing* is defined to be:

$$\begin{aligned} \phi_n : E(K)[n] \times (E(K)/nE(K)) \mapsto K^*/(K^*)^n \\ \phi_n(P, \overline{Q}) = \overline{f_P(D_Q)} \end{aligned}$$

Here $\overline{Q}$ is the equivalence class in $E(K)/nE(K)$ containing Q and $\overline{f_P(D_Q)}$ is the equivalence class in $K^*/(K^*)^n$.

An essential part in computing the Weil/Tate pairing is the evaluation of rational function f_P at some divisor D. Miller gave an efficient algorithm for this calculation.

The main idea of Miller's algorithm is to calculate $f_P(D_Q)$ recursively. Specifically, pick a random point S, and let $D_P = (P+S)-(S)$. Then $div(f_P) \sim nD_P$. For each integer k, there is a rational function f_k s.t.

$$div(f_k) = k(P + S) - k(S) - (kP) + (\mathcal{O}).$$

In particular, $f_n = f_P$.

Let $L_{P,Q}$ be the line passing through points P, Q and L_P be the vertical line passing through point P. Then we have

$$\begin{aligned} div(L_{k_1P,k_2P}) &= (k_1P) + (k_2P) + (-(k_1 + k_2)P)) - 3(\mathcal{O}) \\ div(L_{kP}) &= (kP) + (-kP) - 2(\mathcal{O}). \end{aligned}$$

So

$$div(f_{k_1+k_2}) = div(f_{k_1}) + div(f_{k_2}) + div(L_{k_1P,k_2P}) - div(L_{(k_1+k_2)P}).$$

In other words,

$$f_{k_1+k_2} = f_{k_1} f_{f_2} L_{k_1P,k_2P}/H_{(k_1+k_2)P}.$$

The initial values are: $f_0 = 1$ and $f_1 = L_{P,R}/L_{P+R}$.

Algorithm 1 describes Miller's method(see [9] for details).

Algorithm 1. Miller's Algorithm

```
Input: Elliptic curve E, integer $n = \sum_{i=0}^{t} b_i 2^i, b_t \neq 0$, points $P, Q \in E, order(P) = n$
Output: $f = f_n(S)$
  $f \leftarrow f_1; Q \leftarrow P;$
  for $j \leftarrow t-1$ down to 0 do
    $f \leftarrow f^2 \frac{L_{Q,Q}(S)}{L_{2Q}(S)}; Q \leftarrow 2Q;$
    if $b_j = 1$ then
      $f \leftarrow f_1 f \frac{L_{Q,P}(S)}{L_{Q+P}(S)}; Q \leftarrow Q + P;$
    end if
  end for
  return $f$
```

2.2 Twisted Edwards Curves

Bernstein et al. introduced the twisted Edwards curve in [11]. Here we give a brief description.

For finite field K with character different from 2, the twisted Edwards curve is defined as:

$$E_{a,d} : ax^2 + y^2 = 1 + dx^2y^2, \ where \ a, d \in K^* \ and \ a \neq d$$

The neutral element is $\mathcal{O} = (0, 1)$ and element $\mathcal{O}' = (0, -1)$ has order two. It also has two points at infinity, denoted by $\Omega_1 = (1 : 0 : 0), \Omega_2 = (0 : 1 : 0)$. Notice these two points are singular and have multiplicity two.

The addition law on points of the curve $E_{a,d}$ is

$$(x_1, y_1) + (x_2, y_2) = (\frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - ax_1x_2}{1 - dx_1x_2y_1y_2}) \tag{1}$$

It is proved in [11] that if a is a square and d is not a square, then formula (1) is complete.

[11] also gives explicit formulae for twisted Edwards curves in projective coordinates. In projective coordinates, twisted Edwards curve is defined as:

$$(aX^2 + Y^2)Z^2 = Z^4 + dX^2Y^2$$

For $Z_1 \neq 0$ the homogeneous point $(X_1 : Y_1 : Z_1)$ represents the affine point $(X_1/Z_1, Y_1/Z_1)$ on $E_{a,d}$.

Addition in Projective Twisted Coordinates. The following formulae compute $(X_3 : Y_3 : Z_3) = (X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2)$ in $10\mathbf{M} + 1\mathbf{S} + 2\mathbf{D} + 7\mathbf{add}$, where the $2\mathbf{D}$ are one multiplication by a and one by d:

$A = Z_1 \cdot Z_2; B = A^2; C = X_1 \cdot X_2; D = Y_1 \cdot Y_2; E = dC \cdot D;$
$F = B \cdot E; G = B + E; X_3 = A \cdot F \cdot ((X_1 + Y_1) \cdot (X_2 + Y_2) \cdot C \cdot D);$
$Y_3 = A \cdot G \cdot (D \cdot aC); Z_3 = F \cdot G.$

Doubling in Projective Twisted Coordinates. The following formulae compute $(X3 : Y3 : Z3) = 2(X1 : Y1 : Z1)$ in $3\mathbf{M} + 4\mathbf{S} + 1\mathbf{D} + 7\mathbf{add}$, where the $1\mathbf{D}$ is a multiplication by a:

$$B = (X_1 + Y_1)^2; C = X_1^2; D = Y_1^2; E = aC; F := E + D; H = Z_1^2;$$
$$J = F \cdot 2H; X_3 = (B \cdot C \cdot D) \cdot J; Y_3 = F \cdot (E \cdot D); Z_3 = F \cdot J.$$

2.3 Bilinear Pairing Over Edwards Curves

In [14], the authors gave a geometry explanation for the addition law over twisted Edwards curves and a method to construct rational function with divisor $(P_1) + (P_2) - (P_3) - (\mathcal{O})$, which is essential for Miller's algorithm.

Let h_{P_1,P_2} be the conic passing through $\Omega_1, \Omega_2, \mathcal{O}', P_1, P_2$; ℓ_{1,P_3} be the horizontal line passing through P_3; $\ell_{2,\mathcal{O}}$ be the vertical line passing through $\mathcal{O}$.

From [14], we have the following lemma:

Lemma 1. *Let ℓ_1, ℓ_2, h be defined as above. Then*

$$div(\ell_{1,P_3}) = (P_3) + (-P_3) - 2(\Omega_2) \tag{2}$$
$$div(\ell_{2,\mathcal{O}}) = (\mathcal{O}) + (\mathcal{O}') - 2(\Omega_1) \tag{3}$$
$$div(h_{P_1,P_2}) = (P_1) + (P_2) + (\mathcal{O}') + (-P_3) - 2(\Omega_1) - 2(\Omega_2) \tag{4}$$

It is easy to prove Corollary 1 with Lemma 1.

Corollary 1. *Let ℓ_1, ℓ_2, h be the same as in Lemma 1. Then*

$$\frac{h_{P_1,P_2}}{\ell_{1,P_3}\ell_{2,\mathcal{O}}} = (P_2) + (P_2) - (P_3) - (\mathcal{O}) \tag{5}$$

We get the Edwards edition of Miller's algorithm(Algorithm 2).

Algorithm 2. Miller's Algorithm for Twisted Edwards Curve

```
Input: Twisted Edwards curve $E_{a,d}$, integer $n = \sum_{i=0}^{t} b_i 2^i, b_t \neq 0$, points $P, S \in E, order(P) = n$
Output: $f = f_n(S)$
$f \leftarrow f_1; Q \leftarrow P$;
for $j \leftarrow t - 1$ down to 0 do
    $f \leftarrow f^2 \frac{h_{Q,Q}(S)}{\ell_{2,2Q}(S)\ell_{1,\mathcal{O}}(S)}$
    $Q \leftarrow 2Q$
    if $b_j = 1$ then
        $f \leftarrow f_1 f \frac{h_{Q,P}(S)}{\ell_{2,Q+P}(S)\ell_{1,\mathcal{O}(S)}}$
        $Q \leftarrow Q + P$
    end if
end for
return $f$
```

3 Our Improvements

The main loop of Miller's algorithm (i.e from line 2 to line 8 in Algorithm 2) takes most of the running time. So we focus on the improvements of the operations in the loop.

In 2006 Blake, Murty and Xu([17]) proposed a method to reduce the total number of lines in Miller's algorithm. Though this concept does not dramatically decrease the cost of points adding, it is novel and can be applied to decrease the number of field multiplications.

In this paper we extend their technique to twisted Edwards curves and achieve some improvements.

Specifically, notice that in Miller's algorithm for twisted Edwards curves (Algorithm 2), only one bit of the integer n is considered in one iteration. If we consider two consecutive bits at a time, we can achieve some improvements.

First we give a theorem which is fundamental to the improvements.

Theorem 1. *Let $E_{a,d}$ be a twisted Edwards curve and $Q \in E_{a,d}$ with order$(Q) = n$. Then*[1]

1.

$$(\frac{h_{Q,Q}}{\ell_{1,2Q}\ell_{2,\mathcal{O}}})^2\frac{h_{2Q,2Q}}{\ell_{2,4Q}\ell_{1,\mathcal{O}}}=\frac{h_{Q,Q}^2}{h_{-2Q,-2Q}h_{\mathcal{O},\mathcal{O}}}$$

2.

$$\frac{h_{Q,Q}}{\ell_{2,2Q}\ell_{1,\mathcal{O}}}\frac{h_{2Q,P}}{\ell_{2,2Q+P}\ell_{1,\mathcal{O}}}=\frac{h_{Q,Q}\ell_{2,P}}{h_{2Q+P,-P}\ell_{1,\mathcal{O}}}$$

Proof. 1. The divisor of the rational function

$$(\frac{h_{Q,Q}}{\ell_{1,2Q}\ell_{2,\mathcal{O}}})^2\frac{h_{2Q,2Q}}{\ell_{1,4Q}\ell_{2,\mathcal{O}}}$$

is

$$\begin{aligned}
&\frac{4(Q)+2(-2Q)-4(\mathit{\Omega}_1)-4(\mathit{\Omega}_2)+2(\mathcal{O}')}{2(2Q)+2(-2Q)-4(\mathit{\Omega}_2)+2(\mathcal{O})+2(\mathcal{O}')-4(\mathit{\Omega}_1)}\\
&\quad+\frac{2(2Q)+(-4Q)-2(\mathit{\Omega}_1)-2(\mathit{\Omega}_2)+(\mathcal{O}')}{(4Q)+(-4Q)-2(\mathit{\Omega}_2)+(\mathcal{O})+(\mathcal{O}')-2(\mathit{\Omega}_1)}\\
=&\frac{4(Q)+2(-2Q)-4(\mathit{\Omega}_1)-4(\mathit{\Omega}_2)+2(\mathcal{O}')}{2(-2Q)-2(\mathit{\Omega}_2)+2(\mathcal{O})+2(\mathcal{O}')-2(\mathit{\Omega}_1)+(4Q)-2(\mathit{\Omega}_2)+(\mathcal{O})-2(\mathit{\Omega}_1)}\\
=&4(Q)+2(-2Q)-4(\mathit{\Omega}_1)-4(\mathit{\Omega}_2)+2(\mathcal{O}')+\\
&\frac{1}{2(-2Q)+(4Q)-2(\mathit{\Omega}_1)-2(\mathit{\Omega}_2)+(\mathcal{O}')}+\\
&\frac{1}{3(\mathcal{O})+(\mathcal{O}')-2(\mathit{\Omega}_2)-2(\mathit{\Omega}_1)}
\end{aligned}$$

[1] Notice that the fraction of divisor a/b means $a - b$.

The divisor $4(Q)+2(-2Q)-4(\Omega_1)-4(\Omega_2)+2(\mathcal{O}')$ corresponds to $h_{Q,Q}^2$, $2(-2Q)+(4Q)-2(\Omega_1)-2(\Omega_2)+(\mathcal{O}')$ corresponds to $h_{-2Q,-2Q}$ and $3(\mathcal{O})+(\mathcal{O}')-2(\Omega_2)-2(\Omega_1)$ corresponds to $h_{\mathcal{O},\mathcal{O}}$.
So we have

$$(\frac{h_{Q,Q}}{\ell_{1,2Q}\ell_{2,\mathcal{O}}})^2\frac{h_{2Q,2Q}}{\ell_{2,4Q}\ell_{1,\mathcal{O}}}=\frac{h_{Q,Q}^2}{h_{-2Q,-2Q}h_{\mathcal{O},\mathcal{O}}}.$$

2. The divisor of the rational function

$$\frac{h_{Q,Q}}{\ell_{2,2Q}\ell_{1,\mathcal{O}}}\frac{h_{2Q,P}}{\ell_{2,2Q+P}\ell_{1,\mathcal{O}}}$$

is

$$\begin{aligned}
&\frac{2(Q)+(-2Q)-2(\Omega_1)-2(\Omega_2)+(\mathcal{O}')}{(2Q)+(-2Q)-2(\Omega_2)+(\mathcal{O})+(\mathcal{O}')-2(\Omega_1)}\\
&+\frac{(2Q)+(P)+(-(2Q+P))-2(\Omega_1)-2(\Omega_2)+(\mathcal{O}')}{(2Q+P)+(-(2Q+P))-2(\Omega_2)+(\mathcal{O})+(\mathcal{O}')-2(\Omega_1)}\\
=&2(Q)+(-2Q)-2(\Omega_1)-2(\Omega_2)+(\mathcal{O}')+\\
&\frac{(P)}{(2Q+P)+(-2Q)-2(\Omega_1)-2(\Omega_2)+(\mathcal{O}')+2(\mathcal{O})}\\
=&2(Q)+(-2Q)-2(\Omega_1)-2(\Omega_2)+(\mathcal{O}')+\\
&\frac{(P)+(-P)-2(\Omega_2)}{(2Q+P)+(-2Q)+(-P)-2(\Omega_1)-2(\Omega_2)+(\mathcal{O}')+2(\mathcal{O})-2(\Omega_2)}
\end{aligned}$$

The divisor $2(Q)+(-2Q)-2(\Omega_1)-2(\Omega_2)+(\mathcal{O}')$ corresponds to $h_{Q,Q}$, $(P)+(-P)-2(\Omega_2)$ corresponds to $\ell_{1,P}$, $(2Q+P)+(-2Q)+(-P)-2(\Omega_1)-2(\Omega_2)+(\mathcal{O}')$ correponds to $h_{2Q+P,-P}$, and $2(\mathcal{O})-2(\Omega_2)$ corresponds to $\ell_{1,\mathcal{O}}$.
So we have

$$\frac{h_{Q,Q}}{\ell_{2,2Q}\ell_{1,\mathcal{O}}}\frac{h_{2Q,P}}{\ell_{2,2Q+P}\ell_{1,\mathcal{O}}}=\frac{h_{Q,Q}\ell_{2,P}}{h_{2Q+P,-P}\ell_{1,\mathcal{O}}}.$$

Next we describe the improvements using Theorem 1 in four different cases.

1. If two consecutive bits of n are "00", then according to Algorithm 2, line 3 $\sim$ 4 are executed twice.
The result of the execution is

$$\begin{aligned}
&f\leftarrow(f^2\frac{h_{Q,Q}(S)}{\ell_{2,2Q}(S)\ell_{1,\mathcal{O}}(S)})^2\frac{h_{2Q,2Q}(S)}{\ell_{2,4Q}(S)\ell_{1,\mathcal{O}}(S)}\\
&Q\leftarrow 4Q
\end{aligned}$$

Using the first formula of Theorem 1, the above operations are equal to

$$\begin{aligned}
&f\leftarrow f^4\frac{h_{Q,Q}^2(S)}{h_{-2Q,-2Q}(S)h_{\mathcal{O},\mathcal{O}}(S)}\\
&Q\leftarrow 4Q
\end{aligned}$$

2. If two consecutive bits of n are "01", then according to Algorithm 2, line 3 $\sim$ 4 are executed twice, and line 6 $\sim$ 7 are executed.
   The result of the execution is

$$f \leftarrow f_1(f^2 \frac{h_{Q,Q}(S)}{\ell_{2,2Q}(S)\ell_{1,\mathcal{O}}(S)})^2 \frac{h_{2Q,2Q}(S)}{\ell_{2,4Q}(S)\ell_{1,\mathcal{O}}(S)} \frac{h_{4Q,P}(S)}{\ell_{2,4Q+P}(S)\ell_{1,\mathcal{O}(S)}}$$
$$Q \leftarrow 4Q + P$$

   In this case we have two ways to combine the divisor of the result rational function f.

   (a) Using the first formula of Theorem 1, the above operations are equal to

$$f \leftarrow f_1 f^4 \frac{h_{Q,Q}^2(S)}{h_{-2Q,-2Q}(S)h_{\mathcal{O},\mathcal{O}}(S)} \frac{h_{4Q,P}(S)}{\ell_{2,4Q+P}\ell_{1,\mathcal{O}}(S)}$$
$$Q \leftarrow 4Q + P$$

   (b) We can also use the second formula of Theorem 1, then the above operations are equal to

$$f \leftarrow f_1(f^2 \frac{h_{Q,Q}(S)}{\ell_{2,2Q}(S)\ell_{1,\mathcal{O}}(S)})^2 \frac{h_{2Q,2Q}(S)\ell_{2,P}(S)}{h_{4Q+P,-P}(S)h_{\mathcal{O},\mathcal{O}}(S)}$$
$$Q \leftarrow 4Q + P$$

3. If two consecutive bits of n are "10", then according to Algorithm 2, line 3 $\sim$ 7 are executed, and line 3 $\sim$ 4 are executed.
   The result of the execution is

$$f \leftarrow (f_1 f^2 \frac{h_{Q,Q}(S)}{\ell_{2,2Q}(S)\ell_{1,\mathcal{O}}(S)} \frac{h_{2Q,P}(S)}{\ell_{2,2Q+P}(S)\ell_{1,\mathcal{O}(S)}})^2 \frac{h_{2Q+P,2Q+P}(S)}{\ell_{2,4Q+2P}(S)\ell_{1,\mathcal{O}}(S)}$$
$$Q \leftarrow 4Q + 2P$$

   In this case we have two ways to combine the divisor of the result rational function f.

   (a) Using the first formula of Theorem 1, the above operations are equal to

$$f \leftarrow f_1^2 f^4 \frac{h_{Q,Q}^2(S)}{\ell_{2,2Q}^2(S)\ell_{1,\mathcal{O}}^2(S)} \frac{h_{2Q,P}^2(S)}{h_{-(2Q+P),-(2Q+P)}(S)h_{\mathcal{O},\mathcal{O}}(S)}$$
$$Q \leftarrow 4Q + 2P$$

   (b) Using the second formula of Theorem 1, the above operations are equal to

$$f \leftarrow (f1f^2 \frac{h_{Q,Q}(S)\ell_{2,P}(S)}{h_{2Q+P,-P}(S)\ell_{1,\mathcal{O}}(S)})^2 \frac{h_{2Q+P,2Q+P}(S)}{\ell_{2,4Q+2P}(S)\ell_{1,\mathcal{O}}(S)}$$
$$Q \leftarrow 4Q + 2P$$

4. If two consecutive bits of n are "11", then according to Algorithm 2, line 3 $\sim$ 7 are executed twice.
   The result of the execution is

$$f \leftarrow f_1(f_1 f^2 \frac{h_{Q,Q}(S)}{\ell_{2,2Q}(S)\ell_{1,\mathcal{O}}(S)} \frac{h_{2Q,P}(S)}{\ell_{2,2Q+P}(S)\ell_{1,\mathcal{O}(S)}})^2 \frac{h_{2Q+P,2Q+P}(S)}{\ell_{2,4Q+2P}(S)\ell_{1,\mathcal{O}}(S)} \frac{h_{4Q+2P,P}(S)}{\ell_{2,4Q+3P}(S)\ell_{1,\mathcal{O}(S)}}$$
$$Q \leftarrow 4Q + 3P$$

   As in the cases "01" and "10", we have two ways to combine the divisor of the result rational function f.
   (a) Using the first formula of Theorem 1, the above operations are equal to

$$f \leftarrow f_1^3 f^4 \frac{h^2_{Q,Q}(S)}{\ell^2_{2,2Q}(S)\ell^2_{1,\mathcal{O}}(S)} \frac{h^2_{2Q,P}(S)}{h_{-(2Q+P),-(2Q+P)}(S)h_{\mathcal{O},\mathcal{O}}(S)} \frac{h_{4Q+2P,P}(S)}{\ell_{2,4Q+3P}(S)\ell_{1,\mathcal{O}}(S)}$$
$$Q \leftarrow 4Q + 3$$

   (b) Using the second formula of Theorem 1, the above operations are equal to

$$f \leftarrow f_1(f_1 f^2 \frac{h_{Q,Q}(S)\ell_{2,P}(S)}{h_{2Q+P,-P}(S)\ell_{1,\mathcal{O}}(S)})^2 \frac{h_{2Q+P,2Q+P}(S)\ell_{2,P}(S)}{h_{4Q+3P,-P}(S)h_{\mathcal{O},\mathcal{O}(S)}}$$
$$Q \leftarrow 4Q + 3$$

It is easy to derive the concrete algorithm from the above description(Algorithm 3). To consider two bits of n at a time, we represent n in 4-base. If the number of bits of the integer n is odd, we initialize f with the first bit, otherwise we use the first two bits to initialize f. And we use different method to combine the divisor of f in different cases. The reasons to use different combinations are shown in Section 4.

4 Analysis and Comparison

The cost of the algorithms calculating bilinear pairing over Edwards curves consists of three parts: the cost of updating f, the cost of updating Q, and the cost of evaluating $h_{Q,R}, \ell_{1,Q}, \ell_{2,Q}$ at some point S. Note that here we mainly make comparison with the original Miller's algorithm(Algorithm 2).

Without special treatment, the cost of updating Q in Algorithm 3 is no more than that of Algorithm 2. And the cost of evaluating $h_{Q,R}, \ell_{1,Q}, \ell_{2,Q}$ at some point S is also the same for the two algorithms[2]. So we focus on the cost of updating f.

Let **M** denote the cost of finite field multiplication, **S** denote the cost of finite field square and **I** denote the finite field inversion. Notice that there is always a multiplication following an inversion.

Field inversion is much more expensive compared with multiplication. And square is cheaper than multiplication. We set $\mathbf{S} = 0.8\mathbf{M}$ and $\mathbf{I} > 8\mathbf{M}$. Because

[2] If we take these costs in consideration, there may be room for further improvements.

Algorithm 3. Improved Miller's Algorithm for Edwards Curve

```
Input: Twisted Edwards curve $E_{a,d}$, integer $n = \sum_{i=0}^{t} q_i 4^i, q_t \neq 0$, points $P, S \in E_{a,d}, order(P) = n$
Output: $f = f_n(S)$
  $f \leftarrow f_1$, $Q \leftarrow P$
  if number of bits of $n$ is even then
    if $q_r = 2$ then
      $f \leftarrow f^2 \frac{h_{P,P}(S)}{\ell_{2,2P}(S)\ell_{1,\mathcal{O}}(S)}$, $Q \leftarrow 2P$
    end if
    if $q_r = 3$ then
      $f \leftarrow f^2 \frac{h_{P,P}(S)}{\ell_{2,2P}(S)\ell_{1,\mathcal{O}}(S)} \frac{h_{2P,P}(S)}{\ell_{2,3P}(S)\ell_{1,\mathcal{O}}(S)}$, $Q \leftarrow 3P$
    end if
  end if
  for $j = t-1$ down to 0 do
    if $q_j = 0$ then
      $f \leftarrow f^4 \frac{h_{Q,Q}^2(S)}{h_{-2Q,-2Q}(S)h_{\mathcal{O},\mathcal{O}}(S)}$
      $Q \leftarrow 4Q$
    end if
    if $q_j = 1$ then
      $f \leftarrow f_1 f^4 \frac{h_{Q,Q}^2(S)}{h_{-2Q,-2Q}(S)h_{\mathcal{O},\mathcal{O}}(S)} \frac{h_{4Q,P}(S)}{\ell_{2,4Q+P}\ell_{1,\mathcal{O}}(S)}$
      $Q \leftarrow 4Q + P$
    end if
    if $q_j = 2$ then
      $f \leftarrow f_1^2 f^4 \frac{h_{Q,Q}^2(S)}{\ell_{2,2Q}^2(S)\ell_{1,\mathcal{O}}^2(S)} \frac{h_{2Q,P}^2(S)}{h_{-(2Q+P),-(2Q+P)}(S)h_{\mathcal{O},\mathcal{O}}(S)}$
      $Q \leftarrow 4Q + 2P$
    end if
    if $q_j = 3$ then
      $f \leftarrow f_1(f_1 f^2 \frac{h_{Q,Q}(S)\ell_{2,P}(S)}{h_{2Q+P,-P}(S)\ell_{1,\mathcal{O}}(S)})^2 \frac{h_{2Q+P,2Q+P}(S)\ell_{2,P}(S)}{h_{4Q+3P,-P}(S)h_{\mathcal{O},\mathcal{O}(S)}}$
      $Q \leftarrow 4Q + 3P$
    end if
  end for
  return $f$
```

the inversion operation is so expensive, we keep the middle result in fraction form($f = \frac{a}{b}$) to avoid the inversions.

First, we give a comparison of the effects of different combinations of divisor. The result of different combinations is given in Section 3, and the cost of these combinations is shown in Table 1.

From Table 1, it is easy to see that in the cases "01" and "10", the first combination save 1 **M** than the second. And in the case "11", the second method is faster than the first one(also save 1 **M**). There is only one way to combine the divisor in the case "00".

The construction of Algorithm 3 follows the above observations.

Table 1. Comparison of Different Combinations of Divisor of f

Method to combine the divisor	case "00"	case "01"	case "10"	case "11"
Using the first formula of Theorem 1	$5\mathbf{S} + 3\mathbf{M}$ $= 7\mathbf{M}$	$4\mathbf{S} + 7\mathbf{M}$ $= 10.2\mathbf{M}$	$4\mathbf{S} + 7\mathbf{M}$ $= 10.2\mathbf{M}$	$4\mathbf{S} + 11\mathbf{M}$ $= 14.2\mathbf{M}$
Using the second formula of Theorem 1	$5\mathbf{S} + 3\mathbf{M}$ $= 7\mathbf{M}$	$4\mathbf{S} + 8\mathbf{M}$ $= 11.2\mathbf{M}$	$4\mathbf{S} + 8\mathbf{M}$ $= 11.2\mathbf{M}$	$4\mathbf{S} + 10\mathbf{M}$ $= 13.2\mathbf{M}$

Specifically,

1. In the case "01" and "10", use the first formula of Theorem 1 to combine the divisor.
2. In the case "11", use the second formula of Theorem 1 to combine the divisor.

Next, We compare Algorithm 2 and the original Miller's algorithm(Algorithm 3). The result is showed in Table 2.

We remind that Algorithm 2 is slower than that of [14], where denominator elimination was used. And we include the result of [14] in the last row of Table 2.

Table 2. Comparison of the Algorithms

	case "00"	case "01"	case "10"	case "11"
Algorithm 2	$4\mathbf{S} + 6\mathbf{M}$ $= 9.2\mathbf{M}$	$4\mathbf{S} + 10\mathbf{M}$ $= 13.2\mathbf{M}$	$4\mathbf{S} + 10\mathbf{M}$ $= 13.2\mathbf{M}$	$4\mathbf{S} + 14\mathbf{M}$ $= 17.2\mathbf{M}$
Algorithm 3	$5\mathbf{S} + 3\mathbf{M}$ $= 7\mathbf{M}$	$4\mathbf{S} + 7\mathbf{M}$ $= 10.2\mathbf{M}$	$4\mathbf{S} + 7\mathbf{M}$ $= 10.2\mathbf{M}$	$4\mathbf{S} + 10\mathbf{M}$ $= 13.2\mathbf{M}$
Method of [14]	$2\mathbf{S} + 2\mathbf{M}$ $= 3.6\mathbf{M}$	$2\mathbf{S} + 3\mathbf{M}$ $= 4.6\mathbf{M}$	$2\mathbf{S} + 3\mathbf{M}$ $= 4.6\mathbf{M}$	$2\mathbf{S} + 4\mathbf{M}$ $= 5.6\mathbf{M}$

From Table 2, we can see that the improved Algorithm 3 is more efficient than the original Algorithm 2 in all the four cases. And in the case "11", the new algorithm can save at most 4**M** per iteration. So if there are more "11"s in the binary representation of integer n, Algorithm 3 will save more time.

Let $n = \sum_{i=0}^{t} 2^i b_i$, suppose

$$Prob[b_i = 1] = Prob[b_i = 0] = 1/2, where\ 0 \leq i < t,$$

and b_i are mutually independent.

Then

$$Prob[b_i b_{i+1} = 00] = Prob[b_i b_{i+1} = 01] = Prob[b_i b_{i+1} = 10] =$$
$$Prob[b_i b_{i+1} = 11] = 1/4$$

So the total cost of Algorithm 3 is about 76.8% of that of Algorithm 2.

5 Conclusion

In this paper we propose an improved Miller's algorithm for twisted Edwards curves. And we give a detailed analysis of the improvement. The savings in the number of multiplication in the updating of f noted is important for the performance of algorithms in the pairing based cryptosystems.

At the same time, we pay little attention to the calculation of $4Q+P, 4Q+2P$, and $4Q+3P$. There may be faster methods to calculate these values than simple additions and doubles. So it is possible to make our analysis result better.

Acknowledgements. We thank Professor Tanja Lange and the anonymous reviewers for their precious comments.

References

1. Edwards, H.M.: A Normal Form for Elliptic Curves. Bulletin of the American Mathematical Society 44, 393–442 (2007)
2. Bernstein, D.J., Lange, T.: Faster Addition and Doubleling on Elliptic Curves. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 29–50. Springer, Heidelberg (2007)
3. Menezes, A.J., Okamoto, T., Vanstone, S.A.: Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field. IEEE Transactions on Information Theory (1993)
4. Frey, G., Rück, H.G.: A Remark Concerning m-divisibility and the Discrete Logarithm in the Divisor Class Group of Curves. Mathematics of Computation 62, 865–874 (1994)
5. Boneh, D., Franklin, M.: Identity-based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
6. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
7. Joux, A.: A One Round Protocol for Tripartite Diffie-Hellman. In: Bosma, W. (ed.) ANTS 2000. LNCS, vol. 1838, pp. 385–393. Springer, Heidelberg (2000)
8. Boneh, D., Lynn, B., Shacham, H.: Short Signature from the Weil Pairing. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 514–532. Springer, Heidelberg (2001)
9. Miller, V.S.: The Weil Pairing, and its Efficient Calculation. Journal of Cryptology 17(4), 235–261 (2004)
10. Blake, I.F., Sroussi, G., Smart, P.N.: Advances in Elliptic Curve Cryptography. Cambridge University Press, Cambridge (2005)
11. Bernstein, D.J., Birkner, P., Joye, M., Lange, T., Peters, C.: Twisted Edwards Curves. In: Vaudenay, S. (ed.) AFRICACRYPT 2008. LNCS, vol. 5023, pp. 389–405. Springer, Heidelberg (2008)
12. Das, M.P.L., Sarkar, P.: Pairing Computation on Twisted Edwards form Elliptic Curves. In: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5209, pp. 192–210. Springer, Heidelberg (2008)
13. Ionica, S., Joux, A.: Another Approach to Pairing Computation in Edwards Coordinates. In: Chowdhury, D.R., Rijmen, V., Das, A. (eds.) INDOCRYPT 2008. LNCS, vol. 5365, pp. 400–413. Springer, Heidelberg (2008)

14. Aréne, C., Lange, T., Naehrig, M., Ritzenthaler, C.: Faster Pairing Computation. Cryptology ePrint Archive, Report 2009/155 (2009)
15. Barreto, P.S., Lynn, B., Scott, M.: Efficient Implementation of Pairing-based Cryptosystems. Journal of Cryptology 17, 321–334 (2004)
16. Hartshorne, R.: Algebraic Geometry. Graduate Texts in Mathematics. Springer, Heidelberg (1977)
17. Blake, I.F., Murty, V.K., Xu, G.: Refinements of Miller's Algorithm for Computing the Weil/Tate pairing. Journal of Algorithms 58, 134–149 (2006)