Another approach on pairing computation in Edwards coordinates

Antoine Joux[1,2] and Sorina Ionica[2]

[1] DGA
[2] Université de Versailles Saint-Quentin-en-Yvelines, 45 avenue des États-Unis, 78035 Versailles CEDEX, France
antoine.joux,sorina.ionica@m4x.org

Abstract. The recent introduction of Edwards curves has significantly reduced the cost of addition on elliptic curves. This paper presents new explicit formulae for pairing implementation in Edwards coordinates. We prove our method gives performances similar to those of Miller's algorithm in Jacobian coordinates and is thus of cryptographic interest when one chooses Edwards curve implementations of protocols in elliptic curve cryptography. The method is faster than the recent proposal of Das and Sarkar for computing pairings on supersingular curves using Edwards coordinates.

Keywords: Tate pairing, Miller's algorithm, Edwards coordinates

1 Introduction

Pairings on elliptic curves are currently of great interest due to their applications in a number of cryptographic protocols such as the tripartite Diffie Hellman [15], identity-based encryption [5], short signatures [6] and group signatures [7]. In this paper we propose to reasses the computational cost of pairings in the light of the introduction by Edwards [11] of a new representation of the addition law on elliptic curves. Recently, a method for computing pairings in Edwards and twisted Edwards coordinates for supersingular curves was proposed in [10]. The approach proposed in the present paper is very different from [10].

Our starting point concerning Edwards curves is a generalized result of Bernstein and Lange [3]. They showed that an elliptic curve defined over a field k of characteristic different from 2 is birationally equivalent over some extension of k to an Edwards curve, i.e. a curve of the form $x^2+y^2 = 1+dx^2y^2$ with $d \notin \{0,1\}$. A simple and symmetric addition law can be defined on such a curve:

$$(x_1, y_1), (x_2, y_2) \to \left(\frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - x_1x_2}{1 - dx_1x_2y_1y_2}\right). \tag{1}$$

Bernstein and Lange showed that this addition law is in fact the standard addition law on the corresponding elliptic curve and gave explicit formulae for additions and doublings, faster than all previously known formulae.

The algorithm used in pairing computation was first described by Miller and is an extension of double-and-add method for finding a point multiple. Our goal in this paper is to extend Miller's algorithm to allow computation of pairings on curves given in the Edwards coordinates. For benchmark purposes, we compare the efficiency of our extension to the use of Jacobian coordinates, which is, to the best of our knowledge, the faster existing method for computing pairings. In the case of supersingular curves, we also compare it to results in [10]. We proceed as in [16] and in [13] and count the number of field multiplications and squarings that appear in the doubling part of Miller's algorithm. This seems a fair basis for comparison when we pair points of r-torsion, with r with small Hamming weight. As the addition part of an iteration of Miller's algorithm has become increasingly important with the introduction of loop-shortening techniques, we also give global estimates of the number of operations to be done in a Miller iteration.

The difficulty when trying to express Miller's algorithm in Edwards coordinates is that it is hard to find the equations of rational functions that need to be evaluated at each addition step. On a curve in Weierstrass form, this equations are easily derived from the equation of straight lines. With the Edwards'representation matters are more complex.
Our basic idea is to describe a map of degree 4 from the Edwards curve to a curve of the form $E_{s,p} : s^2p = (1 + dp)^2 - 4p$. This curve has an equation of total degree 3 and just like for the Weierstrass form, we can easily compute the equations of the two lines that appear naturally when adding two points P_1 and P_2, i.e. the line l passing through P_1 and P_2 and the vertical line v that passes through $P_1 + P_2$. We then pullback l and v to the Edwards curve. The output of our algorithm is essentially the desired pairing. More precisely, we obtain the 4-th power of the usual pairing.

The remainder of this paper is organised as follows: Section 2 recalls basic properties of Edwards curves and of the Edwards addition law. It also presents Miller's algorithm on an elliptic curve given by a Weierstrass equation. Section 3 introduces the curve $E_{s,p}$ and explains how to compute pairings on Edwards curves by using this representation. Finally, in section 4 we give estimates of the computational cost of the Tate pairing in Edwards coordinates and compare this cost to that of a pairing implementation in Jacobian coordinates (for a Weierstrass equation). We treat apart the case of curves with even embedding degree k, which is prefered in most of the cryptographic applications. In this case a major part of the computations is performed in a proper subfield of F_{q^k}.

2 Preliminaries

2.1 Edwards coordinates

Edwards showed in [11] that every elliptic curve E defined over an algebraic number field k is birationally equivalent over some extension of k to a curve of equation:

$$x^2 + y^2 = c^2(1 + x^2y^2). \qquad (2)$$

In this paper, we make use of the results concerning elliptic curves over finite fields obtained by Bernstein et al. [2]:

Theorem 1. *Fix a field F_q with char $\neq 2$. Let E be an elliptic curve over F_q. E is birationally equivalent over F_q to an Edwards curve if and only if group $E(F_q)$ has an element of order 4.*

Moreover, it was actually shown in [2] that the curve $x^2 + y^2 = 1 + dx^2y^2$ is birationally equivalent to an elliptic curve $E_d : (1/(1-d))v^2 = u^3 + 2((1+d)/(1-d))u^2 + u$ via the rational map

$$\begin{aligned} \psi : E_d &\to E \\ (u,v) &\to \left(\frac{2u}{v}, \frac{(u-1)}{(u+1)}\right). \end{aligned} \tag{3}$$

One may compute the inverse rational map of ψ given by

$$(x,y) \to \left(\frac{(1+y)}{(1-y)}, \frac{2(1+y)}{(1-y)x}\right). \tag{4}$$

On an Edwards curve we consider the following addition law:

$$(x_1,y_1),(x_2,y_2) \to \left(\frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - x_1x_2}{1 - dx_1x_2y_1y_2}\right). \tag{5}$$

In [3], it was shown that this addition law corresponds to the standard addition law on the birationally equivalent elliptic curve and that the Edwards addition law is *complete* when d is not a square. This means it is defined for all pairs of input points on the Edwards curve with no exceptions for doublings, neutral element etc.

The neutral element of this addition law is $O = (0,1)$. For every point $P = (x,y)$ the opposite element is $-P = (-x,y)$. The curve has a 4-torsion subgroup defined over k. We note $T_2 = (0,-1)$ the point of order 2 and $T_4 = (1,0)$, $-T_4 = (-1,0)$ the two points of order 4.

In the following sections we use projective coordinates. A projective point (X,Y,Z) satisfying $(X^2+Y^2)Z^2 = Z^4 + dX^2Y^2$ and $Z \neq 0$ corresponds to the affine point $(X/Z, Y/Z)$ on the curve $x^2 + y^2 = 1 + dx^2y^2$. The Edwards curve has two points at infinity $(0:1:0)$ and $(1:0:0)$. These points are actually singularities of the curve and resolving singularities produces four points defined over $k(\sqrt{d})$, not over k, as stated in [3].

Edwards curves became interesting for elliptic curve cryptography when it was proven by Bernstein and Lange in [3] that they provide addition and doubling formulae faster than most previously known addition formulae . Table 1 below gives a cost comparison between operations of addition, doubling and mixed addition (i.e. the Z-coordinate of one of the two points is 1) on the Edwards curve and on the Weierstrass form in Jacobian coordinates. We remind the reader that a point (X,Y,Z) in Jacobian coordinates corresponds to the affine point (x, y) with $x = X/Z^2$ and $y = Y/Z^3$. We note by **M** the cost of a field multiplication and by **S** the cost of a field squaring. We assume that the cost of addition and that of multiplication by d negligible (we choose d a small constant).

Table 1. Performance evaluation: Edwards versus Jacobian

	Edwards coordinates	Jacobian coordinates
addition	10**M**+1**S**	11**M**+5**S** (plus **S**-**M** tradeoff)
doubling	3**M**+4**S**	1**M**+8**S** (plus 2 **S**-**M** tradeoffs) or 4**M**+4**S** for $a = -3$
mixed addition	9**M**+1**S**	7**M**+4**S** (plus **M**-**S** tradeoff)

2.2 Background on pairings

In this section we give a brief overview of the definition of the Tate pairing and of Miller's algorithm [17] used in pairing computation. This algorithm heavily relies on the double and add method for finding a point multiple. Let E be an elliptic curve of Weierstrass equation

$$y^2 = x^3 + ax + b, \tag{6}$$

defined over a finite field F_q. Consider r a large prime dividing $\#E(F_q)$ and k the corresponding embedding degree, i.e. the smallest positive integer such that r divides $q^k - 1$.

Let P be a r-torsion point and for every integer i, denote by $f_{i,P}$ the function with divisor $\operatorname{div}(f_{i,P}) = i(P) - (iP) - (i-1)(O)$. Note $f_{r,P}$ is such that $\operatorname{div}(f_{r,P}) = r(P) - r(O)$.

In order to define the Tate pairing we take Q an element of $E(F_{q^k})/rE(F_{q^k})$. Let T be a point on the curve such that the support of the divisor $D = (Q + T) - (T)$ is disjoint from the one of $f_{r,P}$. We then define the Tate pairing as:

$$t_r(P, Q) = f_{r,P}(D). \tag{7}$$

This value is a representative of an element of $F^*_{q^k}/(F^*_{q^k})^r$. However for cryptographic protocols it is essential to have a unique representative so we will raise it to the $((q^k - 1)/r)$-th power, obtaining an r-root of the unity. We call the resulting value the (reduced) Tate pairing:

$$T_r(P, Q) = t_r(P, Q)^{\frac{q^k-1}{r}}.$$

Before going into the details of Miller's algorithm we recall the standard addition law on an elliptic curve of Weierstrass equation. Suppose we want to compute

the sum of iP and jP for $i, j \geq 1$. Let l be the line through iP and jP. Then l intersects the cubic curve E at one further point R according to Bezout's theorem (see [14]). We take v the line between R and O (which is a vertical line when R is not O). Then v intersects E at one more point which is defined to be the sum of iP and jP, that is $(i+j)P$.
The lines l and v are functions on the curve and the corresponding divisors are

$$\begin{aligned} \operatorname{div}(l) &= (iP) + (jP) + (R) - 3(O) \\ \operatorname{div}(v) &= (R) + ((i+j)P) - 2(O). \end{aligned}$$

One can then easily check the following relation:

$$f_{i+j,P} = f_{i,P} f_{j,P} \frac{l}{v}. \tag{8}$$

In the sequel we will call this relation Miller's equation. Turning back to Miller's algorithm, suppose we want to compute $f_{r,P}(D)$. We compute at each step of the algorithm on one side $[m]P$, where m is the integer with binary expansion given by the i topmost bits of the binary expansion of r, and on the other side $f_{m,P}$ evaluated at D, by exploiting the formula above. We call the set of operations executed for each bit i of r a *Miller operation*.

Algorithm 1 Miller's algorithm

```
Choose a random point $T \in E(F_{q^k})$ and compute $Q^{'} = Q + T \in E(F_{q^k})$. Set $n = [\log_2(r)] - 1$, $K \leftarrow P$, $f \leftarrow 1$.
  while $n \geq 1$ do
    Compute equations of $l$ and $v$ arising in the doubling of $K$.
    $K \leftarrow 2K$ and $f \leftarrow f^2(l(Q^{'})v(T))/(v(Q^{'})l(T))$.
    if the $n$th bit of $r$ is one then
      Compute equations of $l$ and $v$ arising in the addition of $K$ and $P$.
      $K \leftarrow P + K, f \leftarrow f(l(Q^{'})v(S))/((l(S)v(Q^{'}))$.
    end if
    $n \leftarrow n - 1$.
  end while
```

The advantage of dealing with the Weierstrass form when running the algorithm is that the equations of l and v are easy to find as they already appear in the addition process. This is obviously not the case with the Edwards curve, whose equation has degree 4. It is difficult to describe the equation of a function with divisor equal to $\operatorname{div}(f_{i+j,P}/f_{i,P}f_{j,P})$ and establish a relation of type (8). An idea would be to consider the Miller equation on the birationally equivalent Weierstrass curve and then to transport this equation on the Edwards curve. However this yields an highly inefficient pairing computation. Our proposal is to map the Edwards curve to another genus 1 curve with an equation of degree 3, get l and v as straight lines and then pull them back to the Edwards curve.

3 Pairings on Edwards curves

In this section E denotes an Edwards curve defined over some finite field F_{q^k} of odd characteristic. Let us take a look at the action of the 4-torsion subgroup defined over k on a fixed point on the Edwards curve $P = (x, y)$, with $xy \neq 0$. A simple computation shows that $P + T_4 = (y, -x)$, $P + T_2 = (-x, -y)$ and $P - T_4 = (-y, x)$. We notice then that by noting $p = (xy)^2$ and $s = x/y - y/x$ we characterize the point P up to an addition with a 4-torsion point. This leads us to consider the following morphism from the Edwards curve to a curve of equation $E_{s,p} : s^2 p = (1 + dp)^2 - 4p$:

$$\begin{aligned} \phi : E \rightarrow E_{s,p} \\ \phi(x, y) \;=\; ((xy)^2, \frac{x}{y} - \frac{y}{x}). \end{aligned}$$

In this section we study the arithmetic of the $E_{s,p}$ curve, establish a Miller equation on this curve and then take its pullback, getting a Miller equation this time on the Edwards curve. This yields all the necessary tools to apply Miller's algorithm on the Edwards curve.

3.1 Arithmetic of the curve $s^2 p = (1 + dp)^2 - 4p$

In this section we study the arithmetic of the curve:

$$E_{s,p} : s^2 p = (1 + dp)^2 - 4p.$$

The equation of $E_{s,p}$ in homogenous coordinates (P, S, Z) is given by $S^2 P = (Z + DP)^2 Z - 4PZ^2$. If we dehomogenize this equation by putting $P = 1$ we get the Weierstrass equation of an elliptic curve

$$s^2 = z^3 + (2d - 4)z^2 + d^2 z. \tag{9}$$

We note $O_{s,p} = (0, 1, 0)$ the point at infinity and $T_{2,s,p} = (1, 0, 0)$ which is a two torsion point.

The following definition is simply another way to write the addition law on an elliptic curve in (p, s) coordinates.

Definition 1. *Let $P_1, P_2 \in E_{s,p}$, L the line connecting P_1 and P_2 (tangent line to $E_{s,p}$ if $P_1 = P_2$), and R the third point of intersection of L with E. Let $L^{'}$ be the vertical line through R (of equation $p = p_R$). Then $P_1 + P_2$ is the point such that $L^{'}$ intersects $E_{s,p}$ at R and $P_1 + P_2$ (the point symmetric to R with respect to the p axis).*

We now show that this addition law corresponds to the addition law induced by the Edwards addition law via the map ϕ.

Theorem 2. *Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be two points on the Edwards curve and P_3 their sum. Then $\phi(P_3)$ is the sum of $\phi(P_1)$ and $\phi(P_2)$ in the addition law of Definition 1.*

Proof. Consider $\psi : E_d \to E$ the map defined in Note 3. Then one can easily see that $\phi \circ \psi$ is an isogeny from E_d to the elliptic curve $E_{s,p}$. Moreover it was shown in Theorem 3.2 of [3] that the Edwards addition law on E is the same as the addition law induced by ψ. It follows that the addition law induced by ϕ is the same as the standard addition law on the elliptic curve, so it corresponds to the addition law described at Definition 1. □

As in the sequel we need to compute the pullback of certain functions on the $E_{s,p}$ curve we now compute the degree of this map.

Proposition 1. *The map $\phi : E \to E_{s,p}$ is separable of degree 4.*

Proof. Let $P = (x, y)$ be a point on the Edwards curve. The doubling formula gives:

$$2P = \left(\frac{2xy}{1 + d(xy)^2}, \frac{y^2 - x^2}{1 - d(xy)^2} \right) = \left(\frac{2xy}{x^2 + y^2}, \frac{y^2 - x^2}{2 - (x^2 + y^2)} \right).$$

If $xy \neq 0$ then by noting $p = (xy)^2$ and $s = x/y - y/x$ we can write:

$$4P = \left(\frac{4ps(1 - d^2p^2)}{(1 - d^2p^2)^2 - 4dp^2s^2}, \frac{4p(1 + dp)^2 - ps^2}{(1 - d^2p^2)^2 + 4dp^2s^2} \right).$$

This means that by defining

$$\begin{aligned} \psi :\ & E_{s,p} \to E \\ (p, s) \to & \left(\frac{4ps(1 - d^2p^2)}{(1 - d^2p^2)^2 - 4dp^2s^2}, \frac{4p(1 + dp)^2 - ps^2}{(1 - d^2p^2)^2 + 4dp^2s^2} \right) \end{aligned}$$

we get a rational map ψ such as $\phi \circ \psi = [4]$ on E. It follows that $\deg \phi$ divides 16. As $\deg_i \phi$ is a power of the characteristic of F_q, we deduct that ϕ is a separable map (we have supposed the characteristic of F_q different from 2). By putting $\phi(P) = Q$ we easily get $\phi^{-1}(Q) = \{P, P + T_2, P + T_4, P - T_4\}$. We conclude that $\deg \phi = 4$.□

3.2 Miller's algorithm on the Edwards curve

Let P be a r-torsion point on the Edwards curve. We consider slightly modified functions $f_{i,P}^{(4)}$:

$$\begin{aligned} f_{i,P}^{(4)} = & \, i((P) + (P + T_4) + (P + T_2) + (P - T_4)) \\ & - ((iP) + (iP + T_4) + (iP + T_2) + (iP - T_4)) \\ & - (i - 1)((O) + (T_4) + (T_2) + (-T_4)). \end{aligned}$$

Then $f_{r,P}^{(4)} = r((P) + (P + T_4) + (P + T_2) + (P - T_4)) - r((O) + (T_4) + (T_2) + (-T_4))$, which means that we can compute the Tate pairing up to a 4-th power:

$$T_r(P, Q)^4 = f_{r,P}^{(4)}(Q)^{\frac{q^k-1}{r}}.$$

We also get the following Miller equation:

$$f_{i+j,P}^{(4)} = f_{i,P}^{(4)} f_{j,P}^{(4)} \frac{l}{v}, \tag{10}$$

where l/v is the function of divisor

$$\begin{aligned}\mathrm{div}(\frac{l}{v}) = &((iP) + (iP + T_4) + (iP + T_2) + (iP - T_4)) \\ &+ ((jP) + (jP + T_4) + (jP + T_2) + (jP - T_4)) \\ &- (((i + j)P) + ((i + j)P + T_4) + ((i + j)P + T_2) + ((i + j)P - T_4))) \\ &- ((O) + (T_4) + (T_2) + (-T_4)).\end{aligned}$$

Let $P^{'} = \phi(P)$ and let $l_{s,p}$ and $v_{s,p}$ be functions on the $E_{s,p}$ curve such that div $(l_{s,p}) = (iP^{'}) + (jP^{'}) + (-(i + j)P^{'}) - 2(T_{2,s,p}) - (O_{s,p})$ and div $(v_{s,p}) = ((i + j)P^{'}) + (-(i + j)P^{'}) - 2(T_{2,s,p})$.

We observe that we have $l/v = \phi^*(l_{s,p}/v_{s,p})$ up to constants. It is easy to find the equations of $l_{s,p}$ and $v_{s,p}$ as they appear naturally in the definition of the sum $iP^{'} + jP^{'}$, namely $l_{s,p}$ is the line connecting $iP^{'}$ and $jP^{'}$, and $v_{s,p}$ is the vertical line through $(i+j)P^{'}$. As we will see in the next section, we can compute their pullback via the map ϕ without any significant computational cost.

4 Pairing computation in Edwards coordinates

In this section we take a look into the details of the computation of pairings in Edwards coordinates and give estimates of the computational costs of the Miller operation. Following [16] and [13], we start by estimating the cost of evaluating the function $f_{r,P}^{(4)}(D)$ in terms of the cost of the doubling part of a Miller operation, which is executed for every bit of r. This seems reasonable, as it gives an evaluation which is independent from any fast exponentiation techniques that might be used in the implementation of the algorithm, such as the sliding window method or the use of a signed Hamming weight representation for r. We recall that a signed representation $(m_{n-1}...m_0)_s$ is said to be in *non-adjacent form*, or NAF for short, if $m_i m_{i+1} = 0$, for all $i \geq 0$. It appears that this representation is unique and on average the number of non-zero terms in a NAF expansion of length n is $n/3$ (see [8]).

Moreover, in many cryptographic applications it is possible to choose r with low Hamming weight. Construction of supersingular curves of embedding degrees 1 and 2 given in [16] enables the choice of a r of low Hamming weight. As for ordinary curves, the construction of Cocks and Pinch as described in [4, p. 210] allows for r to be chosen arbitrarily, so a prime of low Hamming weight can be chosen. Further examples are provided by a construction of Brezing and Weng [9] for prime embeddings degrees k, extended in [12] for all odd $k < 200$.

Example 1. The following example is given in [10]. Consider $E : y^2 = x^3 + x$ over F_q, with $q \cong 3 \bmod 4$. Then this curve is supersingular and its corresponding Edwards form is $x^2 + y^2 = 1 - (xy)^2$, so $d = -1$. One may choose for instance $p = 2^{520} + 2^{363} - 2^{360} - 1$, $r = 2^{160} + 2^3 - 1$ or $p = 2^{1582} + 2^{1551} - 2^{1326} - 1, r = 2^{256} + 2^{225} - 1$.

In order to give a complete evaluation of the complexity, we also count the number of operations in the mixed addition step of the Miller operation and compare it to the mixed addition step in Jacobian coordinates. As to the best of our knowledge, estimates of the cost of the mixed addition step in Jacobian coordinates do not exist in the literature, we also give an operation count of this step in Appendix A.

4.1 The case $k = 1$

We write the functions l and v that appear in (10) as $l = l_1/l_2$ and $v = v_1/v_2$. We show that the denominators l_2 and v_2 are constant. It follows that in the double and add method, after initially setting $K = P$ and $f_1 = f_2 = 1$, we only have to do the following evaluations for the i-th bit of r:

$$\begin{aligned} &K \leftarrow 2K \\ &f_1 \leftarrow f_1^2 l_1(T+Q) v_1(T) \\ &f_2 \leftarrow f_2^2 l_1(T) v_1(Q+T). \end{aligned} \tag{11}$$

Following [3] the doubling formulas for $K = (X_1, Y_1, Z_1)$ are:

$$\begin{aligned} X_3 &= 2X_1Y_1(2Z_1^2 - (X_1^2 + Y_1^2)), \\ Y_3 &= (X_1^2 + Y_1^2)(Y_1^2 - X_1^2), \\ Z_3 &= (X_1^2 + Y_1^2)(2Z_1^2 - (X_1^2 + Y_1^2)). \end{aligned}$$

On the curve $E_{s,p}$ we consider $l_{s,p}$ the tangent line to the curve at $\phi(K) = (p_1, s_1)$ and $v_{s,p}$ the vertical line passing through $\phi(2K) = (p_3, s_3)$. These lines have the following equations:

$$\begin{aligned} l_{s,p}(s,p) &= 2s_1p_1^2(s - s_1) - (p - p_1)(2d(1 + dp_1) - (s_1^2 + 4))p_1, \\ v_{s,p}(s,p) &= p - p_3. \end{aligned}$$

Consequently we get the following equations of l and v on the Edwards curve:

$$\begin{aligned} l(x,y) = l_1(x,y)/l_2 = ((X_1^2 + Y_1^2 - Z_1^2)(X_1^2 - Y_1^2)((2X_1Y_1(x/y - y/x) \\ - 2(X_1^2 - Y_1^2)) - Z_3(dZ_1^2(xy)^2 - (X_1^2 + Y_1^2 - Z_1^2)))/Z_1^6 \\ v(x,y) = v_1(x,y)/v_2 = (dZ_3^2(xy)^2 - (X_3^2 + Y_3^2 - Z_3^2))/Z_3^2. \end{aligned}$$

At each step, we assume that X_1^2, Y_1^2 and Z_1^2, as well as $dZ_1^2(xy)^2 - (X_1^2 + Y_1^2 - Z_1^2)$ have already been computed as a side effect of the function evaluation corresponding to the last addition performed in the Miller operation for the $(i+1)$-th bit of r. Note that if the $(i+1)$-th bit is 1, this addition is not a doubling, but a mixed addition of K and P.

We choose to compute $f_{r,P}(D)$, with $D = (Q+T) - (T)$ where $T = (0,1)$.[3] We will actually make use of the following equation:

$$v_1(T)/l_1(T) = 4Z_1^2(Y_1^2 - X_1^2)/2X_1Y_1.$$

The operations to be done in the doubling step of Miller's algorithm are detailed in table 2:

Table 2. Operations of the doubling part of a Miller operation

$A = (X_1+Y_1)^2,\ B = X_1^2+Y_1^2,\ C = A-B,\ D = Y_1^2 - X_1^2,$	(1**S**)
$E = 2Z_1^2 - B,\ X_3 = C \cdot E,\ Y_3 = B \cdot D,\ Z_3 = B \cdot E$	(3**M**)
$F = X_3^2,\ G = Y_3^2,\ H = Z_3^2, I = Z_1^2 \cdot D,$	(3**S**+1**M**)
$K = C \cdot (x/y - y/x) + 2D,\ L = (Y_3 - I) \cdot K,\ M = Z_3 \cdot v_1$	(3**M**)
$l_1 = L - M, v_1 = dH \cdot (xy)^2 - (B - Z_1^2),$	(1**M**)
$X_1 = X_3, Y_1 = Y_3, Z_1 = Z_3,$	
$f_1 = f_1^2 \cdot l_1 \cdot (4I), f_2 = f_2^2 \cdot v_1 \cdot C.$	(2**S**+4**M**)

The mixed addition step in Miller's algorithm in Edwards coordinates is the following:

$$\begin{aligned} K &\leftarrow K + P \\ f_1 &\leftarrow f_1 l_1(T+Q) v_1(T) \\ f_2 &\leftarrow f_2 l_1(T) v_1(T+Q). \end{aligned} \tag{12}$$

We count the number of operations that must be executed when adding $K = (X_1, Y_1, Z_1)$ and $P = (X_0, Y_0, 1)$. The result is $K + P = (X_3, Y_3, Z_3)$ with:

$$\begin{aligned} X_3 &= Z_1(X_0Y_1 + Y_0X_1)(Z_1^2 + dX_0X_1Y_0Y_1) \\ Y_3 &= Z_1(Y_0Y_1 - X_0X_1)(Z_1^2 - dX_0X_1Y_0Y_1) \\ Z_3 &= (Z_1^2 + dX_0X_1Y_0Y_1)(Z_1^2 - dX_0X_1Y_0Y_1) \end{aligned}$$

[3] In computations we actually work with $l_1^{'}(x,y) = (xy)l_1(x,y)$ instead of $l_1(x,y)$. This trick has no effect on the final result and minimizes the number of operations done when evaluating at point T. On the other hand, when evaluating at point $Q+T$, whose x and y coordinates are both different from 0, we use l_1 and leave the xy factor aside. This will give a $(xy)^r$ factor in the end, which disappears when considering the reduced Tate pairing.

We compute these coordinates in a slightly different way from [3]. We suppose having precomputed X_1^2,Y_1^2 and Z_1^2 at the addition performed before in Miller's algorithm. We also suppose having precomputed X_0Y_0 once and for all in the very beginning.
Next, we consider $l_{s,p}$ and $v_{s,p}$, the straight lines passing through $K^{'} = \phi(K)$ and $P^{'} = \phi(P)$ and the vertical line passing through the point $\phi(K) + \phi(P)$. By putting $K^{'} = (s_1, p_1)$, $P^{'} = (s_0, p_0)$ and $\phi(T) + \phi(P) = (s_3, p_3)$ we get:

$$\begin{aligned} l_{s,p}(s,p) &= (p_0 - p_1)(s - s_1) - (s_0 - s_1)(p - p_1); \\ v_{s,p}(s,p) &= p - p_3. \end{aligned}$$

Consequently we get the following equations for pullbacks:

$$\begin{aligned} l(x,y) = l_1(x,y)/l_2 &= (X_1^2 + Y_1^2 - Z_1^2 - dZ_1^2(X_0Y_0)^2))(X_1Y_1(\frac{x}{y} - \frac{y}{x}) - (X_1^2 - Y_1^2)) \\ &\quad - (X_1^2 - Y_1^2 - X_1Y_1(\frac{X_0}{Y_0} - \frac{Y_0}{X_0})(dZ_1^2(xy)^2 - (X_1^2 + Y_1^2 - Z_1^2))/Z_1^4; \\ v(x,y) = v_1(x,y)/v_2 &= (dZ_3^2(xy)^2 - (X_3^2 + Y_3^2 - Z_3^2))/Z_3^2. \end{aligned}$$

Detailed computations of the mixed addition step are presented in table 3:

Table 3. Operations of the mixed addition step of a Miller operation

$A = X_1 \cdot Y_1, B = d \cdot A \cdot (X_0Y_0)$	(2**M**)
$C = (X_1 + X_0) \cdot (Y_0 + Y_1) - A - X_0Y_0; D = (X_1 + Y_0) \cdot (Y_1 - X_0) - A + X_0Y_0$	(2**M**)
$X_3 = Z_1 \cdot C \cdot (Z_1^2 + B), Y_3 = Z_1 \cdot (Z_1^2 + B) \cdot D, Z_3 = (Z_1^2 - B) \cdot (Z_1^2 + E)$	(5**M**)
$E = X_3^2, F = Y_3^2, G = Z_3^2$	(3**S**)
$H = dZ_1^2 \cdot (X_0Y_0)^2, I = X_1Y_1 \cdot (\frac{x}{y} - \frac{y}{x}) - (X_1^2 - Y_1^2), J = (X_1^2 + Y_1^2 - Z_1^2 - H) \cdot I$	(3**M**)
$K = X_1Y_1 \cdot (\frac{X_0}{Y_0} - \frac{Y_0}{X_0}), L = (X_1^2 - Y_1^2 - K) \cdot (dZ_1^2(xy)^2 - (X_1^2 + Y_1^2 - Z_1^2))$	(2**M**)
$l_1 = J - L, v_1 = dG \cdot (xy)^2 - (E + F - G)$	(1**M**)
$M = (X_1^2 + Y_1^2 - Z_1^2 - H) \cdot A$	(1**M**
$f_1 = f_1 \cdot l_1 \cdot (-E - F + G), f_2 = f_2 \cdot v_1 \cdot M$	(4**M**)

The operation count is presented in table 4.

Table 4. Comparison of costs in the case $k = 1$

	doubling step	mixed addition step
Jacobian coordinates	10**M**+10**S**	18**M**+3**S**
Edwards coordinates	12**M**+6**S**	20**M**+3**S**

4.2 The case of an even embedding degree

Koblitz and Menezes showed in [16] that if q and k are chosen such as $p \equiv 1 \pmod{12}$ and $k = 2^i 3^j$, then the arithmetic of the extension field F_{p^k} can be implemented very efficiently as this field can be built up as a tower of extension fields:

$$F_q \subset F_{q^{d_1}} \subset F_{q^{d_2}} ... \subset F_{q^k}.$$

where the ith field $F_{q^{d_i}}$ is obtained by adjoining a root of some irreducible polynomial $X^{d_i/d_{i-1}} - \beta_i$ and $d_i/d_{i-1} \in \{2,3\}$.

We note by $\mathbf{m}, \mathbf{M}$ (respectively $\mathbf{s}, \mathbf{S}$) the costs of multiplications (respectively squarings) in the field F_p and in extension F_{p^k}. Then according to [16] we get

$$\mathbf{M} \approx v(k)\mathbf{m} \text{ and } \mathbf{S} \approx v(k)\mathbf{s},$$

where $v(k) = 3^i 5^j$. Moreover, a multiplication of an element in F_{q^k} by an element in F_q costs $k\mathbf{m}$ operations. Suppose now that we want to compute the Tate pairing $T_r(P,Q)$.

In most cryptographic protocols there is some flexibility in the choice of the order r subgroups generated by P and Q. P can be chosen such that $< P >$ is the unique subgroup of order r in $E(F_q)$. Moreover, if the embedding degree is even, it was shown that the subgroup $< Q > \subset E(F_{q^k})$ can be taken so that the x-coordinates of all its points lie in $F_{q^{k/2}}$ and the y-coordinates are products of elements of $F_{q^{k/2}}$ with $\sqrt{\beta}$, where β is a nonsquare in $F_{q^{k/2}}$and $\sqrt{\beta}$ is a fixed squareroot in F_{q^k} (see [16] for details). Then the computational cost of the Tate pairing is significantly lower as we can ignore terms that lie in a proper subfield of F_{q^k}. These terms can be ignored because k is the multiplicative order of q modulo r, so $(q^k - 1)/r$ is a multiple of $q^{k'} - 1$ for some proper divisor k' of k. Besides it was shown in [1] that in this case the auxiliary point T can be ignored and the pairing value is given by $f_{r,P}(Q)^{(q^k-1)/r}$. Since the x-coordinate of Q and hence $v_1(Q)$ is in $F_{q^{k/2}}$ it follows that we can also ignore it. Hence the function evaluation step in the doubling part of Miller's algorithm 1 becomes

$$f_1 \leftarrow f_1^2 l_1(Q). \tag{13}$$

The same kind of considerations are to be done for Edwards coordinates. To do this we need to take a look at the birational map that transforms a curve of Weierstrass equation into a curve of Edwards equation. As stated in section 2.1 the curve $x^2 + y^2 = 1 + dx^2y^2$ is birationally equivalent to the curve of equation $(1/(1-d))v^2 = u^3 + 2((1+d)/(1-d))u^2 + u$, via the rational map $(u,v) \rightarrow (\frac{2u}{v}, \frac{u-1}{u+1})$.

It follows that in the case of an even embedding degree, the coordinates of elements of $< P >$ can be chosen in F_q. The subgroup $< Q > \in F_{q^k}$ can be chosen such as its elements have y-coordinates in the quadratic subextension $F_{q^{k/2}}$ and x-coordinates that can be written as products of elements of $F_{q^{k/2}}$ with some squareroot of a nonsquare element β of $F_{q^{k/2}}$. Hence we can do the same denominator elimination trick as above and the function evaluation step in in the doubling part of the Miller operation has the same form as the one in Jacobian coordinates of (13). Because some operations are done in F_{q^k} and others in F_q we compute l_1 as follows:

$$\begin{aligned} l_1(x,y) = {} & ((Y_3 - I)\cdot C)\cdot(x/y - y/x) - (Y_3 - I)\cdot 2D - Z_3\cdot dZ_1^2\cdot(xy)^2 \\ & + Z_3\cdot(X_1^2 + Y_1^2 - Z_1^2), \end{aligned}$$

The detailed computation of the doubling step is given in table 5.

Table 5. Operations of the doubling part of Miller's equation

$A = (X_1 + Y_1)^2,\ B = X_1^2 + Y_1^2,\ C = A - B,\ D = Y_1^2 - X_1^2$	$(1\mathbf{s})$
$E = 2Z_1^2 - B,\ X_3 = C\cdot E,\ Y_3 = B\cdot D,\ Z_3 = B\cdot E$	$(3\mathbf{m})$
$F = X_3^2;\ G = Y_3^2;\ H = Z_3^2; I = Z_1^2\cdot D.$	$(1\mathbf{m}+3\mathbf{s})$
$K = (Y_3 - I)\cdot C,\ L = K\cdot(x/y - y/x), M = (Y_3 - I)\cdot 2D,\ N = Z_3\cdot dZ_1^2$	$(3\mathbf{m}+k/2\mathbf{m})$
$O = M\cdot(xy)^2, P = Z_3\cdot(X_1^2 + Y_1^2 - Z_1^2), l_1 = L - M - O + P,$	$(1\mathbf{m}+k/2\mathbf{m})$
$X_1 = X_3, Y_1 = Y_3, Z_1 = Z_3,$	
$f_1 = f_1^2 l_1$	$(1\mathbf{M}+1\mathbf{S})$

Results are summarized in the table 6.

Table 6. Comparison of costs for the doubling step of the Miller operation in the case of k even

	$k = 2$	$k \geq 4$
Jacobian coordinates	$6\mathbf{s} + 7\mathbf{m} + \mathbf{S} + \mathbf{M}$	$6\mathbf{s} + (k+6)\mathbf{m} + \mathbf{S} + \mathbf{M}$
Jacobian coordinates for $a = -3$	$4\mathbf{s} + 8\mathbf{m} + \mathbf{S} + \mathbf{M}$	$4\mathbf{s} + (k+7)\mathbf{m} + \mathbf{S} + \mathbf{M}$
Das/Sarkar Edwards coordinates (supersingular curves)	$6\mathbf{s} + 9\mathbf{m} + \mathbf{S} + \mathbf{M}$	-
Edwards coordinates	$4\mathbf{s} + 10\mathbf{m} + \mathbf{S} + \mathbf{M}$	$4\mathbf{s} + (k+8)\mathbf{m} + \mathbf{S} + \mathbf{M}$

We can see that in the case of an even embedding degree the cost of an implementation of Miller's algorithm in Edwards coordinates will be comparable to the cost of an implementation in Jacobian coordinates. We find it important to state that, no matter the representation one might choose to implement Miller's algorithm in high embedding degrees, it would be impossible to avoid the costly computation of $\mathbf{M}+\mathbf{S}$ in equation (13), as the final result needs to be an element of F_{q^k}.

For the same reasons as above, the mixed addition in equation (12) becomes:

$$f_1 \to f_1 \cdot l_1(Q) \tag{14}$$

The operation count is quite similar to the one in table 3, so we do not detail it here. Results and performance comparison are presented in the table 7.

Table 7. Comparison of costs for the mixed addition step of the Miller operation in the case of k even

	$k = 2$	$k \geq 4$
Jacobian coordinates	$3\mathbf{s} + 15\mathbf{m} + \mathbf{M}$	$3\mathbf{s} + (k + 13)\mathbf{m} + 1\mathbf{M}$
Das/Sarkar Edwards coordinates (supersingular curves)	$1\mathbf{s} + 18\mathbf{m} + \mathbf{M}$	-
Edwards coordinates	$3\mathbf{s} + 15\mathbf{m} + \mathbf{M}$	$3\mathbf{s} + (k + 13)\mathbf{m} + 1\mathbf{M}$

5 Conclusion

In this paper, we have given a new algorithm to compute pairings on Edwards curves and compared its performance to that of an implementation of Miller's algorithm in Jacobian coordinates and to the method for Edwards coordinates from [10]. Our count shows that our algorithm is slightly slower than Jacobian coordinates for the most frequently used case of curves with an even embedding degree and faster than the previously known approach for Edwards coordinates. Moreover, in the special case of embedding degree one, it becomes faster than Miller's algorithm using Jacobian coordinates.

We did not include in our analysis the case of curves of odd embedding degree. Although examples of such curves do exist in literature (see [9] and [12]), they are less used in practice. We expect similar results to hold in such cases.

References

1. Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. On the selection of pairing-friendly groups. In Mitsuru Matsui and Robert J. Zuccherato, editors, *SAC 2003: 10th Annual International Workshop on Selected Areas in Cryptography*, volume 3006 of *Lecture Notes in Computer Science*, pages 17–25. Springer Verlag, 2004.
2. Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. Twisted edwards curves. In Serge Vaudenay, editor, *AFRICACRYPT*, volume 5023 of *Lecture Notes in Computer Science*, pages 389–405. Springer Verlag, 2008.
3. Daniel J. Bernstein and Tanja Lange. Faster addition and doubling on elliptic curves. In Kaoru Kurosawa, editor, *Advances in Cryptology – ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 29–50. Springer Verlag, 2007.
4. Ian F. Blake, Gadiel Seroussi, and Nigel P. Smart. *Advances in Elliptic Curve Cryptography*, volume 317 of *London Mathetical Society Lecture Note Series*. Cambridge university press, 2005.
5. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer Verlag, 2001.
6. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–532. Springer Verlag, 2001.
7. Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick McDaniel, editors, *ACM CCS 04: 11th Conference on Computer and Communications Security*, pages 168–177. ACM Press, 2004.
8. Wieb Bosma. Signed bits and fast exponentiation. *J. de théorie des nombres de Bordeaux*, 13(1):27–41, 2001.
9. Friederike Brezing and Annegret Weng. Elliptic curves suitable for pairing based cryptography. *Des. Codes Cryptography*, 37(1):133–141, 2005.
10. M. Prem Laxman Das and Palash Sarkar. Pairing computation on twisted Edwards form elliptic curves. In Steven Galbraith and Kenny Paterson, editors, *Pairing*, Lecture Notes in Computer Science. Springer Verlag, 2008. To appear.
11. Harold M. Edwards. A normal form for elliptic curves. *Bull. AMS*, 44:393–422, 2007.
12. David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. Cryptology ePrint Archive, Report 2006/372, 2006. http://eprint.iacr.org/.
13. Robert Granger, Dan Page, and Nigel P. Smart. High security pairing-based cryptography revisited. In Florian Hess, Sebastian Pauli, and Michael E. Pohst, editors, *ANTS*, volume 4076 of *Lecture Notes in Computer Science*, pages 480–494, 2006.
14. Robin Hartshorne. *Algebraic Geometry*, volume 52 of *Graduate texts in Mathematics*. Springer, 1977.
15. Antoine Joux. A one round protocol for tripartite Diffie-Hellman. *Journal of Cryptology*, 17(4):263–276, September 2004.
16. Neal Koblitz and Alfred Menezes. Pairing-based cryptography at high security levels. In Nigel P. Smart, editor, *IMA Int. Conf.*, volume 3796 of *Lecture Notes in Computer Science*, pages 13–36, 2005.
17. Victor S. Miller. The Weil pairing, and its efficient calculation. *Journal of Cryptology*, 17(4):235–261, September 2004.

A Pairing computation in Jacobian coordinates

We begin with the doubling step of the Miller operation for the bit i. We write the functions l and v that appear in (8) as $l = l_1/l_2$ and $v = v_1/v_2$ (we will see that the denominators are constant). So in the double and add method, after initially setting $K = P$ and $f_1 = f_2 = 1$, we have to do the following evaluations for the ith bit of r:

$$\begin{aligned} K &\leftarrow 2K \\ f_1 &\leftarrow f_1^2 l_1(T+Q)v_1(T) \\ f_2 &\leftarrow f_2^2 l_1(T)v_1(Q+T). \end{aligned} \tag{15}$$

A doubling of the point $K = (X_1, Y_1, Z_1)$ takes the form $2K = (X_3, Y_3, Z_3)$ with

$$\begin{aligned} X_3 &= (3X_1^2 + aZ_1^4)^2 - 8X_1Y_1^2, \\ Y_3 &= (3X_1^2 + aZ_1^4)(4X_1Y_1^2 - X_3) - 8Y_1^4, \\ Z_3 &= 2Y_1Z_1. \end{aligned}$$

The functions l and v, corresponding respectively to the tangent line to the curve at K and the vertical line through the point $2K$ have the following equations:

$$\begin{aligned} l(x,y) &= l_1(x,y)/l_2 = (Z_3Z_1^2y - 2Y_1^2 - (3X_1^2 + aZ_1^4)(xZ_1^2 - X_1))/(Z_3Z_1^2) \\ v(x,y) &= v_1(x,y)/v_2 = (Z_3^2x - X_3)/Z_3^2. \end{aligned}$$

As we need to evaluate l and v at $Q + T$ and T, the constant denominators l_2 and v_2 will cancel. According to [16], we suppose $T = (0, 0)$ in order to minimize computations. In the operation count we do not take into consideration the computations of Z_1^2 and $Z_1^2x - X_1$ as they have already been done in the computation of Z_3^2 and $Z_3^2x - X_3$ at the point addition executed just before. Obviously, if the $(i + 1)$th bit was a 1, the addition before was not a doubling, but a mixed addition of K and P. The operations to be performed are described in table A. We count 10**M**+10**S** in all.

Table 8. Operations of the doubling part of a Miller operation

$A = (Z_1^2)^2, B = X_1^2, C = Y_1^2, D = C^2$	(4**S**)
$E = (X_1 + C)^2 - B - D, F = 3B + aA$	(1**S**)
$X_3 = -2E + F^2; Y_3 = -8D + F \cdot (E - X_3); Z_3 = (Y_1 + Z_1)^2 - C - Z_1^2$	(1**M**+2**S**)
$G = Z_3 \cdot Z_1^2, H = G \cdot y, I = F \cdot (Z_1^2x - X_1)$	(3**M**)
$J = Z_3^2, v_1 = J \cdot x - X_3, K = F \cdot X_1$	(2**M**+1**S**)
$f_1 = f_1^2 \cdot l_1 \cdot (-X_3)$	(2**M**+1**S**)
$f_1 = f_1^2 \cdot v_1 \cdot K$	(2**M**+1**S**)

The mixed addition step implies the following operations:

$$
\begin{aligned}
K &\leftarrow K + P \\
f_1 &\leftarrow f_1 l_1(T+Q) v_1(T) \qquad (16) \\
f_2 &\leftarrow f_2 l_1(T) v_1(T+Q)
\end{aligned}
$$

$$(17)$$

When adding $K = (X_1, Y_1, Z_1)$ and $P = (X_0, Y_0, 1)$ the result takes this form $K + P = (X_3, Y_3, Z_3)$ with

$$
\begin{aligned}
X_3 &= (X_1 + X_0 Z_1^2)(X_1 - X_0 Z_1^2)^2 + (Y_0 Z_1^3 - Y_1)^2, \\
Y_3 &= (Y_0 Z_1^3 - Y_1)(X_1(X_1 - X_0 Z_1^2)^2 - X_3) + Y_1(X_1 - X_0 Z_1^2)^2, \\
Z_3 &= Z_1(X_0 Z_1^2 - X_1).
\end{aligned}
$$

The functions l and v have the following equations:

$$
\begin{aligned}
l(x,y) = l_1(x,y)/l_2 &= (X_1 - X_0 Z_1^2)(Z_1^3 y - Y_1) - (Y_0 Z_1^3 - Y_1)(Z_1^2 x - X_1)/Z_1^2 Z_3, \\
v(x,y) = v_1(x,y)/v_2 &= (Z_3^2 x - X_3)/Z_3^2.
\end{aligned}
$$

Computations are detailed in the table A. We count 18**M**+3**S** in all.

Table 9. Operations of the mixed addition step in a Miller operation

$A = X_0 \cdot Z_1^2, B = (X_1 - A)^2, C = Z_1^2 \cdot Z_1$	(2**M**+1**S**)
$D = Y_0 \cdot C, E = (D - Y_1)^2, F = X_1 \cdot B$	(2**M**+1**S**)
$G = Y_1 \cdot B, X_3 = (X_1 + A) \cdot B + E, Y_3 = (D - Y_1) \cdot (F - X_3) + G$	(3**M**)
$Z_3 = Z_1 \cdot (A - X_1), H = C \cdot y, I = Z_3^2, J = I \cdot x$	(3**M**+1**S**)
$K = Y_1 \cdot (X_1 - A), L = X_1 \cdot (D - Y_1)$	(2**M**)
$l_1 = (X_1 - A) \cdot (H - Y_1) - (D - Y_1) \cdot (Z_1^2 x - X_1), v_1 = J - X_3$	(2**M**)
$f_1 = f_1 \cdot l_1 \cdot (-X_3)$	(2**M**)
$f_1 = f_1 \cdot v_1 \cdot (-K - L)$	(2**M**)