Faster Pairing Computation

Christophe Arène[1], Tanja Lange[2], Michael Naehrig[2], and Christophe Ritzenthaler[1]

[1] Institut de mathématiques de Luminy
Université de la Méditerranée
163 Avenue de Luminy Case 907, 13288 Marseille, France
{arene, ritzenth}@iml.univ-mrs.fr
[2] Department of Mathematics and Computer Science
Technische Universiteit Eindhoven, P.O. Box 513, 5600 MB Eindhoven, Netherlands
tanja@hyperelliptic.org, michael@cryptojedi.org

Abstract. This paper proposes new explicit formulas for the doubling and addition step in Miller's algorithm to compute pairings.
For Edwards curves the formulas come from a new way of seeing the arithmetic. We state the first geometric interpretation of the group law on Edwards curves by presenting the functions which arise in the addition and doubling. Computing the coefficients of the functions and the sum or double of the points is faster than with all previously proposed formulas for pairings on Edwards curves. They are even competitive with all published formulas for pairing computation on Weierstrass curves. We also speed up pairing computation on Weierstrass curves in Jacobian coordinates.
Finally, we present examples of pairing-friendly twisted Edwards curves with embedding degree $k = 6$.
Keywords: Pairing, Miller function, explicit formulas, Edwards curves.

1 Introduction

Since their introduction to cryptography by Bernstein and Lange [8], Edwards curves have received a lot of attention because of their very fast group law. The group law in affine form was introduced by Edwards in [14] along with a description of the curve and several proofs of the group law. Remarkably none of the proofs provided a geometric interpretation of the group law while for elliptic curves in Weierstrass form the explanation via the chord-and-tangent method is the standard.

Applications in discrete-logarithm-based systems such as Diffie-Hellman key exchange or digital signatures require efficient computation of scalar multiples and thus have benefited from the speedup in addition and doubling. The situation is significantly different in pairing-based cryptography where Miller's algorithm needs a function whose divisor is $(P) + (Q) - (P + Q) - (\mathcal{O})$, for two

* This work has been supported in part by the European Commission through the ICT Programme under Contract ICT–2007–216646 ECRYPT II, and in part by grant MTM2006-11391 from the Spanish MEC.

input points P and Q and their sum $P+Q$. For curves in Weierstrass form these functions are readily given by the line functions in the usual addition and doubling. Edwards curves have degree 4 and thus any line passes through 4 points instead of 3. This led many to conclude that Edwards curves provide no benefit to pairings and are doomed to be slower than the Weierstrass counterparts.

So far two papers have attempted to compute pairings efficiently on Edwards curves: Das and Sarkar [13] use the birational equivalence to Weierstrass curves to map the points on the Edwards curve to a Weierstrass curve on which the usual line functions are then evaluated. This approach comes at a huge performance penalty as these implicit pairing formulas need many more field operations to evaluate them. Das and Sarkar then focus on supersingular curves with embedding degree $k = 2$ and develop explicit formulas for that case.

Ionica and Joux [21] use a different map to a curve of degree 3 and compute the 4-th power of the Tate pairing. The latter poses no problem in usage in protocols as long as both sides perform the same type of pairing computation. Their results are significantly faster than Das and Sarkar's but they are still much slower than pairings on Weierstrass curves.

In this paper we close several important gaps:

- We provide a geometric interpretation of the Edwards addition law for twisted Edwards curves.
- We study additions, doublings, and all the special cases that appear as part of the geometric addition law for twisted Edwards curves.
- We use the geometric group law to show how to compute the Tate pairing on twisted Edwards curves.
- We give examples of pairing-friendly Edwards and twisted Edwards curves.

Beyond that, we develop explicit formulas for computing pairings on Edwards and twisted Edwards curves that for Edwards curves

- solidly beat the results by Das–Sarkar [13] and Ionica–Joux [21];
- are as fast as the fastest published formulas for the doubling step on Weierstrass curves, namely curves with $a_4 = 0$ (e.g. Barreto-Naehrig curves) in Jacobian coordinates, and faster than other Weierstrass curves;
- need the same number of field operations as the best published formulas for mixed addition in Jacobian coordinates; but need more multiplications and fewer squaring;
- have minimal performance penalty for non-affine base points.

In particular, for even embedding degree k the doubling step on an Edwards curve takes $(k + 6)\mathbf{m} + 5\mathbf{s} + 1\mathbf{M} + 1\mathbf{S}$, where $\mathbf{m}$ and $\mathbf{s}$ denote the costs of multiplication and squaring in the base field while $\mathbf{M}$ and $\mathbf{S}$ denote the costs of multiplication and squaring in the extension field of degree k. A mixed addition step takes $(k + 12)\mathbf{m} + 1\mathbf{M}$ and an addition step takes $(k + 14)\mathbf{m} + 1\mathbf{M}$.

We also speed up the addition and doubling step on Weierstrass curves for all curve shapes by trading several multiplications for squarings and additions. We present the first efficient formulas for (non-mixed) addition steps on Weierstrass curves.

2 Twisted Edwards Curves

In this section and the next one, K will denote a field of characteristic different from 2. A *twisted Edwards curve* over K is a curve given by an affine equation of the form

$$\mathrm{E}_{a,d} : ax^2 + y^2 = 1 + dx^2y^2$$

for $a, d \in K^*$ and $a \neq d$. They were introduced by Bernstein et al. in [6] as a generalization of Edwards curves [8] which are included as $\mathrm{E}_{1,d}$. There is an addition law on points of the curve $\mathrm{E}_{a,d}$ which is given by

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

The neutral element is $\mathcal{O} = (0, 1)$, and the negative of (x_1, y_1) is $(-x_1, y_1)$. The point $\mathcal{O}' = (0, -1)$ has order 2. The points at infinity $\Omega_1 = (1 : 0 : 0)$ and $\Omega_2 = (0 : 1 : 0)$ are singular and blow up to two points each.

The name twisted Edwards curves comes from the fact that the set of twisted Edwards curves is invariant under quadratic twists while a quadratic twist of an Edwards curve is not necessarily an Edwards curve. In particular, let $\delta \in K \setminus K^2$ and let $\alpha^2 = \delta$ for some α in a quadratic extension K_2 of K. The map $\epsilon : (x, y) \mapsto (\alpha x, y)$ defines a K_2-isomorphism between the twisted Edwards curves $\mathrm{E}_{a,d}$ and $\mathrm{E}_{a/\delta,d/\delta}$. Hence, the map ϵ is the prototype of a quadratic twist. Note that twists change the x-coordinate unlike on Weierstrass curves where they affect the y-coordinate.

3 Geometric Interpretation of the Group Law

In this section, we study the intersection of $\mathrm{E}_{,a,d}$ with certain plane curves and explain the Edwards addition law in terms of the divisor class arithmetic. We remind the reader that the divisor class group is defined as the group of degree-0 divisors modulo the group of principal divisors in the function field of the curve, i.e. two divisors are *equivalent* if they differ by a principal divisor. For background reading on curves and Jacobians, we refer to [15] and [31].

We first consider projective lines in $\mathbb{P}^2$. A general line is of the form

$$L : c_X X + c_Y Y + c_Z Z = 0, \tag{1}$$

where $(c_X : c_Y : c_Z) \in \mathbb{P}^2$. A line is uniquely determined by two of its points when they are distinct. We first consider lines which pass through one of the points at infinity and an affine point P. Note that the line through Ω_1 and Ω_2 is the line at infinity $L_\infty : Z = 0$.

Lemma 1. *Let $P = (X_0 : Y_0 : Z_0) \in \mathbb{P}^2(K)$ be an affine point, i. e. $Z_0 \neq 0$, and $L_{1,P}$ be the projective line passing through P and Ω_1. Then $L_{1,P}$ is given by*

$$L_{1,P} : Z_0Y - Y_0Z = 0.$$

Let $L_{2,P}$ be the line through P and Ω_2. Then $L_{2,P}$ is given by

$$L_{2,P} : Z_0 X - X_0 Z = 0.$$

Proof. The points at infinity force $c_X = 0$ and $c_Y = 0$, respectively. The equation then follows from P being a point on the line. □

In the following we describe a special conic which passes through both points at infinity, Ω_1 and Ω_2, the point $\mathcal{O}'$, and two arbitrary affine points P_1 and P_2 on $\mathrm{E}_{a,d}$. Let $\phi(X,Y,Z) = c_{X^2}X^2 + c_{Y^2}Y^2 + c_{Z^2}Z^2 + c_{XY}XY + c_{XZ}XZ + c_{YZ}YZ \in K[X,Y,Z]$ be a homogeneous polynomial of degree 2 and $C : \phi(X,Y,Z) = 0$, the associated plane (possibly degenerate) conic.

Remark 2. Since the points $\Omega_1, \Omega_2, \mathcal{O}'$ are not on a line, the conic C cannot be a double line and ϕ represents C uniquely up to multiplication by a scalar.

Lemma 3. *If a conic C passes through the points Ω_1, Ω_2, and $\mathcal{O}'$, then it has an equation of the form*

$$C : c_{Z^2}(Z^2 + YZ) + c_{XY}XY + c_{XZ}XZ = 0, \tag{2}$$

where $(c_{Z^2} : c_{XY} : c_{XZ}) \in \mathbb{P}^2(K)$.

Proof. We evaluate ϕ at the three points Ω_1, Ω_2, and $\mathcal{O}'$. The fact that Ω_1 lies on the conic, means that $c_{X^2} = 0$. Similarly, $c_{Y^2} = 0$ since Ω_2 lies on C. Further, the condition $\mathcal{O}' \in C$ shows $c_{YZ} = c_{Z^2}$. □

Theorem 4. *Let $\mathrm{E}_{a,d}$ be a twisted Edwards curve over K, and let $P_1 = (X_1 : Y_1 : Z_1)$ and $P_2 = (X_2 : Y_2 : Z_2)$ be two affine, not necessarily distinct, points on $\mathrm{E}_{a,d}(K)$. Let C be the conic passing through Ω_1, Ω_2, $\mathcal{O}'$, P_1, and P_2, i. e. C is given by an equation of the form (2). If some of the above points are equal, we consider C and $\mathrm{E}_{a,d}$ to intersect with at least that multiplicity at the corresponding point. Then the coefficients in (2) of the equation ϕ of the conic C are uniquely (up to scalars) determined as follows:*

(a) If $P_1 \neq P_2$, $P_1 \neq \mathcal{O}'$ and $P_2 \neq \mathcal{O}'$, then

$$\begin{aligned} c_{Z^2} &= X_1X_2(Y_1Z_2 - Y_2Z_1), \\ c_{XY} &= Z_1Z_2(X_1Z_2 - X_2Z_1 + X_1Y_2 - X_2Y_1), \\ c_{XZ} &= X_2Y_2Z_1^2 - X_1Y_1Z_2^2 + Y_1Y_2(X_2Z_1 - X_1Z_2). \end{aligned}$$

(b) If $P_1 \neq P_2 = \mathcal{O}'$, then

$$c_{Z^2} = -X_1,\ c_{XY} = Z_1,\ c_{XZ} = Z_1.$$

(c) If $P_1 = P_2$, then

$$\begin{aligned} c_{Z^2} &= X_1Z_1(Z_1 - Y_1), \\ c_{XY} &= dX_1^2Y_1 - Z_1^3, \\ c_{XZ} &= Z_1(Z_1Y_1 - aX_1^2). \end{aligned}$$

Proof. As for Lemma 3, if the points are distinct, the coefficients are obtained by evaluating the previous equation at the points P_1 and P_2. We obtain two linear equations in c_{Z^2}, c_{XY}, and c_{XZ}

$$c_{Z^2}(Z_1^2 + Y_1Z_1) + c_{XY}X_1Y_1 + c_{XZ}X_1Z_1 = 0,$$
$$c_{Z^2}(Z_2^2 + Y_2Z_2) + c_{XY}X_2Y_2 + c_{XZ}X_2Z_2 = 0.$$

The formulas in (a) follow from the (projective) solutions

$$c_{Z^2} = \begin{vmatrix} X_1Y_1 & X_1Z_1 \\ X_2Y_2 & X_2Z_2 \end{vmatrix}, \; c_{XY} = \begin{vmatrix} X_1Z_1 & Z_1^2 + Y_1Z_1 \\ X_2Z_2 & Z_2^2 + Y_2Z_2 \end{vmatrix}, \; c_{XZ} = \begin{vmatrix} Z_1^2 + Y_1Z_1 & X_1Y_1 \\ Z_2^2 + Y_2Z_2 & X_2Y_2 \end{vmatrix}.$$

If $P_1 = P_2 \neq \mathcal{O}'$, we start by letting $Z_1 = 1, Z = 1$ in the equations. The tangent vectors at the non singular point $P_1 = (X_1 : Y_1 : 1)$ of $\mathrm{E}_{a,d}$ and of C are

$$\begin{pmatrix} dX_1^2Y_1 - Y_1 \\ aX_1 - dX_1Y_1^2 \end{pmatrix}, \quad \begin{pmatrix} -c_{Z^2} - c_{XY}X_1 \\ c_{XY}Y_1 + c_{XZ} \end{pmatrix}.$$

They are collinear if the determinant of their coordinates is zero which gives us a linear condition in the coefficients of ϕ. We get a second condition by $\phi(X_1, Y_1, 1) = 0$. Solving the linear system, we get the projective solution

$$\begin{aligned}
c_{Z^2} &= X_1^3(-dY_1^2 + a) = X_1(1 - Y_1^2) = X_1(Y_1 + 1)(1 - Y_1),\\
c_{XY} &= 2dX_1^2Y_1^2 - Y_1 - Y_1^2 + dX_1^2Y_1 - aX_1^2\\
&= -1 - Y_1 + dX_1^2Y_1^2 + dX_1^2Y_1 = (Y_1 + 1)(dX_1^2Y_1 - 1),\\
c_{XZ} &= -dX_1^2Y_1^3 - aX_1^2 + Y_1^2 + Y_1^3 = (Y_1 + 1)(Y_1 - aX_1^2)
\end{aligned}$$

using the curve equation $aX_1^2 + Y_1^2 = 1 + dX_1^2Y_1^2$ to simplify. Finally, since $P_1 \neq \mathcal{O}'$, we can divide by $1+Y_1$ and homogenize to get the result which provides the formulas as stated. The same formulas hold if $P_1 = \mathcal{O}'$ since intersection multiplicity greater than or equal to 3 at $\mathcal{O}'$ is achieved by setting $\phi = X(Y + Z) = XY + XZ$.

Assume now that $P_1 \neq P_2 = \mathcal{O}'$. Note that the conic C is tangent to $\mathrm{E}_{a,d}$ at $\mathcal{O}'$ if and only if $(\partial\phi/\partial x)(0, -1, 1) = (c_{XY}y + c_{XZ}z)(0, -1, 1) = 0$, i.e. $c_{XY} = c_{XZ}$. Then $\phi = (Y + Z)(c_{Z^2}Z + c_{XY}X)$. Since $P_1 \neq \mathcal{O}'$, it is not on the line $Y + Z = 0$. Then we get $c_{Z^2}Z_1 + c_{XY}X_1 = 0$ which gives the coefficients as in (b). □

Let P_1 and P_2 be two affine K-rational points on a twisted Edwards curve $\mathrm{E}_{a,d}$, and let $P_3 = (X_3 : Y_3 : Z_3) = P_1 + P_2$ be their sum. Let

$$l_1 = Z_3Y - Y_3Z, \quad l_2 = X$$

be the polynomials of the horizontal line L_{1,P_3} and the vertical line $L_{2,\mathcal{O}}$ respectively, (cf. Lemma 1) and let

$$\phi = c_{Z^2}(Z^2 + YZ) + c_{XY}XY + c_{XZ}XZ$$

be the unique polynomial (up to multiplication by a scalar) defined by Theorem 4.
The following theorem shows that the twisted Edwards group law indeed has a geometric interpretation involving the above equations. It gives us an important ingredient to compute Miller functions.

Theorem 5. *Let $a, d \in K^*$, $a \neq d$ and $\mathrm{E}_{a,d}$ be a twisted Edwards curve over K. Let $P_1, P_2 \in \mathrm{E}_{a,d}(K)$. Define $P_3 = P_1 + P_2$. Then we have*

$$\mathrm{div}\left(\frac{\phi}{l_1 l_2}\right) \sim (P_1) + (P_2) - (P_3) - (\mathcal{O}). \tag{3}$$

Proof. Let us consider the intersection divisor $(C \cdot \mathrm{E}_{a,d})$ of the conic $C : \phi = 0$ and the singular quartic $\mathrm{E}_{a,d}$. Bezout's Theorem [16, p. 112] tells us that the intersection of C and $\mathrm{E}_{a,d}$ should have $2 \cdot 4 = 8$ points counting multiplicities over $\overline{K}$. We note that the two points at infinity Ω_1 and Ω_2 are singular points of multiplicity 2. Moreover, by definition of the conic C, $(P_1) + (P_2) + (\mathcal{O}') + 2(\Omega_1) + 2(\Omega_2) \leq (C \cdot \mathrm{E}_{a,d})$. Hence there is an 8th point Q in the intersection. Let $L_{1,Q} : l_Q = 0$ be the horizontal line going through Q. Since the inverse for addition on twisted Edwards curves is given by $(x, y) \mapsto (-x, y)$, we see that $(L_{1,Q} \cdot \mathrm{E}_{a,d}) = (Q) + (-Q) + 2(\Omega_2)$. On the other hand $(L_{2,\mathcal{O}} \cdot \mathrm{E}_{a,d}) = (\mathcal{O}) + (\mathcal{O}') + 2(\Omega_1)$. Hence by combining the above divisors we get

$$\mathrm{div}\left(\frac{\phi}{l_Q l_2}\right) \sim (P_1) + (P_2) - (-Q) - (\mathcal{O}).$$

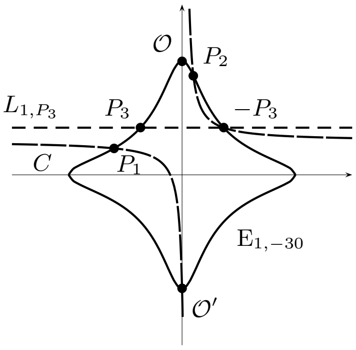
By unicity of the group law with neutral element $\mathcal{O}$ on the elliptic curve $\mathrm{E}_{a,d}$ [31, Prop.3.4], the last equality means that $P_3 = -Q$. Hence $(L_{1,P_3} \cdot \mathrm{E}_{a,d}) = (P_3) + (-P_3) + 2(\Omega_2) = (-Q) + (Q) + 2(\Omega_2)$ and $l_1 = l_Q$. So

$$\mathrm{div}\left(\frac{\phi}{l_1 l_2}\right) \sim (P_1) + (P_2) - (P_3) - (\mathcal{O}).$$

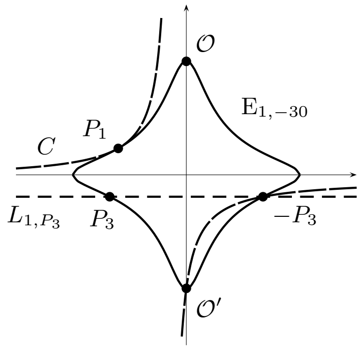
□

Remark 6. From the proof, we see that $P_1 + P_2$ is obtained as the mirror image with respect to the y-axis of the eighth intersection point of $\mathrm{E}_{a,d}$ and the conic C passing through $\Omega_1, \Omega_2, \mathcal{O}', P_1$ and P_2.

Example 7. As an example we consider the Edwards curve $\mathrm{E}_{1,-30} : Z^2(X^2 + Y^2) = Z^4 - 30X^2Y^2$ over the set of real numbers $\mathbb{R}$. Of course, the pictures show the affine part of the curve, dehomogenized by setting $Z = 1$. We choose the point P_1 with x-coordinate $x_1 = -0.6$ and P_2 with x-coordinate $x_2 = 0.1$. Figure 1(a) shows addition of different points P_1 and P_2, and Figure 1(b) shows doubling of the point P_1.



(a) $P_1 \neq P_2$, $P_1, P_2 \neq \mathcal{O}'$, $P_3 = P_1 + P_2$



(b) $P_1 = P_2 \neq \mathcal{O}'$, $P_3 = 2P_1$

Fig. 1. Geometric interpretation of the Edwards group law on $\mathrm{E}_{1,-30} : x^2 + y^2 = 1 - 30x^2y^2$ over $\mathbb{R}$.

4 Background on Pairings

Let p be a prime different from 2 and let $E/\mathbf{F}_p$ be an elliptic curve over $\mathbf{F}_p$ with neutral element denoted by $\mathcal{O}$. Let $n \mid \#E(\mathbf{F}_p)$ be a prime divisor of the group order and let E have embedding degree k with respect to n. For simplicity and speed we assume that $k > 1$.

Let $P \in E(\mathbf{F}_p)[n]$ and let $f_P \in \mathbf{F}_p(E)$ be such that $\mathrm{div}(f_P) = n(P) - n(\mathcal{O})$ and let $\mu_n \subset \mathbf{F}^*_{p^k}$ denote the group of n-th roots of unity. The reduced Tate pairing is given by

$$T_n : E(\mathbf{F}_p)[n] \times E(\mathbf{F}_{p^k})/nE(\mathbf{F}_{p^k}) \to \mu_n;\ (P,Q) \mapsto f_P(Q)^{(p^k-1)/n}.$$

Miller [24] suggested to compute pairings in an iterative manner. Let $n = (n_{l-1}, \ldots, n_1, n_0)_2$ be the binary representation of n and let $g_{R,P} \in \mathbf{F}_p(E)$ be the function arising in addition on E such that $\mathrm{div}(g_{R,P}) = (R)+(P)-(R+P)-\mathcal{O}$, where $\mathcal{O}$ denotes the neutral element in the group of points and $R + P$ denotes the sum of R and P on E while additions of the form $(R) + (P)$ denote formal additions in the divisor group. Miller's algorithm starts with $R = P, f = 1$ and computes

1. for $i = l - 2$ to 0 do
   (a) $f \leftarrow f^2 \cdot g_{R,R}(Q),\ R \leftarrow 2R$ //doubling step
   (b) if $n_i = 1$ then $f \leftarrow f \cdot g_{R,P}(Q),\ R \leftarrow R + P$ //addition step
2. $f \leftarrow f^{(p^k-1)/n}$

For Weierstrass curves and even k, several improvements and speedups are presented in [3] and [4]. In particular it is common to eliminate all denominators by choosing the second point Q such that its x-coordinate is in a subfield of $\mathbf{F}_{p^k}$. The functions $g_{R,P}$ are defined over $\mathbf{F}_p$ and their denominators are functions in x only. Writing $g_{R,P}(Q) = h_{R,P}(x_Q, y_Q)/s_{R,P}(x_Q)$ with polynomial functions

$h_{R,P}$ and $s_{R,P}$, one sees that the complete contribution of all $s_{R,P}(x_Q)$'s will be mapped to 1 by the final exponentiation if x_Q is in a proper subfield of $\mathbf{F}_{p^k}$. The latter is usually enforced by choosing a point Q' on a quadratic twist of E over $\mathbf{F}_{p^{k/2}}$ and defining Q as the image of Q' under the twist.

5 Miller Functions on Edwards Curves

In this section we show how to use the geometric interpretation of the group law derived in Section 3 to compute pairings. We assume that k is even and that the second input point Q is chosen by using the tricks in [3] and [4]. Note that, as explained in Section 2, on twisted Edwards curves $\mathrm{E}_{a,d}$, twists affect the x-coordinate. Let $\mathbf{F}_{p^k}$ have basis $\{1, \alpha\}$ over $\mathbf{F}_{p^{k/2}}$ with $\alpha^2 = \delta \in \mathbf{F}_{p^{k/2}}$ and let $Q' = (X_0 : Y_0 : Z_0) \in \mathrm{E}_{a\delta,d\delta}(\mathbf{F}_{p^{k/2}})$. Twisting Q' with α ensures that the second argument of the pairing is on $\mathrm{E}_{a,d}(\mathbf{F}_{p^k})$ (and no smaller field) and is of the form $Q = (X_0\alpha : Y_0 : Z_0)$, where $X_0, Y_0, Z_0 \in \mathbf{F}_{p^{k/2}}$.

By Theorem 5 we have $g_{R,P} = \frac{\phi}{l_1 l_2}$. So the update in the Miller loop computes $g_{R,P}$, evaluates it at $Q = (X_0\alpha : Y_0 : Z_0)$ and updates f as $f \leftarrow f \cdot g_{R,P}(Q)$ (addition) or as $f \leftarrow f^2 \cdot g_{R,R}(Q)$ (doubling). Given the shape of ϕ and the point $Q = (X_0\alpha : Y_0 : Z_0)$, we see that we need to compute

$$\begin{aligned}\frac{\phi}{l_1 l_2}(X_0\alpha : Y_0 : Z_0) &= \frac{c_{Z^2}(Z_0^2 + Y_0 Z_0) + c_{XY} X_0 \alpha Y_0 + c_{XZ} X_0 Z_0 \alpha}{(Z_3 Y_0 - Y_3 Z_0) X_0 \alpha} \\ &= \frac{c_{Z^2} \frac{Z_0 + Y_0}{X_0 \delta} \alpha + c_{XY} y_0 + c_{XZ}}{Z_3 y_0 - Y_3},\end{aligned}$$

where $(X_3 : Y_3 : Z_3)$ are coordinates of the point $R+P$ or $R+R$ and $y_0 = Y_0/Z_0$. Define $\eta = \frac{Z_0+Y_0}{X_0\delta}$. Note that $\eta \in \mathbf{F}_{p^{k/2}}$ and that it is fixed for the whole computation, so it can be precomputed. The denominator $Z_3 y_0 - Y_3$ is defined over $\mathbf{F}_{p^{k/2}}$; since it enters the function multiplicatively, the final exponentiation will remove all contributions from it. We can thus avoid its computation completely, and only have to evaluate

$$c_{Z^2}\eta\alpha + c_{XY} y_0 + c_{XZ}.$$

The coefficients c_{Z^2}, c_{XY}, and c_{XZ} are defined over $\mathbf{F}_p$, so the evaluation at Q given the coefficients of the conic can be computed in $k\mathbf{m}$ (the multiplication by η and y_0 each need $\frac{k}{2}\mathbf{m}$).

In the next sections we give explicit formulas to efficiently compute c_{Z^2}, c_{XY}, and c_{XZ} for addition and doubling. For applications in cryptography we restrict our considerations to points in a group of prime order. Let the number of points on the curve factor as $\#\mathrm{E}_{a,d}(\mathbf{F}_p) = 4hn$, with n prime, and let the basepoint P have order n. This implies in particular that none of the additions or doublings involves Ω_1, Ω_2, or $\mathcal{O}'$. The neutral element $\mathcal{O}$ is a multiple of P, namely nP, but none of the operations in the Miller loop will have it as its input. This means that without loss of generality we can assume that none of the coordinates of the input points is 0. In fact, for this assumption to hold we only need that P has odd order, so that the points of order 2 or 4 are not multiples of it.

5.1 Addition

Hisil et al. presented new addition formulas for twisted Edwards curves at Asiacrypt 2008 [20]. To save $1\mathbf{m}$ they extended the representation by a further coordinate $T_1 = X_1Y_1/Z_1$ for points $P = (X_1 : Y_1 : Z_1)$ with $Z_1 \neq 0$. In the following section we show how to compute this value as part of the doubling step. As suggested in [20] it is only computed for the last doubling in a sequence of doublings and is not computed after an addition. Note that no addition is ever followed by another addition in the scalar multiplication. Furthermore, we assume that the base point P has odd order so in particular $Z_1, Z_2 \neq 0$. The sum $P_3 = (X_3 : Y_3 : Z_3)$ of two different points $P_1 = (X_1 : Y_1 : Z_1 : T_1)$ and $P_2 = (X_2 : Y_2 : Z_2 : T_2)$ in extended representation is given by

$$\begin{aligned}
X_3 &= (X_1Y_2 - Y_1X_2)(T_1Z_2 + Z_1T_2),\\
Y_3 &= (aX_1X_2 + Y_1Y_2)(T_1Z_2 - Z_1T_2),\\
Z_3 &= (aX_1X_2 + Y_1Y_2)(X_1Y_2 - Y_1X_2).
\end{aligned}$$

Theorem 4 (a) in Section 3 states the coefficients of the conic section for addition. We use T_1, T_2 to shorten the formulas.

$$\begin{aligned}
c_{Z^2} &= X_1X_2(Y_1Z_2 - Y_2Z_1) = Z_1Z_2(T_1X_2 - X_1T_2),\\
c_{XY} &= Z_1Z_2(X_1Z_2 - Z_1X_2 + X_1Y_2 - Y_1X_2),\\
c_{XZ} &= X_2Y_2Z_1^2 - X_1Y_1Z_2^2 + Y_1Y_2(X_2Z_1 - X_1Z_2)\\
&= Z_1Z_2(Z_1T_2 - T_1Z_2 + Y_1T_2 - T_1Y_2).
\end{aligned}$$

Note that all coefficients are divisible by $Z_1Z_2 \neq 0$ and so we scale the coefficients. The explicit formulas for computing $P_3 = P_1 + P_2$ and $(c_{Z^2}, c_{XY}, c_{XZ})$ are given as follows:

$$\begin{aligned}
A &= X_1 \cdot X_2;\ B = Y_1 \cdot Y_2;\ C = Z_1 \cdot T_2;\ D = T_1 \cdot Z_2;\ E = D + C;\\
F &= (X_1 - Y_1) \cdot (X_2 + Y_2) + B - A;\ G = B + aA;\ H = D - C;\ I = T_1 \cdot T_2;\\
c_{Z^2} &= (T_1 - X_1) \cdot (T_2 + X_2) - I + A;\ c_{XY} = X_1 \cdot Z_2 - X_2 \cdot Z_1 + F;\\
c_{XZ} &= (Y_1 - T_1) \cdot (Y_2 + T_2) - B + I - H;\\
X_3 &= E \cdot F;\ Y_3 = G \cdot H;\ Z_3 = F \cdot G.
\end{aligned}$$

With these formulas P_3 and $(c_{Z^2}, c_{XY}, c_{XZ})$ can be computed in $13\mathbf{m} + 1\mathbf{m_a}$, where $\mathbf{m_a}$ denotes the costs of a multiplication by the constant a. If T_3 is desired as part of the output it can be computed in $1\mathbf{m}$ as $T_3 = E \cdot H$. The point P_2 is not changed during pairing computation and can be given in affine coordinates, i. e. $Z_2 = 1$. Applying mixed addition, the above costs reduce to $11\mathbf{m} + 1\mathbf{m_a}$. We used Sage [32] to verify the explicit formulas. Note that there is no extra speed up from choosing $a = -1$ (unlike in [20]) since all subexpressions are also used in the computation of $(c_{Z^2}, c_{XY}, c_{XZ})$.

An addition step in Miller's algorithm for the Tate pairing thus costs $1\mathbf{M} + (k+11)\mathbf{m} + 1\mathbf{m_a}$.

5.2 Doubling

Theorem 4 (c) in Section 3 states the coefficients of the conic section in the case of doubling. To speed up the computation we multiply each coefficient by $-2Y_1/Z_1$; remember that ϕ was unique up to scaling. Note also that $Y_1, Z_1 \neq 0$ because we assume that all points have odd order. The multiplication by Y_1/Z_1 reduces the overall degree of the equations since we can use the curve equation to simplify the formula for c_{XY}; the factor 2 is useful in obtaining an $\mathbf{s}/\mathbf{m}$ tradeoff in the explicit formulas below. We obtain:

$$\begin{aligned}
c_{Z^2} &= X_1(2Y_1^2 - 2Y_1Z_1),\\
c_{XY} &= 2(Y_1Z_1^3 - dX_1^2Y_1^2)/Z_1 = 2(Y_1Z_1^3 - Z_1^2(aX_1^2 + Y_1^2) + Z_1^4)/Z_1\\
&= Z_1(2(Z_1^2 - aX_1^2 - Y_1^2) + 2Y_1Z_1),\\
c_{XZ} &= Y_1(2aX_1^2 - 2Y_1Z_1).
\end{aligned}$$

Of course we also need to compute $P_3 = [2]P_1$. We use the explicit formulas from [6] for the doubling and reuse subexpressions in computing the coefficients of the conic. The formulas were checked for correctness with Sage [32].

$$\begin{aligned}
&A = X_1^2;\ B = Y_1^2;\ C = Z_1^2; D = (X_1 + Y_1)^2;\ E = (Y_1 + Z_1)^2;\\
&F = D - (A + B);\ G = E - (B + C);\ H = aA;\ I = H + B;\\
&J = C - I;\ K = J + C;\ c_{XZ} = Y_1 \cdot (2H - G);\ c_{XY} = Z_1 \cdot (2J + G);\\
&c_{Z^2} = F \cdot (Y_1 - Z_1);\ X_3 = F \cdot K;\ Y_3 = I \cdot (B - H);\ Z_3 = I \cdot K.
\end{aligned}$$

These formulas compute P_3 and $(c_{Z^2}, c_{XY}, c_{XZ})$ in $6\mathbf{m} + 5\mathbf{s} + 1\mathbf{m_a}$. If the doubling is followed by an addition the additional coordinate $T_3 = X_3Y_3/Z_3$ needs to be computed. This is done by additionally computing $T_3 = F \cdot (B - H)$ in $1\mathbf{m}$.

If the input is given in extended form as $P_1 = (X_1 : Y_1 : Z_1 : T_1)$ we can use T_1 in the computation of the conic as

$$\begin{aligned}
c_{Z^2} &= X_1(2Y_1^2 - 2Y_1Z_1) = 2Z_1Y_1(T_1 - X_1),\\
c_{XY} &= Z_1(2(Z_1^2 - aX_1^2 - Y_1^2) + 2Y_1Z_1),\\
c_{XZ} &= Y_1(2aX_1^2 - 2Y_1Z_1) = 2Z_1(aX_1T_1 - Y_1^2),
\end{aligned}$$

and then scale the coefficients by $1/Z_1$. The computation of $P_3 = (X_3 : Y_3 : Z_3 : T_3)$ and $(c_{Z^2}, c_{XY}, c_{XZ})$ is then done in $6\mathbf{m} + 5\mathbf{s} + 2\mathbf{m_a}$ as

$$\begin{aligned}
&A = X_1^2;\ B = Y_1^2;\ C = Z_1^2; D = (X_1 + Y_1)^2;\ E = (Y_1 + Z_1)^2;\\
&F = D - (A + B);\ G = E - (B + C);\ H = aA;\ I = H + B;\ J = C - I;\\
&K = J + C;\ c_{Z^2} = 2Y_1 \cdot (T_1 - X_1);\ c_{XY} = 2J + G;\ c_{XZ} = 2(aX_1 \cdot T_1 - B);\\
&X_3 = F \cdot K;\ Y_3 = I \cdot (B - H);\ Z_3 = I \cdot K;\ T_3 = F \cdot (B - H).
\end{aligned}$$

For computing the Tate pairing this means that a doubling step costs $1\mathbf{M} + 1\mathbf{S} + (k+6)\mathbf{m} + 5\mathbf{s} + 1\mathbf{m_a}$ in twisted Edwards coordinates and $1\mathbf{M} + 1\mathbf{S} + (k+6)\mathbf{m} + 5\mathbf{s} + 2\mathbf{m_a}$ in extended coordinates.

5.3 Miller loop

Miller's algorithm loops over the bits in the representation of n. We follow Hisil et al. [20] and denote the system of projective Edwards coordinates $(X_1 : Y_1 : Z_1)$ by $\mathcal{E}$ and the extended system $(X_1 : Y_1 : Z_1 : T_1)$ by $\mathcal{E}^e$.

If the whole computation is carried out in $\mathcal{E}^e$ each addition step in the Tate pairing needs $1\mathbf{M} + (k+14)\mathbf{m} + 1\mathbf{m_a}$ if both points are projective and $1\mathbf{M} + (k+12)\mathbf{m} + 1\mathbf{m_a}$ if the addition is mixed. A doubling costs $1\mathbf{M} + 1\mathbf{S} + (k+6)\mathbf{m} + 5\mathbf{s} + 2\mathbf{m_a}$.

We can save $1\mathbf{m_a}$ per doubling by using the following idea which is already mentioned by Cohen et al [12]. If we are faced with s consecutive doublings between additions we execute the first $s-1$ doublings as $2\mathcal{E} \to \mathcal{E}$, do the last one as $2\mathcal{E} \to \mathcal{E}^e$ and then perform the addition as $\mathcal{E}^e + \mathcal{E}^e \to \mathcal{E}$. We account for the extra $\mathbf{m}$ needed in $2\mathcal{E} \to \mathcal{E}^e$ when stating the cost for addition. This way each addition step needs $1\mathbf{M} + (k+14)\mathbf{m} + 1\mathbf{m_a}$ if both points are projective and $1\mathbf{M} + (k+12)\mathbf{m} + 1\mathbf{m_a}$ if the addition is mixed. A doubling costs $1\mathbf{M} + 1\mathbf{S} + (k+6)\mathbf{m} + 5\mathbf{s} + 1\mathbf{m_a}$.

6 Operation counts

We give an overview of the best formulas in the literature for pairing computation on Edwards curves and for the different forms of Weierstrass curves in Jacobian coordinates. In the Appendix (Section 9) we present new, faster formulas for pairings on Weierstrass curves. We compare the results with the pairing formulas for Edwards curves from the previous section.

Throughout this section we assume that k is even, that the second input point Q is given in affine coordinates, and that quadratic twists are used to have multiplications with x_Q and y_Q take only $(k/2)\mathbf{m}$ each.

6.1 Overview

Chatterjee, Sarkar, and Barua [9] study pairings on Weierstrass curves in Jacobian coordinates. Their paper does not distinguish between multiplications in $\mathbf{F}_p$

and in $\mathbf{F}_{p^k}$ but their results are easily translated. For mixed addition their formulas need $1\mathbf{M}+(k+9)\mathbf{m}+3\mathbf{s}$. For doublings they need $1\mathbf{M}+(k+7)\mathbf{m}+1\mathbf{S}+4\mathbf{s}$ if $a_4 = -3$. For doubling on general Weierstrass curves (no condition on a_4) the formulas by Ionica and Joux [21] are fastest with $1\mathbf{M} + (k+1)\mathbf{m} + 1\mathbf{S} + 11\mathbf{s}$.

Actually, any mixed addition (mADD) or addition (ADD) needs $1\mathbf{M} + k\mathbf{m}$ for the evaluation at Q and the update of f; each doubling (DBL) needs $1\mathbf{M} + k\mathbf{m} + 1\mathbf{S}$ for the evaluation at Q and the update of f. In the following we do not comment on these costs since they do not depend on the chosen representation and are a fixed offset. We also do not report these expenses in the overview table.

Hankerson, Menezes, and Scott [19] study pairing computation on Barreto-Naehrig [5] curves. All BN curves have the form $y^2 = x^3 + b$ and are thus more special than curves with $a_4 = -3$ or Edwards curves. In their presentation they combine the pairing computation with the extension-field arithmetic and thus the operation for the pure pairing computation is not stated explicitly but the formulas match those in [10]. They need $6\mathbf{m} + 5\mathbf{s}$ for a doubling step and $9\mathbf{m} + 3\mathbf{s}$ for a mixed addition step when computing the update functions for the Tate pairing.

We present new formulas for Weierstrass curves in the appendix (Section 9). The results are stated in the table as "this paper".

Das and Sarkar [13] were the first to publish pairing formulas for Edwards curves. We do not include them in our overview since their study is specific to supersingular curves with $k = 2$.

Ionica and Joux [21] proposed the thus far fastest pairing formulas for Edwards curves. Note that they actually compute the 4th power $T_n(P,Q)^4$ of the Tate pairing. This has almost no negative effect for usage in protocols. So we include their result as pairings on Edwards curves.

We denote Edwards coordinates by $\mathcal{E}$ and Jacobian coordinates by $\mathcal{J}$. The row "$\mathcal{E}$, this paper" reports the results of the previous section using $2\mathcal{E} \to \mathcal{E}$ for the main doublings, $2\mathcal{E} \to \mathcal{E}^e$ for the final doubling, and $\mathcal{E}^e + \mathcal{E}^e \to \mathcal{E}$ for the addition. Using only $\mathcal{E}^e$ for all operations requires $1\mathbf{m_a}$ more per doubling.

	DBL	mADD	ADD
$\mathcal{J}$, [21], [9]	$1\mathbf{m} + 11\mathbf{s} + 1\mathbf{m_{a_4}}$	$9\mathbf{m} + 3\mathbf{s}$	—
$\mathcal{J}$, [21], this paper	$1\mathbf{m} + 11\mathbf{s} + 1\mathbf{m_{a_4}}$	$6\mathbf{m} + 6\mathbf{s}$	$15\mathbf{m} + 6\mathbf{s}$
$\mathcal{J}, a_4 = -3$, [9]	$7\mathbf{m} + 4\mathbf{s}$	$9\mathbf{m} + 3\mathbf{s}$	—
$\mathcal{J}, a_4 = -3$, this paper	$6\mathbf{m} + 5\mathbf{s}$	$6\mathbf{m} + 6\mathbf{s}$	$15\mathbf{m} + 6\mathbf{s}$
$\mathcal{J}, a_4 = 0$, [10], [9]	$6\mathbf{m} + 5\mathbf{s}$	$9\mathbf{m} + 3\mathbf{s}$	—
$\mathcal{J}, a_4 = 0$, this paper	$3\mathbf{m} + 8\mathbf{s}$	$6\mathbf{m} + 6\mathbf{s}$	$15\mathbf{m} + 6\mathbf{s}$
$\mathcal{E}$, [21]	$8\mathbf{m} + 4\mathbf{s} + 1\mathbf{m_d}$	$14\mathbf{m} + 4\mathbf{s} + 1\mathbf{m_d}$	—
$\mathcal{E}$, this paper	$6\mathbf{m} + 5\mathbf{s} + 1\mathbf{m_a}$	$12\mathbf{m} + 1\mathbf{m_a}$	$14\mathbf{m} + 1\mathbf{m_a}$

6.2 Comparison

We point out that all the example curves presented in Section 8 are Edwards curves so that the multiplication costs $\mathbf{m_a}$ vanish. More generally, since both a_4 and a can be chosen within some range and are usually small, multiplications by them are negligible, i.e. we assume $1\mathbf{m_{a_4}} = 1\mathbf{m_a} = 0$ in this comparison.

The overview shows that our new formulas for Edwards curves solidly beat any formulas published for pairing computation on Edwards curves. Our new formulas for Edwards curves need fewer field operations and have a larger percentage of squarings among them.

We first compare our new formulas for Edwards curves with formulas in the literature for Weierstrass curves. For doubling, our formulas are as efficient as the so far most efficient ones ($a_4 = 0$) and faster than the $7\mathbf{m} + 4\mathbf{s}$ for $a_4 = -3$ if $\mathbf{s} \leq \mathbf{m}$. The comparison with the case of general curves depends on the $\mathbf{s}/\mathbf{m}$ ratio. For cheap $\mathbf{s}$ the general formulas are faster while otherwise our Edwards doubling step is faster. Note that the general formulas can be used to double in the special cases of a_4.

For addition, Edwards curves need the same number of field operations as Weierstrass curves, but the formulas have no squarings. So they are slower if $\mathbf{s} \leq \mathbf{m}$. Overall, the new formulas are competitive with the formulas in the literature.

Our own improvements to the doubling and addition formulas for Weierstrass curves keep the same number of field operations as previously published formulas for the same Weierstrass curves but we manage to trade off several multiplications for squarings. Compared to these new formulas given in the appendix the doubling step on Edwards curves is slower or at best as fast as on Weierstrass curves unless the curve has a general a_4 and $\mathbf{m} - \mathbf{s}$ is small. For the mixed addition the gap has widened so that fast squarings make the Edwards mixed addition look worse in comparison.

It is important to note that the $\mathbf{s} - \mathbf{m}$ tradeoffs come at the expense of extra field additions and intermediate variables. If squarings are not particularly cheap or if storage is restricted the tradeoffs might not be worthwhile.

The penalty for computing full additions instead of mixed additions is only $2\mathbf{m}$ for Edwards curves. The gap between mADD and ADD is significantly worse for Jacobian coordinates where an optimized addition step costs $15\mathbf{m} + 6\mathbf{s}$, i.e. $9\mathbf{m}$ more than the best formulas. This is significantly more than the computation in Edwards coordinates. Therefore, Edwards curves are the clear winner if the input point P is not in affine coordinates; e.g. in protocols that compute the first input point using scalar multiplication. This is also the most likely use case for pairings on Edwards curves because the implementation can use the fast scalar multiplication on Edwards curves.

7 Construction of Pairing-Friendly Edwards Curves

We describe a way to generate Edwards curves over finite fields with embedding degree $k = 6$ and cryptographic bitsize. They are constructed as pairing-friendly elliptic curves in Weierstrass form with a group order divisible by 4 [6, Thm. 3.3]. Since all parameterized families of pairing-friendly curves with ρ-value 1 yield curves with odd group order, we use the construction of generalized MNT curves with cofactor 4 given by Galbraith, McKee, and Valença [17]. The following polynomial parameterizations lead to curves over $\mathbf{F}_{q(\ell)}$ with embedding degree $k = 6$, trace $t(\ell)$ and group order $4n(\ell)$:

Case	$q(\ell)$	$t(\ell)$	$n(\ell)$
1	$16\ell^2 + 10\ell + 5$	$2\ell + 2$	$4\ell^2 + 2\ell + 1$
2	$112\ell^2 + 54\ell + 7$	$14\ell + 4$	$28\ell^2 + 10\ell + 1$
3	$112\ell^2 + 86\ell + 17$	$14\ell + 6$	$28\ell^2 + 18\ell + 3$
4	$208\ell^2 + 30\ell + 1$	$-26\ell - 2$	$52\ell^2 + 14\ell + 1$
5	$208\ell^2 + 126\ell + 19$	$-26\ell - 8$	$52\ell^2 + 38\ell + 7$

Constructing a curve with parameters given by the above polynomials requires the CM norm equation

$$t(\ell)^2 - 4q(\ell) = -Dy^2$$

to be fulfilled with a small positive discriminant D. Since choosing ℓ such that $q(\ell)$ and $n(\ell)$ are prime results in too large values for D, the construction has to be done as for MNT curves [25] by first solving the corresponding Pell equation. For instance, in Case 1, we get

$$t(\ell)^2 - 4\,q(\ell) = -D\,y^2 \iff x^2 - 15\,D\,y^2 = -44,$$

where $x = 15\ell + 4$.

Remark 8. The method to construct pairing-friendly Edwards curves described in this section focuses on the case $k = 6$. For efficient implementation, we aim at balancing the difficulty of the DLPs on the curve and in the finite field $\mathbf{F}_{p^6}$. Following the ECRYPT recommendations [30], the "optimal" bitsizes for curves $E/\mathbf{F}_p$ with $\#E(\mathbf{F}_p) = 4hn$ and n prime are shown in Table 1 for the most common security levels. For these parameters, the DLP in the subgroup of $E(\mathbf{F}_p)$ of order n is considered equally hard as the DLP in $\mathbf{F}^*_{p^6}$. Using curves with a cofactor of the given size ensures that the prime n is as small as possible for the corresponding size of p. The bit length of n is equal to the length of the Miller loop. Hence it can be minimized for the given security level by choosing a curve with parameter sizes as indicated in Table 1.

n	p	p^6	h
160	208	1248	46
192	296	1776	102
224	405	2432	179
256	541	3248	283
512	2570	15424	2056

Table 1. "Optimal" bitsizes for the primes n, p and the cofactor h.

8 Examples of Edwards curves with embedding degree 6

We present examples of Edwards curves with embedding degree $k = 6$. They were constructed using the method described in the previous section. Together with David Kohel [1] we have shown that any elliptic curve in Weierstrass form for which 4 divides the number of rational points can be transformed into a plain Edwards curve by applying a sequence of 2-isogenies. Thus it is possible to always find a curve with $a = 1$ and we use this idea to construct the following examples. Note that choosing $a \neq 1$ offers some flexibility in choosing a small basepoint.

In the following we present four examples of pairing-friendly Edwards curves with embedding degree 6. The first one comes from the particularly easy to handle discriminant $D = 1$. The other three have well-balanced parameters and are interesting for cryptographic applications. Notation is as before, where the number of $\mathbf{F}_p$-rational points on the curve is $4hn$.

– $D = 1$, $\lceil\log(n)\rceil = 363$, $\lceil\log(h)\rceil = 7$, $\lceil\log(p)\rceil = 371$

$$\begin{aligned}
p &= 3242890372842743487196063845602840916228193958243257594 5\\
&\quad 306321535594026280100199466816249589739372396374201691 41,\\
n &= 1110578894809158728491802686850287985009655465151800546 0\\
&\quad 62383206431203589781550995148890796453200096599378724 1,\\
h &= 73,\\
d &= 1621445186421371743598031922801420458114096979121628797 2\\
&\quad 6531607677970131400500997334081247948696861981871008457 1.
\end{aligned}$$

– $D = 7230$, $\lceil\log(n)\rceil = 165$, $\lceil\log(h)\rceil = 34$, $\lceil\log(p)\rceil = 201$

$$\begin{aligned}
p &= 2051613663768129606093583432875887398415301962227490187508801,\\
n &= 44812545413308579913957438201331385434743442366277,\\
h &= 7 \cdot 733 \cdot 2230663,\\
d &= 889556570662354157210639662153375862261205379822879716332449.
\end{aligned}$$

– $D = 4630$, $\lceil \log(n) \rceil = 191$, $\lceil \log(h) \rceil = 90$, $\lceil \log(p) \rceil = 283$

$$\begin{aligned}
p &= 120764224732576209996227729242202305356551042856008263578 5\\
&\quad 60701796190315106158863616 01,\\
n &= 2498886235887409414948289020220476887707263210939845485839,\\
h &= 11161 \cdot 19068349 \cdot 5676957216676051,\\
d &= 4597008687866412934970378498245465932931615077893178705320\\
&\quad 7445923055271353005027781 90.
\end{aligned}$$

– $D = 314$, $\lceil \log(n) \rceil = 220$, $\lceil \log(h) \rceil = 98$, $\lceil \log(p) \rceil = 319$

$$\begin{aligned}
p &= 9452707311247707513330618188853923205411626343551115070653265144510\\
&\quad 0440884421216867565977827 2001,\\
n &= 13364958610259914721463310337607104185807430907691125850531645345 99,\\
h &= 3 \cdot 589396220550901517029052719 33,\\
d &= 6987035440681854303189570169230960965572653735366548738901481656990\\
&\quad 6295089670835705731049716 7360.
\end{aligned}$$

References

1. Christophe Arène, David Kohel, Tanja Lange, Michael Naehrig, and Christophe Ritzenthaler. Work in progress, 2009.
2. Roberto M. Avanzi, Henri Cohen, Christophe Doche, Gerhard Frey, Tanja Lange, Kim Nguyen, and Frederik Vercauteren. *The Handbook of Elliptic and Hyperelliptic Curve Cryptography.* CRC, 2005.
3. Paulo S. L. M. Barreto, Hae Yong Kim, Ben Lynn, and Michael Scott. Efficient algorithms for pairing-based cryptosystems. In *CRYPTO 2002 [34]*, pages 354–368, 2002.
4. Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. Efficient implementation of pairing-based cryptosystems. *J. Cryptology*, 17:321–334, 2004.
5. Paulo S. L. M. Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In *SAC 2005 [29]*, pages 319–331, 2006.
6. Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. Twisted Edwards curves. In *Africacrypt [33]*, pages 389–405, 2008. http://cr.yp.to/papers.html#twisted.
7. Daniel J. Bernstein and Tanja Lange. Explicit-formulas database. http://www.hyperelliptic.org/EFD.
8. Daniel J. Bernstein and Tanja Lange. Faster addition and doubling on elliptic curves. In *ASIACRYPT 2007 [23]*, pages 29–50, 2007. http://cr.yp.to/newelliptic/.
9. Sanjit Chatterjee, Palash Sarkar, and Rana Barua. Efficient computation of Tate pairing in projective coordinate over general characteristic fields. In *ICISC 2004 [27]*, pages 168–181, 2005.

10. Zhaohui Cheng and Manos Nistazakis. Implementing pairing-based cryptosystems. In *3rd International Workshop on Wireless Security Technologies IWWST-2005*, 2005.
11. Dipanwita Roy Chowdhury, Vincent Rijmen, and Abhijit Das, editors. *Progress in Cryptology - INDOCRYPT 2008, 9th International Conference on Cryptology in India, Kharagpur, India, December 14-17, 2008, proceedings*, volume 5365 of *Lecture Notes in Computer Science*, Berlin, 2008. Springer.
12. Henry Cohen, Atsuko Miyaji, and Takatoshi Ono. Efficient elliptic curve exponentiation using mixed coordinates. In *ASIACRYPT 1998 [26]*, pages 51–65, 1998. http://www.math.u-bordeaux.fr/∼ cohen/asiacrypt98.dvi.
13. M. Prem Laxman Das and Palash Sarkar. Pairing computation on twisted Edwards form elliptic curves. In *Pairing 2008 [18]*, pages 192–210, 2008.
14. Harold M. Edwards. A normal form for elliptic curves. *Bulletin of the American Mathematical Society*, 44:393–422, 2007. http://www.ams.org/bull/2007-44-03/S0273-0979-07-01153-6/home.html.
15. Gerhard Frey and Tanja Lange. *Background on Curves and Jacobians*, chapter 13 in [2], pages 45–85. 2005.
16. William Fulton. *Algebraic Curves*. W. A. Benjamin, Inc., 1969.
17. Steven D. Galbraith, James F. McKee, and Paula C. Valença. Ordinary abelian varieties having small embedding degree. *Finite Fields and their Applications*, 13:800–814, 2007.
18. Steven D. Galbraith and Kenneth G. Paterson, editors. *Pairing-Based Cryptography - Pairing 2008, Second International Conference, Egham, UK, September 1-3, 2008, Proceedings*, volume 5209 of *Lecture Notes in Computer Science*, Berlin, 2008. Spriger.
19. Darrel Hankerson, Alfred J. Menezes, and Michael Scott. Software implementation of pairings. In *Identity-Based Cryptography [22]*, pages 188–206, 2009.
20. Huseyin Hisil, Kenneth Koon-Ho Wong, Gary Carter, and Ed Dawson. Twisted Edwards curves revisited. In *ASIACRYPT 2008 [28]*, pages 326–343, 2008.
21. Sorina Ionica and Antoine Joux. Another approach to pairing computation in Edwards coordinates. In *INDOCRYPT 2008 [11]*, pages 400–413, 2008. http://eprint.iacr.org/2008/292.
22. Marc Joye and Gregory Neven, editors. *Identity-Based Cryptography*, volume 2 of *Cryptology and Information Security Series*. IOS Press, 2009.
23. Kaoru Kurosawa, editor. *Advances in Cryptology — ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, Berlin Heidelberg, 2007. Springer.
24. Victor S. Miller. The Weil pairing, and its efficient calculation. *Journal of Cryptology*, 17:235–261, 2004.
25. Atsuko Miyaji, Masaki Nakabayashi, and Shunzou Takano. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Transactions on Fundamentals*, E84-A(5):1234–1243, 2001.
26. Kazuo Ohta and Dingyi Pei, editors. *Advances in cryptology — ASIACRYPT'98, International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, October 18–22, 1998, Proceedings*, volume 1514 of *Lecture Notes in Computer Science*, Berlin, 1998. Springer.
27. Choonsik Park and Seongtaek Chee, editors. *Information Security and Cryptology - ICISC 2004, 7th International Conference, Seoul, Korea, December 2-3, 2004, Revised Selected Papers*, volume 3506 of *Lecture Notes in Computer Science*. Springer, 2005.

28. Josef Pieprzyk, editor. *Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 7-11, 2008. Proceedings*, volume 5350 of *Lecture Notes in Computer Science*, Berlin, 2008. Springer.
29. Bart Preneel and Stafford E. Tavares, editors. *Selected Areas in Cryptography, 12th International Workshop, SAC 2005, Kingston, ON, Canada, August 11-12, 2005, Revised Selected Papers*, volume 3897 of *Lecture Notes in Computer Science*. Springer, 2006.
30. Christian Rechberger and Vincent Rijmen. ECRYPT yearly report on algorithms and keysizes (2007-2008). Technical report, 2008. D.SPA.28, IST-2002-507932.
31. Joseph H. Silverman. *The Arithmetic of Elliptic Curves.* Number 106 in Graduate texts in mathematics. Springer-Verlag, 1986.
32. William Stein. Sage mathematics software (version 2.8.12), 2008. The Sage Group, http://www.sagemath.org.
33. Serge Vaudenay, editor. *Progress in Cryptology - AFRICACRYPT 2008, First International Conference on Cryptology in Africa, Casablanca, Morocco, June 11-14, 2008, proceedings.*, Lecture Notes in Computer Science, Berlin, 2008. Springer.
34. Moti Yung, editor. *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, volume 2442 of *Lecture Notes in Computer Science*. Springer, 2002.

9 Appendix: Formulas for Weierstrass curves

To obtain the full speed of pairings on Weierstrass curves it is useful to represent a point by $(X_1 : Y_1 : Z_1 : T_1)$ with $T_1 = Z_1^2$. This allows one $\mathbf{s} - \mathbf{m}$ tradeoff in the doubling step and one in the addition step compared with the usual representation $(X_1 : Y_1 : Z_1)$.

The line function for Weierstrass curves is given by

$$g_{R,P}(X : Y : Z) = \frac{l_{R,P}}{l_{R+P}} = \frac{(YZ_0^3 - Y_0Z^3) - \lambda(XZ_0^2 - X_0Z^2)ZZ_0}{(X - cZ)Z^2},$$

where λ is the slope, $(X_0 : Y_0 : Z_0)$ is a point on the line, and c is some constant. When one computes the Tate pairing, the point $(X_0 : Y_0 : Z_0)$ and the constants λ and c are defined over the base field $\mathbf{F}_p$. The function is evaluated at a point $Q = (X_Q : Y_Q : Z_Q)$ defined over $\mathbf{F}_{p^k}$; if k is even then the field extension $\mathbf{F}_{p^k}$ is usually constructed via a quadratic subfield as $\mathbf{F}_{p^k} = \mathbf{F}_{p^{k/2}}(\alpha)$, with $\alpha^2 = \delta$ and Q is chosen to be of the form $Q = (x_Q : y_Q\alpha : 1)$ with $x_Q, y_Q \in \mathbf{F}_{p^{k/2}}$. Like in the case of Edwards curves only the numerator needs to be considered and all multiplicative contributions from subfields of $\mathbf{F}_{p^k}$ can be discarded. In particular $\lambda = L_1/Z_3$ for curves in Jacobian coordinates and thus the computation simplifies to computing

$$Z_3(y_Q Z_0^3\alpha - Y_0) - L_1(x_Q Z_0^2 - X_0)Z_0.$$

Multiplications with x_Q and y_Q cost $(k/2)\mathbf{m}$ each; for $k > 2$ it is thus useful to rewrite this equation as

$$l = ((Z_3 \cdot Z_0) \cdot Z_0^2) \cdot y_Q\alpha - Y_0 \cdot Z_3 - (L_1 \cdot Z_0) \cdot Z_0^2 \cdot x_Q + X_0 \cdot (L_1 \cdot Z_0)$$

needing at worst $(k+6)\mathbf{m}+1\mathbf{s}$; in most cases some computations can be reused. In particular, if $T_0 = Z_0^2$ and $T_3 = Z_3^2$ are known at worst $(k+5)\mathbf{m}+1\mathbf{s}$ are needed. Additionally $1\mathbf{M}$ or $1\mathbf{M}+1\mathbf{S}$ are needed to update the function f in Miller's algorithm.

9.1 Addition

For addition we use Bernstein and Lange's formulas ("add-2007-bl") from the EFD [7], enhanced with caching $T = Z^2$. The point $(X_0 : Y_0 : Z_0)$ on the line can be chosen to be $(X_2 : Y_2 : Z_2)$; the numerator of λ is r. Then $T_2 = Z_0^2 = Z_2^2 = B$ is already computed and $L_1 \cdot Z_0 = (r+Z_2)^2 - r^2 - B$, where r^2 is computed in computing X_3.

$$
\begin{aligned}
&A = X_1 \cdot T_2;\ B = X_2 \cdot T_1;\ C = Y_1 \cdot Z_2 \cdot T_2;\ D = Y_2 \cdot Z_1 \cdot T_1;\ H = B - A;\\
&I = (2H)^2;\ J = H \cdot I;\ r = 2(D - C);\ R = r^2;\ V = A \cdot I;\ W = (r + Z_2)^2 - R - T_2;\\
&X_3 = R - J - 2V;\ Y_3 = r \cdot (V - X_3) - 2C \cdot J;\ Z_3 = ((Z_1 + Z_2)^2 - T_1 - T_2) \cdot H;\\
&T_3 = Z_3^2;\ c_1 = ((Z_3 + Z_2)^2 - T_2 - T_3) \cdot T_2;\ c_2 = 2Y_2 \cdot Z_3;\ c_3 = W \cdot T_2;\ c_4 = W \cdot X_2;\\
&l = c_1 \cdot y_Q\alpha - c_2 - c_3 \cdot x_Q + c_4.
\end{aligned}
$$

The formulas need $1\mathbf{M} + (k+15)\mathbf{m} + 6\mathbf{s}$ to compute the addition step. To our knowledge this is the first set of formulas for full (non-mixed) addition. Note that the usage of these formulas in pairings required some different optimizations; a naive application would need $(11\mathbf{m}+5\mathbf{s})+1\mathbf{M}+(k+6)\mathbf{m}+1\mathbf{s}$ without the extra coordinates T_i and without the additional tricks.

If $(X_2 : Y_2 : Z_2)$ is fixed throughout the computation and $\mathbf{m} > \mathbf{s}$ then it is worthwhile computing and storing $S_2 = Y_2^2$. This allows to compute $2D = ((Y_2 + Z_1)^2 - S_2^2 - T_1) \cdot T_1$ and $c_2 = (Y_2 + Z_3)^2 - S_2 - T_2$. Note that using $2D$ in place of D requires scaling everything by 2. The complete computation can be done in $1\mathbf{M} + (k+13)\mathbf{m} + 8\mathbf{s}$; we do not report these numbers in the table since projective base points are most likely to happen if the base point is changing and/or the device is constrained.

9.2 Mixed addition

Mixed addition means that the second input point is in affine representation, i.e. $Z_2 = 1$ and thus also $T_2 = 1$. Choosing the point $(X_0 : Y_0 : Z_0)$ on the line as this point $(x_2 : y_2 : 1)$ saves several operations in the addition as well as in the computation of the line function:

$$l = Z_3 \cdot y_Q\alpha - y_2 \cdot Z_3 - L_1 \cdot (x_Q - x_2)$$

which simplifies to only $(k+1)\mathbf{m}$. Note that this is better than computing the first part as $Z_3 \cdot (y_Q\alpha - y_2)$ since a multiplication by $y_Q\alpha - y_2$ costs $k\mathbf{m}$ instead of $(k/2)\mathbf{m}$. This base point is usually fixed throughout the computation and, given that it is provided in affine coordinates, it is likely a long-term input; thus it is worthwhile storing it as (x_2, y_2, y_2^2).

We now state the mixed addition formulas based on Bernstein and Lange's formulas ("add-2007-bl") from the EFD [7]. Mixed additions are the usual case studied for pairings and the evaluation in $(k+1)\mathbf{m}$ is standard. However, most implementations miss the $\mathbf{s}-\mathbf{m}$ tradeoff in the main mixed addition formulas and do not compute the T-coordinate.

$$\begin{aligned}
&B = x_2 \cdot T_1;\ D = ((y_2+Z_1)^2 - y_2^2 - T_1)\cdot T_1;\ H = B - X_1;\\
&I = H^2;\ E = 4I;\ J = H\cdot E;\ r = 2(D - Y_1);\ V = X_1 \cdot E;\\
&X_3 = r^2 - J - 2V;\ Y_3 = r\cdot(V - X_3) - 2Y_1 \cdot J;\ Z_3 = (Z_1+H)^2 - T_1 - I;\\
&T_3 = Z_3^2;\ l = Z_3 \cdot y_Q\alpha - (y_2+Z_3)^2 + y_2^2 + T_3 - r\cdot(x_Q - x_2).
\end{aligned}$$

The formulas need $1\mathbf{M} + (k+6)\mathbf{m} + 6\mathbf{s}$ to compute the mixed addition step.

9.3 Doubling

The main differences between the addition and the doubling formulas are that the doubling formulas depend on the curve shape and that the line function must be computed with $(X_0 : Y_0 : Z_0) = (X_1 : Y_1 : Z_1)$, where generically $Z_0 \neq 1$.

They have in common the general equation of the slope

$$\lambda = (3X_1^2 - a_4 Z_1^2)/(2Y_1 Z_1) = (3X_1^2 - a_4 Z_1^2)/Z_3.$$

Thus Z_3 is divisible by Z_1 and we can replace l by $l' = l/Z_1$ which will give the same result for the pairing computation. The value of

$$l' = (Z_3 \cdot Z_1^2)\cdot y_Q\alpha - 2Y_1^2 - L_1 \cdot Z_1^2 \cdot x_Q + X_1 \cdot L_1$$

can be computed in at worst $(k+3)\mathbf{m} + 1\mathbf{s}$.

The formulas by Ionica and Joux take into account the doubling formulas from the EFD for general Weierstrass curves in Jacobian coordinates. We thus present new formulas for the more special curves with $a_4 = -3$ and for $a_4 = 0$.

Doubling on curves with $\boldsymbol{a_4 = -3}$ The fastest doubling formulas are due to Bernstein (see [7] "dbl-2001-b") and need $3\mathbf{m} + 5\mathbf{s}$ for the doubling.

$$\begin{aligned}
&A = Y_1^2;\ B = X_1 \cdot A;\ C = 3(X_1 - T_1)\cdot(X_1 + T_1);\\
&X_3 = C^2 - 8B;\ Z_3 = (Y_1+Z_1)^2 - A - T_1;\ Y_3 = C\cdot(4B - X_3) - 8A^2;\ T_3 = Z_3^2;\\
&l = (Z_3 \cdot T_1)\cdot y_Q\alpha - 2A - L_1 \cdot T_1 \cdot x_Q + X_1 \cdot L_1
\end{aligned}$$

The complete doubling step thus takes $1\mathbf{M} + 1\mathbf{S} + (k+6)\mathbf{m} + 5\mathbf{s}$.

Doubling on curves with $\boldsymbol{a_4 = 0}$ The following formulas compute a doubling in $1\mathbf{m} + 7\mathbf{s}$. Note that without T_1 and computing $Z_3 = 2Y_1 \cdot Z_1$ a doubling can be computed in $2\mathbf{m} + 5\mathbf{s}$ which is always faster (see [7]) but the line functions make use of Z_1^2. Note that here $L_1 = E = 3X_1^2$ is particularly simple.

$$A = X_1^2;\ B = Y_1^2;\ C = B^2;\ D = 2((X_1 + B)^2 - A - C);\ E = 3A;\ G = E^2;$$
$$X_3 = G - 2D;\ Y_3 = E \cdot (D - X_3) - 8C;\ Z_3 = (Y_1 + Z_1)^2 - B - T_1;\ T_3 = Z_3^2;$$
$$l = 2(Z_3 \cdot T_1) \cdot y_Q \alpha - 4B - 2E \cdot T_1 \cdot x_Q + (X_1 + E)^2 - A - G$$

The complete doubling step thus takes $1\mathbf{M} + 1\mathbf{S} + (k+3)\mathbf{m} + 8\mathbf{s}$.