Faster Computation of the Tate Pairing

Christophe Arène[a], Tanja Lange[*,b], Michael Naehrig[b,c], Christophe Ritzenthaler[a]

[a] *Institut de Mathématiques de Luminy*
163, avenue de Luminy, Case 907
13288 Marseille CEDEX 09
France
[b] *Department of Mathematics and Computer Science*
Technische Universiteit Eindhoven
P.O. Box 513, 5600 MB Eindhoven
Netherlands
[c] *Microsoft Research*
One Microsoft Way
Redmond, 98052 WA
USA

Abstract

This paper proposes new explicit formulas for the doubling and addition steps in Miller's algorithm to compute the Tate pairing on elliptic curves in Weierstrass and in Edwards form. 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 addition and doubling. The Tate pairing on Edwards curves can be computed by using these functions in Miller's algorithm.

Computing the sum of two points or the double of a point and the coefficients of the corresponding functions is faster with our formulas 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 improve the formulas for Tate pairing computation on Weierstrass curves in Jacobian coordinates. Finally, we present several examples of pairing-friendly Edwards curves.

Key words: Pairings, Miller functions, explicit formulas, Edwards curves.

1. Introduction

Since their introduction to cryptography by Bernstein and Lange [7], Edwards curves have received a lot of attention due to the fact that their group law can be computed very efficiently. The group law in affine form was introduced by Edwards in [15] along with a description of the curve and several proofs of correctness. Remarkably none of the proofs provided a geometric interpretation while addition on Weierstrass curves is usually explained via the chord-and-tangent method.

Cryptographic 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 with divisor $(P) + (Q) - (P + Q) - (\mathcal{O})$ for two input points P and Q, their

sum $P + Q$, and neutral element $\mathcal{O}$. 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 curve 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 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 [23] 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 for usage in protocols as long as all participating parties 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 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 interpretation of the group law to show how to compute the Tate pairing on twisted Edwards curves.

- We give examples of ordinary pairing-friendly Edwards curves at several security levels. The curves have embedding degrees between 6 and 22.

Beyond that, we develop explicit formulas for computing the Tate pairing on Edwards curves that

- solidly beat the results by Das and Sarkar [13] and Ionica and Joux [23];

- are as fast as the fastest previously 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; and

- have minimal performance penalty for non-affine base points.

In particular, for even embedding degree k the doubling step on an Edwards curve takes $1\mathbf{M} + 1\mathbf{S} + (k + 6)\mathbf{m} + 5\mathbf{s}$, 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 $1\mathbf{M} + (k + 12)\mathbf{m}$ and an addition step takes $1\mathbf{M} + (k + 14)\mathbf{m}$. Our method for pairing computation on Edwards curves can be used for all curves that can be represented in Edwards form over the base field.

We also improve the addition and doubling steps on Weierstrass curves given by an equation $y^2 = x^3 + a_4 x + a_6$. We present the first explicit formulas for full addition steps on Weierstrass curves. The new formulas need $1\mathbf{M} + 1\mathbf{S} + (k + 6)\mathbf{m} + 5\mathbf{s}$ for a doubling step on curves with coefficient $a_4 = -3$. On such curves a mixed addition step costs $1\mathbf{M} + (k + 6)\mathbf{m} + 6\mathbf{s}$ and an addition step costs $1\mathbf{M} + (k + 9)\mathbf{m} + 6\mathbf{s}$. On curves with $a_4 = 0$, the formulas take $1\mathbf{M} + 1\mathbf{S} + (k + 3)\mathbf{m} + 8\mathbf{s}$ for a doubling step, $1\mathbf{M} + (k + 6)\mathbf{m} + 6\mathbf{s}$ for a mixed addition, and $1\mathbf{M} + (k + 9)\mathbf{m} + 6\mathbf{s}$ for an addition step.

Our new formulas for Weierstrass curves are the fastest when using affine base points (except in the case $a_4 = 0$, $a_6 = b^2$). For projective base points – a common case in pairing-based protocols – it is better to use Edwards curves.

2. Background on Pairings

Let q be a prime power not divisible by 2 and let $E/\mathbf{F}_q$ be an elliptic curve over $\mathbf{F}_q$ with neutral element denoted by $\mathcal{O}$. Let $n \mid \#E(\mathbf{F}_q)$ be a prime divisor of the group order and let E have embedding degree $k > 1$ with respect to n, i.e. k is the smallest integer such that $n \mid q^k - 1$.

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

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

Miller [26] 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, where $n_{l-1} = 1$. Let $g_{R,S} \in \mathbf{F}_q(E)$ be the function arising in the addition of two points R and S on E, i.e. $g_{R,S}$ is a function with $\operatorname{div}(g_{R,S}) = (R) + (S) - (R+S) - (\mathcal{O})$, where $\mathcal{O}$ denotes the neutral element in the group of points, $R+S$ denotes the sum of R and S on E, and additions of the form $(R) + (S)$ 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) $f \leftarrow f^2 \cdot g_{R,R}(Q),\ R \leftarrow [2]R,$ //doubling step

    (b) if $n_i = 1$ then $f \leftarrow f \cdot g_{R,P}(Q),\ R \leftarrow R + P.$ //addition step

2. $f \leftarrow f^{(q^k-1)/n}.$

Note that pairings can be combined with windowing methods by replacing the computation in step (b) by

$$f \leftarrow f \cdot f_{c,P}(Q) \cdot g_{R,[c]P}(Q),\ R \leftarrow R + [c]P,$$

where the current window in the binary representation of n corresponds to the value c. The Miller function $f_{c,P}$ is defined via $\operatorname{div}(f_{c,P}) = c(P) - ([c]P) - (c-1)(\mathcal{O})$. But windowing methods are rarely used because of the extra costs of $1\mathbf{M}$ for updating the variable f.

3. Formulas for Pairings on Weierstrass curves

An elliptic curve over $\mathbf{F}_q$ in short Weierstrass form is given by an equation of the form $y^2 = x^3 + a_4 x + a_6$ with $a_4, a_6 \in \mathbf{F}_q$. In this section we present new formulas for the addition and doubling step in Miller's algorithm that are faster than previous ones. Furthermore, we also cover the case of a non-affine base point.

The fastest formulas for doublings on Weierstrass curves are given in Jacobian coordinates (cf. the EFD [6]). A point is represented as $(X_1 : Y_1 : Z_1)$ which for $Z_1 \neq 0$ corresponds to the affine point (x_1, y_1) with $x_1 = X_1/Z_1^2$ and $y_1 = Y_1/Z_1^3$. 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 addition step compared with the usual representation $(X_1 : Y_1 : Z_1)$. If the intermediate storage is an issue or if $\mathbf{s}$ is not much smaller than $\mathbf{m}$, T_1 should not be cached. We present the formulas including T_1 below; the modifications to omit T_1 are trivial.

For $S \in \{R, P\}$, the function $g_{R,S}$ for Weierstrass curves is given as the fraction of the usual line functions by

$$g_{R,S}(X : Y : Z) = \frac{(YZ_0^3 - Y_0 Z^3) - \lambda(XZ_0^2 - X_0 Z^2)ZZ_0}{(X - cZ^2)Z},$$

where λ is the slope of the line through R and S (with multiplicities), $(X_0 : Y_0 : Z_0)$ is a point on the line, and c is the x-coordinate of $R + S$. 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 $\mathbf{F}_q$. The function is evaluated at a point $Q = (X_Q : Y_Q : Z_Q)$ defined over $\mathbf{F}_{q^k}$.

We assume that k is even. This allows us to use several improvements and speedups that are presented in [2] and [3]. As usual, let the field extension $\mathbf{F}_{q^k}$ be constructed via a quadratic subfield as $\mathbf{F}_{q^k} = \mathbf{F}_{q^{k/2}}(\alpha)$, with $\alpha^2 = \delta$ for a non-square $\delta \in \mathbf{F}_{q^{k/2}}$; and let Q be chosen to be of the form $Q = (x_Q : y_Q\alpha : 1)$ with $x_Q, y_Q \in \mathbf{F}_{q^{k/2}}$. The latter is enforced by choosing a point Q' on a quadratic twist of E over $\mathbf{F}_{q^{k/2}}$ and defining Q as the image of Q' under the twist isomorphism. The denominator of $g_{R,S}(Q)$ is given by $x_Q - c$ which is defined over the subfield $\mathbf{F}_{q^{k/2}}$. Thus only the numerator needs to be considered as all multiplicative contributions from proper subfields of $\mathbf{F}_{q^k}$ are mapped to 1 by the final exponentiation and can be discarded. Furthermore, for addition and doubling in Jacobian coordinates we can write $\lambda = L_1/Z_3$, where Z_3 is the z-coordinate of $R + S$ and L_1 depends on R and S. Since Z_3 is defined over $\mathbf{F}_q$, we can instead compute $Z_3(y_Q Z_0^3\alpha - Y_0) - L_1(x_Q Z_0^2 - X_0)Z_0$ giving $g_{R,S}$ up to factors from subfields of $\mathbf{F}_{q^k}$.

3.1. Addition steps

In Miller's algorithm, all additions involve the base point as one input point so, when computing the line function, $(X_0 : Y_0 : Z_0)$ can be chosen as the base point P and all values depending solely on P and Q can be precomputed at the beginning of the computation. For additions, P is always stated as the second summand, i.e. $P = (X_2 : Y_2 : Z_2)$.

To enable an $\mathbf{m} - \mathbf{s}$ tradeoff we compute $2g_{R,P}(Q)$; this does not change the result of the computation since $2 \in \mathbf{F}_q$. Multiplications with x_Q and y_Q cost $(k/2)\mathbf{m}$ each; for $k > 2$ it is thus useful to rewrite the line function as
$$l = Z_3 \cdot 2y_Q Z_2^3\alpha - 2Z_3 \cdot Y_2 - L_1 \cdot (2(x_Q Z_2^2 - X_2)Z_2),$$
needing $(k+1)\mathbf{m}$ for precomputed $y_Q' = 2y_Q Z_2^3\alpha$ and $x_Q' = 2(x_Q Z_2^2 - X_2)Z_2$. Additionally $1\mathbf{M}$ is needed to update the variable f in Miller's algorithm.

Full addition. We use Bernstein and Lange's formulas ("add-2007-bl") from the EFD [6]. We can cache all values depending solely on P. In particular we precompute (or cache after the first addition or doubling) $R_2 = Y_2^2$ and $S_2 = T_2 \cdot Z_2$. The numerator of λ is $L_1 = D - C$.

$$
\begin{aligned}
A &= X_1 \cdot T_2;\ B = X_2 \cdot T_1;\ C = 2Y_1 \cdot S_2;\ D = ((Y_2 + Z_1)^2 - R_2 - T_1) \cdot T_1; \\
H &= B - A;\ I = (2H)^2;\ J = H \cdot I;\ L_1 = D - C;\ V = A \cdot I; \\
X_3 &= L_1^2 - J - 2V;\ Y_3 = L_1 \cdot (V - X_3) - 2C \cdot J;\ Z_3 = ((Z_1 + Z_2)^2 - T_1 - T_2) \cdot H; \\
T_3 &= Z_3^2;\ l = Z_3 \cdot y_Q' - (Y_2 + Z_3)^2 + R_2 + T_3 - L_1 \cdot x_Q'.
\end{aligned}
$$

The formulas need $1\mathbf{M} + (k+9)\mathbf{m} + 6\mathbf{s}$ to compute the addition step. To our knowledge this is the first set of formulas for full (non-mixed) addition. If $\mathbf{m}$ is not significantly more expensive than $\mathbf{s}$, some computations should be performed differently. In particular, R_2 needs not be stored, D is computed as $D = 2Y_2 \cdot Z_1 \cdot T_1$, l contains the term $-2Y_2 \cdot Z_3$ instead of $-(Y_2 + Z_3)^2 + R_2 + T_3$, and the computation of Z_3 can save some field additions.

If the values T_1, R_2, S_2, T_2, x_Q', and y_Q' cannot be stored, different optimizations are needed; in particular the line function is computed as
$$l = ((Z_3 \cdot Z_2) \cdot Z_2^2) \cdot y_Q\alpha - Y_2 \cdot Z_3 - (L_1 \cdot Z_2) \cdot Z_2^2 \cdot x_Q + X_2 \cdot (L_1 \cdot Z_2)$$
and the computation costs end up as $1\mathbf{M} + (k+17)\mathbf{m} + 6\mathbf{s}$.

Mixed addition. Mixed addition means that the second input point is in affine representation. Mixed additions occur in scalar multiplication if the base point P is given as $(x_2 : y_2 : 1)$.

We now state the mixed addition formulas based on Bernstein and Lange's formulas ("add-2007-bl") from the EFD [6]. Mixed additions are the usual case studied for pairings and the evaluation of the line function 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.

$$
\begin{aligned}
B &= x_2 \cdot T_1;\ D = ((y_2 + Z_1)^2 - R_2 - T_1) \cdot T_1;\ H = B - X_1;\ I = H^2;\ E = 4I;\ J = H \cdot E; \\
L_1 &= (D - 2Y_1);\ V = X_1 \cdot E;\ X_3 = L_1^2 - J - 2V;\ Y_3 = r \cdot (V - X_3) - 2Y_1 \cdot J; \\
Z_3 &= (Z_1 + H)^2 - T_1 - I;\ T_3 = Z_3^2;\ l = 2Z_3 \cdot y_Q\alpha - (y_2 + Z_3)^2 + R_2 + T_3 - 2L_1 \cdot (x_Q - x_2).
\end{aligned}
$$

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

3.2. Doubling steps

The main differences between the addition and the doubling formulas are that the doubling formulas depend on the curve coefficients and that the point $(X_0 : Y_0 : Z_0)$ appearing in the definition of $g_{R,S}$ is $(X_1 : Y_1 : Z_1)$, which is changing at every step. So in particular $Z_0 \neq 1$ and no precomputations (like x'_Q or y'_Q in the addition step) can be done.

For arbitrary a_4 the equation of the slope is $\lambda = (3X_1^2 + a_4 Z_1^4)/(2Y_1 Z_1) = (3X_1^2 + a_4 Z_1^4)/Z_3$. 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 most $(k + 3)\mathbf{m} + 1\mathbf{s}$ for arbitrary a_4 and with slightly less operations otherwise.

The formulas by Ionica and Joux [23] 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 $a_4 = 0$.

Doubling on curves with $a_4 = -3$. The fastest doubling formulas are due to Bernstein (see [6] "dbl-2001-b") and need $3\mathbf{m} + 5\mathbf{s}$ for the doubling.

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

The complete doubling step thus takes $1\mathbf{M} + 1\mathbf{S} + (k + 6)\mathbf{m} + 5\mathbf{s}$. Note that $L_1 = C$.

Doubling on curves with $a_4 = 0$. 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 [6]) but the line functions make use of Z_1^2. Note further that here $L_1 = E = 3X_1^2$ is particularly simple.

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

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

4. Geometric interpretation of the group law on twisted Edwards curves

In this section K denotes a field of characteristic different from 2. A *twisted Edwards curve* over K is a curve given by an affine equation of the form $\mathrm{E}_{a,d} : ax^2 + y^2 = 1 + dx^2 y^2$ for $a, d \in K^*$ and $a \neq d$. Twisted Edwards curves were introduced by Bernstein et al. in [5] as a generalization of Edwards curves [7] which are included as $\mathrm{E}_{1,d}$. An addition law on points of the curve $\mathrm{E}_{a,d}$ is given by

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

The neutral element is $\mathcal{O} = (0, 1)$, 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.

Edwards curves received a lot of attention because the above addition can be computed very efficiently, resulting in highly efficient algorithms to carry out scalar multiplication, a basic tool for many cryptographic protocols.

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 $\mathrm{E}_{a\delta, d\delta}$ and $\mathrm{E}_{a,d}$. 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.

We now 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 [17] and [33].

Let $\mathbb{P}^2(K)$ be the two-dimensional projective space over K, and let $P = (X_0 : Y_0 : Z_0) \in \mathbb{P}^2(K)$ with $Z_0 \neq 0$. Let $L_{1,P}$ be the line through P and Ω_1, i.e. $L_{1,P}$ is defined by $Z_0 Y - Y_0 Z = 0$; and let $L_{2,P}$ be the line through P and Ω_2, i.e. $L_{2,P}$ is defined by $Z_0 X - X_0 Z = 0$.

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. Since the points $\Omega_1, \Omega_2, \mathcal{O}'$ are not on a line, a conic C passing through these points cannot be a double line and ϕ represents C uniquely up to multiplication by a scalar. Evaluating ϕ at Ω_1, Ω_2, and $\mathcal{O}'$, we see that a conic C through these points has the form

$$C : c_{Z^2}(Z^2 + YZ) + c_{XY} XY + c_{XZ} XZ = 0, \tag{1}$$

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

Theorem 1. *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 $\Omega_1, \Omega_2, \mathcal{O}'$, P_1, and P_2, i.e. C is given by an equation of the form (1). 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 (1) 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

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

(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

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

Proof. 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}

$$\begin{aligned}
c_{Z^2}(Z_1^2 + Y_1 Z_1) + c_{XY} X_1 Y_1 + c_{XZ} X_1 Z_1 &= 0, \\
c_{Z^2}(Z_2^2 + Y_2 Z_2) + c_{XY} X_2 Y_2 + c_{XZ} X_2 Z_2 &= 0.
\end{aligned}$$

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

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

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

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

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

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

using the curve equation $aX_1^2 + Y_1^2 = 1 + dX_1^2 Y_1^2$ 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 $c_{Z^2} Z_1 + c_{XY} X_1 = 0$ and the coefficients as in (b). $\square$

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

$$
l_1 = Z_3 Y - Y_3 Z, \quad l_2 = X
$$

be the polynomials of the horizontal line L_{1,P_3} through P_3 and the vertical line $L_{2,\mathcal{O}}$ through $\mathcal{O}$ respectively, 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 1. The following theorem shows that the group law on a twisted Edwards curve indeed has a geometric interpretation involving the above equations. It gives us an important ingredient to compute Miller functions.

Theorem 2. *Let $a, d \in K^*$ with $a \neq d$ and let $\mathrm{E}_{a,d}$ be a twisted Edwards curve over K. Let $P_1, P_2 \in \mathrm{E}_{a,d}(K)$. Define $P_3 = P_1 + P_2$. Let ϕ, l_1, l_2 be defined as above. Then we have*

$$
\mathrm{div}\left(\frac{\phi}{l_1 l_2}\right) \sim (P_1) + (P_2) - (P_3) - (\mathcal{O}). \tag{2}
$$

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 [18, 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 eighth 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 $\mathrm{E}_{a,d}$ [33, 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 $\mathrm{div}\left(\frac{\phi}{l_1 l_2}\right) \sim (P_1) + (P_2) - (P_3) - (\mathcal{O})$.
$\square$

Remark 3. *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 $E_{a,d}$ and the conic C passing through $\Omega_1, \Omega_2, \mathcal{O}', P_1$ and P_2.*

Example 4. *As an example we consider the Edwards curve $E_{1,-30} : x^2 + y^2 = 1 - 30x^2y^2$ over the set of real numbers $\mathbb{R}$. 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) $P_1 = P_2 \neq \mathcal{O}'$, $P_3 = 2P_1$
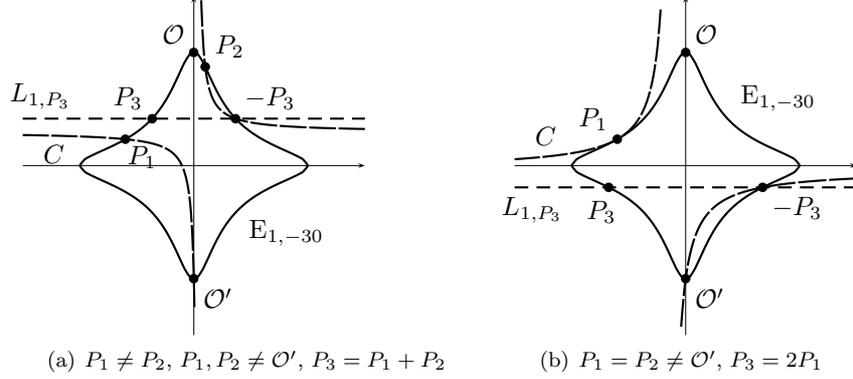
Figure 1: Geometric interpretation of the group law on $x^2 + y^2 = 1 - 30x^2y^2$ over $\mathbb{R}$.

5. Formulas for Pairings on Edwards Curves

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

By Theorem 2 we have $g_{R,S} = \frac{\phi}{l_1 l_2}$. In each step of the Miller loop first $g_{R,S}$ is computed, it is then evaluated at $Q = (X_0\alpha : Y_0 : Z_0)$ and finally f is updated 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

$$
\begin{aligned}
\frac{\phi}{l_1 l_2}(X_0\alpha : Y_0 : Z_0) &= \frac{c_{Z^2}(Z_0^2 + Y_0 Z_0) + c_{XY} X_0 \alpha Y_0 + c_{XZ} X_0 Z_0 \alpha}{(Z_3 Y_0 - Y_3 Z_0) X_0 \alpha} \\
&= \frac{c_{Z^2} \frac{Z_0 + Y_0}{X_0 \delta} \alpha + c_{XY} y_0 + c_{XZ}}{Z_3 y_0 - Y_3}, \\
&\in (c_{Z^2} \eta \alpha + c_{XY} y_0 + c_{XZ}) \mathbf{F}_{p^{k/2}}^*,
\end{aligned}
$$

where $(X_3 : Y_3 : Z_3)$ are coordinates of the point $R + P$ or $R + R$, $y_0 = Y_0/Z_0$, and $\eta = \frac{Z_0 + Y_0}{X_0 \delta}$. Note that $\eta, y_0 \in \mathbf{F}_{q^{k/2}}$ and that they are fixed for the whole computation, so they can be precomputed. The coefficients c_{Z^2}, c_{XY}, and c_{XZ} are defined over $\mathbf{F}_q$, thus the evaluation at Q given the coefficients of the conic can be computed in $k\mathbf{m}$ (multiplications by η and y_0 need $\frac{k}{2}\mathbf{m}$ each).

5.1. Addition steps

Hisil et al. presented new addition formulas for twisted Edwards curves in extended Edwards form at Asiacrypt 2008 [22]. Let $P_3 = P_1 + P_2$ for two different points $P_1 = (X_1 : Y_1 : Z_1 : T_1)$ and $P_2 = (X_2 : Y_2 : Z_2 : T_2)$ with $Z_1, Z_2 \neq 0$ and $T_i = X_i Y_i / Z_i$. Theorem 1 (a) states the coefficients of the conic section for addition. We use T_1, T_2 to shorten the formulas.

$$
\begin{aligned}
c_{Z^2} &= X_1 X_2 (Y_1 Z_2 - Y_2 Z_1) = Z_1 Z_2 (T_1 X_2 - X_1 T_2), \\
c_{XY} &= Z_1 Z_2 (X_1 Z_2 - Z_1 X_2 + X_1 Y_2 - Y_1 X_2), \\
c_{XZ} &= X_2 Y_2 Z_1^2 - X_1 Y_1 Z_2^2 + Y_1 Y_2 (X_2 Z_1 - X_1 Z_2) \\
&= Z_1 Z_2 (Z_1 T_2 - T_1 Z_2 + Y_1 T_2 - T_1 Y_2).
\end{aligned}
$$

8

Note that all coefficients are divisible by $Z_1 Z_2 \neq 0$ 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:

$$
\begin{aligned}
A &= X_1 \cdot X_2; \ B = Y_1 \cdot Y_2; \ C = Z_1 \cdot T_2; \ D = T_1 \cdot Z_2; \ E = D + C; \\
F &= (X_1 - Y_1) \cdot (X_2 + Y_2) + B - A; \ G = B + aA; \ H = D - C; \ I = T_1 \cdot T_2; \\
c_{Z^2} &= (T_1 - X_1) \cdot (T_2 + X_2) - I + A; \ c_{XY} = X_1 \cdot Z_2 - X_2 \cdot Z_1 + F; \\
c_{XZ} &= (Y_1 - T_1) \cdot (Y_2 + T_2) - B + I - H; \\
X_3 &= E \cdot F; \ Y_3 = G \cdot H; \ T_3 = E \cdot H; \ Z_3 = F \cdot G.
\end{aligned}
$$

With these formulas P_3 and $(c_{Z^2}, c_{XY}, c_{XZ})$ can be computed in $1\mathbf{M} + (k+14)\mathbf{m} + 1\mathbf{m_a}$, where $\mathbf{m_a}$ denotes the costs of a multiplication by a. If the base point P_2 has $Z_2 = 1$, the above costs reduce to $1\mathbf{M} + (k+12)\mathbf{m} + 1\mathbf{m_a}$. We used Sage [34] to verify the explicit formulas.

5.2. Doubling steps

Theorem 1 (c) states the coefficients of the conic section in the case of a doubling step. To speed up the computation we multiply each coefficient by $-2Y_1/Z_1$; remember that ϕ is 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:

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

Of course we also need to compute $P_3 = 2P_1$. We use the explicit formulas from [5] for the doubling and reuse subexpressions in computing the coefficients of the conic. The formulas were checked for correctness with Sage [34]. Since 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

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

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 $1\mathbf{M} + 1\mathbf{S} + (k+6)\mathbf{m} + 5\mathbf{s} + 2\mathbf{m_a}$ as

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

Note that like in [22] we can save $1\mathbf{m_a}$ per doubling by changing to the extended representation only before an addition.

6. Operation counts

We give an overview of the best formulas in the literature for computing the Tate pairing on Edwards curves and on the different forms of Weierstrass curves in Jacobian coordinates. We compare the results with our new pairing formulas for Weierstrass and Edwards curves.

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 so that multiplications with η and y_Q take $(k/2)\mathbf{m}$ each.

6.1. Overview

Chatterjee, Sarkar, and Barua [8] study pairings on Weierstrass curves in Jacobian coordinates. Their paper does not distinguish between multiplications in $\mathbf{F}_q$ and in $\mathbf{F}_{q^k}$ but their results are easily translated. For mixed addition steps their formulas need $1\mathbf{M} + (k+9)\mathbf{m} + 3\mathbf{s}$, and for doubling steps they need $1\mathbf{M} + (k+7)\mathbf{m} + 1\mathbf{S} + 4\mathbf{s}$ if $a_4 = -3$. For doubling steps on general Weierstrass curves (no condition on a_4) the formulas by Ionica and Joux [23] are fastest with $1\mathbf{M} + (k+1)\mathbf{m} + 1\mathbf{S} + 11\mathbf{s}$.

Actually, any mixed addition step (mADD) or addition step (ADD) in Miller's algorithm needs $1\mathbf{M} + k\mathbf{m}$ for the evaluation at Q and the update of f; each doubling step (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 [21] study pairing computation on Barreto-Naehrig [4] curves. All BN curves have the form $y^2 = x^3 + a_6$ and are thus more special than curves with $a_4 = -3$ or Edwards curves. They need $6\mathbf{m} + 5\mathbf{s}$ for a doubling step and $9\mathbf{m} + 3\mathbf{s}$ for a mixed addition step. Very recently, Costello et al. [11] presented explicit formulas for pairings on curves of the form $y^2 = x^3 + b^2$, i.e. $a_4 = 0$ and a_6 is a square. Their representation is in projective rather than Jacobian coordinates.

To the best of our knowledge our paper is the first to publish full (non-mixed) addition formulas for Weierstrass curves. Note that [11] started after our results became public.

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 [23] 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 $\mathcal{E}$, projective coordinates by $\mathcal{P}$, and Jacobian coordinates by $\mathcal{J}$. Morain [27] showed that 2-isogenies reach $a = 1$ from any twisted Edwards curve; we therefore omit $\mathbf{m_a}$ in the table.

	DBL	mADD	ADD
$\mathcal{J}$, [23], [8]	$1\mathbf{m} + 11\mathbf{s} + 1\mathbf{m_{a_4}}$	$9\mathbf{m} + 3\mathbf{s}$	—
$\mathcal{J}$, [23], this paper	$1\mathbf{m} + 11\mathbf{s} + 1\mathbf{m_{a_4}}$	$6\mathbf{m} + 6\mathbf{s}$	$9\mathbf{m} + 6\mathbf{s}$
$\mathcal{J}, a_4 = -3$, [8]	$7\mathbf{m} + 4\mathbf{s}$	$9\mathbf{m} + 3\mathbf{s}$	—
$\mathcal{J}, a_4 = -3$, this paper	$6\mathbf{m} + 5\mathbf{s}$	$6\mathbf{m} + 6\mathbf{s}$	$9\mathbf{m} + 6\mathbf{s}$
$\mathcal{J}, a_4 = 0$, [9], [8]	$6\mathbf{m} + 5\mathbf{s}$	$9\mathbf{m} + 3\mathbf{s}$	—
$\mathcal{J}, a_4 = 0$, this paper	$3\mathbf{m} + 8\mathbf{s}$	$6\mathbf{m} + 6\mathbf{s}$	$9\mathbf{m} + 6\mathbf{s}$
$\mathcal{P}, a_4 = 0, a_6 = b^2$ [11]	$3\mathbf{m} + 5\mathbf{s}$	$10\mathbf{m} + 2\mathbf{s} + 1\mathbf{m_b}$	$13\mathbf{m} + 2\mathbf{s} + 1\mathbf{m_b}$
$\mathcal{E}$, [23]	$8\mathbf{m} + 4\mathbf{s} + 1\mathbf{m_d}$	$14\mathbf{m} + 4\mathbf{s} + 1\mathbf{m_d}$	—
$\mathcal{E}$, this paper	$6\mathbf{m} + 5\mathbf{s}$	$12\mathbf{m}$	$14\mathbf{m}$

6.2. Comparison

The overview shows that our new formulas for Edwards curves solidly beat all previous formulas published for Tate pairing computation on Edwards curves.

Our new formulas for pairings on arbitrary Edwards curves are faster than all formulas previously known for Weierstrass curves except for the very special curves with $a_4 = 0$. Specifically mixed additions on Edwards curves are slower by some $\mathbf{s} - \mathbf{m}$ tradeoffs but doublings are much more frequent and gain at least an $\mathbf{s} - \mathbf{m}$ tradeoff each.

The curves considered in [11] are extremely special: For $p \equiv 2 \bmod 3$ these curves are supersingular and thus have $k = 2$. For $p \equiv 1 \bmod 3$ a total of 3 isomorphism classes is covered by this curve shape. They have faster doublings but slower additions and mixed additions than Edwards curves.

Our own improvements to the doubling and addition formulas for Weierstrass curves beat our new formulas for Edwards curves with affine base point by several $\mathbf{s} - \mathbf{m}$ tradeoffs. However, in many protocols the pairing input P is the output of some scalar multiplication and is thus naturally provided in non-affine

form. Whenever converting P to affine form is more expensive than proceeding in non-affine form, all additions are full additions. A full addition on an Edwards curve needs one field operation less than on Weierstrass curves. Depending on the frequency of addition and the $\mathbf{s}/\mathbf{m}$ ratio the special curves with $a_4 = 0$ might or might not be faster. For all other curves, Edwards form is the best representation. Furthermore, scalar multiplications on Edwards curves are significantly faster than on Weierstrass curves.

Our new formulas for mixed addition steps (mADD) and doubling steps (DBL) on Weierstrass curves are faster than all previous ones by several $\mathbf{s} - \mathbf{m}$ tradeoffs. Our formulas for full addition (ADD) are the only ones in the literature for most Weierstrass curves; for those with $a_4 = 0$ and $a_6 = b^2$ they are faster than those in [11] for any $\mathbf{s}/\mathbf{m}$ ratio.

We note here that for curves in Weierstrass form the ate pairing is more efficient than the Tate pairing, in particular when the R-ate pairing or optimal pairings with a very short loop in Miller's algorithm are computed, and when twists of degree 4 and 6 are used to represent torsion points. Our comparison only refers to Tate pairing computation.

Further research needs to focus on how to compute variants of the ate pairing on Edwards curves. To obtain the same or better efficiency as the fastest pairings on Weierstrass curves, it needs to be clarified whether optimal ate pairings can be computed and whether the above mentioned high-degree twists can be used as well for suitable pairing-friendly curves in Edwards form. Some initial results are presented in [12].

7. Construction of Pairing-Friendly Edwards Curves

The previous chapter showed that pairing computation can benefit from Edwards curves. Most constructions of pairing-friendly elliptic curves in the literature aim at a prime group order and thus in particular do not lead to curves with cofactor 4 that can be transformed to Edwards curves. Galbraith, McKee, and Valença [19] showed how to use the MNT construction to produce curves with small cofactor. Some other constructions that allow to find curves with cofactor divisible by 4 are described by Freeman, Scott, and Teske [16].

To ensure security of the pairing based system two criteria must be satisfied: The group $E(\mathbf{F}_p)$ must have a large enough prime order subgroup so that generic attacks are excluded *and* p^k must be large enough so that index calculus attacks in $\mathbf{F}_{p^k}^*$ are excluded. For efficient implementation, we try to minimize p and k to minimize the cost of arithmetic in $\mathbf{F}_p$ and $\mathbf{F}_{p^k}$ and minimize n to minimize the length of the Miller loop. This has the effect of balancing the difficulty of the DLPs on the curve and in the multiplicative group of the finite field $\mathbf{F}_{p^k}$.

Following the ECRYPT recommendations [14], the "optimal" bitsizes of the primes p and n for curves $E/\mathbf{F}_p$ with $n \mid \#E(\mathbf{F}_p)$ 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 $\mathbf{F}_{p^k}^*$. In order to transform the curve to an Edwards curve, we need to have $\#E(\mathbf{F}_p) = 4hn$ for some cofactor h. It follows that the rho-value $\rho = \log(p)/\log(n)$ of E is always larger than 1. The recommendations imply a desired value for $\rho \cdot k$ as displayed in Table 1, which should be achieved with an even embedding degree to favor efficient implementation. This means that p cannot be kept minimal but we managed to minimize n to keep the Miller loop short.

In the following section we present six examples of pairing-friendly Edwards curves with embedding degrees $k \in \{6, 8, 10, 22\}$, which cover the security levels given in Table 1.

security	80	96	112	128	160	256
$\log_2(n)$	160	192	224	256	320	512
$\log_2(p^k)$	1248	1776	2432	3248	4800	15424
$\rho \cdot k$	7.80	9.25	10.86	12.67	15	30.13

Table 1: "Optimal" bitsizes for the primes n and p and the corresponding values for $\rho \cdot k$ for most common security levels.

8. Examples of Pairing-Friendly Edwards Curves

This section presents pairing-friendly Edwards curves. Note that they were constructed for applications using the Tate pairing so that the curve over the ground field has a point of order 4. They are all defined over a prime field $\mathbf{F}_p$, and the ρ values are stated with the curves. Notation is as before, where the number of $\mathbf{F}_p$-rational points on the curve is $4hn$.

The curve examples in this section cover the security levels in Table 1. We used the method and formula in [14] to determine the effective security in bits on the curve and in the finite field.

8.0.1. Security level 80 bits (generic: 82 bits, index calculus: 79 bits):

$k = 6, \rho = 1.22$ following [19]:

$D = 7230$, $\lceil \log(n) \rceil = 165$, $\lceil \log(h) \rceil = 34$, $\lceil \log(p) \rceil = 201$, $k \lceil \log(p) \rceil = 1206$

$$
\begin{aligned}
p &= 2051613663768129606093583432875887398415301962227490187508801, \\
n &= 4481254541330857991395743820133138543474342366277, \\
h &= 7 \cdot 733 \cdot 2230663, \\
d &= 1100661309421493056836745159318889208210931380459417578976626.
\end{aligned}
$$

8.0.2. Security level 96 bits (generic: 95 bits, index calculus: 93 bits):

$k = 6, \rho = 1.48$ following [19]:

$D = 4630$, $\lceil \log(n) \rceil = 191$, $\lceil \log(h) \rceil = 90$, $\lceil \log(p) \rceil = 283$, $k \lceil \log(p) \rceil = 1698$

$$
\begin{aligned}
p &= 12076422473257620999622772924220230535655104285600826357856070179619031510615886361601, \\
n &= 2498886235887409414948289020220476887707263210939845485839, \\
h &= 11161 \cdot 19068349 \cdot 5676957216676051, \\
d &= 2763915426899189358845059350727381504946815286189972438681082636399984067165911590884.
\end{aligned}
$$

8.0.3. Security level 112 bits (generic: 112 bits, index calculus: 117 bits):

$k = 8, \rho = 1.50$ following Example 6.10 in [16]:

$D = 1$, $\lceil \log(n) \rceil = 224$, $\lceil \log(h) \rceil = 111$, $\lceil \log(p) \rceil = 337$, $k \lceil \log(p) \rceil = 2696$

$$
\begin{aligned}
p &= 23377366536991056692603839001569188814245474692929568668962591328909094370357 \\
  &\quad 23487560287788744816042 89 \\
n &= 22985796260053765810955211899935144604417092746113717429138553265289 \\
h &= 315669989 \cdot 558193107149 \cdot 14429732414341 \\
d &= 21373841441636012883551957246343228553489584548232523879997636200280796159999 98 \\
  &\quad 48556640836158104712032
\end{aligned}
$$

8.0.4. Security level 128 bits (generic: 133 bits, index calculus: 127 bits):

$k = 8, \rho = 1.50$ following Example 6.10 in [16]:

$D = 1$, $\lceil \log(n) \rceil = 267$, $\lceil \log(h) \rceil = 133$, $\lceil \log(p) \rceil = 401$, $k\lceil \log(p) \rceil = 3208$

$$
\begin{aligned}
p &= 51065000030527450626711027753965666498558576769353848475638203214584974495354436071209268470508469629312810691036880709, \\
n &= 83370304250867884451007046717638964825493974378500426335965601180405626415044337, \\
h &= 5 \cdot 17 \cdot 1229 \cdot 3181 \cdot 460805316477868978561389227734187, \\
d &= 2553250001526372531335551387698283324927928838467692423781910160729248724767721803560463423525423481465640534551844035587, \\
\end{aligned}
$$

8.0.5. Security level 160 bits (generic: 164 bits, index calculus: 154 bits):

$k = 10, \rho = 1.49$ following Construction 6.5 in [16]:

$D = 1$, $\lceil \log(n) \rceil = 328$, $\lceil \log(h) \rceil = 160$, $\lceil \log(p) \rceil = 490$, $k\lceil \log(p) \rceil = 4900$

$$
\begin{aligned}
p &= 319667071934078971315677746964738362812713703914060344412320604868708613896665173327525254333020975442799087510187984142542764611515759451562949124937, \\
n &= 546812704438652190176048473638362779688423061794499756311925945545462152449512232744941959488864241, \\
h &= 2^4 \cdot 70199^4 \cdot 7831391^4, \\
d &= 366838958032886838857360394166535857747556934852621175164120734346101628194129743602008259319768868802620569094456792293200142806009932471922115210.
\end{aligned}
$$

8.0.6. Security level 256 bits (generic: 259 bits, index calculus: 259 bits):

$k = 22, \rho = 1.39$ following Construction 6.6 in [16]:

$D = 3$, $\lceil \log(n) \rceil = 519$, $\lceil \log(h) \rceil = 204$, $\lceil \log(p) \rceil = 724$, $k\lceil \log(p) \rceil = 15928$

$$
\begin{aligned}
p &= 79324390783653822510191966358195377091376558066284959420357463687451883685827055516014492098382728038686154339121902148247413729605337155986911218807161824591404393367767771926661771139435864150449118516697852906546951237, \\
n &= 9621311878085603778985691952625727109889848694647550025094596661780692626283672821912529731051013737049538186606705506586597903896379176063425017329234863697, \\
h &= 3^5 \cdot 7 \cdot 13^2 \cdot 19^2 \cdot 37^2 \cdot 6421^2 \cdot 7219 \cdot 3498559^2 \cdot 22526869^2 \cdot 78478074679, \\
d &= 26441462754793978081083982672739538325998744498135256075358287708632007468065063378057192037361551803250920085233286421641304132894986501666675972821801945609720468771083104817656092016879614901160245443945786256399518.
\end{aligned}
$$

References

[1] 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.

[2] Paulo S. L. M. Barreto, Hae Yong Kim, Ben Lynn, and Michael Scott. Efficient algorithms for pairing-based cryptosystems. In *CRYPTO 2002 [36]*, pages 354–368, 2002.

[3] Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. Efficient implementation of pairing-based cryptosystems. *J. Cryptology*, 17:321–334, 2004.

[4] Paulo S. L. M. Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In *SAC 2005 [31]*, pages 319–331, 2006.

[5] Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. Twisted Edwards curves. In *Africacrypt [35]*, pages 389–405, 2008. http://cr.yp.to/papers.html#twisted.

[6] Daniel J. Bernstein and Tanja Lange. Explicit-formulas database. http://www.hyperelliptic.org/EFD.

[7] Daniel J. Bernstein and Tanja Lange. Faster addition and doubling on elliptic curves. In *ASIACRYPT 2007 [25]*, pages 29–50, 2007. http://cr.yp.to/newelliptic/.

[8] Sanjit Chatterjee, Palash Sarkar, and Rana Barua. Efficient computation of Tate pairing in projective coordinate over general characteristic fields. In *ICISC 2004 [29]*, pages 168–181, 2005.

[9] Zhaohui Cheng and Manos Nistazakis. Implementing pairing-based cryptosystems. In *3rd International Workshop on Wireless Security Technologies IWWST-2005*, 2005.

[10] 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.

[11] Craig Costello, Huseyin Hisil, Colin Boyd, Juan Manuel Gonzalez Nieto, and Kenneth Koon-Ho Wong. Faster pairings on special Weierstrass curves. In *Pairing 2009 [32]*, pages 89–101, 2009.

[12] Craig Costello, Tanja Lange, and Michael Naehrig. Faster pairing computations on curves with high-degree twists. In *PKC 2010 [28]*, pages 224–242, 2010.

[13] M. Prem Laxman Das and Palash Sarkar. Pairing computation on twisted Edwards form elliptic curves. In *Pairing 2008 [20]*, pages 192–210, 2008.

[14] Nigel Smart (editor). ECRYPT2 yearly report on algorithms and keysizes (2008-2009). Technical report, ECRYPT II – European Network of Excellence in Cryptology, EU FP7, ICT-2007-216676, 2009. published as deliverable D.SPA.7 http://www.ecrypt.eu.org/documents/D.SPA.7.pdf.

[15] 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.

[16] David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. Cryptology ePrint Archive, Report 2006/372, 2006. update 2008, http://eprint.iacr.org/2006/372, to appear in Journal of Cryptology.

[17] Gerhard Frey and Tanja Lange. *Background on Curves and Jacobians*, chapter 4 in [1], pages 45–85. 2005.

[18] William Fulton. *Algebraic Curves*. W. A. Benjamin, Inc., 1969.

[19] 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.

[20] 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. Springer.

[21] Darrel Hankerson, Alfred J. Menezes, and Michael Scott. Software implementation of pairings. In *Identity-Based Cryptography [24]*, pages 188–206, 2009.

[22] Huseyin Hisil, Kenneth Koon-Ho Wong, Gary Carter, and Ed Dawson. Twisted Edwards curves revisited. In *ASIACRYPT 2008 [30]*, pages 326–343, 2008.

[23] Sorina Ionica and Antoine Joux. Another approach to pairing computation in Edwards coordinates. In *INDOCRYPT 2008 [10]*, pages 400–413, 2008. http://eprint.iacr.org/2008/292.

[24] Marc Joye and Gregory Neven, editors. *Identity-Based Cryptography*, volume 2 of *Cryptology and Information Security Series*. IOS Press, 2009.

[25] Kaoru Kurosawa, editor. *Advances in Cryptology — ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, Berlin Heidelberg, 2007. Springer.

[26] Victor S. Miller. The Weil pairing, and its efficient calculation. *Journal of Cryptology*, 17(4):235–261, 2004.

[27] Francois Morain. Edwards curves and CM curves. Technical report, arXiv, 2009.

[28] Phong Nguyen and David Pointcheval, editors. *13th International Conference on Practice and Theory in Public-Key Cryptography, Paris, France, May 26-28, 2010. Proceedings*, volume 6056 of *Lecture Notes in Computer Science*, Berlin, 2009. Springer.

[29] 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.

[30] 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.

[31] 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.

[32] Hovav Schacham and Brent Waters, editors. *Pairing-Based Cryptography - Pairing 2009, Third International Conference, Palo Alto, CA, USA, August 12-14, 2009, Proceedings*, volume 5671 of *Lecture Notes in Computer Science*, Berlin, 2009. Springer.

[33] Joseph H. Silverman. *The Arithmetic of Elliptic Curves*. Number 106 in Graduate texts in mathematics. Springer-Verlag, 1986.

[34] William Stein. Sage mathematics software (version 2.8.12), 2008. The Sage Group, http://www.sagemath.org.

[35] 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.

[36] 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.