Faster Pairing Computations on Curves with High-Degree Twists

Craig Costello[1], Tanja Lange[2], and Michael Naehrig[2]

[1] Information Security Institute
Queensland University of Technology, GPO Box 2434, Brisbane QLD 4001, Australia
craig.costello@qut.edu.au
[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. Research on efficient pairing implementation has focussed on reducing the loop length and on using high-degree twists. Existence of twists of degree larger than 2 is a very restrictive criterion but luckily constructions for pairing-friendly elliptic curves with such twists exist. In fact, Freeman, Scott and Teske showed in their overview paper that often the best known methods of constructing pairing-friendly elliptic curves over fields of large prime characteristic produce curves that admit twists of degree $3, 4$ or 6.
A few papers have presented explicit formulas for the doubling and the addition step in Miller's algorithm, but the optimizations were all done for the Tate pairing with degree-2 twists, so the main usage of the high-degree twists remained incompatible with more efficient formulas.
In this paper we present efficient formulas for curves with twists of degree $2, 3, 4$ or 6. These formulas are significantly faster than their predecessors. We show how these faster formulas can be applied to Tate and ate pairing variants, thereby speeding up all practical suggestions for efficient pairing implementations over fields of large characteristic.
Keywords: Pairings, Miller functions, explicit formulas, Tate pairing, ate pairing, twists, Weierstrass curves.

1 Introduction

Many new protocols are based on pairings and so the construction of pairing-friendly curves and the efficiency of pairing computation has become a field of active research. The first wave of this research exhausted many tricks that can be applied inside a Miller iteration, resulting in significant computational speed ups [4, 6, 7, 34]. The second wave of improvements focussed on constructing pairing-friendly elliptic curves [5, 11, 37, 16, 8, 22, 9, 17, 28], and this research is extended

* This work has been supported in part by a Queensland Government Smart State PhD Scholarship and in part by the European Commission through the ICT Programme under Contract ICT–2007–216646 ECRYPT II. Part of this work was done while the second author visited QUT.

and collected in [18]. The third and more recent wave of research has focussed on reducing the loop length of Miller's algorithm [35, 26, 3, 32] to be as short as possible [42, 25]. Along the way, there have been several other clever optimizations that give faster pairings in certain scenarios, including compressed pairings [36], single coordinate pairings [21], efficient methods of hashing to pairing-friendly groups [38], and techniques that achieve a faster final exponentiation [24, 39].

After the introduction of projective coordinates for pairing computations in [12], very little was heard about low level optimizations. This started to become more interesting lately for alternative curve shapes such as Edwards curves, studied in [14, 27, 2], and curves of the form $y^2 = x^3 + c^2$, studied in [13].

All of these improvements are presented in the context of the Tate pairing on curves with even embedding degrees and using only quadratic twists, since the nature of the Tate pairing allows for a relatively simple exposition and improves efficiency through denominator elimination. At the same time, curves with larger degree twists give much more efficient pairings and choosing special curve shapes was risking this larger benefit. On top of that, Galbraith [20] studied the group orders of curves and their twists and showed that for Edwards curves only quadratic twists could be used, in the sense that the only twist which preserves the existence of a point of order 4 is a quadratic twist. This deterred further research on ate pairings and other variants for special curves. In this paper we show that it is possible to compute a small power of the ate pairing entirely on the twisted curve; so the curve can be chosen so that the *twist* of the curve admits a particular shape. We show the fields of definition for the respective coordinates. This provides a framework for converting Tate-like pairing computation formulas and operation counts to their ate-like analogues.

For BN curves [8], Akane, Nogami, and Morikawa showed in [1] that the ate pairing itself can be computed on the twisted curve. Our result covers more general curves but computes the ate pairing only up to a power. Furthermore, the idea of using twists in order to cover curves of special shapes is new. In the context of Weierstrass curves, our result gives an easy way of computing the cost of evaluating the Miller function.

For all practically useful embedding degrees, the best methods of constructing pairing-friendly curves mostly produce elliptic curves of the form $y^2 = x^3 + ax + b$ with $a = 0$ or $b = 0$ (see [18]). In this paper we consider these two cases separately to give specialized pairing formulas in both scenarios. In particular, we achieve the fastest known formulas for computing pairings on general curves with $b = 0$ in weight-$(1, 2)$ coordinates. In addition, the point doubling formulas we derive for curves of this form are currently the fastest published point doubling formulas [10] across all forms of elliptic curves. For pairings on general curves with $a = 0$, we use standard projective coordinates. The doubling step on these curves is two field multiplications faster than the previous record for such curves. Furthermore, we also consider the case of computing pairings on curves with odd embedding degrees that employ cubic twists, where we present formulas which are significantly faster than their predecessors. Lastly, we also suggest an improvement to the formulas presented in [13]. Note that for ate pairings, speed

ups in the doubling and addition step save computations in fields whose sizes grow proportionately to the embedding degree. This means that applying these faster formulas to the ate pairing variants will give relative speed ups which are consistent across all embedding degrees and savings which do not suffer as the extension field arithmetic becomes more complex.

The rest of this paper is organized as follows. Section 2 provides a brief background on pairings. In Section 3, we present a modified method of computing the ate pairing where all operations involve points only on the twisted curve. This theoretical result is a key ingredient for efficient computation of the ate pairing and has applications outside the scope of this paper, e.g. for Edwards curves. We then show how Tate pairing formulas and operation counts can be easily modified to this method of computing the ate pairing. In Sections 4 and 5, we present faster formulas for pairing computations that employ quadratic, quartic or sextic twists. In Section 6, we present faster formulas for pairings on curves with odd embedding degrees divisible by 3. We compare our results with the state-of-the-art pairing formulas in Section 7.

2 Background on Pairings

Let $p > 3$ be a prime, and let E be an elliptic curve over $\mathbb{F}_q$, $\mathrm{char}(\mathbb{F}_q) = p$, with short Weierstrass equation $E : y^2 = x^3 + ax + b$ and point at infinity $\mathcal{O}$. Let $r \neq p$ be a prime divisor of $n = \#E(\mathbb{F}_q) = q + 1 - t$ and let $k > 1$ be the embedding degree of E with respect to r, i. e. k is minimal with $r \mid q^k - 1$. For the r-torsion subgroup, we have $E[r] \subseteq E(\mathbb{F}_{q^k})$. Let $\mu_r \subseteq \mathbb{F}_{q^k}^*$ be the group of r-th roots of unity. For $m \in \mathbb{Z}$ and $P \in E[r]$, let $f_{m,P}$ be a function with divisor $\mathrm{div}(f_{m,P}) = m(P) - ([m]P) - (m-1)(\mathcal{O})$. The reduced Tate pairing is defined as

$$\tau_r : E(\mathbb{F}_{q^k})[r] \times E(\mathbb{F}_{q^k})/[r]E(\mathbb{F}_{q^k}) \to \mu_r,\ (P, Q) \mapsto f_{r,P}(Q)^{\frac{q^k-1}{r}}.$$

In practice one restricts the arguments to groups of prime order r. If $r^2 \nmid n$, the most common choice is to take the groups

$$G_1 = E[r] \cap \ker(\phi_q - [1]) = E(\mathbb{F}_q)[r],\ G_2 = E[r] \cap \ker(\phi_q - [q]) \subseteq E(\mathbb{F}_{q^k}),$$

where ϕ_q is the q-power Frobenius endomorphism on E. The groups G_1 and G_2 are the eigenspaces of ϕ_q on $E[r]$ and we have $E[r] = G_1 \oplus G_2$. From now on, we consider e_r, the reduced Tate pairing restricted to $G_1 \times G_2$, i. e.

$$e_r : G_1 \times G_2 \to \mu_r,\ (P, Q) \mapsto f_{r,P}(Q)^{\frac{q^k-1}{r}}.$$

Let $T = t - 1$. Restricting the Tate pairing to $G_2 \times G_1$ leads to the ate pairing [26]

$$a_T : G_2 \times G_1 \to \mu_r,\ (Q, P) \mapsto f_{T,Q}(P)^{\frac{q^k-1}{r}}.$$

Note that the parameter r is changed to T. The group G_2 consists of points defined over $\mathbb{F}_{q^k}$. Often G_2 can be represented by a subgroup G_2' of a curve

isomorphic to E over $\mathbb{F}_{q^k}$. Let $d \mid k$; an elliptic curve E' over $\mathbb{F}_{q^{k/d}}$ is called a twist of degree d of $E/\mathbb{F}_{q^{k/d}}$ if there is an isomorphism $\psi : E' \to E$ defined over $\mathbb{F}_{q^k}$, and this is the smallest extension of $\mathbb{F}_{q^{k/d}}$ over which ψ is defined. Depending on the j-invariant $j(E)$ of E, there exist twists of degree at most 6. Pairing-friendly curves with twists of degree higher than 2 arise from constructions with j-invariants $j(E) = 0$ and $j(E) = 1728$.

A twist of E is given by E': $y^2 = x^3 + a\omega^4 x + b\omega^6$ for some $\omega \in \mathbb{F}_{q^k}$. The isomorphism between E' and E is $\Psi : E' \to E : (x', y') \to (x'/\omega^2, y'/\omega^3)$ with inverse $\Psi^{-1} : E \to E' : (x, y) \to (\omega^2 x, \omega^3 y)$. Depending on $j(E)$ and ω, we obtain the possible degrees of a twist E' as summarized in Table 1. The isomorphism Ψ induces a group isomorphism $G_2' \to G_2$, where $G_2' = E'(\mathbb{F}_{q^{k/d}})[r]$. Thus, points in G_2 can be represented by their image under Ψ^{-1}. In what follows, we write P' for the point on the twist E' corresponding to a point $P \in E$, i. e. $P' = \Psi^{-1}(P)$ and $P = \Psi(P')$. The last two columns in Table 1 show the subfields of $\mathbb{F}_{q^k}$ in which the coordinates of the specific points are contained. For example $(\mathbb{F}_{q^{k/2}}, \mathbb{F}_{q^k})$ means that the x-coordinate is in $\mathbb{F}_{q^{k/2}}$ and the y-coordinate is in $\mathbb{F}_{q^k}$. The last column illustrates that the coordinates of P' lie in the same fields as the coordinates of Q. The importance of this becomes evident in Section 3. Since the points in G_2' are defined over a smaller field than those in G_2, curve arithmetic is more efficient in G_2'.

d	$j(E)$ a, b	fields of definition for powers of ω	$Q' = (x_{Q'}, y_{Q'})$ $P = (x_P, y_P)$	$Q = \Psi(Q')$ $P' = \Psi^{-1}(P)$
2	$\notin \{0, 1728\}$ $a \neq 0$, $b \neq 0$	$\omega^2, \omega^4, \omega^6 \in \mathbb{F}_{q^{k/2}}$ $\omega^3 \in \mathbb{F}_{q^k} \setminus \mathbb{F}_{q^{k/2}}$	$(\mathbb{F}_{q^{k/2}}, \mathbb{F}_{q^{k/2}})$ $(\mathbb{F}_q, \mathbb{F}_q)$	$(\mathbb{F}_{q^{k/2}}, \mathbb{F}_{q^k})$ $(\mathbb{F}_{q^{k/2}}, \mathbb{F}_{q^k})$
3	0 $a = 0$, $b \neq 0$	$\omega^6, \omega^3 \in \mathbb{F}_{q^{k/3}}$ $\omega^2 \in \mathbb{F}_{q^k} \setminus \mathbb{F}_{q^{k/3}}$	$(\mathbb{F}_{q^{k/3}}, \mathbb{F}_{q^{k/3}})$ $(\mathbb{F}_q, \mathbb{F}_q)$	$(\mathbb{F}_{q^k}, \mathbb{F}_{q^{k/3}})$ $(\mathbb{F}_{q^k}, \mathbb{F}_{q^{k/3}})$
4	1728 $a \neq 0$, $b = 0$	$\omega^4 \in \mathbb{F}_{q^{k/4}}$, $\omega^2 \in \mathbb{F}_{q^{k/2}}$ $\omega^3 \in \mathbb{F}_{q^k} \setminus \mathbb{F}_{q^{k/2}}$	$(\mathbb{F}_{q^{k/4}}, \mathbb{F}_{q^{k/4}})$ $(\mathbb{F}_q, \mathbb{F}_q)$	$(\mathbb{F}_{q^{k/2}}, \mathbb{F}_{q^k})$ $(\mathbb{F}_{q^{k/2}}, \mathbb{F}_{q^k})$
6	0 $a = 0$, $b \neq 0$	$\omega^6 \in \mathbb{F}_{q^{k/6}}$, $\omega^3 \in \mathbb{F}_{q^{k/3}}$ $\omega^2 \in \mathbb{F}_{q^{k/2}}$	$(\mathbb{F}_{q^{k/6}}, \mathbb{F}_{q^{k/6}})$ $(\mathbb{F}_q, \mathbb{F}_q)$	$(\mathbb{F}_{q^{k/2}}, \mathbb{F}_{q^{k/3}})$ $(\mathbb{F}_{q^{k/2}}, \mathbb{F}_{q^{k/3}})$

Table 1. The nature of the twist isomorphisms for twists of degree d.

Assume that E has a twist of degree d and that $d \mid k$. Let $e = k/d$, $T_e = T^e \bmod r$. The twisted ate pairing is defined as

$$\eta_{T_e} : G_1 \times G_2 \to \mu_r,\ (P, Q) \mapsto f_{T_e,P}(Q)^{\frac{q^k-1}{r}}.$$

The reduced Tate and the twisted ate pairing are both defined on $G_1 \times G_2$, while the ate pairing is defined on $G_2 \times G_1$. We aim to simultaneously treat both concepts of pairings by respectively fixing R and S as the first and second arguments of either pairing. For both variants, we thus write $f_{m,R}(S)^{(q^k-1)/r}$, where m, R, S are chosen according to the desired pairing. Miller's algorithm is

used to compute the pairing as follows: Let $m = (m_{l-1}, \ldots, m_1, m_0)_2$ be the binary representation of m, initialize $U = R$, $f = 1$ and compute

1. for $i = l - 2$ to 0 do
   (a) $f \leftarrow f^2 \cdot f_{\mathrm{DBL}(U)}(S)$, $U \leftarrow [2]U$, //doubling step (DBL)
   (b) if $m_i = 1$ then $f \leftarrow f \cdot f_{\mathrm{ADD}(U,R)}(S)$, //addition step (ADD)
   $U \leftarrow U + R$.
2. $f \leftarrow f^{(q^k-1)/r}$.

The function $f_{\mathrm{DBL}(U)}$ is defined as $f_{\mathrm{DBL}(U)} = l_{\mathrm{DBL}(U)}/v_{\mathrm{DBL}(U)}$, where $l_{\mathrm{DBL}(U)}$ is the function of the line tangent to E at the point U and $v_{\mathrm{DBL}(U)}$ is the function of the vertical line through $[2]U$. Analogously, the function $f_{\mathrm{ADD}(U,R)}$ is defined as $f_{\mathrm{ADD}(U,R)} = l_{\mathrm{ADD}(U,R)}/v_{\mathrm{ADD}(U,R)}$, where $l_{\mathrm{ADD}(U,R)}$ is the function of the line through the points U and R and $v_{\mathrm{ADD}(U,R)}$ is the function of the vertical line through $U+R$. If one of the inputs to the addition is given in affine representation we speak of a "mixed addition" and use the abbreviation mADD.

Step 1 in the above algorithm is called the Miller loop; it computes the function value $f_{m,R}(S)$ up to r-th powers. Step 2, the final exponentiation, determines the final pairing value.

The number of iterations of the Miller loop is equal to $l - 1$, where l is the bitlength of m. Therefore, reducing the bitlength of m reduces the number of iterations in the Miller loop which reduces the cost of the pairing computation. Several papers have proposed methods for loop shortening [30, 32, 43, 42, 25]. For example, for the twisted ate pairing one can replace T_e by any of its powers modulo r and choose the smallest of those. A good choice for the ate pairing is to use the R-ate pairing [30], which often achieves an optimal loop length of $\log(r)/\varphi(k)$, yielding an optimal pairing [42].

3 Computing the Ate Pairing Entirely on the Twisted Curve

Several authors have presented new formulas that achieve faster iterations of the Miller loop on certain curves [12, 14, 27, 2, 13]. The operation counts presented in these papers are given in the context of Tate pairing computations on curves with even embedding degrees, where all elliptic curve operations occur in the base field $\mathbb{F}_q$ and the functions in the Miller loop are evaluated at a point which has one coordinate in $\mathbb{F}_{q^{k/2}}$ and one in the full extension field $\mathbb{F}_{q^k}$. This allows for a relatively simple exposition. However, the ate pairing reverses the roles of the points involved and employs twisted curves. This means that some of the optimizations can not be applied in the same fashion. The purpose of this section is to tidy up this discussion and to show how operation counts for the Tate pairing can be easily modified to give the analogous ate pairing count.

The usual practice when computing the ate pairing $a_T(Q, P)$ of the points $P \in E(\mathbb{F}_q)$ and $Q \in E(\mathbb{F}_{q^k})$ is to map the point Q to the twisted curve using the isomorphism Ψ^{-1}, so that the point operations (doubling/addition) in the Miller loop can be performed more efficiently using the point $Q' = \Psi^{-1}(Q) \in E'(\mathbb{F}_{q^{k/d}})$,

whose coordinates are defined over the smaller field $\mathbb{F}_{q^{k/d}}$. When it is time to compute the Miller line, Q' is "untwisted" back to the full extension field via $Q = \Psi(Q')$. Operation counts for the Tate pairing do not carry over directly to the ate pairing. In particular, for the Tate pairing it is the y-coordinate of the second argument that is in the full extension field $\mathbb{F}_{q^k}$, whereas one of the coordinates of the first argument in the ate pairing is in $\mathbb{F}_{q^k}$. This means that all optimizations that were based on eliminating subfield elements have to be revised.

Furthermore, pairings on special curves such as Edwards curves and the curves in [13] pose conditions on cofactors of the group order. Galbraith [20] pointed out to us that for twists of degree larger than 2, E and E' can not both simultaneously be in Edwards form. His arguments also apply to the curves in [13] with sextic twists. So far this meant that the formulas used for the point operations and the formulas derived for the Miller functions must be treated separately which usually results in a greater overall operation count.

We show that a small (≤ 6) power of the ate pairing can be computed entirely on the twisted curve, rendering the above concerns obsolete. Our pairing can make use of loop shortening techniques just like the ate pairing, but only requires one curve (the twisted curve) to have particular properties. Furthermore, Table 1 shows that most coordinates of the twisted points P' and Q' are defined over subfields. Note that the computation of a small power of pairings for efficiency reasons has been addressed in previous work, see for example [15].

Theorem 1. *Let $E/\mathbb{F}_q : y^2 = x^3+ax+b$ and let $E'/\mathbb{F}_{q^{k/d}} : y^2 = x^3+a\omega^4x+b\omega^6$, a degree-$d$ twist of E. Let Ψ be the associated twist isomorphism $\Psi : E' \to E : (x', y') \to (x'/\omega^2, y'/\omega^3)$. Let $P \in G_1$, $Q \in G_2$, and let $Q' = \Psi^{-1}(Q)$ and $P' = \Psi^{-1}(P)$. Let $a_T(Q, P)$ be the ate pairing of Q and P. Then*

$$a_T(Q, P)^{\gcd(d,6)} = a_T(Q', P')^{\gcd(d,6)},$$

where $a_T(Q', P') = f_{T,Q'}(P')^{(q^k-1)/r}$ uses the same loop parameter as $a_T(Q, P)$ on E, but takes the two twisted points Q' and P' as inputs, instead of Q and P.

Proof. Since all factors of the Miller values that lie in a proper subfield of $\mathbb{F}_{q^k}$ vanish under the final exponentiation, it suffices to show that the Miller function updates at each iteration are equal, up to a constant defined over any proper subfield of $\mathbb{F}_{q^k}$. The computation of $a_T(Q, P)$ is composed of addition and doubling steps. Consider the gradients of the lines at either the addition or doubing stage of the Miller loop respectively. We have

$$\frac{y'_2 - y'_1}{x'_2 - x'_1} = \frac{\omega^3(y_2 - y_1)}{\omega^2(x_2 - x_1)} = \omega\frac{y_2 - y_1}{x_2 - x_1} \text{ and } \frac{3x_1'^2 + a\omega^4}{2y'_1} = \frac{\omega^4(3x_1^2 + a)}{2\omega^3 y_1} = \omega\frac{3x_1^2 + a}{2y_1}$$

for addition and doubling. We write the update to the Miller function at the doubling step, $f_{\mathtt{DBL}(U')}(P')$, as

$$\begin{aligned}(l_{\mathtt{DBL}(U')}(P'))/(v_{\mathtt{DBL}(U')}(P')) &= (y_{U'} - y_{P'} - \lambda'(x_{U'} - x_{P'}))/(x_{P'} - x_{[2]U'}) \\ = (\omega^3 y_U - \omega^3 y_P - \omega\lambda(\omega^2 x_U - \omega^2 x_P))/(\omega^2 x_P - \omega^2 x_{[2]U}) &= \omega \cdot f_{\mathtt{DBL}(U)}(P),\end{aligned}$$

where λ and λ' are the gradients determined before. We also have $f_{\mathtt{ADD}(U',Q')}(P') = \omega \cdot f_{\mathtt{ADD}(U,Q)}(P)$. For twists of degree $d = 2$ and $d = 4$, observe that $\omega^2 = \omega^{\gcd(d,6)}$ is in a subfield of $\mathbb{F}_{q^k}$ and thus vanishes in the final exponentiation. Similarly, for $d = 3$ and $d = 6$, ω^3 and ω^6 are both in subfields of $\mathbb{F}_{q^k}$ so that introducing a factor of 3 and 6 respectively to the exponent of $a_T(Q', P')$ will give an identical result to the computation of the same power of $a_T(Q, P)$. □

Corollary 2. *If $a_T(Q, P)$ is bilinear and non-degenerate, then so is $a_T(Q', P')$.*

Remark 3. Note that for $d = 6$ both ω^2 and ω^3 are in proper subfields of $\mathbb{F}_{q^k}$. Thus their contributions to the denominator and numerator vanish in the final exponentiation, so there is no need to introduce a factor of 6 to the final exponent. That is, for sextic twists it is actually always the case that $a_T(Q, P) = a_T(Q', P')$. If denominator elimination is used for $d = 6$, the values differ by ω^3 which lies in a subfield. For $k = 12$ and BN curves this case was considered by Akane, Nogami, and Morikawa [1] who showed that up to constants from subfields $a_T(Q, P) = a_T(Q', P')$.

For the other cases either ω^2 or ω^3 lie in a proper subfield $\mathbb{F}_{q^e}$ of $\mathbb{F}_{q^k}$. If 4 or 9 divides $\prod_{d|k} \Phi_d(q)/(q^e - 1)$, respectively, we obtain $\omega^{(q^k-1)/r} = 1$ and thus automatically $a_T(Q, P) = a_T(Q', P')$. However, in general these conditions are not satisfied, and the extra power of 2 or 3 is needed to obtain the same result.

Computing the ate pairing as $a_T(Q', P')$ and using twists as in Table 1 implies (for $d < 6$) that the only coordinate that lies in the full extension field $\mathbb{F}_{q^k}$ belongs to the second argument; for $d = 6$ all coordinates are defined over subfields. In this sense, the field operations encountered in computing the ate pairing $a_T(Q', P')$ on E' mimic the field operations encountered in computing the Tate pairing $e_r(P, Q)$ on E. Thus, point operation and line computation formulas that work in the Tate pairing can directly be applied to the ate pairing.

Inversions in $\mathbb{F}_{q^k}$ are prohibitively expensive and so we will show for all curve types a way to eliminate denominators. Therefore, at the doubling or addition stage of a Miller iteration the update function is given by a polynomial $f = \sum_{i,j} L_{i,j} \cdot x_S^i y_S^j$, where the $L_{i,j}$ are functions solely of the intermediate point U (doubling) or of the intermediate point U and the base point R (addition). In the Tate pairing computation of $e_r(P, Q)$, the $L_{i,j}$ are functions of some multiple of the point $P \in E(\mathbb{F}_q)$ and therefore all calculations required to compute the $L_{i,j}$ are performed in the base field $\mathbb{F}_q$. Similarly, in the modified definition of the ate pairing computation of $a_T(Q', P')$, the $L_{i,j}$ are functions of some multiple of the point $Q' \in E'(\mathbb{F}_{q^e})$ and therefore all calculations required to compute the $L_{i,j}$ in this case are performed in the subfield $\mathbb{F}_{q^e}$. Thus, if the computations of the $L_{i,j}$ in an iteration of the Tate pairing require $m\mathbf{m}_1 + s\mathbf{s}_1$, where $\mathbf{m}_1$ and $\mathbf{s}_1$ denote multiplication and squaring in $\mathbb{F}_q$, then the equivalent computations in an iteration of the ate pairing will require $m\mathbf{m}_e + s\mathbf{s}_e$, where $\mathbf{m}_e$ and $\mathbf{s}_e$ denote multiplication and squaring in $\mathbb{F}_{q^e}$; a multiplication by the curve constant a costs $\mathbf{d}_a$.

For even embedding degrees (admitting quadratic, sextic or quartic twists) the function update always simplifies to $f = L_{1,0}x + L_{0,1}y + L_{0,0}$, so that we have

two extra multiplications required here ($L_{1,0}$ by x and $L_{0,1}$ by y). In the Tate pairing as well as in the ate pairing each of these multiplications costs $e = k/d$ base field multiplications if field extensions are represented in a suitable way. If k is odd and divisible by three and if the curve admits a cubic twist, the function update requires more terms. For comparison, let there be h_{ADD} non-zero terms (excluding $L_{0,0}$) in the addition step and h_{DBL} in the doubling step, each of which costs $e = k/3$ base field multiplications. We summarize the situation for different twists in Table 2.

k even	DBL	ADD/ mADD
Tate: $e_r(P,Q)$	$m_1\mathbf{m}_1 + s_1\mathbf{s}_1 + 2e\mathbf{m}_1 + \mathbf{m}_k + \mathbf{s}_k$	$m_2\mathbf{m}_1 + s_2\mathbf{s}_1 + 2e\mathbf{m}_1 + \mathbf{m}_k$
Ate: $a_T(Q',P')$	$m_1\mathbf{m}_e + s_1\mathbf{s}_e + 2e\mathbf{m}_1 + \mathbf{m}_k + \mathbf{s}_k$	$m_2\mathbf{m}_e + s_2\mathbf{s}_e + 2e\mathbf{m}_1 + \mathbf{m}_k$
k odd, $3 \mid k$	DBL	ADD/ mADD
Tate: $e_r(P,Q)$	$m_1\mathbf{m}_1 + s_1\mathbf{s}_1 + h_{\mathrm{DBL}}e\mathbf{m}_1 + \mathbf{m}_k + \mathbf{s}_k$	$m_2\mathbf{m}_1 + s_2\mathbf{s}_1 + h_{\mathrm{ADD}}e\mathbf{m}_1 + \mathbf{m}_k$
Ate: $a_T(Q',P')$	$m_1\mathbf{m}_e + s_1\mathbf{s}_e + h_{\mathrm{DBL}}e\mathbf{m}_1 + \mathbf{m}_k + \mathbf{s}_k$	$m_2\mathbf{m}_e + s_2\mathbf{s}_e + h_{\mathrm{ADD}}e\mathbf{m}_1 + \mathbf{m}_k$

Table 2. Converting operation counts for single addition and doubling steps in the Tate pairing $e_r(P,Q)$ and ate pairing $a_T(Q',P')$.

In what follows, whenever we omit the subscripts from the operation costs and write $\mathbf{m}$ and $\mathbf{s}$, we mean $\mathbf{m}_1$, $\mathbf{s}_1$ for Tate pairing computation and $\mathbf{m}_e$, $\mathbf{s}_e$ for ate pairing computation.

Remark 4. Note that by Theorem 1 the computation of $a_T(Q',P')^{\gcd(d,6)}$ can be done entirely on the twisted curve. This means that Edwards curves can be employed in the ate setting if we choose the original curve such that the twisted curve can be written in Edwards form.

All curves we consider in the following are defined over the prime field $\mathbb{F}_p$. We therefore restrict to the case $q = p$ from now on.

4 Pairings on $y^2 = x^3 + ax$ with even embedding degrees

The only curves which admit quartic twists over $\mathbb{F}_p$ are of the form $E : y^2 = x^3 + ax$. In this section we assume that the embedding degree k is even and so by Table 1 we can use that the x-coordinates of Q (used in the Tate pairing) and of P' (used in our modified ate pairing) are defined over a subfield of $\mathbb{F}_{p^k}$. Using the naming convention introduced in Section 2, x_S is defined over a subfield of $\mathbb{F}_{p^k}$ while y_S is minimally defined over $\mathbb{F}_{p^k}$.

Curves of the form $E : y^2 = x^3 + ax$ have not received much attention, even for simple elliptic curve arithmetic, e.g. no special formulas were reported in the EFD [10] before our paper. We present new formulas for addition and doubling in a new coordinate system, which we call "weight-$(1,2)$ coordinates".

The point $(X : Y : Z)$ corresponds to the affine point (x, y), where $x = X/Z$ and $y = Y/Z^2$. The projective curve equation for these weights is $Y^2 = X^3Z + aXZ^3$. Lopez and Dahab studied such coordinates in the context of elliptic curves over binary fields but these weights have not been used in the context of curves over odd-characteristic fields.

It is quite remarkable that our doubling formulas are faster than any doubling formulas reported for elliptic curves in the EFD.

We extend the explicit formulas for curve operations to compute the doubling and the addition step on these curves. The resulting pairing computations are also significantly faster than their predecessors.

Doubling formulas. For this curve shape the affine doubling formulas to compute $(x_3, y_3) = [2]U = [2](x_1, y_1)$ simplify to $x_3 = \lambda^2 - 2x_1,\ y_3 = \lambda(x_1 - x_3) - y_1$, where $\lambda = (3x_1^2 + a)/(2y_1)$. In weight-$(1, 2)$ coordinates the doubling formulas to compute $(X_3 : Y_3 : Z_3) = [2](X_1 : Y_1 : Z_1)$ become

$$X_3 = (X_1^2 - aZ_1^2)^2,\ Y_3 = 2Y_1(X_1^2 - aZ_1^2)((X_1^2 + aZ_1^2)^2 + 4aZ_1^2X_1^2),\ Z_3 = 4Y_1^2.$$

The point doubling needs $1\mathbf{m} + 6\mathbf{s} + 1\mathbf{d}_a$ using the following sequence of operations.

$$\begin{aligned} &A = X_1^2,\ B = Y_1^2,\ C = Z_1^2,\ D = aC,\ X_3 = (A - D)^2, \\ &E = 2(A + D)^2 - X_3,\ F = ((A - D + Y_1)^2 - B - X_3),\ Y_3 = E \cdot F,\ Z_3 = 4B. \end{aligned} \tag{1}$$

These formulas are now the fastest doubling formulas reported in the EFD [10]. They are faster by 1 **s-m** tradeoff. than the previous champion, "dbl-20090311-hwcd" due to Hisil, Wong, Carter, and Dawson. Those formulas are optimized for "Doubling-oriented XXYZZR coordinates for Jacobi quartics" and need $2\mathbf{m} + 5\mathbf{s} + 1\mathbf{d}_a$, where a is some curve constant.

Line computation for doubling. In the doubling step of the pairing computation we need to compute $[2]U$ and to compute the line function at U and evaluate it at $S = (x_S, y_S)$. The affine formula for the computation of $f_{\mathrm{DBL}(U)}(S)$ is given as $\frac{\lambda(X_1/Z_1 - x_S) + y_S - Y_1/Z_1^2}{x_S - (\lambda^2 - 2X_1/Z_1)} = -\frac{2Y_1(-(3X_1^2Z_1 + aZ_1^3)\cdot x_S + (2Y_1Z_1)\cdot y_S + X_1^3 - aZ_1^2X_1)}{-(4Y_1^2Z_1)\cdot x_S + 9X_1^4Z_1 + 6aX_1^2Z_1^3 + a^2Z_1^5 - 8X_1Y_1^2}$. Since any element except for y_S is in a proper subfield of $\mathbb{F}_{p^k}$, we can omit computing the entire denominator and also the multiplication by $-Y_1$. We leave the factor of 2 to obtain an **s-m** tradeoff. The simplified line function is

$$f'_{\mathrm{DBL}(U)}(S) = -2(3X_1^2Z_1 + aZ_1^3) \cdot x_S + (4Y_1Z_1) \cdot y_S + 2(X_1^3 - aZ_1^2X_1).$$

We write $f'_{\mathrm{DBL}(U)}(S)$ as $f'_{\mathrm{DBL}(U)}(S) = L_{1,0} \cdot x_S + L_{0,1} \cdot y_S + L_{0,0}$ and compute $L_{1,0}$, $L_{0,1}$ and $L_{0,0}$ as

$$L_{1,0} = -2Z_1 \cdot (3 \cdot A + D),\ L_{0,1} = 2((Y_1 + Z_1)^2 - B - C),\ L_{0,0} = (X_1 + A - D)^2 - X_3 - A,$$

using the values computed in (1) at an additional cost of $1\mathbf{m} + 2\mathbf{s}$, so that the total operation count for point doubling with line computation is $2(k/d)\mathbf{m}_1 + 2\mathbf{m} + 8\mathbf{s} + 1\mathbf{d}_a$.

Addition and mixed addition. In affine coordinates, the sum $(x_3, y_3) = U + R = (x_1, y_1) + (x_2, y_2)$ is given by $x_3 = \lambda^2 - x_1 - x_2,\ y_3 = \lambda(x_1 - x_3) - y_1$, where $\lambda = (y_1 - y_2)/(x_1 - x_2)$. In weight-$(1, 2)$ coordinates this becomes $(X_3 : Y_3 : Z_3) = (X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2)$

$$\begin{aligned} X_3 &= (Y_1Z_2^2 - Y_2Z_1^2)^2 - (X_1Z_2 + X_2Z_1)T, \\ Y_3 &= ((Y_1Z_2^2 - Y_2Z_1^2)(X_1Z_2T - X_3) - Y_1Z_2^2TU)UZ_1Z_2, \\ Z_3 &= (UZ_1Z_2)^2, \end{aligned}$$

where $T = (X_1Z_2 - X_2Z_1)^2Z_1Z_2$ and $U = (X_1Z_2 - X_2Z_1)$. This addition can be computed in $10\mathbf{m} + 7\mathbf{s}$ using

$$\begin{aligned} &A = Z_1^2,\ B = Z_2^2,\ C = (Z_1 + Z_2)^2 - A - B,\ D = X_1 \cdot Z_2,\ E = X_2 \cdot Z_1, \\ &F = Y_1 \cdot B,\ G = Y_2 \cdot A,\ H = (D - E),\ I = 2(F - G),\ II = I^2,\ J = C \cdot H, \\ &K = 4J \cdot H,\ X_3 = 2II - (D + E) \cdot K,\ Z_3 = J^2, \\ &Y_3 = ((J + I)^2 - Z_3 - II) \cdot (D \cdot K - X_3) - F \cdot K^2,\ Z_3 = 2Z_3. \end{aligned}$$

For mixed addition, i.e. $Z_2 = 1$, the number of operations reduces to $8\mathbf{m} + 5\mathbf{s}$ omitting computation of B, C, D and F.

Line computation for addition and mixed addition. For affine points U, R, and S the line function is given by $f_{\mathrm{ADD}(U,R)}(S) = \frac{\lambda(x_2 - x_S) + y_S - y_2}{x_S - (\lambda^2 - x_1 - x_2)}$. Again, we can omit the denominator because it is entirely defined over a subfield of $\mathbb{F}_{p^k}$. In weight-$(1, 2)$ coordinates the modified line function becomes $f'_{\mathrm{ADD}(U,R)}(S) = I \cdot X_2Z_2 - I \cdot x_SZ_2^2 + J \cdot y_SZ_2^2 - J \cdot Y_2$. The values $X_2Z_2, x_SZ_2^2$, and $y_SZ_2^2$ do not change during the computation and can thus be precomputed. For the Tate pairing the cost of one addition step (computation of addition and line function) therefore is $(2k/d)\mathbf{m}_1 + 12\mathbf{m} + 7\mathbf{s}$. If $d = 2$ it is possible to save $1\mathbf{m}$ by computing $I \cdot (X_2Z_2 - x_SZ_2^2)$. When computing the ate pairing, the multiplications in $I \cdot X_2Z_2 - I \cdot x_SZ_2^2 + J \cdot y_SZ_2^2 - J \cdot Y_2$ cost $1\mathbf{m}$ each, given the shape of x_S and y_S. The cost of one addition step (computation of addition and line function) in the ate pairing therefore is $14\mathbf{m} + 7\mathbf{s}$.

For mixed additions ($Z_2 = 1$) this simplifies to $f'_{\mathrm{mADD}(U,R)}(S) = I \cdot X_2 - I \cdot x_S + J \cdot y_S - J \cdot Y_2$, costing $(2k/d)\mathbf{m}_1 + 10\mathbf{m} + 5\mathbf{s}$ for both the ate and the Tate pairing for a complete mixed addition step. For $d = 2$ again $1\mathbf{m}$ can be saved in the Tate pairing.

If R is reused several times in the Tate pairing it might be worthwhile to precompute $1/Y_2$ for longterm usage. At the beginning of a pairing computation $\tilde{X}_2 = X_2/Y_2, \tilde{x}_S = x_S/Y_2$ and $\tilde{y}_S = y_S/Y_2$ are computed. Since Y_2 lies in $\mathbb{F}_p$, so $f'_{\mathrm{mADD}(U,R)}(S)$ can be replaced by

$$f'_{\mathrm{mADD}(U,R)}(S)/Y_2 = I \cdot \tilde{X}_2 - I \cdot \tilde{x}_S + J \cdot \tilde{y}_S - J$$

without changing the pairing value. Note also that Table 1 shows that $\tilde{x}_S$ and $\tilde{y}_S$ are defined over the same fields as x_S and y_S are. In this case a mixed addition step costs only $(2k/d)\mathbf{m}_1 + 9\mathbf{m} + 5\mathbf{s}$.

If instead S is reused several times in the ate pairing, similar savings are possible. It is useful to precompute $1/y'_S$ and update the function by

$$\bar{f}_{\mathrm{mADD}(U,R)}(S)/\bar{y}_S = I \cdot \bar{X}_2 - I \cdot \bar{x}_S + J\omega^3 - J \cdot \bar{Y}_2,$$

where $\bar{X}_2 = X_2/\bar{y}_S, \bar{x}_S = x_S/\bar{y}_S$ and $\bar{Y}_2 = Y_2/\bar{y}_S$, and $y_S = \bar{y}_S\omega^3$ with $\bar{y}_S \in \mathbb{F}_p$. In this case a mixed addition step costs only $(2k/d)\mathbf{m}_1 + 9\mathbf{m} + 5\mathbf{s}$.

Note that these savings are compatible with the saving for $d = 2$.

Depending on the representation of $\mathbb{F}_{p^k}$ over $\mathbb{F}_{p^{k/2}}$ and $\mathbb{F}_{p^{k/d}}$ it is possible to save operations in the other cases.

5 Pairings on $y^2 = x^3 + b$ with even embedding degrees

The only curves which can have sextic twists over $\mathbb{F}_p$ are of the form $E : y^2 = x^3 + b$. In this section we assume that the embedding degree k is even and so by Table 1 we can use that the x-coordinate of Q (in $e_r(P, Q)$) and of P' (in $a_T(Q', P')$) is defined over a subfield of $\mathbb{F}_{p^k}$. Using the naming convention introduced in Section 2, x_S is defined over a subfield of $\mathbb{F}_{p^k}$ while y_S might be defined over $\mathbb{F}_{p^k}$. Note that if $d = 6$, y_S is also defined over a proper subfield, namely $\mathbb{F}_{p^{k/3}}$. For these curves we obtained the best results in standard projective coordinates where the curve equation $y^2 = x^3 + b$ becomes $Y^2Z = X^3 + bZ^3$.

When b is a square in $\mathbb{F}_p$, the curve E always has a point of order 3, otherwise such a point never exists in $E(\mathbb{F}_p)$. The former case was extensively studied in [13] in the context of the Tate pairing. The addition formulas are independent of the nature of the curve constant b and can therefore also be used for non-square b. We slightly improve these addition formulas in the second half of this section and use these formulas for all curves with $a = 0$. The first part of this section focuses on achieving faster operation counts at the Miller doubling stage on general curves of the form $E : y^2 = x^3 + b$, where we make no assumptions about the nature of the curve constant b (and consequently the order of E).

Point doubling and line computation. The affine doubling formulas differ from those in Section 4 in the definition of λ. We have $\lambda = 3x_1^2/2y_1$. In projective coordinates and after eliminating powers of X_1^3 via the curve equation, we obtain $(X_3 : Y_3 : Z_3) = [2](X_1 : Y_1 : Z_1)$ as

$$X_3 = 2X_1Y_1(Y_1^2 - 9bZ_1^2),\ Y_3 = Y_1^4 + 18bY_1^2Z_1^2 - 27b^2Z_1^4,\ Z_3 = 8Y_1^3Z_1.$$

We homogenize the affine doubling line using $x_1 = X_1/Z_1$ and $y_1 = Y_1/Z_1$ and get

$$f'_{\mathrm{DBL}(U)}(S) = 3X_1^2 \cdot x_S - 2Y_1Z_1 \cdot y_S + 3bZ_1^2 - Y_1^2.$$

We write $f'_{\mathrm{DBL}(U)}(S) = L_{1,0} \cdot x_S + L_{0,1} \cdot y_S + L_{0,0}$ and compute $L_{1,0}$, $L_{0,1}$, $L_{0,0}$ and the point $(X_3 : Y_3 : Z_3)$ using the following sequence of operations.

$$\begin{aligned}
&A = X_1^2,\ B = Y_1^2,\ C = Z_1^2,\ D = 3bC,\ E = (X_1 + Y_1)^2 - A - B,\\
&F = (Y_1 + Z_1)^2 - B - C,\ G = 3D,\ X_3 = E \cdot (B - G),\\
&Y_3 = (B + G)^2 - 12D^2,\ Z_3 = 4B \cdot F,\ L_{1,0} = 3A,\ L_{0,1} = -F,\ L_{0,0} = D - B.
\end{aligned}$$

The total count for the above sequence of operations is $2\mathbf{m}+7\mathbf{s}+1\mathbf{d}_b$ in addition to the multiplications by x_S and y_S. Note that doubling outside the context of pairings would omit the computation of A and would obtain $E = 2X_1Y_1$, needing a total of $3\mathbf{m} + 5\mathbf{s} + 1\mathbf{d}_b$. As doubling formulas they are not competitive with those in the EFD but they are almost the fastest for the doubling step in pairings, second only to $y^2 = x^3 + c^2$ in [13].

Addition, mixed addition and line computation. For the addition of points on $y^2 = x^3 + b$, we adopt the formulas obtained in [13] for curves of the form $y^2 = x^3 + c^2$. These addition and line computation formulas are independent of b being a square. The cost for an addition is $12\mathbf{m} + 2\mathbf{s}$ The addition line in [13] can be written as $f'_{\mathrm{ADD}(U,R)}(S) = (Y_1Z_2 - Y_2Z_1) \cdot X_2 - (Y_1Z_2 - Y_2Z_1) \cdot x_S Z_2 + (X_1Z_2 - X_2Z_1) \cdot y_S Z_2 - (X_1Z_2 - X_2Z_1) \cdot Y_2$. Note that the coefficients appear as subexpressions in the mixed addition of U and R, so computing $f'_{\mathrm{ADD}(U,R)}(S)$ as above costs an extra $(2k/d)\mathbf{m}_1 + 2\mathbf{m}$ for the Tate pairing and an extra $4\mathbf{m}$ the ate pairing.

If $R = (X_2 : Y_2 : 1)$, the addition $U + R$ becomes a mixed addition and costs $9\mathbf{m} + 2\mathbf{s}$. Computing the addition and the line as $f'_{\mathrm{mADD}(U,R)}(S) = (Y_1 - Y_2Z_1) \cdot X_2 - (Y_1 - Y_2Z_1) \cdot x_S + (X_1 - X_2Z_1) \cdot y_S - (X_1 - X_2Z_1) \cdot Y_2$ costs an extra $(2k/d)\mathbf{m}_1 + 2\mathbf{m}$ for both the Tate and the ate pairing.

If R or S is fixed in the mixed addition, similar comments to Section 4 apply, reducing the extra costs to only $(2k/d)\mathbf{m}_1 + \mathbf{m}$.

6 Fast Formulas for Pairing Computations with Cubic Twists

For an odd embedding degree k, the only possible non-trivial twists are cubic twists and these only exist for curves of the form $y^2 = x^3 + b$, requiring also that $3|k$. Table 1 shows that in this scenario the point $S = (x_S, y_S)$ has x_S defined over the full extension field $\mathbb{F}_{p^k}$ and y_S defined over a subfield. The formulas obtained in most publications including the previous sections use denominator elimination based on x_S being in a subfield.

In this section we present fast formulas for addition and doubling steps for $y^2 = x^3 + b$ and optimize them using the fact that y_S, y_U and x_U are in a proper subfield of $\mathbb{F}_{p^k}$, while x_S is not. Our results are significantly faster than other studies of this case, but nevertheless the cases with even embedding degree offer more advantages. For curves of the form $y^2 = x^3 + b$, Lin *et al.* [31] observed that $1/v_{\mathrm{DBL}(U)}(S)$ can be written as

$$\frac{1}{v_{\mathrm{DBL}(U)}(S)} = \frac{1}{x_S - x_{[2]U}} = \frac{x_S^2 + x_S x_{[2]U} + x_{[2]U}^2}{(y_S - y_{[2]U})(y_S + y_{[2]U})}.$$

Since $(y_S - y_{[2]U})(y_S + y_{[2]U})$ lies in a subfield, the line function can be multiplied by $x_S^2 + x_S x_{[2]U} + x_{[2]U}^2$, instead of dividing it by $v_{\mathrm{DBL}(U)}(S)$. Analogously, the addition step becomes $f'_{\mathrm{ADD}(U,R)}(S) = l_{\mathrm{ADD}(U,R)}(S) \cdot (x_S^2 + x_S x_{U+R} + x_{U+R}^2)$.

Point doubling and line computation. In projective coordinates $x_U = X_1/Z_1$ and $y_U = Y_1/Z_1$, we replace $X_1^3 = Y_1^2 Z_1 - bZ_1^3$ and factor $f'_{\mathrm{DBL}(U)}(S)$ to see that $f'_{\mathrm{DBL}(U)}(S)$ equals

$$\alpha \cdot \left(X_1 Z_1 (Y_1^2 - 9bZ_1^2) \cdot x_S + (4Y_1^2 Z_1^2) \cdot x_S^2 - (6X_1^2 Y_1 Z_1) \cdot y_S + X_1^2 (Y_1^2 + 9bZ_1^2)\right),$$

where $\alpha = (18bY_1^2 Z_1^2 - 27b^2 Z_1^4 + Y_1^4 + 8Y_1^3 Z_1 \cdot y_S)/(32Y_1^5 Z_1^3) \in \mathbb{F}_{p^{k/3}}$ does not contain x_S and can be discarded. The values for X_1 and Z_1 are defined over subfields of $\mathbb{F}_{p^k}$ and we obtain more efficient formulas by computing $f''_{\mathrm{DBL}(U)}(S) = f'_{\mathrm{DBL}(U)}(S) X_1/(Z_1 \alpha)$ as

$$f''_{\mathrm{DBL}(U)}(S) = X_1^2(Y_1^2 - 9bZ_1^2) \cdot x_S + 4X_1 Y_1^2 Z_1 \cdot x_S^2 - 6X_1^3 Y_1 \cdot y_S + (Y_1^2 - bZ_1^2)(Y_1^2 + 9bZ_1^2).$$

For cubic twists, the term $x_S^2 \in \mathbb{F}_{p^k}$ appears in the simplified doubling line function so we write $f''_{\mathrm{DBL}(U)}(S) = L_{1,0} \cdot x_S + L_{2,0} \cdot x_S^2 + L_{0,1} \cdot y_S + L_{0,0}$. We compute $(X_3 : Y_3 : Z_3) = [2](X_1 : Y_1 : Z_1)$ and the necessary $L_{i,j}$ coefficients using $6\mathbf{m} + 7\mathbf{s} + 1\mathbf{d}_b$ in addition to the multiplications by x_S, x_S^2, and y_S.

$$\begin{aligned}
&A = X_1^2,\ B = Y_1^2,\ C = Z_1^2,\ D = bC,\ E = 3D,\ F = (X_1 + Y_1)^2 - A - B,\\
&G = (Y_1 + Z_1)^2 - B - C,\ H = 3E,\ X_3 = F \cdot (B - H),\\
&Y_3 = (B + H)^2 - 3(2E)^2,\ Z_3 = 4B \cdot G,\ L_{1,0} = A \cdot (B - H),\ L_{2,0} = F \cdot G,\\
&L_{0,1} = -3A \cdot F,\ L_{0,0} = (B - D) \cdot (B + H).
\end{aligned}$$

Note that the formulas in [33] require $8\mathbf{m}+9\mathbf{s}+1\mathbf{d}_b$ in addition to the multiplications by x_S, x_S^2, y_S, y_S^2, $x_S y_S$, and $x_S^2 y_S$, i.e. they need 6 multiplications costing $k/3$ base field multiplications each while we only need 3 such multiplications. This means that the overall saving is $2\mathbf{m} + 2\mathbf{s} + k\mathbf{m}_1$.

Addition and line computation. For additions we break with the conventional wisdom that the line function should be given in terms of the base point. For even embedding degrees where denominator elimination does not require further adjustment, that approach is suitable and particularly helps if the base point is given in affine coordinates. For the curves in this section we show that building the line function on the resulting point $(X_3 : Y_3 : Z_3)$ gives better operation counts in spite of Z_3 not being equal to 1.

The default line function is given by $\left(\frac{(y_1 - y_2)}{(x_1 - x_2)} \cdot (x_1 - x_S) + y_S - y_1\right)/(x_3 - x_S)$. Using the above denominator elimination technique this gets transformed to

$$\left(\frac{(y_2 - y_1)}{(x_2 - x_1)} \cdot (x_1 - x_S) + y_S - y_1\right) \cdot \left(x_3^2 + x_3 x_S + x_S^2\right)/(y_3^2 - y_S^2).$$

This approach leads to a polynomial of the form $L_{2,0} \cdot x_S^2 + L_{1,0} \cdot x_S + L_{1,1} \cdot x_S y_S + L_{2,1} \cdot x_S^2 y_S + L_{0,2} \cdot y_S + L_{0,1} \cdot y_S + L_{0,0}$ which requires $(6k/3)\mathbf{m}_1$ after the computation of the coefficients $L_{i,j}$.

In the representation

$$\left(\frac{(y_1 - y_2)}{(x_1 - x_2)} \cdot (x_3 - x_S) + y_S + y_3\right) \cdot \left(x_3^2 + x_3 x_S + x_S^2\right)/(y_3^2 - y_S^2)$$

using the coordinates x_3, y_3 instead of x_1, y_1, it becomes obvious that the factor $(x_3 - x_S)(x_3^2 + x_3x_S + x_S^2) = y_3^2 - y_S^2$ appears in the left term of the numerator and that thus the whole numerator is divisible by the subfield element $y_S + y_3$. (Note the sign change on y_3 because the line goes through $(x_3, -y_3)$ by the geometric addition law on E.) This means that the line function is of the form $L_{2,0} \cdot x_S^2 + L_{1,0} \cdot x_S + L_{0,1} \cdot y_S + L_{0,0}$, requiring only $(3k/3)\mathbf{m}_1$ after the computation of the coefficients $L_{i,j}$.

We obtain in projective coordinates that $f'_{\mathrm{ADD}(U,R)}(S)$ equals

$$\frac{(Y_1Z_2 - Y_2Z_1)Z_3(Y_3 - y_SZ_3) + (X_3^2 + X_3Z_3x_S + Z_3^2x_S^2)(X_1Z_2 - X_2Z_1)}{(Y_3 - y_SZ_3)(X_1Z_2 - X_2Z_1)Z_3}.$$

The denominator can be discarded. To compute the numerator more efficiently we observe that $Z_3 = Z_1Z_2(X_1Z_2 - X_2Z_1)^3$ so that we can divide by $(X_1Z_2 - X_2Z_1)$; furthermore we scale the function by 2 to allow an **s**-**m** tradeoff. This gives

$$\begin{aligned} f''_{\mathrm{ADD}(U,R)}(S) = {} & 2Z_3^2x_S^2 + 2X_3Z_3x_S - 2Z_1Z_2(X_1Z_2 - X_2Z_1)^2(Y_1Z_2 - Y_2Z_1)Z_3y_S \\ & + 2X_3^2 + 2Z_1Z_2(X_1Z_2 - X_2Z_1)^2(Y_1Z_2 - Y_2Z_1)Y_3. \end{aligned}$$

We compute the addition and line computation using the following sequence of operations.

$$\begin{aligned} & A = X_1 \cdot Z_2,\ B = Y_1 \cdot Z_2,\ C = Z_1 \cdot Z_2,\ D = Z_1 \cdot X_2 - A,\ E = B - Z_1 \cdot Y_2, \\ & F = D^2,\ G = E^2,\ H = -D \cdot F,\ I = F \cdot A,\ J = H + C \cdot G - 2I, \\ & K = C \cdot F \cdot E;\ X_3 = -D \cdot J,\ Y_3 = E \cdot (I - J) - (H \cdot B),\ Z_3 = C \cdot H, \\ & L = X_3^2,\ M = Z_3^2,\ N = (X_3 + Z_3)^2 - L - M,\ L_{2,0} = 2M,\ L_{1,0} = N, \\ & L_{0,0} = 2(L + K \cdot Y_3),\ L_{0,1} = -2K \cdot Z_3. \end{aligned}$$

The explicit formulas for computing $(X_3 : Y_3 : Z_3)$ are the same as in the EFD [10]; they use $12\mathbf{m} + 2\mathbf{s}$ and use the intermediate variables $A, \ldots, J$; the values $K, \ldots, N$ are used in the computation of the line function. The total operation count for the above sequence of operations is $16\mathbf{m} + 5\mathbf{s}$ in addition to the multiplications by x_S^2, x_S, and y_S. Mixed addition is cheaper saving one $\mathbf{m}$ in each of A, B, and C and needing only $13\mathbf{m} + 5\mathbf{s}$.

In the pairing computation each addition is followed by a doubling. Thus $L = X_3^2$ and $M = Z_3^2$ should be cached and used in the doubling computation. This reuse reduces the effective costs of the addition step by $2\mathbf{s}$ and similarly for the mixed-addition step. Accordingly we report $16\mathbf{m} + 3\mathbf{s}$ and $13\mathbf{m} + 3\mathbf{s}$ in the comparison in Section 7.

7 Comparisons

This section compares the speed of our pairing formulas with the literature in the following categories:

- **(i)**: Curves of the form $y^2 = x^3 + ax$ have twists of degree $d = 2$ and 4. We compare operation counts with the results given by Ionica and Joux [27] and Arène *et al.* [2]; note that those papers cover general Weierstrass curves but we are not aware of any other study covering this case.
- **(ii)**: Curves of the form $y^2 = x^3 + c^2$ have a point of order 3 and admit twists of degrees $d = 2$ and 6. These curves were studied in detail very recently by Costello *et al.* in [13] and we only found faster mixed addition formulas than those originally proposed.
- **(iii)**: Curves of the form $y^2 = x^3 + b$ do not necessarily have a point of order 3. We study operation counts for twists of degree 2 and 6. These curves cover in particular BN curves [8]. We compare our new formulas with those given for the same curve shape in [2].
- **(iv)** Curves of the form $y^2 = x^3 + b$ also have twists of degree 3. This case requires very different optimizations and has not been studied much in the literature. The first paper studying pairing computation on curves admitting cubic twists [31] did not pay close attention to the operation count itself, so we compare our formulas with the results presented by El Mrabet *et al.* in [33], although that paper did not present addition formulas.

The above papers for even d give $k\mathbf{m}_1$ for evaluating the line function. This can almost always be done in $(2k/d)\mathbf{m}_1$, so we adjust their results accordingly. In the general addition case, Table 3 only gives counts for the Tate pairing. For $d = 2$ it is possible to save $1\mathbf{m}$ in each ADD and each mADD. For the ate pairing in this case the costs are different and the operation counts should be modified by $-(2k/d)\mathbf{m}_1 + 2\mathbf{m}$.

For mixed additions we use our improved precomputations, assuming that one of the input points is fixed.

Curve Curve order Twist deg.	Best Coord.	DBL ADD mADD	Prev. best Coord.	DBL ADD mADD
$y^2 = x^3 + ax$	Sec. 4	$(2k/d)\mathbf{m}_1 + 2\mathbf{m} + 8\mathbf{s} + 1\mathbf{d}_a$	[27],	$(2k/d)\mathbf{m}_1 + 1\mathbf{m} + 11\mathbf{s} + 1\mathbf{d}_a$
-	$\mathcal{W}_{(1,2)}$	$(2k/d)\mathbf{m}_1 + 12\mathbf{m} + 7\mathbf{s}$	[2]	$(2k/d)\mathbf{m}_1 + 10\mathbf{m} + 6\mathbf{s}$
$d = 2, 4$		$(2k/d)\mathbf{m}_1 + 9\mathbf{m} + 5\mathbf{s}$	$\mathcal{J}$	$(2k/d)\mathbf{m}_1 + 7\mathbf{m} + 6\mathbf{s}$
$y^2 = x^3 + c^2$	Sec. 5	$(2k/d)\mathbf{m}_1 + 3\mathbf{m} + 5\mathbf{s}$	[13]	$(2k/d)\mathbf{m}_1 + 3\mathbf{m} + 5\mathbf{s}$
$3 \mid \#E$	& [13]	$(2k/d)\mathbf{m}_1 + 14\mathbf{m} + 2\mathbf{s} + 1\mathbf{d}_c$	$\mathcal{P}$	$(2k/d)\mathbf{m}_1 + 14\mathbf{m} + 2\mathbf{s} + 1\mathbf{d}_c$
$d = 2, 6$	$\mathcal{P}$	$(2k/d)\mathbf{m}_1 + 10\mathbf{m} + 2\mathbf{s} + 1\mathbf{d}_c$		$(2k/d)\mathbf{m}_1 + 11\mathbf{m} + 2\mathbf{s} + 1\mathbf{d}_c$
$y^2 = x^3 + b$	Sec. 5	$(2k/d)\mathbf{m}_1 + 2\mathbf{m} + 7\mathbf{s} + 1\mathbf{d}_b$	[2]	$(2k/d)\mathbf{m}_1 + 3\mathbf{m} + 8\mathbf{s}$
$3 \nmid \#E$	& [13]	$(2k/d)\mathbf{m}_1 + 14\mathbf{m} + 2\mathbf{s}$	$\mathcal{J}$	$(2k/d)\mathbf{m}_1 + 10\mathbf{m} + 6\mathbf{s}$
$d = 2, 6$	$\mathcal{P}$	$(2k/d)\mathbf{m}_1 + 10\mathbf{m} + 2\mathbf{s}$		$(2k/d)\mathbf{m}_1 + 7\mathbf{m} + 6\mathbf{s}$
$y^2 = x^3 + b$	Sec. 6	$k\mathbf{m}_1 + 6\mathbf{m} + 7\mathbf{s} + 1\mathbf{d}_b$	[33]	$2k\mathbf{m}_1 + 8\mathbf{m} + 9\mathbf{s} + 1\mathbf{d}_b$
-	$\mathcal{P}$	$k\mathbf{m}_1 + 16\mathbf{m} + 3\mathbf{s}$	$\mathcal{P}$	*ADD/mADD*
$d = 3$		$k\mathbf{m}_1 + 13\mathbf{m} + 3\mathbf{s}$		*not reported*

Table 3. Comparisons of our pairing formulas with the previous fastest formulas.

We point out that all new doublings are faster than the previous ones. In (i) and (iii) this comes at the expense of somewhat slower additions. In the Miller loop, doublings are significantly more frequent than additions so that this disadvantage is amply mitigated by the faster doublings. Note that the doublings save entire field operations, and do not just present **s**–**m** tradeoffs.

In Table 4 we determine the operation counts for both the Tate and ate pairings in a typical iteration of Miller's algorithm, based on the fastest operation counts summarized in Table 3. In optimized pairing implementations, the loop parameter is chosen to have a low Hamming weight so that only few additions are encountered throughout the loop. Thus, the operation counts presented in Table 4 are for the doubling stage of Miller's algorithm. The column titled *Tate* gives the equivalent number of total base field operations (multiplications and squarings in $\mathbb{F}_p$) for a Miller iteration, based on the fact that the first argument is $R \in E(\mathbb{F}_p)$ and the second argument is $S \in E(\mathbb{F}_{p^k})$; for the fields of the individual coordinates see Table 1. The column titled *ate* gives the equivalent number of base field operations for an iteration where the first argument is $R \in E'(\mathbb{F}_{p^e})$ and the second argument is $S \in E'(\mathbb{F}_{p^k})$. If $s = 2^i 3^j$, then we can quantify the cost of a multiplication in the field $\mathbb{F}_{p^s}$ as $3^i 5^j$ multiplications in $\mathbb{F}_p$ using Karatsuba and/or Toom-Cook multiplication, and we do the same for squarings, cf. [29] for details. To compare across operations we follow the EFD [10] and report two sets of numbers: the first ones are assuming that $1\mathbf{s} = 1\mathbf{m}$ and the second ones are assuming that $1\mathbf{s} = 0.8\mathbf{m}$. In the second case, we assume that squarings in $\mathbb{F}_{p^k}$ do not make use of special properties of the field extension. Thus we approximate the ratio of squaring to multiplication costs to be 0.8 as well. In both cases we assume multiplications by curve constants to be virtually free.

We use the optimal methods of curve construction for each embedding degree, which were originally presented in [18], to determine which categories (**(i)-(iv)**) E and E' belong to. We note that constructions 6.11-6.14 in [18] are due to [28]. The construction of BN curves for $k = 12$ was given in [8] and construction 6.10 for $k = 8$ curves is due to [41]. For each embedding degree, we also present the loop length ratios $m_{opt} : T_e : r$, where m_{opt} is the loop parameter of the optimal ate pairing, T_e is the loop parameter of the twisted ate pairing and r is the loop parameter of the standard Tate pairing. For all construction methods shown in Table 4 there is an optimal ate pairing achieving the minimal loop length in Miller's algorithm. For the twisted ate pairing we used the shortest loop length found by considering the powers of $(t-1)^e \mod r$. In the last column, we compare the optimal ate pairing and twisted ate pairing and present a factor that approximates how many times faster the computation of the Miller loop is under the faster pairing option.

References

1. Masataka Akane, Yasuyuki Nogami, and Yoshitaka Morikawa. Fast ate pairing computation of embedding degree 12 using subfield-twisted elliptic curve. *IEICE Transactions*, 92-A(2):508–516, 2009.

k	Const. [18]	$\varphi(k)$	ρ	d	E	E'	$m_{opt} : T_e : r$ (log)	Tate : ate $\mathbf{s} = \mathbf{m}$	Tate : ate $\mathbf{s} = 0.8\mathbf{m}$	$a_{m_{opt}}$ vs. η_{T_e}
4	6.4	2	2.000	4	**(i)**	**(i)**	1 : 1 : 2	30 : 30	26.6 : 26.6	Even
6	6.6	2	2.000	6	**(ii)**	**(iii)**	1 : 1 : 2	40 : 41	36 : 36.6	η_{T_e} (1.02)
8	6.10	4	1.500	4	**(i)**	**(i)**	3 : 3 : 4	68 : 88	61 : 77.8	η_{T_e} (1.3)
9	6.6	6	1.333	3	**(iv)**	**(iv)**	1 : 3 : 6	72 : 124	65.6 : 112	$a_{m_{opt}}$ (1.7)
12	6.8	4	1.000	6	**(iii)**	**(iii)**	1 : 2 : 4	103 : 121	92.6 : 107.8	$a_{m_{opt}}$ (1.7)
16	6.11	8	1.250	4	**(i)**	**(i)**	1 : 4 : 8	180 : 260	162.2 : 229.4	$a_{m_{opt}}$ (2.8)
18	6.12	6	1.333	6	**(iii)**	**(ii)**	1 : 3 : 6	165 : 196	148.6 : 176	$a_{m_{opt}}$ (2.5)
24	6.6	8	1.250	6	**(ii)**	**(iii)**	1 : 4 : 8	286 : 359	258 : 319.4	$a_{m_{opt}}$ (3.2)
27	6.6	18	1.111	3	**(iv)**	**(iv)**	1 : 9 : 18	290 : 602	263.6 : 542	$a_{m_{opt}}$ (4.4)
32	6.13	16	1.125	4	**(i)**	**(i)**	1 : 8 : 16	512 : 772	461.8 : 680.2	$a_{m_{opt}}$ (5.3)
36	6.14	12	1.167	6	**(iii)**	**(iii)**	1 : 6 : 12	471 : 597	424.6 : 531	$a_{m_{opt}}$ (4.7)
48	6.6	16	1.125	6	**(ii)**	**(iii)**	1 : 8 : 16	834 : 1069	752 : 950.2	$a_{m_{opt}}$ (6.2)

Table 4. Comparison of optimal ate pairing and twisted ate pairing.

2. Christophe Arene, Tanja Lange, Michael Naehrig, and Christophe Ritzenthaler. Faster pairing computation. Cryptology ePrint Archive, Report 2009/155, 2009. http://eprint.iacr.org/.
3. Paulo S. L. M. Barreto, Steven D. Galbraith, Colm O'hEigeartaigh, and Michael Scott. Efficient pairing computation on supersingular abelian varieties. *Des. Codes Cryptography*, 42(3):239–271, 2007.
4. Paulo S. L. M. Barreto, Hae Yong Kim, Ben Lynn, and Michael Scott. Efficient algorithms for pairing-based cryptosystems. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 354–368. Springer, 2002.
5. Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. Constructing elliptic curves with prescribed embedding degrees. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *SCN*, volume 2576 of *Lecture Notes in Computer Science*, pages 257–267. Springer, 2002.
6. Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. On the selection of pairing-friendly groups. In Mitsuru Matsui and Robert J. Zuccherato, editors, *Selected Areas in Cryptography*, volume 3006 of *Lecture Notes in Computer Science*, pages 17–25. Springer, 2003.
7. Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. Efficient implementation of pairing-based cryptosystems. *J. Cryptology*, 17(4):321–334, 2004.
8. Paulo S. L. M. Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In Bart Preneel and Stafford E. Tavares, editors, *Selected Areas in Cryptography*, volume 3897 of *Lecture Notes in Computer Science*, pages 319–331. Springer, 2005.
9. Waldyr D. Benits Junior and Steven D. Galbraith. Constructing pairing-friendly elliptic curves using Gröbner basis reduction. In Galbraith [19], pages 336–345.
10. Daniel J. Bernstein and Tanja Lange. Explicit-formulas database. http://www.hyperelliptic.org/EFD.
11. Friederike Brezing and Annegret Weng. Elliptic curves suitable for pairing based cryptography. *Des. Codes Cryptography*, 37(1):133–141, 2005.
12. Sanjit Chatterjee, Palash Sarkar, and Rana Barua. Efficient computation of Tate pairing in projective coordinate over general characteristic fields. In Choonsik Park

and Seongtaek Chee, editors, *ICISC*, volume 3506 of *Lecture Notes in Computer Science*, pages 168–181. Springer, 2004.
13. Craig Costello, Hüseyin Hisil, Colin Boyd, Juan Manuel González Nieto, and Kenneth Koon-Ho Wong. Faster pairings on special Weierstrass curves. In Shacham and Waters [40], pages 89–101.
14. M. Prem Laxman Das and Palash Sarkar. Pairing computation on twisted Edwards form elliptic curves. In Galbraith and Paterson [23], pages 192–210.
15. Kirsten Eisenträger, Kristin Lauter, and Peter L. Montgomery. Improved Weil and Tate pairings for elliptic and hyperelliptic curves. In Duncan A. Buell, editor, *ANTS*, volume 3076 of *Lecture Notes in Computer Science*, pages 169–183. Springer, 2004.
16. David Freeman. Constructing pairing-friendly elliptic curves with embedding degree 10. In Florian Hess, Sebastian Pauli, and Michael E. Pohst, editors, *ANTS*, volume 4076 of *Lecture Notes in Computer Science*, pages 452–465. Springer, 2006.
17. David Freeman. A generalized Brezing-Weng algorithm for constructing pairing-friendly ordinary abelian varieties. In Galbraith and Paterson [23], pages 146–163.
18. David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. *J. Cryptology*, 23(2):224–280, 2010.
19. Steven D. Galbraith, editor. *Cryptography and Coding, 11th IMA International Conference, Cirencester, UK, December 18-20, 2007, Proceedings*, volume 4887 of *Lecture Notes in Computer Science*. Springer, 2007.
20. Steven D. Galbraith. Twists of Edwards curves. unpublished manuscript, 2009.
21. Steven D. Galbraith and Xibin Lin. Computing pairings using x-coordinates only. *Des. Codes Cryptography*, 50(3):305–324, 2009.
22. 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.
23. 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*. Springer, 2008.
24. Steven D. Galbraith and Michael Scott. Exponentiation in pairing-friendly groups using homomorphisms. In Galbraith and Paterson [23], pages 211–224.
25. Florian Hess. Pairing lattices. In Galbraith and Paterson [23], pages 18–38.
26. Florian Hess, Nigel P. Smart, and Frederik Vercauteren. The eta pairing revisited. *IEEE Transactions on Information Theory*, 52(10):4595–4602, 2006.
27. Sorina Ionica and Antoine Joux. Another approach to pairing computation in Edwards coordinates. In Dipanwita Roy Chowdhury, Vincent Rijmen, and Abhijit Das, editors, *INDOCRYPT*, volume 5365 of *Lecture Notes in Computer Science*, pages 400–413. Springer, 2008. http://eprint.iacr.org/2008/292.
28. Ezekiel J. Kachisa, Edward F. Schaefer, and Michael Scott. Constructing Brezing-Weng pairing-friendly elliptic curves using elements in the cyclotomic field. In Galbraith and Paterson [23], pages 126–135.
29. Neal Koblitz and Alfred Menezes. Pairing-based cryptography at high security levels. In Nigel P. Smart, editor, *IMA Int. Conf.*, volume 3796 of *Lecture Notes in Computer Science*, pages 13–36. Springer, 2005.
30. Eunjeong Lee, Hyang-Sook Lee, and Cheol-Min Park. Efficient and generalized pairing computation on abelian varieties. Cryptology ePrint Archive, Report 2008/040, 2008. http://eprint.iacr.org/2008/040.

31. Xibin Lin, Changan Zhao, Fangguo Zhang, and Yanming Wang. Computing the ate pairing on elliptic curves with embedding degree $k = 9$. *IEICE Transactions*, 91-A(9):2387–2393, 2008.
32. Seiichi Matsuda, Naoki Kanayama, Florian Hess, and Eiji Okamoto. Optimised versions of the ate and twisted ate pairings. In Galbraith [19], pages 302–312.
33. Nadia El Mrabet, Nicolas Guillermin, and Sorina Ionica. A study of pairing computation for elliptic curves with embedding degree 15. Cryptology ePrint Archive, Report 2009/370, 2009. http://eprint.iacr.org/.
34. Michael Scott. Computing the Tate pairing. In Alfred Menezes, editor, *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 293–304. Springer, 2005.
35. Michael Scott. Faster pairings using an elliptic curve with an efficient endomorphism. In Subhamoy Maitra, C. E. Veni Madhavan, and Ramarathnam Venkatesan, editors, *INDOCRYPT*, volume 3797 of *Lecture Notes in Computer Science*, pages 258–269. Springer, 2005.
36. Michael Scott and Paulo S. L. M. Barreto. Compressed pairings. In Matthew K. Franklin, editor, *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 140–156. Springer, 2004.
37. Michael Scott and Paulo S. L. M. Barreto. Generating more MNT elliptic curves. *Des. Codes Cryptography*, 38(2):209–217, 2006.
38. Michael Scott, Naomi Benger, Manuel Charlemagne, Luis J. Dominguez Perez, and Ezekiel J. Kachisa. Fast hashing to G_2 on pairing-friendly curves. In Shacham and Waters [40], pages 102–113.
39. Michael Scott, Naomi Benger, Manuel Charlemagne, Luis J. Dominguez Perez, and Ezekiel J. Kachisa. On the final exponentiation for calculating pairings on ordinary elliptic curves. In Shacham and Waters [40], pages 78–88.
40. Hovav Shacham 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*. Springer, 2009.
41. Satoru Tanaka and Ken Nakamula. Constructing pairing-friendly elliptic curves using factorization of cyclotomic polynomials. In *Pairing '08: Proceedings of the 2nd international conference on Pairing-Based Cryptography*, pages 136–145, Berlin, Heidelberg, 2008. Springer-Verlag.
42. Frederik Vercauteren. Optimal pairings. *IEEE Transactions on Information Theory*, 56(1):455–461, 2010.
43. Chang-An Zhao, Fangguo Zhang, and Jiwu Huang. A note on the ate pairing. *International Journal of Information Security*, 7(6):379–382, 2008.