Pairing-Friendly Twisted Hessian Curves

Chitchanok Chuengsatiansup[1] and Chloe Martindale[2]

[1] INRIA and ENS de Lyon
46 Allée d'Italie 69364 Lyon Cedex 07, France
chitchanok.chuengsatiansup@ens-lyon.fr
[2] Department of Mathematics and Computer Science
Technische Universiteit Eindhoven
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
c.r.martindale@tue.nl

Abstract. This paper presents efficient formulas to compute Miller doubling and Miller addition utilizing degree-3 twists on curves with j-invariant 0 written in Hessian form. We give the formulas for both odd and even embedding degrees and for pairings on both $\mathbb{G}_1 \times \mathbb{G}_2$ and $\mathbb{G}_2 \times \mathbb{G}_1$. We propose the use of embedding degrees 15 and 21 for 128-bit and 192-bit security respectively in light of the NFS attacks and their variants. We give a comprehensive comparison with other curve models; our formulas give the fastest known pairing computation for embedding degrees 15, 21, and 24.

Keywords: twisted Hessian curves, pairing-friendly curves, ate pairing, degree-3 twists, explicit formulas

1 Introduction

Pairings on elliptic curves have various applications in cryptography, ranging from very basic key exchange protocols, such as one round tripartite Diffie–Hellman [29] [30], to complicated protocols, such as identity-based encryption [8] [26] [22] [47]. Pairings also help to improve currently existing protocols, such as signature schemes, to have shortest possible signatures [9].

Curves that are suitable for pairings are called *pairing-friendly curves*, and these curves must satisfy specific properties. It is extremely rare that a randomly generated elliptic curve is pairing-friendly, so pairing-friendly curves have to be generated in a specific way. Examples of famous and commonly used pairing-friendly curves include Barreto-Naehrig curves [5]

Chitchanok Chuengsatiansup acknowledges the support of Bpifrance in the context of the national projet RISQ (P141580). Chloe Martindale was supported by the Commission of the European Communities through the Horizon 2020 program under CHIST-ERA USEIT (NWO project 651.002.004).

(BN curves), Barreto-Lynn-Scott curves [4] (BLS curves), and Kachisa-Schaefer-Scott curves [33] (KSS curves).

The performance of pairing-based cryptography relies on elliptic-curve-point arithmetic, computation of line functions and pairing algorithms. A pairing is a bilinear map from two elliptic curve groups $\mathbb{G}_1$ and $\mathbb{G}_2$ to a target group $\mathbb{G}_T$. To achieve a good performance, as well as having an efficient pairing algorithm, it is desirable to have a fast elliptic-curve-point arithmetic in both $\mathbb{G}_1$ and $\mathbb{G}_2$.

The security of pairings depends mainly on the cost of solving the discrete logarithm problem (DLP) in the three groups previously mentioned, namely, $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$. Since one can attack pairing-based protocols by attacking any of these three groups, the cost of solving DLP must be sufficiently high in all of these three groups.

1.1 Choice of curves and embedding degrees

One way to improve the performance of pairings is to improve the performance of the underlying point arithmetic. Many authors have studied efficient point arithmetic via the representation of elliptic curves in a specific model, for example, Hessian form [50] [32] and Edwards form [17] [7].

Pairings based on Edwards curves, along with examples of pairing-friendly Edwards curves, were proposed by Arene, Lange, Naehrig and Ritzenthaler [1]. They found that the computation of line functions necessary to compute the pairing is much more complicated than if the curves were written in Weierstrass form. In other words, even though Edwards curves allow faster point arithmetic, this gain is somewhat outweighed by the slower computation of line functions. Li, Wu, and Zhang [40] proposed the use of quartic and sextic twists for Edwards curves, improving the efficiency of both the point arithmetic and the computation of the line functions.

Pairings based on Hessian curves with even embedding degrees were proposed by Gu, Gu and Xie [23]. They provided a geometric interpretation of the group law on Hessian curves along with an algorithm for computing Tate pairing on elliptic curves in Hessian form. However, no pairing-friendly curves in Hessian form were given.

Bos, Costello and Naehrig [10] investigated the possibility of using a model of a curve (such as Edwards or Hessian) allowing for fast point arithmetic and transforming to Weierstrass form for the actual computation of the pairing. They found that for every elliptic curve E in the BN-12, BLS-12, and KSS-18 families of pairing-friendly curves, if E is isomorphic over $\mathbb{F}_q$ to a curve in Hessian or Edwards form, then it is not isomorphic over $\mathbb{F}_{q^k}$ to a curve in Hessian or Edwards form, where k is the embedding degree. This implies that the point arithmetic has to be performed on curves in Weierstrass form — not all curves can be written in special forms such as Hessian or Edwards form. This idea of using different curve models comes at a cost of at least one conversion between other curve models into Weierstrass form.

In this article we study the efficiency of curves in Hessian form for pairing computations. Hessian curves with j-invariant 0 have degree-3 twists that can also be written in Hessian form. This means that we can take full advantage of speed-up techniques for point arithmetic and pairing computations that move arithmetic to subfields via the twist, e.g., as studied for Edwards curves in [40], without the expensive curve conversion to Weierstrass form. We use the families proposed by [20], in which we could find three families that can be written in Hessian form.

Regardless of which model of elliptic curve was being studied, most of the previous articles on this topic were considering even embedding degrees. One of the main advantages of even embedding degrees is the applicability of a denominator elimination technique in the pairing computation (avoiding a field inversion) which does not directly apply to odd embedding degrees. Examples of pairing algorithms for curves in Weierstrass form with odd embedding degree include the work by Lin, Zhao, Zhang and Wang in [41], by Mrabet, Guillermin and Ionica in [43], and by Fouotsa, Mrabet and Pecha in [19].

1.2 Attacks on solving DLP over finite fields

Due to recent advances in number field sieve (NFS) techniques for attacking the discrete logarithm problem for pairing-friendly elliptic curves over finite fields [31] [35] [2] [3] (NFS attacks and their variants), it is necessary to re-evaluate the security of pairing-friendly curves. In [18], Fotiadis and Konstantinou propose countering these attacks by using families with a higher ρ-value. In this paper, we investigate the feasibility of an alternative method: increasing the embedding degree. This has the advantage of keeping the low ρ-value of previously proposed families, but it is disadvantaged by the less efficient pairing computations. This article attempts to analyze the use of Hessian curves in combating this. Previous research on computing pairings with Hessian curves addressed only even embedding degrees, and in order to make use of degree-3 twists the embedding degree should be divisible by 6. Prior to the NFS attacks and their variants, the favoured embedding degree for 128-bit security was 12, so that to increase the embedding degree while making use of cubic twists the next candidate is 15. However, as 15 is odd the formulas of [23] do not apply; for this reason one focus of this article is to provide formulas for embedding degree 15. Similarly, the pre-NFS favourite embedding degree for 192-bit security was 18, which we propose to increase to 21. Observe further that for 192-bit security, the families of [18] all require the embedding degree to be greater than 21.

1.3 Our contributions

We present formulas for computing pairings on both $\mathbb{G}_1 \times \mathbb{G}_2$ and $\mathbb{G}_2 \times \mathbb{G}_1$ for a curve given in Hessian form that admits degree-3 twists. These formulas exploit the degree-3 twists where possible: in moving the point arithmetic in $\mathbb{F}_{q^k}$ to $\mathbb{F}_{q^{k/3}}$ and performing the computations for the line functions in $\mathbb{F}_{q^{k/3}}$ in place of $\mathbb{F}_{q^k}$. For efficient curve arithmetic (before applying the use of twists) we refer to Bernstein, Chuengsatiansup, Kohel, and Lange [6].

We analyze the efficiency of the pairing computation in each case, focussing on the embedding degrees that should correspond to 128- and 192-bit security. Our analysis shows that for embedding degree 12, Hessian curves are outperformed by twisted Edwards curves, but for embedding degrees 15, 21, and 24 our formulas give the most efficient known pairing computation. We do not consider 18 as we do not know of any curve constructions for this case. As explained above, our main focus is on odd embedding degrees, as we propose the use of $k = 15$ and $k = 21$ as a countermeasure against the NFS attacks and their variants.

We also give concrete constructions of pairing-friendly Hessian curves for both embedding degrees and a proof-of-concept implementation of the optimal ate pairing for these cases.

2 Background on Pairings

Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$ where q is a prime. Let r be the largest prime factor of $n = \#E(\mathbb{F}_q) = q + 1 - t$ where t is the trace of Frobenius. The *embedding degree* with respect to r is defined to be the smallest positive integer k such that $r | (q^k - 1)$. 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 $\operatorname{div}(f_{m,P}) = m(P) - ([m]P) - (m-1)(\mathcal{O})$, where $\mathcal{O}$

denotes the neutral element of E. The reduced Tate pairing is defined as

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

We address the computation of the reduced Tate pairing restricted to $\mathbb{G}_1 \times \mathbb{G}_2$, where

$$\mathbb{G}_1 = E[r] \cap \ker(\phi_q - [1]) \text{ and } \mathbb{G}_2 = E[r] \cap \ker(\phi_q - [q]) \subseteq E(\mathbb{F}_{q^k}).$$

Here ϕ_q denotes the q-power Frobenius morphism on E. We denote the restriction of τ_r to $\mathbb{G}_1 \times \mathbb{G}_2$ by

$$e_r : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mu_r.$$

Let $T = t - 1$. We define the *ate pairing* a_T by restricting the Tate pairing to $\mathbb{G}_2 \times \mathbb{G}_1$ so that

$$\begin{aligned}a_T : \mathbb{G}_2 \times \mathbb{G}_1 &\longrightarrow \mu_r \\ (P, Q) &\mapsto\ f_{T,P}(Q)^{\frac{q^k-1}{r}}.\end{aligned}$$

Note that in addition to $\mathbb{G}_1$ and $\mathbb{G}_2$ being switched, the subscript r (i.e., the number of loops) is also changed to T.

Algorithm 1 shows Miller's algorithm to compute the reduced Tate pairing or the ate pairing. Let $m \in \{r, T\}$ and represent the binary format of m by $(m_{n-1}, \ldots, m_1, m_0)_2$. For any two points R, S on E denote by $l_{R,S}$ the line passing through R and S, and by v_R the line passing through R and $-R$. We further define $\ell_{2R} = l_{R,R}/v_{2R}$ and $\ell_{R,P} = l_{R,P}/v_{R+P}$. Miller's algorithm outputs the Tate pairing if $m = r$, $P \in \mathbb{G}_1$, and $Q \in \mathbb{G}_2$, and outputs the ate pairing if $m = T$, $P \in \mathbb{G}_2$, and $Q \in \mathbb{G}_1$.

3 Curve constructions

Even though every elliptic curve can be written in Weierstrass form, only those that contain points of order 3 can be written in (twisted) Hessian form. Almost all methods to generate pairing-friendly curves are for generating pairing-friendly Weierstrass curves, so we find pairing-friendly Hessian curves by searching through constructions of pairing-friendly Weierstrass curves for curves that have points of order 3, and converting those curves into Hessian form. The families that we present below are guaranteed to have points of order 3.

Algorithm 1 Miller's algorithm

```
Require: $m = (m_{n-1}, \ldots, m_1, m_0)_2$ and $P, Q \in E[r]$ with $P \neq Q$
Initialize $R = P$ and $f = 1$
for $i := n-2$ down to $0$ do
    $f \leftarrow f^2 \cdot \ell_{2R}(Q)$
    $R \leftarrow 2R$
    if $m_i = 1$ then
        $f \leftarrow f \cdot \ell_{R,P}(Q)$
        $R \leftarrow R + P$
$f \leftarrow f^{(q^k-1)/r}$
```

In order to give fast formulas for curve arithmetic, it is desirable for the pairing-friendly curves that we consider to have *twists*. Recall that a *degree-d twist* of an elliptic curve $E/\mathbb{F}_q$ is an elliptic curve $E'/\mathbb{F}_{q^e}$ that is isomorphic to E over a degree-d extension of $\mathbb{F}_{q^e}$ but not over any smaller field. Recall also (e.g., [49]) that the only degrees of twists that occur for elliptic curves are $d \in \{2, 3, 4, 6\}$ such that $d|k$, and that degree 3 and 6 twists occur only for elliptic curves with j-invariant 0. We concentrate in this article on twists of degree 3, partly motivated by our aforementioned interest in embedding degrees $k = 15$ and 21. Twisted Hessian curves with j-invariant 0 are of the form

$$\mathcal{H}_a : aX^3 + Y^3 + Z^3 = 0.$$

Suppose that $a \in \mathbb{F}_q$ is a non-cube such that for $\omega \in \mathbb{F}_{q^3}$ with $a = \omega^3$, the element ω generates $\mathbb{F}_{q^k}$ as a $\mathbb{F}_{q^{k/3}}$-vector space. Then $\mathcal{H}_a$ is a degree-3 twist of $\mathcal{H}_1$; the two curves are isomorphic via

$$\begin{array}{rcccl} \varphi : & \mathcal{H}_a & \to & \mathcal{H}_1 & \\ & (X : Y : Z) & \mapsto & (\omega X : Y : Z). & \end{array} \tag{1}$$

In particular, if $R' \in \mathcal{H}_a(\mathbb{F}_{q^{k/3}})$, then $\varphi(R') \in \mathbb{G}_2$. Analogously to [4], we choose the $\mathbb{G}_2$ input point for the pairing from $\varphi(\mathcal{H}_a(\mathbb{F}_{q^{k/3}}))$. The simplicity of the twist isomorphism allows us to do many calculations in $\mathbb{F}_{q^{k/3}}$ instead of $\mathbb{F}_{q^k}$, as explained in detail on a case-by-case basis in Section 4.

3.1 Degree six twists of Hessian curves

In this article we include, for completeness, formulas for computing pairings of Hessian curves with even embedding degree. As we want to make use of the natural twist of degree 3, the embedding degrees that we consider are also divisible by 3, so that we are in fact considering embedding degrees divisible by 6.

As mentioned above, degree-6 twists only occur for elliptic curves with j-invariant 0. Let a and ω be as in the previous section and let $\alpha \in \mathbb{F}_{q^2}$ generate $\mathbb{F}_{q^{k/3}}$ as a $\mathbb{F}_{q^{k/6}}$-vector space. Then

$$\mathbb{F}_{q^k} = \mathbb{F}_{q^{k/6}} + \alpha\mathbb{F}_{q^{k/6}} + \omega\mathbb{F}_{q^{k/6}} + \alpha\omega\mathbb{F}_{q^{k/6}} + \omega^2\mathbb{F}_{q^{k/6}} + \alpha\omega^2\mathbb{F}_{q^{k/6}}.$$

Define the triangular elliptic curve $T/\mathbb{F}_q : \alpha^2 VW(V + aW) = U^3$. Then we can adapt the isomorphism of [6, Theorem 5.3] to see that T is a degree-2 twist of $\mathcal{H}_a$ via the isomorphism

$$\begin{array}{rccc} \psi : & T & \to & \mathcal{H}_a \\ & (U : V : W) & \mapsto & (U : \beta(\alpha V - 54W) : \beta(-\alpha V + 54\zeta_3^2 W)), \end{array} \tag{2}$$

where $\beta = \zeta_3 - \zeta_3^2$ and $\zeta_3 \in \mathbb{F}_q$ is a primitive cube root of unity. In particular, the triangular elliptic curve T is a degree-6 twist of $\mathcal{H}_1$ via the composition $\varphi \circ \psi$, where φ is as given in Equation 1.

3.2 Checking for points of order 3

Let $E/\mathbb{F}_q$ be an elliptic curve. There is a Hessian model of E if and only if $E(\mathbb{F}_q)$ contains a point of order 3. To apply the formulas in the following sections we require both E and the degree-3 twist of E that we consider to have order 3. Recall that $\#E(\mathbb{F}_q) = q + 1 - t$, where t is the trace of Frobenius; by [24] the two non-trivial degree-3 twists E' satisfy:

$$\begin{aligned} \#E'(\mathbb{F}_q) &= q + 1 - (3f - t)/2 && \text{with} \quad t^2 - 4q = -3f^2, \\ \#E'(\mathbb{F}_q) &= q + 1 - (-3f - t)/2 && \text{with} \quad t^2 - 4q = -3f^2. \end{aligned}$$

It is also necessary that, for the twist E' that we use, $\#E'(\mathbb{F}_q)$ is divisible by r (recall that r was the largest prime factor of $\#E(\mathbb{F}_q)$); exactly one of the two possible twists satisfies this condition. So to choose a family for which the elliptic curve E can be rewritten in Hessian form together with a degree-3 twist, it suffices to check that 3 divides $q + 1 - t$ and that $3r$ divides $q + 1 - (\pm 3f - t)/2$ (for one choice of sign).

3.3 Generating curves

Recall that E is an elliptic curve defined over a finite field $\mathbb{F}_q$ where q is prime, and r is the largest prime factor of $\#E(\mathbb{F}_q)$. The embedding degree

k is the smallest integer k such that $r|q^k - 1$. Constructions of parametric families of pairing-friendly curves give an elliptic curve E with integral coefficients and polynomials $q(x)$ and $r(x)$, where for each x_0 such that $q(x_0)$ is prime and $r(x_0)$ has a large prime factor, the reduction of E mod $q(x_0)$ is a pairing-friendly curve with parameters $q = q(x_0)$ and $r = r(x_0)$.

Cyclotomic families are families of curves where the underlying field K is a cyclotomic field, the size r of the largest prime-order subgroup of the group of $\mathbb{F}_q$-point is a cyclotomic polynomial, and the field K contains $\sqrt{-D}$ for some small discriminant D. We searched through [20] and found three cyclotomic-family constructions that satisfy the conditions outlined in the previous section; for each family $D = 3$. The following constructions generate pairing-friendly Weierstrass curves which have a (twisted) Hessian model [6, Section 5]. Note that twists of these curves (see Section 3.2) are also expressible in twisted Hessian form. We denote the cyclotomic polynomial of degree n by $\Phi_n(x)$.

Recall the L-notation: $L_N[\ell, c] = \exp\big((c + o(1))(\ln N)^\ell(\ln\ln N)^{1-\ell}\big)$. The best complexity for NFS attacks up until recently was $L_{q^k}[1/3, 1.923]$, but now due to work of [35] the best complexity for composite k is reduced to $L_{q^k}[1/3, 1.526]$. In particular, with the earlier figure, a 256-bit prime q together with embedding degree $k = 12$ gave a security complexity of 139 bits, but that has now been brought down to 110 bits. To compensate, a pairing implementation using embedding degree 12 aiming for 128-bit security would have to increase the size of the base field to about 364 bits. We propose increasing the embedding degree instead to $k = 15$, for which the base field does not have to increased so dramatically; see details below. Similarly, with the earlier figure, a 384-bit prime q together with embedding degree $k = 18$ gave a security complexity of 194 bits, but that has now been brought down to 154 bits. To compensate, a pairing implementation using embedding degree 18 aiming for 192-bit security now requires $\log(q) \approx 653$, giving $k\log(q) \approx 11754$. We propose increasing the embedding degree instead to $k = 21$ or $k = 24$, for which the base field does not have to be increased so dramatically; see details below.

Construction 1: $k \equiv 3 \pmod{18}$. This construction follows Construction 6.6 in [20]. Pairing-friendly curves with embedding degree $k \equiv 3 \pmod{18}$ can be constructed using the following polynomials:

$$\begin{aligned} r(x) &= \Phi_{2k}(x), \\ t(x) &= x^{k/3+1} + 1, \\ q(x) &= \tfrac{1}{3}(x^2 - x + 1)(x^{2k/3} - x^{k/3} + 1) + x^{k/3+1}. \end{aligned}$$

For this construction, the resulting curves and their twists all have points of order 3. However, there is no such x_0 for which both $q(x_0)$ and $r(x_0)$ are prime. This means that $r(x_0)$ factors, and the largest prime-order subgroup of $E(\mathbb{F}_q)$ actually has less than $r(x_0)$ elements. Recall that the discriminant $D = 3$: the curves are defined by an equation of the form $y^2 = x^3 + b$ and have cubic twists. The ρ-value of this family is $\rho = (2k/3 + 2)/\varphi(k)$ where φ is the Euler φ-function. For $k = 21$ this gives $\rho = 4/3$. To get 192-bit security we have to take r about 420 bits, for which we get $\log(q) \approx 560$ and $k\log(q) \approx 11760$.

Construction 2: $\boldsymbol{k \equiv 9, 15 \pmod{18}}$. This construction follows Construction 6.6 in [20]. Pairing-friendly curves with embedding degree $k \equiv 9, 15 \pmod{18}$ can be constructed using the following polynomials:

$$\begin{aligned} r(x) &= \Phi_{2k}(x), \\ t(x) &= -x^{k/3+1} + x + 1, \\ q(x) &= \tfrac{1}{3}(x+1)^2(x^{2k/3} - x^{k/3} + 1) - x^{2k/3+1}. \end{aligned}$$

This satisfies all the same properties as Construction 1. For $k = 15$ the ρ-value is $\rho = 3/2$. To get 128-bit security we have to take $\log(r) \approx 256$. Then $\log(q) \approx 384$, and $k\log(q) \approx 5760$. This actually gives 143-bit security; a family with a lower ρ-value would be more efficient.

Construction 3: $\boldsymbol{k \equiv 0 \pmod{6}}$ and $\boldsymbol{18 \nmid k}$. This construction follows Construction 6.6 in [20]. Pairing-friendly curves with embedding degree $k \equiv 0 \pmod{6}$ where $18 \nmid k$ can be constructed using the following polynomials:

$$\begin{aligned} r(x) &= \Phi_k(x), \\ t(x) &= x + 1, \\ q(x) &= \tfrac{1}{3}(x-1)^2(x^{k/3} - x^{k/6} + 1) + x. \end{aligned}$$

For this construction, the resulting curves and their twists all have points of order 3. There also exists x_0 such that both $q(x_0)$ and $r(x_0)$ are prime. The curves generated by this construction admit sextic twists. The ρ-value for this construction is given by $\rho = (k/3 + 2)/\varphi(k)$ where φ is the Euler φ-function. For $k = 12$ this gives $\rho = 3/2$ and for $k = 24$ this gives $\rho = 5/4$. To get 192-bit security with $k = 24$ we need $\log(r) \approx 392$, for which $\log(q) \approx 490$ and $k\log(q) \approx 11760$.

For all the constructions outlined above, the curves are given in Weierstrass form as $v^2 = u^3 + b$. To convert a pairing-friendly Weierstrass curve of the above form that has a point (u_3, v_3) of order 3 into twisted Hessian form, we refer to [6]. The authors give explicit transformations showing that there is a Hessian model of the above curve given by $aX^3 + Y^3 + Z^3 = 0$, where $a = 27(u_3^6/v_3^3 - 2v_3)$. Let $\mathbf{m}, \mathbf{s}$ and $\mathbf{m}_c$ denote field multiplication, field squaring and field multiplication by a small constant respectively. They compute the total cost for the whole conversion to be $9\mathbf{m} + 2\mathbf{s} + 5\mathbf{m}_c$ plus one inversion and one cube root computation.

4 Computation of line functions

Each iteration of Miller's loop (Algorithm 1) includes a *Miller doubling* step and some of the iterations also include a *Miller addition* step. The Miller doubling step has four costly parts: computing the double of a point R on the curve, computing the Miller function $\ell_{R,R} = l_{R,R}/v_{2R}$, squaring an element $f \in \mathbb{F}_{q^k}$, and multiplying f^2 by $\ell_{R,R}$. The Miller addition step has three costly parts: computing the sum of two points P and R on the curve, computing the Miller function $\ell_{P,R} = l_{P,R}/v_{P+R}$, and multiplying an element $f \in \mathbb{F}_{q^k}$ by $\ell_{P,R}$. We attempt in the following sections to optimize each of these parts for Hessian curves $\mathcal{H}/\mathbb{F}_q : X^3 + Y^3 + Z^3 = 0$ of j-invariant 0 for pairings on both $\mathbb{G}_1 \times \mathbb{G}_2$ (such as the Tate pairing) and $\mathbb{G}_2 \times \mathbb{G}_1$ (such as the ate pairing).

4.1 Denominator elimination

It is, of course, desirable to avoid the field inversion that results from dividing by $v_{P_1+P_2}(Q)$, with $P_2 = R$ and $P_1 \in \{R, P\}$, which we can do (to some extent). For curves in (twisted) Hessian form, the neutral group element is given by $(0 : -1 : 1)$, and negation by $-(x, y) = (x/y, 1/y)$ (in affine coordinates). This means that the line $v_{P_1+P_2}$ passing through $P_3 = P_1 + P_2$ and $(0 : -1 : 1)$ has a more complicated form than for many other popular curve shapes (such as short Weierstrass or Edwards). Namely, writing $(X_3 : Y_3 : Z_3) = P_3$ and $(X_Q : Y_Q : Z_Q)$, we have

$$v_{P_3}(Q) : (Z_3 + Y_3)X_Q - (Z_Q + Y_Q)X_3.$$

When considering pairings on $\mathbb{G}_1 \times \mathbb{G}_2$, we have that $P_3 \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$, and when considering pairings on $\mathbb{G}_2 \times \mathbb{G}_1$, we have that $P_3 \in \mathbb{G}_2$ and $Q \in \mathbb{G}_1$. As $v_{P_3}(Q) = v_Q(P_3)$, exactly the same arguments apply to $\mathbb{G}_1 \times \mathbb{G}_2$ as to $\mathbb{G}_2 \times \mathbb{G}_1$ in this case; say for simplicity that $P_3 \in \mathbb{G}_1$ and

$Q \in \mathbb{G}_2$. Suppose that we have chosen Q such that there exists $Q' \in \mathcal{H}_a(\mathbb{F}_{q^{k/3}})$ for which $Q = \varphi(Q')$, where φ is the cubic twist isomorphism from Equation 1.

Even embedding degrees. The following is essentially a rephrasing of the denominator elimination technique presented in [23] (although they do not mention pairings on $\mathbb{G}_2 \times \mathbb{G}_1$).

Assume now that $6|k$. In particular, by the discussion in Section 3.1, the triangular curve $T : \alpha^2 VW(V + \omega^3 W) = U^3$, with α and ω as in Section 3.1, defines a quadratic twist of $\mathcal{H}_{\omega^3}$ via the isomorphism ψ of Equation 2. We choose our point $Q' \in \mathcal{H}_{\omega^3}(\mathbb{F}_{q^{k/3}})$ from the image under ψ of $T(\mathbb{F}_{q^{k/6}})$, so that there exist $U, V, W \in \mathbb{F}_{q^{k/6}}$ for which $Q' = (U : \beta(\alpha V - 54W) : \beta(-\alpha V + 54\zeta_3^2 W))$, where $\beta = \zeta_3 - \zeta_3^2$ and $\zeta_3 \in \mathbb{F}_q$ is a primitive cube root of unity. Evaluation of v_{P_3} at $Q = \varphi(Q')$ then gives

$$v_{2R}(Q) : (Z_3 + Y_3)U\omega - 54\beta(\zeta_3^2 - 1)WX_3 \in \mathbb{F}_{q^{k/2}}.$$

This value will go to 1 in the final expontentiation step of Miller's algorithm (Algorithm 1), so without loss of generality we can set it to 1 throughout the computation.

Odd embedding degrees. Unfortunately the denominator elimination technique of [23] does not apply to this case; instead we extend ideas of [41] and [43]. Observe that $\frac{1}{x-y} = \frac{x^2+xy+y^2}{x^3-y^3}$. Let $Q' = (X_{Q'}, Y_{Q'}, Z_{Q'})$. Plugging $x = (Z_3+Y_3)X_{Q'}\omega$ and $y = (Z_{Q'}+Y_{Q'})X_3$ in $\frac{1}{v_{P_3}(Q)}$ with $Q = \varphi(Q')$, we get that the denominator $x^3 - y^3$ is in $\mathbb{F}_{q^{k/3}}$ so will go to 1 in the final exponentiation, hence can be set to 1 for the whole computation. That is, we replace $\frac{1}{v_{P_3}(Q)}$ by the numerator

$$n_{P_3}(Q) = ((Z_3+Y_3)X_{Q'})^2\omega^2 + (Z_3+Y_3)X_{Q'}(1+Y_{Q'})X_3\omega + ((1+Y_{Q'})X_3)^2,$$

and we replace the Miller function $\ell_{P_1,P_2}(Q)$ by $n_{P_3}(Q) \cdot l_{P_1,P_2}(Q)$. The numerator $n_{P_3}(Q)$ can be computed with cost $\frac{2k}{3}\mathbf{m} + \frac{1}{9}\mathbf{M} + \frac{2}{9}\mathbf{S}$ via

$$u = (Z_3 + Y_3)X_{Q'}; \quad v = (1 + Y_{Q'})X_3; \quad n = u^2\omega^2 + (u \cdot v)\omega + v^2.$$

4.2 Miller doubling

Let $R = (X_1 : Y_1 : Z_1) \in \mathcal{H}_b(K)$ for $b \in \{1, a\}$. The fastest known formulas to compute $2R = (X_3 : Y_3 : Z_3)$ (due to [6]) are as follows:

$$T = Y_1^2; \qquad A = Y_1 \cdot T; \qquad S = Z_1^2; \qquad B = Z_1 \cdot S;$$
$$X_3 = X_1 \cdot (A - B); \quad Y_3 = -Z_1 \cdot (2A + B); \quad Z_3 = Y_1 \cdot (A + 2B).$$

The cost for point doubling with the above formulas is $5\mathbf{m} + 2\mathbf{s}$ in K.

In all that follows we denote multiplication and squaring in $\mathbb{F}_q$ by $\mathbf{m}$ and $\mathbf{s}$ respectively, and multiplication and squaring in $\mathbb{F}_{q^k}$ by $\mathbf{M}$ and $\mathbf{S}$ respectively. We also assume always that $3|k$.

Pairings on $\mathbb{G}_1 \times \mathbb{G}_2$. The Miller doubling function is given by

$$\ell_{R,R}(Q) = l_{R,R}(Q)/v_{2R}(Q).$$

For pairings on $\mathbb{G}_1 \times \mathbb{G}_2$ the input points are $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$, and R will be a multiple of P.

We first address the computation of $l_{R,R}(Q)$. This line is the tangent line to $\mathcal{H}_1$ at R evaluated at Q, which is given by $l_{R,R}(Q) : X_1^2 X_Q + TY_Q + S$, where $R = (X_1 : Y_1 : Z_1)$ and $T = Y_1^2$, and $S = Z_1^2$ are the values that were computed in the point doubling computation. Set $Q' = (X_{Q'} : Y_{Q'} : 1)$ and $Q = \varphi(Q')$, where $\varphi : \mathcal{H}_a \to \mathcal{H}_1$ is the twist isomorphism Equation 1 (this is possible as $3|k$). Then we can write $l_{R,R}(Q)$ as

$$l_{R,R}(Q) : (SY_{Q'} + T) + aX_{Q'} X_1^2 \omega,$$

which can be computed with cost $\frac{2k}{3}\mathbf{m} + \mathbf{s}$ via

$$U = X_1^2; \quad V = SY_{Q'}; \quad W = \eta U; \quad l_{R,R}(Q) = V + T + W\omega,$$

where $\eta = aX_{Q'}$ and can be precomputed. We now split into cases.

Even embedding degrees. By Section 4.1, we can set the denominator of the Miller doubling function to 1, so that the computation of the line function $l_{R,R}(Q)$ is in fact the computation of the whole Miller (doubling) function $\ell_{R,R}(Q)$.

Furthermore, a general element of $\mathbb{F}_{q^k}$ considered as element of the $\mathbb{F}_{q^{k/3}}$-vector space generated by ω will be of the form $c_1\omega + c_2\omega^2 + c_3\omega^3$, but for $\ell_{2R}(Q)$ we have that $c_2 = 0$. In particular, the multiplication of $\ell_{2R}(Q)$ with f^2 in Step 3 of Algorithm 1 will not be the full cost of a general multiplication in $\mathbb{F}_{q^k}$ (that is, approximately $k^2\mathbf{m}$), but by schoolbook multiplication will cost 6 multiplications in $\mathbb{F}_{q^{k/3}}$, which amounts to $6\left(\frac{k}{3}\right)^2\mathbf{m} = \frac{2}{3}\mathbf{M}$. Putting together all of the above, the Miller doubling step for even embedding degrees costs

$$\left(5 + \frac{2k}{3}\right)\mathbf{m} + 3\mathbf{s} + \frac{2}{3}\mathbf{M} + \mathbf{S}.$$

Odd embedding degrees. By Section 4.1, we have $\ell_{2R}(Q) = n_{2R}(Q) \cdot l_{R,R}(Q)$, where $n_{2R}(Q)$ is as given in Section 4.1. Putting the above together, the Miller doubling step for odd embedding degrees costs

$$\left(5 + \frac{4k}{3}\right)\mathbf{m} + 3\mathbf{s} + \frac{16}{9}\mathbf{M} + \frac{11}{9}\mathbf{S}.$$

Pairings on $\mathbb{G}_2 \times \mathbb{G}_1$. In this case, the input points are $P \in \mathbb{G}_2$ and $Q \in \mathbb{G}_1$, and R will be a multiple of P. We choose $P = (X_P : Y_P : 1) \in \varphi(\mathcal{H}_a(\mathbb{F}_{p^{k/3}}))$, where φ is the twist isomorphism given in Equation 1. As $R = (X_1 : Y_1 : Z_1)$ is a multiple of P, it is also in the image of $\mathcal{H}_a(\mathbb{F}_{p^{k/3}})$ under φ; let $R' \in \mathcal{H}_a(\mathbb{F}_{p^{k/3}})$ be the pre-image of R under φ. As $2R = 2\varphi(R') = \varphi(2R')$, we can perform the doubling operation on the cubic twist $\mathcal{H}_a$, so that the operation count occurs in $\mathbb{F}_{q^{k/3}}$. That is, point doubling can be performed using 5 multiplications and 2 squarings in $\mathbb{F}_{q^{k/3}}$, which amounts to $\frac{5}{9}\mathbf{M} + \frac{2}{9}\mathbf{S}$. For even embedding degrees this can be done slightly faster, which we address below.

As for pairings on $\mathbb{G}_1 \times \mathbb{G}_2$, we address the computations of the line function

$$l_{R,R}(Q) : X_1^2 X_Q + T Y_Q + S, \qquad (3)$$

where $T = Y_1^2$ and $S = Z_1^2$, in order to compute the Miller doubling function.

Even embedding degrees. Assume now that $6|k$. As described in Section 4.1 we choose the input point from $\mathbb{G}_2$, in this case $P = \varphi(P')$, such that P' is in the image of the quadratic twist isomorphism ψ given in Section 3.1. This implies that $R' = \varphi^{-1}(R)$, as a multiple of P', also lies in this image, so that there exist $U_1, V_1, W_1 \in \mathbb{F}_{q^{k/6}}$ for which

$$R' = (X_1' : Y_1' : Z_1') = (U_1 : \beta(\alpha V_1 - 54 W_1) : \beta(-\alpha V_1 + 54\zeta_3^2 W_1)), \quad (4)$$

where $\beta = \zeta_3 - \zeta_3^2$ and $\zeta_3 \in \mathbb{F}_q$ is a primitive cube root of unity. Here ω and α are as in Section 3.1. We also have $X_1' \in \mathbb{F}_{q^{k/6}}$ and $Y_1', Z_1' \in \mathbb{F}_{q^{k/3}}$.

This gives us a small saving in the point doubling calculation. In the preamble we stated that all the point doubling arithmetic is performed in $\mathbb{F}_{q^{k/3}}$. However, the final step in the computation of X_3' (the X-coordinate of $2R'$) is not a full multiplication in $\mathbb{F}_{q^{k/3}}$ but a multiplication of a $\mathbb{F}_{q^{k/6}}$-element X_1' with a $\mathbb{F}_{q^{k/3}}$-element $(A - B)$, costing $2\left(\frac{k}{6}\right)^2 \mathbf{m} = \frac{1}{18}\mathbf{M}$ using schoolbook multiplication instead of $\frac{1}{9}\mathbf{M}$. So we save $\frac{1}{18}\mathbf{M}$ on the point doubling for even embedding degrees, resulting in $\frac{1}{2}\mathbf{M} + \frac{2}{9}\mathbf{S}$.

As shown in Section 4.1, the Miller doubling function $\ell_{R,R}(Q)$ is just given by the line function $l_{R,R}(Q)$ in this case, the computation of which we now address. As above we have that $R = (X_1 : Y_1 : Z_1) = (X_1'\omega : Y_1' : Z_1')$ so that Equation 3 becomes

$$l_{R,R}(Q) : (X_1')^2 X_Q \omega^2 + TY_Q + S.$$

The values S and T are computed during the point doubling computation and lie in $\mathbb{F}_{q^{k/3}}$, so the computation of $\ell_{R,R}(Q) = l_{R,R}(Q)$ costs an additional squaring in $\mathbb{F}_{q^{k/6}}$, multiplication of a $\mathbb{F}_{q^{k/6}}$-element with a $\mathbb{F}_q$-element, and multiplication of a $\mathbb{F}_{q^{k/3}}$-element with a $\mathbb{F}_q$-element, giving $\frac{k}{2}\mathbf{m} + \frac{1}{36}\mathbf{S}$ via

$$c_1 = (X_1')^2; \quad c_2 = c_1 X_Q; \quad c_3 = TY_Q.$$

Additionally, the formula for $\ell_{R,R}(Q)$ considered as an element of the $\mathbb{F}_{q^{k/6}}$-vector space generated by ω and α has no coefficient of ω, $\alpha\omega$, or $\alpha\omega^2$. Therefore the multiplication of $\ell_{R,R}(Q)$ with a general element (i.e., f^2) of $\mathbb{F}_{q^k}$ costs only $3 \cdot 6 \left(\frac{k}{6}\right)^2 \mathbf{m} = \frac{1}{2}\mathbf{M}$ with schoolbook arithmetic.

Putting the above together, the full Miller doubling step for even embedding degrees costs

$$\frac{k}{2}\mathbf{m} + \mathbf{M} + \frac{5}{4}\mathbf{S}.$$

Odd embedding degrees. By Section 4.1, the Miller doubling function $\ell_{R,R}(Q)$ is given by $\ell_{R,R}(Q) = n_{2R}(Q) \cdot l_{R,R}(Q)$, where $n_{2R}(Q)$ is as given in Section 4.1. As described above for even embedding degrees, we have that

$$l_{R,R}(Q) : (X_1')^2 X_Q \omega^2 + TY_Q + S,$$

where $S, T \in \mathbb{F}_{q^{k/3}}$ and are computed during the point doubling computation. In the case of odd embedding degrees, we have that $X_1' \in \mathbb{F}_{q^{k/3}}$, so that the cost of commutating $l_{R,R}(Q)$ via c_1, c_2, and c_3 as above is $\frac{2k}{3}\mathbf{m} + \frac{1}{9}\mathbf{S}$. Putting the above together, the whole Miller doubling step for odd embedding degrees costs

$$\frac{4k}{3}\mathbf{m} + \frac{7}{3}\mathbf{M} + \frac{14}{9}\mathbf{S}.$$

4.3 Miller addition

Let $P_1 = P = (X_1 : Y_1 : 1)$ and $P_2 = R = (X_2 : Y_2 : Z_2) \in \mathcal{H}_b(K)$ for $b \in \{1, a\}$. The fastest known formulas to compute $P_1 + P_2 = P_3 = (X_3 :$

$Y_3 : Z_3)$ for $P_1 \neq P_2$ (due to [25]) are as follows:

$$\begin{aligned}
&A = X_1 \cdot Z_2; \qquad C = Y_1 \cdot X_2; \qquad D = Y_1 \cdot Y_2; \qquad F = \eta \cdot X_2;\\
&G = (D + Z_2) \cdot (A - C); \quad H = (D - Z_2) \cdot (A + C); \quad X_3 = G - H;\\
&J = (D + F) \cdot (A - Y_2); \quad K = (D - F) \cdot (A + Y_2); \quad Y_3 = K - J;\\
&Z_3 = J + K - G - H - 2(Z_2 - F) \cdot (C + Y_2),
\end{aligned}$$

where $\eta = aX_1$ can be precomputed. The cost for point addition with the above formulas is $9\mathbf{m}$ in K.

Pairings on $\mathbb{G}_1 \times \mathbb{G}_2$. The Miller addition function is given by

$$\ell_{P_1,P_2}(Q) = l_{P_1,P_2}(Q)/v_{P_1+P_2}(Q).$$

For pairings on $\mathbb{G}_1 \times \mathbb{G}_2$ the input points are $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$, and for addition we have that $P_1 = P = (X_1 : Y_1 : 1)$ and $P_2 = R = (X_2 : Y_2 : Z_2)$ is a multiple of P.

The line $l_{P_1,P_2}(Q)$ is the line passing through P and R evaluated at Q. As above we write $Q = (\omega X'_Q : Y'_Q : 1)$ with $Q' = (X_{Q'}, Y_{Q'} : 1) \in \mathcal{H}_a(\mathbb{F}_{q^{k/3}})$. Then

$$l_{P,R}(Q) : (E - Y_2)X_1 + (Y_{Q'} - Y_1)(A - X_2) - (E - Y_2)X_{Q'}\omega,$$

where $E = Y_1 Z_2$, and where A is the value that was computed during the computation of $P + R$. In particular, the cost of computing $l_{P,R}(Q)$ is $\left(2 + \frac{2k}{3}\right)\mathbf{m}$ via

$$\begin{aligned}
&E = Y_1 \cdot Z_2; \qquad L = (E - Y_2) \cdot X_1; \qquad M = (Y_{Q'} - Y_1) \cdot (A - X_2);\\
&N = (E - Y_2) \cdot X_{Q'}; \qquad l_{P,R}(Q) = L + M - N\omega.
\end{aligned}$$

Even embedding degrees. By Section 4.1, the Miller addition function $\ell_{P_1,P_2}(Q)$ is just given by $l_{P_1,P_2}(Q)$ in this case. Also, exactly as for the Miller doubling function, multiplying a general element of $\mathbb{F}_{q^k}$ with $l_{P,R}(Q)$ costs only $\frac{2}{3}\mathbf{M}$. Putting together all of the above, the entire Miller addition step costs

$$\left(11 + \frac{2k}{3}\right)\mathbf{m} + \frac{2}{3}\mathbf{M}.$$

Odd embedding degrees. By Section 4.1, the Miller addition function $\ell_{P_1,P_2}(Q)$ is given by $\ell_{P_1,P_2}(Q) = n_{P_1+P_2}(Q) \cdot l_{P_1,P_2}(Q)$, where $n_{P_1+P_2}(Q)$ is as given in Section 4.1. Putting together all of the above, the entire Miller addition step costs

$$\left(11 + \frac{4k}{3}\right)\mathbf{m} + \frac{16}{9}\mathbf{M} + \frac{2}{9}\mathbf{S}.$$

Pairings on $\mathbb{G}_2 \times \mathbb{G}_1$. For pairings on $\mathbb{G}_2 \times \mathbb{G}_1$ the input points $P \in \mathbb{G}_2$ and $Q \in \mathbb{G}_1$, and in the Miller addition function $\ell_{P_1,P_2}(Q)$ we have that $P_1 = P = (X_1 : Y_1 : 1)$ and $P_2 = R = (X_2 : Y_2 : Z_2)$, which is some multiple of P. In exactly the same way as discussed for the Miller doubling function, the point addition can be performed in the group $\mathcal{H}_a(\mathbb{F}_{q^{k/3}})$ in place of $\mathcal{H}(\mathbb{F}_{q^k})$, so that the operation count occurs in $\mathbb{F}_{q^{k/3}}$. That is, point addition can be performed using 9 multiplications in $\mathbb{F}_{q^{k/3}}$, which amounts to $1\mathbf{M}$. For even embedding degrees this can be done faster, which we address below. As for pairings on $\mathbb{G}_1 \times \mathbb{G}_2$, we will need to compute the line function

$$l_{P,R}(Q) : -(E - Y_2)X_Q + (E - Y_2)X_1 + (Y_Q - Y_1)(A - X_2),$$

where $E = Y_1 Z_2$ and $A = X_1 Z_2$. Let $P = \varphi(P')$ and $R = \varphi(R')$ be the images of $P' = (X_1', Y_1', 1)$ and $R' = (X_2', Y_2', Z_2') \in \mathcal{H}_a(\mathbb{F}_{q^{k/3}})$ respectively under the twist isomorphism φ of Equation 1. Then

$$l_{P,R}(Q) : -(E' - Y_2')X_Q + (C' - Y_2'X_1' + Y_Q(A' - X_2'))\omega,$$

where $E' = Y_1'Z_2'$, $A = A'\omega$, $C = C'\omega$ and $A' = X_1'Z_2'$ and $C' = Y_1'X_2'$ are the values that were computed during the point addition. This can be computed in $\frac{2k}{3}\mathbf{m} + \frac{2}{9}\mathbf{M}$ via

$$E' = Y_1' \cdot Z_2'; \quad d_1 = Y_2' \cdot X_1'; \quad d_2 = (E' - Y_2') \cdot X_Q; \quad d_3 = (A' - X_2') \cdot Y_Q.$$

Even embedding degrees. Suppose now that $6|k$. As described already for Miller doubling, we may choose $U_2, V_2, W_2 \in \mathbb{F}_{q^{k/6}}$ such that

$$R' = (U_2 : \beta(\alpha V_2 - 54W_2) : \beta(-\alpha V_2 + 54\zeta_3^2 W_2)),$$

where $\beta = \zeta_3 - \zeta_3^2$ and $\zeta_3 \in \mathbb{F}_q$ is a primitive cube root of unity (c.f. Equation 4). Note that we do not apply this to P because we want to make use of the mixed addition with $Z_1 = 1$.

This gives us a small saving in the point addition calculation: the computations of C and of F now cost $\frac{1}{18}\mathbf{M}$ each instead of $\frac{1}{9}\mathbf{M}$ each, saving $\frac{1}{18}\mathbf{M}$; the cost for point addition is therefore $\frac{8}{9}\mathbf{M}$.

As shown in Section 4.1, the Miller addition function $\ell_{P,R}(Q)$ is just given by the line function $l_{P,R}(Q)$ in this case. Multiplication of a general element in $\mathbb{F}_{q^k}$ with $\ell_{P,R}(Q)$ costs only $\frac{2}{3}\mathbf{M}$ as $\ell_{P,R}(Q)$ has no coefficient of ω^2. Putting together all of the above, we get the cost for the whole Miller addition step

$$\frac{2k}{3}\mathbf{m} + \frac{16}{9}\mathbf{M}.$$

Odd embedding degrees. By Section 4.1, the Miller addition function $\ell_{P,R}(Q)$ is given by $n_{P+R}(Q) \cdot l_{P,R}(Q)$ in this case, where $n_{P+R}(Q)$ is as given in Section 4.1. Putting together all of the above, we get the cost for the full Miller addition step

$$\frac{4k}{3}\mathbf{m} + 3\mathbf{M} + \frac{2}{9}\mathbf{S}.$$

5 Comparison

As this paper primarily concerns cubic twists, we only discuss results for embedding degrees that are divisible by 3. To our knowledge, most of the previous work on the optimization of operation counts for one iteration of Miller's loop concentrated on pairings for $\mathbb{G}_1 \times \mathbb{G}_2$. To properly compare different results, we need to take into account the number of iterations of Miller's loop, which differs greatly between $\mathbb{G}_1 \times \mathbb{G}_2$ and $\mathbb{G}_2 \times \mathbb{G}_1$.

For pairings on $\mathbb{G}_1 \times \mathbb{G}_2$, the lowest number of iterations occurs for the twisted ate pairing when twists are available, or the reduced Tate pairing when twists are not available. In this paper, we explicitly address the first case, so the twisted ate pairing gives the minimal number of iterations. Let t be the trace of Frobenius, let $T = t - 1$, and let d be the degree of the twist. The number of iterations of Miller's loop for the twisted ate pairing is given by $\log(T_e)$, where $T_e \equiv T^e \pmod{r}$ and $1 < e|d$. Also T is a d-th root of unity in $\mathbb{F}_r$, so when $d = 6$ the smallest value of $\log(T_e)$ is $\log(T_2) \approx \log(r)/3$, and when $d = 3$ the smallest value of $\log(T_e)$ is $\log(T_3) \approx \log(r)$. For more details on the twisted ate pairing see [24].

For pairings on $\mathbb{G}_2 \times \mathbb{G}_1$, the lowest number of iterations occurs for the optimal ate pairing. The best-case-scenario (which can in principle occur for any embedding degree) is $\log(r)/\varphi(k)$ iterations of Miller's loop, where φ is the Euler φ-function. This scenario takes x as the input for the Miller's algorithm (e.g., in place of $r = r(x)$ as in Tate). For more details on the optimal ate pairing see [51].

We compared previous results in this area for Weierstrass curves with Jacobian coordinates [27] [1], Weierstrass curves with projective coordinates [15], Edwards curves [1], Edwards curves with sextic twists [40], and Hessian curves with quadratic twists [23]. Most of these papers considered only pairings on $\mathbb{G}_1 \times \mathbb{G}_2$ (many of them were written before Vercauteren's paper [51] on optimal pairings) and only even embedding degree (to avoid dealing with denominators).

5.1 Comparing results for $\mathbb{G}_2 \times \mathbb{G}_1$

The only other paper containing operation counts for pairings on $\mathbb{G}_2 \times \mathbb{G}_1$ and embedding degree divisible by 3, to our knowledge, is [15], which considers projective Weierstrass coordinates. In that paper they look at even embedding degrees, so we only compare our results for the optimal ate pairing when $k = 12$ and 24 (c.f. Construction 3). Assume for simplicity that $\mathbf{s} \approx 0.8\mathbf{m}$. The formulas presented in [15] give an operation count of

$$\tfrac{41}{36}\mathbf{M} + \tfrac{41}{36}\mathbf{S} \approx \begin{cases} 295.2\mathbf{m} & k = 12 \\ 1180.8\mathbf{m} & k = 24 \end{cases} \text{ for Miller doubling and}$$

$$\tfrac{4}{3}\mathbf{M} + \tfrac{1}{18}\mathbf{S} \approx \begin{cases} 198.4\mathbf{m} & k = 12 \\ 793.6\mathbf{m} & k = 24 \end{cases} \text{ for Miller addition.}$$

The formulas presented in this paper give an operation count of

$$\tfrac{k}{2}\mathbf{m} + \mathbf{M} + \tfrac{5}{4}\mathbf{S} \approx \begin{cases} 294.0\mathbf{m} & k = 12 \\ 1164.0\mathbf{m} & k = 24 \end{cases} \text{ for Miller doubling and}$$

$$\tfrac{2k}{3}\mathbf{m} + \tfrac{16}{9}\mathbf{M} \approx \begin{cases} 264.0\mathbf{m} & k = 12 \\ 1040.0\mathbf{m} & k = 24 \end{cases} \text{ for Miller addition.}$$

As the formulas for Hessian form are faster for doubling but slower for adding (with respect to projective Weierstrass form), there is a tradeoff to assess. Suppose that we wish to compute the optimal ate pairing and that we have an example for which the input for Miller's algorithm is x. The pairing can then be computed in $\log(x) = \log(r)/\varphi(k)$ iterations of Miller's loop — this amounts to $O(\log(x))$ Miller doubling steps, $O(\mathsf{Ham}(x))$ Miller addition steps, where $\mathsf{Ham}(x)$ denotes the Hamming weight of x, and the final exponentiation. When $k = 12$, the formulas presented in [15] compute the pairing in $\approx 295.2 \cdot O(\log(x)) + 198.4 \cdot O(\mathsf{Ham}(x))$ multiplications in $\mathbb{F}_q$ and an exponentiation, and the formulas presented in this paper compute the pairing in $\approx 294.0 \cdot O(\log(x)) + 264.0 \cdot O(\mathsf{Ham}(x))$ multiplications in $\mathbb{F}_q$ and an exponentiation. That is, the formulas using Hessian curves outperform the projective Weierstrass curves only for an x-value such that $\log(x) > 54.67 \cdot \mathsf{Ham}(x)$. When $k = 24$, the formulas using Hessian curves outperform the projective Weierstrass curves for an x-value such that $\log(x) > 14.67 \cdot \mathsf{Ham}(x)$.

5.2 Comparing results for $\mathbb{G}_1 \times \mathbb{G}_2$

Comparing the aforementioned papers [27], [1], [15], [40], [23], and our results, we see that the fastest curve model for embedding degree divisible by 6 together with a $\mathbb{G}_1 \times \mathbb{G}_2$ pairing is the Edwards form with sextic twists [40] using $\left(\frac{4k}{3}+4\right)\mathbf{m}+7\mathbf{s}+\frac{1}{3}\mathbf{M}+\mathbf{S}$ for one Miller doubling step and $\left(\frac{4k}{3}+12\right)\mathbf{m}+\frac{1}{3}\mathbf{M}$ for one Miller addition step. The fastest curve model for odd embedding degree divisible by 3 together with a $\mathbb{G}_1 \times \mathbb{G}_2$ pairing is the projective Weierstrass form [15] using $(k+6)\mathbf{m}+7\mathbf{s}+\mathbf{M}+\mathbf{S}$ for one Miller doubling step and $(k+13)\mathbf{m}+3\mathbf{s}+\mathbf{M}$ for one Miller addition step.

5.3 Comparing $\mathbb{G}_1 \times \mathbb{G}_2$ and $\mathbb{G}_2 \times \mathbb{G}_1$

In the following table we compare the operation counts from the *most efficient* curve shape for each subcase (optimal ate vs. twisted ate and even vs. odd) in what we hope is a meaningful way: we give the number of $\mathbb{F}_q$-multiplications per Miller doubling/addition multiplied by $\frac{1}{\log(r)}\times$ the number of iterations. We call these numbers DBLc (for doubling compare) and ADDc (for addition compare). We assume here that $\mathbf{s} = 0.8\mathbf{m}$ for simplicity. The most efficient option for each subcase is as follows.

Table 1. Best operation counts for DBLc and ADDc for each embedding degree and type of pairing

k	pairing	Model	# iterations	DBLc	ADDc
12	twisted ate	Edwards [40]	$\log(r)/3$	62.9	25.3
	optimal ate	Projective [15]	$\log(r)/4$	73.8	49.6
	optimal ate	Hessian (this paper)	$\log(r)/4$	73.5	66.0
15	twisted ate	Projective [15]	$\log(r)$	431.6	255.4
	optimal ate	Hessian (this paper)	$\log(r)/8$	103.1	120.0
21	twisted ate	Projective [15]	$\log(r)$	826.4	477.4
	optimal ate	Hessian (this paper)	$\log(r)/12$	133.8	155.9
24	twisted ate	Edwards [40]	$\log(r)/3$	231.5	78.7
	optimal ate	Projective [15]	$\log(r)/8$	147.5	99.2
	optimal ate	Hessian (this paper)	$\log(r)/8$	140.7	134.0

For embedding degree 12, [40] is clearly the most efficient. For embedding degrees 15 and 21, our results are clearly the most efficient. For embedding degree 24, doubling is more efficient in Hessian form with optimal ate while addition is more efficient in Edwards form with twisted ate. We could assess this trade-off in a similar way to the trade-off that was required to compare results for even embedding degrees for optimal ate pairings; our results will outperform those of [40] when the Hamming weight of x is sufficiently low compared to $\log(x)$.

Not included in Table 1 are the precomputation costs (which are relatively low for our constructions) and the final exponentiation costs (which are roughly uniform across all curve shapes). A significant part of the precomputation cost for many models is the conversion between curve models, which is not necessary for our constructions. (Recall that for BN, BLS, and KSS, this conversion is always necessary if one wants to take advantage of the fast point arithmetic on Hessian or Edwards curves, as proven in [10].)

6 Concluding remarks

This paper presents efficient formulas to compute Miller doubling and Miller addition on curves of j-invariant 0 with embedding degree divisible by 3 when written in Hessian form. This paper presents formulas for both pairings of the form $\mathbb{G}_1 \times \mathbb{G}_2$ and $\mathbb{G}_2 \times \mathbb{G}_1$ and compares the efficiency of these formulas to the best known formulas of previous research. We present the first formulas for pairings on $\mathbb{G}_2 \times \mathbb{G}_1$ that utilize twists of degree 3 in the case of odd embedding degrees, and the first formulas that utilize twists of degree 3 for Hessian curves in all cases. Our formulas for embedding degrees 15, 21, and (subject to trade-offs) 24 are the most efficient among known choices.

Curves generated by the methods used in this paper (originally due to [20]) are guaranteed to have twists of degree 3 and have embedding degree $k \equiv 3, 9, 15 \pmod{18}$ or $k \equiv 0 \pmod{6}$ where $18 \nmid k$. We suggest updating the use of embedding degree 12 to 15 for 128-bit security and 18 to 21 for 192-bit security in light of the NFS attacks and their variants. This allows us to keep the relatively small primes for the base field and a low ρ-value. We additionally suggest including $k = 24$ in any future (more precise) comparisons, as our results show that this may be competitive with $k = 21$ (since the ρ-value for $k = 24$ is lower than that of $k = 21$).

In future work, we plan to study precisely how the NFS attacks and their variants apply to our constructions in order to be able to properly evaluate the security and propose concrete parameters. A comparison between the larger embedding degrees (but low ρ-value) that we suggest in this paper and the higher ρ-value (but small embedding degrees) suggested in [18] would be very interesting, but we leave this for future work. It would also be interesting to evaluate the performance of other curve models with degree 3 twists on $\mathbb{G}_2 \times \mathbb{G}_1$ pairings. We also consider the optimized implementation as future work.

References

1. Christophe Arene, Tanja Lange, Michael Naehrig, and Christophe Ritzenthaler. Faster Computation of the Tate Pairing. *IACR Cryptology ePrint Archive*, 2009:155, 2009. http://eprint.iacr.org/2009/155.
2. Razvan Barbulescu, Pierrick Gaudry, Aurore Guillevic, and François Morain. Improving NFS for the Discrete Logarithm Problem in Non-prime Finite Fields. In *Eurocrypt 2015 [44]*, pages 129–155, 2015.
3. Razvan Barbulescu, Pierrick Gaudry, and Thorsten Kleinjung. The Tower Number Field Sieve. In *Asiacrypt 2015 [28]*, pages 31–55, 2015.
4. Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. On the Selection of Pairing-Friendly Groups. In *SAC 2003 [42]*, pages 17–25, 2003.
5. Paulo S.L.M. Barreto and Michael Naehrig. Pairing-Friendly Elliptic Curves of Prime Order. In *SAC 2005 [45]*, pages 319–331, 2006. http://cryptosith.org/papers/pfcpo.pdf.
6. Daniel J. Bernstein, Chitchanok Chuengsatiansup, David Kohel, and Tanja Lange. Twisted Hessian Curves. In *LATINCRYPT 2015 [39]*, pages 269–294, 2015. http://cr.yp.to/papers.html#hessian.
7. Daniel J. Bernstein and Tanja Lange. Faster addition and doubling on elliptic curves. In *Asiacrypt 2007 [37]*, pages 29–50, 2007. http://cr.yp.to/newelliptic/newelliptic-20070906.pdf.
8. Dan Boneh and Matthew K. Franklin. Identity-Based Encryption from the Weil Pairing. In *CRYPTO 2001 [34]*, pages 213–229, 2001. http://www.iacr.org/archive/crypto2001/21390212.pdf.
9. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. *Journal of Cryptology*, 17(4):297–319, 2004. http://crypto.stanford.edu/~dabo/pubs/papers/weilsigs.ps.
10. Joppe W. Bos, Craig Costello, and Michael Naehrig. Exponentiating in Pairing Groups. In *SAC 2013 [38]*, 2013. https://eprint.iacr.org/2013/458.pdf.
11. Wieb Bosma, editor. *Algorithmic Number Theory, 4th International Symposium, ANTS-IV, Leiden, The Netherlands, July 2–7, 2000, proceedings*, volume 1838 of *Lecture Notes in Computer Science*. Springer, 2000.
12. Zhenfu Cao and Fangguo Zhang, editors. *Pairing-Based Cryptography — Pairing 2013, 6th International Conference, Beijing, China, November 22–24, 2013, Revised Selected Papers*, volume 8365 of *Lecture Notes in Computer Science*. Springer, 2014.
13. Çetin Kaya Koç, David Naccache, and Christof Paar, editors. *Cryptographic hardware and embedded systems — CHES 2001, third international workshop, Paris, France, May 14–16, 2001, proceedings*, volume 2162 of *Lecture Notes in Computer Science*. Springer, 2001.
14. 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*. Springer, 2008.
15. Craig Costello, Hüseyin Hisil, Colin Boyd, Juan Manuel González Nieto, and Kenneth Koon-Ho Wong. Faster Pairings on Special Weierstrass Curves. In *Pairing 2009 [48]*, pages 89–101, 2009.
16. Ronald Cramer, editor. *Advances in Cryptology — EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22–26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*. Springer, 2005.

17. 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.
18. Georgios Fotiadis and Elisaveth Konstantinou. TNFS resistant families of pairing-friendly elliptic curves. *Journal of Theoretical Computer Science*, 2018. (to appear).
19. Emmanuel Fouotsa, Nadia El Mrabet, and Aminatou Pecha. Optimal Ate Pairing on Elliptic Curves with Embedding Degree 9, 15 and 27. *IACR Cryptology ePrint Archive*, 2016:1187, 2016. http://eprint.iacr.org/2016/1187.
20. David Freeman, Michael Scott, and Edlyn Teske. A Taxonomy of Pairing-Friendly Elliptic Curves. *Journal of Cryptology*, 23(2):224–280, 2010. http://eprint.iacr.org/2006/372/.
21. 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.
22. Craig Gentry and Alice Silverberg. Hierarchical ID-Based Cryptography. In *Asiacrypt 2002 [52]*, pages 548–566, 2002. http://www.cs.ucdavis.edu/~franklin/ecs228/pubs/extra_pubs/hibe.pdf.
23. Haihua Gu, Dawu Gu, and WenLu Xie. Efficient Pairing Computation on Elliptic Curves in Hessian Form. In *ICISC 2010*, pages 169–176, 2010.
24. Florian Hess, Nigel P. Smart, and Frederik Vercauteren. The Eta Pairing Revisited. *IEEE Transactions on Information Theory*, 52(10):4595–4602, 2006. http://eprint.iacr.org/2006/110.
25. Hüseyin Hışıl. *Elliptic curves, group law, and efficient computation.* PhD thesis, Queensland University of Technology, 2010.
26. Jeremy Horwitz and Ben Lynn. Toward Hierarchical Identity-Based Encryption. In *Eurocrypt 2002 [36]*, pages 466–481, 2002. http://theory.stanford.edu/~horwitz/pubs/hibe.pdf.
27. Sorina Ionica and Antoine Joux. Another Approach to Pairing Computation in Edwards Coordinates. In *INDOCRYPT 2008 [14]*, pages 400–413, 2008.
28. Tetsu Iwata and Jung Hee Cheon, editors. *Advances in Cryptology — ASIACRYPT 2015, 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 – December 3, 2015, Proceedings, Part II*, volume 9453 of *Lecture Notes in Computer Science*. Springer, 2015.
29. Antoine Joux. A One Round Protocol for Tripartite Diffie-Hellman. In *ANTS-IV [11]*, pages 385–393, 2000. http://cgi.di.uoa.gr/~aggelos/crypto/page4/assets/joux-tripartite.pdf.
30. Antoine Joux. A One Round Protocol for Tripartite Diffie-Hellman. *Journal of Cryptology*, 17(4):263–276, 2004.
31. Antoine Joux and Cécile Pierrot. The Special Number Field Sieve in $\mathbb{F}_{p^n}$, Application to Pairing-Friendly Constructions. In *Pairing 2013 [12]*, pages 45–61, 2013.
32. Marc Joye and Jean-Jacques Quisquater. Hessian elliptic curves and side-channel attacks. In *CHES 2001 [13]*, pages 402–410, 2001. http://joye.site88.net/.
33. Ezekiel J. Kachisa, Edward F. Schaefer, and Michael Scott. Constructing Brezing-Weng Pairing-Friendly Elliptic Curves Using Elements in the Cyclotomic Field. In *Pairing 2008 [21]*, pages 126–135, 2008.
34. Joe Kilian, editor. *Advances in Cryptology — CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19–23,*

2001, Proceedings, volume 2139 of *Lecture Notes in Computer Science*. Springer, 2001.

35. Taechan Kim and Razvan Barbulescu. Extended Tower Number Field Sieve: A New Complexity for the Medium Prime Case. In *CRYPTO 2016 [46]*, pages 543–571, 2016.
36. Lars R. Knudsen, editor. *Advances in Cryptology — EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 – May 2, 2002, proceedings*, volume 2332 of *Lecture Notes in Computer Science*. Springer, 2002.
37. Kaoru Kurosawa, editor. *Advances in Cryptology — ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2–6, 2007, Proceedings*, volume 4833 of *Lecture Notes in Computer Science*. Springer, 2007.
38. Tanja Lange, Kristin Lauter, and Petr Lisonek, editors. *Selected areas in cryptography, 20th international conference, SAC 2013, Burnaby, BC, Canada, August 14–16, 2013, Revised Selected Papers*, volume 8282 of *Lecture Notes in Computer Science*. Springer, 2014.
39. Kristin E. Lauter and Francisco Rodríguez-Henríquez, editors. *Progress in Cryptology — LATINCRYPT 2015, 4th International Conference on Cryptology and Information Security in Latin America, Guadalajara, Mexico, August 23–26, 2015, Proceedings*, volume 9230 of *Lecture Notes in Computer Science*. Springer, 2015.
40. Liangze Li, Hongfeng Wu, and Fan Zhang. Pairing Computation on Edwards Curves with High-Degree Twists. In *Inscrypt 2013*, 2014. https://doi.org/10.1007/978-3-319-12087-4_12.
41. 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.
42. Mitsuru Matsui and Robert J. Zuccherato, editors. *Selected Areas in Cryptography, 10th Annual International Workshop, SAC 2003, Ottawa, Canada, August 14–15, 2003, Revised Papers*, volume 3006 of *Lecture Notes in Computer Science*. Springer, 2004.
43. Nadia El Mrabet, Nicolas Guillermin, and Sorina Ionica. A study of pairing computation for elliptic curves with embedding degree 15. *IACR Cryptology ePrint Archive*, 2009:370, 2009. http://eprint.iacr.org/2009/370.
44. Elisabeth Oswald and Marc Fischlin, editors. *Advances in Cryptology — EUROCRYPT 2015, 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26–30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*. Springer, 2015.
45. Bart Preneel and Stafford E. Tavares, editors. *Selected Areas in Cryptography, 12th International Conference, SAC 2005, Kingston, ON, Canada, August 11–12, 2005, Revised Selected Papers*, volume 3897 of *Lecture Notes in Computer Science*. Springer, 2006.
46. Matthew Robshaw and Jonathan Katz, editors. *Advances in Cryptology — CRYPTO 2016, 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14–18, 2016, Proceedings, Part I*, volume 9814 of *Lecture Notes in Computer Science*. Springer, 2016.
47. Amit Sahai and Brent Waters. Fuzzy Identity-Based Encryption. In *Eurocrypt 2005 [16]*, pages 457–473, 2005. http://eprint.iacr.org/2004/086/.
48. Hovav Shacham and Brent Waters, editors. *Pairing-Based Cryptography — Pairing 2009, Third International Conference, Palo Alto, California, USA, August 12–14,*

2009, proceedings, volume 5671 of *Lecture Notes in Computer Science*. Springer, 2009.

49. Joseph H. Silverman. *The Arithmetic of Elliptic Curves*, volume 106 of *Graduate Texts in Mathematics*. Springer, 2009.
50. Nigel P. Smart. The Hessian form of an elliptic curve. In *CHES 2001 [13]*, pages 118–125, 2001.
51. Frederik Vercauteren. Optimal pairings. In *IEEE Transactions on Information Theory 56(1)*, pages 455–461, 2010.
52. Yuliang Zheng, editor. *Advances in Cryptology — ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1–5, 2002, Proceedings*, volume 2501 of *Lecture Notes in Computer Science*. Springer, 2002.