Pairing-Friendly Twisted Hessian Curves

No Author Given

No Institute Given

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 applied to the optimal ate pairing give the fastest known pairing implementation for embedding degrees 15 and 21.

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] [23] [48]. 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] (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 also 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 [51] [32] or Edwards form [18] [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 [24]. 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, for example as studied for Edwards curves in [40], without the expensive curve conversion to Weierstrass form. We use the families proposed by [21], 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 [20].

1.2 Attacks on solving DLP over finite fields

Due to recent advances in number field sieve techniques for attacking the discrete logarithm problem for pairing-friendly elliptic curves over finite fields [31] [35] [2] [3] (NFS attacks), it is necessary to re-evaluate the security of pairing-friendly curves. In [19], Fotiadis and Konstantinou propose countering these attacks by using families with a higher ρ-value. In this paper, we investigate the feasability 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 combatting this. Previous research on computing pairings with Hessian curves addressed only even emebedding 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 [24] do not apply; for this reason one focus of this article is to provide formulas for embedding degree 15. Simlarly, 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 [19] 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. Our analysis shows that for even embedding degrees, Hessian curves are outperformed by twisted Edwards curves for pairings on $\mathbb{G}_1 \times \mathbb{G}_2$, and by Weierstrass curves for pairings on $\mathbb{G}_2 \times \mathbb{G}_1$. However, as explained above, our 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. For embedding degrees 15 and 21 we show that the most efficient pairing computation available is the optimal ate pairing with the Hessian form proposed in this paper.

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.

The setup of this paper is as follows: in Section 2, we recall Miller's algorithm to compute pairings on elliptic curves. In Section 3, we state constructions of families of pairing-friendly elliptic curves given in [21] that can be written in twisted Hessian form, together with the conversion to twisted Hessian form. In Section 4, we provide explicit formulas for the computation of pairings for both $\mathbb{G}_1 \times \mathbb{G}_2$ and $\mathbb{G}_2 \times \mathbb{G}_1$ on twisted Hessian curves using the state-of-the-art point arithmetic formulas presented in [6]. In Section 5, we compare our results to previous work. In Section 6, we present our conclusions.

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 $\mathrm{div}(f_{m,P}) = m(P) - ([m]P) - (m-1)(\mathcal{O})$, where $\mathcal{O}$

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

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

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 a 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 $P \neq Q$

1: Initialize $R = P$ and $f = 1$
2: **for** $i := n - 2$ **down to** 0 **do**
3: $\quad f \leftarrow f^2 \cdot \ell_{2R}(Q)$
4: $\quad R \leftarrow 2R$
5: $\quad$ **if** $m_i = 1$ **then**
6: $\quad\quad f \leftarrow f \cdot \ell_{R,P}(Q)$
7: $\quad\quad R \leftarrow R + P$
8: $f \leftarrow f^{(q^k-1)/r}$

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. [50]) 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, as motivated by our aforementioned interest in embedding degrees $k = 15$ and 21. Hessian curves with j-invariant zero are of the form

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

This motivates our interest in Hessian curves in particular: Let

$$\mathcal{H}_\omega : a\omega^3 X^3 + Y^3 + Z^3 = 0,$$

where $\omega \in \mathbb{F}_{q^k}$ generates $\mathbb{F}_{q^k}$ as a $\mathbb{F}_{q^{k/3}}$-vector space. Then $\mathcal{H}_\omega$ is a degree-3 twist of $\mathcal{H}$; the two curves are isomorphic via

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

In particular, if $R' \in \mathcal{H}_\omega(\mathbb{F}_{q^{k/3}})$, then $\varphi(R') \in \mathbb{G}_2$. Analogously to [4], we choose the $\mathbb{G}_2$ input point for the pairing from $\varphi(\mathcal{H}_\omega(\mathbb{F}_{q^{k/3}}))$. 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 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 [25] the two non-trivial degree-3 twists E' satisfy:

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

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 $\#E(\mathbb{F}_q)$) and 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).

3.2 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 coeffiients 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 $\mathbb{F}_q$-points is a cyclotomic polynomial, and the field K contains $\sqrt{-D}$ for some small discriminant D. We searched through [21] and found three cyclotomic-family constructions that satisfy the conditions outlined in the previous section. These constructions are based on a cyclotomic field containing a cube root of unity, i.e., fields contain $\sqrt{-3}$. Therefore, we choose the discriminant $D = 3$.

The following constructions are for generating pairing-friendly Weierstrass curves which have a (twisted) Hessian model [6, Section 5]. Note

that twists of these curves (see Section 3.1) are also expressible in twisted Hessian form. We categorized these constructions by embedding degrees. Constructions 1 and 2 are for embedding degree divisible by 3, but not 6; our analysis shows that for embedding degree divisible by 6 there are other choices of curve shapes that are preferable to Hessian form (with respect to the efficiency of the pairing implementation), so these are the most relevant constructions for this paper. Construction 3 is for even embedding degree divisible by 6 which we include only for completeness. Note that $q(x)$ is a prime, $r(x)$ is the size of a subgroup of $E(\mathbb{F}_{q(x)})$ (which may not have prime order), and $t(x)$ is a trace of Frobenius of $E/\mathbb{F}_{q(x)}$. We denote the cyclotomic polynomial of degree n by $\Phi_n(x)$.

Construction 1: $k \equiv 3 \pmod{18}$. This construction follows Construction 6.6 in [21]. Pairing-friendly curves with embedding degree $k \equiv 3 \pmod{18}$ can be constructed using the following polynomials:

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

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 $D = 3$. This implies in particular that the curves are defined by an equation of the form $y^2 = x^3 + b$. The only possible twists are cubic ($d = 3$) 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 $\rho = 4/3$.

Construction 2: $k \equiv 9, 15 \pmod{18}$. This construction follows Construction 6.6 in [21]. Pairing-friendly curves with embedding degree $k \equiv 9, 15 \pmod{18}$ can be constructed using the following polynomials:

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

This satisfies all the same properties as Construction 1. For $k = 15$ the ρ-value is $\rho = 3/2$.

Construction 3: $k \equiv 0 \pmod 6$ and $18 \nmid k$. This construction follows Construction 6.6 in [21]. Pairing-friendly curves with embedding degree $k \equiv 0 \pmod 6$ where $18 \nmid k$ can be constructed using the following polynomials:

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

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 $\rho = 5/4$.

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

The efficiency of pairings relies heavily on the computation of the line functions (as denoted by $\ell_{2R} = l_{R,R}/v_{2R}$ and $\ell_{R,P} = l_{R,P}/v_{R+P}$ in Algorithm 1). Recall that $l_{R,P}$ is notation for the line passing through R and P and that v_R is notation for the line passing through R and $-R$. In many other representations of elliptic curves, for example short Weierstrass form or Edwards form, the negation of a point is its reflection in the x-axis due to the fact that the identity group element is the point at

infinity. In particular, the line v_R has a particularly simple form, meaning that very often the field inversion can be avoided using demoninator elimination techniques.

For curves in Hessian form, the neutral group element is given by $(-1 : 0 : 1)$, and negation by

$$-(x, y) = (x/y, 1/y)$$

(in affine coordinates). Below, we give an optimized algorithm to compute the line functions for Hessian curves

$$\mathcal{H}/\mathbb{F}_q : aX^3 + Y^3 + Z^3 = 0$$

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 Arithmetic on Hessian curves

For the benefit of the reader, we include the most efficient (known) formulas for doubling and adding points on elliptic curves in Hessian form.

Let $R = (X_1 : Y_1 : Z_1) \in \mathcal{H}(K)$, and denote by $\mathbf{m}$ and $\mathbf{s}$ the field multiplication and field squaring in K respectively. The following formulas compute $2R = (X_3 : Y_3 : Z_3)$.

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

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

Let $P = (X_1 : Y_1 : 1)$ and $R = (X_2 : Y_2 : Z_2) \in \mathcal{H}(K)$. The following formulas compute $P + R = (X_1 : Y_1 : 1) + (X_2 : Y_2 : Z_2) = (X_3 : Y_3 : Z_3)$.

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

where $\eta = a \cdot X_1$ can be precomputed. The cost for point addition with the above formulas is $9\mathbf{m}$.

4.2 Even embedding degrees with pairings on $\mathbb{G}_1 \times \mathbb{G}_2$

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.

Recall that Miller's algorithm computes the line function

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

for pairings on $\mathbb{G}_1 \times \mathbb{G}_2$ we have that $P_1, P_2 \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$.

To avoid the field inversion we apply the denominator elimination technique of [24], described briefly below. It is shown in [24, Section 4] that if $\alpha \in \mathbb{F}_{q^k} \setminus \mathbb{F}_{q^{k/2}}$, then without loss of generality we may assume that a point $Q \in E(\mathbb{F}_{q^k})$ is of the form $Q = (c + d\alpha, c - d\alpha)$, where $c, d \in \mathbb{F}_{q^{k/2}}$. In particular, the line v_R passing through $R = (x, y) \in E(\mathbb{F}_q)$ and $-R = (x/y, 1/y)$ evaluated at Q is given by $2c(y-x) + x^2 - y^2 \in \mathbb{F}_{q^{k/2}}$. Hence in the final exponentiation step of Miller's algorithm the denominator will become 1, so can be ignored.

So, we replace the line funtion $\ell_{P_1,P_2}(Q)$ by just $l_{P_1,P_2}(Q)$. When $P_1 = P_2 = R$, this is the tangent line at R evaluated at Q, which is given by

$$l_{R,R}(Q) : aX_1^2 X_Q + TY_Q + S.$$

where $R = (X_1 : Y_1 : Z_1)$ and S and T are the values that were computed in the point doubling computation. Set $Q' = (X_{Q'} : Y_{Q'} : 1)$ and $Q = \varphi(Q')$, where φ is the twist isomorphism (1). Then we can write $l_{R,R}(Q)$ as

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

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

$$U = X_1^2; V = S \cdot Y_{Q'}; W = \eta \cdot U;$$

$$l_{R,R}(Q) = V + T + W\omega,$$

where $\eta = aX_{Q'}$ and can be precomputed.

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 $l_{2R}(Q)$ we have that $c_2 = 0$. In particular, the multiplication of $l_{2R}(Q)$ 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 $6\left(\frac{k}{3}\right)^2\mathbf{m} = \frac{2}{3}\mathbf{M}$ where $\mathbf{M}$ is the cost of field multiplication in $\mathbb{F}_{q^k}$. Putting together all of the above, the entire Miller doubling costs

$$(5\mathbf{m} + 2\mathbf{s}) + \left(1\mathbf{s} + \frac{2k}{3}\mathbf{m}\right) + \frac{2}{3}\mathbf{M} + 1\mathbf{S} = \left(5 + \frac{2k}{3}\right)\mathbf{m} + 3\mathbf{s} + \frac{2}{3}\mathbf{M} + 1\mathbf{S}.$$

When $P_1 = P = (X_1 : Y_1 : 1)$ and $P_2 = R = (X_2 : Y_2 : Z_2)$, the line $l_{P_1,P_2}(Q)$ is the line passing through P and R evaluated at Q. As above we write $Q = (\omega X'_Q : Y'_Q : 1)$ so that

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

where $E = Y_1 \cdot Z_2$, 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

$$E = Y_1 \cdot Z_2;\ L = (E - Y_2) \cdot X_1;\ M = (Y_{Q'} - Y_1) \cdot (A - X_2);$$

$$N = (E - Y_2) \cdot X_{Q'};\ l_{P,R}(Q) = L + M - N\omega.$$

Also exactly as for the the doubling line 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

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

4.3 Odd embedding degrees with pairings on $\mathbb{G}_1 \times \mathbb{G}_2$

Unfortunately the denominator elimination technique of [24] does not apply to this case; instead we extend ideas of [41] and [43]. Write $P_1 + P_2 = (X_3 : Y_3 : Z_3)$ and $Q = (X_Q : Y_Q : 1)$. Recall that Miller's algorithm starts from a chosen point $Q \in \mathbb{G}_2$; as above we use the twist isomorphism (1) to write $Q = (X_{Q'}\omega : Y_{Q'} : 1)$, where $X_{Q'}$ and $Y_{Q'} \in \mathbb{F}_{q^{k/3}}$. The line $v_{P_1+P_2}(Q)$ passing through $P_1 + P_2$ and $-(P_1 + P_2)$ evaluated at Q is hence given by

$$v_{P_1+P_2}(Q) : (Z_3 + Y_3)X_{Q'}\omega - (1 + Y'_Q)X_3.$$

To avoid inversion, we simply apply a trick of [41] and [43], namely observing that

$$\frac{1}{x - y} = \frac{x^2 + xy + y^2}{x^3 - y^3}.$$

Plugging in $x = (Z_3 + Y_3)X_{Q'}\omega$ and $y = (1 + Y_{Q'})X_3$ in $\frac{1}{v_{P_1+P_2}(Q)}$, we get that the denominator $x^3 - y^3$ is in $\mathbb{F}_{q^{k/3}}$ so will disappear in the final exponentiation, hence can be ignored. So we replace $\frac{1}{v_{P_1+P_2}(Q)}$ by the numerator

$$n_{P_1+P_2}(Q) = ((Z_3+Y_3)X_{Q'})^2\omega^2+(Z_3+Y_3)X_{Q'}(1+Y_{Q'})X_3\omega+((1+Y_{Q'})X_3)^2.$$

That is, we replace the line function $\ell_{P_1,P_2}(Q)$ by $n_{P_1+P_2}(Q)\cdot l_{P_1,P_2}(Q)$. As for even embedding degrees, the computation of $l_{P_1,P_2}(Q)$ costs $\mathbf{s} + \frac{2k}{3}\mathbf{m}$ if $P_1 = P_2$ and $2\mathbf{m} + \frac{2k}{3}\mathbf{m}$ if $P_1 \neq P_2$. The computation of $n_{P_1+P_2}(Q)$ can be computed in $\frac{2k}{3}\mathbf{m} + \frac{2}{9}\mathbf{S} + \frac{1}{9}\mathbf{M}$ via

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

and the multiplication of $n_{P_1+P_2}(Q)$ with $l_{P_1,P_2}(Q)$ costs $\frac{2}{3}\mathbf{M}$ as $l_{P_1+P_2}(Q)$ has no coefficient of ω^2.

Putting the above together, the full Miller doubling step costs

$$\begin{aligned}&(5\mathbf{m} + 2\mathbf{s}) + \left(\mathbf{s} + \frac{4k}{3}\mathbf{m} + \frac{2}{9}\mathbf{S} + \frac{7}{9}\mathbf{M}\right) + 1\mathbf{S} + 1\mathbf{M}\\ =&3\mathbf{s} + \left(5 + \frac{4k}{3}\right)\mathbf{m} + \frac{11}{9}\mathbf{S} + \frac{16}{9}\mathbf{M},\end{aligned}$$

and the full Miller addition step costs

$$9\mathbf{m} + \left(\left(2 + \frac{4k}{3}\right)\mathbf{m} + \frac{2}{9}\mathbf{S} + \frac{7}{9}\mathbf{M}\right) + 1\mathbf{M} = \left(11 + \frac{4k}{3}\right)\mathbf{m} + \frac{2}{9}\mathbf{S} + \frac{16}{9}\mathbf{M}.$$

4.4 Pairings on $\mathbb{G}_2 \times \mathbb{G}_1$

As with the previous two sections, we compute the line function

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

Also as above, we write $\mathbf{m}$ and $\mathbf{s}$ for the cost of multiplication and squaring in $\mathbb{F}_q$ respectively and we write $\mathbf{M}$ an $\mathbf{S}$ for the cost of multiplication and squaring in $\mathbb{F}_{q^k}$ respectively. For pairings on $\mathbb{G}_2 \times \mathbb{G}_1$ we have that $P_1, P_2 \in \mathbb{G}_2$ and $Q \in \mathbb{G}_1$.

As with odd embedding degrees for pairings on $\mathbb{G}_1 \times \mathbb{G}_2$, the denominator elimination technique of [24] does not apply and we again use the ideas of [41] and [43]. Write $P_1+P_2 = (X_3 : Y_3 : Z_3)$ and $Q = (X_Q : Y_Q : 1)$. Recall that Miller's algorithm starts from a chosen point $P \in \mathbb{G}_2$ from which

P_1 and P_2 are derived – in particular $P_1 + P_2$ is multiple of P, say mP. Using the twist isomorphism (1) we can assume that P is the image of a point $P' \in \mathcal{H}_\omega(\mathbb{F}_{q^{k/3}})$, so that $P_1+P_2 = mP$ is given by $\varphi(mP')$. It follows that there exist $X_3', Y_3', Z_3' \in \mathbb{F}_{q^{k/3}}$ such that $P_1 + P_2 = (X_3'\omega : Y_3' : Z_3')$. The line $v_{P_1+P_2}(Q)$ passing through $P_1 + P_2$ and $-(P_1 + P_2)$ evaluated at Q is hence given by

$$v_{P_1+P_2}(Q) : (Z_3' + Y_3')X_Q - (1 + Y_Q)X_3'\omega.$$

Analogously to the denominator elimination in the previous, we replace $\frac{1}{v_{P_1+P_2}(Q)}$ by the numerator

$$n_{P_1+P_2}(Q) = ((Z_3'+Y_3')X_Q)^2+(Z_3'+Y_3')X_Q(1+Y_Q)X_3'\omega+((1+Y_Q)X_3')^2\omega^2.$$

That is, we replace the line function $\ell_{P_1,P_2}(Q)$ by $n_{P_1,P_2}(Q)\cdot l_{P_1,P_2}(Q)$. When $P_1 = P_2 = R$, the line $l_{P_1,P_2}(Q)$ is the tangent line at R evaluated at Q, which is given by

$$l_{R,R}(Q) : aX_1'^2X_Q\omega^2 + TY_Q + S.$$

where S and T are the values that were computed in the point doubling computation. Observe also that when computing $2R$ we can instead compute $\varphi(2\varphi^{-1}(R))$ due to the simplicity of the twist isomorphism (1), so that our operation count for point doubling occurs in $\mathbb{F}_{q^{k/3}}$, not $\mathbb{F}_{q^k}$; also $S, T \in \mathbb{F}_{q^{k/3}}$. The numerator $n = n_{2R}(Q)$ can be computed in $\frac{2k}{3}\mathbf{m} + \frac{1}{9}\mathbf{M} + \frac{2}{9}\mathbf{S}$ via

$$\alpha = (1 + Y_Q) \cdot X_3'; \beta = X_Q \cdot (Z_3' + Y_3'); n = \beta^2 + (\alpha \cdot \beta)\omega + \alpha^2\omega^2$$

and the line $l = l_{R,R}(Q)$ can be computed in $\frac{2k}{3}\mathbf{m} + \frac{1}{9}\mathbf{S}$ via

$$\gamma = X_1'^2; \delta = \gamma \cdot \eta; l = S + T \cdot Y_Q + \delta\omega^2,$$

where $\eta = aX_Q$ and can be precomputed. The multiplication of n with l costs $\frac{2}{3}\mathbf{M}$ as l has no coefficient of ω, hence we compute $\ell_{R,R}(Q)$ in $\frac{1}{3}\mathbf{S} + \frac{7}{9}\mathbf{M} + \frac{4k}{3}\mathbf{m}$. Putting together the above, we get a total cost of

$$\left(\frac{5}{9}\mathbf{M} + \frac{2}{9}\mathbf{S}\right) + \left(\frac{1}{3}\mathbf{S} + \frac{7}{9}\mathbf{M} + \frac{4k}{3}\mathbf{m}\right) + \mathbf{M} + \mathbf{S} = \frac{7}{3}\mathbf{M} + \frac{14}{9}\mathbf{S} + \frac{4k}{3}\mathbf{m}$$

for the whole Miller doubling step.

When $P_1 = P = (X_1 : Y_1 : 1)$ and $P_2 = R = (X_2 : Y_2 : Z_2)$, the line $l_{P_1,P_2}(Q)$ is the line passing through P and R evaluated at Q. As

above we write $P = (\omega X_1' : Y_1' : 1)$ and $R = (\omega X_2' : Y_2' : Z_2')$ where $X_1', Y_1', X_2', Y_2', Z_2' \in \mathbb{F}_{q^{k/3}}$ so that

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

where $E = Y_1' \cdot Z_2'$, and where $A = A'\omega = X_1' Z_2' \omega$ is the value that was computed during the computation of $P + R$. We compute $\ell_{P,R}(Q)$ in $k\mathbf{m} + \mathbf{M} + \frac{2}{9}\mathbf{S}$ as follows:

$$\alpha = (1 + Y_Q) \cdot X_3'; \beta = X_Q \cdot (Z_3' + Y_3'); \epsilon = (E - Y_2') \cdot X_Q;$$

$$\zeta = (E - Y_2') \cdot X_1'; \theta = (Y_Q - Y_1') \cdot (A' - X_2');$$

$$n = \beta^2 + (\alpha \cdot \beta)\omega + \alpha^2\omega^2; l = -\epsilon + (\zeta + \theta)\omega; \ell_{R,R}(Q) = n \cdot l.$$

Observe further that when computing $P + R$ we can instead compute $\varphi(\varphi^{-1}(P) + \varphi^{-1}(R))$ due to the simplicity of the twist isomorphism (1), so that our operation count for point addition occurs in $\mathbb{F}_{q^{k/3}}$, not $\mathbb{F}_{q^k}$, that is, point addition costs $9(\frac{k}{3})^2\mathbf{m} = \mathbf{M}$.

Putting the together above, we see that the whole Miller addition step takes

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

5 Comparison

We would like to emphasize that:

- Most of the previous work on this topic concentrated on even embedding degrees.
- Most of the previous work on the optimization of operation counts for doubling and addition concentrated on pairings for $\mathbb{G}_1 \times \mathbb{G}_2$.
- To the best of our knowledge, previous research on Hessian curves for pairings did not utilize cubic twists and was only applicable for even embedding degrees.
- To the best of our knowledge, the only previous research on pairings for curves with odd embedding degrees used curves in Weierstrass form.

Table 1 shows the costs of the computation of the Miller doubling step (DBL) and Table 2 shows the costs of the Miller adding step (mADD) using mixed addition. The first column gives the conditions on the embedding degree k. Note that as this paper particularly concerns cubic twists,

we include only embedding degrees that are divisble by 3. The second column gives the curve models, where $\mathcal{J}$, $\mathcal{P}$, $\mathcal{E}$, $\mathcal{H}$ denote the Weierstrass model with Jacobian coordinates, the Weierstrass model with projective coordinates, the Edwards model and the Hessian model respectively. We denote the cost of field multiplication and field squaring over the base field $\mathbb{F}_q$ by $\mathbf{m}$ and $\mathbf{s}$ respectively and $\mathbf{M}$ and $\mathbf{S}$ denote the cost of field multiplication and field squaring over the extension field $\mathbb{F}_{q^k}$ respectively. The cost of multiplication by curve parameters is omitted from the table.

Table 1. Comparison of DBL for $\mathbb{G}_1 \times \mathbb{G}_2$ with different curve models and embedding degrees

k	Curve models	DBL
even, $6\|k$	$\mathcal{J}$, [27] [1]	$(k+1)\mathbf{m} + 11\mathbf{s} + \mathbf{M} + \mathbf{S}$
	$\mathcal{J}, a_4 = -3$, [1]	$(k+6)\mathbf{m} + 5\mathbf{s} + \mathbf{M} + \mathbf{S}$
	$\mathcal{J}, a_4 = 0$, [1]	$(k+3)\mathbf{m} + 8\mathbf{s} + \mathbf{M} + \mathbf{S}$
	$\mathcal{P}, (a_4, a_6) = (0, b^2)$, [15]	$(k+3)\mathbf{m} + 5\mathbf{s} + \mathbf{M} + \mathbf{S}$
	$\mathcal{E}$, [1]	$(k+6)\mathbf{m} + 5\mathbf{s} + \mathbf{M} + \mathbf{S}$
	$\mathcal{E}, j = 0$, [40]	$\left(\frac{4k}{3} + 4\right)\mathbf{m} + 7\mathbf{s} + \frac{1}{3}\mathbf{M} + \mathbf{S}$
	$\mathcal{H}$, [24]	$(k+3)\mathbf{m} + 6\mathbf{s} + \mathbf{M} + \mathbf{S}$
	$\mathcal{H}, j = 0$, this paper	$\left(\frac{2k}{3} + 5\right)\mathbf{m} + 3\mathbf{s} + \frac{2}{3}\mathbf{M} + \mathbf{S}$
odd, $3\|k$	$\mathcal{P}$, [16]	$(k+6)\mathbf{m} + 7\mathbf{s} + \mathbf{M} + \mathbf{S}$
	$\mathcal{H}, j = 0$, this paper	$\left(\frac{4k}{3} + 5\right)\mathbf{m} + 3\mathbf{s} + \frac{16}{9}\mathbf{M} + \frac{11}{9}\mathbf{S}$

Table 2. Comparison of mADD for $\mathbb{G}_1 \times \mathbb{G}_2$ with different curve models and embedding degrees

k	Curve models	mADD
even, $6\|k$	$\mathcal{J}$, [27] [1]	$(k+6)\mathbf{m} + 6\mathbf{s} + \mathbf{M}$
	$\mathcal{J}, a_4 = -3$, [1]	$(k+6)\mathbf{m} + 6\mathbf{s} + \mathbf{M}$
	$\mathcal{J}, a_4 = 0$, [1]	$(k+6)\mathbf{m} + 6\mathbf{s} + \mathbf{M}$
	$\mathcal{P}, (a_4, a_6) = (0, b^2)$, [15]	$(k+10)\mathbf{m} + 2\mathbf{s} + \mathbf{M}$
	$\mathcal{E}$, [1]	$(k+12)\mathbf{m} + \mathbf{M}$
	$\mathcal{E}, j = 0$, [40]	$\left(\frac{4k}{3} + 12\right)\mathbf{m} + \frac{1}{3}\mathbf{M}$
	$\mathcal{H}$, [24]	$(k+10)\mathbf{m} + \mathbf{M}$
	$\mathcal{H}, j = 0$, this paper	$\left(\frac{2k}{3} + 11\right)\mathbf{m} + \frac{2}{3}\mathbf{M}$
odd, $3\|k$	$\mathcal{P}$, [16]	$(k+13)\mathbf{m} + 3\mathbf{s} + \mathbf{M}$
	$\mathcal{H}, j = 0$, this paper	$\left(\frac{4k}{3} + 11\right)\mathbf{m} + \frac{16}{9}\mathbf{M} + \frac{2}{9}\mathbf{S}$

Table 3. Comparison of the costs of DBL and mADD for $\mathbb{G}_2 \times \mathbb{G}_1$ with different curve models and embedding degrees

k	Curve models	DBL	mADD
even, $6\|k$	$\mathcal{P}, j = 0$, [1]	$\frac{41}{36}\mathbf{M} + \frac{41}{36}\mathbf{S}$	$\frac{4}{3}\mathbf{M} + \frac{1}{18}\mathbf{S}$
	$\mathcal{H}, j = 0$, this paper	$\frac{4k}{3}\mathbf{m} + \frac{7}{3}\mathbf{M} + \frac{14}{9}\mathbf{S}$	$k\mathbf{m} + 3\mathbf{M} + \frac{2}{9}\mathbf{S}$
odd, $3\|k$	$\mathcal{H}, j = 0$, this paper	$\frac{4k}{3}\mathbf{m} + \frac{7}{3}\mathbf{M} + \frac{14}{9}\mathbf{S}$	$k\mathbf{m} + 3\mathbf{M} + \frac{2}{9}\mathbf{S}$

As is probably clear from Table 3, to our knowledge there is very little research on the optimization of curve arithmetic with regards the computation of the Miller function that utilizes twists. In particular we believe that this is the first paper that studies the use of cubic twists for the efficient computation of the Miller line function for a pairing on $\mathbb{G}_2 \times \mathbb{G}_1$ for curves with odd embedding degree. Curve arithmetic in this context is discussed to some extent in [1], but the formulas given in Section 6 for the line function correspond to pairings on $\mathbb{G}_1 \times \mathbb{G}_2$, so it appears that the authors chose to concentrate on pairings for $\mathbb{G}_1 \times \mathbb{G}_2$ for odd embedding degree.

Comparing Tables 1, 2, and 3, it is clear that one iteration of Miller's loop is faster for a $\mathbb{G}_1 \times \mathbb{G}_2$ pairing than for a $\mathbb{G}_2 \times \mathbb{G}_1$ pairing. However, there has been much research on minimizing the number of iterations of Miller's loop for both types of pairing which affects our analysis considerably.

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 are explictly addressing 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 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 T_e is T_2 where $T_2^3 = r - 1$, and when $d = 3$ the smallest value of T_e is $T_3 = r - 1$. For more details on the twisted ate pairing see [25].

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 x iterations of Miller's loop. For more details on the optimal ate pairing see [52].

To motivate the study of pairings of $\mathbb{G}_2 \times \mathbb{G}_1$, we compare the number of iterations for $\mathbb{G}_1 \times \mathbb{G}_2$ and $\mathbb{G}_2 \times \mathbb{G}_1$ when k is odd and divisible by 3. In

all the pairing-friendly families given in this paper, the polynomial $r(x)$ is the k^{th} or $2k^{\text{th}}$ cyclotomic polynomial, both of which have degree given by the Euler function $\varphi(k)$. That is, where the twisted ate pairing (on $\mathbb{G}_1 \times \mathbb{G}_2$) requires $\log(r-1)$ iterations of Miller's loop, the optimal ate pairing (on $\mathbb{G}_2 \times \mathbb{G}_1$) requires only $\log(r)/\phi(k)$ iterations of Miller's loop. So, if we wish to deduce from Tables 1, 2, and 3 which pairing should be used, as a rough guide we could divide the operation counts of Table 3 by $\phi(k)$. We give a comparison in the two aforementioned cases $k = 15$ and $k = 21$ with the NFS attacks and their variants in mind.

Example 1. Suppose that $\mathbf{s} = \mathbf{m}$ and set $k = 15$. The best operation count for one iteration of the Miller loop to compute the twisted ate pairing (on $\mathbb{G}_1 \times \mathbb{G}_2$) is given by projective coordinates at $478\mathbf{m}$ for doubling and $256\mathbf{m}$ for addition. The (only) operation count for one iteration of the Miller loop to compute the optimal ate pairing (on $\mathbb{G}_2 \times \mathbb{G}_1$) is given by Hessian coordinates at $895\mathbf{m}$ for doubling and $740\mathbf{m}$ for addition. However, if optimal parameters can be found, the computation of the twisted ate pairing will require $8 = \phi(15)$ times as many iterations of the Miller loop as the computation of the optimal ate, so we expect the optimal ate pairing with Hessian coordinates to outperform the twisted ate pairing with projective coordinates by a factor of between 3 and 4 (exactly how much better the performance is depends on the Hamming weights of x and T_3).

Example 2. Suppose that $\mathbf{s} = \mathbf{m}$ and set $k = 21$. The best choices for each pairing type are as in the previous example: for $\mathbb{G}_1 \times \mathbb{G}_2$ projective coordinates uses $916\mathbf{m}$ for doubling and $478\mathbf{m}$ for addition per iteration loop, and for $\mathbb{G}_2 \times \mathbb{G}_1$ Hessian coorindates uses $1743\mathbf{m}$ for doubling and $1442\mathbf{m}$ for addition per iteration loop. The twisted ate pairing ($\mathbb{G}_1 \times \mathbb{G}_2$) requires $12 = \phi(21)$ times as many iterations of the Miller loop as the optimal ate pairing ($\mathbb{G}_2 \times \mathbb{G}_1$), so for a rough comparison of the performance of the two pairings we divide the operation counts for optimal ate by 12. Given the above, we expect the optimal ate pairing with Hessian coorinates to be the most efficient choice for $k = 21$ as it should outperform the twisted ate pairing with projective Weierstrass coordinates by a factor of approximately 4.

Not included in Tables 1, 2, and 3 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 in 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 tables suggest that, for embedding degrees divisible by 6, the most efficient choices from the known implementations (including those in this paper) are degree 6 twists of Edwards curves for pairings on $\mathbb{G}_1 \times \mathbb{G}_2$ and degree 6 twists of Weierstrass curves with projective coordinates for pairings on $\mathbb{G}_2 \times \mathbb{G}_1$. For embedding degrees divisible by 3 but not 6, the most efficient choices from the known implementations (including those is in this paper) are degree 3 twists of Weierstrass curves with projective coordinates for pairings on $\mathbb{G}_1 \times \mathbb{G}_2$ and degree 3 twists of Hessian curves (that is, the constructions from this paper) for pairings on $\mathbb{G}_2 \times \mathbb{G}_1$. Furthermore, for embedding degrees 15 and 21 we expect the optimal ate pairing with Hessian curves to be the fastest choice; this outperforms the twisted ate pairing (the best choice for $\mathbb{G}_1 \times \mathbb{G}_2$) with projective coordinates of Weierstrass curves by a factor of between 3 and 4, as explained in the previous section.

Curves generated by the methods used in this paper (originally due to [21]) 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 $18 \nmid k$. As explained in earlier sections, 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 the low ρ-value of the families described above. The added security comes from the fact that 15 and 21 are slightly larger than 12 and 18 respectively (additionally, as both have only two prime factors the NFS techniques may be less effective). As both of these embedding degrees are odd, we suggest the use of Hessian curves with degree 3 twists in combination with the optimal ate pairing. This the most

efficient available implementation choice in this context – as explained in the discussion of the previous section.

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-values) that we suggest in this paper and the higher ρ-values (but small embedding degrees) suggested in [19] 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 [45]*, 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 [46]*, 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 [49]*, pages 89–101, 2009.
16. Craig Costello, Tanja Lange, and Michael Naehrig. Faster pairing computations on curves with high-degree twists. In *PKC 2010 [44]*, pages 224–242, 2010.
17. 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.
18. 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.
19. Georgios Fotiadis and Elisaveth Konstantinou. Tnfs resistant families of pairing-friendly elliptic curves. *Journal of Theoretical Computer Science*, page 224, 2018. (to appear).
20. 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.
21. 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/.
22. 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.
23. Craig Gentry and Alice Silverberg. Hierarchical ID-Based Cryptography. In *Asiacrypt 2002 [53]*, pages 548–566, 2002. http://www.cs.ucdavis.edu/~franklin/ecs228/pubs/extra_pubs/hibe.pdf.
24. Haihua Gu, Dawu Gu, and WenLu Xie. Efficient pairing computation on elliptic curves in hessian form. In *ICISC 2010*, pages 169–176, 2010.
25. 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.
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 - 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 [22]*, 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 [47]*, 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. Phong Q. Nguyen and David Pointcheval, editors. *Public Key Cryptography — PKC 2010, 13th International Conference on Practice and Theory in Public Key*

Cryptography, Paris, France, May 26–28, 2010. Proceedings, volume 6056 of *Lecture Notes in Computer Science*. Springer, 2010.

45. 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.
46. 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.
47. 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.
48. Amit Sahai and Brent Waters. Fuzzy Identity-Based Encryption. In *Eurocrypt 2005 [17]*, pages 457–473, 2005. http://eprint.iacr.org/2004/086/.
49. 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.
50. Joseph H. Silverman. *The Arithmetic of Elliptic Curves*, volume 106 of *Graduate Texts in Mathematics*. Springer, 2009.
51. Nigel P. Smart. The Hessian form of an elliptic curve. In *CHES 2001 [13]*, pages 118–125, 2001.
52. Frederik Vercauteren. Optimal pairings. In *IEEE Transactions on Information Theory 56(1)*, pages 455–461, 2010.
53. 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.