Fast Pairing-Friendly Hessian Curves

May 12, 2017

Abstract

This paper presents explicit formulas to generate pairing-friendly Hessian curves having embedding degree $k \equiv 3, 9, 15 \pmod{18}$ and $k \equiv 0 \pmod{6}$ where $18 \nmid k$. Curves generated by our constructions are guaranteed to have the twist of degree 3. We also explain a method to eliminate denominators for odd embedding degrees and provide explicit formulas for computing Tate pairing on twisted Hessian curves.

Keywords: twisted Hessian curves, pairing-friendly curves, Tate pairing, odd embedding degree, denominator elimination

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 [?] [?], to complicated protocols, such as identity-based encryption [?] [?] [?] [?]. Pairings also help to improve currently existing protocols, such as signature schemes, to have shortest possible signatures [?].

Curves that are suitable for pairings are called *pairing-friendly curves*, and these curves must satisfy specific properties. It is extremely rare that any randomly generated elliptic curves result in pairing-friendly curves. Therefore, pairing-friendly curves need to be generated in a special way. Examples of famous and commonly used pairing-friendly curves include Barreto-Naehrig curves [?] (BN-curves), Barreto-Lynn-Scott curves [?] (BLS-curves), and Kachisa-Schaefer-Scott curves [?] (KSS-curves). However, recent advances in the number field sieve techniques ([?]) to attack the discrete logarithm problem for elliptic curves have significantly reduced the security of pairing-based cryptography for elliptic curves with even embedding degree. In this paper, we outline compute the reduced Tate pairing on elliptic curves in Hessian form, which have a natural degree 3 twist, so that the most natural embedding degrees to consider are multiples of 3.

Performance of pairing-based cryptography relies on both elliptic-curve-point arithmetic and a computation of line functions. A pairing is a bilinear map from two elliptic curve groups $\mathbb{G}_1$ and $\mathbb{G}_2$ to a target group $\mathbb{G}_T$. Therefore, it is desirable to have efficient arithmetic in both $\mathbb{G}_1$ and $\mathbb{G}_2$. To improve the performance of the underlying point arithmetic, alternative curve models other than the conservative Weierstrass curves have been proposed, for example, Hessian curves [?] [?] and Edwards curves [?] [?].

Pairings based on Edwards curves along with examples of pairing-friendly Edwards curves were proposed by Arene, Lange, Naehrig and Ritzenthaler [?]. They found that the computation of line functions is much more complicated in the situation of Edwards curves than Weierstrass curves. In other words, even though Edwards curves allow faster point arithmetic, this gain is outweighed by the slower computation of line functions.

Bos, Costello and Naehrig [?] investigated a possibility of using faster curve models to compute operations prior to pairings, e.g., group exponentiations, then mapping to Weierstrass curves only for pairing computations. They found that among BN-12, BLS-12, and KSS-18 families of pairing-friendly curves either $\mathbb{G}_1$ or $\mathbb{G}_2$ has to be represented in Weierstrass curves. This means that either $\mathbb{G}_1$ or $\mathbb{G}_2$ has to use slower formulas to compute point operations. Moreover, this idea of using different curve models comes at a cost of at least one conversion between other curve models into Weierstrass curves.

The above raises the question whether it is possible to construct pairing-friendly curves that could be represented in other curve models in both $\mathbb{G}_1$ and $\mathbb{G}_2$ and preferably with fast point arithmetic and fast computation of line functions. Twisted Hessian curves with faster point arithmetic have recently been proposed by Bernstein, Lange, Chuengsatiansup and Kohel [?]. This leads to questions whether the newer point arithmetic would also lead to faster formulas for computing line functions, and whether these formulas would be applicable to pairing on Hessian curves.

The setup of this article is as follows: in ? we detail how to compute the reduced Tate pairing on an elliptic curve in Hessian form, making use of the existence of degree 3 twists. In ?, we observe that the efficiency of computing line functions on curves in Hessian form is similar to the Weierstrass case. In ?, we recall the methods given in [?] to generate pairing-friendly curves and propose a method to generate high-security pairing-friendly curves that can be represented in the Hessian form. In ?, we state explicit formulae for the computation of the reduced Tate pairing on Hessian curves based on the state-of-the-art point arithmetic formulas. In ?, we compare the efficiency of our algorithm with the Weierstrass case.

2 The reduced Tate pairing

Let E be an elliptic curve defined over a finite field $\mathbb{F}_p$ where p is a prime, and let r be the size of the largest prime-order subgroup of $E(\mathbb{F}_p)$. We define the *embedding degree* of E to be

$$\min\{k \in \mathbb{Z} : r | p^k - 1\}.$$

For an integer $0 < d \leq k$ and a point $T \in E(\mathbb{F}_{p^d})$, we define the function $f_{r,T}$ to be the unique normalized function such that

$$\mathrm{div}(f_{r,T}) = r[T] - [rT] - (r-1)[T_\infty],$$

where T_∞ denotes the point at infinity on E.

Definition 1. Let E, p, k, r, and for $P \in E(\mathbb{F}_p)$, the function $f_{r,P}$ be as above. We define the *reduced Tate pairing on E* to be

$$\begin{array}{rccc} e_r : & E(\mathbb{F}_p) \times E(\mathbb{F}_{p^k}) & \longrightarrow & \mathbb{F}_{p^k} \\ & (P, Q) & \mapsto & f_{r,P}(Q)^{\frac{q^k-1}{r}}. \end{array}$$

If E is ordinary, then this is a non-degenerate bilinear pairing, as shown in e.g. [**?**]. Furthermore, the following facts, due to Miller ([**?**]), allow us to explicitly compute this pairing. Let $0 < d \leq k$ be an integer, and let $T, T' \in E(\mathbb{F}_{q^d})$. We will write $L_{T,T'}$ for the projective line passing through T and T', and we will write $g_{T,T'}$ for the function

$$g_{T,T'} = \frac{L_{T,T'}}{L_{T+T',-(T+T')}}.$$

Lemma 1. With notation as above, for $P \in E(\mathbb{F}_p)$, we have that $f_{0,P} = f_{1,P} = 1$ and for $n \geq 2$, that $f_{n+1,P} = f_{n,P}g_{P,nP}$.

Proof. Straightforward computation on divisors. See [**?**] for more details. □

Corollary 1. With notation as above, for $P \in E(\mathbb{F}_p)$ and $Q \in E(\mathbb{F}_{p^k})$, we get that

$$e_r(P,Q) = \prod_{m=1}^{r-1} \frac{L_{P,mP}(Q)^{\frac{p^k-1}{r}}}{L_{(m+1)P,-(m+1)P}(Q)^{\frac{p^k-1}{r}}}.$$

Hence, we only need to give formulae for addition and point doubling to compute the reduced Tate pairing.

3 Pairing-friendly Hessian curves

Recall that in Section **??**, we define the reduced Tate pairing

$$e_r : E(\mathbb{F}_p) \times E(\mathbb{F}_{p^k}) \longrightarrow \mathbb{F}_{p^k}$$

on an elliptic curve $E/\mathbb{F}_p$ in the case that

- p is prime,
- r is the size of the largest prime order subgroup of $E(\mathbb{F}_p)$, and
- k is the smallest integer for which $r|p^k - 1$.

We will refer to an elliptic curve $E/\mathbb{F}_p$ where p, r, k are known and satisfy these properties as *pairing-friendly*. To construct our pairing, we search for pairing-friendly elliptic curves with a point of order 3 so that it can be written in Hessian form. There are many constructions of parametric families of pairing-friendly curves listed in [**?**]. We recall below those families for which the curves also have a point of order 3, as shown in [**?**, Section 5].

3.1 Construction 1: $k \equiv 3 \pmod{18}$

This construction follows Construction 6.6 in [**?**] under the first subcase where $k \equiv 3 \pmod 6$. Define

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

where $\Phi_{2k}(x)$ denotes the cyclotomic polynomial of degree $2k$. For infinitely many $x_0 \in \mathbb{Z}$, we can construct an elliptic curve $E/\mathbb{F}_{p(x_0)}$ such that the integers $r = r(x_0)$, $k = k(x_0)$, and $q = q(x_0)$ satisfy the properties listed in 3 (hence E is pairing-friendly). Furthermore, for such x_0, we have that $k(x_0) \equiv 3 \pmod{18}$.

3.2 Construction 2: $k \equiv 9, 15 \pmod{18}$

This construction follows Construction 6.6 in [**?**] under the second subcase where $k \equiv 3 \pmod 6$. Define

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

where $\Phi_{2k}(x)$ denotes the cyclotomic polynomial of degree $2k$. For infinitely many $x_0 \in \mathbb{Z}$, we can construct an elliptic curve $E/\mathbb{F}_{p(x_0)}$ such that the integers $r = r(x_0)$, $k = k(x_0)$, and $q = q(x_0)$ satisfy the properties listed in 3 (hence E is pairing-friendly). Furthermore, for such x_0, we have that $k(x_0) \equiv 9, 15 \pmod{18}$.

3.3 Construction 3: $k \equiv 0 \pmod 6$ and $18 \nmid k$

This construction follows the last case of Construction 6.6 in [?]. Define

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

where $\Phi_{2k}(x)$ denotes the cyclotomic polynomial of degree $2k$. For infinitely many $x_0 \in \mathbb{Z}$, we can construct an elliptic curve $E/\mathbb{F}_{p(x_0)}$ such that the integers $r = r(x_0)$, $k = k(x_0)$, and $q = q(x_0)$ satisfy the properties listed in 3 (hence E is pairing-friendly). Furthermore, for such x_0, we have that $k(x_0) \equiv 0 \pmod 6$, where $18 \nmid k(x_0)$.

3.4 Twists of degree 3

Let E and E' be elliptic curves over $\mathbb{F}_q$. We call E' a *twist* of E if E and E' are isomorphic over some field extension of $\mathbb{F}_q$. More precisely, E' is a *degree-d twist* of E if they are isomorphic over a degree d extension and not over any smaller field. Suppose that

$$E : x^3 + y^3 + z^3 = 0$$

is an elliptic curve defined over a field $\mathbb{F}_q$. Then if $q \equiv 1 \bmod 3$, the group $\mathbb{F}_q^*/(\mathbb{F}_q^*)^3$ consists of 3 elements, and hence there exist 2 degree 3 twists of E defined by

$$E_a : aX^3 + Y^3 + Z^3 = 0,$$

where $1 \neq a \in \mathbb{F}_q^*/(\mathbb{F}_q^*)^3$. In particular, if $3|k$, where k is the embedding degree of $E/\mathbb{F}_q$, and $\#E(\mathbb{F}_q) \pmod 3 \neq 0$, then by [?, ?, ?, Theorem 3], we have that

$$E(\mathbb{F}_{q^k}) \cong \bigoplus_{a \in \mathbb{F}_q^*/(\mathbb{F}_q^*)^3} E_a(\mathbb{F}_{q^{k/3}}).$$

Hence, we can speed up the evaluation of the line function at a point $Q \in E(\mathbb{F}_{q^k})$ in the pairing computation by instead evaluating at a point $Q' \in E_a(\mathbb{F}_{q^{k/3}})$.

is this really necessary? To express twists of curves in the Hessian form, those twists must also contain points of order 3. To check whether the twisted curve E' of E contains points of order 3 or not, we use the formulas in [?] which state the number of points on twisted curves. The formulas for calculating the number of points on twisted curves always come in pairs. For example, the formulas for $d = 3$ stated in [?] are as follows:

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

To determine the right twist, we use the fact that $\#E'(\mathbb{F}_q)$ must also be divisible by r which is the subgroup of E. There is exactly one of those two possible twists that satisfies this condition.

The only possible degree of twist for curves generated by Construction 1 and Construction 2 is of degree 3. Twists of curves generated by these constructions also contain points of order 3. For Construction 3, even though it allows many possible degrees of twist, only twist of degree 3 that contain points of order 3. In summary, for all pairing-friendly Hessian curve constructions presented in previous sections, the only possible degree of twist is of degree $d = 3$.

4 Computation of line functions

We now move to the explicit computation of the line functions $\ell_1 = L_{P,Q}$ and $\ell_2 = L_{P,-P}$ given in the formula for the reduced Tate pairing. Recall from **??** that the denominator of the reduced Tate pairing was given by

$$L_{nP,-nP}(Q)^{\frac{q^k-1}{r}},$$

where $P \in E(\mathbb{F}_q)$ and $Q \in E(\mathbb{F}_{q^k})$. Recall also from **??** that, possibly replacing E by a twist, we may assume without loss of generality that $Q \in E(\mathbb{F}_{q^{k/3}})$. Write $nP = (x_{nP}, y_{nP}, z_{nP})$ and $Q = (x_Q, y_Q, z_Q)$. Then applying the group law we compute the equation of $L_{nP,-nP}$ to be

$$Y + Z + (y_{nP} + z_{nP})X = 0,$$

which evaluated at Q gives an element b of $\mathbb{F}_{q^{k/3}}$. Now observe that

$$(q^k - 1) = (q^{k/3} - 1)(\sum_{i=0}^{2k/3} q^i),$$

and by minimality of the embedding degree that $r \not| q^{k/3} - 1$, hence

$$L_{nP,-nP}(Q)^{\frac{q^k-1}{r}} = b^{\frac{q^k-1}{r}} = (b^{q^{k/3}-1})^{\frac{\sum_{i=0}^{2k/3} q^i}{r}} = 1.$$

So this does not contribute to the reduced Tate pairing.

4.1 Denominator elimination

We follow a similar technique as in [?] and [?] by rewriting an inversion into a fraction having denominator lies in a subfield. Define a finite field extension $\mathbb{F}_{q^m}$ of a finite field $\mathbb{F}_q$. Observe that

$$\frac{1}{a} = \frac{a^q \cdot a^{q^2}}{a \cdot a^q \cdot a^{q^2}}$$

where $a \cdot a^q \cdot a^{q^2} = N(a)$, i.e., a *norm* of a which lies in $\mathbb{F}_q$. Because $N(a) \in \mathbb{F}_q$ which is a subfield of $\mathbb{F}_{q^m}$, the final exponentiation forces $N(a)$ to become 1. This means that we can ignore the computation of the denominator $N(a)$. Therefore, instead of computing $\frac{1}{a}$, we compute $a^q \cdot a^{q^2}$ where a^q and a^{q^2} can be easily computed using Frobenius.

4.2 Explicit formulas

Recall the twsited Hessian curve equation in projective coordiates

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

This subsection shows formulas to compute point doubling, point addition, and line functions associated to these operations. These formulas work under an assumption that the curve parameter $d = 0$.

Symbols $\mathbf{m}, \mathbf{s}, \mathbf{A}, \mathbf{m}_a, \mathbf{m}_2$ denote field multiplication, field squaring, field addition, field multiplication by curve constant a and field multiplication by 2 over $\mathbb{F}_p$, whereas $\mathbf{m}_k$ and $\mathbf{A}_k$ denote field multiplication and field addition over $\mathbb{F}_{p^k}$, and $\mathbf{m}_e$ and $\mathbf{A}_e$ denote field multiplication and field addition over $\mathbb{F}_{p^e}$ where $e = k/d$.

4.2.1 Doubling.

Given a point $P = (X_1, Y_1, Z_1)$ on $E/\mathbb{F}_p$ and a point $Q = (X_Q, Y_Q, 1)$ on $E'/\mathbb{F}_{p^{k/d}}$, the following formulas compute the doubling of point $P = 2P = (X_3, Y_3, Z_3)$ and the line function l.

$$\begin{aligned}
&U = Y_1^2; \qquad A = Y_1 \cdot U; \qquad V = Z_1^2; \qquad B = Z_1 \cdot V;\\
&X_3 = X_1 \cdot (A - B); \quad Y_3 = -Z_1 \cdot (2A + B); \quad Z_3 = Y_1 \cdot (A + 2B);\\
&l_1 = -X_Q \cdot (A + B) + X_1 \cdot (Y_Q \cdot U + V);\\
&l_2 = X_3 \cdot (Y_Q + 1) - X_Q \cdot (Y_3 + Z_3);\\
&\ l = l_1 \cdot Frobenius(l_2, 7) \cdot Frobenius(l_2, 14);
\end{aligned}$$

The total number of operations is $5\mathbf{m} + 2\mathbf{s} + 3\mathbf{A} + 2\mathbf{m}_2 +$ $(2\mathbf{m}_k + 3\mathbf{m}_e + 2\mathbf{A}_k + 2\mathbf{A}_e + 2\mathbf{A}) +$ $(2\mathbf{m}_k + 2Frobenius)$.

4.2.2 Addition.

Given points $P = (X_1, Y_1, 1)$ and $R = (X_2, Y_2, Z_2)$ on $E/\mathbb{F}_p$ and a point $Q = (X_Q, Y_Q, 1)$ on $E'/\mathbb{F}_{p^{k/d}}$, the following formulas compute the addition of points $P + R = (X_1, Y_1, Z_1) + (X_2, Y_2, Z_2) = (X_3, Y_3, Z_3)$ and the line function l.

$$
\begin{aligned}
&A = X_1 \cdot Z_2; \qquad C = Y_1 \cdot X_2; \qquad D = Y_1 \cdot Y_2; \qquad F = aX_1 \cdot X_2;\\
&G = (D + Z_2) \cdot (A - C); \quad H = (D - Z_2) \cdot (A + C);\\
&J = (D + F) \cdot (A - Y_2); \quad K = (D - F) \cdot (A + Y_2);\\
&X_3 = G - H; \qquad Y_3 = K - J;\\
&Z_3 = J + K - G - H - 2(B - F) \cdot (C + E);\\
&l_1 = (Y_1 \cdot Z_2 - Y_2) \cdot (X_1 - X_Q) + (Y_Q - Y_1) \cdot (X_1 \cdot Z_2 - X_2);\\
&l_2 = X_3 \cdot (Y_Q + 1) - X_Q \cdot (Y_3 + Z_3);\\
&l = l_1 \cdot Frobenius(l_2, 7) \cdot Frobenius(l_2, 14);
\end{aligned}
$$

The total number of operations is $9\mathbf{m} + 16\mathbf{A} + 1\mathbf{m}_a + 1\mathbf{m}_2 +$ $(2\mathbf{m}_k + 2\mathbf{m}_e + 2\mathbf{m} + 3\mathbf{A}_k + 2\mathbf{A}_e + 3\mathbf{A}) +$ $(2\mathbf{m}_k + 2Frobenius)$.

5 Comparison

Different curve models

Table 1: Cost of pairing computation

Curve models	DBL	mADD
$\mathcal{J}$, [?] [?]	$1\mathbf{M} + 1\mathbf{S} + k\mathbf{m} + 1\mathbf{m} + 11\mathbf{s} + 1\mathbf{m}_a$	$1\mathbf{M} + k\mathbf{m} + 6\mathbf{m} + 6\mathbf{s}$
$\mathcal{J}, a = -3$, [?]	$1\mathbf{M} + 1\mathbf{S} + k\mathbf{m} + 6\mathbf{m} + 5\mathbf{s}$	$1\mathbf{M} + k\mathbf{m} + 6\mathbf{m} + 6\mathbf{s}$
$\mathcal{J}, a = 0$, [?]	$1\mathbf{M} + 1\mathbf{S} + k\mathbf{m} + 3\mathbf{m} + 8\mathbf{s}$	$1\mathbf{M} + k\mathbf{m} + 6\mathbf{m} + 6\mathbf{s}$
$\mathcal{P}, a = 0, b = b^2$, [?]	$1\mathbf{M} + 1\mathbf{S} + k\mathbf{m} + 3\mathbf{m} + 5\mathbf{s}$	$1\mathbf{M} + k\mathbf{m} + 10\mathbf{m} + 2\mathbf{s} + 1\mathbf{m}_b$
$\mathcal{E}$, [?]	$1\mathbf{M} + 1\mathbf{S} + k\mathbf{m} + 6\mathbf{m} + 5\mathbf{s}$	$1\mathbf{M} + k\mathbf{m} + 12\mathbf{m}$
$\mathcal{H}$, [?]	$1\mathbf{M} + 1\mathbf{S} + k\mathbf{m} + 3\mathbf{m} + 6\mathbf{s}$	$1\mathbf{M} + k\mathbf{m} + 10\mathbf{m}$
$\mathcal{H}$, this paper	$1\mathbf{M} + 1\mathbf{S} + k\mathbf{m} + 3\mathbf{m} + 6\mathbf{s}$	$1\mathbf{M} + k\mathbf{m} + 10\mathbf{m}$

Different embedding degrees

6 Discussion and future work

Although pairing on Hessian curves has already been considered, for example, by Gu, Gu, and Xie in [?] and by Li and Zhang in [?], this work appears to be the first to explain methods to generate pairing-friendly Hessian curves. Moreover, we applied state-of-the-art point arithmetic formulas from [?] and present faster formulas for line function and pairing computations.

Our initial analysis shows that the speed of line function computations on Hessian curves is comparable to those on Weierstrass curves. However, Hessian curves have an advantage of faster point arithmetic, which may lead to faster overall performance of pairing computations. We expect pairing on Hessian curves to be faster than on Weierstrass curves and consider the optimized implementation as future work.

7 Conclusions

This paper presents concrete methods to generate pairing-friendly Hessian curves. Curves generated by these methods have twists of degree 3 and have embedding degree $k \equiv 3 \pmod{18}$, $k \equiv 9, 15 \pmod{18}$ or $k \equiv 0 \pmod{6}$ where $18 \nmid k$. We describe techniques to eliminate intermediate denominators by using Frobenius, which results in removing all division operations and hence leads to faster overall computation. We also provide explicit formulas to compute line functions in pairing computations. Our constructions of pairing-friendly Hessian curves together with presented techniques and formulas offer fast point arithmetic and fast line function computations competitive with pairing-friendly Weierstrass and Edwards curves.