ARAB JOURNAL OF
MATHEMATICAL SCIENCES

Arab J Math Sci 25 (1) (2019) 29–42

Parallelizing pairings on Hessian elliptic curves☆

EMMANUEL FOUOTSA

Higher Teacher Training College, The University of Bamenda, P.O. Box 39, Bambili, Cameroon

Received 12 January 2018; accepted 5 June 2018
Available online 15 June 2018

Abstract. This paper considers the computation of the Ate pairing on the Hessian model of elliptic curves. Due to the many important properties making the model attractive in cryptography, we compute for the first time the Ate pairing on this model and show how both the Tate and the Ate pairings can be parallelized on this curve. We wrote codes in the Sage software to ensure the correctness of formulas in this work.

Keywords: Hessian curves; Tate and ate pairings; Parallel computation

1. INTRODUCTION

In cryptography, pairings are bilinear maps defined on the group of points of an elliptic of hyperelliptic curve. These mathematical objects were first used to solve the discrete logarithm problem [16], but they now constitute an interesting research topic in cryptography as they allow to construct many other cryptographic protocols [6,7,14,22]. Several papers exist on the efficient computation and implementation of pairings on various models of elliptic curves [1–3,5,9–11,13,29] leading sometimes to other type of interesting pairings such as the Ate pairing (see [19,20,28]). The most common used model of elliptic curves is the so called Weierstrass model in its short equation $y^2 = x^3 + ax + b$ defined over a finite field $\mathbb{F}_q$. Several other models exist in the literature such as the Hessian model,

☆ This work was supported by the Simons Foundation through Pole of Research in Mathematics with applications to Information Security, Sub-Saharan Africa and LIRIMA-FAST project.
E-mail address: emmanuelfouotsa@yahoo.fr.
Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

https://doi.org/10.1016/j.ajmsc.2018.06.001

the Edward model, the Jacobi model. These curves are almost all birationally equivalent to the Weierstrass model but depending on the properties of each curve such as arithmetic of points, a careful choice of the model may be necessary. For example, an elliptic curve with complete addition formulas and/or unified addition formulas ensure protection against exceptional procedure attacks [21] and side-channel attacks respectively on protocols based on the curves used. Also, addition formulas that can be parallelized may bring advantage in terms of efficiency of the computations. The Hessian model of elliptic curves [26] has been proven to have unified addition formulas [23] which can be computed in a parallel way [26]. Also this model presents a nice geometric interpretation of the group law that allows to obtain competitive costs in pairing's computation with respect to well known models of curves such as the Weierstrass and the Edward model [18]. Also, some standard curves from IEEE, SECG can be transformed to Hessian curves as pointed out by Smart [26]. This work continues to emphasize on these nice properties of the Hessian form of elliptic curve. In particular, we show in this paper that one can efficiently parallelize the computation of both the Tate and the Ate pairings on this curve. More precisely, our contribution is as follows:

1. Considering the fact that in the computation of the Ate pairing, addition of points is performed in an extension field $\mathbb{F}_{q^k}$ of the base field $\mathbb{F}_q$ where the elliptic curve is defined, we used a specific representation of points to efficiently compute this pairing on the Hessian curves, by first rewriting adequate addition formulas. This is similar to the twist technique that enables to avoid some inversions and to perform some computations rather in a subfield of $\mathbb{F}_{q^k}$. We also provide codes written with the software Sage [27] to ensure the correctness of all the formulas in this work (http://www.camercrypt.org/sagehessian.txt)
2. Using the aforementioned adapted addition formulas, we succeed to efficiently compute the Ate pairing on Hessian curves. Also we consider the previous result [18] (and the only existing one to our knowledge) on the computation of the Tate pairing on Hessian curves and we succeed to show how both the Tate and the Ate pairings can be efficiently computed in a parallel manner. The results are summarized in Table 7.

The rest of this work is organized as follows: In Section 2 we define pairings on elliptic curves and the Miller algorithm for its efficient computation. Section 3 presents the Hessian model of elliptic curves with some important properties (addition of points, formulas for the computation of the Tate pairing). In Section 4, we give details for the first time in the literature for the computation of the Ate pairing on Hessian curves. We then show how to parallelize the computation of both the Tate and Ate pairings in Section 5. Section 6 concludes our work.

Notations. In this work the following notations m_l, s_l and m_c represent the cost of multiplication, squaring and multiplication with a constant in the field $\mathbb{F}_{q^l}$ respectively.

2. Preliminaries on elliptic curves and pairings

This section reviews fundamental elements on pairings over elliptic curves useful to understand the work. We define the two pairings concerned in this work namely the Tate and the Ate pairings.

Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$. The identity element of the additive group law defined on the set of rational points of E is denoted $\mathcal{O}$. We denote r a large prime

Algorithm 1: Miller's Algorithm

Input : $P \in E(\mathbb{F}_q)[r]$, $Q \in E(\mathbb{F}_{q^k})[r]$, $r = (1, r_{n-2}, ...r_1, r_0)_2$.

Output: The reduced Tate pairing of P and Q : $f_{r,P}(Q)^{\frac{q^k-1}{r}}$

1: Set $f \leftarrow 1$ and $R \leftarrow P$
2: **For** $i = n-2$ **down to** 0 **do**
3: $f \leftarrow f^2 \cdot h_{R,R}(Q)$
4: $R \leftarrow 2R$
5: **if** $r_i = 1$ **then**
6: $f \leftarrow f \cdot h_{R,P}(Q)$
7: $R \leftarrow R + P$
8: **end if**
9: **end for**
10: **return** $f^{\frac{q^k-1}{r}}$

Fig. 1. The Miller algorithm for the computation of the reduced Tate pairing.

divisor of the group order $\sharp E(\mathbb{F}_q)$ and k the smallest integer such that r divides $q^k - 1$. The set $E\left(\overline{\mathbb{F}_q}\right)[r] = \{P \in E\left(\overline{\mathbb{F}_q}\right) : [r]P = \mathcal{O}\}$ is the set of r-torsion points with coordinates in an algebraic closure $\overline{\mathbb{F}_q}$ of $\mathbb{F}_q$, where $[\,] : P \longmapsto [m]P$ is the endomorphism defined on $E(\overline{\mathbb{F}_q})$ which maps P to itself m times.

The Tate pairing. Consider a point $P \in E(\mathbb{F}_q)[r]$ and the principal divisor $D = r(P) - r(\mathcal{O})$ such that Div $(f_{r,P}) = D$ for a certain function $f_{r,P}$. Let $Q \in E(\mathbb{F}_{q^k})[r]$ and μ_r be the group of rth roots of unity in $\mathbb{F}_{q^k}^*$.

Definition 1 (*The Tate Pairing*)**.** [25] The reduced Tate pairing e_r is a bilinear and non degenerate map defined by

$$\begin{array}{rccc} e_r : & E(\mathbb{F}_q)[r] \times E(\mathbb{F}_{q^k})[r] & \to & \mu_r \\ & (P, Q) & \mapsto & f_{r,P}(Q)^{\frac{q^k-1}{r}} \end{array}$$

The value $f_{r,P}(Q)$ can be determined efficiently using Miller's algorithm [24] (see Fig. 1) which uses the rational function $h_{R,S}$ such that

$$\text{Div}(h_{R,S}) = (R) + (S) - (S + R) - (\mathcal{O})$$

with R and S two arbitrary points on the elliptic curve. In the case of elliptic curves in Weierstrass form, $h_{R,S} = \frac{\ell_{R,S}}{v_{R+S}}$ where $\ell_{R,S}$ is the straight line defining $R + S$ and v_{R+S} is the corresponding vertical line passing through $R + S$. So simple, this is slightly the same thing in the case of Hessian curves but this function is much more complicated in other cases (degree-two curves like conics in the case of Jacobi or Edwards curves).

More information on pairings can be found in [17] and [12].

The Ate pairing. We briefly present in this section the Ate pairing and we refer to [20] for more details. Let t be the trace of the Frobenius endomorphism defined by

$$\begin{array}{rccc} \pi_q : & E\left(\overline{\mathbb{F}_q}\right) & \to & E\left(\overline{\mathbb{F}_q}\right) \\ & (x, y) & \mapsto & \left(x^q, y^q\right) \end{array}$$

The Frobenius endomorphism π_q has exactly two eigenvalues 1 and q. This enables to consider points $P \in \mathbb{G}_1 = E\left(\overline{\mathbb{F}_q}\right)[r] \cap \text{Ker}(\pi_q - [1]) = E(\mathbb{F}_q)[r]$ and $Q \in \mathbb{G}_2 = E\left(\overline{\mathbb{F}_q}\right)[r] \cap \text{Ker}(\pi_q - [q])$. The Ate pairing is defined as follows:

Definition 2 (*The Ate Pairing*)**.** [20] The reduced Ate pairing is the map:

$$\begin{array}{rccc} e_A: & \mathbb{G}_2 \times \mathbb{G}_1 & \to & \mu_r \\ & (Q, P) & \mapsto & f_{T,Q}(P)^{\frac{q^k-1}{r}} \end{array}$$

where $T = t - 1$.

Theorem 1 makes sense to Definition 2 as it states that the Ate pairing is a power of a Tate pairing and therefore is a pairing. A complete proof can be found in [20].

Theorem 1 (*[20]*)**.** *Let $N = gcd(T^k - 1, q^k - 1)$ and $T^k - 1 = LN$. We have*

- *$e_A(Q, P)^{rc} = (f_{r,Q}(P)^{(q^k-1)/r})^{LN}$ where $c = \sum_{i=0}^{k-1} T^{k-1-i} q^i \equiv kq^{k-1} \bmod r$.*
- *for $r \nmid L$, the Ate pairing e_A is non-degenerate.*

Remark 1. During the execution of the Miller algorithm the point addition is performed in an extension field of $\mathbb{F}_q$ in Ate pairing computation whereas it is performed in the base field $\mathbb{F}_q$ in the case of the Tate pairing. Therefore each step of the Ate pairing is more expensive than the Tate pairing. However the Miller loop length in the case of the Ate pairing is $\log_2 T$ which is less (generally the half) than $\log_2 r$, the loop length for the Tate pairing.

3. The Hessian model of elliptic curves

In this section we recall the definition of an Hessian curve together with addition formulas. We also recall from [18] the Miller function on this curve which is essential for the computation of various pairings.

3.1. Definition of the Hessian model of elliptic curves

Definition 3. The Hessian model of elliptic curve over a finite field $\mathbb{F}_q$ (or any field) is defined in [26] by a homogeneous equation of the form

$$X^3 + Y^3 + Z^3 = 3DXYZ$$

or the affine equation

$$x^3 + y^3 + 1 = Dxy$$

where $D \in \mathbb{F}_q^\star$ and $D^3 \neq 1$.

The map between the Hessian curves $\mathcal{H}_D$ and Weierstrass curves is given in Proposition 1.

Proposition 1. *The Hessian curve $\mathcal{H}_D : x^3 + y^3 + 1 = Dxy$ over $\mathbb{F}_q$ is birationally equivalent to the Weierstrass curve $W_D : v^3 = u^3 - 27D(D^3 + 8)u + 54(D^6 - 20D^3 - 8)$ under the*

maps

$$\begin{array}{rccc} \varphi: & \mathcal{H}_D & \longrightarrow & W_D \\ & (x, y) & \longmapsto & (\eta(x + 9D^2), -1 + \eta(3D^3 - Dx - 12)); \end{array}$$

$$\begin{array}{rccc} \varphi^{-1}: & W_D & \longrightarrow & \mathcal{H}_D \\ & (u, v) & \longmapsto & (-9D^2 + \xi u, 3\xi(v - 1)) \end{array}$$

where

$$\eta = \frac{6(D^3 - 1)(y + 9D^3 - 3Dx - 36)}{(x + 9D^2)^2(3D^3 - Dx - 12)^3} \text{ and } \xi = \frac{12(D^3 - 1)}{Du + v + 1}$$

3.2. Addition formulas on Hessian curves

The geometric interpretation of the group law on Hessian curves is similar to the one on Weierstrass curves with a little difference. Given two points P and Q on $\mathcal{H}_D$, the sum $P + Q$ is obtained as the reflection with respect to the line $y = x$ of the third intersection point of the line (PQ) with $\mathcal{H}_D$ [18]. The identity point is denoted $\mathcal{O} := (-1 : 1 : 0)$. Explicitly, given a point $P_1 = (X_1 : Y_1 : Z_1)$ on $\mathcal{H}_D$, its opposite is given as $-P_1 = (Y_1 : X_1 : Z_1)$.

Point addition

Assume $P_1 = (X_1 : Y_1 : Z_1)$, $P_2 = (X_2 : Y_2 : Z_2)$ and $P_1 + P_2 = P_3 = (X_3 : Y_3 : Z_3)$, then

$$\begin{cases} X_3 = Y_1^2 X_2 Z_2 - X_1 Z_1 Y_2^2 \\ Y_3 = X_1^2 Y_2 Z_2 - Y_1 Z_1 X_2^2 \\ Z_3 = Z_1^2 X_2 Y_2 - X_1 Y_1 Z_2^2 \end{cases}$$

Point doubling

Assume $P_1 = (X_1 : Y_1 : Z_1)$, $2P_2 = P_3 = (X_3 : Y_3 : Z_3)$, then

$$\begin{cases} X_3 &= Y_1(X_1^3 - Z_1^3) \\ Y_3 &= X_1(Z_1^3 - Y_1^3) \\ Z_3 &= Z_1(Y_1^3 - X_1^3) \end{cases}$$

For a complete understanding of addition formulas on Hessian curves one can read [8,23,26] and [15].

3.3. Miller's function on Hessian curves

As we earlier mentioned, the computation of pairings with the Miller algorithm requires a function $h_{R,S}$ with divisor $div(h_{R,S}) = (R)+(S)-(R+S)-(\mathcal{O})$ where R and S are two points on the elliptic curve. This function is in fact the description of the geometric interpretation of the group law on the elliptic curve. In the case of Hessian curves, such function is given by Theorem 2.

Theorem 2 (*[18]*). *Let $\mathcal{H}_D$ be the Hessian curve over $\mathbb{F}_q$; $P_1 = (X_1 : Y_1 : Z_1)$ and $P_2 = (X_2 : Y_2 : Z_2)$ two points on $\mathcal{H}_D(\mathbb{F}_q)$. Define $P_3 = (X_3 : Y_3 : Z_3) = P_1 + P_2$. Then the function h_{P_1,P_2} is given by*

$$h_{P_1,P_2}(X, Y, Z) = \frac{c_X X + c_Y Y + c_Z Z}{(Y_3 Z_3 - Z_3 X_3)(X + Y) + (X_3^2 - Y_3^2)Z}, \tag{1}$$

where the coefficients are given as follows:

a. *If* $P_1 \neq P_2$ *then*

$$c_X = Y_1Z_2 - Z_1Y_2$$
$$c_Y = Z_1X_2 - X_1Z_2$$
$$c_Z = X_1Y_2 - Y_1X_2$$

b. *If* $P_1 = P_2$, *then* $c_X = 3X_1^2 - 3DY_1Z_1$, $c_Y = 3Y_1^2 - 3DX_1Z_1$ *and* $c_Z = 3Z_1^2 - 3DX_1Y_1$.

The function in Theorem 2 is given for the first time by Gu et *al.* [18] to compute the Tate pairing on Hessian curves. This is to our knowledge the only result in the literature on pairing computation on Hessian curve. In the next section we extend their work to the computation of the Ate pairing.

4. Computation of the Ate pairing on $\mathcal{H}_D$

In this section we begin by rewriting the point addition, the point doubling formulas and Miller's function suitable for the computation of the Ate pairing.

4.1. Explicit addition formulas for the Ate pairing on Hessian curve

As earlier mentioned, we observe that the addition and doubling of points are performed in $\mathbb{F}_{q^k}$. But thanks to the approach explained in [18] if k is even then points with coordinates in $\mathbb{F}_{q^k}$ can take the form $(A + B\alpha : A - B\alpha : C)$ with $\alpha \in \mathbb{F}_{q^k}$ and $A, B, C \in \mathbb{F}_{q^{\frac{k}{2}}}$. This form will particularly allow to make some computation rather in a subfield $\mathbb{F}_{q^{k/2}}$of $\mathbb{F}_{q^k}$, as $\mathbb{F}_{q^k}$ is now viewed as a $\mathbb{F}_{q^{k/2}}$-vector space. Thus, to compute the Ate pairing, it is necessary to rewrite addition and doubling of points, with the difference that the points have the form $(A_i + B_i\alpha : A_i - B_i\alpha : C_i)$ where $A_i, B_i, C_i \in \mathbb{F}_{q^{\frac{k}{2}}}$, $i = 1, 2, 3$. The sage code for the correctness of the formulas is available at http://www.camercrypt.org/sagehessian.txt.

Point addition

Let $P_1 = (A_1 + B_1\alpha : A_1 - B_1\alpha : C_1)$ and $P_2 = (A_2 + B_2\alpha : A_2 - B_2\alpha : C_2)$ be two points on $\mathcal{H}_D(\mathbb{F}_{q^k})[r]$ with $A_i, B_i, C_i \in \mathbb{F}_{q^{\frac{k}{2}}}$, $i = 1, 2$. Denote $P_1 + P_2 = P_3 = (X_3 : Y_3 : Z_3)$ then the coordinates of P_3 are computed as follows:

$$\begin{aligned} X_3 &= (A_1 - B_1\alpha)^2(A_2 + B_2\alpha)C_2 - (A_1 + B_1\alpha)(A_2 + B_2\alpha)^2C_1 \\ &= (A_1C_2 - A_2C_1)(A_1A_2 - 2B_1B_2\alpha^2) + (A_2B_1^2C_2 - A_1B_2^2C_1)\alpha^2 \\ &+ [(B_1C_2 - B_2C_1)(B_1B_2\alpha^2 - 2A_1A_2) + (A_1^2B_2C_2 - A_2^2B_1C_1)]\alpha \end{aligned}$$

$$\begin{aligned} Y_3 &= (A_1 - B_1\alpha)^2(A_2 + B_2\alpha)C_2 - (A_1 - B_1\alpha)(A_2 + B_2\alpha)^2C_1 \\ &= (A_1C_2 - A_2C_1)(A_1A_2 - 2B_1B_2\alpha^2) + (A_2B_1^2C_2 - A_1B_2^2C_1)\alpha^2 \\ &- [(B_1C_2 - B_2C_1)(B_1B_2\alpha^2 - 2A_1A_2) + A_1^2B_2C_2 - A_2^2B_1C_1]\alpha \end{aligned}$$

$$\begin{aligned} Z_3 &= C_1^2(A_2^2 - B_2^2\alpha^2) - (A_1^2 - B_1^2\alpha^2)C_2^2 \\ &= C_1^2A_2^2 - C_1^2B_2^2\alpha^2 - C_2^2A_1^2 - C_2^2B_1^2\alpha^2C_2^2 \\ &= C_1^2A_2^2 - C_2^2A_1^2 + (C_2^2B_1^2 - C_1^2B_2^2)\alpha^2 \end{aligned}$$

Hence $P_1 + P_2 = P_3 = (A_3 + B_3\alpha : A_3 - B_3\alpha : C_3)$ where:

$$\begin{cases} A_3 &= (A_1C_2 - A_2C_1)(A_1A_2 - 2B_1B_2\alpha^2) + (A_2B_1^2C_2 - A_1B_2^2C_1)\alpha^2 \\ B_3 &= (B_1C_2 - B_2C_1)(B_1B_2\alpha^2 - 2A_1A_2) + (A_1^2B_2C_2 - A_2^2B_1C_1) \\ C_3 &= C_1^2A_2^2 - C_2^2A_1^2 + (C_2^2B_1^2 - C_1^2B_2^2)\alpha^2 \end{cases}$$

Point doubling

Let $P_1 = (A_1 + B_1\alpha : A_1 - B_1\alpha : C_1)$ be a point on $\mathcal{H}_D(\mathbb{F}_{q^k})[r]$, $2P_1 = P_3 = (X_3 : Y_3 : Z_3)$ where the coordinates of P_3 are computed as follows:

$$\begin{aligned} X_3 &= (A_1 - B_1\alpha)((A_1 + B_1\alpha)^3 - C_1^3) \\ &= (A_1 - B_1\alpha)(A_1^3 + 3A_1^2B_1\alpha + 3A_1B_1^2\alpha^2 + B_1^3\alpha^3 - C_1^3) \\ &= 2A_1^3B_1\alpha - 2A_1B_1^3\alpha^3 + B_1C_1^3\alpha + (\frac{A_1^4}{\alpha} - B_1^4\alpha^3 - \frac{C_1^3A_1}{\alpha})\alpha \\ &= [2A_1^3B_1 - 2A_1B_1^3\alpha^2 + B_1C_1^3 + (\frac{A_1^4}{\alpha^2} - B_1^4\alpha^2 - \frac{C_1^3A_1}{\alpha^2})\alpha]\alpha \end{aligned}$$

$$\begin{aligned} Y_3 &= (A_1 + B_1\alpha)(C_1^3 - (A_1 - B_1\alpha)^3) \\ &= (A_1 + B_1\alpha)(C_1^3 - A_1^3 + 3A_1^2B_1\alpha - 3A_1B_1^2\alpha^2 + B_1^3\alpha^3) \\ &= 2A_1^3B_1\alpha - 2A_1B_1^3\alpha^3 + B_1C_1^3\alpha - (\frac{A_1^4}{\alpha} - B_1^4\alpha^3 - \frac{C_1^3A_1}{\alpha})\alpha \\ &= [2A_1^3B_1 - 2A_1B_1^3\alpha^2 + B_1C_1^3 - (\frac{A_1^4}{\alpha^2} - B_1^4\alpha^2 - \frac{C_1^3A_1}{\alpha^2})\alpha]\alpha \end{aligned}$$

$$\begin{aligned} Z_3 &= C_1((A_1 - B_1\alpha)^3 - (A_1 + B_1\alpha)^3) \\ &= C_1(-6A_1^2B_1\alpha - 2B_1^3\alpha^3) \\ &= -6A_1^2B_1C_1\alpha - 2B_1^3C_1\alpha^3 \end{aligned}$$

Therefore $2P_1 = P_3 = (A_3 + B_3\alpha : A_3 - B_3\alpha : C_3)$ where:

$$\begin{cases} A_3 &= 2A_1^3B_1 - 2A_1B_1^3\alpha^2 + B_1C_1^3 \\ B_3 &= (\frac{A_1^4}{\alpha^2} - B_1^4\alpha^2 - \frac{C_1^3A_1}{\alpha^2}) \\ C_3 &= -6A_1^2B_1C_1 - 2B_1^3C_1\alpha^2 \end{cases}$$

4.2. Simplification of the Miller function for the Ate pairing computation

In this section we simplify the Miller function using the representation of points previously mentioned. We show how this representation allows to completely ignore the denominator of the function given in Theorem 2 leading to efficient computations.

The addition step

Let $R = (A_1 + B_1\alpha : A_1 - B_1\alpha : C_1)$ and $Q = (A_2 + B_2\alpha : A_2 - B_2\alpha : C_2)$ be two points on $\mathcal{H}_D(\mathbb{F}_{q^k})[r]$, $R + P = (A_3 + B_3\alpha : A_3 - B_3\alpha : C_3)$ with $A_i, C_i, B_i \in \mathbb{F}_{q^{\frac{k}{2}}}$.

Let $P = (x_P, y_P)$ be a point on $\mathcal{H}_D(\mathbb{F}_q)[r]$, in projective coordinates P is denoted $P = (x_P : y_P : 1)$. In the addition step the Miller function is defined as:

$$h_{R,Q}(P) = \frac{l_1(P)}{l_2(P)} = \frac{C_X x_P + C_Y y_P + C_Z}{(Y_3Z_3 - Z_3X_3)(x_P + y_P) + (X_3^2 - Y_3^2)}$$

So we have: $l_2(P) = (Y_3Z_3 - Z_3X_3)(x_P + y_P) + (X_3^2 - Y_3^2)$. Let us show that $l_2(P) \in \mathbb{F}_{q^{\frac{k}{2}}}$. X_3, Y_3, Z_3 are the coordinates of $R+P$ then $X_3 = A_3 + B_3\alpha$, $Y_3 = A_3 - B_3\alpha$, $Z_3 = C_3$, therefore

$$\begin{aligned} Y_3 - X_3 &= A_3 - B_3\alpha - A_3 - B_3\alpha \\ &= -2B_3\alpha \\ Z_3(Y_3 - X_3) &= -2B_3C_3\alpha \end{aligned}$$

$$\begin{aligned} X_3^2 - Y_3^2 &= (A_3 + B_3\alpha)^2 - (A_3 - B_3\alpha)^2 \\ &= 4A_3B_3\alpha \end{aligned}$$

Thus

$$\begin{aligned} l_2(P) &= (Y_3Z_3 - Z_3X_3)(x_P + y_P) + (X_3^2 - Y_3^2) \\ &= -2B_3C_3(x_P + y_P)\alpha - 4A_3B_3\alpha \\ &= (-2B_3C_3(x_P + y_P) + 4A_3B_3)\alpha \end{aligned}$$

Substituting c_X, c_Y, c_Z by their values we obtain

$$\begin{aligned} c_X &= (A_1 - B_1\alpha)C_2 - C_1(A_2 - B_2\alpha) \\ &= A_1C_2 - A_2C_1 + (B_2C_1 - B_1C_2)\alpha \end{aligned}$$

$$\begin{aligned} c_Y &= C_1(A_2 + B_2\alpha) - C_2(A_1 + B_1\alpha) \\ &= A_2C_1 - A_1C_2 + (B_2C_1 - B_1C_2)\alpha \end{aligned}$$

$$\begin{aligned} c_Z &= (A_1 + B_1\alpha)(A_2 - B_2\alpha) - (A_1 - B_1\alpha)(A_2 + B_2\alpha) \\ &= 2A_2B_1\alpha - 2A_1B_2\alpha \\ &= (2A_2B_1 - 2A_1B_2)\alpha \end{aligned}$$

$$\begin{aligned} h_{R,Q}(P) &= \frac{c_X x_P + c_Y y_P + c_Z}{(-2B_3C_3(x_P + y_P) - 4A_3B_3)\alpha} \\ &= \frac{[c_X x_P + c_Y y_P + c_Z]\alpha^{-1}}{-2B_3C_3(x_P + y_P) + 4A_3B_3} \end{aligned}$$

Set $S = -2B_3C_3(x_P + y_P) + 4A_3B_3$ then $S \in \mathbb{F}_{q^{\frac{k}{2}}}$ since $A_i, B_i, C_i \in \mathbb{F}_{q^{\frac{k}{2}}}$. Therefore as $q^{k/2} - 1$ divides $q^k - 1$, S will tend to 1 during the final exponentiation (line 10 in the Miller algorithm). Thus S can be simply ignored in the algorithm. Thus

$$\begin{aligned} h_{R,Q}(P) &= l_1(P) \\ &= (A_1C_2 - A_2C_1)(x_P - y_p)\alpha^{-1} + (B_2C_1 - B_1C_2)(x_P + y_P) + \\ &+ 2A_2B_1 - 2A_1B_2 \\ &\quad \frac{(A_1C_2 - A_2C_1)}{\alpha^2}(x_P - y_P)\alpha + (B_2C_1 - B_1C_2)(x_P + y_P) + \\ &+ 2A_2B_1 - 2A_1B_2 \end{aligned}$$

Table 1
Combined formulas for point addition and Miller function evaluation for the Ate pairing.

Operations	Values	cost
$E \coloneqq A_1C_2$	$E = A_1C_2$	$1m_{k/2}$
$F \coloneqq A_2C_1$	$F = A_2C_1$	$1m_{k/2}$
$G \coloneqq A_1A_2$	$G = A_1A_2$	$1m_{k/2}$
$H \coloneqq B_1B_2$	$H = B_1B_2$	$1m_{k/2}$
$I \coloneqq B_1C_2$	$E = B_1C_2$	$1m_{k/2}$
$J \coloneqq B_2C_1$	$E = B_2C_1$	$1m_{k/2}$
$K \coloneqq A_2B_1$	$E = A_2B_1$	$1m_{k/2}$
$L \coloneqq A_1B_2$	$E = A_1B_2$	$1m_{k/2}$
$M \coloneqq H\alpha^2$	$M = B_1B_2\alpha^2$	$1m_c$
$N \coloneqq E - F$	$N = A_1C_2 - A_2C_1$	–
$O \coloneqq N(G - 2M)$	$O = (A_1C_2 - A_2C_1)\times$ $(A_1A_2 - 2B_1B_2\alpha^2)$	$1m_{k/2}$ –
$P_{11} \coloneqq KI$	$P_{11} = A_2B_1^2C_2$	$1m_{k/2}$
$P_{12} \coloneqq LJ$	$P_{12} = A_1B_2^2C_1$	$1m_{k/2}$
$P_{13} \coloneqq (P_{11} - P_{12})\alpha$	$P_{13} = (A_2B_2^2C_2 - A_1B_2^2C_1)\alpha^2$	$1m_c$
$Q1 \coloneqq I - J$	$Q1 = B_1C_2 - B_2C_1$	–
$R \coloneqq M - 2G$	$R = B_1B_2\alpha^2 - 2A_1A_2$	–
$S \coloneqq Q1R$	$S = (B_1C_2 - B_2C_1)\times$ $(B_1B_2\alpha^2 - 2A_1A_2)$	$1m_{k/2}$
$T_1 \coloneqq LE$	$T_1 = A_1^2B_2C_2$	$1m_{k/2}$
$T_2 \coloneqq KF$	$T_2 = A_2^2B_1C_1$	$1m_{k/2}$
$U_1 \coloneqq E + F$	$U_1 = A_1C_2 + A_2C_1$	–
$U_2 \coloneqq J + I$	$U_2 = B_2C_1 + B_1C_2$	–
$U_3 \coloneqq Q1U_2\alpha^2$	$U_3 = (B_1^2C_2^2 - B_2^2C_1^2)\alpha^2$	$1m_{k/2} + 1m_c$
$U_4 \coloneqq NU_1$	$U_4 = (A_1^2C_2^2 - A_2^2C_1^2)$	$1m_{k/2}$
$A_3 \coloneqq O + P_{13}, C_3 \coloneqq U_3 - U_4$	–	–
$B_3 \coloneqq S + T_1 - T_2$	–	–
$h_{R,Q}(P) \coloneqq \frac{N}{\alpha^2}(x_P - y_P)\alpha -$ $-Q1(x_P + y_P) + 2(K - L)$	– –	$\frac{k}{2}m_1 + \frac{k}{2}m_1$
$f \coloneqq f.h_{R,Q}(P)$	–	$1m_k$

Cost of the addition step

The main operations for this iteration of the Miller algorithm are $f \leftarrow f.h_{R,Q}(P)$ and $R \leftarrow R + Q$ where $R = (A_1 + B_1\alpha : A_1 - B_1\alpha : C_1)$, $Q = (A_2 + B_2\alpha : A_2 - B_2\alpha : C_2)$ and $P = (x_P : y_P : 1)$. Table 1 shows how to compute the two operations simultaneously and also gives the cost.

From Table 1, the cost of the Miller addition step for Ate pairing is $16m_{k/2} + 3m_c + km_1 + 1m_k + 3m_c$. When applying fixed point (mixed addition) the cost reduces to $14m_{k/2} + 3m_c + km_1 + 1m_k$

4.3. Doubling step in the Miller algorithm for Ate pairing

Let $R = (A_1 + B_1\alpha : A_1 - B_1\alpha : C_1)$ be a point on $\mathcal{H}_D$. Let $2R = (A_3 + B_3\alpha : A_3 - B_3\alpha : C_3)$ where $A_i, C_i, B_i \in \mathbb{F}_{q^{\frac{k}{2}}}$, $i = 1, 3$ and $P = (x_P, y_P)$. In the doubling step the Miller

function is given by

$$h_{R,R}(P) = \frac{C_X x_P + C_Y y_P + C_Z}{(Y_3Z_3 - Z_3X_3)(x_P + y_P) + (X_3^2 - Y_3^2)}$$

where X_3, Y_3 and Z_3 are the coordinates of $2R$ and

$$\begin{aligned} c_X &= (A_1 + B_1\alpha)^2 - DC_1(A_1 - B_1\alpha) \\ &= (A_1^2 - DC_1A_1) + (2A_1B_1 + DB_1C_1)\alpha + B_1^2\alpha^2 \\ c_Y &= (A_1 - B_1\alpha)^2 - DC_1(A_1 + B_1\alpha) \\ &= (A_1^2 - DC_1A_1) - (2A_1B_1 + DB_1C_1)\alpha + B_1^2\alpha^2 \\ c_Z &= C_1^2 - D(A_1 + B_1\alpha)(A_1 - B_1\alpha) \\ &= (C_1^2 - DA_1^2) + DB_1^2\alpha^2 \end{aligned}$$

Substituting X_3, Y_3 and Z_3 by their values we obtain

$$h_{R,Q}(P) = \frac{[c_X x_P + C_Y y_P + C_Z]\alpha^{-1}}{-2B_3C_3(x_P + y_P) - 4A_3B_3}$$

Set $S = -2B_3C_3(x_P + y_P) - 4A_3B_3$ then $S \in \mathbb{F}_{q^{\frac{k}{2}}}$ since $A_i, B_i, C_i \in \mathbb{F}_{q^{\frac{k}{2}}}$ and because $q^{k/2} - 1$ divides $q^k - 1$, S will tend to 1 during the final exponentiation (line 10 of the Miller algorithm) and therefore can be ignored during the execution of the algorithm. Thus we just consider

$$h_{R,R}(P) = [T(x_P + y_P) + V]\alpha + (2A_1B_1 + DB_1C_1)(x_P - y_P)$$

where

$$\begin{aligned} T &= \frac{A_1^2 - DC_1A_1}{\alpha^2} + B_1^2 \\ V &= \frac{C_1^2 - DA_1^2}{\alpha^2} + DB_1^2 \end{aligned}$$

Cost of the doubling step

In the Miller doubling step for Ate pairing we have the following operations $f \leftarrow f^2.h_{R,R}(P)$ and $R \leftarrow 2R$ where $R = (A_1+B_1\alpha : A_1-B_1\alpha : C_1)$, $P = (x_P : y_P : 1)$. Table 2 shows how to compute these two operations simultaneously and also gives the cost. Thus the total cost of the Miller doubling step for Ate pairing is $6m_{k/2}+8s_{k/2}+4m_c+km_1+1s_k+1m_k$.

5. Parallel computation of pairings on Hessian curves

Parallel computation refers to a group of independent processors working together to solve a large computational problem. The most important feature of such parallel computers is that all the processors share a single global memory space which is realized either at the hardware level or at the software level. The parallel computation is motivated by the need to reduce the execution time and to utilize large memory.

5.1. Parallel computation of the Tate pairing

In this section we show how to perform the computation of the Tate pairing when three processors are used. The Sage code to ensure the correctness of the algorithms in

Table 2
Combined formulas for point doubling Miller function evaluation for Ate pairing.

Operations	Values	cost
$E_1 \coloneqq A_1^2$	$E_1 = A_1^2$	$1s_{k/2}$
$F \coloneqq B_1^2$	$F = B_1^2$	$1s_{k/2}$
$G \coloneqq C_1^2$	$G = C_1^2$	$1s_{k/2}$
$H \coloneqq 1/2(A_1 + C_1)^2 - E_1 - G$	$H = A_1 C_1$	$1s_{k/2}$
$I \coloneqq 1/2(A_1 + B_1)^2 - E_1 - F$	$E = A_1 B_1$	$1s_{k/2}$
$J \coloneqq 1/2(B_1 + C_1)^2 - F - G$	$E = B_1 C_1$	$1s_{k/2}$
$K_1 \coloneqq F\alpha^2$	$K_1 = B_1^2\alpha^2$	$1m_c$
$K_2 \coloneqq K_1^2$	$K_2 = B_1^4\alpha^4$	$1s_{k/2}$
$L \coloneqq E_1 I$	$L = A_1^3 B_1$	$1m_{k/2}$
$M \coloneqq I K_1$	$M = A_1 B_1^3 \alpha^2$	$1m_{k/2}$
$N \coloneqq GJ$	$N = C_1^3 B_1$	$1m_{k/2}$
$E_2 \coloneqq E_1^2$	$E_2 = A_1^4$	$1s_{k/2}$
$O \coloneqq GH$	$O = C_1^3 A_1$	$1m_{k/2}$
$P_{11} \coloneqq E_1 J$	$P_{11} = A_1^2 B_1 C_1$	$1m_{k/2}$
$Q_1 \coloneqq K_1 J$	$Q_1 = C_1 B_1^3 \alpha^2$	$1m_{k/2}$
$T \coloneqq \frac{E_1 - DH}{\alpha^2} + F$	$T = \frac{A_1^2 - DA_1C_1}{\alpha^2} + B_1^2$	m_c
$V \coloneqq \frac{G - DE_1}{\alpha^2} + DF$	$V = \frac{C_1^2 - DA_1^2}{\alpha^2} + DB_1^2$	m_c
$A_3 \coloneqq 2L - 2M + N$	$A_3 = 2A_1^3 B_1 - 2A_1 B_1^3 \alpha^2 + C_1^3 B_1$	–
$B_3 \coloneqq \frac{1}{\alpha^2}(E_2 - O - K_2)$	$B_3 = \frac{A_1^4}{\alpha^2} - \frac{C_1^3 A_1}{\alpha^2} - B_1^4\alpha^2$	$1m_c$
$C_3 \coloneqq -6P_{11} - 2Q_1$	$C_3 = -6A_1^2 B_1 C_1 - 2C_1 B_1^3 \alpha^2$	–
$h_{R,R}(P) \coloneqq (T(x_P + y_P) + V)\alpha +$	–	–
$+(2I + DJ)(x_P - y_P)$	–	km_1
$f_1 \coloneqq f^2$	–	$1s_k$
$f_1 \coloneqq f_1 . h_{R,R}(P)$	–	$1m_k$

Table 3
Parallel execution of addition step for Tate pairing.

Processor1	Processor2	Processor3	Cost
$o_1 = Y_1 Z_2$	$o_2 = Z_1 Y_2$	$o_3 = Z_1 X_2$	$1m_1$
$o_4 = X_1 Z_2$	$o_5 = X_1 Y_2$	$o_6 = Y_1 X_2$	$1m_1$
$l_1 = o_1 o_4$	$l_2 = o_2 o_5$	$l_3 = o_3 o_6$	$1m_1$
$l_4 = o_4 o_5$	$l_5 = o_1 o_6$	$l_6 = o_2 o_3$	$1m_1$
$c_X = o_1 - o_2$	$c_Y = o_3 - o_4$	$c_Z = o_5 - o_6$	–
$X_3 = l_5 - l_2$	$Y_3 = l_4 - l_3$	$Z_3 = l_6 - l_1$	–
$t_1 = c_X x_Q$	$t_2 = c_Y y_Q$	–	$\frac{k}{2}m_1$
$f = f.(t_1 + t_2 + c_Z)$	–	–	$1m_k$

this section is available at http://www.camercrypt.org/sagehessian.txt. Table 3 presents the parallel execution of the addition step in Miller's algorithm and Table 4 the parallel execution of the addition step. From Table 3 the total cost of the parallel execution of the addition step is $4m_1 + \frac{k}{2}m_1 + 1m_k$ while the normal execution of the addition requires $12m_1 + km_1 + 1m_k$ when a single processor is used [18].

Table 4 presents the parallel execution of the doubling step in Miller's algorithm and its cost for the Tate pairing. From Table 4 the total cost of parallel execution of the doubling step is $1m_k + 1s_k + \frac{k}{2}m_1 + 1m_1 + 2s_1$.

Table 4
Parallel execution of doubling step for Tate pairing.

Processor1	Processor2	Processor3	Cost
$l_1 = X_1^2$	$l_2 = Y_1^2$	$l_3 = Z_1^2$	s_1
$l_4 = X_1Y_1$	$l_5 = X_1Z_1$	$l_6 = Y_1Z_1$	s_1
$c_X = 3l_1 - 3Dl_6$	$c_Y = 3l_2 - 3Dl_5$	$c_Z = 3l_3 - 3Dl_4$	–
$X_3 = (l_4 - l_6)\times$ $(l_5 + l_1 + l_3)$	$Y_3 = (l_5 - l_4)\times$ $(l_6 + l_2 + l_3)$	$Z_3 = (l_6 - l_5)\times$ $(l_4 + l_1 + l_2)$	$1m_1$ –
$t_1 = c_X x_Q$ $f = f.(t_1 + t_2 + c_Z)$	$t_2 = c_Y y_Q$ –	$f_1 = f^2$ –	$\frac{k}{2}m_1 + 1s_k$ $1m_k$

Table 5
Parallel execution of addition step for Ate pairing.

Processor1	Processor2	Processor3	Cost
$l_1 = A_1C_2$	$l_2 = A_2C_1$	$l_3 = A_1B_2$	$1m_{k/2}$
$l_4 = B_1C_2$	$l_5 = B_2C_1$	$l_6 = A_2B_1$	$1m_{k/2}$
$l_7 = A_1A_2$	$l_8 = B_2B_1$	$l_9 = l_6l_4$	$1m_{k/2}$
$l_{10} = l_3l_5$	$l_{11} = l_3l_1$	$l_{12} = l_2l_6$	$1m_{k/2}$
$a_1 = l_4 - l_5$	$a_2 = l_4 + l_5$	–	–
$l_{13} = a_1a_2$	–	–	$1m_{k/2}$
$c_1 = l_8\alpha^2$	$c_2 = (l_9 - l_{10})\alpha^2$	$c_3 = a_1a_2\alpha^2$	$1m_c$
$a_3 = (l_1 - l_2)$	$a_4 = l_1 + l_2$	$a_5 = l_3 - 2c_1$	–
$l_{14} = a_3a_5$	$l_{15} = a_1(c_1 - 2l_7)$	$l_{16} = a_3a_4$	$1m_{k/2}$
$A_3 = l_{14} + c_2$	$B_3 = l_{15} + l_{11} - l_{12}$	$C_3 = -l_{16} + c_3$	–
$t_1 = \frac{a_3}{\alpha^2}(x_P - y_P)$	$t_2 = -a_1(x_P + y_P) + 2(l_6 - l_3)$	–	$\frac{k}{2}m_1$
$f = f.[t_1\alpha + t_2]$	–	–	$1m_k$

5.2. *Parallel computation of the Ate pairing*

This section concerns the parallel computation of the Ate pairing. We proceed the same manner as we previously did for the Tate pairing.

The following Table 5 shows how the Ate pairing can be computed using three processors. From Table 5 the total cost of parallel execution of the addition step in Ate pairing is $6m_{k/2} + 1m_c + \frac{k}{2}m_1 + 1m_k$.

Table 6 presents the parallel execution of the doubling step of the Miller algorithm. Thus the total cost of parallel execution of the doubling step is $1m_k + 2m_{k/2} + 2s_{k/2} + \frac{k}{2}m_1 + 1s_k + 2m_c$.

5.3. *Conclusion*

Table 7 summarizes all the costs obtained in this work and compare with previous results.

6. Conclusion

In this work we extended the results of Gu et *al.* [18] on the computation of the Tate pairing to the computation of the Ate pairing on the Hessian model of elliptic curves. We also succeeded to show how the computation of the two pairings can be parallelized. We provided a Sage code to ensure the correctness of all the formulas and the algorithms in this work.

Table 6
Parallel execution of doubling step for Ate pairing.

Processor1	Processor2	Processor3	Cost
$o_1 = A_1^2$	$o_2 = B_1^2$	$o_3 = C_1^2$	$1s_{k/2}$
$o_4 = A_1 B_1$	$o_5 = A_1 C_1$	$o_6 = B_1 C_1$	$1s_{k/2}$
$c_1 = o_2 \alpha^2$	–	–	$1m_c$
$a_1 = o_1 - Do_5$	$a_2 = 2o_4 + Do_6$	$a_3 = o_3 - Do_1$	–
$a_4 = o_1 - c_1$	$a_5 = o_1 + c_1$	–	–
$l_1 = o_3 o_6$	$l_2 = o_3 o_5$	$l_3 = a_4 a_5$	$1m_{k/2}$
$A_3 = 2o_4(o_1 - c_1) + l_1$	$C_3 = -o_6(6o_1 + 2c_1)$	–	$1m_{k/2}$
$B_3 = \frac{1}{\alpha^2}(L3 - l_2)$	$T = \frac{a_1}{\alpha^2} + o_2$	$V = \frac{a_3}{\alpha^2} + Do_2$	$1m_c$
$t_1 = T(x_P + y_P) + V$	$t_2 = a_2(x_P - y_P)$	–	$\frac{k}{2}m_1$
$f_1 = f^2$	–	–	$1s_k$
$f = f.[t_1\alpha + t_2]$	–	–	$1m_k$

Table 7
Summary of results and comparison with previous results.

Pairings	Doubling	Addition	Mixed addition
Tate(P,Q) [18]	$1m_k + 1s_k + km_1 +$ $3m_1 + 6s_1$	$1m_k + km_1 + 12m_1$	$1m_k + km_1 + 10m_1$
Tate(P,Q) **Parallelization** (**This work**)	$1m_k + 1s_k + \frac{k}{2}m_1 +$ $1m_1 + 2s_1$	$1m_k + \frac{k}{2}m_1 + 4m_1$	–
Ate(P,Q) (**This work**)	$6m_{k/2} + 8s_{k/2} + 4m_c +$ $km_1 + 1s_k + 1m_k$	$16m_{k/2} + 3m_c + km_1 +$ $1m_k$	$14m_{k/2} + 3m_c + km_1 +$ $1m_k$
Ate(P,Q) **Parallelization** (**This work**)	$2m_{k/2} + 2s_{k/2} + 2m_c +$ $\frac{k}{2}m_1 + 1s_k + 1m_k$	$6m_{k/2} + 1m_c + \frac{k}{2}m_1 +$ $1m_k$	–

ACKNOWLEDGMENTS

The author thanks the anonymous reviewers for comments which help to improve the quality of this work.

REFERENCES

[1] Diego F. Aranha, Laura Fuentes-Castaneda, Edward Knapp, Alfred Menezes, Francisco Rodríguez-Henríquez, (2012) Implementing pairings at the 192-bit security level, in: Pairing-Based Cryptography - Pairing 2012 - 5th International Conference, Cologne, Germany, May 16–18, 2012, Revised Selected Papers, pp. 177–195.

[2] Diego F. Aranha, Koray Karabina, Patrick Longa, Catherine H. Gebotys, Julio López, Faster explicit formulas for computing pairings over ordinary curves, in: Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15–19, 2011. Proceedings, 2011, pp. 48–68.

[3] C. Arene, T. Lange, M. Naehrig, C. Ritzenthaler, Faster computation of the Tate pairing, J. Number Theory 131 (5) (2011) 842–857.

[4] R. Avanzi, H. Cohen, C. Doche, G. Frey, T. Lange, K. Nguyen, F. Vercauteren, Handbook of elliptic and hyperelliptic curve cryptography, Discrete Math. Appl. (2006).

[5] P.S.L.M. Barreto, B. Lynn, M. Scott, Efficient implementation of pairing-based cryptosystems, J. Cryptol. 17 (4) (2004) 321-334.

[6] D. Boneh, M. Franklin, Identity-based encryption from the Weil pairing., SIAM J. Comput. 32 (3) (2003) 586-615.

[7] Dan Boneh, Matthew K. Franklin, Identity-based encryption from the weil pairing, in: Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19–23, 2001, Proceedings, 2001, pp. 213–229.
[8] D.V. Chudnovsky, G.V. Chudnovky, Sequences of numbers generated by addition in formal groups and new primality and factorization tests, Adv. Appl. Math. 7 (4) (1986) 385-434.
[9] C. Costello, T. Lange, M. Naehrig, Faster pairing computations on curves with high-degree twists, in: PKC 2010, in: LNCS, vol. 6056, 2010, pp. 224–242.
[10] M.P.L. Das, P. Sarkar, Pairing computation on twisted Edwards form elliptic curves, in: Pairing 2008, in: LNCS, vol. 5209, 2008, pp. 192–210.
[11] S. Duquesne, N. El Mrabet, E. Fouotsa, Efficient pairing computation on Jacobi quartic elliptic curve, J. Math. Cryptol. 8 (4) (2014) 331–362.
[12] S. Duquesne, G. Frey, Background on pairings, in [4], 2005, pp. 115–124.
[13] Sylvain Duquesne, Loubna Ghammam, Memory-saving computation of the pairing final exponentiation on BN curves, Groups Complex. Cryptol. 8 (1) (2016) 75–90.
[14] Ratna Dutta, Rana Barua, Palash Sarkar, Pairing-based cryptographic protocols : A survey, in: IACR Cryptology EPrint Archive, 2004, p. 64.
[15] Reza Rezaeian Farashahi, Marc Joye, Efficient arithmetic on Hessian curves, in: Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26–28, 2010. Proceedings, 2010, pp. 243–260.
[16] Gerhard Frey, Michael Müller, Hans-Georg Rück, The tate pairing and the discrete logarithm applied to elliptic curve cryptosystems, IEEE Trans. Inf. Theory 45 (5) (1999) 1717–1719.
[17] S.D. Galbraith, Pairings, in: London Mathematics Society Lecture Note Series - Cambridge University Press, vol. 317, 2005, pp. 183–213.
[18] Haihua Gu, Dawu Gu, WenLu Xie, Efficient pairing computation on elliptic curves in hessian form, in: Information Security and Cryptology - ICISC 2010 - 13th International Conference, Seoul, Korea, December 1–3, 2010, Revised Selected Papers, 2010, pp. 169–176.
[19] Florian Hess, Pairing lattices, in: Pairing-Based Cryptography - Pairing 2008, Second International Conference, Egham, UK, September 1–3, 2008. Proceedings, 2008, pp. 18–38.
[20] Florian Hess, Nigel P. Smart, Frederik Vercauteren, The eta pairing revisited, IEEE Trans. Inf. Theory 52 (10) (2006) 4595–4602.
[21] T. Izu, T. Takagi, Exceptional procedure attack on elliptic curve cryptosystems, in: PKC 2003, in: LNCS, vol. 2567, Springer, 2003, pp. 224–239.
[22] Antoine Joux, A one round protocol for Tripartite Diffie-Hellman, in: Algorithmic Number Theory, 4th International Symposium, ANTS-IV, Leiden, the Netherlands, July 2–7, 2000, Proceedings, pp. 385–394.
[23] Marc Joye, Jean-Jacques Quisquater, Hessian elliptic curves and side-channel attacks, in: Cryptographic Hardware and Embedded Systems - CHES 2001, Third International Workshop, Paris, France, May 14–16, 2001, Proceedings, Generators, 2001, pp. 402–410.
[24] V. Miller, Short programs for functions on curves, IBM Watson, T.J. Research Center http://crypto.stanford.edu/miller/miller.pdf, vol., 1986.
[25] V. Miller, A remark concerning m-divisibility and the discrete logarithm problem in the divisor class group of groups, Math. Comp. 62 (1994) 865–874.
[26] Nigel P. Smart, The hessian form of an elliptic curve, in: Cryptographic Hardware and Embedded Systems - CHES 2001, Third International Workshop, Paris, France, May 14-16, 2001, Proceedings, Generators, 2001, pp. 118–125.
[27] W. Stein, Sage Mathematics Software (Version 4.8), The Sage Group, 2012. http://www.sagemath.org.
[28] Frederik Vercauteren, Optimal pairings, IEEE Trans. Inf. Theory 56 (1) (2010) 455–461.
[29] Lijun Zhang, Kunpeng Wang, Hong Wang, Dingfeng Ye, Another elliptic curve model for faster pairing computation, in: Information Security Practice and Experience - 7th International Conference, ISPEC 2011, Guangzhou, China, May 30 - June 1, 2011. Proceedings, 2011, pp. 432–446.