Constructing Symmetric Pairings over Supersingular Elliptic Curves with Embedding Degree Three

Tadanori Teruya[1], Kazutaka Saito[2,⋆], Naoki Kanayama[3],
Yuto Kawahara[4], Tetsutaro Kobayashi[4], and Eiji Okamoto[3]

[1] Research Institute for Secure Systems,
National Institute of Advanced Industrial Science and Technology,
1-1-1 Umezono, Tsukuba-shi, Ibaraki-ken 305-8568, Japan
[2] Internet Initiative Japan Inc.,
Jinbocho Mitsui Bldg., 1-105 Kanda Jinbo-cho, Chiyoda-ku, Tokyo 101-0051, Japan
[3] Faculty of Systems and Information Engineering,
University of Tsukuba,
1-1-1, Ten-nohdai, Tsukuba-shi, Ibaraki-ken, 305-8573 Japan
[4] NTT Secure Platform Laboratories,
3-9-11, Midori-cho, Musashino-shi, Tokyo 180-8585, Japan

Abstract. In the present paper, we propose constructing symmetric pairings by applying the Ate pairing to supersingular elliptic curves over finite fields that have large characteristics with embedding degree three. We also propose an efficient algorithm of the Ate pairing on these curves. To construct the algorithm, we apply the denominator elimination technique and the signed-binary approach to the Miller's algorithm, and improve the final exponentiation. We then show the efficiency of the proposed method through an experimental implementation.

Keywords: supersingular elliptic curves, symmetric pairings.

1 Introduction

Since Sakai et al. [26] and Boneh et al. [6,7] independently proposed pairing-based cryptosystems, many other novel cryptographic schemes that use pairings have been proposed.

An admissible pairing e is a mapping from two source groups $\mathbb{G}_1$ and $\mathbb{G}_2$, both of order r, to target group $\mathbb{G}_T$, also of order r. The mapping must be bilinear, nondegenerate, and able to be computed efficiently. Typically, $\mathbb{G}_1$ and $\mathbb{G}_2$ are denoted as additive groups, and $\mathbb{G}_T$ is denoted as a multiplicative group. The bilinearity is described as follows:

$$e(P_1 + P_2, Q) = e(P_1, Q)e(P_2, Q),$$
$$e(P, Q_1 + Q_2) = e(P, Q_1)e(P, Q_2),$$

⋆ Part of this work was done while the second author was a student at the University of Tsukuba.

Z. Cao and F. Zhang (Eds.): Pairing 2013, LNCS 8365, pp. 97–112, 2014.

where $P, P_1, P_2 \in \mathbb{G}_1$ and $Q, Q_1, Q_2 \in \mathbb{G}_2$. In the present paper, the case $\mathbb{G}_1 = \mathbb{G}_2$ of pairings from $\mathbb{G}_1 \times \mathbb{G}_1$ to $\mathbb{G}_T$ is referred to as a *symmetric pairing* (the "type 1" pairing in [11]), and the other case, i.e., $\mathbb{G}_1 \neq \mathbb{G}_2$, is referred to as an *asymmetric pairing*. Symmetric pairings and asymmetric pairings are similar in some ways, but they differ in their mathematical structures and the security assumptions used to construct cryptographic schemes. It has been reported for several implementations that asymmetric pairings are the best choice for higher levels of security. However, symmetric pairings are often used to construct cryptographic schemes because their mathematical structures are simpler than asymmetric pairings. Currently, the most popular way to construct symmetric pairings is to use supersingular (hyper)elliptic curves. These curves have many properties that are friendly to the computations for symmetric parings, for example, the existence of distortion maps. In particular, supersingular elliptic curves over finite fields of small characteristic have been widely used for computing symmetric pairings.

However, there have also been several proposals of security analysis for solving the discrete logarithm problem (DLP) on $\mathbb{G}_T$ in the case of small characteristic [15,1]. Hayashi et al. [15] showed that the DLP over $\mathbb{F}_{3^{97 \cdot 6}}$ can be solved. Subsequently, Adj et al. [1] reported that the actual security level of the curves with characteristic 3 is lower than was previously estimated. In the case of characteristic 2, Joux [17] reported that the DLP in $\mathbb{F}_{2^{254 \cdot 24}}$ can be solved in practical time. $\mathbb{G}_T$ is included in the extension field of degree 4 or 12, thus $\mathbb{G}_T$ is also included in $\mathbb{F}^*_{2^{254 \cdot 24}}$. These results will lead to the reevaluation of their security level, and the key length, and performance of them are expected to be worse.

As mentioned above, asymmetric pairings currently perform the best. The constructions of cryptographic schemes on asymmetric pairings that are similar to those that have been proposed on symmetric pairings have been considered. Chatterjee et al. [9] investigated the construction of several cryptographic schemes built on asymmetric pairings and compared their performance. The most interesting result of Chatterjee et al. is the construction of a Waters signature scheme [31] on an asymmetric pairing. The original Waters scheme is constructed on a symmetric pairing, and the public key, private key, and signature are all very small. On the other hand, in order to construct, the modified Waters signature schemes proposed by Chatterjee et al., they require either larger public and private keys or a public parameter generated by a trusted third party. Hence, there are several trade-offs between using symmetric or asymmetric pairings.

Contribution

In the present paper, we consider efficient algorithms for supersingular elliptic curves that are defined over extension fields that have large characteristics. Supersingular curves defined over finite fields that have large characteristics are classified into two types. These curves are summarized in Table 1. The type 1 curve is defined over prime fields, and the type 2 curve is defined over extension fields.

The use of the type 1 curve in the construction of pairing-based cryptosystems was demonstrated by Boneh et al. [6,7]. The type 2 curve was introduced by Verheul [29,30] in a different context. However, using these curves for pairing-based cryptosystems is not as popular as using supersingular curves over fields with small characteristics. One of the reasons is that the type 1 elliptic curves have not been commonly used in recent cryptographic pairings (such as the η_T pairing [3] and the Ate pairing [16]). For supersingular elliptic curves over small-characteristic finite fields, we can use the η_T pairing $f_{T,P}(Q)$ instead of the Tate pairing $f_{r,P}(Q)$, since, in this case, the bit length of T is half that of r. Thus, for these curves, the η_T pairing can be computed much faster than the Tate pairing. There is, however, almost no advantage to using the Eta or the Ate pairing for type 1 supersingular elliptic curves because their trace is 0.

On the other hand, computing pairings over type 2 supersingular elliptic curves has not been extensively investigated. One of the reasons for this is that the embedding degree k of such curves is 3. This is smaller than that of the supersingular elliptic curves over small characteristic fields (in these cases, $k = 4, 6$), and it thus would seem that there would not be much advantage to using type 2 elliptic curves. Furthermore, the η_T pairing is not applicable for type 2 curves because their k is odd, and we cannot directly use the denominator elimination technique [4] that is used when k is even. However, Lin et al. [21] proposed a denominator elimination technique for elliptic curves with an odd embedding degree. Also note that the embedding degree $k = 3$ of a type 2 elliptic curve is slightly larger than the degree $k = 2$ of elliptic curves over prime fields.

Another advantage of using a type 2 elliptic curve is that we can use the efficient method for scalar multiplication that was proposed by Gallant et al. [12] because the group order r is of the form $r = p^2 \pm p + 1$. This can save much computation time.

In the present paper, we propose a method for efficiently computing symmetric pairings over type 2 elliptic curves.

The remainder of this paper is organized as follows. Section 2 presents a brief mathematical description of pairings. Section 3 presents the reduced Ate pairings on type 2 elliptic curves; this is the main result of the present study. Section 4 presents an experimental implementation of the proposed method. Finally, conclusions are presented in Section 5.

2 Mathematical Preliminaries

2.1 Pairings

Let E be an elliptic curve over a finite field $\mathbb{F}_q$ with q elements. The set of $\mathbb{F}_q$-rational points of E is denoted as $E(\mathbb{F}_q)$. Let $E(\mathbb{F}_q)[r]$ denote the subgroup of r-torsion points in $E(\mathbb{F}_q)$. We write O for the point at infinity on E. Consider a large prime r such that $r \mid \#E(\mathbb{F}_q)$, and denote the embedding degree by k, which is the smallest positive integer such that r divides $q^k - 1$. Let π_q be the q-power Frobenius endomorphism $\pi_q : E \to E, (x, y) \mapsto (x^q, y^q)$. We denote the

Table 1. Summary of supersingular elliptic curves defined over large characteristic finite fields

Type	1		2
Base Field	$\mathbb{F}_p$, where $p > 3$ and $p \equiv 3 \pmod 4$	$\mathbb{F}_p$, where $p > 3$ and $p \equiv 2 \pmod 3$	$\mathbb{F}_{p^2}$, where $p > 3$ and $p \equiv 5 \pmod 6$
Curve	$E/\mathbb{F}_p : Y^2 = X^3 + X$	$E/\mathbb{F}_p : Y^2 = X^3 + 1$	$E/\mathbb{F}_{p^2} : Y^2 = X^3 + b$, where b is a square but not a cube in $\mathbb{F}_{p^2}$
Order	$\#E(\mathbb{F}_p) = p + 1$	$\#E(\mathbb{F}_p) = p + 1$	$\#E(\mathbb{F}_{p^2}) = p^2 + 1 - t$, $t = \pm p$
Embedding Degree	2	2	$\begin{cases} 3 & \text{if } t = p, \\ 3/2 & \text{otherwise} \end{cases}$
Distortion Map	$\iota : (x, y) \mapsto (-x, \zeta_4 y)$, where ζ_4 is a proper element in $\mathbb{F}_{p^2}$ and $\zeta_4^4 = 1$	$\iota : (x, y) \mapsto (\zeta_3 x, y)$, where ζ_3 is a proper element in $\mathbb{F}_{p^2}$ and $\zeta_3^3 = 1$	$\iota : (x, y) \mapsto (u^2 x^p, u^3 y^p)$, where u is a proper element in $\mathbb{F}_{p^6}$ and $u^6 = b/b^p$

trace of Frobenius by t, i.e., $\#E(\mathbb{F}_q) = q + 1 - t$. Finally, let $\mu_r (\subset \mathbb{F}_{q^k}^\times)$ be the group of r-th roots of unity.

Tate Pairing. Let $P \in E(\mathbb{F}_{q^k})[r]$ and $Q \in E(\mathbb{F}_{q^k})$. Choose a point $R \in E(\mathbb{F}_{q^k})$ such that the supports of $\mathrm{div}(f_{r,P}) = r(P) - r(O)$ and $D_Q := (Q + R) - (R)$ are disjoint. Then, the Tate pairing (Tate–Lichtenbaum pairing) is defined by:

$$\langle \cdot, \cdot \rangle_r : E(\mathbb{F}_{q^k})[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) \to \mathbb{F}_{q^k}^\times / (\mathbb{F}_{q^k}^\times)^r,$$
$$(P, Q) \mapsto \langle P, Q \rangle_r := f_{r,P}(D_Q) \bmod (\mathbb{F}_{q^k}^\times)^r .$$

It has been shown that $\langle P, Q \rangle_r$ is bilinear and nondegenerate.

For cryptography applications, it is convenient to define pairings for which the outputs are unique values rather than equivalence classes. Thus, we consider the reduced Tate pairing defined by:

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

We call the operation $z \mapsto z^{(q^k - 1)/r}$ final exponentiation.

Ate Pairing. The Ate pairing, proposed by Hess et al. [16], is a generalization of the η_T pairing [3]. The Ate pairing can be applied to not only supersingular but also to ordinary elliptic curves.

Let $T = t - 1$. We choose integers N and L such that $N = \gcd(T^k - 1, q^k - 1)$ and $T^k - 1 = LN$. We assume that r^2 does not divide $q^k - 1$.

Definition 1. *The reduced Ate pairing (on $\mathbb{G}_2 \times \mathbb{G}_1$) is defined by*

$$a_T: \ \mathbb{G}_2 \times \mathbb{G}_1 \to \mu_r;$$
$$(Q,\ P) \mapsto f_{T,Q}(P)^{(q^k-1)/r},$$

where the rational function $f_{T,Q}$ on E is the normalized function that satisfies

$$(f_{T,Q}) = T(Q) - ([T]Q) - (T-1)(O).$$

The definition for the normalization of rational functions is given in [22].

Many variants of the Ate pairing have been proposed, including the Ate$_i$ pairing [32], the R-Ate pairing [20], and the optimal pairing [28]. These pairings are defined on $\mathbb{G}_2 \times \mathbb{G}_1$ using normalization functions. The Ate pairing and its variants are also defined in $\mathbb{G}_1 \times \mathbb{G}_2$[1], there is no need to consider normalization [25].

2.2 Supersingular Elliptic Curves Defined over an Extension Field

We propose a method for the efficient computation of a symmetric pairing over a supersingular elliptic curve $E/\mathbb{F}_q$, as characterized in [30]:

$$E/\mathbb{F}_q : Y^2 = X^3 + b, \tag{1}$$

where $q = p^2$ and the quantities in (1) satisfy the following conditions:

- p is a prime larger than 3;
- $p \equiv 5 \pmod 6$;
- $b \in \mathbb{F}_q$ is a square in $\mathbb{F}_q$ but is not a cube in $\mathbb{F}_q$.

The trace t of the q-power Frobenius endmorphism π_q on $E/\mathbb{F}_q$ and the cardinality $\#E(\mathbb{F}_q)$ are determined, respectively, by:

$$\begin{aligned} t &= p, \\ \#E(\mathbb{F}_q) &= p^2 - p + 1. \end{aligned} \tag{2}$$

Therefore, the embedding degree of $E/\mathbb{F}_q$ is $k = 3$.

Let r be the largest prime divisor of $\#E(\mathbb{F}_q)$, and let $h = \#E(\mathbb{F}_q)/r$. We assume that $r^2 \nmid \#E/\mathbb{F}_q$. Hereafter, we write $\mathbb{G}_1 := E(\mathbb{F}_q)[r]$ and call $\mathbb{G}_1$ the source group of pairings.

[1] When E is supersingular, the Ate pairing is defined using the same formula. When E is ordinary, the Ate pairing is defined using a slightly different formula. In this case, the Ate pairing is called the twisted Ate pairing; for more information see [16].

2.3 Distortion Map

The distortion map on $E/\mathbb{F}_q$ is defined as follows.

Lemma 1 (distortion map, [30]). *Let $E/\mathbb{F}_q : Y^2 = X^3 + b$ be an elliptic curve, and let u be a proper element in $\mathbb{F}_{q^3}$ such that $u^6 = b/b^p$.*
Then

$$\iota : E(\mathbb{F}_q) \to E(\mathbb{F}_{q^3}) \setminus E(\mathbb{F}_q), (x, y) \mapsto (u^2 x^p, u^3 y^p) \tag{3}$$

is a distortion map on E.

We can construct a symmetric pairing $e(\cdot, \cdot)$ by "compositing" the distortion map ι to the Tate pairing $\langle \cdot, \cdot \rangle$ on E, that is,

$$e(\cdot, \cdot) := \langle \cdot, \iota(\cdot) \rangle .$$

3 The Main Result

As mentioned in Section 1, there is almost no advantage to using the Ate pairing for type 1 supersingular elliptic curves defined over prime fields, because $t = 0$ for them. However, the Ate pairing for a type 2 curve, as discussed in Section 2.2, can be computed efficiently. In the present section, we propose an algorithm for computing Ate pairings over type 2 curves.

First, we compare type 2 curves with type 1 curves from the viewpoint of pairing-based cryptography.

3.1 Comparison between Type 1 and Type 2 Curves

When we use elliptic curves over $\mathbb{F}_{p^2}$, we need to consider the hardness of the elliptic curve discrete logarithm problem (ECDLP) on $E/\mathbb{F}_{p^2}$ against a Gaudry–Hess–Smart (GHS) attack or an attack by one of its variants. Let $E/\mathbb{F}_{p^2} : Y^2 = F(X)$ be an elliptic curve. According to Momose et al. [23], if $F(X)$ is irreducible over $\mathbb{F}_{p^2}$ or can be factored as a product of linear factors, then E is equivalent to the elliptic curves of the Scholten form [27], and we can use degree 2 Weil restrictions to make a genus 2 hyperelliptic curve $C/\mathbb{F}_p$. Hence, the ECDLP on $E/\mathbb{F}_{p^2}$ is reduced to the hyperelliptic curve discrete logarithm problem (HECDLP) on the Jacobian group of $C/\mathbb{F}_p$. In the case of our target curve $\mathbb{F}_{p^2}$, $F(X) = X^3 + b$ is generally irreducible since b is not a cube in $\mathbb{F}_{p^2}$. Hence, degree 2 Weil restrictions are applicable to $E/\mathbb{F}_{p^2}$, and we must choose parameters $(q(= p^2), r, t)$ to protect against this attack. When we solve the HECDLP on the Jacobian of $C/\mathbb{F}_p$, which is obtained by applying degree 2 Weil restrictions to $E/\mathbb{F}_{p^2}$ and using the double-large prime variation-of-index calculus of Gaudry et al. [13] and Nagao [24]. The running cost is $\tilde{O}(q)$ when the genus of C is 2.

When we choose (q, r, t) such that q^3 is at least 960 bits, then $q = p^2$ is at least 320 bits. Hence, the running cost $\tilde{O}(q)$ is larger than $O(2^{320})$ when the characteristic p is 160 bits. We now need to choose a larger q; for example, if p

is 200 bits, we can choose a q^3 that is 1200 bits. We can thus obtain parameters that are secure against the Weil restrictions.

Next, we consider the hardness of finite-field discrete logarithm problem (FFDLP) on $\mathbb{G}_T$. To guarantee security, the FFDLP must be hard. The elliptic curve introduced in Section 2.2 is defined over a large characteristic extension field. Freeman et al. [10] suggested that the size of q^k needs 2200-3600 bits in order to guarantee the 112-bit level of security. We can also consider another setting, which based on the function-field sieve attack [2], and its complexity is:

$$\exp\left(\left(\frac{32}{9}+o(1)\right)^{\frac{1}{3}} \cdot (\log q^k)^{\frac{1}{3}} \cdot (\log\log q^k)^{\frac{2}{3}}\right). \tag{4}$$

Recently, Joux and Pierrot [18] proposed the extended special number field sieve to compute FFDLP in $\mathbb{F}_{p^n}$, where p has an adequate sparse representation. The concern with the security analysis of FFDLP has been growing by their investigations. It is interesting to follow up their results further, but it is not our present concern.

Next, we compare the parameters of the type 1 and type 2 elliptic curves for the 112-bit level of security based on Equation (4). We suppose $o(1)$ in Equation (4) is 0, namely, we need that the size of the resulting $\mathbb{F}_{p^k}$, which includes $\mathbb{G}_T$, is around 3132 bits. The summary of the comparison of parameters is shown in Table 2. The base field of the type 2 curve is smaller than that of the type 1 curve. Moreover, the base field of the type 2 curve is an extension field. Thus, the characteristic of the type 2 curve is small, its arithmetic is implementation friendly, and the representation of the elements in $\mathbb{G}_1$ is smaller than it is for the type 1 curve. However, the order of the type 2 curve is larger than that of the type 1 curve. If the method proposed by Gallant et al. [12] (GLV) is used for scalar multiplication on $\mathbb{G}_1$ for the type 2 elliptic curves, then the length of this operation is cut in half; nevertheless, the reduced length is still larger than that for type 1 curves. Scalar multiplication on type 2 curves is considerably slower than it is for type 1 curves. But the final exponentiation is faster for type 2 curves because the costly part of this operation on type 2 curves is smaller than it is for type 1 curves. Hence, the Weil pairing is considerable for type 1 curves. This means that Miller's algorithm is evaluated in twice the time it takes to calculate a pairing on type 1 curves. The actual Miller loop parameters for the type 1 and type 2 curves are $2 \cdot 224$ bits and 522 bits, respectively, so that of the type 2 curves is still larger. However, the arithmetic of the type 2 curves can be implemented efficiently by using the pseudo-Mersenne prime [14], and we show several instances of them in Section 4.1.

3.2 Miller's Algorithm

We now present an algorithm for computing the Ate pairing over type 2 curves.

In this algorithm, we use a denominator elimination technique based on the following lemma.

Table 2. Summary of parameter comparison for the 112-bit security level which is discussed in Section 3.1, where "GLV Method" is the method proposed by Gallant et al. [12], "Miller Loop Parameter" is the integer that determines the number of iterations of Miller's algorithm, and "Final Exp." is the exponents of operations in the final exponentiation

Type	1	2
Base Field	$\mathbb{F}_p$: p is a 1566-bit prime number	$\mathbb{F}_{p^2}$: 1044-bit size and p is a 522-bit prime number
Order	r: 224-bit prime number such that $p+1=hr$	r: prime number such that hr is a 1044-bit integer and h is small
GLV Method	Not applicable	Applicable by using $\phi : (x,y) \mapsto (\zeta_3 x, y)$, where $\zeta_3 \in \mu_3 \subset \mathbb{F}_{p^2}^*$
Miller Loop Parameter	r: 224-bit prime number with low Hamming weight	$p-1$: 522-bit integer with small number of non-zero components in NAF encoding
Final Exp.	$(p^2-1)/r = (p-1)h$, where h is a 1342-bit integer	$(p^6-1)/r = (p^3-1)(p+1)h$, where h is a small integer

Lemma 2 (**[21]**)

$$\frac{1}{x_P - x_Q} = \frac{x_P^2 + x_P x_Q + x_Q^2}{(y_P + y_Q)(y_P - y_Q)} \tag{5}$$

Lemma 1 and Lemma 2 derive the following theorem.

Theorem 1 (denominator elimination). *Let* $P = (x_P, y_P)$ *and* $Q = (x_Q, y_Q) \in \mathbb{G}_1$*, let* ι *be a distortion map defined as in Equation* (3)*, and let* $Q' = \iota(Q)$*.*

Then, without changing the output of the reduced Tate pairing, division by $x_P - x_{Q'}$ *can be replaced with multiplication by* $x_P^2 + x_P x_{Q'} + x_{Q'}^2$*.*

Proof. In Equation (5), $x_P - x_{Q'} \neq 0$ and $x_P^2 + x_P x_{Q'} + x_{Q'}^2 \neq 0$ for all possible $x_P, x_{Q'}$ in the Miller loop. Then, the denominator in Miller's algorithm is replaced as in Lemma 2, and we note that the denominator in Equation (5) is as follows:

$$\begin{aligned}(y_P + y_{Q'})(y_P - y_{Q'}) &= (y_P + u^3 y_Q)(y_P - u^3 y_Q) \\ &= y_P^2 - u^6 y_Q^2 \in \mathbb{F}_q.\end{aligned} \tag{6}$$

In the final exponentiation, the exponent can be decomposed as $(q^3-1)/r = (q-1)(p^2+p+1)h$, and resulting value of the final exponentiation with input the value of Equation (6) becomes one. □

Miller's Algorithm with Signed-Binary Representation. Miller's algorithm to compute $f_{p-1,P}\big(\iota(Q)\big)$ is defined on the standard binary representation, and it is also known as the double-and-add approach. It can be extended to the signed-binary representation, and it is then known as the double-and-add/subtract approach. If the number of non-zero components of the non-adjacent form (NAF) of $p-1$ is smaller than the Hamming weight of its binary representation, then the computation time can be improved.

Beuchat et al. [5] proposed using Miller's algorithm on the signed-binary representation of the Miller's algorithm on the Barreto–Naehrig curves; however, their algorithm does not work on the curves introduced in Section 2.2. As the definition of the Miller function implies,

$$(f_{-a,P}) = \left(\frac{1}{f_{a,P} \cdot v_{[a]P}}\right). \tag{7}$$

The algorithm presented by Beuchat et al. does not handle $v_{[a]P}$.

To extend the original Miller's algorithm for the signed-binary representation, we consider the subtraction of Miller's formula as follows:

$$\begin{aligned}(f_{a-1,P}) &= \left(f_{a,P} \cdot f_{-1,P} \cdot \frac{l_{[a]P,-P}}{v_{[a-1]P}}\right) \\ &= \left(f_{a,P} \cdot \frac{l_{[a]P,-P}}{v_{-P} \cdot v_{[a-1]P}}\right).\end{aligned} \tag{8}$$

Theorem 1 derives the following subtraction procedure:

$$f_{a-1,P}(Q) = \big(f_{a,P} \cdot l_{[a]P,-P} \cdot S_{[a-1]P} \cdot S_{-P}\big)(Q), \tag{9}$$

where S_V is a polynomial function on the elliptic curve defined as $S_V(Q) = x_V^2 + x_V x_Q + x_Q^2$. Equation (9) allows us to extend Miller's algorithm for the signed-binary representation with the elimination of the denominator to the curves introduced in Section 2.2.

3.3 Final Exponentiation

The output of Miller's algorithm is defined as an element of $\mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^r$. An exponentiation by $(q^3-1)/r$ is necessary in order to obtain a unique value of $\mu_r \in \mathbb{F}_{q^3}^*$, where μ_r is the r-th roots of unity. Typically, this exponentiation is called *final exponentiation*. This operation is computed in $\mathbb{F}_{q^3}$, and so it is one of the more expensive parts of a pairing computation.

From the definition of type 2 elliptic curves in Section 2.2, we can transform the exponent for the final exponentiation as follows:

$$\begin{aligned}(p^6-1)/r &= h(p^6-1)/\#E(\mathbb{F}_q) \\ &= h(p^6-1)/(p^2-p+1) \\ &= h(p^3-1)(p+1),\end{aligned} \tag{10}$$

where $h = \#E(\mathbb{F}_q)/r$. Hence, the final exponentiation is efficiently calculated by one inversion over $\mathbb{F}_{q^k}$, two multiplications over $\mathbb{F}_{q^k}$, two Frobenius maps, and an exponentiation by h. The most expensive part is the exponentiation by h. However, since we can choose an elliptic curve such that h is a very small integer in almost all cases, this operation can be done quickly. We call this faster version *fast final exponentiation.*

Algorithm 1. Reduced Ate pairing on $E/\mathbb{F}_{p^2}$

Input: T, P, Q: $T = t - 1 = 2^\ell + \sum_{i=0}^{\ell-1} s_i 2^i$, where $s_i \in \{0, \pm 1\}$, and $P, Q \in \mathbb{G}_1$.

Output: Reduced Ate pairing $f_{T,P}\big(\iota(Q)\big)^{(q^k-1)/r} \in \mathbb{G}_T$.

1: $Q' \leftarrow \iota(Q)$; // $6M_2$
2: $t_0 \leftarrow x_{Q'}^2$; // S_6
3: $t_1 \leftarrow S'_{-P}(Q', t_0)$; // $3M_2$
4: $V \leftarrow P$;
5: $f \leftarrow 1$;
6: **for** $i \leftarrow \ell - 1$ **down to** 0 **do**
7: $\quad (f, V) \leftarrow \Big(f^2 \cdot l_{V,V}(Q') \cdot S'_{[2]V}(Q', t_0), [2]V\Big)$;
8: $\quad$ **if** $s_i = 1$ **then**
9: $\quad\quad (f, V) \leftarrow \Big(f \cdot l_{V,P}(Q') \cdot S'_{V+P}(Q', t_0), V + P\Big)$;
10: $\quad$ **else if** $s_i = -1$ **then**
11: $\quad\quad (f, V) \leftarrow \Big(f \cdot l_{V,-P}(Q') \cdot S'_{V-P}(Q', t_0) \cdot t_1, V - P\Big)$;
12: $\quad$ **end if**;
13: **end for**;
14: $f \leftarrow f^{p^3} \cdot f^{-1}$; // $\pi_{p^3} + I_6 + M_6$
15: $f \leftarrow f \cdot f^p$; // $\pi_p + M_6$
16: $f \leftarrow f^h$; // Exp_h
17: **return** f;

3.4 Estimation of Computational Cost

In this section, we estimate computational cost of our algorithm performing the reduced Ate pairing. We will show the algorithm for the reduced Ate pairing on the elliptic curve $E/\mathbb{F}_{p^2}$ introduced in Section 2.2; see Algorithm 1. We note that $S'_V(Q, t) := x_V(x_V + x_Q) + t$ and $S'_P(Q, x_Q^2) = x_P^2 + x_P x_Q + x_Q^2 = S_P(Q)$ in Algorithm 1. In Algorithm 1, lines 1-13 and lines 14-16 correspond to the Miller's algorithm and the final exponentiation, respectively.

In this paper, we use the affine coordinate to implement the group operation of $\mathbb{G}_1$. The details of lines 7 and 9 in Algorithm 1 are described in Algorithm 2 and 3, respectively. The detail of line 11 in Algorithm 1 is easily derived by Algorithm 3, the difference is a multiplication by $t_1 \in \mathbb{F}_{p^6}$ and P is replaced by $-P$. We then show the computational cost of Algorithm 1 at Table 3. We note that the number of additions and subtractions are ignored and assume two Frobenius maps π_p and π_{p^3} over $\mathbb{F}_{p^6}$ have same computational cost in Table 3.

4 Experimental Implementation

In this section, we show the results from an experimental implementation of our proposed method. First, we show the environment in Table 4.

Algorithm 2. Doubling step of the reduced Ate pairing on $E/\mathbb{F}_{p^2}$ (at the line 7 in Algorithm 1)

Input: f, V, Q', t_0: $f \in \mathbb{F}_{p^6}$, $V \in E(\mathbb{F}_{p^2})$, $Q' = \iota(Q) \in E(\mathbb{F}_{p^6})$, and $t_0 = x_{Q'}^2 \in \mathbb{F}_{p^6}$. Note that Q' and t_0 are computed at lines 1 and 2, respectively, in Algorithm 1.

Output: $\left(f^2 \cdot l_{V,V}(Q') \cdot S'_{[2]V}(Q', t_0), [2]V\right) \in \mathbb{F}_{p^6} \times E(\mathbb{F}_{p^2})$.

$m \leftarrow 3x_V^2$; // S_2
$n \leftarrow 2y_V$;
$\lambda \leftarrow m/n$; // $I_2 + M_2$
$g \leftarrow y_{Q'} - y_V - \lambda(x_{Q'} - x_V)$; // $3M_2$
$f \leftarrow f^2$; // S_6
$f \leftarrow fg$; // M_6
$\lambda' \leftarrow \lambda^2$; // S_2
$x_{V'} \leftarrow \lambda' - 2x_V$;
$y_{V'} \leftarrow \lambda(x_V - x_{V'}) - y_V$; // M_2
$V' \leftarrow (x_{V'}, y_{V'})$;
$v \leftarrow x_{V'}(x_{V'} + x_{Q'}) + t_0$; // $3M_2$
$f \leftarrow fv$; // M_6
return (f, V');

4.1 Parameters

In our experiment, we generated two parameters, Curve 1 and Curve 2. In the class of our target elliptic curves described in Section 3, the characteristic p of a base field can be chosen as the pseudo-Mersenne prime ($p = 2^n - c$ and $\log_2 |c| \leq n/2$) [14]. Moreover, a tower field $\mathbb{F}_{q^3} = \mathbb{F}_{p^6}$ containing $\mathbb{G}_T$ can be defined by an irreducible binomial of $W^3 - \beta \in \mathbb{F}_q[W]$.

For our experiments, we generated two elliptic curves, Curves 1 and 2, as defined above. The length of their characteristics are $n = 367$ and 522, respectively. The parameter setting of Curve 1 is based on the least size of suggestions described in [10], and Curve 2 is based on Equation (4) with the assumption described in Section 3.1. Note that these two curves were generated randomly. We note that w_{NAF}^+ and w_{NAF}^- denote the numbers of 1 components and -1 components, respectively, in NAF encoding of $p - 1$.

Curve 1 (the sizes of p, r, and q^3 are 367 bits, 718 bits, and 2202 bits, respectively):

$E/\mathbb{F}_{p^2} : Y^2 = X^3 + \beta$,
$p = 2^{367} - c$, where $c = 6441$,
$w_{\mathrm{NAF}}^+ = 2$ and $w_{\mathrm{NAF}}^- = 5$,
$q = p^2$, and $t = p$,
$r = \#E(\mathbb{F}_{p^2})/h = (p^2 - p + 1)/h$, where $h = 110937$,
$\mathbb{F}_q = \mathbb{F}_{p^2} := \mathbb{F}_p[V]/(V^2 - \alpha)$, where α is
2674245158309532807325674069454972905651716022739308862
8789216699870470962170359843916380575606965024714761972 2,

Algorithm 3. Addition step of the reduced Ate pairing on $E/\mathbb{F}_{p^2}$ (at the line 9 in Algorithm 1)

Input: f, V, P, Q', t_0: $f \in \mathbb{F}_{p^6}$, $V, P \in E(\mathbb{F}_{p^2})$, $Q' = \iota(Q) \in E(\mathbb{F}_{p^6})$, and $t_0 = x_{Q'}^2 \in \mathbb{F}_{p^6}$. Note that Q' and t_0 are computed at lines 1 and 2, respectively, in Algorithm 1, and P is a one of inputs of Algorithm 1.

Output: $\left(f \cdot l_{V,P}(Q') \cdot S'_{V+P}(Q', t_0), V+P\right) \in \mathbb{F}_{p^6} \times E(\mathbb{F}_{p^2})$.

```
m ← (y_P − y_V);
n ← (x_P − x_V);
λ ← m/n;                                  // I_2 + M_2
g ← y_Q' − y_V − λ(x_Q' − x_V);           // 3M_2
f ← fg;                                   // M_6
λ' ← λ^2;                                 // S_2
x_V' ← λ' − x_V − x_P;
y_V' ← λ(x_P − x_V') − y_P;               // M_2
V' ← (x_V', y_V');
v ← x_V'(x_V' + x_Q') + t_0;              // 3M_2
f ← fv;                                   // M_6
return f;
```

$\mathbb{F}_{q^3} := \mathbb{F}_q[W]/(W^3 - \beta)$, where β is
2528964409087109586735370294508436849691017597126041538
6550722365991977183853605246047387340418369769543384088 2$V+$
2058841674231253025987668201602254081903020106910309523
5245994850270079586875401480868413416144232203483283360 6,

and distortion map is $\iota : (x, y) \mapsto (u^2 x^p, u^3 y^p)$ where u is
9914330293514571516462572069203799078797519193318327503
588178011015271568479578245047076030877204117816758990 0W.

Curve 2 (the sizes of p, r, and q^3 are 522 bits, 1038 bits, and 3132 bits, respectively):

$E/\mathbb{F}_{p^2} : Y^2 = X^3 + \beta$,
$p = 2^{522} - c$, where $c = 29087$,
$w_{\mathrm{NAF}}^{+} = 3$ and $w_{\mathrm{NAF}}^{-} = 3$,
$q = p^2$, and $t = p$,
$r = \#E(\mathbb{F}_{p^2})/h = (p^2 - p + 1)/h$, where $h = 93$,
$\mathbb{F}_q = \mathbb{F}_{p^2} := \mathbb{F}_p[V]/(V^2 - \alpha)$, where α is
2583834559853811459432166124427683502167391574858989654
5214442003228999316236159397036115676140967350980743986
5701651847527304215126376997355248221059380187 9,

$\mathbb{F}_{q^3} := \mathbb{F}_q[W]/(W^3 - \beta)$, where β is
5540496805234858649054077930128599436615709048884769387
6603968620597741702054737057676736328177323553483431937
910113639593360925402578513145105442802971714 01$V+$
5729611582621237878678119907084390704267702847871726214

Table 3. Computational cost of our algorithm, where M_k, S_k, and I_k denote the multiplication, squaring, and inversion over $\mathbb{F}_{p^k}$, π denotes Frobenius map over $\mathbb{F}_{p^6}$, $p = 2^\ell - c$ and it is a prime number, w^+_{NAF} denotes the number of 1 components and w^-_{NAF} denotes the number of -1 components in NAF encoding of $p-1$, and Exp_h denotes exponentiation by h over $\mathbb{F}_{p^6}$

Part of Algorithm 1	Computational Cost
$l_{V,V}(Q')$ and $[2]V$ in line 7	$5M_2 + 2S_2 + I_2$
$S'_{[2]V}(Q', t_0)$ in line 7	$3M_2$
$l_{V,\pm P}(Q')$ and $V \pm P$ in lines 9 and 11	$5M_2 + S_2 + I_2$
$S'_{V\pm P}(Q', t_0)$ in lines 9 and 11	$3M_2$
Line 7	$8M_2 + 2S_2 + I_2 + 2M_6 + S_6$
Line 9	$8M_2 + S_2 + I_2 + 2M_6$
Line 11	$8M_2 + S_2 + I_2 + 3M_6$
Miller's algorithm (lines 1-13)	$9M_2 + S_6 + (8M_2 + 2S_2 + I_2 + 2M_6 + S_6)\ell$ $+(w^+_{\mathrm{NAF}} + w^-_{\mathrm{NAF}})(8M_2 + S_2 + I_2 + 2M_6)$ $+w^-_{\mathrm{NAF}} M_6$
Final exponentiation (lines 14-16)	$2M_6 + 2\pi + I_6 + \mathrm{Exp}_h$

Table 4. Experimental environment

	Environment
OS	Linux 3.5.0-37 (Ubuntu 12.04.2 LTS)
CPU	Core i7-4770 (3.4 GHz)
Memory	32 GB
Language	Magma version 2.19-8 [8]

3429775029040573419091832483405499148515483815456512633
32728406562347176934945350917989445472195196929, and
distortion map is $\iota : (x, y) \mapsto (u^2 x^p, u^3 y^p)$ where u is
1810455431901709610502451144154632135017017586718473396
1873794180953915455128081305700723007474055399866147491
22579794730213310737853381173392719765819055455W

4.2 Performance of the Proposed Method

We computed the pairings and compared the running time of the Tate pairing and the Ate pairing with the signed-binary approach on $E/\mathbb{F}_q$. The parameters used in Miller's algorithm were r and $p-1$, and these were represented in NAF encoding. We ran the pairings 1000 times and computed the averages of Miller's algorithm for the Tate pairing, the Ate pairing, and the fast final exponentiation. Table 5 shows these averages. It is clear that the Ate pairing computation on $E/\mathbb{F}_q$ is efficiently computable. We note that our experimental implementation is written in Magma [8], we did not implement efficient arithmetic based on the pseudo-Mersenne prime, and generated curves are randomly generated. Thus, there is room for further optimization.

Table 5. Running time of pairing computations (unit: milliseconds)

	Curve 1	Curve 2
$f_{r,P}\big(\iota(Q)\big)$ with NAF	88.28	157.87
$f_{T,P}\big(\iota(Q)\big)$ with NAF	34.38	62.06
Fast Final Exp.	0.25	0.21
Reduced Tate	88.53	158.08
Reduced Ate	34.63	62.27

5 Conclusion

In the present paper, we proposed a method to construct symmetric pairings by applying the Ate pairing to supersingular elliptic curves over finite fields with large characteristics and embedding degree three. We also proposed an efficient algorithm of the Ate pairing on these curves. We then generated several curves in order to show the existence of curves that our method is applicable to, and implemented experimental programs of our method and demonstrated that it is efficiently computable.

Acknowledgements. The authors would like to thank Goichiro Hanaoka and Takahiro Matsuda for the valuable comments. We gratefully thank the members of Shin-Akarui-Angou-Benkyou-Kai for the valuable discussion and comments. We also thank the anonymous reviewers of Pairing 2013 for the valuable comments.

References

1. Adj, G., Menezes, A., Oliveira, T., Rodríguez-Henríquez, F.: Weakness of $\mathbb{F}_{3^{6\cdot 509}}$ for discrete logarithm cryptography. In: Cao, Z., Zhang, F. (eds.) Pairing 2013. LNCS, vol. 8365, pp. 19–43. Springer, Heidelberg (2014)
2. Adleman, L.M.: The function field sieve. In: Huang, M.-D.A., Adleman, L.M. (eds.) ANTS 1994. LNCS, vol. 877, pp. 108–121. Springer, Heidelberg (1994)
3. Barreto, P.S.L.M., Galbraith, S.D., ÓhÉigeartaigh, C., Scott, M.: Efficient pairing computation on supersingular abelian varieties. Des. Codes Cryptography 42(3), 239–271 (2007)
4. Barreto, P.S.L.M., Kim, H.Y., Lynn, B., Scott, M.: Efficient algorithms for pairing-based cryptosystems. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 354–368. Springer, Heidelberg (2002)
5. Beuchat, J.-L., González-Díaz, J.E., Mitsunari, S., Okamoto, E., Rodríguez-Henríquez, F., Teruya, T.: High-speed software implementation of the optimal ate pairing over barreto–naehrig curves. In: Joye, M., Miyaji, A., Otsuka, A. (eds.) Pairing 2010. LNCS, vol. 6487, pp. 21–39. Springer, Heidelberg (2010)
6. Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. In: [19], pp. 213–229
7. Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. SIAM J. Comput. 32(3), 586–615 (2003)

8. Bosma, W., Cannon, J., Playoust, C.: The Magma algebra system. I. The user language. J. Symbolic Comput. 24(3-4), 235–265 (1997); Computational algebra and number theory, London (1993)
9. Chatterjee, S., Hankerson, D., Knapp, E., Menezes, A.: Comparing two pairing-based aggregate signature schemes. Des. Codes Cryptography 55(2-3), 141–167 (2010)
10. Freeman, D., Scott, M., Teske, E.: A taxonomy of pairing-friendly elliptic curves. J. Cryptology 23(2), 224–280 (2010)
11. Galbraith, S.D., Paterson, K.G., Smart, N.P.: Pairings for cryptographers. Discrete Applied Mathematics 156(16), 3113–3121 (2008)
12. Gallant, R., Lambert, R., Vanstone, S.: Faster point multiplication on elliptic curves with efficient endomorphisms. In: [19], pp. 190–200 (2001)
13. Gaudry, P., Thomé, E., Thériault, N., Diem, C.: A double large prime variation for small genus hyperelliptic index calculus. Mathematics of Computation 76, 475–492 (2004)
14. Hankerson, D., Menezes, A.J., Vanstone, S.: Guide to Elliptic Curve Cryptography. Springer-Verlag New York, Inc., Secaucus (2004)
15. Hayashi, T., Shimoyama, T., Shinohara, N., Takagi, T.: Breaking pairing-based cryptosystems using η_T pairing over $GF(3^{97})$. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 43–60. Springer, Heidelberg (2012)
16. Hess, F., Smart, N.P., Vercauteren, F.: The eta pairing revisited. IEEE Transactions on Information Theory 52(10), 4595–4602 (2006)
17. Joux, A.: Discrete logarithms in $\mathrm{GF}(2^{6168})$ [$= \mathrm{GF}((2^{257})^{24})$]. NMBRTHRY list (May 21, 2013), `https://listserv.nodak.edu/cgi-bin/wa.exe?A2=NMBRTHRY;49bb494e.1305`
18. Joux, A., Pierrot, C.: The special number field sieve in $\mathbb{F}_{p^n}$, application to pairing-friendly constructions. In: Cao, Z., Zhang, F. (eds.) Pairing 2013. LNCS, vol. 8365, pp. 45–61. Springer, Heidelberg (2014)
19. Kilian, J. (ed.): CRYPTO 2001. LNCS, vol. 2139. Springer, Heidelberg (2001)
20. Lee, E., Lee, H.S., Park, C.M.: Efficient and generalized pairing computation on abelian varieties. IEEE Transactions on Information Theory 55(4), 1793–1803 (2009)
21. Lin, X., Zhao, C., Zhang, F., Wang, Y.: Computing the ate pairing on elliptic curves with embedding degree $k = 9$. IEICE Transactions 91-A(9), 2387–2393 (2008)
22. Miller, V.S.: The Weil pairing, and its efficient calculation. J. Cryptology 17(4), 235–261 (2004)
23. Momose, F., Chao, J.: Scholten forms and elliptic/hyperelliptic curves with weak Weil restrictions. Cryptology ePrint Archive, Report 2005/277 (2005), `http://eprint.iacr.org/2005/277`
24. Nagao, K.: Improvement of Thériault algorithm of index calculus for Jacobian of hyperelliptic curves of small genus. Cryptology ePrint Archive, Report 2004/161 (2004), `http://eprint.iacr.org/2004/161`
25. Ogura, N., Uchiyama, S., Kanayama, N., Okamoato, E.: A note on the pairing computation using normalized Miller functions. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E95-A(1), 196–203 (2012)
26. Sakai, R., Ohgishi, K., Kasahara, M.: Cryptosystems based on pairing. In: 2000 Symposium on Cryptography and Information Security (SCIS 2000), pp. 26–28 (January 2000) C20

27. Scholten, J.: Weil restriction of an elliptic curve over a quadratic extension (2003) (preprint), `http://www.esat.kuleuven.ac.be/~jscholte/weilres.ps`
28. Vercauteren, F.: Optimal pairings. IEEE Transactions on Information Theory 56(1), 455–461 (2010)
29. Verheul, E.R.: Evidence that XTR is more secure than supersingular elliptic curve cryptosystems. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 195–210. Springer, Heidelberg (2001)
30. Verheul, E.R.: Evidence that XTR is more secure than supersingular elliptic curve cryptosystems. J. Cryptology 17(4), 277–296 (2004)
31. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
32. Zhao, C., Zhang, F., Huang, J.: A note on the ate pairing. Int. J. Inf. Sec. 7(6), 379–382 (2008)