Implementing Cryptographic Pairings Over Curves of Embedding Degrees 8 and 10

Christine Abegail Antonio, Satoru Tanaka, and Ken Nakamula

Department of Mathematics and Information Science
Tokyo Metropolitan University
Minami-Osawa, Hachioji-shi
Tokyo, Japan
abby@tnt.math.metro-u.ac.jp, satoru@tnt.math.metro-u.ac.jp,
nakamula@tnt.math.metro-u.ac.jp

Abstract. In this paper, we will describe efficient implementations of the Tate and Ate pairings over ordinary elliptic curves of embedding degrees 8 and 10. We will discuss the possible curve-dependent optimizations that can be applied to evaluate the pairings. We pay particular attention to the use of elliptic curve twists and the denominator elimination method to make computations more efficient. Our main goal is to draw together the best possible optimizations that can be used to efficiently evaluate the Tate and the Ate pairings in both curves and to give timings and appropriate interpretation on the rate of change on the running time of our programs for both curves. To come up with an adequate conclusion, we will compare the performance of the curves we chose to an already experimented curve of embedding degree 12.

key words and phrases. bilinear pairings, cryptography, pairing-friendly curves

1 Introduction

The efficient implementation of bilinear pairings has been a significant topic of research because they are being used in recently developed cryptographic protocols. The standard method of computing these pairings is based on the algorithm presented by Miller in [9]. A lot of pairing researches are based on improving this standard method. One optimization that can be done is to improve the Miller loop such as the denominator elimination method which is discussed in [1]. Another improvement is to use pairing-friendly curves with parameters which are appropriate to use in pairing-based cryptography. Supersingular and ordinary elliptic curves are both suitable curves for this kind of cryptography. Many papers have been published on the construction of these kinds of curves and a survey on this was written by Freeman, Scott and Teske [5]. In [7], they suggested that bilinear pairing computations can be improved by choosing suitable fields where the field arithmetic can be made relatively faster.

In [3], J. Devegili, M.Scott and R. Dahab described an efficient implementation of both the Tate and the Ate pairings using curves of embedding degree 12, which are pairing-friendly elliptic curves of embedding degree 12 with prime order. In this paper, we will do a similar experiment using the family of curves generated by Freeman[4] and the family of curves generated by Tanaka [13], both of which are pairing friendly elliptic curves with embedding degrees 10 and 8, respectively. To adequately compare the efficiency of the optimizations we applied, we will include in our implementations the curve of embedding degree 12 used in [3]. We chose these curves because, as we write this paper, there are still no published researches on the efficient implementation of bilinear pairings using curves with those embedding degrees. Furthermore, explicit methods on how to generate such curves are discussed in their papers and good numerical examples are already provided which make them suitable to use in our implementations.

Unlike supersingular elliptic curves, ordinary curves do not have distortion maps that can make computations much faster. However, for ordinary curves of even embedding degree, we can use the twist of an elliptic curve to efficiently evaluate bilinear pairings. Since curves of embedding degrees 8 and 12 fall under this category, we will use this idea in our experiment and we will show that, indeed, the use of twists make the pairing evaluations much faster. We will also use the denominator elimination method proposed in [1]. These are the two main optimizations that we will use in this experiment and we will show that they reduce the running time of the algorithms to compute the pairings.

For comparison, we will use the same curve they used in [3]. Preliminary ideas on the turnout of the experiment can be expected and they will be discusssed in Section 4.5. Our main interest is to compare the rate of the speed up of the algorithms using the two main optimizations discussed above and how they perform on the three curves that we will use. The denominator elimination method reduced the running time about 20 percent, in all the cases. On the other hand, the effect of the use of twist varied between the three curves, depending on the parameters.

This paper is organized as follows. Section 2 gives the notations that we will use in this paper, a brief mathematical background of bilinear pairings, in particular, the Tate and Ate pairings, and algorithms to compute these pairings. In Section 3 we will introduce some optimizations to make the algorithms efficient. Section 4 will give details on the curves that we will use in this experiment and how to implement the Tate and the Ate pairings on these curves. In Section 5 we will give our numerical results. Finally, we will discuss the conclusions that we will draw from our implementations in Section 6.

2 Notations and Algorithms

Let E be an elliptic curve over a finite field $\mathbb{F}_q$. A *divisor* D is defined as $D = \sum_{P \in E} m_P(P)$, $m_P \in \mathbb{Z}$ and $m_P = 0$ for almost all points P. These divisors form an additive group $\mathbf{D}$(E). We say that a divisor D is principal if there exists a

rational function f such that m_P gives the order of vanishing of f at P. These principal divisors are of degree 0 and form a subgroup of the group $\mathbf{D}^0(\mathrm{E})$ of degree zero divisors. The support of a divisor $D = \sum_{P \in E} n_P(P)$, denoted by $\mathrm{supp}(D)$, is the set of points P with $n_P \neq 0$. See [6] for a detailed discussion on divisors.

Now, let r be the largest prime dividing $|E(\mathbb{F}_q)|$ such that r is relatively prime to the characteristic of the field $\mathbb{F}_q$. We denote by k the *embedding degree*, namely, the smallest positive integer such that $r|q^k - 1$. A pairing is a function which maps bilinearly, a pair of elliptic curve points P, Q to an element in the finite multiplicative group ${\mathbb{F}_{q^k}}^*$. The most commonly used bilinear pairing is called the Tate pairing.

We denote by $E(\mathbb{F}_{q^k})[r]$ the group of r-torsion points of E. For a pair of elliptic curve points $P \in E(\mathbb{F}_{q^k})[r]$ and $Q \in E(\mathbb{F}_{q^k})$, we let the divisor $(f_P) = r(P) - r(P_\infty)$ where P_∞ denotes the point at infinity on E. This divisor is principal since P is an r-torsion point. Choose another divisor $D_Q = (Q) - (P_\infty)$ with support disjoint from the support of (f_P). The Tate pairing is defined as

$$e(*,*)_r : E(\mathbb{F}_{q^k})[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) \rightarrow {\mathbb{F}_{q^k}}^*/{\mathbb{F}_{q^k}}^{*r}$$

by $e(P,Q)_r = f_P(D_Q)$. This pairing is well-defined, bilinear and non-degenerate. Note that the Tate pairing evaluates as an element of one of the cosets of ${\mathbb{F}_{q^k}}^*/{\mathbb{F}_{q^k}}^{*r}$. To produce a unique value, we raise the output to the power $(q^k - 1)/r$. This process is called the *final exponentiation*.

In our algorithms to compute the Tate pairing, we will use the functions $l_{A,B}$ and v_{A+B}. These are just the lines computed when evaluating the elliptic curve point addition $A + B = C$. The values for these functions are solved using the formulas

$$l_{A,B}(Q) = (y_Q - y_A) - \lambda(x_Q - x_A)$$

and

$$v_C(Q) = (x_Q - x_C)$$

where A=(x_A, y_A), C=(x_C, y_C), Q=(x_Q, y_Q), and λ is the slope of the line through A and B. Furthermore, let

$$r = (1, r_{\lfloor \log_2 r \rfloor - 1}, ..., r_0)_2$$

be the binary expansion of r. Given below is a standard algorithm to compute the Tate pairing.

Algorithm 1. Standard Algorithm for Computing Tate Pairing.

INPUT: $P \in E(\mathbb{F}_q)[r]$, $P \neq P_\infty$, $Q \in E(\mathbb{F}_{q^k})$.

OUTPUT: $e(P,Q)_r^{(q^k-1)/r}$.

1: $T \leftarrow P$, $f \leftarrow 1$.
2: for $i \leftarrow \lfloor \log_2(r) \rfloor - 1$ down to 0 do
3: $\quad f = f^2 \cdot l_{T,T}(Q)$
4: $\quad T = 2T$
5: $\quad g = g^2 \cdot v_T(Q)$
6: $\quad$ if $r_i = 1$ then
7: $\quad\quad f = f \cdot l_{T,P}(Q)$
8: $\quad\quad T = T + P$
9: $\quad\quad g = g \cdot v_T(Q)$
10: $\quad$ end if
11: end for
12: $w \leftarrow (f/g)^{(q^k-1)/r}$
13: Return w.

Another kind of pairing that we will use in our implementations is a variant of the Tate pairing called Ate pairing introduced in [8]. If we denote by t the trace of Frobenius of the curve E, then the Ate pairing is defined as

$$a(*,*)_r : E(\mathbb{F}_{q^k})[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) \rightarrow {\mathbb{F}_{q^k}}^*/{\mathbb{F}_{q^k}}^{*r}$$

by $a(Q,P)_{t-1} = f_Q(D_P)$. This pairing is well-defined, bilinear and non-degenerate. Similar to the Tate pairing, a final exponentiation is needed to obtain a unique value. The only difference is that we interchange the fields from which we take the points P, Q and instead of using r to control the Miller loop, we use $s = t-1$. For simplicity, we assume $t > 1$ and we will compute on such curves. Below is a standard algorithm to compute the Ate pairing. Note that

$$s = (1, s_{\lfloor \log_2 s \rfloor - 1}, ..., s_0)_2.$$

Algorithm 2. Standard Algorithm for Computing Ate Pairing.

INPUT: $Q \in E(\mathbb{F}_q)[r]$, $P \neq P_\infty$, $P \in E(\mathbb{F}_{q^k})[r]$.

OUTPUT: $a(P,Q)_r^{(q^k-1)/r}$.

1: $T \leftarrow P$, $f \leftarrow 1$.
2: for $i \leftarrow \lfloor \log_2(s) \rfloor - 1$ down to 0 do
3: $\quad f = f^2 \cdot l_{T,T}(Q)$
4: $\quad T = 2T$
5: $\quad g = g^2 \cdot v_T(Q)$
6: $\quad$ if $r_i = 1$ then
7: $\quad\quad f = f \cdot l_{T,P}(Q)$
8: $\quad\quad T = T + P$
9: $\quad\quad g = g \cdot v_T(Q)$
10: $\quad$ end if
11: end for
12: $w \leftarrow (f/g)^{(q^k-1)/r}$
13: Return w.

3 OPTIMIZATIONS

The standard algorithm used to compute bilinear pairings is based on Miller's algorithm introduced in [9]. Many researches discuss the optimizations that can be applied to the standard algorithm, see for example [11] about a survey on the efficient implementation of bilinear pairings. In this section, we will discuss the optimizations we used to efficiently evaluate the pairings.

1. Since we are using curves of even embedding degrees in our experiment, then $k = 2k'$, so we may assume that the extension field $\mathbb{F}_{q^k}$ is built as a quadratic extension over $\mathbb{F}_{q^{k'}}$. Using this fact, we can use the denominator elimination method, namely, there is no need to compute for the vertical line $v_{A,B}(Q)$ in Algorithms 1 and 2. See [11] and [1] for details on this optimization.
2. We say that E admits a twist of degree d if there exists an E' defined over $\mathbb{F}_q$ and an isomorphism $\psi : E' \rightarrow E$ defined over $\mathbb{F}_{q^{k/d}}$. The use of the twist of ordinary curves appear to speed up pairing evaluations because for the case of the Tate pairing, instead of taking $Q \in E(\mathbb{F}_{q^k})$, we can choose it to come from $E'(\mathbb{F}_{q^{k/d}})$. Similarly for the case of the Ate pairing, we can take P from $E'(\mathbb{F}_{q^{k/d}})$. These choices in our input reduce the running time of the algorithms because we can avoid full $E(\mathbb{F}_{q^k})$ arithmetic to compute $l_{A,B}$. Therefore, in our experiment, we will compare its performance with and without using twists. Details on the appropriate twist to use on both curves of embedding degrees 8 and 10 will be discussed in Section 4.

Below are the optimized algorithms to compute the Tate and Ate pairings.

Algorithm 3. Optimized Algorithm for Computing Tate Pairing.

INPUT: $P \in E(\mathbb{F}_q)[r]$, $P \neq P_\infty$, $Q \in E'(\mathbb{F}_{q^{k/d}})$.
OUTPUT: $\langle P, Q \rangle_r^{(q^k-1)/r}$
1: $T \leftarrow P$, $f \leftarrow 1$.
2: for $i \leftarrow \lfloor \log_2(r) \rfloor - 1$ down to 0 do
3: $f = f^2 \cdot l_{T,T}(Q)$
4: $T = 2T$
5: if $r_i = 1$ then
6: $f = f \cdot l_{T,P}(Q)$
7: $T = T + P$
8: end if
9: end for
10: $w \leftarrow (f)^{(q^k-1)/r}$
11: Return w.

Algorithm 4. Optimized Algorithm for Computing Ate Pairing.

INPUT: $Q \in E(\mathbb{F}_q)[r]$, $P \neq P_\infty$, $P \in E'(\mathbb{F}_{q^{k/d}})[r]$.
OUTPUT: $\langle P, Q \rangle_r^{(q^k-1)/r}$
1: $T \leftarrow P$, $f \leftarrow 1$.
2: for $i \leftarrow \lfloor \log_2(s) \rfloor - 1$ down to 0 do
3: $f = f^2 \cdot l_{T,T}(Q)$
4: $T = 2T$
5: if $s_i = 1$ then
6: $f = f \cdot l_{T,P}(Q)$
7: $T = T + P$
8: end if
9: end for
10: $w \leftarrow (f)^{(q^k-1)/r}$
11: Return w.

4 Pairing-Friendly Curves

In this section, we will discuss the pairing-friendly curves that we will use in our implementations. After giving details on these curve, we will provide a table which contains the difference in the parameters of the sample curves that we used in this experiment.

4.1 Curves with Embedding Degree 10

Freeman, in [5] proposed a way to generate a whole family of pairing-friendly ordinary elliptic curves of the form $E_1 : Y^2 = X^3 + aX + b$, $b \neq 0$ of embedding degree 10 with prime orders. This is an addition to the work of Miyaji, Nakabayashi and Takano[10] who gave a complete characterization of ordinary elliptic curves with embedding degrees 3, 4 and 6 of prime order, and Baretto–Naehrig[2] who provided a method to produce curves of prime order with embedding degree 12.

We can generate a 'pairing-friendly' elliptic curve of embedding degree 10 with prime order by using the following parameters.

$$\begin{aligned} t(x) &= 10x^2 + 5x + 3 \\ r(x) &= 25x^4 + 25x^3 + 15x^2 + 5x + 1 \\ q(x) &= 25x^4 + 25x^3 + 25x^2 + 10x + 3 \end{aligned}$$

where $t(x)$ is the trace of Frobenius of the curve, $r(x)$ is the large prime which divides the order of the group, and at the same time, the number of points of the elliptic curve E_1 , and $q(x)$ is the characteristic of the finite field. Since this curve is of embedding degree 10, we are computing pairings over points in the field $\mathbb{F}_{q^{10}}$. To make pairing implementations more efficient, we will use the twist of these curves. Let $\zeta \in \mathbb{F}_{q^5}$ such that $Z^2 - \zeta$ is irreducible over $\mathbb{F}_{q^5}[Z]$ whenever $q \equiv 1 \pmod 2$. Then there exists a curve $E_2/\mathbb{F}_{q^5} : Y'^2 = X'^3 + \frac{a}{\zeta^2}X' + \frac{b}{\zeta^3}$ which is a quadratic twist of $E_1/\mathbb{F}_{q^{10}}$. So for curves of embedding degree 10, we can compress some of the points from $E_1(\mathbb{F}_{q^{10}})$ to points in the quadratic twist $E_2(\mathbb{F}_{q^5})$.

Let $V \in \mathbb{F}_{q^{10}}$ be a root of $Z^2 - \zeta$. The we can construct a homomorphism

$$\begin{aligned} \psi : E_2(\mathbb{F}_{q^5}) &\to E_1(\mathbb{F}_{q^{10}}) \\ (X', Y') &\mapsto (V^2X, V^3Y) \end{aligned}$$

which maps points on the quadratic twist $E_2(\mathbb{F}_{q^5})$ to the points of the original curve $E_1(\mathbb{F}_{q^{10}})$. This map will make pairing evaluations faster which we will show later in our results.

In our implementations, we will use a 234-bit curve (published example [5])

$$E : Y^2 = X^3 + AX + B$$

with the following parameters.

$$\begin{aligned} r &= 182116508039694720644932643473759500459342546966570904207262300432038 03 \\ q &= 182116508039694720644932643473759500459342546966570904207262300432038 03 \\ A &= -3 \\ B &= 157486680949134011847779644735228590869008312749229489733206849959032 75 \end{aligned}$$

Below is the table for the construction of the extension fields that we used for this curve.

4.2 Curves with Embedding Degree 8

In [13, 14], Tanaka developed an algorithm to generate pairing-friendly elliptic curves of the form $E_3 : Y^2 = X^3 + aX$, with embedding degree 8 over finite

Table 1. Extension Field Construction, k=10

Extension	Construction	Representation
$\mathbb{F}_{q^5}$	$\mathbb{F}_q[X]/(X^5 - X - 1)$	$a = a_0 + a_1X + a_2X^2 + a_3X^3 + a_4X^4$
$\mathbb{F}_{q^{10}}$	$\mathbb{F}_{q^5}[Y]/(Y^2 - \alpha)$	$a = a_0 + a_1Y$

prime fields by improving the method of Brezing and Weng (see [4] and [14] for further details). The curve can be generated using the following parameters.

$$\begin{aligned} t(x) &= -8x^2 - 108x^2 - 54x - 8 \\ r(x) &= 82x^4 + 108x^3 + 54x^2 + 12x + 1 \\ q(x) &= 379906x^6 + 799008x^5 + 705346x^4 + 333614x^3 + 88945x^2 \\ &\quad +12636x + 745 \\ n(x) &= q(x) + 1 - t(x), \end{aligned}$$

where the notations are the same as the ones we used for curves of embedding degree 10, except that $r(x)$ is the large prime divisor of $n(x)$ which is the number of points of the elliptic curve E.

As in the case of curves of embedding degree 10, we can use twists to make implementations more efficient. These curves are of embedding degree 8, so we are evaluating pairings over points in $E_3(\mathbb{F}_{q^8})$. Whenever $q \equiv 1 \pmod 4$, let $\mu \in \mathbb{F}_{q^2}$ such that $W^4 - \mu$ is irreducible over $\mathbb{F}_{q^2}[W]$. Then there exists $E_4/\mathbb{F}_{q^2} : Y'^2 = X'^3 + \frac{a}{\mu}X'$ that is a quartic twist of $E_3/\mathbb{F}_q$. Therefore, for the case of the curves of embedding degree 8, we can compress some of the points from $E_3(\mathbb{F}_{q^8})$ to points in the quartic twist $E_4(\mathbb{F}_q^2)$.

Similarly, we can constuct a homomorphism which maps points on the quartic twist $E_4(\mathbb{F}_{q^2})$ to the points of the original curve $E_3(\mathbb{F}_{q^8})$. Let $U \in \mathbb{F}_{q^8}$ be a root of $W^2 - \mu$. Then the following is the homomorphism map for the case of the curves of embedding degree 8.

$$\begin{aligned} \psi : E_4(\mathbb{F}_{q^2}) &\rightarrow E_3(\mathbb{F}_{q^8}) \\ (X', Y') &\mapsto (U^2X, U^3Y) \end{aligned}$$

This map will make pairing evaluations more efficient which will be shown later in our results.

For our implementation, we will use the elliptic curve with the following parameters. Note that the large prime divisor r of the group order is 224 bits. It is easy to verify that this curve is pairing-friendly once the parameters are given numerically as follows.

$$E : Y^2 \equiv X^3 + aX \pmod q \quad (a \neq 0)$$

For $x = -72057594037930756$ ($\log_2(-x) \approx 56.0$), we generated the curve with the following parameters.

$$
\begin{aligned}
q &= 5318077912637504134292767901251647400395578540 \\
&\quad 3827730100050941212371435046023372666628598916 \\
&\quad 049952969199369 \\
r &= 2210715626706698491377041180063927762099958931 \\
&\quad 72260380547480590742 4817 \\
t &= 3067984237085391549834039420816298507616442947 \\
&\quad 7994640 \\
n &= 5318077912637504134292767901251647400395578540 \\
&\quad 3827730069371098841517519547682978458465613839 \\
&\quad 885523491204730 \\
a &= 1/3.
\end{aligned}
$$

For this curve, we have $\lg r \approx 230.4$ and $\lg q \approx 354.5$. It is easy to verify that this curve is pairing-friendly once the parameters are given numerically as above.

Below is the table for the construction of the extension fields that we used for this curve.

Table 2. Extension Field Construction, k=8

Extension	Construction	Representation
$\mathbb{F}_{q^2}$	$\mathbb{F}_q[X]/(X^2-\alpha)$	$a = a_0 + a_1 X$
$\mathbb{F}_{q^4}$	$\mathbb{F}_{q^2}[Y]/(Y^2-\beta)$	$a = a_0 + a_1 Y$
$\mathbb{F}_{q^8}$	$\mathbb{F}_{q^4}[Z]/(Z^2-\gamma)$	$a = a_0 + a_1 Z$

4.3 Curves with Embedding Degree 12

In [3], they used Barreto-Naehrig curves in their implementations. These are pairing-friendly curves of embedding degree 12 with prime order. The equation of the curve is given by $y^2 = x^3 + b$, $b \neq 0$. The parameters of the curve are as follows, where the notations are the same as that of curves of embedding degree 10.

$$
\begin{aligned}
t(x) &= 6x^2 + 1 \\
r(x) &= 36x^4 - 36x^3 + 18x^2 - 6x + 1 \\
q(x) &= 36x^4 - 36x^3 + 24x^2 - 6x + 1
\end{aligned}
$$

Similar to both curves discussed above, we can use twists to make implementations more efficient. Since these curves are of embedding degree 12 we are evaluating pairings over points in $E_5(\mathbb{F}_{q^{12}})$. Whenever $q \equiv 1 \pmod 6$, let $\lambda \in \mathbb{F}_{q^2}$ such that $W^6 - \lambda$ is irreducible over $\mathbb{F}_{q^2}[W]$. Then there exists $E_6/\mathbb{F}_{q^2} : Y'^2 = X'^3 + \frac{b}{\lambda}$

that is a sextic twist of $E_5/\mathbb{F}_q$. Therefore, for the case of BN curves, we can compress some of the points from $E_5(\mathbb{F}_{q^{12}})$ to points in the quadratic twist $E_6(\mathbb{F}_q^2)$.

Similarly, we can constuct a homomorphism which maps points on the sextic twist $E_6(\mathbb{F}_{q^2})$ to the points of the original curve $E_5(\mathbb{F}_{q^{12}})$. Let $A \in \mathbb{F}_{q^{12}}$ be a root of $A^6 - \lambda$. Then the following is the homomorphism map for BN curves.

$$\begin{aligned} \psi : E_6(\mathbb{F}_{q^2}) &\to E_5(\mathbb{F}_{q^{12}}) \\ (X', Y') &\mapsto (A^2X, A^3Y) \end{aligned}$$

In our experiment, we will use the same curve they used in [3] and it is given numerically as follows.

$$E : Y^2 = X^3 + B$$

$$\begin{aligned} q &= 824340166543006797212173535031900388365717818113862289211\\ &\quad 673224128190\\ r &= 824340166543006797212173535031900388362846685642966864301145100525564\\ t &= 28711324708954249105281236026262811941 5\\ B &= 3 \end{aligned}$$

Table 3. Extension Field Construction, k=12

Extension	Construction	Representation
$\mathbb{F}_{q^2}$	$\mathbb{F}_q[X]/(X^2-\alpha)$	$a = a_0 + a_1X$
$\mathbb{F}_{q^6}$	$\mathbb{F}_{q^2}[Y]/(Y^3-\beta)$	$a = a_0 + a_1Y + a_2Y^2$
$\mathbb{F}_{q^{12}}$	$\mathbb{F}_{q^6}[Z]/(Z^2-\gamma)$	$a = a_0 + a_1Z$

4.4 The Tate and Ate Pairings

The Tate pairing $e(P, Q)$ takes points $P \in E(\mathbb{F}_q)[r]$ and $Q \in E(\mathbb{F}_{q^k})$. For curves of embedding degree 10, since it has a quadratic twist, we can take $Q \in E_2(\mathbb{F}_{q^5})$ which will reduce the computing time to evaluate the Tate pairing. Similarly for curves of embedding degree 8, since it has a quartic twist, we can choose $Q \in E_4(\mathbb{F}_{q^2})$ which significantly reduces the time needed to evaluate the Tate pairing. For both curves, these input choices will assure us that arithmetic in $\mathbb{F}_{q^k}$ can be much more decreased when computing $l_{A,B}(Q)$. For an explicit formula to compute $l_{A,B}(Q)$ using projective coordinates, see [3].

The Ate pairing $a(P, Q)$ is similar to the Tate pairing but we swap the fields from where we take the points P and Q, i.e., we choose $Q \in E(\mathbb{F}_q)$ and $P \in E(\mathbb{F}_{q^k})$. As in the Tate pairing, we can choose $Q \in E_2(\mathbb{F}_{q^5})$ for curves of embedding degree 10 and $Q \in E_4(\mathbb{F}_{q^2})$ for the case of curves of embedding degree 8. Furthermore, instead of using r to control Miller's loop, the Ate pairing uses $s = t - 1$. For an explicit formula to compute $l_{A,B}(Q)$ using affine coordinates, see [3]

4.5 Expected Reductions

Below is the table containing a detailed comparison of the parameters of the sample curves that we will use. Note that the ρ-value pertains to the ratio of the order of the elliptic curve and the largest prime factor dividing the order.

Table 4. Parameter Comparison of the Curves Used in This Experiment

Embedding Degree	Type	ρ-value	Degree of Twist
8	$y^2 = x^3 + ax$	1.54	4
10	$y^2 = x^3 + ax + b$	1	2
12	$y^2 = x^3 + b$	1	6

Since the main operation that we use is multipication, we count the number of multiplications in $\mathbb{F}_{q^k}$ and $\mathbb{F}_{q^{k/d}}$ in the loops of the algorithms carefully. By using the twist of the curves mentioned above, the bitsize of the values in the algorithms will be reduced, and the running time of the algorithms is expected to be $1/2 - 3/4(1/d^2)$ faster. The algorithms will perform 45 percent faster for the curves of embedding degree 8, 31 percent faster for the curves of embedding degree 10 and 48 percent faster for the curves of embedding degree 12. The use of the denominator elimination method is expected to make the optimized algorithms 20 percent faster than the standard ones. Also, the ρ-value of the curve plays a big role in the pairing computations. Since the curves of embedding degrees 10 and 12 have ρ-value 1, namely, the bitsizes of r and q are equal, the computation time will not change. However, for curves of embedding degree 8, we expect the running time of the algorithm to be slower because the ρ-value of this family of curves is 1.5. We also expect the Ate pairing evaluation to be faster than the Tate pairing because the binary length of t is less compared to the binary length of r.

Of course these are just theoretical assumptions. One factor that truly affects the performance of the algorithms is the choice of extension fields. We will compare these expectations to the turnout of our experiment and we will discuss in detail the reasons why some of these expectations are not met.

5 Results

We implemented six algorithms, the standard Tate and Ate pairings, the Tate and Ate pairings with denominator elimination (d.e.) and the optimized Tate and Ate pairings using twists and denominator elimination. Their efficiency are measured over curves of embedding degrees 8 and 10. We generated 10 pairs of points for each curve as inputs in our programs. Our programs are written in MAGMA and ran on an AMD Opteron, 2GHZ dual core machine with 4GB of memory. Below are the tables for the average timings of the algorithms in seconds.

Table 5. Timings for the Tate/Ate Pairings

	k=8	k=10	k=12
Tate Pairing	0.0794	0.1173	0.1090
Tate Pairing (w/ twist)	0.0660	0.1169	0.0708
Tate Pairing (w/twist and d.e.)	0.0535	0.0797	0.0574
Ate Pairing	0.1624	0.1658	0.1557
Ate Pairing (w/ twist)	0.0687	0.1666	0.0443
Ate Pairing (w/twist and d.e.)	0.0587	0.1420	0.0375

6 Observations

We have demonstrated the first efficient implementation of the Tate/Ate pairings on ordinary curves of embedding degrees 8 and 10. For comparison, we also implemented both pairings on the curve used in [3]. Below is a table comparing the parameters r and t which contol the Miller loop of the Tate and Ate pairings, respectively.

Table 6. Parameter Comparison

k	$\lg r$	$\lg t$	$\omega = \log r / \log t$	d	k/d	result
8	230	175	1.31	4	2	$\omega < k/d$
10	234	118	1.98	2	5	$\omega < k/d$
12	256	128	2	6	2	$\omega \geq k/d$

As expected, Tate pairing computations require more time than the Ate pairing for the curve of embedding degree 12. This is because the Hamming weight of the parameter t, which is what we use to control the Miller loop of the Ate pairing algorithm, is much less than the parameter r, which on the other hand, is used in the Tate pairing algorithm. For both curves of embedding degrees 8 and 10, the binary length of the parameter r is almost twice as much compared to the parameter t so we expected the Ate pairing to perform better than the Tate pairing. However, in our experiment, the Tate pairing computation is faster than the Ate pairing even when the twist of the curves are used and the Hamming weight of the parameter r is much larger compared to the parameter t. In fact, for the curve of embedding degree 10, the use of twist for the Ate pairing does not improve the running time of the algorithm. From the table above, if the value of $\omega < k/d$, the Tate pairing will perform better. On the other hand, if the value of $\omega \geq k/d$, the evaluation of the Ate pairing will be much faster compared to the Tate pairing. Therefore, the performance of both pairings not only depend on the binary length of the controlling parameters. The bitsize of the parameters and the degree of the twist also affect the computation time of both pairings.

Furthermore, the optimizations we applied indeed reduces the running time of the programs. For the curves of embedding degrees 8 and 12, our expected running time for the algorithms are not far from the outcome of our experiment, especially for the Ate pairing. However, for the curve of embedding degree 10, the use of twist does not seem to improve the algorithms for both pairings. On the other hand, using the denominator elimination method on both the Tate and Ate pairings for all three curves, the expected reduction is not far from we expected which is 20 percent. The probable cause of the differences is our choice of extension fields to which we evaluate the pairings and the parameters of the curve that we chose.

References

1. P. Baretto, H. Kim, B. Lynn, M. Scott: Efficient Algorithms for Pairing Based Cryptosystems. In: M. Yung (ed.) CRYPTO 2002, pp. 354–368, LNCS vol. 2442, Springer-Verlag, 2002.
2. P. Baretto, M. Naehrig: Pairing-Friendly Elliptic Curves of Prime Order. In: B. Preneel, S. Tavares (eds.), SAC 2005, LNCS vol. 3897, pp. 319–331, Springer-Verlag 2006.
3. A.J. Devegili, M.Scott, R. Dahab: Implementing Cryptographic Pairings over Baretto-Naehrig Curves. In: T. Takagi, T. Okamoto, E. Okamoto, T. Okamoto (Eds.) Pairing-Based Cryptography Pairing 2007. LNCS vol. 4575, pp. 197–207, Springer-Verlag 2007.
4. D. Freeman: Constructing Pairing Friendly Elliptic Curves with Embedding Degree 10. In: F. Hess , S. Pauli, M. Pohst (eds.) Algorithmic Number Theory, LNCS vol. 4076, pp. 452–465, Springer Heidelberg, 2006.
5. D. Freeman, M. Scott, E. Teske: A Taxonomy of Pairing-Friendly Elliptic Curves. Preprint, 2006. Available at http://math.berkeley.edu/dfreeman/papers/taxonomy.pdf
6. S. Galbraith: Pairings. In: Advances in Elliptic Curve Cryptography, pp. 183–213, Cambridge University Press, 2005.
7. K. Harrison, D. Page, N.P. Smart: Software Implementation of Finite Fields of Characteristic 3, for Use in Pairing Based Cryptosystem. In: LMS Journal of Computation and Mathematics, London, vol.5(1), pp.181–193, London Mathematical Society, London 2002.
8. Hess, F., Smart, N.P., Vercauteren, F., The Eta Pairing Revisited. IEEE Transactions on Information Theory 52(10), 4595–4602, 2006.
9. V. Miller: Short Programs for Functions on Curves, unpublished manuscript, 1986. Available at http://crypto.stanford.edu/miller/miller.pdf.
10. A. Miyaji, M. Nakabayashi, S. Takano: New Explicit Conditions of Elliptic Curve Traces for FR-Reduction. IEICE Taransactions on Fundamentals, pp. 1234–1243, E84–A(5)(2001).
11. M. Scott: Implementing Cryptographic Pairings. In: T. Takagi, T. Okamoto, E. Okamoto, T. Okamoto (Eds.) Pairing-Based Cryptography Pairing 2007. LNCS vol. 4575, pp. 177–196, Springer-Verlag 2007.
12. M Scott, P. Baretto: Generating More MNT Elliptic Curves. Designs, Code and Cryptography, Vol. 38, No.2, pp.209–217, 2006.

13. S. Tanaka: More Constructing Pairing-Friendly Elliptic Curves for Cryptography. Masters Thesis, Tokyo Metropolitan Univerity (2007). Available at www.tnt.math.metro-u.ac.jp/labo/master/2006/satoru/thesis2006r1.pdf (in japanese).
14. S. Tanaka, K. Nakamula: More Constructing Pairing-Friendly Elliptic Curves for Cryptography. To appear in Transactions of the Japan Society for Industrial and Applied Mathematics (JSIAM), vol. 17, No. 4, 2007 (in japanese).