Computing the Ate Pairing on Elliptic Curves with Embedding Degree $k = 9$

Xibin Lin[1], Chang-An Zhao[2,3], Fangguo Zhang[2,3], and Yanming Wang[1]

[1] School of Mathematics and Computational Science
Sun Yat-Sen University Guangzhou, 510275, P.R.China
[2] Guangdong Key Laboratory of Information Security Technology
[3] School of Information Science and Technogoly,
Sun Yat-Sen University, Guangzhou 510275, P.R.China
linxibin@mail2.sysu.edu.cn
zhcha@mail2.sysu.edu.cn
isszhfg@mail.sysu.edu.cn
stswym@mail.sysu.edu.cn

Abstract. For AES 128 security level there are several natural choices for pairing-friendly elliptic curves. In particular, as we will explain, one might choose curves with $k = 9$ or curves with $k = 12$. The case $k = 9$ has not been studied in the literature, and so it is not clear how efficiently pairings can be computed in that case. In this paper, we present efficient methods for the $k = 9$ case, including generation of elliptic curves with the shorter Miller loop, the denominator elimination and speed up of the final exponentiation. Then we compare the performance of these choices. From the analysis, we conclude that for pairing-based cryptography at the AES 128 security level, the Barreto-Naehrig curves are the most efficient choice, and the performance of the case $k = 9$ is comparable to the Barreto-Naehrig curves.

Keywords: Ate pairing, Denominator elimination, Final exponentiation.

1 Introduction

Pairing computation is an important primitive in cryptographic systems. Much work has been done on related topics, including an denominator elimination

method [3], the selection of pairing-friendly groups [6], the construction of pairing-friendly curves [4][7][8][10][19], the methods to shorten the Miller loop [2][9] [14] and etc.

In [14], Hess et al. extended the η_T pairing [2] over ordinary curves, and proposed the Ate pairing. The Miller loop in the Ate pairing can be reduced to T, where $T = t - 1$ and t is the Frobenius trace of the elliptic curve. So the performance of the Ate pairing is dependent on the selection of elliptic curves. Duan et al. extended Brezing-Weng's method [7] to generate curves with $t \approx r^{1/\varphi(k)}$ when $k = 2^i \cdot 3$ [8]. Here and in the following of this paper, $\varphi(x)$ will be the Euler totient function, and $\Phi_k(x)$ be the kth cyclotomic polynomial [17].

In pairing based cryptography, it is required that r and p^k should be sufficiently large, and k, $\rho = \lg p / \lg r$ should be appropriately small. Here, r is the size of the prime subgroup, $\mathbb{F}_p$ is the underlying field of the elliptic curve and k is the embedding degree, which is the smallest integer that $r \mid p^k - 1$. An elliptic curve satisfying these conditions is called a "pairing-friendly elliptic curve". For different AES security levels, we can choose different k and p to achieve the balance of security and performance efficiency.

Although the constructed curves in [8][11] provide efficient Ate pairing implementation due to the shorter loop, we extend the method to generate more elliptic curves with embedding degree $k = 3^i$. These curves might offer better efficiency due to the following reasons:

1. We constructed such pairing-friendly elliptic curves with $t \approx r^{1/\varphi(k)}$. It's well known that $\varphi(3^i) = 2 \cdot 3^{i-1}$ and $\varphi(2^i \cdot 3) = 2^i$. For embedding degrees of cryptographic interest and the same security level, the size of r is the same. So the Ate pairing of elliptic curves with $k = 3^i$ would require a much shorter Miller loop.
2. By the method, the curves constructed in [8] for $k = 12$ have $\rho = 1.5$, while the curves constructed in our extended method for $k = 9$ have $\rho = 1.33$. This is important to save computation cost and bandwidth.

For odd embedding degrees, the denominator elimination method and the speed up of final exponentiation can not be easily applied. And the efficiency of pairing computation in odd embedding degrees has not been studied in detail in the literature.

In this paper, we first extend Duan's technique to generate curves of $k = 3^i$ with $t \approx r^{1/\varphi(k)}$. Second, we propose a method that is analogous to the denomi-

nator elimination method using cubic twist elliptic curves. Third, we describe the method to speed up the final exponentiation using the explicit representation of coefficients of the final exponentiation. Finally, following Koblitz's analysis [15], we analyze the performance over the proposed Ate pairing-friendly elliptic curves and Barreto-Naehrig curves [4] at AES 128 security level.

The contributions of this paper are as follows. First, we present the efficient methods to compute the Ate pairing on elliptic curves with $k = 9$, and this has not be studied in detail in the literature. Second we compare the performance of different choices of elliptic curves under the AES 128 security level. From the analysis, we conclude that for pairing-based cryptography at the AES 128 security level, the Barreto-Naehrig curves are the most efficient choice, and the performance of the case $k = 9$ is comparable to the Barreto-Naehrig curves.

This paper is organized as follows. Section 2 introduces the mathematical preliminaries, including the Ate pairing and the CM method for curve generation and the AES security levels. Section 3 describes the method of generating Ate pairing-friendly elliptic curves with $k = 3^i$. Section 4 describes the method of denominator elimination over the proposed curves. Section 5 describes the method to speed up the final exponentiation. Section 6 compares the efficiency of the choices of elliptic curves under the AES 128 security level. Section 7 summarizes our work.

2 Mathematical Preliminaries

In this section, we briefly recall the Ate pairing, the CM method for generating elliptic curves and the matching of AES security.

2.1 The Ate Pairing

In this section, we briefly recall the Ate pairing [14]. Let $\mathbb{F}_p$ be a finite field with p elements, where p is a prime. Let E be an ordinary elliptic curve over $\mathbb{F}_p$, r a large prime with $r \mid \#E(\mathbb{F}_p)$ and let t denote the trace of Frobenius, i.e. $\#E(\mathbb{F}_p) = p+1-t$. Let π_p be the Frobenius endomorphism, $\pi_p : E \to E : (x, y) \mapsto (x^p, y^p)$. For $T = t-1$, $Q \in \mathbb{G}_2 = E[r] \cap Ker(\pi_p - [p])$ and $P \in \mathbb{G}_1 = E[r] \cap Ker(\pi_p - [1])$, we have the following:

- $f_{T,Q}(P)$ defines a bilinear pairing, which is called the Ate pairing.

- let $N = gcd(T^k - 1, p^k - 1)$ and $T^k - 1 = LN$, with k the embedding degree, then
$$e(Q, P)^L = f_{T,Q}(P)^{c(p^k-1)/N}$$
where $c = \sum_{i=0}^{k-1} T^{k-1-i} p^i \equiv kp^{k-1} \bmod r$.
- for $r \nmid L$, the Ate pairing is non-degenerate.

Note that the length of Miller loop of the pairing computation is determined by the trace of Frobenius t. Therefore, choosing the elliptic curves with a small t can speed up the pairing computation.

2.2 Complex Multiplication Method

The CM method can be used to generate elliptic curves of given order [1]. The integer equation $4p = t^2 - Dy^2$ is called CM equation, here D is a fundamental discriminant. The elliptic curve has endomorphism ring equal to an order $\mathcal{O}$ in number field $Q(\sqrt{D})$.

For completeness we recall the following lemma for construction of pairing-friendly elliptic curves observed by Cocks and Pinch and stated in [11].

Lemma 1. *Let k be a positive integer, $E/\mathbb{F}_p$ an elliptic curve with $\#E(F_p) = hr$, where r is prime, and let t be the trace of E/F_p. Assume that $r \nmid k$. Then $E/\mathbb{F}_p$ has embedding degree k with respect to r if and only if $\Phi_k(p) \equiv 0 \bmod r$,or equivalently, if and only if $\Phi_k(t-1) \equiv 0 \bmod r$.*

This observation leads to an efficient construction algorithm due to Brezing and Weng. In the following we briefly recall their method(Section 1 of [7]).

Given k, D and the notations as above, take a prime r that splits in $\mathcal{O}$, and $r \equiv 1 \bmod k$. Choose ζ_k as a primitive kth root of unity modulo r. Set $t = \zeta_k + 1 \bmod r$ and $b = \pm\frac{t-2}{\delta} \bmod r$ where δ is a square root of D modulo r. Choose different r and ζ_k until $Norm_{Q(\sqrt{D})/Q}(\frac{t+b\sqrt{D}}{2})$ is a prime p. Using the CM method with D and p gives $E(\mathbb{F}_p)$. Then $E(\mathbb{F}_p)$ has a subgroup of order r, and embedding degree k with respect to r.

For convenience, we also follow the language of "family of pairing -friendly elliptic curves" [10] to give a definition of family of Ate pairing-friendly elliptic curves mentioned in [11].

Definition 1. (Family of Ate Pairing-Friendly Elliptic Curves) *Let $p(x)$, $n(x)$, $t(x)$, $r(x)$ and $h(x)$ be polynomials with integer coefficients. For a given*

positive integer k, the triple (p, r, t) represents a family of Ate pairing-friendly elliptic curves with embedding k with respect to r if the following conditions are satisfied:

1. *$n(x) = p(x) - t(x) + 1$ and $n(x) = r(x) \cdot h(x)$;*
2. *$r(x)$ and $p(x)$ are irreducible;*
3. *$r(x)$ divides $\Phi_k(t(x) - 1)$, where Φ_k is the kth cyclotomic polynomial;*
4. *$deg(t(x)) \leq deg(r(x)) \cdot 1/(\varphi(k))$;*
5. *The equation $Dy^2 = 4p(x) - t(x)^2$ has infinitely many integer solutions (x,y) for some proper discriminant D.*

2.3 Matching AES Security Using Public Key Systems

In [16], Lenstra analyzed the matching of the AES security levels to public key systems. This matching becomes important when considering the choice of embedding degrees. For later analysis, we recall Koblitz's table about different security levels [15].

Table 1. Minimum bitlengths of r and p^k

AES security level	80	128	192	256
bitlength of b_r	160	256	384	512
bitlength of b_{p^k}	1024	3072	8192	15360
$\gamma = \frac{b_{p^k}}{b_r}$	6.4	12	$21\frac{1}{3}$	30

3 Generating the Ate Pairing-Friendly Elliptic Curves with $k = 3^i$

In this section, following Brezing and Weng's method [7], we generate the Ate pairing-friendly curves with $k = 3^i$.

It is well known that $\Phi_{3^i}(x) = x^{2 \cdot 3^{i-1}} + x^{3^{i-1}} + 1$, so we have the following,

Theorem 1. *Let $t(x) = x + 1$, $r(x) = \Phi_{3^i}(x)$, $p(x) = \{(x+1)^2 + [(x-1)^2 \cdot (2 \cdot x^{3^{i-1}} + 1)^2/3]\}/4$ and $D = -3$, then (t, r, p) represents a family of Ate pairing-friendly elliptic curves with embedding degree $k = 3^i$.*

Proof. The proof of this theorem is a direct application of Brezing and Weng's method [7].

Let $r(x) = \Phi_{3^i}(x)$, then any integer x_0 is a kth primitive root of unity modulo $r(x_0)$. Let $t = x + 1$ and $h_1(x) \in Z[x]$ such that $h_1(x)^2 \equiv -3 \pmod{r(x)}$. By choosing $r(x)$ in the above way, we have

$$4r(x) - 3 = (2 \cdot x^{3^{i-1}} + 1)^2 = h_1^2(x).$$

Let $b'(x) = (t-2)h_1(x) = (x-1)(2 \cdot x^{3^{i-1}} + 1)$, so that

$$p = Norm_{Q(\sqrt{D})/Q}(\frac{t + \frac{b'}{D} \cdot \sqrt{D}}{2}) = \frac{(x+1)^2 + \frac{(x-1)^2 \cdot (2 \cdot x^{3^{i-1}} + 1)^2}{3}}{4}.$$

If there exists some integer x_0 such that $r(x_0)$ and $p(x_0)$ are primes, then $E(\mathbb{F}_{p(x_0)})$ can be constructed by CM method. By the construction, (t, r, p) satisfies Definition 1. □

Note that this theorem is an extension of Duan et al.'s result. But in their paper [8], they did not consider the case of $k = 3^i$.

Remark 1. It's important that our construction gives curves with $D = -3$, so that we can use cubic twist curves. The construction of Freeman in section 6.2 of [11] generates Ate pairing-friendly elliptic curves with $D = -1$. It is unclear whether the method by Theorem 6.19 of [11] can give curves with $D = -3$. Even if this can be done, their method gives elliptic curves with $\rho = \frac{k+2}{\varphi(k)}$, and our constructed curves have $\rho = \frac{\varphi(k)+2}{\varphi(k)}$. Using elliptic curves with smaller ρ can achieve more efficiency.

Here we give an example of elliptic curves constructed by our method.

Example 1. **(Embedding Degree $k = 9$ at Security Level of AES 128)**

Using the proposed construction, we can let

$t(x) = x + 1;$
$r(x) = x^6 + x^3 + 1;$
$p(x) = \{(x+1)^2 + [(x-1)^2 \cdot (2 \cdot x^3 + 1)^2]/3\}/4;$

In this example, we set the security level at AES 128, which requires the size of r equals to about 256 bits, and the size of the finite field $\mathbb{F}_{p^k}$ more than 3072 bits [15]. By our construction, we have $\rho = 1.33$ and $\omega = 6 = \varphi(9)$. In the following

example, r is 260 bits, p is 348 bits, and $T = t - 1$ is 44 bits, and the hamming weight of T is 7. Here we choose $D = -3$. Then the curve will admit a cubic twist. We can easily find the desired curve of the form $y^2 = x^3 + 1$ with

- T =C0000001831
- r =F300000B7B418039DD7C67EB89AC47F31ED80AE3BC07BE91EF26898B BA86F4E91
- p =88B000089C5A583CC0E78CD1F13D85701832C21FCA17C75C976D037B 4E661B69A0177F059FF0B46731C3131
- $\sharp E(F_p)$=88B000089C5A583CC0E78CD1F13D85701832C21FCA1 7C75C976D037B4E661B69A0177F059FEFF46731C1900

Note that curves over $\mathbb{F}_p$ with the same j *invariant* are isomorphic over $\overline{\mathbb{F}}_p$. So if the curve is of the form $y^2 = x^3 + b$, then defined over $\mathbb{F}_p$, it will only have six possible group orders. It's easy to choose the value of b that would give the prescribed order.

4 Denominator Elimination Using Cubic Twisted Elliptic Curves

In this section, we propose a method to eliminate the denominator computation in each Miller step.

4.1 Miller's Algorithm

First, we recall Miller's algorithm to compute the pairing.

Algorithm 1: Miller's Algorithm for the Ate Pairing

Input: $T = \sum_{i=0}^{n} l_i 2^i$, where $l_i \in \{0, 1\}$. $Q \in G_2$ and $P \in G_1$
Output: $f_{T,Q}(P)^{(p^k-1)/r}$
1 $R \leftarrow Q$, $f_1 \leftarrow 1$
2 for $i = n-1, n-2, \ldots, 1, 0$ do
2.1 $f_1 \leftarrow f_1^2 \cdot \frac{l_{R,R}(P)}{v_{2R}(P)}$, $R \leftarrow 2R$
2.2 if $l_i = 1$ then
2.3 $f_1 \leftarrow f_1 \cdot \frac{l_{R,Q}(P)}{v_{R+Q}(P)}$, $R \leftarrow R + Q$
3 return $f_1^{(p^k-1)/r}$

Here, $l_{R,Q}$ means the line through R and Q, and v_{R+Q} means the vertical line through $R+Q$.

In [3], Barreto et al. proposed an elegant method to eliminate the denominator. They observed that when choosing a point Q defined over a proper subfield, the valuation of the line $v_{R+Q} = x_{R+Q} - x_P$ would be in a proper subfield of $\mathbb{F}_{p^k}$. Thus the denominator would become 1 when the final exponentiation is performed.

When $k = 2d$ is even, it is very efficient to implement such idea by using quadratic twisted curves. Let $E(\mathbb{F}_{p^d}) : y^2 = x^3 + ax + b$, then the quadratic twisted curve would be defined by $E'(\mathbb{F}_{p^d}) : Dy^2 = x^3 + ax + b$, where D is a quadratic non residue over $\mathbb{F}_{p^d}$. We can first choose $Q' = (x, y) \in E'(\mathbb{F}_{p^d})$, then $Q = \Psi(Q') = (x, \sqrt{D} \cdot y) \in E(\mathbb{F}_{p^k})$. So $\langle Q \rangle$ is a subgroup with $x \in \mathbb{F}_{p^d}$, and $y \in \mathbb{F}_{p^k}$. So $v_{R+Q} = x_{R+Q} - x_P$ will always lies in the proper subfield of $\mathbb{F}_{p^k}$, thus can be eliminated.

4.2 Denominator Elimination for Elliptic Curves with $k = 3^i$ Admitting Cubic Twist

Up to our knowledge, there does not exist such a method when the embedding degree is odd. So curves with odd embedding degrees are precluded in many pairing applications. The following shows a method to eliminate the denominator for curves with $k = 3^i$ admitting cubic twist.

Let $E(\mathbb{F}_{p^{k/3}}) : y^2 = x^3 + b$, then E admits a cubic twisted curve $E'(\mathbb{F}_{p^{k/3}}) : y^2 = x^3 + b/D$. Here D is not a cubic residue but a quadratic residue over $\mathbb{F}_{p^{k/3}}$. The monomorphism

$$\begin{aligned} \Psi_3 : E'(\mathbb{F}_{p^{k/3}}) &\rightarrow E(\mathbb{F}_{p^k}). \\ (x', y') &\rightarrow (D^{1/3}x', D^{1/2}y') = (x, y) \end{aligned}$$

maps the points on the cubic twisted curve to the curve defined over the cubic extension. Note that, in pairing implementation, we first choose $Q' \in E'(\mathbb{F}_{p^{k/3}}) \setminus E'(\mathbb{F}_p)$ of prime order r. Then under this monomorphism, we have a cyclic subgroup of order r generated by $Q = \Psi_3(Q')$. The group $\langle Q \rangle$has the nice property that when $(x, y) \in \langle Q \rangle$, then $x \in \mathbb{F}_{p^k}$, while $y \in \mathbb{F}_{p^{k/3}}$. This is fundamental to our method of denominator elimination.

Proposition 1. *Let $H \in E(\mathbb{F}_{p^k})$ and $P \in E(\mathbb{F}_p)$. Then $\frac{1}{x_H - x_P} = \frac{x_H^2 + x_H x_P + x_P^2}{(y_H - y_P)(y_H + y_P)}$.*

Proof. Because $H \in E(\mathbb{F}_{p^k})$ and $P \in E(\mathbb{F}_p)$, then we have

$$y_H^2 = x_H^3 + b;$$
$$y_P^2 = x_P^3 + b;$$

then subtract the above two equation would lead the result. □.

By choosing H using the monomorphism, $(y_H - y_P)(y_H + y_P)$ lies in the proper subfield of $\mathbb{F}_{p^k}$. So the denominator would become 1 in the final exponentiation step. This observation will easily lead to the following useful lemma.

Lemma 2. *Over the curves defined as above, let $\langle Q \rangle$ be the subgroup of $E(\mathbb{F}_{p^k})$ using the above monomorphism, $H \in \langle Q \rangle$ and $P \in E(\mathbb{F}_p)$. Let $S_H(P) = x_H^2 + x_H x_P + x_P^2$. Then first part of Step 2.1 in Algorithm 1 can be replaced with $f_1 \leftarrow f_1^2 \cdot l_{R,R}(P) \cdot S_{2R}(P)$, and first part of Step 2.3 in Algorithm 1 can be replaced with $f_1 \leftarrow f_1 \cdot l_{R,Q}(P) \cdot S_{R+Q}(P)$* □

From the above analysis, a modification of algorithm 1 might be given in the following:

Algorithm 2:Algorithm for Elliptic Curves over Cubic Extension

Input: $T = \sum_{i=0}^{n} l_i 2^i$, where $l_i \in \{0, 1\}$. $Q \in G_2$ and $P \in G_1$
Output: $f_{T,Q}(P)^{(p^k-1)/r}$
1 $R \leftarrow Q$, $f_1 \leftarrow 1$.
2 for $i = n-1, n-2, \ldots, 1, 0$ do
2.1 $f_1 \leftarrow f_1^2 \cdot l_{R,R}(P) \cdot S_{2R}(P)$, $R \leftarrow 2R$
2.2 if $l_i = 1$ then
2.3 $f_1 \leftarrow f_1 \cdot l_{R,Q}(P) \cdot S_{R+Q}(P)$, $R \leftarrow R + Q$
3 return $f_1^{(p^k-1)/r}$

Remark 2. Algorithm 2 replaces the inversion in Algorithm 1 with one full extension field multiplication and one base field multiplication(See line 2.1 and 2.4 in two algorithms). While the full denominator elimination totally eliminates the inversion. So the efficiency between our denominator elimination method and the full denominator elimination method is small. At the same security level, the Miller loop is much shorter when the embedding degree is 9, so the efficiency of the total Miller loop in the case of $k = 9$ is still comparable to the Ate paring on elliptic curves with even embedding degrees.

5 Explicit Coefficients for Frobenius Expansion of the Final Exponentiation

In the section, we describe the method to speed up the final exponentiation.

5.1 Explicit Coefficients of Frobenius Expansion for $k = 9$

The original parameter setting can be easily derived from our construction. For the polynomials of p and r to be primes more likely, we substitue x with $6x + 1$, so the parameters would be

$t(x) = 6x + 2$
$r(x) = 15552x^6 + 15552x^5 + 6480x^4 + 1512x^3 + 216x^2 + 18x + 1$
$p(x) = 559872x^8+559872x^7+233280x^6+54432x^5+7776x^4+648x^3+36x^2+6x+1$

For the final exponentiation, we need to compute $f^{\frac{p^9-1}{r}}$. Let $e = f^{p^3-1}$, so the hard part of the exponentiation is to compute $e = \frac{p^6+p^3+1}{r} \in F_p[x]$, since we don't use this value to do arithmetic, we omit the explicit polynomial of e.

We expand e under base p, so we have $e = a_5p^5+a_4p^4+a_3p^3+a_2p^2+a_1p^1+a_0$. We show that a_i can be defined explicitly as follows:

$a_5 = 36x^2$
$a_4 = 216x^3 + 36x^2$
$a_3 = 1296x^4 + 432x^3 + 36x^2$
$a_2 = 7776x^5 + 3888x^4 + 648x^3 + 72x^2$
$a_1 = 46656x^6 + 31104x^5 + 7776x^4 + 1080x^3 + 72x^2$
$a_0 = 279936x^7 + 233280x^6 + 77760x^5 + 14256x^4 + 1512x^3 + 72x^2 + 3$

These can be easily verified by an explicit computation.

5.2 Speeding up the Final Exponentiation Using Explicit Expansion

In this section, we show that the explicit polynomials provide a method to compute the final exponentiation. It is used by Scott in the case of the Barreto-Naehrig curves(MIRACL) [20].

Let k be the embedding degree, define $D_k = Max\{deg(a_0), \cdots, deg(a_{\varphi(k)-1})\}$. We describe the method in the following algorithm.

Algorithm 3: Algorithm for Compute $g^e = g^{\sum_{i=0}^{\varphi(k)-1} a_i p^i}$

Input: g, e
Output: g^e
1 Compute and store $M_i = g^{x^i}$ for $i = 0, 1, \cdots, D_k$;
2 Compute and store $S_i = g^{a_i}$ for $i = 0, 1, \cdots, \varphi(k) - 1$ from M_i and the explicit polynomials;
3 Compute $S_0 \cdot S_1^p \cdot \cdots S_{\varphi(k)-1}^{p^{\varphi(k)-1}}$;

Given g^{x^i}, it is easy to compute g^{a^i} in the cost of some limited storage using the explicit polynomials, because there exist relations with the coefficients of a_i. A direct method to use these relations is to compute a_5 first, then compute a_4, a_3, a_2, a_1, and finally a_0. During the computation, store the intermediate results for later use. So to estimate the cost of computing all these g^{a^i}, it is sufficient to count the total length of the binary representation of the coefficients of a_0. We will discuss this in detail in the following section.

And since all the exponentiation in the pairing computation are in the cyclotomic subgroup $G_{\Phi_k(p)}$, the Frobenius map is cheap.

So the dominant steps of the final exponentiation in this method are computing the exponent g^{x^i}. Using the standard binary method, the cost is about $D_k \log x$ square and multiplication operations.

The bit-length of the hard part of the final exponentiation e is estimated by $\varphi(k) \log p - \log r$. In the case of $k = 9$, the bit length of the hard part of final exponentiation is about 1796, which means that by naive implementation, it would require about 1796 exponentiation steps. But since the bit-length of x is only 41. So by Algorithm 3, the dominant steps only require 287 exponentiation steps. When x is of low hamming weight, in most of the steps, only squaring is performed. This method greatly improve the speed of the final exponentiation.

6 Efficiency Analysis at AES 128 Security Level

In this section, we compare the efficiency of different choices of curves under the AES 128 security level.

6.1 AES Security Levels and Different Embedding Degrees

As in Table 1, when implementing cryptographic protocols, one must pay attention to the balance between efficiency and security. In the table, it seems that

$k = 6, 12, 24$ offer a perfect matching to AES 80, 128, 192. But the embedding degrees are derived under an important assumption.

Assumption 1 *Let the curves defined by $E(\mathbb{F}_p)$ and r is the size of the prime subgroup. Then we assume that $r \approx p$.*

Generating pairing-friendly curves of prime order for more general k is still an open question [4][11][10].

The Barreto-Naehrig curves [4] have prime order and some other advantages. But the Frobenius trace is $t \approx r^{1/2}$, which requires a longer Miller loop for the Ate pairing.

On the other side the existing Ate pairing-friendly elliptic curves do not have prime order group. Up to our knowledge, the existing Ate pairing-friendly elliptic curves are curves in [8] and the curves summarized in section 8.4 of [11]. For curves in [8], we have $\rho = \frac{\varphi(k)+2}{\varphi(k)}$. Curves in section 8.4 of [11] have ρ between $\frac{k+2}{\varphi(k)}$ to $\frac{\varphi(k)+2}{\varphi(k)}$ depending on the various methods. And the curves constructed in this paper have same ρ value as [8] too. But as $k = 3^i$, this ρ would be relatively small.

So for the Ate pairing implementation, we must reconsider this table using the existing elliptic curves in the literature. And we do our following analysis accordingly. Table 2 shows the matching to the AES security levels. And we see that for pairing base cryptography at AES 128 security level, these three families of elliptic curves are the natural choices.

Table 2. Security Matching for the AES 128

Curves	embedding degree k	bits of r	bits of q	bits of q^k
Ate Friendly	9	256	342	3076
Ate Friendly	12	256	384	4608
Barreto-Naehrig	12	256	256	3072

6.2 Efficiency Estimation of the Ate Pairing Computation

In this subsection, we compare the efficiency of the above three cases. we compare the total computational cost of the Miller loop and the final exponentiation.

The Cost of the Miller Loop For $k = 9$ we consider the curves admitting cubic twist, and for $k = 12$ we consider the curves admitting sextic twist. These curves have the similar equation: $y^2 = x^3 + b$. They are curves with j *invariant* equal to zero. Following the analysis in [14], let C^A_{Full} denote the cost of Miller loop in Ate pairing computation using affine coordinate, M_k, S_k and I_k denote the multiplication, squaring and inversion in the finite field $\mathbb{F}_{p^k}$ respectively. Using our proposed denominator elimination method in section 4. we have

$$C^A_{Full} = (2S_3 + 3M_3 + I_3 + 6M_1 + S_9 + 2M_9)log_2T \quad k = 9$$
$$C^A_{Full} = (2S_2 + 3M_2 + I_2 + 2M_1 + S_{12} + M_{12})log_2T \quad k = 12$$

For the case $k = 9$, compare line 2.1 in algorithm 1 and 2. In algorithm 2, line 2.1 requires one more M_9 and one more $3M_1$. So we have the estimation formula above.

Following the rough estimation of extension field arithmetic. In the case of $k = 9$, we assume $M_9 = S_9, M_1 = S_1$ and $M_9 = 25M_1, S_9 = 25S_1$. In the case of $k = 12$, we assume $M_{12} = S_{12}, M_1 = S_1$ and $M_{12} = 45M_1, S_{12} = 45S_1$ [14]. Here, we assume $1I_1 = 10M_1$

In the double step of each Miller loop, the case $k = 9$ requires $130M_1$ and the case $k = 12$ requires $121M_1$. (Here we only consider the double step, because we can choose x of low hammming weight)

The Cost of the Final Exponentiation We use algorithm 3 to compute the hard part of the final exponentiation. The case for the Barreto-Naehrig curves has been implemented, and the case for the Ate pairing-friendly elliptic curves with $k = 12$ is similar.

Firstly, we need to compute the easy part, which need $9M_1+I_9+M_9$, $36M_1+3M_{12}+I_{12}$, and $36M_1 + 3M_{12} + I_{12}$ respectively.

For step 1, because we are able to generate curves with x of low hamming weight, we assume only squaring is required in the binary method.

Ate friendly $k = 9$ $D_k = 7$, $\log x = 41$ the computation cost $287S_9$
Ate friendly $k = 12$ $D_k = 5$, $\log x = 64$ the computation cost $320S_{12}$
Barreto-Naehrig curve $k = 12$ $D_k = 3$, $\log x = 64$ the computation cost $192S_{12}$

For step 2, as mentioned in Section 5.2, we first need to count the binary representation of the coefficients of a_0, which is the exponentiation steps we need. Then for each two monomials of these a_i's, one extension field multiplication M_k is needed. So for step 2, we need $88(S_9 + 1/2M_9) + 16M_9$ for the case of $k = 9$. The cost for the other two cases are $35(S_{12}+1/2M_{12})+5M_{12}$, $18(S_{12}+1/2M_{12})+7M_{12} + 24M_1$ respectively.

For step 3, the Frobenius map of raising to p^i would cost kM_1. So the cost would be $45M_1$, $36M_1$ and $36M_1$. Computing the final value g^e = needs $(\phi(k) - 1)M_k$, so the cost is $5M_9$, $3M_{12}$ and $3M_{12}$ respectively.

Note that in the final exponentiation all the computation is performed in the cyclotomic subgroup $G_{\phi_k(p)}$. When k is even, we can use the method introduced by Granger et.al. [12] to estimate $S_{12} = 30M_1$. For $k = 9$, we do not have an analogue method.

The Total Cost Comparision The following table summarizes our efficiency consideration and thus show the theoretical results on the Ate pairing computation. The size of p is different for such elliptic curves, so let M denote the 32×32 bit multiplication operation.

Table 3. Efficiency Comparision at AES 128

Curve	Exponen	Miller	Total	Basic Multiplication
Ate friendly $k = 9$	$11185M_1$	$5590M_1$	$17650M_1$	4428600M
Ate friendly $k = 12$	$12039M_1$	$7744M_1$	$19783M_1$	6172296M
Barreto-Naehrig $k = 12$	$7398M_1$	$15488M_1$	$22886M_1$	4119480M

From the table, we see that at the AES 128 security level, the most efficient choice is the Barreto-Naehrig curves, but the efficiency of the proposed Ate pairing friendly elliptic curves with $k = 9$ is comparable to the Barreto-Naehrig curves. Note also that in this estimation, we use the computation in the torus [12] to speed up exponentiation in the $k = 12$ case, but we have not considered any trick in the $k = 9$ case. Further research might find similar methods for the computation in the cyclotomic group $G_{\phi_9(p)}$, this might further speed up pairing computation in the case $k = 9$.

7 Conclusion

In this paper, we consider the computation of the Ate pairing over the proposed curves with $k = 9$, including the generation of such curves, the denominator elimination and the speed up of the final exponentiation. Our proposed methods make the performance of such curves comparable to the Barreto-Naehrig curves at AES 128 security level.

Acknowledgement

We would like to thank Steven Galbraith for his valuable comments and encouragements on this work.

References

1. A. O. L. Atkin and F. Morain. *Elliptic Curves and Primality Proving*. Math. Comput. Volumn 61, pp.29-68, 1993.
2. P.S.L.M. Barreto, S. Galbraith, C. ÓhÉigeartaigh, and M. Scott. *Efficient pairing computation on supersingular abelian varieties*. Designs, Codes and Cryptography. Volume 42, Number 3, pp.239-271, Springer-Verlag, 2007.
3. P.S.L.M. Barreto, H.Y. Kim, B. Lynn, and M. Scott. *Efficient algorithms for pairing-based cryptosystems*. Crypto'2002, LNCS 2442, pp.354-368, Springer-Verlag, 2002.
4. P.S.L.M. Barreto and M. Naehrig. *Pairing-Friendly Elliptic Curves of Prime Order*. SAC'2005, LNCS 3897, pp.319-331, Springer-Verlag, 2006.
5. P.S.L.M. Barreto, B. Lynn, and M. Scott. *Constructing Elliptic Curves with Prescribed Embedding Degrees*. SCN, LNCS 2576, pp.257-267, Springer-Verlag, 2002.
6. P.S.L.M. Barreto, B. Lynn, and M. Scott. *On the selection of pairing-friendly groups*. SAC'2003, LNCS 3006, pp.17-25, Springer-Verlag, 2004.
7. F. Brezing and A. Weng. *Elliptic curves suitable for pairing based cryptography*. Designs, Codes and Cryptography. Volume 37, Number 1, pp.133-141, Springer-Verlag, 2005.
8. P. Duan, S. Cui, and C.W. Chan. *Special Polynomial Families for Generating More Suitable Elliptic Curves for Pairing-Based Cryptosystems* . Proceedings of 5th WSEAS International Conference on Electronics, Hardware, Wireless and Optical Communications (EHAC 2006).
9. I. M. Duursma and H. S. Lee. *Tate Pairing Implementation for Hyperelliptic Curves* $y^2 = x^p - x + d$. Asiacrypt'2003, LNCS 2894, pp.111-123, Springer-Verlag, 2003.

10. D. Freedman. *Constructing pairing-friendly elliptic curves with embedding degree 10*. ANTS-VII, LNCS 4076, pp.452-465, Springer-Verlag, 2006.
11. D. Freeman, M. Scott, and E. Teske. *A Taxonomy of pairing–friendly elliptic curves*. Cryptology ePrint Archive, Report 2006/372 .
12. R. Granger, D. Page, and N.P. Smart. *High Security Pairing-Based Cryptography. Revisited*. ANTS-VII, LNCS 4076, pp.480-494, Springer-Verlag, 2006.
13. D. Hankerson, A. Menezes, and S. Vanstone. *Guide to elliptic curve cryptography*. Springer-Verlag New York, 2004.
14. F. Hess, N.P. Smart, and F. Vercauteren. *The Eta Pairing Revisited*. IEEE Transactions on Information Theory, vol 52, pp. 4595-4602, Oct. 2006. Also available from http://eprint.iacr.org/2006/110.
15. N. Koblitz and A. Menezes. *Pairing-Based Cryptography at High Security Levels*. IMA Int. Conf. LNCS 3796, pp.13-36, Springer-Verlag, 2005.
16. A. K. Lenstra. *Unbelievable Security. Matching AES Security Using Public Key Systems*. Asiacrypt'2001, LNCS 2248, pp.67-86, Springer-Verlag, 2001.
17. R. Lidl and H. Niederreiter. *Finite Fields*. Cambridge University Press, 1997.
18. S. Matsuda, N. Kanayama, F. Hess, and E. Okamoto. *Optimised Versions of the Ate and Twisted Ate Pairings*. http://eprint.iacr.org/2007/013.pdf, to appear in the Eleventh IMA International Conference on Cryptography and Coding.
19. A. Miyaji, M. Nakabayashi, and S. Takano. *New explicit condition of elliptic curve trace for FRreduction*. IEICE Trans. Fundamentals, Vol. E84 A, No.5, May 2001.
20. *Multiprecision Integer and Rational Arithmetic C/C++ Library*. http://www.shamus.ie/index.php