Accelerating the Final Exponentiation in the Computation of the Tate Pairings

Taechan Kim, Sungwook Kim, and Jung Hee Cheon

Department of Mathematical Sciences and ISaC-RIM,
Seoul National University, Seoul, 151-747, Korea
{tckim1458,kim.sungwook0630}@gmail.com,jhcheon@snu.ac.kr

Abstract. Tate pairing computation consists of two parts: Miller step and final exponentiation step. In this paper, we investigate how to accelerate the final exponentiation step. Consider an order r subgroup of an elliptic curve defined over $\mathbb{F}_q$ with embedding degree k. The final exponentiation in the Tate pairing is an exponentiation of an element in $\mathbb{F}_{q^k}$ by $(q^k-1)/r$. The hardest part of this computation is to raise to the power $\lambda := \varphi_k(q)/r$. Write it as $\lambda = \lambda_0 + \lambda_1 q + \cdots + \lambda_{d-1}q^{d-1}$ in the q-ary representation. When using multi-exponentiation techniques with precomputation, the final exponentiation cost mostly depends on $\kappa(\lambda)$, the size of the maximum of $|\lambda_i|$.

In many parametrized pairing-friendly curves, the value κ is about $\left(1-\frac{1}{\rho\varphi(k)}\right)\log q$ where $\rho = \log q/\log r$, while random curves will have $\kappa \approx \log q$. We analyze how this small κ is obtained for parametrized elliptic curves, and show that $\left(1-\frac{1}{\rho\varphi(k)}\right)\log q$ is almost optimal in the sense that for all known construction methods of parametrized pairing-friendly curves it is the lower bound. This method is useful, but has a limitation that it can only be applied to only parametrized curves and excludes many of elliptic curves.

In the second part of our paper, we propose a method to obtain a modified Tate pairing with smaller κ for *any elliptic curves*. More precisely, our method finds an integer m such that $\kappa(m\lambda) = \left(1-\frac{1}{\rho\varphi(k)}\right)\log q$ efficiently using lattice reduction. Using this modified Tate pairing, we can reduce the number of squarings in the final exponentiation by about $\left(1-\frac{1}{\rho\varphi(k)}\right)$ times from the usual Tate pairing. We apply our method to several known pairing friendly curves to verify the expected speedup.

Key words: Tate pairing, bilinear maps, final exponentiation, optimal pairing, pairing-friendly curves, elliptic curves, Miller length

1 Introduction

Non-degenerate bilinear pairings have played a key role in public-key cryptography since they have been used to construct identity-based encryption schemes [2] and one-round three-way key exchange protocols [12]. Because the performance of pairing-based cryptosystems relies heavily on the efficiency of pairing computation, the development of efficient pairings has been an important mathematical issue in cryptographic research areas.

The desired pairings are obtained from the Weil and Tate pairings defined on the rational points on elliptic curves over finite fields. Especially the most widely used pairing is the Tate pairing. When given an elliptic curve E defined over a finite field $\mathbb{F}_q$ and two points P and Q on $E(\mathbb{F}_q)$, the value of the Tate pairing at (P, Q) is given by $e(P,Q) = f_{r,P}(Q)^{\frac{q^k-1}{r}}$ where $f_{r,P} \in \mathbb{F}_{q^k}[x, y]$ where k is the embedding degree. The algorithm that computes Tate pairing consists of two steps. One is to compute $f_{r,P}(Q)$ using Miller's algorithm, called the Miller step [15], and the other is to exponentiate by $\frac{q^k-1}{r}$, called the final exponentiation step.

There have been numerous works on shortening the loop of Miller's algorithm, including Ate pairings [10], R-ate pairings [14], and optimal pairings [24], to name a few (For more variants of Tate pairings, refer to [24]). Especially, in [24], Vercauteren suggested a method to obtain a shortest Miller length for *any pairing-friendly elliptic curves*, called optimal pairings. On accelerating the final exponentiation step, however, only little work has been done [21].

On supersingular curves, the final exponentiation is relatively easy compared to the Miller step. On ordinary curves, however, the final exponentiation step takes about 40–50% of the whole Tate pairing computation [1]. The final exponentiation in Tate pairing is an exponentiation of an element in $\mathbb{F}_{q^k}$ by $(q^k-1)/r$. The hardest part of this computation is to raise power by $\lambda := \varphi_k(q)/r$. Write it as $\lambda = \lambda_0 + \lambda_1 q + \cdots + \lambda_{d-1}q^{d-1}$ in the q-ary representation. When using Multi-exponentiation technique, one can compute the exponentiation by λ with $\log_2 q$ squarings and $(\log_2 q)/w + 2^{dw}$ multiplications with $O(2^{dw})$ storage. In this paper, we focus on how to reduce the number of squaring, which is directly related to $\kappa(\lambda)$, the size of $\max_i |\lambda_i|$.

Interestingly, we observed that for most existing families of pairing-friendly curve $\kappa(\lambda)$ is much less than $\log q$, for instance about $(\log q)/2$ for $k = 6$ and about $3(\log_2 q)/4$ for $k = 12$. Thus our concerns in this work have two folds: One is to investigate when parameterized families of pairing-friendly curves have small $\kappa(\lambda)$'s and establish a formula of the smallest $\kappa(\lambda)$, and the other is to find a general way to speed-up the final exponentiation, that is, how to reduce $\kappa(\lambda)$ for any given pairing parameters.

Our contributions: Consider an order r subgroup of an elliptic curve $E(\mathbb{F}_{q^k})$ with embedding degree k, and let $\rho = \log q/\log r$. First, by analyzing previous construction methods, we give a sufficient condition that parametrized pairing-friendly elliptic curves have $\kappa(\lambda) = \left(1 - \frac{1}{\rho\varphi(k)}\right)\log q$. Furthermore, we show that this is the lower bound of $\kappa(\lambda)$ under a certain condition, which is satisfied by all the known construction methods of parametrized pairing-friendly elliptic curves. Second, we propose a method to obtain a modified Tate pairing with smaller κ for any elliptic curves. More precisely, our method finds an integer m with $\gcd(m, r) = 1$ such that $\kappa(m\lambda) = \frac{1}{d}\log\varphi_k(q)/r$, which is about $\left(1 - \frac{1}{\rho\varphi(k)}\right)\log q$, efficiently using lattice reduction. [1] In this case, $\bar{e}(P, Q) := e(P, Q)^m$ defines a non-degenerate bilinear pairing. Using this modified Tate pairing, we can reduce the number of squarings in the final exponentiation by $\left(1 - \frac{1}{\rho\varphi(k)}\right)$ times from the usual Tate pairing. We verify our argument by applying to the parameters suggested by Dupont, Enge, and Morain [6] and by Park and Lee [19].

Outline of the paper: This paper is organized as follows. In Section 2, we briefly introduce some backgrounds of pairings, pairing-friendly curves, and exponentiation method we use to analyze the number of squarings in the final exponentiation step. In Section 3, we give the analysis on parameterized families of paring-friendly curves in the sense of the final exponentiation-efficiency. In Section 4 we propose a general method to accelerate the final exponentiation and present examples in Section 5. Finally we conclude in Section 6.

[1] In [13], Kim showed that the minimum value of $\kappa(m\lambda)$ is about $\left(1 - \frac{1}{\rho\varphi(k)}\right)\log q$ when m runs though all the integers relatively prime to r. It is interesting that this bound almost equals the lower bound in the first part.

2 Preliminaries

Throughout this paper, we denote $\log_2(\cdot)$ by $\log(\cdot)$.

2.1 Pairings

Let E be an elliptic curve defined over $\mathbb{F}_q$ where $q = p^n$ for some prime p and a positive integer n. For any extension field L of $\mathbb{F}_q$, $E(L)$ denotes the set of L-rational points on E, *i.e.,* the points with coordinates in L, together with the point at infinity ∞. Then $E(L)$ forms a group with identity ∞. Let $\#E(L)$ be the order of this group. Now consider a large prime r dividing $\#E(\mathbb{F}_q)$. Let k be an embedding degree, *i.e.,* the smallest positive integer such that $r \mid q^k - 1$. Consider the r-torsion subgroup $E(\mathbb{F}_{q^k})[r]$. The Tate pairing is a well-defined non-degenerate bilinear map

$$\begin{aligned}\langle\cdot,\cdot\rangle : E(\mathbb{F}_{q^k})[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) &\to \quad \mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^r\\ \big(P, Q + rE(\mathbb{F}_{q^k})\big) \quad &\mapsto \langle P, Q\rangle_r = f_{r,P}(D),\end{aligned}$$

where D is a divisor equivalent to $(Q) - (\infty)$ and $f_{r,P}$ is a function with divisor

$$div(f_{r,P}) = r(P) - (rP) - (r-1)(\infty).$$

Since the image of the pairing is represented by a coset element, to avoid this one can use the reduced Tate pairing

$$e(P,Q) = f_{r,P}(D)^{(q^k-1)/r}.$$

Futhermore, if $(u_\infty f_{r,P})(\infty) = 1$ for some uniformizer u_∞ at ∞, we say that $f_{r,P}$ is normalized. In the case that $f_{r,P}$ is normalized one can simply work with point Q instead of using divisor D

$$e(P,Q) = f_{r,P}(Q)^{(q^k-1)/r}.$$

From now on, we call the function $f_{r,P}$ Miller function and always assume that it is normalized.

Miller algorithm computes Miller function in $\log r$ operations, called Miller length. As in [10, 14, 24], Miller length can be further reduced by defining new pairings based on the Tate pairing. All those variations of the Tate pairing have Miller length at least $\log r/\varphi(k)$. On this line of research, Vercauteren defined the notion of optimal pairings which achieves $\log r/\varphi(k)$ Miller length and proposed an algorithm to obtain a pairing with optimal Miller length for any parametrized pairing-friendly elliptic curve. The notion of pairing-friendly curves will be introduced in the next subsection.

2.2 Pairing-Friendly Elliptic Curves

For the security of pairing-based cryptosystems, the discrete logarithm problems (DLP) in the group $E(\mathbb{F}_q)$ and in the multiplicative group $\mathbb{F}_{q^k}^*$ must be infeasible. To avoid DL attack on $E(\mathbb{F}_q)$, r must be sufficiently large where r is the largest prime dividing $\#E(\mathbb{F}_q)$. And q^k should be chosen large enough so that index calculus attack is infeasible. So k needs to be large enough to avoid index calculus attack but small enough for efficient pairing implementation in extension field arithmetic. Thus in pairing-based cryptography one must find elliptic curves with sufficiently large subgroup of order r and small embedding degree k. We call them pairing-friendly curves. Formal definition is as follows.

Definition 1 ([7]). *Suppose that E is an elliptic curve defined over a finite field $\mathbb{F}_q$. E is said to be pairing-friendly if*

– *there is a prime $r \geq \sqrt{q}$ dividing $\#E(\mathbb{F}_q)$, and*
– *the embedding degree of E with respect to r is less than* $(\log_2 r)/8$.

In the construction of pairing-friendly curves, one first find t, r, q such that there exists an elliptic curve E defined over $\mathbb{F}_q$ that has trace t and a subgroup of order r with prescribed embedding degree k, then uses the complex multiplication method to find an elliptic curve equation.

Definition 2 ([7]). *Let $t(x), q(x), r(x)$ be polynomials with rational coefficients where $q(x) = p(x)^n$ for some polynomial $p(x)$ and some positive integer n. If there is an elliptic curve E defined over $\mathbb{F}_{q(x_0)}$ with trace $t(x_0)$ that has a subgroup of order $r(x_0)$ for some integer x_0, then we say that E is a curve in family (t, r, q) or (t, r, q) parameterizes a family of elliptic curves with embedding degree k. Here $p(x)$ and $r(x)$ represent primes.*

In ordinary pairing friendly curves defined over an extesion field $\mathbb{F}_{p^n}$ with $n > 1$, result values of the Tate pairing could be contained in a smaller embedding field (for example, $\mathbb{F}_{p^k}$ in the worst case) than expected, *i.e.,* $\mathbb{F}_{p^{nk}}$ [9]. To avoid this potential security loss of the DLP in the embedding field ordinary pairing friendly elliptic curves are prefered to be defined over a prime field $\mathbb{F}_p$. Thus, in the remainder of this paper, we deal with only ordinary elliptic curves defined over a prime field.

2.3 Exponentiation Method

The final exponent appearing in the Tate pairing is of the form $(p^k - 1)/r$. The exponent splits into

$$(p^k - 1)/r = [(p^k - 1)/\Phi_k(p)] \cdot [\Phi_k(p)/r],$$

where $\Phi_k(x)$ is the k-th cyclotomic polynomial. By definition of the cyclotomic polynomial

$$(p^k - 1)/\Phi_k(p) = \prod_{j|k, j\neq k} \Phi_j(p).$$

Note that $\Phi_j(x)$ is a polynomial in x with coefficients in $\{-1, 0, 1\}$ for $j < 105$ [11]. Thus raising to the exponent $\Phi_j(p)$ takes only a few Frobenius mapping and some inversions in field arithmetic. Furthermore one can replace an inversion with a few Frobenius mappings and multiplications [1]. Hence, the exponentiation by $(p^k - 1)/\Phi_k(p)$ can be done easy. In this paper, we focus on the exponentiation by $\Phi_k(p)/r$.

Define $\lambda := \Phi_k(p)/r$ and express λ as base p representation $\lambda = \sum_{i=0}^{\ell-1} \lambda_i p^i$ where $l = \lceil \log_p \lambda \rceil$. Then

$$g^\lambda = g^{\lambda_0}(g^p)^{\lambda_1} \cdots (g^{p^{\ell-1}})^{\lambda_{\ell-1}}$$

where the element g to be exponentiated is not a fixed element, but depends on the input P and Q. Without any notification all the exponentiation in this paper is considered to be computed using multi-exponentiation. Note that calculating g^{p^i} can be done easily using Frobenius map when g is an element of a finite field with characteristic p.

When ignoring p-power computation, computing g^λ takes at most $(\log_2 p)$ squarings and $(\log_2 p)$ multiplications to compute the exponentiation. Note that $2^\ell - \ell - 1$ multiplications are required to compute $g^{i_0}(g^p)^{i_1} \cdots (g^{p^{\ell-1}})^{i_{\ell-1}}$ where $i_j \in \{0, 1\}, j = 0, 1, \ldots, l-1$ for precomputation. In fact the number of squarings is related with the bit length of λ_i's. More precisely, an exponentiation by λ requires $\max_i(\log_2 \lambda_i)$ squarings. Furthermore, if we use the width w sliding window method, the number of multiplications reduces to $(1/w) \cdot \log p$ with 2^{dw} precomputed elements stored. This leads us to a natural question, that is, how to reduce the maximum size of λ_i.

3 Polynomial representation of the base-p coefficients

For any given integer λ, the coefficients of λ in the base-p representation have almost same size with base p on average. In this case, an exponentiation by λ has almost $\log_2 p$ squarings. However for many families of pairing friendly curves the number of squarings is quite smaller than $\log_2 p$.

As an instance, let us consider the final exponentiation step of the BN family of curves which has embedding degree $k = 12$. The final exponent $\lambda(x)$ is equal to $(p(x)^4 - p(x)^2 + 1)/r(x)$. Write $\lambda(x)$ as the base-$p(x)$ representation, say $\lambda(x) = \lambda_0(x)+\lambda_1(x)p(x)+\lambda_2(x)p(x)^2+\lambda_3(x)p(x)^3$, where

$$\begin{aligned}\lambda_3(x) &= 1,\\ \lambda_2(x) &= 6x^2 + 1,\\ \lambda_1(x) &= -36x^3 - 18x^2 - 12x + 1,\\ \lambda_0(x) &= -36x^3 - 30x^2 - 18x - 2.\end{aligned}$$

For the choice of $x = -4647714815446351873$, p is 254-bit and both λ_0 and λ_1 are 192-bit (*i.e.,* $\lambda_0, \lambda_1 \approx p^{192/254}$). Thus the required number of squarings are 192, not 254. Roughly speaking, this comes from the fact that $\lambda_0(x)$ and $\lambda_1(x)$ has small coefficients so that they are close to X^3 rather than X^4 for a large number X.

The above example shows that the polynomial representations of $\lambda(x)$ may give advantages in the final exponentiation step. In this section we examine polynomial representations of coefficients and investigate the conditions of coefficients under which the final exponentiation is efficiently computable. This gives another view point on parameterized families of pairing-friendly curves.

Through this section, we use notations d_f, $LC(f)$, and $||f||_\infty$ for a polynomial $f(x) = f_0 + f_1x + \cdots + f_nx^n$ which denote the degree of f, the leading coefficient f_n of f, and $\max\{|f_0|, \ldots, |f_n|\}$, respectively. Sometimes we simply write f as a evaluated value of $|f(x)|$ at $x = X$.

As indicated above the size of the value of $f(x)$ at $x = X$ for large X is determined by its degree. The following lemma asserts this.

Lemma 1. *Suppose $f(x) = f_nx^n + \cdots + f_1x + f_0$, $f_n \neq 0$. For any given $\epsilon > 0$, if $|x| = X$ is large so that $X \geq \frac{K}{\epsilon|f_n|} > 1$, then*

$$(1 - \epsilon)|f_n|X^n \leq |f(x)| \leq (1 + \epsilon)|f_n|X^n.$$

Here, $K := |f_{n-1}| + \cdots + |f_1| + |f_0|$.

Proof. Let $|f(x)| = |x|^n \cdot \left|f_n + \frac{f_{n-1}}{x} + \cdots + \frac{f_0}{x^n}\right|$, then by triangle inequality,

$$X^n \left(|f_n| - \left|\frac{f_{n-1}}{x} + \cdots + \frac{f_0}{x^n}\right|\right) \leq |f(x)| \leq X^n \left(|f_n| + \left|\frac{f_{n-1}}{x} + \cdots + \frac{f_0}{x^n}\right|\right).$$

From the assumption

$$\left|\frac{f_{n-1}}{x} + \cdots + \frac{f_0}{x^n}\right| \leq \frac{|f_{n-1}| + \cdots + |f_0|}{X} = \frac{K}{X} \leq \epsilon \cdot |f_n|.$$

Thus

$$(1 - \epsilon)|f_n|X^n \leq |f(x)| \leq (1 + \epsilon)|f_n|X^n.$$

□

If the $X = |x|$ is sufficiently large i.e. ϵ is close to 0, then $|f(x)|$ becomes asymptotically close to $|f_n|X^n$. Thus by the lemma we can regard $|f(X)|$ as $|LC(f)| \cdot |X|^{d_f}$.

Lemma 2. *Let $(p(x), r(x), t(x))$ be a family of pairing friendly curves with embedding degree k. Let $\varphi := \varphi(k)$ and $\lambda(x) := \Phi_k(p(x))/r(x)$. And let $\lambda_0(x)+\lambda_1(x)p(x)+\cdots\lambda_{\varphi-1}(x)p(x)^{\varphi-1}$ be the base-$p(x)$ representation of $\lambda(x)$. If $X^{-\alpha_i} \leq |LC(\lambda_i)| \leq X^{\alpha_i}$ and $X^{-\beta} \leq |LC(p)| \leq X^{\beta}$ for $\alpha_i, \beta > 0$, then the number of squarings, denoted by κ, to exponentiate by $\lambda(x)$ is bounded as follows,*

$$\frac{\max_i\{d_{\lambda_i}\} - \alpha_i}{d_p + \beta} \log p \leq \kappa \leq \frac{\max_i\{d_{\lambda_i}\} + \alpha_i}{d_p - \beta} \log p.$$

Proof. By the assumption, for sufficiently large X

$$\begin{aligned} X^{-\alpha_i} &\leq |LC(\lambda_i)| \leq X^{\alpha_i}, \\ X^{-\beta} &\leq\ |LC(p)| \leq\ X^{\beta}. \end{aligned}$$

We have

$$\begin{aligned} X^{d_{\lambda_i}-\alpha_i} &\leq |\lambda_i(x)| \leq X^{d_{\lambda_i}+\alpha_i}, \\ X^{d_p-\beta} &\leq\ |p(x)| \leq\ X^{d_p+\beta}. \end{aligned}$$

Thus

$$\frac{d_{\lambda_i} - \alpha_i}{d_p + \beta} \leq \frac{\log \lambda_i}{\log p} \leq \frac{d_{\lambda_i} + \alpha_i}{d_p - \beta}.$$

Since $\kappa = \max_i \log \lambda_i$, the remains of proof is obvious. □

Note that if α_i and β are sufficiently small so that $|\lambda_i(x)| \approx X^{d_{\lambda_i}}$ and $|p(x)| \approx X^{d_p}$, then we may assume that $\kappa \approx \frac{\max_i\{d_{\lambda_i}\}}{d_p} \log p$. Thus Lemma 2 implies that if the coefficients of $\lambda_i(x)$ and $p(x)$ are well-bounded then family accelerates the computation of final exponentiation step. This let us consider the specific class of families of pairing-friendly curves as below.

Definition 3. *Let $(p(x), r(x), t(x))$ be a family of pairing friendly curves. Let k be the embedding degree and $\lambda(x) := \Phi_k(p(x))/r(x)$. Let $\lambda(x) = \lambda_0(x)+\lambda_1(x)p(x)+\cdots\lambda_{\varphi-1}(x)p(x)^{\varphi-1}$ be the polynomial representations of coefficients in the base p. If κ is equal to $\frac{\max_i\{d_{\lambda_i}\}}{d_p} \log p$ then we say that the family is final-exponent friendly (FE-friendly).*

We note that in many existing families $\lambda_i(x)$'s have small coefficients. Before precisely analyzing the final exponentiation-efficiency of polynomial representations, we give an almost alternative expression of polynomial representations. The expression is useful to have some intuition on in which condition the polynomial representations show the superior final exponentiation-efficiency to numerical representations.

Recall that r is prime that divides the order of elliptic curve group $\#E(\mathbb{F}_p) = p + 1 - t$ where t is the trace of Frobenius map. Thus we can write $p+1-t = hr$ *i.e.*, $p = hr+(t-1) = hr + u$ for some cofactor h. By Hasse's bound, $|u + 1| < 2\sqrt{p}$.

Lemma 3. *Let $p(x) = h(x)r(x) + u(x)$, then*

$$\begin{aligned} \frac{p(x)^i - u(x)^i}{r(x)} &= h(x) \sum_{j=0}^{i-1} p(x)^j \cdot u(x)^{i-j-1} \\ &= h(x)(p(x)^{i-1} + u(x)p(x)^{i-2} + \cdots + u(x)^{i-1}), \end{aligned}$$

for $i > 1$ and $\frac{p(x)^i - u(x)^i}{r(x)} = h(x)$ for $i = 1$.

Proof. In the proof we abbreviate polynomial $f(x)$ simply to f. The proof uses an induction on i. If $i = 1$ then it is obvious. For $i > 1$, note that

$$\frac{p^{i+1} - u^{i+1}}{r} = \frac{p(p^i - u^i) + u^i(p-u)}{r}.$$

By induction hypothesis, the remains of proof is obvious. □

Let $f(x)$, $g(x)$ be polynomials with rational coefficients. We denote by$\lfloor f(x)/g(x) \rfloor$ the quotient when $f(x)$ divided by $g(x)$. For example, $\lfloor \frac{ax^2+bx+c}{x} \rfloor = ax + b$. Now we have an alternative expression of polynomial representations.

Lemma 4. *Let* $\lambda(x) := \frac{\Phi_k(p(x))}{r(x)}$, *then*

$$\lambda(x) = h(x)\left(p(x)^{\varphi-1} + \left\lfloor \frac{\Phi_k(u(x))}{u(x)^{\varphi-1}} \right\rfloor p(x)^{\varphi-2} + \cdots + \left\lfloor \frac{\Phi_k(u(x))}{u(x)} \right\rfloor\right) + \frac{\Phi_k(u(x))}{r(x)}.$$

Proof. Let $\Phi_k(x) := x^{\varphi} + a_{\varphi-1}x^{\varphi-1} + \cdots + a_1x + a_0$, where $\varphi := \varphi(k)$. Simply write $f(x)$ as f.

$$\begin{aligned} \frac{\Phi_k(p)}{r} &= \frac{p^{\varphi} + a_{\varphi-1}p^{\varphi-1} + \cdots + a_1p + a_0}{r} \\ &= \frac{(p^{\varphi} - u^{\varphi}) + a_{\varphi-1}(p^{\varphi-1} - u^{\varphi-1}) + \cdots + a_1(p-u)}{r} + \frac{\Phi_k(u)}{r} \end{aligned}$$

Using Lemma 3, the remains of proof is just calculations. □

We should note that $\lambda(x)$ in the above lemma is not the perfect base-p representation since the degree of $\lfloor \frac{\Phi_k(u(x))}{u(x)^i} \rfloor$ may exceed or be equal to the degree of $p(x)$ for some i. However, when $\varphi = 2$ or some specific cases overflow does not happen. Now let us analyze the case $\varphi = 2$, *i.e.,* $k = 3, 4, 6$. Let $\Phi_k(x) = x^2 + ax + b$, where $a, b \in \{0, \pm 1\}$. From Lemma 4, we see that

$$\Phi_k(p(x))/r(x) = h(x)p(x) + \{h(x)(u(x) + a) + (u(x)^2 + au(x) + b)/r(x)\}.$$

Note that $d_u < d_r \le d_p$ and

$$\begin{aligned} \deg\{h(x)(u(x) + a) + (u(x)^2 + au(x) + b)/r(x)\} &= \max\{d_h + d_u, 2d_u - d_r\} \\ &= d_h + d_u \\ &= (d_p - d_r) + d_u \\ &\le d_p - 1, \end{aligned}$$

where the second equality comes from

$$(d_h + d_u) - (2d_u - d_r) = d_h + d_r - d_u = d_p - d_u \ge 0.$$

Thus if we let $\lambda_1(x)p(x) + \lambda_0(x)$ be the base-p representation of $\Phi_k(p(x))/r(x)$, then $\lambda_1(x) = h(x)$ and $\lambda_0(x) = h(x)(u(x) + a) + (u(x)^2 + au(x) + b)/r(x)$. So, families of the embedding degree k with $\varphi(k) = 2$ yields the efficient final exponentiation step if $LC(h)$ and $LC(hu) = LC(h)LC(u)$ are both small.

For a larger $\varphi(k)$, it seems hard to control $LC(\lambda_i)$'s because of huge coefficients explosion and frequent overflows occuring in the computation of $\Phi_k(u(x))/(u(x)^i)$'s and $\Phi_k(u(x))/r(x)$ of Lemma 4. However, one can expect that if $\varphi(k)$, $||q||_\infty$, $||r||_\infty$, and $||u||_\infty$ are small enough, so $LC(\lambda_i)$'s are.

Now we are in a position to describe the lower bound of the number of squarings in the final exponentiation for the polynomial representations.

Theorem 1. *Suppose $(p(x), r(x), t(x))$ is a family of FE-friendly curves. Let $\rho := d_p/d_r$. If $\max\{d_{\lambda_i} : i = 0, 1, \ldots, \varphi - 1\} \geq d_p - \frac{d_r}{\varphi}$, then*

$$\kappa \geq \left(1 - \frac{1}{\rho\varphi}\right) \log p(x).$$

Proof. By Definition 3, $\kappa = \frac{\max_i\{d_{\lambda_i}\}}{d_p} \log p \geq \frac{d_p - \frac{d_r}{\varphi}}{d_p} \log p \geq \left(1 - \frac{1}{\rho\varphi}\right) \log p$. □

At first sight the bound in Theorem 1 may look unnatural. However, this bound is captured in most cases. More precisely, with high probability $\max_i\{d_{\lambda_i}\} = d_p - 1$ in most cases. And all known methods to construct the family of pairing friendly curves use a irreducible polynomial $r(x)$ to define the extexstion field $L := \mathbb{Q}[x]/(r(x))$ in order that it contains $\mathbb{Q}(\zeta_k)$ with k-th primitive root of unity ζ_k. Thus all the known families of curves satisfy $d_r \geq \varphi(k)$.Then,

$$\kappa = \frac{d_p - 1}{d_p} \log p = \left(1 - \frac{1}{d_p}\right) \log p = \left(1 - \frac{1}{\rho d_r}\right) \log p \geq \left(1 - \frac{1}{\rho\varphi}\right) \log p.$$

However, if $\max_i\{d_{\lambda_i}\} = d_p - N$ for $N \geq 2$, the bound can be overcome. For example, in the case that $d_r = \varphi$, we have

$$\kappa = \left(1 - \frac{N}{\rho\varphi}\right) \log p < \left(1 - \frac{1}{\rho\varphi}\right) \log p.$$

But the case that $\max_i\{d_{\lambda_i}\} \leq d_p - 2$ looks quite exceptional. To the best of our knowledge, there is no known family of curves which overcomes the bound in Theorem 1. We leave finding this family of curves as a further research work.

Example 1. Consider the BN family of curves again. BN curve has $k = 12$ and $\rho = 1$. The number of squarings is expected to be $\left(1 - \frac{1}{\varphi(12)}\right) \log p = (3/4) \log p$. In fact as seen in the front of this section, the required squarings are $192 \approx \frac{3}{4} \cdot 254$.

Example 2. Consider the cyclotomic family of curves given by [7] (Construction 6.2) with embedding degree k.

$$\begin{aligned} r(x) &= \Phi_{4k}(x), \\ t(x) &= -x^2 + 1, \\ p(x) &= \tfrac{1}{4}\left(x^{2k+4} + 2x^{2k+2} + x^{2k} + x^4 - 2x^2 + 1\right). \end{aligned}$$

Let us compute $\Phi_k(p)/r$ using Lemma 4, then $\Phi_k(u)/r = 1$ and

$$\begin{aligned} \max_i\{d_{\lambda_i}\} &= d_h + (\varphi(k) - 1)d_u \\ &= (2k + 4 - \varphi(4k)) + (\varphi(k) - 1) \cdot 2 \\ &= 2k + 2 < 2k + 4. \end{aligned}$$

In this case, $\max_i\{d_{\lambda_i}\} = d_p - 2$. However the number of squarings is expected to be $\frac{\max_i\{d_{\lambda_i}\}}{d_p} \log p = \left(1 - \frac{1}{\rho\varphi(k)}\right) \log p = \frac{k+1}{k+2} \log p$.

Remark 1. Although one exponentiate by λ using addition chain as described in [21], the number of squarings has no much difference with multi-exponentiation method. Since in the addition chain method, one computes $\max_i\{d_{\lambda_i}\}$ numbers of exponentiation by x, thus total number of squarings is $\max_i\{d_{\lambda_i}\}(\log x) \approx (\max_i\{d_{\lambda_i}\}/d_p) \log p$ which is exactly same with $\max_i \log \lambda_i$, the number of squarings when multi-exponentiation is used.

4 Reducing the size of base p coefficients

In this section, we propose a general method to reduce κ in computing the final exponentiation by $\lambda := \Phi_k(p)/r$. The main idea is to reduce the maximum size of coefficients of base p representation of λ since κ depends on the maximum bit length of λ_i's. Since the pairing $e(P,Q)^m$ also defines a non-degenerate bilinear pairing map with m relatively prime to r, we use the exponent $m\lambda$ instead of λ. Using lattice basis reduction algorithm one can find $m\lambda$ whose coefficients in base p representation are small. Throughout this section p, r, t are integers not polynomials.

Observations Since the reduced Tate pairing is non-degenerate, the map $\bar{e}$ also defines non-degenerate bilinear pairing

$$\bar{e}(P,Q) = e(P,Q)^m = f_{r,P}(Q)^{m(p^k-1)/r},$$

if $\gcd(r,m) = 1$. Let $g := f_{r,P}(Q)^{(p^k-1)/\Phi_k(p)}$, then $\bar{e}(P,Q) = g^{m\lambda}$. We want to find $m\lambda$ with $\gcd(r,m) = 1$ such that

$$m\lambda = \sum_{i=0}^{d-1} v_i p^i$$

where v_i's are as small as possible. (The choice of d will be given later.) With abuse of notations, we write $\sum_{i=0}^{d-1} v_i p^i = (v_0, v_1, \ldots, v_{d-1})$.

Reducing the coefficients of base p representation Motivated by [24], $m\lambda$ with small coefficients in base p representation can be obtained by using lattice basis reduction algorithm. Let L be the lattice of dimension d spanned by rows of the matrix

$$L := \begin{pmatrix} \lambda & 0 & 0 & \cdots & 0 \\ -p & 1 & 0 & \cdots & 0 \\ -p^2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & & \\ -p^{d-1} & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

It is easily verified that $v := (v_0, v_1, \cdots, v_{d-1}) \in L$ if and only if $\sum_{i=0}^{d-1} v_i p^i = m\lambda$ for some integer m. Now finding $m\lambda$ with small coefficients reduces to find the short vector in lattice L. By Minkowski's theorem [16] there is a shortest vector v in L satisfies $||v||_\infty \leq |\det(L)|^{1/d}$ where $||v||_\infty = \max\{|v_i| : i = 0, 1, \ldots, d-1\}$. Then there exists $m\lambda = \sum_{i=0}^{d-1} v_i p^i$ with

$$\max\{|v_i|\} \leq |\det(L)|^{1/d} = |\lambda|^{1/d} = \left(\frac{\Phi_k(p)}{r}\right)^{1/d} \approx (p^{\varphi(k)-1/\rho})^{1/d}.$$

Since $\Phi_k(p) \equiv 0 \bmod \lambda$, any powers p^i for $i \geq \varphi(k)$ can be represented by a linear combination of $1, p, \ldots, p^{\varphi(k)-1}$ modulo λ and since $\Phi_k(p) = r\lambda$ has small coefficietnts in base p representation to avoid degenerate pairing maps, it sufficies to consider the lattice with dimension $d = \varphi(k)$. Thus κ reduces to at most $[(\rho \cdot \varphi(k) - 1)/d\rho] \log p = \left(1 - \frac{1}{\rho\varphi(k)}\right) \log p$.

Finding a shortest vector in L Finding a short vector in a given lattice L can be done using LLL algorithm. Let $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d\}$ be an ordered basis of lattice L. Then LLL algorithm returns the reduced basis $\{\bar{\mathbf{b}}_1, \bar{\mathbf{b}}_2, \ldots, \bar{\mathbf{b}}_d\}$ where $||\bar{\mathbf{b}}_1|| \leq ||\bar{\mathbf{b}}_2|| \leq \cdots \leq ||\bar{\mathbf{b}}_d||$. In particular, $||\mathbf{b}||$ denotes the Euclidean norm. The lemma below shows that LLL algorithm has good property.

Lemma 5. *Let $\{\bar{\mathbf{b}}_1, \bar{\mathbf{b}}_2, \ldots, \bar{\mathbf{b}}_d\}$ be a LLL reduced basis for lattice L and λ_1 be a length of the shortest vector in L. Then $||\bar{\mathbf{b}}_1|| \leq 2^{(d-1)/2} \cdot \lambda_1$*

This means that LLL algorithm can be used to find almost shortest vector for low dimensional lattice. Especially, for a lattice whose dimension is less than or equal to 4, a shortest vector can be always found quite efficiently [8, 23, 22, 18].

m* is relatively prime to *r $m\lambda$ with small coefficients in base p representation can be obtained efficiently using LLL algorithm. For nondegeneracy of the pairing m must be relatively prime to r. This is equivalent that m is not a multiplie of r since r is prime. The following lemma asserts this property.

Lemma 6. *Let $\lambda := \Phi_k(p)/r$ and $\varphi := \varphi(k)$. Suppose that r is a prime larger than $2^{\varphi(\varphi+1)}$ and p is a prime larger than 3. If $m\lambda = \sum_{i=0}^{\varphi-1} v_i p^i$ with $|v_i| \leq \lambda^{1/\varphi}$ and assume that $m = n \cdot r$ for some integer n, then n must be 0.*

Proof. We will use the inequality $(p-1)^\varphi \leq \Phi_k(p) \leq (p+1)^\varphi$ for all k. First observe that

$$\left(\frac{p}{p-1}\right)^\varphi \cdot \left(\frac{p+1}{p-1}\right) \leq 2^{\varphi+1} < r^{1/\varphi}$$

from $p/(p-1) < (p+1)/(p-1) \leq 2$ and $r > 2^{\varphi(\varphi+1)}$. From this

$$\frac{p+1}{r^{1/\varphi}} \cdot \frac{p^\varphi}{p-1} < (p-1)^\varphi.$$

Then

$$\begin{aligned} |n|\Phi_k(p) = |m\lambda| = |\textstyle\sum_{i=0}^{\varphi-1} v_i p^i| \leq \textstyle\sum_{i=0}^{\varphi-1} \lambda^{1/\varphi} p^i < \lambda^{1/\varphi} \cdot \tfrac{p^\varphi}{p-1} &\leq \left(\tfrac{(p+1)^\varphi}{r}\right)^{1/\varphi} \cdot \tfrac{p^\varphi}{p-1} \\ = \tfrac{p+1}{r^{1/\varphi}} \cdot \tfrac{p^\varphi}{p-1} \qquad < (p-1)^\varphi \quad &\leq \Phi_k(p). \end{aligned}$$

Hence $|n|\Phi_k(p) < \Phi_k(p)$ and n must be 0. □

In the pairing based cryptosystem, for the 80-bit security r is usually chosen to be 160 bits prime. In this case, if $d = \varphi(k) \leq 12$ then r is always larger than $2^{d(d+1)}$. Thus the assumption in lemma holds.

Let $m\frac{\Phi_k(p)}{r} := \sum_{i\leq 0}^{\varphi(k)-1} \lambda_i q^i$. Then [13, Theorem 4.4.1] have shown that $\max_i |\lambda_i| \geq \frac{1}{\varphi(k)} \left(\frac{\Phi_k(p)}{r}\right)^{1/\varphi(k)}$. Then we have $\max_i\{\log|\lambda_i|\} \geq \frac{1}{\varphi(k)} \log\left(\frac{\Phi_k(p)}{r}\right) - \log\varphi(k)$. This means that $\left(1 - \frac{1}{\rho\varphi(k)}\right) \log p$ is the almost optimized number of squarings in the final exponentiation step.

5 Examples

In this section we give some examples investigated by lattice basis reduction. All results satisfy the Minkowski's bounds well as we have shown that theoretically. Our approach using lattice reduction reduces the number of squarings nicely for the curves which are not in the family.

First and second example show the case when our method is applied to DEM curves which is not in the family.

Example 3. Dupont, Enge, and Morain proposed some parameters for pairing-friendly curves in [6]. The following p and r parameterize the pairing-friendly curve for $k = 5$:

$$\begin{aligned} p &= 91600022435668881297760819108273609(117 \text{ bits}),\\ r &= 1040375393410195481(60 \text{ bits}). \end{aligned}$$

Then the final exponent is of the form $\lambda = (p^4 + p^3 + p^2 + p + 1)/r = a_0 + a_1 p + a_2 p^2 + a_3 p^3$ where

$$\begin{aligned} a_0 &= 48298402242066861357969209793319103(116 \text{ bits}),\\ a_1 &= 68283809547505356824804028665198693(116 \text{ bits}),\\ a_2 &= 53294610661059016732355697881722241(116 \text{ bits}),\\ a_3 &= 88045164289610560(57 \text{ bits}). \end{aligned}$$

Note that the maximum bit length of a_0, a_1, a_2, a_3 is 116 bits. The naive implementation takes totally 115 squarings and 118 multiplications. However, our method finds $m\lambda = b_0 + b_1 p + b_2 p^2 + b_3 p^3$ where

$$\begin{aligned} b_0 &= -2868147363431539633026293965700(102 \text{ bits}),\\ b_1 &= -179610012117759028207462943(88 \text{ bits}),\\ b_2 &= 89797974551946435080337006(87 \text{ bits}),\\ b_3 &= 14058171382122118208099(74 \text{ bits}),\\ m &= 159670. \end{aligned}$$

The implementation requires total 101 squarings and 96 multiplications. Consequently our method reduces the number of squarings by 12% and the number of multiplications by 18.6%.

Example 4. Another example in [6] proposes parameters of the curves for $k = 10$:

$$\begin{aligned} p = {} & 26583877300690675075645839413139198533414446909174086061240198580 0\\ & 1080573263503000190636119494020100362575727175540808493 69(407 \text{ bits}),\\ r = {} & 25621456065075422729511299019214902729542591998892393498858941(204 \text{ bits}) \end{aligned}$$

where $\lambda = (p^4 - p^3 + p^2 - p + 1)/r$. The naive implementation requires 405 squarings and 367 multiplications. When our method is applied to the λ, computing the final exponent needs 354 squarings and 339 multiplications with

$$m = 67378873396743296140989477656148344170131747 05.$$

Thus our method reduces the number of squarings by 12.5% and the number of multiplications by 7.6%.

Next example shows the case when the lattice basis reduction is applied to the families of curves.

Example 5. Consider the BN curves with $x = -4647714815446351873$.

$$\begin{aligned} p = {} & 16798108731015832284940804142231733909889187121439069848933715\\ & 426072753864723(254 \text{ bits}),\\ r = {} & 16798108731015832284940804142231733909759579603404752749028378\\ & 864165570215949(254 \text{ bits}). \end{aligned}$$

Let $\lambda = (p^4 - p^2 + 1)/r = a_0 + a_1 p + a_2 p^2 + a_3 p^3$, then a_0 and a_1 have 192 bits. So the number of squarings is 192. After the lattice basis reduction we get $m\lambda = b_0 + b_1 p + b_2 p^2 + b_3 p^3$ where b_0 and b_2 have 190 bits with

$$m = 129607518034317099886745702645398241283.$$

As we have noted in previous section, BN curve already attains Minkowski's bound. The example shows that there is no noticeable difference by lattice reduction for FE-friendly curves.

Example 6. In [19], they proposed the method to find the parameters of pairing friendly curves which has minimal security loss against Cheon's attack on strong DH [5].

$$\begin{aligned}
p = {} & 168117645147302822689933583299826302980183409639586768617985750798 \\
& 949133918522866373(277 \text{ bits}), \\
r = {} & 744872153953490574612022965440081094285259191927642272735639568165 \\
& 35141597060317(266 \text{ bits}).
\end{aligned}$$

In this case, $\lambda = (p^4 - p^2 + 1)/r = a_0 + a_1 p + a_2 p^2 + a_3 p^3$ and a_0, a_1 have 220 bits. The reduction shows that the maximum bit length of $m\lambda$ is 210 bits, so reduces the number of squarings by 4.5%.

6 Conclusion

In this paper we have suggested a general method for reducing the number of squarings by $\left(1 - \frac{1}{\rho\varphi(k)}\right) \log p$ in the calculation of final exponentiation for the Tate pairing. We also have shown that if the maximum of degree of $\lambda_i(x)$ is greater than $\deg p - \deg r/\varphi(k)$, then the required number of squarings in the final exponentiation step is bounded below by Minkowski's bound $\left(1 - \frac{1}{\rho\varphi(k)}\right) \log p$. All the known families have this low bound, however it remains open whether one can construct a family of curves whose computation cost for squarings in the final exponentiation is much less than the Minkowski's bound.

References

1. Aranha, D. F., Karabina, K., Longa, P., Gebotys, C. H., López, J.: Faster Explicit Formulas for Computing Pairings over Ordinary Curves. In: Paterson, K.G. (ed.) Eurocrypt 2011. LNCS, vol. 6632, pp. 48–68. (2011)
2. Boneh, D., Franklin, M.K.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Berlin (2001)
3. 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, Berlin (2002)
4. Barreto, P .S. L. M., Naehrig, M.: Pairing-friendly elliptic curves of prime order. In: Preneel, B., Tavares, S. (Eds.) SAC 2005. LNCS, vol. 3897. pp. 319–331. Springer, Heidelberg (2006)
5. Cheon, J.: Security Analysis of the Strong Diffie-Hellman Problem. EUROCRYPT 2006, LNCS 4004, pp. 1-11, 2006
6. Dupont, R., Enge, A., Morain, F.: Building curves with arbitrary small MOV degree over finite prime fields. Journal of Cryptology 18, pp. 79–89 (2005)
7. Freeman, D., Scott, M., Teske, E.: A Taxonomy of Pairing-Friendly Elliptic Curves. Journal of Cryptology 23(2), pp. 224–280 (2010)
8. Gauss, C. F.: Disquistiones Arithmetica. Leipziq (1801)
9. Hitt, L.: On the minimal embedding field. In: Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T. (ed.) Pairing 2007. LNCS, vol. 4575, 294–301. Springer, Berlin (2007)
10. Hess, F., Smart, N.P., and Vercauteren, F.: The eta pairing revisited. IEEE Transactions on Information Theory 52(10), 4595–4602 (2006)
11. Isaacs, I.M.: Algebra : A Graduate Course, AMS p. 310. ISBN 9780821847992. (2009)
12. Joux, A.: A One Round Protocol for Tripartite Diffie-Hellman. In: Bosma, W. (ed.) ANTS 2000. LNCS, vol. 1838, 385–393. Springer, Berlin (2000)

13. Kim, M.: Integer Factorization and Discrete Logarithm with Additional Information. PhD Thesis. Seoul National University (2011)
14. Lee. E., Lee, H.-S., and Park, C.-M.: Efficient and Generalized Pairing Computation on Abelian Varieties. IEEE Transactions on Information Theory 55(4), 1793–1803 (2009)
15. Miller, V.S.: Short Programs for Functions on Curves. Unpulished manuscript (1986), http://crypto.stanford.edu/miller/miller.pdf
16. Minkowski, H.: Geometrie der Zahlen. Leipzig und Berlin: Druck und Verlag von B.G. Teubner (1910)
17. Miyaji, A., Nakabayashi, M., Takano, S.: New explicit conditions of elliptic curve traces for FR-reduction. IEICE Trans. Fundam. E84–A(5), 1234–1243 (2001)
18. Nguyen, P. Q., Stehlé D.: Low-Dimensional Lattice Basis Reduction Revisited. In: Buell, D. A. (ed.) ANTS 2004. LNCS, vol. 3076, pp. 338–357. Springer, Heidelverg (2004)
19. Park, C.-M., Lee, H.-S.: Pairing–friendly curves with minimal security loss by Cheon's algorithm. ETRI Journal. 33(4), 656–659 (2011)
20. Scott, M., Barreto, P.S.L.M.: Generating more MNT elliptic curves. Designs, Codes and Cryptography 38, 209–217 (2006)
21. Scott, M., Benger, N., Charlemagne, M., Perez, L. J. D., Kachisa, E.J.: On the Final Exponentiation for Calculating Pairings on Ordinary Elliptic Curves. In: Shacham, H., Waters, B. (eds.) Pairing 2009. LNCS, vol. 5671, pp. 78–88. Springer, Heidelberg (2009)
22. Semaev, I.: A 3-dimensional Lattice Reduction Algorithm. In: Silverman, J. H. (ed.) CALC 2001. LNCS, vol. 2146, pp. 181–193. Springer, Heidelberg (2001)
23. Vallée, B.: Une Approche Géométrique de la Réduction de Réseaux en Petite Dimension. PhD thesis, Université de Caen (1986)
24. Vercauteren, F.: Optimal Pairings. IEEE Transactions on Information Theory 56(1), 455–461 (2010)