Optimal Pairings on BN Curves

by

Kewei Yu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics & Optimization

Waterloo, Ontario, Canada, 2011

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Bilinear pairings are being used in ingenious ways to solve various protocol problems. Much research has been done on improving the efficiency of pairing computations. This thesis gives an introduction to the Tate pairing and some variants including the ate pairing, Vercauteren's pairing, and the R-ate pairing. We describe the Barreto-Naehrig (BN) family of pairing-friendly curves, and analyze three different coordinates systems (affine, projective, and jacobian) for implementing the R-ate pairing. Finally, we examine some recent work for speeding the pairing computation and provide improved estimates of the pairing costs on a particular BN curve.

Acknowledgements

I would like to thank all the little people who made this possible.

Dedication

This is dedicated to the one I love.

Table of Contents

List of Tables

Chapter 1

Introduction

Since 2000, non-degenerate bilinear pairings have been used in ingenious ways to solve various protocol problems that do not have efficient solutions using conventional cryptographic techniques. Among these protocols are identity-based encryption, aggregate signature schemes, and attribute-based encryption.

The desired pairings are derived from the classic Weil and Tate pairings defined on the rational points on low-embedding degree elliptic curves defined over finite fields. In the past 10 years, several families of low-embedding degree elliptic curves have been discovered. Moreover, there have been many proposal for faster pairings, including the ate pairing, the eta paring, the R-ate pairing, and Vercauteren's pairing. At present, it appears that the fastest pairing that meets the 128-bit security level is Vercauteren's pairing on Barreto-Naehrig (BN) elliptic curves.

The purpose of this thesis is to give a complete description of the mathematics required to understand Vercauteren's pairing, and the numerous optimizations available for accelerating the pairing on BN curves.

The remainder of this thesis is organized as follows. In Chapter 2, we provide some elementary background on elliptic curves, define the classic Tate pairing and describe Miller's basic algorithm for computing it. Finally, we present two fundamental pairing-based cryptographic protocols, namely the Boneh-Lynn-Shacham short signature scheme and the Boneh-Franklin identity-based encryption scheme.

In Chapter 3, we outline Vercauteren's general construction for optimal pairings, and then describe the ate pairing, the R-ate pairing, and Vercauteren's optimal pairing. We also note that the R-ate pairing is not derivable from Vercauteren's general framework.

In Chapter 4, we present the BN family of elliptic curves, define their sextic twists, and outline a method for efficiently constructing BN curves.

In Chapter 5, we give a detailed algorithm for computing the R-ate pairing on a specially-chosen BN curve. We describe techniques for efficiently performing field arithmetic in the extension fields $\mathbb{F}_{p^2}$, $\mathbb{F}_{p^6}$ and $\mathbb{F}_{p^{12}}$, and for the Miller loop and final exponentiation. We give a careful estimate of the number of $\mathbb{F}_p$ arithmetic operations needed for the pairing computation.

The purpose of Chapter 6 is to evaluate several recent papers that presented techniques for purportedly speeding up the pairing computation. In particular, we examine a WAIFI 2010 paper and an AFRICACRYPT 2010 paper by Costello, Boyd, Nieto and Wong on delaying full field multiplications, a PKC 2010 paper by Costello, Lange and Naehrig that presented new formulas using ordinary projective coordinates for the doubling operation in the Miller loop, and a Pairing 2010 paper by Lauter, Montogmery and Naehrig that uses affine coordinates instead of projective coordinates. All these papers used very crude methods to estimate the advantages of their new methods. We provide much more careful estimates of the new methods, and as a result conclude that only the new formulas by Costello, Lange and Naehrig offer speed ups over the previous methods for single pairing computation.

Chapter 2

Mathematical Background

2.1 Elliptic Curves

We begin by summarizing some essential properties of elliptic curves that will be needed in this thesis. A standard reference for this background material is Washington's book [31].

Definition 2.1.1 *An elliptic curve E over a field $\mathbb{F}$ is defined by an equation*

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{2.1}$$

where $a_1, a_2, a_3, a_4, a_6 \in \mathbb{F}$ and $\triangle \neq 0$, and where $\triangle$ is the discriminant of E and is defined as follows:

$$\triangle = -d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6$$

$$d_2 = a_1^2 + 4a_2$$

$$d_4 = 2a_4 + a_1a_3$$

$$d_6 = a_3^2 + 4a_6$$

$$d_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2.$$

If $\mathbb{F}'$ is any extension of $\mathbb{F}$, then the set of $\mathbb{F}'$-rational points on E is

$$E(\mathbb{F}') = \{(x, y) \in \mathbb{F}' \times \mathbb{F}' : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\} \cup \{\infty\}$$

where ∞ is the point at infinity. In particular, the set of all points on E is $E(\overline{\mathbb{F}})$, where $\overline{\mathbb{F}}$ is the algebraic closure of $\mathbb{F}$; we will often denote $E(\overline{\mathbb{F}})$ by E itself.

The field $\mathbb{F}$ can be the rational numbers $\mathbb{Q}$, the real numbers $\mathbb{R}$, the complex numbers $\mathbb{C}$, etc. In this thesis, we mainly consider $\mathbb{F}$ to be a finite field $\mathbb{F}_{p^k}$ with prime p and $k \geq 1$. Equation (2.1) is called the generalized Weierstrass equation. If the characteristic of the field is not 2, then we can complete the square on the left hand side and move the extra terms to the right to get:

$$\left(y + \frac{a_1 x + a_3}{2}\right)^2 = x^3 + \left(a_2 + \frac{a_1^2}{4}\right) x^2 + \left(a_4 + \frac{a_1 a_3}{2}\right) x + \left(\frac{a_3^2}{4} + a_6\right).$$

Letting $y_1 = y + a_1 x/2 + a_3/2$, the equation can be written as

$$y_1^2 = x^3 + a_2' x^2 + a_4' x + a_6',$$

for some constants a_2', a_4', $a_6' \in \mathbb{F}$. Further, if the characteristic is also not 3, then letting $x_1 = x + a_2'/3$, we get

$$y_1^2 = x_1^3 + a x_1 + b$$

for some constants a, $b \in \mathbb{F}$.

In this thesis, the elliptic curves we are mainly focusing on are the Barreto-Naehrig (BN) curves, which will be introduced in Chapter 4. These curves are defined over $\mathbb{F}_p$, where $p \neq 2, 3$. So for most of this thesis, elliptic curves will be of the form

$$y^2 = x^3 + ax + b,$$

where a, b are in some finite field $\mathbb{F}_{p^k}$. The discriminant is

$$\triangle = -16(4a^3 + 27b^2),$$

and $\triangle \neq 0$ implies that the polynomial $x^3 + ax + b$ has no multiple roots. The group law for points on the elliptic curve defined by $y^2 = x^3 + ax + b$ will be introduced in Section 2.1.1.

2.1.1 Group Law

There is a chord-and-tangent rule for adding two points on an elliptic curve. Let P and Q be two points on an elliptic curve E, with $P = (x_1, y_1)$ and $Q = (x_2, y_2)$. We use $y^2 = x^3 + ax + b$ as the curve equation. Let point $R = (x_3, y_3)$ be the sum of P and Q; then R is defined as follows. We first give a rough geometric description of this addition rule. It is understood that all vertical lines intersect the point ∞, and that the reflection of ∞ in the x-axis is ∞ itself. Now, to add P and Q, one draws the line ℓ through P and Q. The line ℓ intersects E at a third point R'. Then R is the reflection of R' about the x-axis.

We now give algebraic formulas for the group law. First assume that $P \neq Q$. If $x_1 = x_2$, then ℓ is the vertical line through P. Therefore, ℓ intersects E at ∞, and $P + Q = \infty$. If $x_1 \neq x_2$, the slope of ℓ is

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}.$$

The equation of ℓ is then

$$y = \lambda(x - x_1) + y_1.$$

Hence, $R = (x_3, -y_3)$ can be obtained by solving the system of equations

$$\begin{cases} y = \lambda(x - x_1) + y_1 \\ y^2 = x^3 + ax + b. \end{cases}$$

Substitute the line function into the curve function to get

$$(\lambda(x - x_1) + y_1)^2 = x^3 + ax + b.$$

From this, we have a cubic polynomial in x that equals 0:

$$x^3 - \lambda^2 x^2 + (a + 2\lambda^2 x_1 - 2\lambda y_1)x + (b - \lambda^2 x_1^2 + 2\lambda x_1 y_1 - y_1^2) = 0.$$

Since we already know that the three roots of the polynomial are x_1, x_2 and x_3, the polynomial must have the form

$$(x - x_1)(x - x_2)(x - x_3) = x^3 - (x_1 + x_2 + x_3)x^2 + (x_1x_3 + x_2x_3 + x_1x_2)x - x_1x_2x_3.$$

Therefore, we have

$$x_1 + x_2 + x_3 = \lambda^2$$

and obtain

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2 \\ y_3 = -(\lambda(x_3 - x_1) + y_1). \end{cases}$$

In the case that $P = Q = (x_1, y_1)$, we take the tangent line through P as the line ℓ. Taking the derivative with respect to x of the curve equation gives

$$2y\frac{dy}{dx} = 3x^2 + a.$$

If $y_1 = 0$, then ℓ is a vertical line. As before, we obtain $P + P = \infty$. If $y_1 \neq 0$, the slope λ of ℓ is

$$\lambda = \frac{dy}{dx} = \frac{3x_1^2 + a}{2y_1}.$$

Similarly to the case $P \neq Q$, we obtain

$$\begin{cases} x_3 = \lambda^2 - 2x_1 \\ y_3 = -(\lambda(x_3 - x_1) + y_1). \end{cases}$$

Finally, if $Q = \infty$, the line through P and ∞ is the vertical line which intersects E at the point $P' = (x_1, -y_1)$, where P' denotes the reflection of P in the x-axis. Hence, when we reflect P' to get $R = P + \infty$, we are back at P again. Therefore,

$$P + \infty = P$$

for all points P on E. Moreover, it is easy to see that

$$P + P' = \infty$$

for all points P on E.

Theorem 2.1.1 *The points $E(\overline{\mathbb{F}})$ on an elliptic curve E form an additive abelian group under point addition.*

Proof: The point addition on E satisfies:

1. Commutativity: Since the line through P and Q is the same as the line through Q and P, we have $P + Q = Q + P$ for all P, Q on E.

2. Existence of Identity: Since $P + \infty = P$ for all P on E, ∞ is the group identity.

3. Existence of Inverses: For any point $P = (x, y)$ on E, there exists a point $P' = (x, -y)$ on E such that $P + P' = \infty$. This point P' is also denoted as $-P$.

4. Associativity: It is not obvious that $(P + Q) + R = P + (Q + R)$ for all P, Q, R on E. Since the proof of this property is not needed in the rest of this thesis, the details are omitted; the interested reader can refer to [31]. □

Now, if E is defined over $\mathbb{F}$ and P, $Q \in E(\mathbb{F})$, then $P + Q \in E(\mathbb{F})$. This shows that $E(\mathbb{F})$ is also an abelian group.

Summarizing all the above information, we have the group law as follows.

Let $E/\mathbb{F}$ be an elliptic curve defined by equation $y^2 = x^3 + ax + b$ over field $\mathbb{F}$ whose characteristic is neither 2 nor 3.

1. Identity: $P + \infty = \infty + P = P$ for all $P \in E(\mathbb{F})$.

2. Negation: If $P = (x, y) \in E(\mathbb{F})$, then there exists $-P = (x, -y)$ such that $P+(-P) = (x + y) + (x, -y) = \infty$. Also, $-\infty = \infty$.

3. Point Addition: Let $P = (x_1, y_1) \in E(\mathbb{F})$ and $Q = (x_2, y_2) \in E(\mathbb{F})$ with $P \neq \pm Q$. Then $P + Q = (x_3, y_3)$ where

$$\begin{cases} x_3 = \left(\dfrac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2 \\ y_3 = \left(\dfrac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1. \end{cases}$$

4. Point Doubling: Let $P = (x_1, y_1) \in E(\mathbb{F})$ with $P \neq -P$. Then $2P = (x_3, y_3)$, where

$$\begin{cases} x_3 = \left(\dfrac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1 \\ y_3 = \left(\dfrac{3x_1^2 + a}{2y_1}\right)(x_1 - x_3) - y_1. \end{cases}$$

The formulas for point addition and point doubling are given in the group law. In the remainder of this thesis, for $a \in \mathbb{N}$ and $P \in E$, we denote by aP the a-fold sum of P with itself. The efficiency of these formulas will be discussed in Chapter 5. Alternate formulas for the group law can be derived by writing the points in other coordinate systems. Two such systems are introduced in Section 2.1.2.

2.1.2 Projective Coordinates and Jacobian Coordinates

The two-dimensional projective space $\mathbb{P}^2(\mathbb{F})$ over a field $\mathbb{F}$ consists of the equivalence classes of non-zero triples in $\mathbb{F} \times \mathbb{F} \times \mathbb{F}$, where triples (x_1, y_1, z_1) and (x_2, y_2, z_2) are said to be equivalent if there exists a nonzero element $\lambda \in \mathbb{F}$ such that

$$(x_1, y_1, z_1) = (\lambda x_2, \lambda y_2, \lambda z_2).$$

An equivalence class containing (x, y, z), called a projective point, will be denoted as $(x : y : z)$.

Let $P = (x : y : z)$ be a point in $\mathbb{P}^2(\mathbb{F})$. If $z \neq 0$, then $P = (x : y : z) = (x/z : y/z : 1)$. These points are called the finite points. If $z = 0$, then $P = (x : y : z)$ is called a point at infinity. In this way,

$$(x, y) \leftrightarrow (x : y : 1)$$

is a bijection between the finite points in the 2-dimensional affine plane and the finite points in the 2-dimensional projective space.

Let projective point $(x : y : z)$ with $z \neq 0$ represent the affine point $(x/z, y/z)$. Since $(x : y : z) = (\lambda x : \lambda y : \lambda z)$ for any $\lambda \in \mathbb{F}^*$, all terms in the curve equation should have the same degree. Therefore, the projective equation of the elliptic curve is

$$y^2 z = x^3 + axz^2 + bz^3.$$

For this specific curve, $z = 0$ implies $x^3 = 0$. Hence, the point at infinity corresponds to the class $(0 : 1 : 0)$.

Similarly, in Jacobian coordinates, we use $(x : y : z)$ to represent the affine point $(x/z^2, y/z^3)$. The elliptic curve equation becomes

$$y^2 = x^3 + axz^4 + bz^6.$$

Naturally, the point addition and point doubling formulas in projective coordinates and Jacobian coordinates are different from those in affine coordinates. These formulas are not unique. Different formulas and their efficiency will be discussed in Chapter 5.

2.1.3 Group Order and Torsion Points

Let E be an elliptic curve defined over $\mathbb{F}_q$. We define the order of E over $\mathbb{F}_q$ to be the number of points in $E(\mathbb{F}_q)$, denoted $\#E(\mathbb{F}_q)$. Examining the curve equation, we see that there are at most two roots y for each $x \in \mathbb{F}_q$. Hence, together with the point at infinity, the group order must be between 1 and $2q+1$. A tighter bound of the group order is given by Hasse's Theorem.

Theorem 2.1.2 *(Hasse's Theorem) Let E be an elliptic curve defined over $\mathbb{F}_q$. Then*

$$q + 1 - 2\sqrt{q} \leq \#E(\mathbb{F}_q) \leq q + 1 + 2\sqrt{q}.$$

Let t be the integer such that $\#E(\mathbb{F}_q) = q + 1 - t$. This integer is called the trace of the Frobenius endomorphism and will be introduced in Section 2.1.4. From Hasse's Theorem, we can see that $|t| \leq 2\sqrt{q}$. Since $2\sqrt{q} \ll q$ for large q, we have $\#E(\mathbb{F}_q) \approx q$. However, when a curve is used for cryptographic implementations, we are only interested in the points with specific order, namely, the n-torsion points.

For $n \in \mathbb{Z}$ define the endomorphism

$$[n] : E \to E$$

to be the multiplication-by-n map. Let $P \in E$ be an arbitrary point on the curve. If $n = 0$, then $[n]P = \infty$. If $n > 0$, then $[n]P = nP$. If $n < 0$, then $[n]P = -[-n]P$. For $n \in \mathbb{Z}$, $n \neq 0$, define the n-torsion points to be the kernel of the multiplication-by-n map

$$E[n] = \ker([n]) = \{P \in E \mid [n]P = \infty\}.$$

Moreover, $E(\mathbb{F}_{q^m})[n]$ is defined to be $E(\mathbb{F}_{q^m}) \cap E[n]$.

Obviously, $E[n] \subseteq E(\overline{\mathbb{F}}_q)$. However, the coordinates of all points in $E[n]$ are in fact contained in some finite extension of $\mathbb{F}_q$. Now, assume that $\gcd(q, n) = 1$ and define the embedding degree of E with respect to n to be the smallest integer k such that $n \mid (q^k - 1)$. A theorem by Balasubramanian and Koblitz [2] related to k is given as follows.

Theorem 2.1.3 *Let $E/\mathbb{F}_q$ be an elliptic curve with $n \mid \#E(\mathbb{F}_q)$, where n is a prime and $n \nmid (q-1)$. Then $E[n] \subseteq E(\mathbb{F}_{q^k})$ if and only if $n \mid (q^k - 1)$.*

By Theorem 2.1.3, if $n \nmid (q-1)$, then the embedding degree k is the smallest extension of $\mathbb{F}_q$ over which all n-torsion points of E are defined.

2.1.4 The Frobenius Map

The Frobenius map ϕ_q for a finite field $\mathbb{F}_q$ is defined as follows:

$$\phi_q : \overline{\mathbb{F}}_q \to \overline{\mathbb{F}}_q,$$

$$x \mapsto x^q.$$

Now let E be an elliptic curve defined over $\mathbb{F}_q$. The Frobenius map

$$\phi_q : E(\overline{\mathbb{F}}_q) \to E(\overline{\mathbb{F}}_q),$$

$$(x, y) \mapsto (x^q, y^q), \qquad \infty \mapsto \infty$$

is an endomorphism of E, called the Frobenius endomorphism. Since the qth power map is the identity on $\mathbb{F}_q$, the set of points fixed by ϕ_q is the group $E(\mathbb{F}_q)$. The Frobenius endomorphism ϕ_q of $E/\mathbb{F}_q$ satisfies

$$\phi_q^2 - [t] \circ \phi_q + [q] = 0.$$

Hence, the characteristic polynomial of ϕ_q is $X^2 - tX + q \in \mathbb{Z}[X]$.

Theorem 2.1.4 *Let $E/\mathbb{F}_q$ be an elliptic curve with $n \mid \#E(\mathbb{F}_q)$, where n is a prime and $n \nmid (q-1)$. Let $k > 1$ be the embedding degree of E with respect to n, and let ϕ_q be the Frobenius endomorphism. Then $\phi_q : E[n] \to E[n]$ is a bijective map and has two eigenvalues $\lambda_1 = 1$ and $\lambda_2 = q$. The decomposition of $E[n]$ into eigenspaces is*

$$E[n] = (\ker(\phi_q - [1]) \cap E[n]) \oplus (\ker(\phi_q - [q]) \cap E[n]).$$

The corresponding eigenspaces are $\ker(\phi_q - [1]) \cap E[n] = E(\mathbb{F}_q)[n]$ *and* $\ker(\phi_q - [q]) \cap E[n] \subseteq E(\mathbb{F}_{q^k})[n]$.

Proof: It is clear that there are n-torsion points in $E(\mathbb{F}_q)$ since $n \mid \#E(\mathbb{F}_q)$. Moreover, $E[n] \nsubseteq E(\mathbb{F}_q)$ since $k > 1$. Points defined over $\mathbb{F}_q$ are fixed under ϕ_q. Hence, 1 is an eigenvalue and the corresponding eigenspace is $\ker(\phi_q - [1]) \cap E[n] = E(\mathbb{F}_q)[n]$.

Recall the characteristic polynomial of ϕ_q. Since $n \mid (q + 1 - t)$, we have $t \equiv q + 1 \pmod{n}$. Hence, over $\mathbb{F}_n$,

$$X^2 - tX + q = X^2 - (q+1)X + q = (X-1)(X-q)$$

gives that the other eigenvalue of ϕ_q on $E[n]$ is q. Therefore, $E[n]$ is the sum of the two eigenspaces $\ker(\phi_q - [1]) \cap E[n] = E(\mathbb{F}_q)[n]$ and $\ker(\phi_q - [q]) \cap E[n] \subseteq E(\mathbb{F}_{q^k})[n]$. □

2.2 Tate Pairing

This section presents the definition and basic properties of the Tate pairing. Proofs of the results stated here can be found in Washington's book [31].

2.2.1 Divisors

Let E be an elliptic curve defined over a field $\mathbb{F}$. Define a symbol (P) for each point $P \in E$. A divisor D on E is a linear combination of such symbols:

$$D = \sum_{P \in E} a_P (P)$$

with all $a_P \in \mathbb{Z}$, and where only finitely many a_P's are non-zero. The set of all divisors on E is denoted $\mathrm{Div}(E)$. Define the support of a divisor to be the set of all points P such that $a_P \neq 0$. Define the degree of a divisor by

$$\deg\left(\sum_{P \in E} a_P (P)\right) = \sum_{P \in E} a_P \in \mathbb{Z}.$$

Define the sum of a divisor by

$$\text{sum}\left(\sum_{P\in E} a_P(P)\right) = \sum_{P\in E} a_P P \in E(\overline{\mathbb{F}}).$$

Here, the sum function simply uses the group law on E to add up the points inside the symbols.

An important subset of $\text{Div}(E)$ is the set of divisors with degree 0, denoted $\text{Div}^0(E)$. The sum function

$$\text{sum} : \text{Div}^0(E) \mapsto E(\overline{\mathbb{F}})$$

is surjective because

$$\text{sum}((P) - (\infty)) = P$$

for all $P \in E(\overline{\mathbb{F}})$.

Let $E/\mathbb{F}$ be an elliptic curve defined by equation $y^2 = x^3 + ax + b$. The function field $\overline{\mathbb{F}}(E)$ of E is the field of fractions of $\overline{\mathbb{F}}[x, y]/(y^2 - x^3 - ax - b)$. Note that a function $f(x, y) \in \overline{\mathbb{F}}(E)$ is defined at a finite point P if one can write $f = u/v$ with $u, v \in \overline{\mathbb{F}}(x, y)$ and $v(P) \neq 0$; otherwise f is not defined at P and we write $f(P) = \infty$.

Example 2.2.1 *Suppose $y^2 = x^3 + 3x$ is the equation of E. The function*

$$f(x, y) = \frac{x}{y}$$

is not defined at (0,0). However, on E,

$$\frac{x}{y} = \frac{xy}{y^2} = \frac{xy}{x^3 + 3x} = \frac{y}{x^2 + 3} = 0$$

which is defined at $(0, 0)$ and takes the value 0 at $(0, 0)$.

A function can always be transformed in this way so that its value at any point is neither $0/0$ nor ∞/∞. The function takes values in $\overline{\mathbb{F}} \cup \{\infty\}$. A function is said to have a zero at a point P if it equals 0 at P. A function is said to have a pole at P if it takes the value ∞ at P. Define a uniformizer at P, denoted u_P, to be a function such that $u_P(P) = 0$ and such that every function $f(x, y)$ can be written in the form

$$f = u_P^r g,$$

where $r \in \mathbb{Z}$ and $g(P) \neq 0, \infty$. It is known that the integer r is independent of the choice of u_P. Now, define the order of f at P by

$$\text{ord}_P(f) = r.$$

For $P = (x_0, y_0) \in E$, a natural choice of u_P is $u_P = x - x_0$ when $y_0 \neq 0$ and $u_P = y$ when $y_0 = 0$.

Example 2.2.2 *On $y^2 = x^3+3$, $x-1$ is a uniformizer at $(1,2)$. Consider $f(x,y) = y-2$. We have*

$$y^2 - 4 = x^3 - 1,$$

so

$$(y+2)(y-2) = (x-1)(x^2+x+1),$$

and

$$f(x,y) = y - 2 = (x-1)\left(\frac{x^2+x+1}{y+2}\right)$$

with $\frac{x^2+x+1}{y+2} \neq 0, \infty$ at $(1,2)$. Hence,

$$\mathrm{ord}_{(1,2)}(y-2) = 1.$$

For the elliptic curve E given by $y^2 = x^3+ax+b$, we take $u_\infty = x/y$ as the uniformizer at ∞. For example, suppose the curve equation is $y^2 = x^3+3$. The curve equation can be written as

$$\left(\frac{x}{y}\right)^3 = y^{-1}\left(1 - \frac{3}{x^3+3}\right).$$

Since $1 - \dfrac{3}{x^3+3} \neq 0, \infty$ at ∞, we have

$$y = \left(\frac{x}{y}\right)^{-3}\left(1 - \frac{3}{x^3+3}\right),$$

so

$$\mathrm{ord}_\infty(y) = -3.$$

Similarly,

$$x = \frac{x}{y}\cdot\left(\frac{x}{y}\right)^{-3}\left(1 - \frac{3}{x^3+3}\right)$$

gives

$$\mathrm{ord}_\infty(x) = -2.$$

For a non-zero function f on E, define the divisor of f, denoted (f), to be

$$(f) = \sum_{P\in E} \mathrm{ord}_P(f)(P) \in \mathrm{Div}(E).$$

The divisor of a function is called a principal divisor.

Proposition 2.2.1 *Let E be an elliptic curve and let f, g be non-zero functions on E. We have:*

1. *f has only finitely many zeros and poles.*
2. *$deg((f)) = 0$.*
3. *f has no zeros or poles, i.e., $(f) = 0$, if and only if f is a constant.*
4. *$(f \cdot g) = (f) + (g)$.*
5. *$(f/g) = (f) - (g)$.*
6. *$(f) - (g) = 0$ if and only if f is a constant multiple of g.*

The following example illustrates some of these properties. Suppose that P, Q and R are three points on E that lie on the line $y = ax + b$. Hence, the function

$$f(x, y) = y - ax - b$$

has zeros at P, Q, R and a triple pole at ∞, so

$$(y - ax - b) = (P) + (Q) + (R) - 3(\infty).$$

Suppose that $R = (x_R, y_R)$. Then $-R = (x_R, -y_R)$ and the vertical line $x = x_R$ passes through R and $-R$. The divisor of $x - x_R$ is then

$$(x - x_R) = (R) + (-R) - 2(\infty),$$

so

$$\left(\frac{y - ax - b}{x - x_R}\right) = (y - ax - b) - (x - x_R) = (P) + (Q) - (-R) - (\infty).$$

By definition of point addition in Section 2.1.1, $P + Q = -R$. Let $\ell_{P,Q}$ denote the function of the line through P and Q. Let v_P denote the function of the vertical line through P. We have the following result:

$$(P) + (Q) = (P + Q) + (\infty) + \left(\frac{\ell_{P,Q}}{v_{P+Q}}\right). \tag{2.2}$$

Theorem 2.2.1 *Let E be an elliptic curve defined over a field F. Let $D \in \mathrm{Div}^0(E)$. Then there is a function f on E with $(f) = D$ if and only if* $\mathrm{sum}(D) = \infty$.

Two divisors D and D' are said to be equivalent, denoted $D \sim D'$, if $D = D' + (f)$ for some function f. Hence, by Theorem 2.2.1, two equivalent divisors must have the same degree.

Let f be a function. Let $D = \sum_{P \in E} a_P(P)$ be a divisor of degree 0 such that the support of D is distinct from the support of (f). Define $f(D)$ to be

$$f(D) = \prod_{P \in E} f(P)^{a_P}.$$

Note that $f(D) \neq 0, \infty$. Let f and g be two functions such that $g = cf$ for some constant $c \in \overline{\mathbb{F}}$. Then

$$\begin{aligned} g(D) &= cf(D) \\ &= \prod_{P \in E} (cf(P))^{a_P} \\ &= \prod_{P \in E} c^{a_P} \cdot \prod_{P \in E} f(P)^{a_P} \\ &= c^{\sum_{P \in E} a_P} \cdot f(D) \\ &= f(D), \qquad \text{since } \deg(D) = 0. \end{aligned}$$

This shows that the value of a function evaluated at a zero divisor does not change if the function is multiplied by a non-zero field element.

2.2.2 The Tate Pairing

Let $E/\mathbb{F}_q$ be an elliptic curve with $n \mid \#E(\mathbb{F}_q)$, where n is a prime, $\gcd(n, q) = 1$ and $n \nmid (q-1)$. Let $k > 1$ be the embedding degree of E with respect to n, so $E[n] \subseteq E(\mathbb{F}_{q^k})$. Recall from Section 2.1.3 that the set of all n-torsion points on E, denoted $E[n]$, is

$$E[n] = \{P \in E(\mathbb{F}_{q^k}) | nP = \infty\}.$$

Define

$$nE(\mathbb{F}_{q^k}) = \{nP | P \in E(\mathbb{F}_{q^k})\}.$$

Then, $nE(\mathbb{F}_{q^k})$ is a subgroup of $E(\mathbb{F}_{q^k})$ and the quotient group $E(\mathbb{F}_{q^k})/nE(\mathbb{F}_{q^k})$ is a group of exponent n. Here, $E(\mathbb{F}_{q^k})/nE(\mathbb{F}_{q^k})$ can be considered as a set of equivalence classes of points in $E(\mathbb{F}_{q^k})$, where P is equivalent to Q if and only if $(P - Q) \in nE(\mathbb{F}_{q^k})$. An element $Q' \in E(\mathbb{F}_{q^k})/nE(\mathbb{F}_{q^k})$ is a set of such equivalent points.

Let $P \in E[n]$ and $Q' \in E(\mathbb{F}_{q^k})/nE(\mathbb{F}_{q^k})$. Let D_P, D_Q be two divisors with disjoint supports such that $D_P \sim (P) - (\infty)$ and $D_Q \sim (Q) - (\infty)$, where $Q \in Q'$. Since $nP = \infty$, by Theorem 2.2.1 there exists a function f with divisor $(f) = n(P) - n(\infty) = nD_P$. Since the supports of (f) and D_Q are disjoint, we have $f(D_Q) \neq 0, \infty$. We can now define the *Tate pairing*. The Tate pairing is a function

$$\langle \cdot, \cdot \rangle_n : E(\mathbb{F}_{q^k})[n] \times E(\mathbb{F}_{q^k})/nE(\mathbb{F}_{q^k}) \longrightarrow (\mathbb{F}_{q^k}^*)/(\mathbb{F}_{q^k}^*)^n,$$

with

$$\langle P, Q' \rangle_n = f(D_Q).$$

From now on, we assume that $E(\mathbb{F}_{q^k})$ does not contain any points of order n^2. Then the set $E(\mathbb{F}_{q^k})[n]$ of n-torsion points forms a set of distinct representatives for the equivalence classes in $E(\mathbb{F}_{q^k})/nE(\mathbb{F}_{q^k})$. Let P, Q be two points in $E[n]$. Let D_P, D_Q be two divisors with disjoint supports such that $D_P \sim (P) - (\infty)$ and $D_Q \sim (Q) - (\infty)$. Let f be a function with divisor $(f) = n(P) - n(\infty)$. We then define the simplified Tate pairing as

$$\langle \cdot, \cdot \rangle_n : E(\mathbb{F}_{q^k})[n] \times E(\mathbb{F}_{q^k})[n] \to (\mathbb{F}_{q^k}^*)/(\mathbb{F}_{q^k}^*)^n,$$

with

$$\langle P, Q \rangle_n = f(D_Q).$$

A drawback of the simplified Tate pairing is that pairing values are unique up to membership in a coset of $\mathbb{F}_{q^k}^*$. Let μ_n denote the order-n subgroup of $\mathbb{F}_{q^k}^*$. Since $\mathbb{F}_{q^k}^*$ is a cyclic group of order $q^k - 1$, the $(q^k - 1)/n$-th power map gives an isomorphism

$$(\mathbb{F}_{q^k}^*)/(\mathbb{F}_{q^k}^*)^n \to \mu_n.$$

This motivates the definition of the *reduced Tate pairing* as follows:

$$e_n : E(\mathbb{F}_{q^k})[n] \times E(\mathbb{F}_{q^k})[n] \to \mu_n$$

with

$$e_n(P, Q) = \langle P, Q \rangle_n^{(q^k-1)/n} = f(D_Q)^{(q^k-1)/n}.$$

This $(q^k - 1)/n$-th power map is called the *final exponentiation*.

Since $n \mid \#E(\mathbb{F}_q)$, there are n-torsion points in $E(\mathbb{F}_q)[n]$. We restrict the first argument to be taken from this set. From Section 2.1.4, we can see that $\ker(\phi_q - [q]) \cap E[n] \subseteq E(\mathbb{F}_{q^k})[n]$. For the second pairing argument, one could choose elements in this eigenspace of the Frobenius, since choosing both points from the other eigenspace results in a trivial pairing value. Then the reduced Tate pairing can be defined to be:

$$e_n : G_1 \times G_2 \to G_3$$

with

$$e_n(P, Q) = \langle P, Q \rangle_n^{(q^k-1)/n} = f(D_Q)^{(q^k-1)/n},$$

where

$$G_1 = \ker(\phi_q - [1]) \cap E[n] = E(\mathbb{F}_q)[n],$$
$$G_2 = \ker(\phi_q - [q]) \cap E[n] \subseteq E(\mathbb{F}_{q^k})[n],$$

and

$$G_3 = \mu_n \subseteq \mathbb{F}_{q^k}^*.$$

In the remainder of this thesis, the 'Tate pairing' refers to the reduced Tate pairing defined over two n-torsion points from $G_1 \times G_2$.

2.2.3 Properties of the Tate Pairing

In this section, the properties of abstract bilinear pairings will be introduced. Let G_1 and G_2 be abelian groups written in additive notation with identity element ∞. Suppose G_1 and G_2 have exponent n, and G_3 is a cyclic group of order n written in multiplicative notation with identity element 1. A bilinear pairing is a function

$$e : G_1 \times G_2 \to G_3$$

with the following properties:

1. Bilinearity: For all $P_1, P_2 \in G_1$ and all $Q_1, Q_2 \in G_2$ we have
   - $e(P_1 + P_2, Q_1) = e(P_1, Q_1)e(P_2, Q_1)$ and
   - $e(P_1, Q_1 + Q_2) = e(P_1, Q_1)e(P_1, Q_2)$.
2. Non-degeneracy:
   - For each $P \in G_1$ with $P \neq \infty$, there is some $Q \in G_2$ such that $e(P, Q) \neq 1$; and
   - For each $Q \in G_2$ with $Q \neq \infty$, there is some $P \in G_1$ such that $e(P, Q) \neq 1$.
3. For all $P \in G_1$ and $Q \in G_2$,
   - $e(P, \infty) = e(Q, \infty) = 1$,
   - $e(-P, Q) = e(P, Q)^{-1} = e(P, -Q)$,
   - $e([j]P, Q) = e(P, Q)^j = e(P, [j]Q)$ for all $j \in \mathbb{Z}$.

Property 3 follows from property 1. Since $e(P, Q) = e(P + \infty, Q) = e(P, Q)e(\infty, Q)$, we have $e(\infty, Q) = 1$. Similarly, $e(P, \infty) = 1$. Furthermore, since $1 = e(\infty, Q) = e(P - P, Q) = e(P, Q)e(-P, Q)$, we have $e(-P, Q) = e(P, Q)^{-1}$. Similarly, $e(P, -Q) = e(P, Q)^{-1}$. Therefore, the key properties of bilinear pairings are bilinearity and non-degeneracy.

Theorem 2.2.2 *Let E be an elliptic curve over $\mathbb{F}_q$, and let n be a prime with $\gcd(n, q) = 1$. The Tate pairing satisfies:*

1. *Bilinearity: For all $P_1, P_2 \in G_1$ and all $Q_1, Q_2 \in G_2$ we have*
   - *$e_n(P_1 + P_2, Q_1) = e_n(P_1, Q_1)e_n(P_2, Q_1)$; and*
   - *$e_n(P_1, Q_1 + Q_2) = e_n(P_1, Q_1)e_n(P_1, Q_2)$.*
2. *Non-degeneracy:*

- *For each $P \in G_1$ with $P \neq \infty$, there is some $Q \in G_2$ such that $e_n(P,Q) \neq 1$; and*
- *For each $Q \in G_2$ with $Q \neq \infty$, there is some $P \in G_1$ such that $e_n(P,Q) \neq 1$.*

The proof of non-degeneracy is briefly discussed in [16]. We prove bilinearity here. Let $P_3 = P_1 + P_2$ and let g be a function such that $(P_3) - (\infty) = (P_1) - (\infty) + (P_2) - (\infty) + (g)$. As introduced in Section 2.2.2, two functions f_1, f_2 with $(f_1) = n(P_1) - n(\infty)$ and $(f_2) = n(P_2) - n(\infty)$ are used in the definitions of $e_n(P_1, Q_1)$ and $e_n(P_2, Q_1)$. Hence, we have

$$(f_1 f_2 g^n) = n(P_1) - n(\infty) + n(P_2) - n(\infty) + n(g) = n(P_3) - n(\infty).$$

Let $D_{Q_1} \sim (Q_1) - (\infty)$ have support disjoint to the set $\{P_1, P_2, P_3, \infty\}$. We have

$$\begin{aligned} e_n(P_1 + P_2, Q_1) &= e_n(P_3, Q_1) \\ &= (f_1 f_2 g^n(D_{Q_1}))^{(q^k-1)/n} \\ &= f_1(D_{Q_1})^{(q^k-1)/n} \cdot f_2(D_{Q_1})^{(q^k-1)/n} \cdot g(D_{Q_1})^{q^k-1} \\ &= e_n(P_1, Q_1) \cdot e_n(P_2, Q_1) \qquad \text{since } g(D_{Q_1}) \in \mathbb{F}_{q^k}^*. \end{aligned}$$

Let $Q_3 = Q_1 + Q_2$ and let $D_{Q_1} \sim (Q_1) - (\infty)$, $D_{Q_2} \sim (Q_2) - (\infty)$. We have

$$(Q_3) - (\infty) = (Q_1) - (\infty) + (Q_2) - (\infty) + (h)$$

for some function h. Hence, $D_{Q_1} + D_{Q_2} \sim (Q_3) - (\infty)$. Then

$$\begin{aligned} e_n(P_1, Q_1 + Q_2) &= e_n(P_1, Q_3) \\ &= f_1(D_{Q_1} + D_{Q_2})^{(q^k-1)/n} \\ &= f_1(D_{Q_1})^{(q^k-1)/n} \cdot f_1(D_{Q_2})^{(q^k-1)/n} \\ &= e_n(P_1, Q_1) \cdot e_n(P_1, Q_2). \end{aligned}$$

This proves bilinearity of the Tate pairing.

2.3 Miller's Algorithm

2.3.1 Miller's Function

Miller's algorithm [21] is the most famous algorithm for computing the Tate pairing. The main idea is to use the double-and-add method to construct a function f such that $(f) = n(P) - n(\infty)$. Let $P \in E(\mathbb{F}_{q^k})$ and $\lambda \in \mathbb{Z}$. A *Miller function* $f_{\lambda,P}$ is a function such that

$$(f_{\lambda,P}) = \lambda(P) - ([\lambda]P) - (\lambda - 1)(\infty).$$

As we discussed in Section 2.2.1, such a function is uniquely defined up to multiplication by constants in $\mathbb{F}_{q^k}$. We use a recurrence relation to define the Miller functions as follows:

1. $f_{0,P} = f_{1,P} = 1$.

2. If $P = \infty$, then $f_{s,P} = 1$.

3. If a, b are positive integers and $P \neq \infty$, then

$$f_{a+b,P} = f_{a,P} \cdot f_{b,P} \cdot \frac{l_{[a]P,[b]P}}{v_{[a+b]P}},$$

where $l_{[a]P,[b]P}$ is the equation of the line through $[a]P$ and $[b]P$, and $v_{[a+b]P}$ is the equation of the vertical line through $[a+b]P$.

4. We provide further details on the lines $l_{[a]P,[b]P}$ and $v_{[a+b]P}$. Let the equation of E be $y^2 = x^3 + Ax + B$. Let $[a]P = (x_1, y_1)$, $[b]P = (x_2, y_2)$ and let $[a+b]P = (x_3, y_3)$.

   - If $[a]P \neq \pm[b]P$, then $l_{[a]P,[b]P} = Y - y_1 - \dfrac{y_2 - y_1}{x_2 - x_1}(X - x_1)$ and $v_{[a+b]P} = X - x_3$.
   - If $[a]P = [b]P$, then $l_{[a]P,[b]P} = Y - y_1 - \dfrac{3x_1^2 + A}{2y_1}(X - x_1)$ and $v_{[a+b]P} = X - x_3$.
   - If $[a]P = -[b]P$, then $l_{[a]P,[b]P} = X - x_1$ and $v_{[a+b]P} = 1$.

In all cases,

$$\left(\frac{l_{[a]P,[b]P}}{v_{[a+b]P}}\right) = ([a]P) + ([b]P) - ([a+b]P) - (\infty)$$

so that

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

Algorithm 2.3.1 ***Miller's Algorithm***

Input: $P, Q \in E[n]$ *and* $\lambda = (\lambda_{l-1}\lambda_{l-2}\ldots\lambda_1\lambda_0)_2 \in \mathbb{N}$

Output: $f_{\lambda,P}(Q)$

1. $T \leftarrow P$, $f \leftarrow 1$

2. *For* $i = l - 2$ *to* 0

(a) $f \leftarrow f^2 \cdot \frac{l_{T,T}(Q)}{v_{[2]T}(Q)}$

(b) $T \leftarrow [2]T$

(c) *If* $\lambda_i \neq 0$ *then*

i. $f \leftarrow f \cdot \frac{l_{T,P}(Q)}{v_{T+P}(Q)}$

ii. $T \leftarrow T + P$

3. *Return* f

In this algorithm, Step 2 is called the Miller loop. There are $\log_2(\lambda)$ iterations in the Miller loop. Steps 2(a) and 2(b) are called the doubling step, while Step 2(c) is called the addition step. Clearly, the doubling step is processed $\log_2(\lambda)$ times and the hamming weight of λ determines the number of times that the addition step is processed. In later chapters of this thesis, ideas for shortening $\log_2(\lambda)$ and decreasing the hamming weight of λ are introduced to get faster pairings.

2.3.2 Computing the Tate Pairing

While computing the Tate pairing, we can use the Miller function $f_{n,P}$ since $(f_{n,P}) = n(P) - n(\infty)$. Let $P, Q \in E[n]$, and let $R \in E(\mathbb{F}_q)$ with $R \notin \{P, Q, -P, -Q, \infty\}$. Let $D_P = (P) - (\infty)$ and $D_Q = (Q + R) - (R) \sim (Q) - (\infty)$; note that D_P and D_Q have disjoint supports. Then

$$e_n(P, Q) = \left(\frac{f_{n,P}(Q + R)}{f_{n,P}(R)} \right)^{(q^k - 1)/n}.$$

In this way, two Miller functions and an inversion need to be computed. However, if $P \in E(\mathbb{F}_q)[n]$ then $f_{n,P}(R)$ and the inversion can be eliminated using the following 'denominator elimination' idea introduced by Barreto, Lynn and Scott [3].

Lemma 2.3.1 *Let $E/\mathbb{F}_q$ be an elliptic curve with $n \mid \#E(\mathbb{F}_q)$, where n is a prime and $n \nmid (q-1)$. Let $k > 1$ be the embedding degree of E with respect to n, and d be a proper factor of k. Then any nonzero element in $\mathbb{F}_{q^d}$ equals 1 after the final exponentiation by $(q^k - 1)/n$ while computing the pairing.*

Proof: The proof begins with the factorization

$$q^k - 1 = (q^d - 1) \cdot \sum_{i=0}^{k/d-1} q^{id}.$$

Since k is the smallest integer such that $n \mid q^k - 1$, we have $n \nmid q^d - 1$. Hence, $n \mid \sum_{i=0}^{k/d-1} q^{id}$. Thus, $q^d - 1$ divides $(q^k - 1)/n$. Let x be a nonzero element in $\mathbb{F}_{q^d}$. We have $x^{q^d-1} = 1$. Therefore, $x^{(q^k-1)/n} = 1$. □

Let $P \in E(\mathbb{F}_q)[n], Q \in E[n]$, and let $R \in E(\mathbb{F}_q)$ with $R \notin \{P, Q, -P, -Q, \infty\}$. Let $D_P = (P + R) - (R) \sim (P) - (\infty)$ and $D_Q = (Q) - (\infty)$. Since $(P + R) - (R) \sim (P) - (\infty)$, there exists a function g such that $(P + R) - (R) = (P) - (\infty) + (g)$. Hence, $n(P + R) - n(R) = n(P) - n(\infty) + (g^n)$. Therefore, setting

$$f = f_{n,P} \cdot g^n,$$

we have

$$\begin{aligned}
e_n(P, Q) &= f(D_Q)^{(q^k-1)/n} \\
&= \left(\frac{f(Q)}{f(\infty)}\right)^{(q^k-1)/n} \\
&= \left(\frac{f_{n,P}(Q) \cdot g(Q)^n}{f_{n,P}(\infty) \cdot g(\infty)^n}\right)^{(q^k-1)/n} \\
&= \frac{f_{n,P}(Q)^{(q^k-1)/n} \cdot g(Q)^{q^k-1}}{1 \cdot g(\infty)^{q^k-1}} \qquad \text{since } f_{n,P}(\infty) \in \mathbb{F}_q \\
&= f_{n,P}(Q)^{(q^k-1)/n}.
\end{aligned}$$

Now, the computation of the Tate pairing requires a single Miller function evaluation with $\log_2(n)$ iterations of Miller loop and a final exponentiation to the power $(q^k - 1)/n$.

2.4 Pairing-Based Cryptography

Pairings are being used to solve protocol problems. Recall the 3 groups G_1, G_2 and G_3 defined in Section 2.2.2. Let

$$e : G_1 \times G_2 \to G_3$$

be a bilinear pairing defined over the three n-torsion groups. Recall the *Diffie Hellman problem (DHP)* in an additive group $G = \langle P \rangle$ to be: Given aP and bP where $a, b \in \mathbb{Z}_n$, compute abP. The co-DHP is defined to be: Given $M, aP \in G_1$ and $aQ \in G_2$, compute aM. The *bilinear Diffie-Hellman problem (BDHP)* is defined to be: Given $P \in G_1, Q \in G_2$, aQ and bQ where $a, b \in \mathbb{Z}$, compute $e(P, Q)^{ab}$. As is generally assumed in the literature, we shall assume co-DHP and BDHP to be as hard as DHP in G_1, G_2 and G_3. Some protocols using bilinear pairings are introduced in this section.

2.4.1 Short Signatures

The BLS signature scheme [7] was proposed by Boneh, Lynn and Shacham.

1. *Public Parameters.* Let $E/\mathbb{F}_q$ be an elliptic curve with $n \mid \#E(\mathbb{F}_q)$, where n is a prime and $n \nmid (q-1)$. Let $k > 1$ be the embedding degree of E with respect to n. Let $e_n : G_1 \times G_2 \to G_3$ be a non-degenerate bilinear pairing with $G_1 = \ker(\phi_q - [1]) \cap E[n] = E(\mathbb{F}_q)[n]$, $G_2 = \ker(\phi_q - [q]) \cap E[n] \subseteq E(\mathbb{F}_{q^k})[n]$, and $G_3 = \mu_n \subseteq \mathbb{F}_{q^k}^*$. Let $P \in G_1^*$ and $Q \in G_2^*$. Let $H : \{0,1\}^* \to G_1$ be a hash function.

2. *Key generation.*

   (a) Alice randomly picks $x \in [1, n-1]$ as her private key.

   (b) Alice computes $W = xP$, $X = xQ$ as her public key.

   (c) Alice sends her public (W, X) to a certification authority.

   (d) The certification authority verifies that $W \in G_1$, $X \in G_2$, $W \neq 1$, $X \neq 1$ and $e(W, Q) = e(P, X)$ and issues a certificate for (W, X).

3. *Signature generation.* To sign a message $m \in \{0,1\}^*$, Alice does the following:

   (a) Compute $M = H(m) \in G_1$.

   (b) Compute $S = xM$.

   (c) The signed message is (m, S).

4. *Signature verification.* To verify (m, S), Bob does the following:

   (a) Obtain Alice's public key (W, X) from Alice's certificate.

   (b) Compute $M = H(m) \in G_1$.

   (c) Accept the signature if and only if $e(M, X) = e(S, Q)$.

If an attacker wants to forge Alice's signature on a message m', he needs to compute $S = xM'$ given P, xP, $M' = H(m') \in G_1$ and $xQ \in G_2$. This is an instance of co-DHP which is infeasible to solve. The BLS signature scheme is called a short signature scheme because it is the first scheme whose signatures are comprised of a single group element. Moveover, signatures can be aggregated [6]. The BGLS signature scheme is an aggregate signature scheme based on the BLS signature scheme. We describe BGLS next.

1. *Public Parameters.* Let $E/\mathbb{F}_q$ be an elliptic curve with $n \mid \#E(\mathbb{F}_q)$, where n is a prime and $n \nmid (q-1)$. Let $k > 1$ be the embedding degree of E with respect to n. Let $e_n : G_1 \times G_2 \to G_3$ be a non-degenerate bilinear pairing with $G_1 = \ker(\phi_q - [1]) \cap E[n] = E(\mathbb{F}_q)[n]$, $G_2 = \ker(\phi_q - [q]) \cap E[n] \subseteq E(\mathbb{F}_{q^k})[n]$, and $G_3 = \mu_n \subseteq \mathbb{F}_{q^k}^*$. Let $P \in G_1^*$ and $Q \in G_2^*$. Let $H : \{0,1\}^* \to G_1$ be a hash function.

2. *Key generation.*

   (a) Each user A_i has private key $x_i \in [1, n-1]$.

   (b) Each user A_i computes $W_i = x_i P$, $X_i = x_i Q$ as the public key, where $G_1 = \langle P \rangle$ and $G_2 = \langle Q \rangle$.

   (c) Each user A_i sends the public (W_i, X_i) to a certification authority.

   (d) The certification authority verifies that $W_i \in G_1$, $X_i \in G_2$, $W_i \neq 1$, $X_i \neq 1$ and $e(W_i, Q) = e(P, X_i)$ and issues certificates for all (W_i, X_i).

3. *Signature generation.* To sign messages m_i's, each user A_i does the following:

   (a) Compute $M_i = H(m_i) \in G_1$.

   (b) Compute $S_i = x_i M_i$.

   (c) A_i's signature on m_i is S_i.

4. *Signature aggregation.* Given (m_1, S_1), (m_2, S_2), $\ldots$, (m_t, S_t), the aggregated signature is $S = \sum_{i=1}^{t} S_i$.

5. *Signature verification.* To verify the aggregated signature S on $(m_1, m_2, \ldots, m_t)$, Bob does the following:

   (a) Obtain each A_i's public key (W_i, X_i) from their certificates.

   (b) Compute $M_i = H(m_i) \in G_1$.

   (c) Accept the signature if and only if $\prod_i e(M_i, X_i) = e(S, Q)$.

Only one pairing evaluation is needed in the right hand side of step 4(c) to obtain the assurance that each message m_i was signed by A_i. It is shown in [6] and [8] that this scheme is secure if co-DHP in G_1, G_2 is hard and H is a random function.

2.4.2 Identity-Based Encryption

Generally, in a public-key cryptosystem, Alice first generates her public key and private key. Bob uses her public key to encrypt the secret message and sends to Alice. Finally, Alice decrypts the ciphertext using her private key. Identity-based cryptosystems were first introduced by Shamir in 1984 [28]. Different from traditional public-key cryptosystems, the private key of Alice is generated by a trusted third party (TTP) using her identity information ID_A. Bob encrypts for Alice using ID_A and the TTP's public key. In this case, Bob does not need to worry whether Alice's public key is authenticated or not, on indeed whether Alice has actually generated a public key.

In 2001, the first practical identity-based encryption scheme was proposed by Boneh and Franklin [5].

1. *Public Parameters.* Let $E/\mathbb{F}_q$ be an elliptic curve with $n \mid \#E(\mathbb{F}_q)$, where n is a prime and $n \nmid (q-1)$. Let $k > 1$ be the embedding degree of E with respect to n. Let $e_n : G_1 \times G_2 \to G_3$ be a non-degenerate bilinear pairing with $G_1 = \ker(\phi_q - [1]) \cap E[n] = E(\mathbb{F}_q)[n]$, $G_2 = \ker(\phi_q - [q]) \cap E[n] \subseteq E(\mathbb{F}_{q^k})[n]$, and $G_3 = \mu_n \subseteq \mathbb{F}_{q^k}^*$. Let $P \in G_1^*$ and $Q \in G_2^*$. Let $H : \{0,1\}^* \to G_1$ and $H' : G_3 \to \{0,1\}^l$ be two hash functions.

2. *Key generation.*

   (a) TTP randomly picks $t \in [1, n-1]$ to be TTP's private key.

   (b) TTP computes $T = tQ \in G_2$ as TTP's public key, where $G_2 = \langle Q \rangle$.

   (c) TTP generates Alice's private key d_A using her identity information ID_A: $d_A = tH(ID_A) \in G_1$.

   (d) TTP securely sends d_A to Alice.

3. *Encryption.*

   (a) To encrypt message $m \in \{0,1\}^l$, Bob computes $P_A = H(ID_A) \in G_1$ and obtains TTP's public key T.

   (b) Bob randomly picks $r \in [1, n-1]$.

   (c) Bob computes $R = rQ \in G_2$.

   (d) Bob encrypts the message using $c = m \oplus H'(e(P_A, T)^r)$.

   (e) Bob then sends the ciphertext (R, c) to Alice.

4. *Decryption.*

   (a) Alice gets d_A from TTP using a secure channel and receives the ciphertext (R, c) from Bob.

   (b) Alice decrypts the message using $m = c \oplus H'(e(d_A, R))$.

The decryption is correct because

$$e(d_A, R) = e(tP_A, rQ) = e(P_A, Q)^{rt} = e(P_A, tQ)^r = e(P_A, T)^r.$$

If an attacker wants to recover $m \in \{0,1\}^l$ from $c \in \{0,1\}^l$, $P_A \in G_1$ and R, $T \in G_2$, he needs to compute $e(P_A, T)^r$, i.e, solve an instance of BDHP. The identity-based encryption scheme basically solves the key management problem for the certifying authority, but a secure channel between TTP and receiver is needed.

Chapter 3

Optimal Pairings

Some results in this thesis are only true if the Miller functions are normalized. We start this chapter by defining what it means for a function to be normalized.

Let $f \in \overline{\mathbb{F}}_q(E)$ be a function, and suppose that f has a pole of order a at ∞. Take $u = x/y$ to be the uniformizer at ∞. Define the leading coefficient $lc_\infty(f)$ of f to be $(u^a f)(\infty)$. Then f is said to be normalized if $lc_\infty(f) = 1$. By Lemma 2.3.1, since the pairing values are in $\mathbb{F}_{q^k}$ after the final exponentiation, we can say that $f_{a,P}$ is normalized if $lc_\infty(f_{a,P})$ is in a proper subfield of $\mathbb{F}_{q^k}$. Recall that using Miller's algorithm introduced in Section 2.3, a Miller function is a product of lines with leading coefficients 1. Hence, the Miller functions we constructed in this thesis are already normalized. Unless otherwise stated, we will assume that all Miller functions are normalized.

3.1 Vercauteren's Construction

To clearly understand this section, the following lemma is a good place to start.

Lemma 3.1.1 *For every $Q \in E(\mathbb{F}_{q^k})$ and integer s, let $f_{s,Q}$ be an $\mathbb{F}_{q^k}$-rational function with divisor*

$$(f_{s,Q}) = s(Q) - ([s]Q) - (s-1)(\infty).$$

Then, for all $a, b \in \mathbb{Z}$, we have

$$f_{ab,Q} = f_{a,Q}^b \cdot f_{b,[a]Q}. \tag{3.1}$$

Proof: The divisors of both sides of (3.1) can be written as follows:

$$(f_{ab,Q}) = ab(Q) - ([ab]Q) - (ab-1)(\infty)$$

and

$$\begin{aligned}(f_{a,Q}^{b} \cdot f_{b,[a]Q}) &= b(a(Q) - ([a]Q) - (a-1)(\infty)) + (b([a]Q) - ([ab]Q) - (b-1)(\infty) \\ &= ab(Q) - b([a]Q) - (ab-b)(\infty) + b([a]Q) - ([ab]Q) - (b-1)(\infty) \\ &= ab(Q) - ([ab]Q) - (ab-1)(\infty).\end{aligned}$$

Therefore, (3.1) holds in general. □

In [30], it is noted that the Tate pairing over $G_2 \times G_1$

$$e_n : G_2 \times G_1 \to \mu_n$$

can be defined by

$$e_n(Q,P) = f_{n,Q}(P)^{(q^k-1)/n},$$

provided that $f_{n,Q}$ is normalized. The central idea for Miller loop reduction is to raise the Tate pairing $e_n(Q,P)$ to some fixed integer power m; here, $P \in G_1$ and $Q \in G_2$. Using Lemma 3.1.1, we have

$$\begin{aligned}e_n^m(Q,P) &= f_{n,Q}(P)^{m(q^k-1)/n} \\ &= \frac{f_{mn,Q}(P)^{(q^k-1)/n}}{f_{m,[n]Q}(P)^{(q^k-1)/n}} \qquad \text{by (3.1)} \\ &= \frac{f_{mn,Q}(P)^{(q^k-1)/n}}{1^{(q^k-1)/n}} \qquad \text{since } [n]Q = \infty \\ &= f_{mn,Q}(P)^{(q^k-1)/n}.\end{aligned}$$

Hence,

$$e_n^m(Q,P) = f_{mn,Q}(P)^{(q^k-1)/n}. \tag{3.2}$$

Since the Tate pairing is non-degenerate, $f_{mn,Q}(P)^{(q^k-1)/n}$ also defines a non-degenerate pairing whenever $n \nmid m$. The main idea is then to find an m such that $f_{mn,Q}(P)$ can be written as some multiple and/or power of simpler non-degenerate functions $f_{\lambda_i,Q}(P)$. If the λ_i's are small, the pairing can be computed in fewer Miller iterations. The increment of exponentiation cost can be reduced, since the q-th powering of $f_{\lambda_i,Q}(P)$ corresponds to multiplication by q in $\langle Q \rangle$. Further, multiplication by q in $\langle Q \rangle$ can be reduced to multiplication by any integer $a \equiv q \pmod{n}$.

Vercauteren first introduced the notion of an optimal pairing in his paper [30]. The definition is as follows:

Definition 3.1.1 *Let $e : G_1 \times G_2 \to G_T$ be a non-degenerate bilinear pairing with $|G_1| = |G_2| = |G_T| = n$, where the field of definition of G_T is $\mathbb{F}_{q^k}$. Then $e(\cdot,\cdot)$ is called an optimal pairing if it can be computed in $\log_2 n/\varphi(k)+\varepsilon(k)$ basic Miller iterations, with $\varepsilon(k) \leq \log_2 k$. Here, $\varphi(k)$ is the Euler totient function.*

By Definition 3.1.1, it is easy to see that, as long as all $f_{\lambda_i,Q}(P)$'s can be computed in $\log_2 n/\varphi(k) + \varepsilon(k)$ basic Miller iterations, the resulting pairing is optimal. Based on this idea, some efficient pairings are introduced in the following sections.

3.2 Ate Pairing

The ate pairing [17] is an optimized version of the Tate pairing. The missing "T" means it is faster. Using results in the previous section, the ate pairing can be derived based on Vercauteren's construction. Consider a fixed power of the Tate pairing $e_n^m(Q, P)$.

Let $\lambda \in \mathbb{Z}$, $\lambda \equiv q \pmod{n}$; then we have

$$q^k - 1 \equiv \lambda^k - 1 \pmod{n}.$$

Since $n \mid q^k - 1$, we have $n \mid \lambda^k - 1$. Letting $m' = (\lambda^k - 1)/n$, we have

$$\begin{aligned} e_n^{m'}(Q, P) &= f_{nm',Q}(P)^{(q^k-1)/n} \qquad \text{by (3.2)} \\ &= f_{\lambda^k-1,Q}(P)^{(q^k-1)/n}. \end{aligned}$$

For non-degeneracy, we need $n \nmid m'$; note that $n^2 \nmid q^k - 1$ is not sufficient. Recall the observation

$$f_{a+b,Q} = f_{a,Q} \cdot f_{b,Q} \cdot \frac{l_{[a]Q,[b]Q}}{v_{[a+b]Q}}, \tag{3.3}$$

where $a, b \in \mathbb{Z}$, $l_{[a]Q,[b]Q}$ is the equation of the line through $[a]Q$ and $[b]Q$, and $v_{[a+b]Q}$ is the equation of the vertical line through $[a + b]Q$. Then

$$\begin{aligned} e_n^{m'}(Q, P) &= f_{\lambda^k-1,Q}(P)^{(q^k-1)/n} \\ &= \left(\frac{f_{\lambda^k,Q}(P)}{f_{1,Q}(P) \cdot \frac{l_{[\lambda^k-1]Q,Q}(P)}{v_{[\lambda^k]Q}(P)}} \right)^{(q^k-1)/n} \qquad \text{by (3.3)} \\ &= \left(\frac{f_{\lambda^k,Q}(P)}{1 \cdot \frac{v_{[\lambda^k]Q}(P)}{v_{[\lambda^k]Q}(P)}} \right)^{(q^k-1)/n} \qquad \text{since } [\lambda^k - 1]Q = \infty \\ &= f_{\lambda^k,Q}(P)^{(q^k-1)/n}. \end{aligned}$$

By repeatedly applying (3.1), we obtain

$$\begin{aligned} e_n^{m'}(Q,P) &= f_{\lambda^k,Q}(P)^{(q^k-1)/n} \\ &= \left(f_{\lambda,Q}^{\lambda^{k-1}}(P)\cdot f_{\lambda,[\lambda]Q}^{\lambda^{k-2}}(P)\cdot f_{\lambda,[\lambda^2]Q}^{\lambda^{k-3}}(P)\cdot\cdots\cdot f_{\lambda,[\lambda^{k-2}]Q}^{\lambda}(P)\cdot f_{\lambda,[\lambda^{k-1}]Q}(P)\right)^{(q^k-1)/n} \\ &= \left(\prod_{i=0}^{k-1} f_{\lambda,[\lambda^{k-1-i}]Q}^{\lambda^i}(P)\right)^{(q^k-1)/n} \\ &= \left(\prod_{i=0}^{k-1} f_{\lambda,[q^{k-1-i}]Q}^{q^i}(P)\right)^{(q^k-1)/n} \qquad \text{since } \lambda \equiv q \pmod{n}. \end{aligned}$$

Notice that for $a \in \mathbb{Z}$,

$$(f_{a,Q}(P))^q = f_{a,\pi_q(Q)}(\pi_q(P)) = f_{a,[q]Q}(P),$$

since $P \in G_1$ whence $\pi_q(P) = P$ and since $Q \in G_2$ whence $\pi_q(Q) = [q]Q$. Hence for $i \geq 0$,

$$f_{a,[q^i]Q}(P) = f_{a,Q}^{q^i}(P). \tag{3.4}$$

Thus, we have

$$\begin{aligned} e_n^{m'}(Q,P) &= \left(\prod_{i=0}^{k-1} f_{\lambda,[q^{k-1-i}]Q}^{q^i}(P)\right)^{(q^k-1)/n} \\ &= \left(\prod_{i=0}^{k-1} f_{\lambda,Q}^{q^i\cdot q^{k-1-i}}(P)\right)^{(q^k-1)/n} \qquad \text{by (3.4)} \\ &= \left(\prod_{i=0}^{k-1} f_{\lambda,Q}^{q^{k-1}}(P)\right)^{(q^k-1)/n} \\ &= (f_{\lambda,Q}^{k\cdot q^{k-1}}(P))^{(q^k-1)/n} \\ &= (f_{\lambda,Q}(P))^{k\cdot q^{k-1}\cdot(q^k-1)/n}. \end{aligned}$$

Now, $f_{\lambda,Q}(P)^{(q^k-1)/n}$ is of prime order n. Also, k and q are relatively prime to n, and so $n \nmid k \cdot q^{k-1}$. Hence, $(k \cdot q^{k-1})^{-1} \mod n$ exists. Letting

$$m = m' \cdot (k \cdot q^{k-1})^{-1} \mod n,$$

the ate pairing $a(Q,P) = e_n^m(Q,P)$ can be defined as follows:

$$\begin{aligned} a(Q,P) &= e_n^m(Q,P) \\ &= (e_n^{m'}(Q,P))^{(k\cdot q^{k-1})^{-1}} \\ &= (f_{\lambda,Q}(P)^{(q^k-1)/n})^{k\cdot q^{k-1}\cdot(k\cdot q^{k-1})^{-1}} \\ &= (f_{\lambda,Q}(P))^{(q^k-1)/n}. \end{aligned}$$

λ is chosen so that $n^2 \nmid \lambda^k - 1$, and thus we have $n \nmid m'$ and $n \nmid m$. Since the Tate pairing is non-degenerate, the ate pairing is also non-degenerate. For example, λ can be chosen to be $t-1$ since $\#E(\mathbb{F}_q) = q+1-t$ and $n \mid q+1-t$. By Theorem 2.1.2, $|t| \leq 2\sqrt{q}$. Hence, $n \approx q$ gives $\lambda \approx \sqrt{n}$. From this point of view, the Miller length of the ate pairing is significantly shorter than that of the Tate pairing. However, the cost of the doubling and addition steps in Miller's Algorithm becomes larger.

The above process also works for $\lambda_i \equiv q^i \pmod{n}$. Zhao, Zhang and Huang [32] introduced variations of the ate pairing using this idea. They define the ate$_i$ pairing to be

$$a_i(Q,P) = f_{\lambda_i,Q}(P)^{(q^k-1)/n}$$

where $\lambda_i \equiv q^i \pmod{n}$. By trying different i's, one can hope to find a λ_i that is smaller than the λ in the ate pairing. However, according to Definition 3.1.1, the ate pairing and ate$_i$ pairings are not optimal pairings.

3.3 R-ate Pairing

The R-ate Pairing [20] was discovered before the publication of Vercauteren's paper [30]. The "R" here can be regarded as a ratio of two pairings, yet it is still considered a fixed power $e_n^m(Q,P)$ of the Tate pairing.

Let $A, B, a, b \in \mathbb{Z}$ with $A = aB + b$. Consider the Miller function $f_{A,Q}(P)$. We have

$$\begin{aligned} f_{A,Q}(P) &= f_{aB+b,Q}(P) \\ &= f_{aB,Q}(P) \cdot f_{b,Q}(P) \cdot \frac{l_{[aB]Q,[b]Q}(P)}{v_{[A]Q}(P)} \qquad \text{by (3.3)} \\ &= f_{B,Q}^a(P) \cdot f_{a,[B]Q}(P) \cdot f_{b,Q}(P) \cdot \frac{l_{[aB]Q,[b]Q}(P)}{v_{[A]Q}(P)} \qquad \text{by (3.1).} \end{aligned}$$

Define the function $R_{A,B}(Q,P)$ to be

$$\begin{aligned} R_{A,B}(Q,P) &= \left(f_{a,[B]Q}(P) \cdot f_{b,Q}(P) \cdot \frac{l_{[aB]Q,[b]Q}(P)}{v_{[A]Q}(P)} \right)^{(q^k-1)/n} \\ &= \left(\frac{f_{A,Q}(P)}{f_{B,Q}^a(P)} \right)^{(q^k-1)/n}. \end{aligned}$$

If $f_{A,Q}(P)$ and $f_{B,Q}(P)$ are Miller functions for non-degenerate pairings, then the new pairing $R_{A,B}(Q,P)$ is also a non-degenerate pairing.

Let $L_1, L_2, M_1, M_2 \in \mathbb{Z}$ such that

$$e_n^{L_1}(Q,P) = f_{A,Q}(P)^{M_1 \cdot (q^k-1)/n}$$

and

$$e_n^{L_2}(Q,P) = f_{B,Q}(P)^{M_2 \cdot (q^k-1)/n}.$$

Let $M = \mathrm{lcm}(M_1, M_2)$ and $m = \frac{M}{M_1} \cdot L_1 - a\frac{M}{M_2} \cdot L_2$. For non-degeneracy, n should not divide the integer power m. We have

$$\begin{aligned} e_n^m(Q,P) &= e_n^{\frac{M}{M_1} \cdot L_1 - a\frac{M}{M_2} \cdot L_2}(Q,P) \\ &= \frac{e_n(Q,P)^{L_1 \cdot \frac{M}{M_1}}}{e_n(Q,P)^{aL_2 \cdot \frac{M}{M_2}}} \\ &= \left(\frac{f_{A,Q}(P)}{f_{B,Q}(P)^a}\right)^{M \cdot (q^k-1)/n}. \end{aligned}$$

Now, it is easy to see that $e_n^m(Q,P) = R_{A,B}(Q,P)^M$. $R_{A,B}(Q,P)$ is defined to be the R-ate pairing. However, arbitrary integers A, B do not give a non-degenerate pairing in general. Four possible choices for integer pairs (A,B) are given in [20] as follows:

1. $(A,B) = (q^i, n)$,
2. $(A,B) = (q, T_1)$,
3. $(A,B) = (T_i, T_j)$,
4. $(A,B) = (n, T_i)$,

where $T_i \equiv q^i \pmod{n}$ for $i \in \mathbb{Z}$ with $0 < i < k$. Consider each case one by one.

First, $(A,B) = (q^i, n)$. Since $A = aB + b$, we have $q^i = an + b$. Hence, $b \equiv q^i \pmod{n}$ and

$$\left(\frac{f_{q^i,Q}(P)}{f_{n,Q}^a(P)}\right)^{(q^k-1)/n} = R_{A,B}(Q,P) = \left(f_{a,[n]Q}(P) \cdot f_{b,Q}(P) \cdot \frac{l_{[an]Q,[b]Q}(P)}{v_{[q^i]Q}(P)}\right)^{(q^k-1)/n}.$$

Since $b \equiv q^i \pmod{n}$, $l_{[an]Q,[b]Q}(P)$ is the same as $v_{[q^i]Q}(P)$. Moreover, $f_{a,[n]Q}(P) = 1$. Thus

$$R_{A,B}(Q,P) = \left(f_{q^i,Q}(P)\right)^{(q^k-1)/n}. \tag{3.5}$$

Second, $(A, B) = (q, T_1)$ that is $q = aT_1 + b$. Then

$$\left(\frac{f_{q,Q}(P)}{f^a_{T_1,Q}(P)}\right)^{(q^k-1)/n} = R_{A,B}(Q,P) = \left(f_{a,[T_1]Q}(P) \cdot f_{b,Q}(P) \cdot \frac{l_{[aT_1]Q,[b]Q}(P)}{v_{[q]Q}(P)}\right)^{(q^k-1)/n}.$$

By (3.4), $f_{a,[T_1]Q}(P) = f^q_{a,Q}(P)$, and hence

$$R_{A,B}(Q,P) = \left(f^q_{a,Q}(P) \cdot f_{b,Q}(P) \cdot \frac{l_{[aT_1]Q,[b]Q}(P)}{v_{[q]Q}(P)}\right)^{(q^k-1)/n}. \tag{3.6}$$

Third, $(A, B) = (T_i, T_j)$ that is $T_i = aT_j + b$. We have

$$\left(\frac{f_{T_i,Q}(P)}{f^a_{T_j,Q}(P)}\right)^{(q^k-1)/n} = R_{A,B}(Q,P) = \left(f_{a,[T_j]Q}(P) \cdot f_{b,Q}(P) \cdot \frac{l_{[aT_j]Q,[b]Q}(P)}{v_{[q^i]Q}(P)}\right)^{(q^k-1)/n}.$$

Similarly, by (3.4), $f_{a,[T_j]Q}(P) = f^{q^j}_{a,Q}(P)$, and so

$$R_{A,B}(Q,P) = \left(f^{q^j}_{a,Q}(P) \cdot f_{b,Q}(P) \cdot \frac{l_{[aT_j]Q,[b]Q}(P)}{v_{[q^i]Q}(P)}\right)^{(q^k-1)/n}. \tag{3.7}$$

Finally, $(A, B) = (n, T_i)$ that is $n = aT_i + b$. Thus

$$\left(\frac{f_{n,Q}(P)}{f^a_{T_i,Q}(P)}\right)^{(q^k-1)/n} = R_{A,B}(Q,P) = \left(f_{a,[T_i]Q}(P) \cdot f_{b,Q}(P) \cdot \frac{l_{[aT_i]Q,[b]Q}(P)}{v_{[n]Q}(P)}\right)^{(q^k-1)/n}.$$

Similarly, by (3.4), $f_{a,[T_j]Q}(P) = f^{q^j}_{a,Q}(P)$, and so

$$R_{A,B}(Q,P) = \left(f^{q^i}_{a,Q}(P) \cdot f_{b,Q}(P) \cdot \frac{l_{[aT_i]Q,[b]Q}(P)}{v_{[n]Q}(P)}\right)^{(q^k-1)/n}. \tag{3.8}$$

The resulting R-ate pairing in the first case (3.5) is the ate$_i$ Pairing introduced in Section 2.2. While computing the pairings in (3.6), (3.7) and (3.8), two Miller loops of lengths $\log a$ and $\log b$ need to be evaluated. There is only one integer parameter i we can change to get efficient pairings in (3.6) and (3.8), while we can change two parameters i and j in (3.7). Therefore, on most of the pairing-friendly curves, we choose $(A, B) = (T_i, T_j)$. For the purpose of shortening the Miller length, by trying different i and j, integers a and b can be small enough that the loop length in Miller's algorithm is as small as $\log(r^{1/\phi(k)})$. Hence, the R-ate pairing is an optimal pairing by definition.

3.4 Vercauteren's Pairing

Vercauteren introduced a specific optimal pairing after his general construction [30]. This pairing is called Optimal ate pairing in [30] and some other papers. In order to be clear, we call it Vercauteren's Pairing in this thesis. To begin with, consider a fixed power $m \in \mathbb{Z}$ of the Tate pairing $e_n^m(Q,P)$. We have

$$e_n^m(Q,P) = f_{mn,Q}(P)^{(q^k-1)/n} \qquad \text{by (3.2).}$$

Let $\alpha = mn$ and write it in base-q expansion $\alpha = \sum_{i=0}^{l} c_i q^i$. Using (3.3), (3.1) and (3.4), we have

$$\begin{aligned}
e_n^m(Q,P) &= f_{\alpha,Q}(P)^{(q^k-1)/n} \\
&= f_{\sum_{i=0}^{l} c_i q^i,Q}(P)^{(q^k-1)/n} \\
&= \left(\prod_{i=0}^{l} f_{c_i q^i,Q}(P) \cdot \prod_{i=0}^{l-1} \frac{l_{[\sum_{j=i+1}^{l} c_j q^j]Q,[c_i q^i]Q}(P)}{v_{[\sum_{j=i}^{l} c_j q^j]Q}(P)}\right)^{(q^k-1)/n} \\
&= \left(\prod_{i=0}^{l} \left(f_{q^i,Q}^{c_i}(P) \cdot f_{c_i,[q^i]Q}(P)\right) \cdot \prod_{i=0}^{l-1} \frac{l_{[\sum_{j=i+1}^{l} c_j q^j]Q,[c_i q^i]Q}(P)}{v_{[\sum_{j=i}^{l} c_j q^j]Q}(P)}\right)^{(q^k-1)/n} \\
&= \left(\prod_{i=0}^{l} f_{q^i,Q}^{c_i}(P)\right)^{(q^k-1)/n} \cdot \left(\prod_{i=0}^{l} f_{c_i,Q}^{q^i}(P) \cdot \prod_{i=0}^{l-1} \frac{l_{[\sum_{j=i+1}^{l} c_j q^j]Q,[c_i q^i]Q}(P)}{v_{[\sum_{j=i}^{l} c_j q^j]Q}(P)}\right)^{(q^k-1)/n}.
\end{aligned}$$

Now define Vercauteren's Pairing $a_{[c_0,c_1,\ldots,c_l]}$ to be

$$a_{[c_0,c_1,\ldots,c_l]} = \left(\prod_{i=0}^{l} f_{c_i,Q}^{q^i}(P) \cdot \prod_{i=0}^{l-1} \frac{l_{[\sum_{j=i+1}^{l} c_j q^j]Q,[c_i q^i]Q}(P)}{v_{[\sum_{j=i}^{l} c_j q^j]Q}(P)}\right)^{(q^k-1)/n}.$$

We can see that Vercauteren's Pairing is a ratio of pairings we already know, namely

$$a_{[c_0,c_1,\ldots,c_l]} = \frac{e_n^m(Q,P)}{\prod_{i=0}^{l} f_{q^i,Q}^{c_i}(P)}.$$

Therefore, Vercauteren's Pairing is bilinear and non-degenerate. For efficiency, we need to carefully choose the fixed power m so that all the c_i's are small. By Definition 3.1.1, as long as all $c_i \leq n^{\frac{1}{l+1}}$, Vercauteren's Pairing is an optimal pairing. Vercauteren [30] gives an algorithm to derive such m.

Consider the following $\varphi(k)$-dimensional lattice

$$L = \begin{pmatrix} n & 0 & 0 & \cdots & 0 \\ -q & 1 & 0 & \cdots & 0 \\ -q^2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \\ -q^l & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

In [30], it is concluded that finding the short vectors in the lattice L is easy, since k is small.

Let a short vector V be

$$\begin{aligned} V &= (\ v_0,\ v_1,\ v_2,\ \cdots,\ v_l\ ) \cdot \begin{pmatrix} n & 0 & 0 & \cdots & 0 \\ -q & 1 & 0 & \cdots & 0 \\ -q^2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \\ -q^l & 0 & \cdots & 0 & 1 \end{pmatrix} \\ &= (\ v_0 n - v_1 q - v_2 q^2 - \cdots - v_l q^l,\ v_1,\ v_2,\ \cdots,\ v_l\ ) \\ &= (\ c_0,\ c_1,\ c_2,\ \cdots,\ c_l\ ), \end{aligned}$$

where all $v_i \in \mathbb{Z}$. By Minkowski's theorem, there exists such short vector $V \in L$ with $\|V\|_\infty \leq n^{1/\varphi(k)}$, where $\|V\|_\infty = \max_i |c_i|$. Hence, we can have

$$v_0 n = c_0 + c_1 q + c_2 q^2 + \cdots + c_l q^l$$

with all $c_i \leq n^{1/\varphi(k)}$. Therefore, if we choose v_0 to be our fixed power m, Vercauteren's Pairing

$$a_{[c_0,c_1,\ldots,c_l]} = \left(\prod_{i=0}^{l} f_{c_i,Q}^{q^i}(P) \cdot \prod_{i=0}^{l-1} \frac{l_{[\sum_{j=i+1}^{l} c_j q^j]Q,[c_i q^i]Q}(P)}{v_{[\sum_{j=i}^{l} c_j q^j]Q}(P)} \right)^{(q^k-1)/n}$$

with $mn = \sum_{i=0}^{l} c_i q^i$ is an optimal pairing.

The relationship between Vercauteren's Pairing and the R-ate pairing is not clear. Both pairings can be written as a ratio of known pairings and a product of Miller functions and line functions. For the R-ate pairing, we look at the third case listed in the previous section, since it is the most used case for many pairing-friendly curves including BN curves. Vercauteren's Pairing can be written as

$$a_{[c_0,c_1,\ldots,c_l]} = \frac{e_n^m(Q,P)}{\prod_{i=0}^{l} f_{q^i,Q}^{c_i}(P)} = \left(\prod_{i=0}^{l} f_{c_i,Q}^{q^i}(P) \cdot \prod_{i=0}^{l-1} \frac{l_{[\sum_{j=i+1}^{l} c_j q^j]Q,[c_i q^i]Q}(P)}{v_{[\sum_{j=i}^{l} c_j q^j]Q}(P)} \right)^{(q^k-1)/n},$$

while the R-ate pairing can be written as

$$R_{A,B}(Q,P) = \left(\frac{f_{T_i,Q}(P)}{f^a_{T_j,Q}(P)}\right)^{(q^k-1)/n} = \left(f_{a,[T_j]Q}(P) \cdot f_{b,Q}(P) \cdot \frac{l_{[aT_j]Q,[b]Q}(P)}{v_{[q^i]Q}(P)}\right)^{(q^k-1)/n},$$

where $T_i \equiv q^i \pmod{n}$ for $i \in \mathbb{Z}$ with $0 < i < k$.

On one hand, there could be more than one coefficient among the c_i's not equal to 1, so it is not always possible to get Vercauteren's Pairing from the R-ate pairing. On the other hand, if the fixed power m in Vercauteren's Pairing could be chosen so that

$$a_{[c_0,c_1,\ldots,c_l]} = \frac{e^m_n(Q,P)}{\prod_{i=0}^{l} f^{c_i}_{q^i,Q}(P)} = \left(\frac{f_{T_i,Q}(P)}{f^a_{T_j,Q}(P)}\right)^{(q^k-1)/n},$$

then we need $mn = aq^j + b$ which might not be true in general. We conclude that Vercauteren's Pairing and the R-ate pairing are distinct optimal pairings.

Chapter 4

BN Curves

In this chapter, we study a family of elliptic curves defined over a prime field $\mathbb{F}_p$ such that the group of $\mathbb{F}_p$-rational points has prime order n. The curves have embedding degree $k = 12$ with respect to n. The curves were discovered in 2005 by Barreto and Naehrig [4].

4.1 Family of Curves

The family of elliptic curves is described in the following theorem [4].

Theorem 4.1.1 *Let $u \in \mathbb{Z}$ be an integer such that*

$$p = p(u) = 36u^4 + 36u^3 + 24u^2 + 6u + 1$$

and

$$n = n(u) = 36u^4 + 36u^3 + 18u^2 + 6u + 1$$

are prime numbers. Then there exists an ordinary elliptic E defined over $\mathbb{F}_p$ with $\#E(\mathbb{F}_p) = n$. The embedding degree of E with respect to n is $k = 12$, and the curve can be given by an equation

$$E : y^2 = x^3 + b, \qquad b \in \mathbb{F}_p.$$

The trace of the Frobenius endomorphism over $\mathbb{F}_p$ is given by $t = t(u) = 6u^2 + 1$.

A pair (p, n) of prime numbers as described in Theorem 4.1.1 is called a BN prime pair. Finding a BN prime pair of a certain bit size is not difficult. One just randomly chooses different numbers for parameter u until both $p(u)$ and $n(u)$ are prime. The values of u can be chosen so that $p(u)$ and $n(u)$ have a desired bit size.

For the curve equation $y^2 = x^3 + b$, the distribution of b is discussed as follows. Let $p \in \mathbb{N}$ be a prime. For $x_0 \in \mathbb{F}_p$, $b \in \mathbb{F}_p^*$ can be chosen such that $x_0^3 + b$ is a square y_0^2 in $\mathbb{F}_p$. Then, $P = (x_0, y_0)$ is an affine point on the curve $E : y^2 = x^3 + b$. If we randomly choose $b \in \mathbb{F}_p^*$, there is a chance of 50% to obtain a square $y_0^2 = x_0^3 + b$. Similarly, for $y_0 \in \mathbb{F}_p$, $b \in \mathbb{F}_p^*$ should be chosen so that $y_0^2 - b$ is a cube in $\mathbb{F}_p$. If we randomly choose $b \in \mathbb{F}_p^*$, there is a chance of $1/3$ to obtain a cube $x_0^3 = y_0^2 - b$. A lemma about b is given as follows.

Lemma 4.1.1 *Let $E : y^2 = x^3 + b$ be a BN curve defined over $\mathbb{F}_p$. Then b is neither a square nor a cube in $\mathbb{F}_p$. In particular, it is not a 6th power. If $P = (x_0, y_0) \in E(\mathbb{F}_p)$, then $x_0 \neq 0$ and $y_0 \neq 0$.*

Proof: First assume b is a square. Then $P = (0, \sqrt{b})$ is in $E(\mathbb{F}_p)$. Note that by the group law, $[2]P = -P$ so that P is a point of order 3. Since $\#E(\mathbb{F}_p) = n$ is a prime and $3 \nmid n$, this is a contradiction. Now assume that b is a cube. Then $Q = (-\sqrt[3]{b}, 0)$ is in $E(\mathbb{F}_p)$. Since the order of Q is 2, this contradicts $2 \nmid n$. Conversely, if $P = (x_0, y_0) \in E(\mathbb{F}_p)$, then $x_0 \neq 0$ and $y_0 \neq 0$. □

The condition $x_0 y_0 \neq 0$ is the only restriction for (x_0, y_0) to be a generator point. Let (p, n) be a BN prime pair, and let $x_0 \in \mathbb{F}_p^*$. Then, on average we expect 12 random choices for $b \in \mathbb{F}_p^*$ until the curve $E : y^2 = x^3 + b$ has order n and a generator with x-coordinate x_0.

With all the information given above, an algorithm for constructing BN curves can be formulated.

Algorithm 4.1.1 ***Algorithm to construct BN curves:***

Input: The expected bit length ℓ of the curve order n.

Output: Parameters p, n, b, y_0 such that the curve $E : y^2 = x^3 + b$ has order n over $\mathbb{F}_p$, the point $P = (1, y_0)$ is a generator of the curve, and n has bitlength at least ℓ.

1. *Let $p(u) = 36u^4 + 36u^3 + 24u^2 + 6u + 1$ and $n(u) = p(u) - 6u^2$.*
2. *Compute the smallest integer u such that $\log_2 n(-u) \geq \ell$.*
3. *Loop until p and n are prime.*
   (a) *Compute $t \leftarrow 6u^2 + 1$.*
   (b) *Compute $p \leftarrow p(-u)$ and $n \leftarrow p + 1 - t$.*
   (c) *If p and n are prime, then go to Step 4.*
   (d) *Compute $p \leftarrow p(u)$ and $n \leftarrow p + 1 - t$.*

(e) *If p and n are prime, then go to Step 4.*

(f) *Compute $u \leftarrow u + 1$ and go back to Step 3(a).*

4. *Loop until the curve $E : y^2 = x^3 + b$ has order n.*

   (a) *Choose $b \in \mathbb{F}_p^*$ at random.*

   (b) *If $b + 1$ is not a quadratic residue* $\bmod p$*, then go back to Step 4(a).*

   (c) *Compute y_0 such that $y_0^2 \equiv b + 1 \pmod{p}$ and set $P \leftarrow (1, y_0)$.*

   (d) *If $nP \neq \infty$, then go back to Step 4(a).*

5. *Return p, n, b, y_0.*

This algorithm gives a curve which has a generator with x-coordinate equal to 1.

4.2 Properties of BN Curves

Let E be an elliptic curve defined over $\mathbb{F}_q$. An elliptic curve E' defined over $\mathbb{F}_q$ is called a twist of E if there is an isomorphism ψ from E' to E that is defined over an extension field $\mathbb{F}_{q^d}$. The minimal extension degree d for which there exists such an isomorphism is called the degree of the twist E'.

The most important property of BN curves is the existence of a twist of degree 6. The second pairing argument defined over the field $\mathbb{F}_{p^{12}}$ is usually taken from the p-eigenspace of the Frobenius endomorphism on the n-torsion subgroup.

Lemma 4.2.1 *[23] Let $E/\mathbb{F}_p$ be a BN curve. The curve E has a unique twist $E'/\mathbb{F}_{p^2}$ of degree $d = 6$ with the following properties:*

1. *The order $\#E'(\mathbb{F}_{p^2})$ is divisible by n.*

2. *The twist can be represented by the equation*

$$E' : y^2 = x^3 + b/\xi,$$

   where $\xi \in \mathbb{F}_{p^2} \backslash ((\mathbb{F}_{p^2})^2 \cup (\mathbb{F}_{p^2})^3)$.

3. *The corresponding isomorphism ψ is given by*

$$\psi : E' \to E,$$
$$(x', y') \mapsto (\xi^{1/3} x', \xi^{1/2} y').$$

4. *A point $Q' \in E'(\mathbb{F}_{p^2})$ of order n is mapped via ψ into the p-eigenspace of the Frobenius endomorphism ϕ_p, i.e., $\phi_p(\psi(Q')) = [p]\psi(Q')$.*

In Section 2.1.4, we defined $G_1 = \ker(\phi_p - [1]) = E(\mathbb{F}_p)$ and $G_2 = E[n] \cap \ker(\phi_p - [p]) \subseteq E(\mathbb{F}_{p^{12}})[n]$. Lemma 4.2.1 shows that we can represent the group G_2 by the $\mathbb{F}_{p^2}$-rational points of order n on the twist E'. We can perform the elliptic curve operations on the twist instead of doing them in G_2. We define G_2' to be the group of $\mathbb{F}_{p^2}$-rational n-torsion points on the twist E',

$$G_2' = E'(\mathbb{F}_{p^2})[n].$$

A twisted pairing on a BN curve is then defined on $G_1 \times G_2'$ or $G_2' \times G_1$, where G_1, G_2 and G_2' are all cyclic groups of prime order n and ψ is a group isomorphism

$$\psi : G_2' \to G_2.$$

Since the twist E' is defined over $\mathbb{F}_{p^2}$, we construct the finite field $\mathbb{F}_{p^{12}}$ as an extension of $\mathbb{F}_{p^2}$. As in Lemma 4.2.1, $\xi \in \mathbb{F}_{p^2} \backslash ((\mathbb{F}_{p^2})^2 \cup (\mathbb{F}_{p^2})^3)$, and so the polynomials $x^2 - \xi$ and $x^3 - \xi$ are irreducible over $\mathbb{F}_{p^2}$ since otherwise ξ would be a square or a cube.

Lemma 4.2.2 *Let q be a prime power such that $q \equiv 1 \pmod 6$, and $\xi \in \mathbb{F}_{q^2} \backslash ((\mathbb{F}_{p^2})^2 \cup (\mathbb{F}_{q^2})^3)$. Then $x^6 - \xi \in \mathbb{F}_q[x]$ is irreducible over $\mathbb{F}_q$.*

Proof: Since $q \equiv 1 \pmod 6$, all 6-th roots of unity are in $\mathbb{F}_q$. Let ω be a root of $x^6 - \xi$ and $u \in \mathbb{F}_q$ be a primitive 6-th root of unity. Thus,

$$x^6 - \xi = \prod_{i=0}^{5} (x - u^i \omega).$$

Assume $x^6 - \xi$ has a cubic factor over $\mathbb{F}_q$. Then, the constant term of the cubic factor has the form $u^{i+j+k}\omega^3$ where i, j, k are distinct integers between 0 and 5. Thus, we have $\omega^3 \in \mathbb{F}_q$ so that $\xi = (\omega^3)^2$ is a square, which is a contradiction. Hence, $x^6 - \xi$ does not have a cubic factor. Similarly, $x^6 - \xi$ does not have a quadratic factor. Therefore, $x^6 - \xi$ is irreducible over $\mathbb{F}_p$. □

Let $\omega \in \mathbb{F}_{p^{12}}$ be a root of the irreducible polynomial $x^6 - \xi$, i.e., $\omega^6 = \xi$. The curve isomorphism ψ can be written as

$$\psi : E' \to E,$$
$$(x', y') \mapsto (\omega^2 x', \omega^3 y').$$

This map is needed during the pairing computation. The curve arithmetic in G_2 can be replaced by arithmetic in G_2'. For example, we compute the Tate pairing using the function

$$e_n : G_1 \times G_2' \to G_3,$$

$$e_n(P, Q') = f_{n,P}(\psi(Q'))^{(p^{12}-1)/n},$$

where $P \in G_1$, $Q' \in G_2'$. It will be shown in Chapter 5 that the map ψ from G_2' to G_2 can be computed at negligible cost. The efficiency of using BN curves will also be further analyzed in Chapter 5.

4.3 More on Curve Construction

The algorithm introduced in Section 4.1 gives a curve that works in general. However, for efficient pairing implementation, Pereira et al. defined a subfamily of BN curves [25].

Definition 4.3.1 *A BN curve $E/\mathbb{F}_p : y^2 = x^3 + b$ is called friendly if $p \equiv 3 \pmod 4$ and if there exist c, $d \in \mathbb{F}_p^*$ such that either $b = c^4 + d^6$ or $b = c^6 + 4d^4$.*

Such friendly BN curves have the following properties:

1. Since $p \equiv 3 \pmod 4$, we can represent $\mathbb{F}_{p^2}$ by $\mathbb{F}_p[i]/(i^2 + 1)$.
2. $\xi = c^2 + d^3 i$ or $\xi = c^3 + 2d^2 i$ is provided by c and d to represent $\mathbb{F}_{p^6} = \mathbb{F}_{p^2}[v]/(v^3 - \xi)$.
3. The sextic twist of correct order is given by $E' : y^2 = x^3 + \overline{\xi}$, where $\overline{\xi}$ is the complex conjugate of ξ.
4. $(-d^2, c^2)$ and $(-c^2, 2d^2)$ are generators of G_2.
5. $(-di, c)$ and $(-c, d(1 - i))$ are generators of G_2'.

Note that with the help of these two parameters c and d, we can efficiently define the group generators, sextic twist, and field extensions. Some other suggestions for improving efficiency are provided in [25].

1. The Hamming weights of either the BN parameter u or the loop order $6u + 2$ of the optimal pairings are minimal for a given bitlength.
2. b is as small as possible.
3. Carefully choose c and d so that b and ξ has low Hamming weight. Thus, multiplication by b or ξ consists only of shifts and additions.

We use a specific BN curve suggested in [25] for implementation in Chapter 5 and Chapter 6.

Chapter 5

Implementing the R-ate Pairing using BN Curves

5.1 R-ate Pairing on a Particular BN Curve

In this section, a particular BN curve is introduced and will be used for the remainder of this thesis. We have

$$E : y^2 = x^3 + 2, \tag{5.1}$$

and

$$u = -(2^{65} + 2^{55} + 1) < 0.$$

Recall the R-ate pairing introduced in Section 3.3. In the third case (see (3.7)), we choose $(A, B) = (T_{10}, T_1)$. Thus,

$$R_{A,B}(Q, P) = \left(f_{a,Q}^{p}(P) \cdot f_{b,Q}(P) \cdot \frac{\ell_{[aT_1]Q,[b]Q}(P)}{v_{[p^{10}]Q}(P)} \right)^{(p^{12}-1)/n},$$

where $a = 6u + 3$, $b = 6u + 2$, $T_1 = p \mod n$ and $T_{10} = p^{10} \mod n$. Since for any point $Q \in G_2$ we have $[p]Q = \phi_p(Q)$, the line function

$$\ell_{[aT_1]Q,[b]Q}(P) = \ell_{[ap]Q,[b]Q}(P) = \ell_{\phi_p([a]Q),[b]Q}(P)$$

can be transformed so that its evaluation is faster. Hence, we can write the R-ate pairing as

$$R_{A,B}(Q, P) = \left(\left(f_{b,Q}(P) \cdot \frac{\ell_{[b]Q,Q}(P)}{v_{[b+1]Q}(P)} \right)^{p} \cdot f_{b,Q}(P) \cdot \frac{\ell_{\phi_p([a]Q),[b]Q}(P)}{v_{[q^{10}]Q}(P)} \right)^{(p^{12}-1)/n}.$$

We call $b = 6u + 2$ the R-ate parameter. Note that the R-ate parameter is negative, so we compute $f_{|b|,Q}$ instead of $f_{b,Q}$ in the Miller Loop. Since

$$\begin{aligned}(f_{|b|,Q}^{-1}) &= -(|b|(Q) - ([|b|]Q) - (|b| - 1)(\infty)) \\ &= b(Q) + ([-b]Q) + (-b - 1)(\infty)\end{aligned}$$

and

$$(f_{b,Q}) = b(Q) - ([b]Q) - (b - 1)(\infty),$$

we have

$$(f_{|b|,Q}^{-1}) - (f_{b,Q}) = ([-b]Q) + ([b]Q) - 2(\infty),$$

and thus $(f_{|b|,Q}^{-1})$ is equivalent to $(f_{b,Q})$. In order not to affect the pairing value, it is required to compute a curve point negation in G_2 and an inversion in G_3 before we apply the final exponentiation. The curve parameters are summarized here:

1. Embedding degree $k = 12$.
2. $u = -(2^{65} + 2^{55} + 1)$.
3. $p = 36u^4 + 36u^3 + 24u^2 + 6u + 1$ is a 254-bit prime of Hamming weight 43 and $p \equiv 1 \pmod 6$, $p \equiv 3 \pmod 4$.
4. $n = 36u^4 + 36u^3 + 18u^2 + 6u + 1$ is a 254-bit prime of Hamming weight 51.
5. $t = 6u^2 + 1$ is a 127-bit integer of Hamming weight 13.
6. $b = 6u + 2$ is a 65-bit integer of Hamming weight 5.

It will be shown in Section 5.2 that all vertical lines are equal to 1 after the final exponentiation. Hence, we can ignore the evaluation of the vertical lines during the pairing computation and rewrite the function as

$$R_{A,B}(Q, P) = \left(\left(f_{b,Q}(P) \cdot l_{[b]Q,Q}(P)\right)^p \cdot f_{b,Q}(P) \cdot l_{[aT_1]Q,[b]Q}(P)\right)^{(p^{12}-1)/n}.$$

The algorithm to compute this particular pairing is given as follows.

Algorithm 5.1.1 ***Algorithm to compute the R-ate pairing on BN curves (where the R-ate parameter is negative):***

Input: $P \in G_1$, $Q \in G_2$, $b = |6u + 2| = \sum_{i=0}^{\log_2(b)} b_i 2^i$

Output: $R_{A,B}(Q, P)$

1. $T \leftarrow Q$, $f \leftarrow 1$
2. *for* $i = \lfloor \log_2(b) \rfloor - 1$ *to* 0
   (a) $f \leftarrow f^2 \cdot \ell_{T,T}(P)$, $T \leftarrow 2T$
   (b) *if* $b_i = 1$, *then* $f \leftarrow f \cdot \ell_{T,Q}(P)$, $T \leftarrow T \leftarrow T + Q$
3. $T \leftarrow -T$, $f \leftarrow f^{-1}$
4. $f \leftarrow f \cdot (f \cdot \ell_{T,Q}(P))^p \cdot \ell_{\pi(T+Q),T}(P)$
5. $f \leftarrow f^{(p^{12}-1)/n}$
6. *Return* $R_{A,B}(Q, P) = f$.

5.2 Tower Extension

To achieve a high performance, a standard method is to represent $\mathbb{F}_{p^{12}}$ using tower extensions. It is clear that $x^2 - (-1)$ is irreducible in $\mathbb{F}_p$. Thus, the extension field $\mathbb{F}_{p^2}$ can be represented as $\mathbb{F}_p[i]/(i^2 + 1)$. We use the field extension

- $\mathbb{F}_{p^2} = \mathbb{F}_p[i]/(i^2 - \beta)$, where $\beta = -1$,
- $\mathbb{F}_{p^4} = \mathbb{F}_{p^2}[s]/(s^2 - \xi)$, where $\xi = 1 + i$,
- $\mathbb{F}_{p^6} = \mathbb{F}_{p^2}[v]/(v^3 - \xi)$,
- $\mathbb{F}_{p^{12}} = \mathbb{F}_{p^2}[\omega]/(\omega^6 - \xi)$

suggested and implemented in [1][25]. We also have $\mathbb{F}_{p^{12}} = \mathbb{F}_{p^4}[\omega]/(\omega^3 - s)$ and $\mathbb{F}_{p^{12}} = \mathbb{F}_{p^6}[\omega]/(\omega^2 - v)$. Here is a short proof that ξ is neither a square nor a cube in $\mathbb{F}_{p^2}$.

Proof: Assume that $\xi \in \mathbb{F}_{p^2}$ is a square. In our chosen curve, we have $b = 2 = (1+i)\cdot(1-i)$. Then,

$$\begin{aligned} b &= \xi \cdot \overline{\xi} \\ &= s^2 \cdot \overline{s^2} \qquad \text{for some } s \in \mathbb{F}_{p^2} \\ &= (s \cdot \overline{s})^2 \end{aligned}$$

Hence, b is square which contradicts Lemma 4.1.1. We conclude that ξ is not a square. Similarly, ξ is not a cube. □

An element $\alpha \in \mathbb{F}_{p^{12}}$ can be represented in the following ways:

$$\begin{aligned}
\alpha &= a_0 + a_1\omega \qquad \text{where } a_0, a_1 \in \mathbb{F}_{p^6} \\
&= (a_{0,0} + a_{0,1}v + a_{0,2}v^2) + (a_{1,0} + a_{1,1}v + a_{1,2}v^2)\omega \qquad \text{where } a_{i,j} \in \mathbb{F}_{p^2} \\
&= (a_{0,0} + a_{1,1}s) + (a_{1,0} + a_{0,2}s)\omega + (a_{0,1} + a_{1,2}s)\omega^2 \\
&= a_{0,0} + a_{1,0}\omega + a_{0,1}\omega^2 + a_{1,1}\omega^3 + a_{0,2}\omega^4 + a_{1,2}\omega^5.
\end{aligned}$$

Note that converting from one towering $\mathbb{F}_{p^2} \to \mathbb{F}_{p^6} \to \mathbb{F}_{p^{12}}$ to the other $\mathbb{F}_{p^2} \to \mathbb{F}_{p^4} \to \mathbb{F}_{p^{12}}$ is simply permuting the order of the $\mathbb{F}_{p^2}$ coefficients.

Recall the existence of sextic twists introduced in Section 4.2. The twist can be represented by the equation $E' : y^2 = x^3 + 2/\xi$. Since $(1+i)(1-i) = 2$, we have $1 - i = 2/\xi$. The sextic twist can thus be represented by equation $E'/\mathbb{F}_{p^2} : y^2 = x^3 + 1 - i$ and the corresponding isomorphism ψ is given by

$$\psi : E' \to E,$$

$$(x', y') \mapsto (x'\omega^2, y'\omega^3).$$

Mapping a point in G_2' to a point in G_2 can be considered as changing both the coordinates of the former into two 6-dimensional vectors,

$$(x', y') \mapsto ((0, 0, x', 0, 0, 0), (0, 0, 0, y', 0, 0)) \tag{5.2}$$

Therefore, the cost of the map ψ is negligible.

Let $P = (x_1, y_1) \in G_1$ and $Q = (x_2, y_2) \in G_2$. Obviously, we have $x_1, y_1 \in \mathbb{F}_p$. By (5.2), we see that x_2 and y_2 have the form $x_2 = x_2'\omega^2 \in \mathbb{F}_{p^6}$ and $y_2 = y_2'\omega^3 \in \mathbb{F}_{p^4}$ where x_2' and $y_2' \in \mathbb{F}_{p^2}$. For any $P = (x_1, y_1) \in G_1$ and $Q = (x_2, y_2) \in G_2$, recall that $v(P) = x_1 - x_2$ is the formula of the vertical line through Q evaluated at P. Since $x_1 \in \mathbb{F}_p$ and $x_2 \in \mathbb{F}_{p^6}$, these function values $x_1 - x_2$ are all in $\mathbb{F}_{p^6} \subset \mathbb{F}_{p^{12}}$. By Lemma 2.3.1, after we apply the final exponentiation, we can ignore the evaluation of the vertical lines during the pairing computation. This is called *denominator elimination*.

Let (M, S, I) denote the cost of multiplication, squaring and inversion in $\mathbb{F}_p$. Let (M_d, S_d, I_d) for $d \in 2, 4, 6, 12$ denote the cost of multiplication, squaring and inversion in $\mathbb{F}_{p^d}$. To roughly determine the efficiency of the whole pairing computation, we convert all operation counts into M. Since the cost of addition is negligible compared with the cost of multiplication, we only count the number of multiplications. Experimentally [15], $S \approx 0.9M$ and $I \approx 41m$. In this thesis, we assume $S \approx M$.

Lemma 5.2.1 *If $a \in \mathbb{F}_p$ and $b \in \mathbb{F}_{p^d}$ for $d \in 2, 4, 6, 12$, then the cost of computing $a \cdot b$ is dM.*

Proof: Suppose $\mathbb{F}_{p^d} = \mathbb{F}_p[\omega]$. We have

$$b = \sum_{i=0}^{d-1} b_i \omega^i$$

where $b_i \in \mathbb{F}_p$. Thus,

$$a \cdot b = a \cdot \sum_{i=0}^{d-1} b_i \omega^i = \sum_{i=0}^{d-1} (a b_i) \omega^i;$$

the last expression has d multiplications in $\mathbb{F}_p$. □

This lemma can be extended as follows: if $a \in \mathbb{F}_{p^d}$ and $b \in \mathbb{F}_{p^{ed}}$ for $d \in \{2, 4, 6, 12\}$ and $e \in \mathbb{N}$, then the cost of computing $a \cdot b$ is eM_d.

5.2.1 $\mathbb{F}_{p^2}$ Arithmetic

Let $a+bi$ and $a'+b'i$ be two abritrary $\mathbb{F}_{p^2}$ elements with a, b, a', $b' \in \mathbb{F}_p$. Using Karatsuba's method,

$$\begin{aligned} (a + bi) \cdot (a' + b'i) &= aa' + bb'i^2 + ab'i + a'bi \\ &= (aa' - bb') + ((a + b) \cdot (a' + b') - aa' - bb')\, i, \end{aligned}$$

which reduces one multiplication in a quadratic extension to three small field multiplications (i.e., $a \cdot a'$, $b \cdot b'$ and $(a + b) \cdot (a' + b')$); thus we have $M_2 \approx 3M$. For squaring in $\mathbb{F}_{p^2}$, we have

$$\begin{aligned} (a + bi)^2 &= a^2 + 2abi + b^2 i^2 \\ &= (a^2 - b^2) + 2abi \\ &= (a + b) \cdot (a - b) + 2abi, \end{aligned}$$

and $S_2 \approx 2M$ (i.e., the two $\mathbb{F}_p$ multiplications are $(a + b) \cdot (a - b)$ and $a \cdot b$). Since $(a + bi)(a - bi) = (a^2 + b^2)$, we have

$$(a + bi)^{-1} = \frac{a - bi}{a^2 + b^2}.$$

Hence, to compute an inversion in a quadratic extension, we need to compute two squarings (i.e., a^2 and b^2), one inversion (i.e., $(a^2 + b^2)^{-1}$), and two multiplications (i.e., by Lemma 5.2.1, $(a^2 + b^2)^{-1} \cdot (a - bi)$) in the small field, so $I_2 \approx 2S + I + 2M \approx I + 4M$.

Lemma 5.2.2 *Let $a + bi$ be an arbitrary $\mathbb{F}_{p^2}$ element with a, $b \in \mathbb{F}_p$. Then the cost of p-th powering is negligible in $\mathbb{F}_{p^2}$.*

Proof: The binomial theorem gives

$$\begin{aligned}
(a+bi)^p &= \sum_{d=0}^{p} \left(\begin{array}{c} p \\ i \end{array} \right) \cdot a^d \cdot (bi)^{p-d} \\
&= a^p + b^p i^p \qquad \text{since all other terms are multiples of } p \\
&= a + bi \cdot i^{p-1} \\
&= a + bi \cdot (\beta)^{\frac{p-1}{2}} \\
&= a + bi \cdot (-1) \qquad \text{since } \beta \text{ is a quadratic non-residue} \\
&= a - bi.
\end{aligned}$$

Thus, one can raise an element in $\mathbb{F}_{p^2}$ to the p-th power by just changing the sign of the second component. □

Lemma 5.2.2 can be extended as follows: the cost of p^d-th powering is negligible in $\mathbb{F}_{p^{2d}}$.

5.2.2 $\mathbb{F}_{p^6}$ Arithmetic

Let $a + bv + cv^2$ and $a' + b'v + c'v^2$ be two arbitrary $\mathbb{F}_{p^6}$ elements with a, b, c, a', b' and $c' \in \mathbb{F}_{p^2}$. Using Karatsuba's method,

$$\begin{aligned}
(a + bv + cv^2) \cdot (a' + b'v + c'v^2) &= aa' + (ab' + a'b)v + (ac' + a'c + bb')v^2 + (bc' + b'c)v^3 \\
&\quad + cc'v^4 \\
&= (aa' + (bc' + b'c)\xi) + (cc'\xi + (ab' + a'b))\, v \\
&\quad + ((ac' + a'c) + bb'))\, v^2 \\
&= (aa' + ((b+c)(b'+c') - bb' - cc')\xi) \\
&\quad + (cc'\xi + ((a+b)(a'+b') - aa' - bb'))\, v \\
&\quad + (((a+c)(a'+c') - aa' - cc') + bb'))\, v^2,
\end{aligned}$$

which reduces one multiplication in a cubic extension to six small field multiplications (i.e., $a \cdot a'$, $b \cdot b'$, $c \cdot c'$, $(a+b) \cdot (a'+b')$, $(a+c) \cdot (a'+c')$ and $(b+c) \cdot (b'+c')$); we have $M_6 \approx 6M_2 \approx 18M$. Note that we did not count an $\mathbb{F}_{p^2}$ multiplication by ξ as a multiplication for the following reason. Let $a + bi$ be an arbitrary $\mathbb{F}_{p^2}$ element; then

$$\begin{aligned}
(a+bi) \cdot \xi &= (a+bi)(1+i) \\
&= a + bi + ai + bi^2 \\
&= (a-b) + (a+b)i,
\end{aligned}$$

which means the cost of multiplication by ξ is as cheap as two additions in $\mathbb{F}_{p^2}$. For squaring in $\mathbb{F}_{p^6}$, we have the following formula from [9]:

$$\begin{aligned}(a+bv+cv^2)^2 &= a^2+b^2v^2+c^2v^4+2abv+2acv^2+2bcv^3\\ &= (a^2+2bc\xi)+(2ab+c^2\xi)v+(b^2+2ac)v^2\\ &= (a^2+2bc\xi)+(2ab+c^2\xi)v+\left((a-b+c)^2-a^2-c^2+2ab+2bc\right)v^2\end{aligned}$$

Hence, to compute a squaring in $\mathbb{F}_{p^6}$ we need to compute two multiplications (i.e., $a\cdot b$ and $b\cdot c$) and three squarings (i.e., a^2, c^2 and $(a-b+c)^2$) in $\mathbb{F}_{p^2}$. Thus, $S_6 \approx 2M_2+3S_2 \approx 12M$. Finally, the formula for inversion in cubic extension field is provided in [27] as

$$(a+bv+cv^2)^{-1} = \frac{A+Bv+Cv^2}{bC\xi+aA+cB\xi},$$

where $A=a^2-bc\xi$, $B=c^2\xi-ab$, and $C=b^2-ac$. Hence, inversion in $\mathbb{F}_{p^6}$ can be reduced to three squarings (i.e., a^2, b^2 and c^2), nine multiplications (i.e., six multiplications for $b\cdot c$, $a\cdot b$, $a\cdot c$, $b\cdot C$, $a\cdot A$, $c\cdot B$ and three multiplications for $(A+Bv+Cv^2)\cdot(bC\xi+aA+cB\xi)^{-1}$ by Lemma 5.2.1) and one inversion (i.e., $(bC\xi+aA+cB\xi)^{-1}$) in $\mathbb{F}_{p^2}$. Thus we have $I_6 \approx 3S_2+9M_2+I_2 \approx 37M+I$.

5.2.3 $\mathbb{F}_{p^{12}}$ Arithmetic

Since $\mathbb{F}_{p^{12}}$ is a tower of quadratic, cubic and quadratic extensions. Karatsuba's method gives $M_{12} \approx 3M_6 \approx 54M$. As in the $\mathbb{F}_{p^2}$ case, we have $S_{12} \approx 2M_6 \approx 36M$.

Lemma 5.2.3 *[29] If $\alpha = a+b\omega \in \mathbb{F}_{p^{12}}$ with a, $b \in \mathbb{F}_{p^6}$ satisfies $\alpha^{p^6+1}=1$, then α^2 can be computed in roughly $24M$.*

Proof: Note that, by Lemma 5.2.2, for any $\alpha = a+b\omega \in \mathbb{F}_{p^{12}}$ with a, $b \in \mathbb{F}_{p^6}$, we have $\alpha^{p^6} = a-b\omega$. Thus

$$\begin{aligned}1 &= \alpha^{p^6+1}\\ &= (a+b\omega)\cdot(a-b\omega)\\ &= a^2-b^2\omega^2,\end{aligned}$$

which means a^2 can be written as $a^2 = b^2\omega^2+1 = b^2v+1$. Then

$$\begin{aligned}\alpha^2 &= (a+b\omega)^2\\ &= a^2+2ab\omega+b^2\omega^2\\ &= (a^2+b^2v)+(2ab)\omega\\ &= (2b^2v+1)+\left((a+b)^2-a^2-b^2\right)\\ &= (2b^2v+1)+\left((a+b)^2-(b^2v+1)-b^2\right).\end{aligned}$$

Hence, squaring such α can be accomplished with two $\mathbb{F}_{p^6}$ squarings (i.e., b^2 and $(a+b)^2$) for a total cost of $24M$. □

We denote such squaring by S'_{12}. It is used in the final exponentiation steps during the pairing computation. Inverting such α is essentially free since we have

$$\alpha^{-1} = \alpha^{p^6} = a - b\omega.$$

Finally, inversion in $\mathbb{F}_{p^{12}}$ can be reduced to two squarings, one inversion and two multiplications in $\mathbb{F}_{p^6}$. Thus we have $I_{12} \approx 2S_6 + I_6 + 2M_6 \approx 97M + I$.

5.2.4 Summary

The costs of arithmetic operation in $\mathbb{F}_p$, $\mathbb{F}_{p^2}$, $\mathbb{F}_{p^6}$ and $\mathbb{F}_{p^{12}}$ are summarized in Table 5.1.

Operation	Cost
Multiplication in $\mathbb{F}_p$	M
Squaring in $\mathbb{F}_p$	S $\approx$ M
Inversion in $\mathbb{F}_p$	I
Multiplication in $\mathbb{F}_{p^2}$	3M
Squaring in $\mathbb{F}_{p^2}$	2M
Inversion in $\mathbb{F}_{p^2}$	I+4M
Multiplication in $\mathbb{F}_{p^6}$	18M
Squaring in $\mathbb{F}_{p^6}$	12M
Inversion in $\mathbb{F}_{p^6}$	I+37M
Multiplication in $\mathbb{F}_{p^{12}}$	54M
Squaring in $\mathbb{F}_{p^{12}}$	36M
Inversion in $\mathbb{F}_{p^{12}}$	I+97M

Table 5.1: Cost estimates for arithmetic operations in $\mathbb{F}_p$, $\mathbb{F}_{p^2}$, $\mathbb{F}_{p^6}$ and $\mathbb{F}_{p^{12}}$

5.3 Operation Count for R-ate Pairings

In order to carefully count the operations, we restate the algorithm to compute the R-ate pairing on this specific BN curve.

Algorithm 5.3.1 ***Algorithm to compute the R-ate pairing on BN curve (5.1):***

Input:

- $P = (x_P, y_P) \in G_1$ *with* x_P, $y_P \in \mathbb{F}_p$,
- $Q = (x_Q\omega^2, y_Q\omega^3) \in G_2$ *with* x_Q, $y_Q \in \mathbb{F}_{p^2}$,
- $b = |6u+2| = \sum_{i=0}^{65} b_i 2^i$.

Output: $R_{A,B}(Q,P)$.

1. $T \leftarrow (x_Q, y_Q, 1)$, $f \leftarrow 1$.
2. *For* i *from* 63 *to* 0 *do:*
   (a) $f \leftarrow f^2 \cdot \ell_{T,T}(P)$.
   (b) $T \leftarrow 2T$.
   (c) *If* $b_i = 1$, *then* $f \leftarrow f \cdot \ell_{T,Q}(P)$, $T \leftarrow T + Q$.
3. $T \leftarrow -T$, $f \leftarrow f^{-1}$.
4. *Compute* $f \cdot (f \cdot \ell_{T,Q}(P))^p \cdot \ell_{\phi_p(T+Q),T}(P)$ *as follows:*
   (a) *Convert* T *to affine coordinates and store as* T'.
   (b) $f' \leftarrow f \cdot \ell_{T,Q}(P)$, $T \leftarrow T + Q$.
   (c) $f' \leftarrow (f')^p$.
   (d) $f' \leftarrow f' \cdot \ell_{\phi_p(T),T'}(P)$.
   (e) $f \leftarrow f \cdot f'$.
5. *Compute* $f^{(p^{12}-1)/n}$ *as follows (see Section 5.3.3):*
   (a) $f \leftarrow f^{p^6-1}$.
   (b) $f \leftarrow f^{p^2+1}$.
   (c) $a \leftarrow f^{-6u-5}$.
   (d) $b \leftarrow a^p$.
   (e) $b \leftarrow a \cdot b$.
   (f) $f \leftarrow f^{p^3} \cdot [b \cdot (f^p)^2 \cdot f^{p^2}]^{6u^2+1} \cdot b \cdot (f^p \cdot f)^9 \cdot a \cdot f^4$.
6. *Return* $R_{A,B}(Q,P) = f$.

In this algorithm, Step 2 is called the Miller loop. Step 3 and Step 4 are the adjustment steps for this specific pairing. Step 5 is called the final exponentiation. Next we perform the operation counts for these three parts.

5.3.1 Operation Count for the Miller Loop

Recall the sextic twist E' of E over $\mathbb{F}_{p^2}$,

$$E'/\mathbb{F}_{p^2} : y^2 = x^3 + 1 - i,$$

and the corresponding isomorphism ψ given by

$$\psi : E' \to E,$$

$$(x', y') \mapsto (x'\omega^2, y'\omega^3).$$

Note that ψ as well as its inverse can be computed at negligible cost. At the beginning of the algorithm, we assign T to be in $E'/\mathbb{F}_{p^2}$. To avoid inversions, G_2' is usually represented in projective coordinates and jacobian coordinates. We shall use jacobian coordinates. A point (X, Y, Z) in jacobian coordinates corresponds to the point (x, y) in affine coordinates with $x = X/Z^2$ and $y = Y/Z^3$. Recall the point doubling formula introduced in Section 2.1.1. Let $P = (x, y)$ with $P \neq -P$. Then $2P = (x_3, y_3)$, where

$$x_3 = \left(\frac{3x^2}{2y}\right)^2 - 2x \tag{5.3}$$

$$y_3 = \left(\frac{3x^2}{2y}\right)(x - x_3) - y. \tag{5.4}$$

In jacobian coordinates we derive the formula for doubling a point $T = (X, Y, Z)$ to $2T = (X_3, Y_3, Z_3)$. Substituting $x = X/Z^2$, $y = Y/Z^3$, $x_3 = X_3/Z_3^2$ and $y_3 = Y_3/Z_3^3$ into (5.3) gives

$$\begin{aligned}
\frac{X_3}{Z_3^2} &= \left(\frac{3(\frac{X}{Z^2})^2}{2(\frac{Y}{Z^3})}\right)^2 - 2\left(\frac{X}{Z^2}\right) \\
&= \frac{9X^4}{Z^8} \cdot \frac{Z^6}{4Y^2} - \frac{2X}{Z^2} \\
&= \frac{9X^4}{4Y^2Z^2} - \frac{8XY^2}{4Y^2Z^2} \\
&= \frac{9X^4 - 8XY^2}{4Y^2Z^2}.
\end{aligned}$$

Similarly, substituting $x = X/Z^2$, $y = Y/Z^3$, $x_3 = X_3/Z_3^2$ and $y_3 = Y_3/Z_3^3$ into (5.4) gives

$$\begin{aligned}
\frac{Y_3}{Z_3^3} &= \left(\frac{3(\frac{X}{Z^2})^2}{2(\frac{Y}{Z^3})}\right) \cdot \left(\frac{X}{Z^2} - \frac{X_3}{Z_3^2}\right) - \frac{Y}{Z^3} \\
&= \left(\frac{3X^2}{Z^4} \cdot \frac{Z^3}{2Y}\right) \cdot \left(\frac{X}{Z^2} - \frac{X_3}{Z_3^2}\right) - \frac{Y}{Z^3} \\
&= \frac{3X^2}{2YZ} \cdot \left(\frac{4XY^2}{4Y^2Z^2} - \frac{9X^4 - 8XY^2}{4Y^2Z^2}\right) - \frac{8Y^4}{8Y^3Z^3} \\
&= \frac{3X^2(4XY^2 - (9X^4 - 8XY^2)) - 8Y^4}{8Y^3Z^3}.
\end{aligned}$$

Hence we have $2T = (X_3, Y_3, Z_3)$ where

$$\begin{cases}
X_3 = 9X^4 - 8XY^2 \\
Y_3 = (3X^2)(4XY^2 - X_3) - 8Y^4 \\
Z_3 = 2YZ.
\end{cases}$$

Since we are doing the curve arithmetic on $E'/\mathbb{F}_{p^2}$, one point doubling requires four squarings (i.e., X^2, Y^2, $(X^2)^2$ and $(Y^2)^2$) and three multiplications (i.e., $X \cdot Y^2$, $(3X^2) \cdot (4XY^2 - X_3)$ and $Y \cdot Z$) in $\mathbb{F}_{p^2}$. Thus, each point doubling in G_2' costs $4S_2 + 3M_2 = 17M$.

The value of the tangent line through T at P is always computed together with the point doubling. Recall the line function from Section 2.3.1: for $P = (x_P, y_P) \in G_1$, $T = (x_1, y_1) \in G_2$, the formula of the tangent line through Q evaluated at P is

$$\ell_{T,T}(P) = y_P - y_1 - \frac{3x_1^2}{2y_1}(x_P - x_1).$$

Different from the point doubling formula, this line must be evaluated in $\mathbb{F}_{p^{12}}$. Let $T = (x_1, y_1) = (x_T\omega^2, y_T\omega^3)$, and let (X, Y, Z) be the point we stored in jacobian coordinates such that $x_T = X/Z^2$ and $y_T = Y/Z^3$. Then

$$\begin{aligned}
\ell_{T,T}(P) &= y_P - y_1 - \frac{3x_1^2}{2y_1}(x_P - x_1) \\
&= y_P - y_T\omega^3 - \frac{3(x_T\omega^2)^2}{2y_T\omega^3}(x_P - x_T\omega^2) \\
&= y_P - \frac{Y}{Z^3}\omega^3 - \frac{3(\frac{X}{Z^2}\omega^2)^2}{2\frac{Y}{Z^3}\omega^3}(x_P - \frac{X}{Z^2}\omega^2) \\
&= \frac{y_PZ^3 - Y\omega^3}{Z^3} - \frac{3X^2\omega^4}{Z^4} \cdot \frac{Z^3}{2Y\omega^3} \cdot \frac{x_PZ^2 - X\omega^2}{Z^2} \\
&= \frac{2y_PYZ^3 - 2Y^2\omega^3 - 3X^2\omega(x_PZ^2 - X\omega^2)}{2YZ^3}.
\end{aligned}$$

Since $2YZ^3 \in \mathbb{F}_{p^2}$, the final exponentiation will convert it to 1. Thus, we take

$$\begin{aligned} \ell_{T,T}(P) &= 2y_P YZ^3 - 2Y^2\omega^3 - 3X^2\omega(x_P Z^2 - X\omega^2) \\ &= 2y_P YZ^3 - 3x_P X^2 Z^2 \omega + (3X^3 - 2Y^2)\omega^3. \end{aligned}$$

Note that X^2, Y^2 and YZ are also computed while doubling T, so we can share these results here. Therefore, to compute $\ell_{T,T}(P)$ we need one squaring (i.e., Z^2) in $\mathbb{F}_{p^2}$, three multiplications (i.e., $X \cdot X^2$, $Z^2 \cdot X^2$ and $Z^2 \cdot YZ$) in $\mathbb{F}_{p^2}$ and four multiplications (i.e., two for $y_P \cdot YZ^3$ and two for $x_P \cdot X^2Z^2$) in $\mathbb{F}_p$, for a total of $S_2 + 3M_2 + 4M \approx 15M$.

Let $T = (x_1, y_1) \in G_2$ and $Q = (x_2, y_2) \in G_2$ with $T \neq \pm Q$. The point addition formula in Section 2.1.1 is given as $T + Q = (x_3, y_3)$ where

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2 \tag{5.5}$$

$$y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1. \tag{5.6}$$

Let $T = (X_1, Y_1, Z_1) \in G_2'$ be the point we stored in jacobian coordinates and $Q = (x_Q\omega^2, y_Q\omega^3) \in G_2$ in affine coordinates. Substituting $x_1 = X_1/Z_1^2$, $y_1 = Y_1/Z_1^3$, $x_3 = X_3/Z_3^2$ and $y_3 = Y_3/Z_3^3$ into (5.5) gives

$$\begin{aligned} \frac{X_3}{Z_3^2} &= \left(\frac{y_Q - \frac{Y_1}{Z_1^3}}{x_Q - \frac{X_1}{Z_1^2}}\right)^2 - \frac{X_1}{Z_1^2} - x_Q \\ &= \left(\frac{y_Q Z_1^3 - Y_1}{Z_1^3} \cdot \frac{Z_1^2}{x_Q Z_1^2 - X_1}\right)^2 - \frac{X_1 \cdot (x_Q Z_1^2 - X_1)^2}{Z_1^2 \cdot (x_Q Z_1^2 - X_1)^2} - \frac{x_Q \cdot Z_1^2 (x_Q Z_1^2 - X_1)^2}{Z_1^2 (x_Q Z_1^2 - X_1)^2} \\ &= \frac{(y_Q Z_1^3 - Y_1)^2 - (X_1 + x_Q Z_1^2)(x_Q Z_1^2 - X_1)^2}{Z_1^2 (x_Q Z_1^2 - X_1)^2} \end{aligned}$$

Similarly, substituting $x_1 = X_1/Z_1^2$, $y_1 = Y_1/Z_1^3$, $x_3 = X_3/Z_3^2$ and $y_3 = Y_3/Z_3^3$ into (5.6) gives

$$\begin{aligned} \frac{Y_3}{Z_3^2} &= \left(\frac{y_Q - \frac{Y_1}{Z_1^3}}{x_Q - \frac{X_1}{Z_1^2}}\right) \cdot \left(\frac{X_1}{Z_1^2} - \frac{X_3}{Z_3^2}\right) - \frac{Y_1}{Z_1^3} \\ &= \left(\frac{y_Q Z_1^3 - Y_1}{Z_1^3} \cdot \frac{Z_1^2}{x_Q Z_1^2 - X_1}\right) \cdot \left(\frac{X_1}{Z_1^2} - \frac{(y_Q Z_1^3 - Y_1)^2 - (X_1 + x_Q Z_1^2)(x_Q Z_1^2 - X_1)^2}{Z_1^2 (x_Q Z_1^2 - X_1)^2}\right) \\ &\quad - \frac{Y_1}{Z_1^3} \\ &= \frac{(y_Q Z_1^3 - Y_1)\{X_1 (x_Q Z_1^2 - X_1)^2 - [(y_Q Z_1^3 - Y_1)^2 - (X_1 + x_Q Z_1^2)(x_Q Z_1^2 - X_1)^2]\}}{Z_1^3 (x_Q Z_1^2 - X_1)^3} \\ &\quad - \frac{Y_1 (x_Q Z_1^2 - X_1)^3}{Z_1^3 (x_Q Z_1^2 - X_1)^3}. \end{aligned}$$

Hence we have $T+Q=(X_3,Y_3,Z_3)$ with

$$\begin{cases} X_3=(y_QZ_1^3-Y_1)^2-(X_1+x_QZ_1^2)(x_QZ_1^2-X_1)^2 \\ Y_3=(y_QZ_1^3-Y_1)[X_1(x_QZ_1^2-X_1)^2-X_3]-Y_1(x_QZ_1^2-X_1)^3 \\ Z_3=Z_1(x_QZ_1^2-X_1). \end{cases}$$

We rewrite X_3 as

$$\begin{aligned} X_3 &= (y_QZ_1^3-Y_1)^2-(X_1+x_QZ_1^2)(x_QZ_1^2-X_1)^2 \\ &= (y_QZ_1^3-Y_1)^2-(x_QZ_1^2-X_1+2X_1)(x_QZ_1^2-X_1)^2 \\ &= (y_QZ_1^3-Y_1)^2-(x_QZ_1^2-X_1)^3+2X_1(x_QZ_1^2-X_1)^2. \end{aligned}$$

Since $(x_QZ_1^2-X_1)^3$ and $X_1(x_QZ_1^2-X_1)^2$ are needed for computing Y_3 and Z_3, we can save one multiplication here. Hence, we need three squarings (i.e Z_1^2, $(x_QZ_1^2-X_1)^2$ and $(y_QZ_1^3-Y_1)^2$) and eight multiplications (i.e $x_Q\cdot Z_1^2$, $Z_1^2\cdot Z_1$, $y_Q\cdot Z_1^3$, $X_1\cdot(x_QZ_1^2-X_1)^2$, $(x_QZ_1^2-X_1)\cdot(x_QZ_1^2-X_1)^2$, $(y_QZ_1^3-Y_1)\cdot[X_1(x_QZ_1^2-X_1)^2-X_3]$, $Y_1\cdot(x_QZ_1^2-X_1)^3$ and $Z_1\cdot(x_QZ_1^2-X_1)$) in $\mathbb{F}_{p^2}$, that is $3S_2+8M_2=30M$ for computing one point addition in G_2'.

Let $T=(x_1,y_1)=(x_T\omega^2,y_T\omega^3)\in G_2$, $Q=(x_2,y_2)=(x_Q\omega^2,y_Q\omega^3)\in G_2$, $P=(x_P,y_P)$ and let $T+Q=(x_3,y_3)$. Recall the formula for evaluating the line through T and Q at P in Section 2.3:

$$\ell_{T,Q}(P)=y_P-y_1-\frac{y_2-y_1}{x_2-x_1}(x_P-x_1).$$

Let (X,Y,Z) be the point we stored in jacobian coordinates such that $x_T=X/Z^2$ and $y_T=Y/Z^3$. Then

$$\begin{aligned} \ell_{Q,T}(P) &= y_P-y_2-\frac{y_2-y_1}{x_2-x_1}(x_P-x_2) \\ &= y_P-y_Q\omega^3-\frac{y_Q\omega^3-y_T\omega^3}{x_Q\omega^2-x_T\omega^2}\cdot(x_P-x_Q\omega^2) \\ &= (y_P-y_Q\omega^3)-\frac{y_Q\omega^3-\frac{Y}{Z^3}\omega^3}{x_Q\omega^2-\frac{X}{Z^2}\omega^2}\cdot(x_P-x_Q\omega^2) \\ &= (y_P-y_Q\omega^3)-\frac{(y_QZ^3-Y)\omega^3}{Z^3}\cdot\frac{Z^2}{(x_QZ^2-X)\omega^2}\cdot(x_P-x_Q\omega^2) \\ &= \frac{(y_P-y_Q\omega^3)Z(x_QZ^2-X)-(y_QZ^3-Y)\omega(x_P-x_Q\omega^2)}{Z(x_QZ^2-X)}. \end{aligned}$$

Since $Z(x_QZ^2-X)\in\mathbb{F}_{p^2}$, the denominator will equal 1 after the final exponentiation. Thus, we take

$$\begin{aligned} \ell_{Q,T}(P) &= (y_P-y_Q\omega^3)Z(x_QZ^2-X)-(y_QZ^3-Y)\omega(x_P-x_Q\omega^2) \\ &= y_PZ(x_QZ^2-X)-x_P(y_QZ^3-Y)\omega-(y_QZ(x_QZ^2-X)-x_Q(y_QZ^3-Y))\omega^3. \end{aligned}$$

Note that since Z^2, $x_Q \cdot Z^2$, $Z \cdot (x_Q Z^2 - X)$, $Z \cdot Z^2$ and $y_Q \cdot Z^3$ are also needed for computing $T+Q$, we can save one squaring and four multiplications in $\mathbb{F}_{p^2}$. Hence, to evaluate $\ell_{Q,T}(P)$ we need two more multiplications (i.e., $y_Q \cdot Z(x_Q Z^2 - X)$ and $x_Q \cdot (y_Q Z^3 - Y)$) in $\mathbb{F}_{p^2}$ and four more multiplications (i.e., two for $y_P \cdot Z(x_Q Z^2 - X)$ and two for $x_P \cdot (y_Q Z^3 - Y)$) in $\mathbb{F}_p$, that is $2M_2 + 4M \approx 10M$. If $T + Q$ is not computed at the same time, then the cost of evaluating $\ell_{Q,T}(P)$ is $S_2 + 6M_2 + 4M \approx 24M$.

Recall that both $\ell_{T,T}(P)$, $\ell_{Q,T}(P) \in \mathbb{F}_{p^{12}}$ have the form $a + b\omega + c\omega^3$ with a, b, $c \in \mathbb{F}_{p^2}$. Let ℓ be an element in $\mathbb{F}_{p^{12}}$ of the form $\ell = a + b\omega + c\omega^3$ with a, b, $c \in \mathbb{F}_{p^2}$ and let f be a general element in $\mathbb{F}_{p^{12}}$ of the form $f = f_0 + f_1\omega$ with f_0, $f_1 \in \mathbb{F}_{p^6}$. Then ℓ can be written as $\ell = a + (b + cv)\omega$ so that the product $\ell \cdot f$ can be expressed as

$$\begin{aligned} \ell \cdot f &= (a + (b + cv)\omega) \cdot (f_0 + f_1\omega) \\ &= (af_0 + (b + cv)f_1 v) + (af_1 + (b + cv)f_0)\omega \\ &= (af_0 + (b + cv)f_1 v) + [(a + b + cv)(f_0 + f_1) - af_0 - (b + cv)f_1]\omega. \end{aligned}$$

From Lemma 5.2.1 we can see that af_0 can be computed in three $\mathbb{F}_{p^2}$ multiplications. Let $f_0 = f_{0,0} + f_{0,1}v + f_{0,2}v^2$ with $f_{0,0}$, $f_{0,1}$, $f_{0,2} \in \mathbb{F}_{p^2}$. Then

$$\begin{aligned} (b + cv) \cdot f_0 &= (b + cv) \cdot (f_{0,0} + f_{0,1}v + f_{0,2}v^2) \\ &= (bf_{0,0} + cf_{0,2}\xi) + (cf_{0,0} + bf_{0,1})v \\ &\quad + [(b + c)(f_{0,0} + f_{0,1} + f_{0,2}) - bf_{0,0} - cf_{0,2} - cf_{0,0} - bf_{0,1}]v^2 \end{aligned}$$

gives that five multiplications (i.e., $b \cdot f_{0,0}$, $c \cdot f_{0,2}$, $c \cdot f_{0,0}$, $b \cdot f_{0,1}$ and $(b+c) \cdot (f_{0,0} + f_{0,1} + f_{0,2})$) in $\mathbb{F}_{p^2}$ are needed for computing $(b + cv) \cdot f_0$. Similarly, another five multiplications in $\mathbb{F}_{p^2}$ are needed for $(a + b + cv) \cdot (f_0 + f_1)$. Therefore, the multiplication between line functions $\ell_{T,T}(P)$, $\ell_{Q,T}(P) \in \mathbb{F}_{p^{12}}$ and $f \in \mathbb{F}_{p^{12}}$ requires only 13 multiplications in $\mathbb{F}_{p^2}$, that is $13M_2 \approx 39M$.

The cost of Steps 2(a) and 2(b) is $S_{12} \approx 36M$ for the squaring, $17M$ for point doubling, $15M$ for the line evaluation, and $39M$ for the multiplication between f^2 and $\ell_{T,T}P$. The cost of Step 2(c) is $30M$ for point addition, $10M$ for point the line evaluation, and $39M$ for the multiplication between f and $\ell_{T,Q}P$. Recall that, $|6u + 2|$ is a 65-bit integer of Hamming weight 5. Steps 2(a) and 2(b) are processed 64 times, and $b_i = 1$ is encountered four times in Step 2(c). Hence, the cost of Miller loop is

$$64 \cdot (36M + 17M + 15M + 39M) + 4 \cdot (30M + 10M + 39M) \approx 7164M.$$

The operation cost for the Miller loop is summarized in Table 5.2.

5.3.2 Operation Count for Adjustment Steps

As the BN parameter is chosen to be $u = -(2^{65} + 2^{55} + 1)$, the R-ate parameter $6u + 2$ is negative. In order to obtain the correct pairing value, we add Step 3 right after the Miller

Operations	Cost
Point Doubling in G_2' $(T+T)$	17M
Evaluating Line $\ell_{T,T}(P)$ while computing $(T+T)$	15M
Point Addition in G_2' $(T+Q)$	30M
Evaluating Line $\ell_{T,Q}(P)$ while computing $(T+Q)$	10M
Evaluating Line $\ell_{T,Q}(P)$	24M
Multiplication between f and ℓ	39M

Table 5.2: Cost estimates for the Miller loop

loop. Note that $T \in G_2'$ is stored in jacobian coordinates. Let $T = (X, Y, Z)$ with X, Y, $Z \in \mathbb{F}_{p^2}$. We can compute

$$-T = (X, -Y, Z)$$

with a cheap negation in $\mathbb{F}_{p^2}$. The inversion for computing $f^{-1} \in \mathbb{F}_{p^{12}}$ is expensive. However, in the final exponentiation step, the inversion of f is required. Therefore, we can store f for later use before we apply the expensive inversion. The cost of Step 3 is $I_{12} \approx 97M + I$.

To convert T in jacobian coordinates to T' in affine coordinates, we use the map

$$T \to T',$$

$$(X, Y, Z) \mapsto (XZ^{-2}, YZ^{-3}).$$

Thus, one inversion (i.e., Z^{-1}), one squaring (i.e., $(Z^{-1})^2$) and three multiplications (i.e., $Z^{-1} \cdot Z^{-2}$, $X \cdot Z^{-2}$ and $Y \cdot Z^{-3}$) in $\mathbb{F}_{p^2}$ are needed. The cost of Step 4(a) is $I_2 + S_2 + 3M_2 \approx 15M + I$. Similarly to Step 2(c), the cost of Step 4(b) is $79M$.

Let $A = \sum_{i=0}^{5} a_i \omega^i$ be an arbitrary element in $\mathbb{F}_{p^{12}}$ with $a_i \in \mathbb{F}_{p^2}$. Then, the Frobenius

map is given as

$$\begin{aligned} A^p &= \left(\sum_{i=0}^{5} a_i \omega^i\right)^p \\ &= \sum_{i=0}^{5} a_i^p \omega^{ip} \\ &= \sum_{i=0}^{5} a_i^p (\omega \cdot \omega^{p-1})^i \\ &= \sum_{i=0}^{5} a_i^p (\omega \cdot \xi^{\frac{p-1}{6}})^i \qquad \text{since } p \equiv 1 \pmod 6 \\ &= \sum_{i=0}^{5} (a_i^p \cdot \xi^{i\frac{p-1}{6}}) \omega^i. \end{aligned}$$

Note that if the $\xi^{i\frac{p-1}{6}} \in \mathbb{F}_{p^2}$ are precomputed, then the p-th powering in $\mathbb{F}_{p^{12}}$ can be computed at a cost of roughly 5 multiplications (i.e., $a_i^p \cdot \xi^{i\frac{p-1}{6}}$ for $i \in \{1, 2, 3, 4, 5\}$) in $\mathbb{F}_{p^2}$, since a_i^p can be computed at a negligible cost and $\xi^{i\frac{p-1}{6}} = 1$ when $i = 0$. Hence, Step 4(c) can be accomplished at a cost of $5M_2 \approx 15M$. Similarly, the p^e-th powering of A with $e \in \mathbb{N}$ can be computed in $15M$ by using

$$A^{p^e} = \sum_{i=0}^{5} (a_i^{p^e} \cdot \xi^{i\frac{p^e-1}{6}}) \omega^i.$$

For $T = (X, Y, Z)$ with X, Y, $Z \in \mathbb{F}_{p^2}$, the cost of applying the Frobenius map

$$\phi_p(T) = (X^p, Y^p, Z^p)$$

is almost free. According to Section 5.3.1, the line evaluation without point addition cost $24M$. Thus, together with multiplication, Step 4(d) is computed as a cost of roughly $24M + 39M \approx 63M$. Finally, the multiplication between f and f' in Step 4(e) costs a full $\mathbb{F}_{p^{12}}$ multiplication $M_{12} \approx 54M$. Therefore, the total cost of the adjustment steps is $323M + 2I$.

5.3.3 Operation Count for Final Exponentiation

The algorithm we use for the final exponentiation is from [13]. The main idea of this algorithm is to use the fact that exponentiations to powers of p are efficiently computed.

We first factor $(p^{12}-1)/n$ into three parts,

$$\frac{p^{12}-1}{n} = (p^6-1)\cdot(p^2+1)\cdot\frac{p^4-p^2+1}{n}.$$

Note that the first two exponentiations are easy to compute. In Step 5(a), the cost of powering f^{p^6} is negligible since $f \in \mathbb{F}_{p^{12}}$. Since f^{-1} is pre-stored before the adjustment steps, the cost of Step 5(a) is 1 multiplication in $\mathbb{F}_{p^{12}}$, $M_{12} \approx 54M$. As analyzed in Step 4(c), the powering f^{p^2} cost $5M_2 \approx 15M$. We only need one more multiplication in $\mathbb{F}_{p^{12}}$ to compute $f^{p^2+1} = f \cdot f^{p^2}$. Hence, the cost of Step 5(b) is $5M_2 + M_{12} \approx 69M$.

Now, the only problem left is to raise $f \in \mathbb{F}_{p^{12}}$ to the power $(p^4-p^2+1)/n$, which is call the "hard exponentiation". Recall that, for BN curves we have $p = 36u^4+36u^3+24u^2+6u+1$ and $n = 36u^4+36u^3+18u^2+6u+1$. If we substitute these two polynomials into $(p^4-p^2+1)/n$, we will get a high-degree polynomial in terms of u. Then we write this polynomial in base p as

$$p^3 + (6u^2+1)p^2 + (-36u^3-18u^2-12u+1)p + (-36u^3-30u^2-18u-2).$$

Then

$$\begin{aligned}
f^{\frac{p^4-p^2+1}{n}} &= f^{p^3}\cdot f^{(6u^2+1)p^2}\cdot f^{(-36u^3-18u^2-12u+1)p}\cdot f^{-36u^3-30u^2-18u-2} \\
&= f^{p^3}\cdot (f^{p^2})^{6u^2+1}\cdot (f^p)^{-36u^3-18u^2-12u+1} + f^{-36u^3-30u^2-18u-2} \\
&= f^{p^3}\cdot (f^{p^2})^{6u^2+1}\cdot (f^p)^{(-36u^3-30u^2-6u-5)+(12u^2+2)+(-6u-5)+9} \\
&\quad \cdot f^{(-36u^3-30u^2-6u-5)+(-6u-5)+(-6u-5)+9+4} \\
&= f^{p^3}\cdot (f^{p^2})^{6u^2+1}\cdot (f^p)^{(-6u-5)(6u^2+1)+2(6u^2+1)+(-6u-5)+9} \\
&\quad \cdot f^{(-6u-5)(6u^2+1)+(-6u-5)+(-6u-5)+9+4} \\
&= f^{p^3}\cdot [f^{p^2}\cdot (f^p\cdot f)^{-6u-5}\cdot (f^p)^2]^{6u^2+1}\cdot (f^p\cdot f)^{-6u-5}\cdot f^{-6u-5}\cdot (f^p\cdot f)^9\cdot f^4
\end{aligned}$$

Exponentiations to powers of p can be efficiently computed using Frobenius. Other exponentiations in terms of u can be computed using the square and multiply method. In the curve we chose, $-6u-5$ is a 65-bit integer of Hamming weight 5 and $6u^2+1$ is a 127-bit integer of Hamming weight 13. Step 5(c) to Step 5(f) evaluate the "hard exponentiation."

Note that after we apply Step 5(a), f satisfies the condition that $f^{p^6+1} = 1$. Thus, the cost of squaring such f is $S' \approx 24M$. Step 5(c) costs $64S' + 4M_{12} \approx 1752M$. The cost of Step 5(d) is $5M_2 \approx 15M$ and the cost of Step 5(e) is $M_{12} \approx 54M$.

For Step 5(f), we need $15M_2 \approx 45M$ to compute f^p, f^{p^2} and f^{p^3}, $2M_{12}+S' \approx 132M$ to compute $b\cdot(f^p)^2\cdot f^{p^2}$, $126S' + 12M_{12} \approx 3672M$ for the $6u^2+1$-th powering, $3S' + M_{12} \approx 126M$ for the 9-th power, and $2S' \approx 48M$ to compute f^4. Finally, $6M_{12} \approx 324M$ extra cost is needed to multiply the terms together. Therefore, the total cost of the "hard exponentiation" is $6168M$ and the cost of the whole final exponentiation step is $6291M$.

The total cost of computing the R-ate pairing is

$$7164M + 323M + 2I + 6291M = 13778M + 2I \approx 13860M.$$

Chapter 6

Recent Work

In this chapter, we analyze some recent work [10] [11] [12] [19] [18] on speeding up the pairing computation. These speed-ups are applied to compute the R-ate pairing on the BN curve described in Chapter 5, and conclusions are drawn about the effectiveness of the speed-ups.

6.1 R-ate Pairings with Projective Coordinates

In 2010, Costello, Lange and Naehrig provided some improved formulas for using projective coordinates instead of jacobian coordinates [12]. A point (X, Y, Z) in projective coordinates corresponds to the point (x, y) in affine coordinates with $x = X/Z$ and $y = X/Z$. In these coordinates, the equation of the BN curve is

$$\left(\frac{Y}{Z}\right)^2 = \left(\frac{X}{Z}\right)^3 + b,$$

or

$$X^3 = Z(Y^2 - bZ^2).$$

Recall the point doubling formula introduced in Section 2.1.1. Let $P = (x, y)$ with $P \neq -P$. Then $2P = (x_3, y_3)$, where

$$x_3 = \left(\frac{3x^2}{2y}\right)^2 - 2x \tag{6.1}$$

$$y_3 = \left(\frac{3x^2}{2y}\right)(x - x_3) - y. \tag{6.2}$$

In projective coordinates we derive the formula for doubling a point $T = (X, Y, Z)$ to $2T = (X_3, Y_3, Z_3)$. Substituting $x = X/Z$, $y = Y/Z$, $x_3 = X_3/Z_3$ and $y_3 = Y_3/Z_3$ into (6.1) gives

$$\begin{aligned}
\frac{X_3}{Z_3} &= \left(\frac{3(\frac{X}{Z})^2}{2(\frac{Y}{Z})}\right)^2 - 2\left(\frac{X}{Z}\right) \\
&= \frac{9X^4}{Z^4} \cdot \frac{Z^2}{4Y^2} - \frac{2X}{Z} \\
&= \frac{9X^4}{4Y^2Z^2} - \frac{8XY^2Z}{4Y^2Z^2} \\
&= \frac{9XZ(Y^2 - bz^2) - 8XY^2Z}{4Y^2Z^2} \\
&= \frac{X(Y^2 - 9bz^2)}{4Y^2Z}.
\end{aligned}$$

Similarly, substituting $x = X/Z$, $y = Y/Z$, $x_3 = X_3/Z_3$ and $y_3 = Y_3/Z_3$ into (6.2) gives

$$\begin{aligned}
\frac{Y_3}{Z_3} &= \left(\frac{3(\frac{X}{Z})^2}{2(\frac{Y}{Z})}\right) \cdot \left(\frac{X}{Z} - \frac{X_3}{Z_3}\right) - \frac{Y}{Z} \\
&= \left(\frac{3X^2}{Z^2} \cdot \frac{Z}{2Y}\right) \cdot \left(\frac{X}{Z} - \frac{X(Y^2 - 9bz^2)}{4Y^2Z}\right) - \frac{Y}{Z^3} \\
&= \frac{3X^2}{2YZ} \cdot \frac{X(4Y^2 - Y^2 + 9bZ^2)}{4Y^2Z} - \frac{Y}{Z} \\
&= \frac{3X^3(3Y^2 + 9bZ^2) - 8Y^4}{8Y^3Z^2} - \frac{Y}{Z} \\
&= \frac{3Z(Y^2 - bZ^2)(3Y^2 + 9bZ^2) - 8Y^4}{8Y^3Z^2} - \frac{Y}{Z} \\
&= \frac{9Y^4 + 27bY^2Z^2 - 9bY^2Z^2 - 27b^2Z^2}{8Y^3Z} - \frac{8Y^4}{8Y^3Z} \\
&= \frac{Y^4 + 18bY^2Z^2 - 27b^2Z^2}{8Y^3Z}.
\end{aligned}$$

Hence we have $2T = (X_3, Y_3, Z_3)$ where

$$\begin{cases}
X_3 = 4XY(Y^2 - 9bz^2) \\
Y_3 = Y^4 + 18bY^2Z^2 - 27b^2Z^2 \\
Z_3 = 8Y^3Z.
\end{cases}$$

Note that since we are doing the curve arithmetic on the twisted curve $E'/\mathbb{F}_{p^2}$, one point doubling requires seven squarings (i.e., X^2, Y^2, Z^2, $2XY = (X+Y)^2 - X^2 - Y^2$, $2YZ =$

$(Y+Z)^2-Y^2-Z^2$, $(YZ)^2$ and $Y^4=(Y^2)^2$) and two multiplications (i.e., $2XY\cdot(Y^2-9bZ^2)$ and $(8YZ)\cdot(Y^2)$) in $\mathbb{F}_{p^2}$. Thus, a point doubling in G_2' costs $7S_2+2M_2=20M$. Recall the line function from Section 2.3.1: for $P=(x_P,y_P)\in G_1$, $T=(x_1,y_1)\in G_2$, the formula of the tangent line through Q evaluated at P is

$$\ell_{T,T}(P)=y_P-y_1-\frac{3x_1^2}{2y_1}(x_P-x_1).$$

Let $T=(x_1,y_1)=(x_T\omega^2,y_T\omega^3)$, and let (X,Y,Z) be the point stored in projective coordinates such that $x_T=X/Z$ and $y_T=Y/Z$. Then

$$\begin{aligned}
\ell_{T,T}(P) &= y_P-y_1-\frac{3x_1^2}{2y_1}(x_P-x_1)\\
&= y_P-y_T\omega^3-\frac{3(x_T\omega^2)^2}{2y_T\omega^3}(x_P-x_T\omega^2)\\
&= y_P-\frac{Y}{Z}\omega^3-\frac{3(\frac{X}{Z}\omega^2)^2}{2\frac{Y}{Z}\omega^3}(x_P-\frac{X}{Z}\omega^2)\\
&= \frac{y_PZ-Y\omega^3}{Z}-\frac{3X^2\omega^4}{Z^2}\cdot\frac{Z}{2Y\omega^3}\cdot\frac{x_PZ-X\omega^2}{Z}\\
&= \frac{2y_PYZ^2-2Y^2Z\omega^3-3X^2\omega(x_PZ-X\omega^2)}{2YZ^2}\\
&= \frac{Z(2y_PYZ-2Y^2\omega^3-3x_PX^2\omega)+3\omega^3Z(Y^2-bZ^2)}{2YZ^2}.
\end{aligned}$$

Since $2YZ\in\mathbb{F}_{p^2}$, the final exponentiation will convert it to 1. Thus, we take

$$\ell_{T,T}(P)=2y_PYZ-3x_PX^2\omega+(Y^2-b3Z^2)\omega^3.$$

Note that X^2, Y^2, Z^2 and YZ are already computed when doubling T. Therefore, to compute $\ell_{T,T}(P)$ we need only four multiplications (i.e., two for $y_P\cdot 2YZ$ and two for $x_P\cdot 3X^2$) in $\mathbb{F}_p$. The doubling step thus requires $2M_2+7S_2+4M\approx 24M$ using projective coordinates. Although more additions and subtractions are used in these formulas, they are more efficient compared to $32M$ using jacobian coordinates.

Let $T=(x_1,y_1)\in G_2$ and $Q=(x_2,y_2)\in G_2$ with $T\neq\pm Q$. The point addition formula in Section 2.1.1 is given as $T+Q=(x_3,y_3)$ where

$$x_3 = \left(\frac{y_2-y_1}{x_2-x_1}\right)^2-x_1-x_2 \tag{6.3}$$

$$y_3 = \left(\frac{y_2-y_1}{x_2-x_1}\right)(x_1-x_3)-y_1. \tag{6.4}$$

Let $T = (X, Y, Z) \in G_2'$ be the point stored in projective coordinates and $Q = (x_Q\omega^2, y_Q\omega^3) \in G_2$ in affine coordinates. Substituting $x = X/Z$, $y = Y/Z$, $x_3 = X_3/Z_3$ and $y_3 = Y_3/Z_3$ into (6.3) gives

$$\begin{aligned}\frac{X_3}{Z_3} &= \left(\frac{y_Q - \frac{Y}{Z}}{x_Q - \frac{X}{Z}}\right)^2 - \frac{X}{Z} - x_Q \\ &= \left(\frac{y_Q Z - Y}{Z} \cdot \frac{Z}{x_Q Z - X}\right)^2 - \frac{X + x_Q Z}{Z} \\ &= \frac{Z(y_Q Z - Y)^2 - (X + x_Q Z)(x_Q Z - X)^2}{Z(x_Q Z - X)^2}.\end{aligned}$$

Similarly, substituting $x = X/Z$, $y = Y/Z$, $x_3 = X_3/Z_3$ and $y_3 = Y_3/Z_3$ into (6.4) gives

$$\begin{aligned}\frac{X_3}{Z_3} &= \left(\frac{y_Q - \frac{Y}{Z}}{x_Q - \frac{X}{Z}}\right) \cdot \left(x_Q - \frac{X_3}{Z_3}\right) - y_Q \\ &= \left(\frac{y_Q Z - Y}{x_Q Z - X}\right) \cdot \left(\frac{x_Q Z(x_Q Z - X)^2 - Z(y_Q Z - Y)^2 + (X + x_Q Z)(x_Q Z - X)^2}{Z(x_Q Z - X)^2}\right) \\ &\quad - y_Q \\ &= \frac{(y_Q Z - Y)[x_Q Z(x_Q Z - X)^2 - Z(y_Q Z - Y)^2 + (X + x_Q Z)(x_Q Z - X)^2]}{Z(x_Q Z - X)^3} \\ &\quad - \frac{y_Q Z(x_Q Z - X)^3}{Z(x_Q Z - X)^3}.\end{aligned}$$

Hence we have $T + Q = (X_3, Y_3, Z_3)$ with

$$\begin{cases} X_3 = (x_Q Z - X)[Z(y_Q Z - Y)^2 - (X + x_Q Z)(x_Q Z - X)^2] \\ Y_3 = (y_Q Z - Y)[x_Q Z(x_Q Z - X)^2 - Z(y_Q Z - Y)^2 + (X + x_Q Z)(x_Q Z - X)^2] \\ \qquad -y_Q Z(x_Q Z - X)^3 \\ Z_3 = Z(x_Q Z - X)^3. \end{cases}$$

Hence, we need ten multiplications (i.e., $x_Q \cdot Z$, $y_Q \cdot Z$, $(X + x_Q Z) \cdot (x_Q Z - X)^2$, $Z \cdot (y_Q Z - Y)$, $X_3 = (x_Q Z - X) \cdot [Z(y_Q Z - Y)^2 - (X + x_Q Z)(x_Q Z - X)^2]$, $(x_Q Z - X)^3 = (x_Q Z - X) \cdot (x_Q Z - X)^2$, $Z_3 = Z \cdot (x_Q Z - X)^3$, $y_Q Z \cdot (x_Q Z - X)^3$, $x_Q Z \cdot (x_Q Z - X)^3$ and $(y_Q Z - Y) \cdot [x_Q Z(x_Q Z - X)^2 - Z(y_Q Z - Y)^2 + (X + x_Q Z)(x_Q Z - X)^2]$) and two squarings (i.e., $(x_Q Z - X)^2$ and $(y_Q Z - Y)^2$) in $\mathbb{F}_{p^2}$, that is $10M_2 + 2S_2 \approx 34M$ for computing one point addition in G_2'.

Let $T = (x_1, y_1) = (x_T\omega^2, y_T\omega^3) \in G_2$, $Q = (x_2, y_2) = (x_Q\omega^2, y_Q\omega^3) \in G_2$, $P = (x_P, y_P)$ and let $T + Q = (x_3, y_3)$. Recall the formula for evaluating the line through T and Q at P in Section 2.3:

$$\ell_{T,Q}(P) = y_P - y_2 - \frac{y_2 - y_1}{x_2 - x_1}(x_P - x_2).$$

Let (X, Y, Z) be the point stored in projective coordinates such that $x_T = X/Z$ and $y_T = Y/Z$. Then

$$\begin{aligned}
\ell_{Q,T}(P) &= y_P - y_2 - \frac{y_2 - y_1}{x_2 - x_1}(x_P - x_2) \\
&= y_P - y_Q\omega^3 - \frac{y_Q\omega^3 - y_T\omega^3}{x_Q\omega^2 - x_T\omega^2} \cdot (x_P - x_Q\omega^2) \\
&= (y_P - y_Q\omega^3) - \frac{y_Q\omega^3 - \frac{Y}{Z}\omega^3}{x_Q\omega^2 - \frac{X}{Z}\omega^2} \cdot (x_P - x_Q\omega^2) \\
&= (y_P - y_Q\omega^3) - \frac{(y_QZ - Y)\omega^3}{Z} \cdot \frac{Z}{(x_QZ - X)\omega^2} \cdot (x_P - x_Q\omega^2) \\
&= \frac{(y_P - y_Q\omega^3)(x_QZ - X) - (y_QZ - Y)\omega(x_P - x_Q\omega^2)}{x_QZ - X}.
\end{aligned}$$

Since $(x_QZ - X) \in \mathbb{F}_{p^2}$, the denominator will equal to 1 after the final exponentiation. Thus, we take

$$\begin{aligned}
\ell_{Q,T}(P) &= (y_P - y_Q\omega^3)(x_QZ - X) - (y_QZ - Y)\omega(x_P - x_Q\omega^2) \\
&= y_P(x_QZ - X) - x_P(y_QZ - Y)\omega - (x_QY - y_QX)\omega^3.
\end{aligned}$$

Note that x_QZ and y_QZ are already computed when computing $T+Q$. Hence, we need two more multiplications (i.e., $y_Q \cdot X$ and $x_Q \cdot Y$) in $\mathbb{F}_{p^2}$ and four more multiplications (i.e., two for $y_P \cdot (x_QZ - X)$ and two for $x_P \cdot (y_QZ - Y)$) in $\mathbb{F}_p$, that is $2M_2 + 4M \approx 10M$. If $Q+T$ is not computed at the same time, then the cost of evaluating $\ell_{Q,T}(P)$ is $4M_2 + 4M \approx 16M$. Both $\ell_{T,T}(P)$, $\ell_{Q,T}(P) \in \mathbb{F}_{p^{12}}$ have the form $a + b\omega + c\omega^3$ with a, b, $c \in \mathbb{F}_{p^2}$. Therefore, as discussed in Section 5.3.1, the multiplication between line functions $\ell_{T,T}(P)$, $\ell_{Q,T}(P) \in \mathbb{F}_{p^{12}}$ and $f \in \mathbb{F}_{p^{12}}$ requires only 13 multiplications in $\mathbb{F}_{p^2}$, that is $13M_2 \approx 39M$.

We compare the multiplication counts in Table 6.1. We can see that the doubling step is more efficient while the addition step is less efficient using projective coordinates. Based on the operation count, projective coordinates is more efficient than jacobian coordinates even when each doubling step is followed by an addition step. However, the loop parameter is always chosen to have a low Hamming weight so that only a few addition steps are required in the whole process.

Using the BN curve introduced in Chapter 5, the cost of the Miller loop is

$$64 \cdot (20M + 4M + 36M + 39M) + 4 \cdot (34M + 10M + 39M) \approx 6668M.$$

There is one point addition and two line evaluations (one followed by the point addition and another computed alone) in the adjustment steps. Hence, $4M$ is saved so that the

Operation	Jacobian Coordinates	Projective Coordinates
Point Doubling in G_2' $(T+T)$	17M	20M
Evaluating Line $\ell_{T,T}(P)$ (while computing $(T+T)$)	15M	4M
Point Addition in G_2' $(T+Q)$	30M	34M
Evaluating Line $\ell_{T,Q}(P)$ (while computing $(T+Q)$)	10M	10M
Evaluating Line $\ell_{T,Q}(P)$	24M	16M
Multiplication between f and ℓ	39M	39M
Doubling Step in Miller's Algorithm	32M + 36M + 39M	24M + 36M + 39M
Addition Step in Miller's Algorithm	40M + 39M	44M + 39M

Table 6.1: Cost comparison: Jacobian coordinates vs. projective coordinates

adjustment steps cost $319M + 2I$. The cost of computing one pairing is decreased from $13860M$ to

$$6668M + 319M + 2I + 6291M \approx 13360M.$$

Therefore, the pairing computation is roughly 3.74% faster using projective coordinates.

Costello et al. compared the optimal ate pairings and the twisted ate pairings defined in [12]. The results are summarized in Table 4 of that paper. We focus on the row with $k = 12$, since BN curves have embedding degree 12. For optimal ate pairings, the first pairing argument is in G_2 and the second pairing argument is in G_1. Conversely, for twisted ate pairings, the first pairing argument is in G_1 and the second pairing argument is in G_2. We list the operation counts for the twisted ate pairing in Table 6.2.

Operations	Twisted ate Pairing
Point Doubling in G_1 $(T+T)$	7S + 2M
Evaluating Line $\ell_{T,T}(P)$ while computing $(T+T)$	4M
Point Addition in G_2' $(T+Q)$	10M + 2S
Evaluating Line $\ell_{T,Q}(P)$ while computing $(T+Q)$	2M + 4M
Evaluating Line $\ell_{T,Q}(P)$	4M + 4M
Multiplication between f and ℓ	39M
Doubling Step in Miller's Algorithm	13M + 36M + 39M
Addition Step in Miller's Algorithm	18M + 39M

Table 6.2: Operation counts for the twisted ate pairing

In Costello et al.'s comparison table [12], the column "Tate : ate $(s = m)$" gives

the base field multiplication count of one doubling step for both twisted ate pairings and optimal ate pairings. Costello et al. [12] used the Toom-Cook method for computing the multiplication $f^2 \cdot \ell$ using $45M$. Furthermore, he assumed $s = m$ in all extension fields. Hence, one also needs $45M$ to compute the squaring f^2. Therefore, the cost of computing one doubling step (point doubling $T + T$ and line $\ell_{T,T}(P)$ evaluation, full field squaring f^2 and full field multiplication $f^2 \cdot \ell$) in the twisted ate pairing is

$$13M + 45M + 45M = 103M.$$

In contrast, the cost of computing one doubling step in the optimal ate pairing is

$$31M + 45M + 45M = 121M.$$

The cost of point doubling and line evaluation is estimated in [12] to be $31M$ instead of $24M$, because the cost of seven squarings in $\mathbb{F}_{p^2}$ is assumed to be as much as the cost of seven multiplications in $\mathbb{F}_{p^2}$. Costello et al.'s cost estimates should be updated as follows:

1. The Toom-Cook method requires fewer multiplications than Karatsuba's method, but is slower in practice [13]. Hence, it is better to use Karatsuba's method.
2. The multiplications between the pairing value f and the line functions can exploit the sparseness of the line function values, and thus can be computed more efficiently than general multiplications in $\mathbb{F}_{p^{12}}$.
3. Assuming $s = m$ in all extension fields is not quite accurate. It is better to only assume that $s = m$ in the base field.

With these improved counts, the estimated cost of one doubling step in the twisted ate pairing is

$$13M + 36M + 39M = 88M,$$

whereas the estimated cost of one doubling step in the optimal ate pairing is

$$24M + 36M + 39M = 99M.$$

Therefore, the value in column "Tate : ate $(s = m)$" should be $88 : 99$ instead of $103 : 121$.

The column "$m_{opt} : T_e : r$" in Costello et al.'s comparison table represents the loop length ratios for optimal ate pairings, twisted ate pairings, and Tate pairing, respectively. For BN curves, we have

1. $m_{opt} = \log_2 \mid 6u - 2 \mid$,
2. $T_e = (t-1)^2 = \log_2 \mid 36u^3 + 18u^2 + 6u + 1 \mid$,

3. $r = \log_2 \mid 36u^4 + 36u^3 + 18u^2 + 6u + 1 \mid$.

Powers of $(t-1)^2 \bmod r$ do not give shorter loop length so that the ratio $m_{opt} : T_e : r$ for BN curves is approximately $1 : 3 : 4$. This is different from the entry "$1 : 2 : 4$" in [12].

The last column "$a_{m_{opt}}$ vs. η_{T_e}" represents a factor of how many times faster the computation of the Miller loop is for the optimal ate pairing than for the twisted ate pairing. For the BN curve (5.1), T_e is a 192-bit integer of Hamming weight 36. One Miller loop for twisted ate pairing costs

$$191 \cdot (13M + 36M + 39M) + 35 \cdot (18M + 39M) = 18803M.$$

Hence, the last column should be $a_{m_{opt}}(2.8)$, not $a_{m_{opt}}(1.7)$.

6.2 R-ate Pairings with Affine Coordinates

Inversion costs much more than other operations in the same field. Naturally, we choose projective coordinates or jacobian coordinates to avoid inversions in pairing computations. In 2010, Lauter, Montgomery and Naehrig analyzed and implemented the optimal ate pairing using affine coordinates [19]. Recall the point doubling formula introduced in Section 2.1.1. Let $T = (x, y) \in G_2'$ with $T \neq -T$. Then $2T = (x_3, y_3)$, where

$$\begin{cases} \lambda = \dfrac{3x^2}{2y} \\ x_3 = \lambda^2 - 2x \\ y_3 = \lambda(x - x_3) - y. \end{cases}$$

Since in optimal ate pairings we are doing the curve arithmetic on the twisted curve $E'/\mathbb{F}_{p^2}$, we have x, y, x_3 and $y_3 \in \mathbb{F}_{p^2}$. We need one inversion (i.e., $(2y)^{-1}$), two squarings (i.e., x^2 and λ^2) and two multiplications (i.e., $3x^2 \cdot (2y)^{-1}$ and $\lambda \cdot (x - x_3)$) in $\mathbb{F}_{p^2}$. Thus, point doubling in G_2' cost $2S_2 + 2M_2 + I_2 \approx I_2 + 10M$ using affine coordinates. Recall from Section 2.3.1 that for $P = (x_P, y_P) \in G_1$ and $T = (x\omega^2, y\omega^3) \in G_2$, the formula for the tangent line through T evaluated at P is

$$\begin{aligned} \ell_{T,T}(P) &= y_P - y\omega^3 - \frac{3(x\omega^2)^2}{2y\omega^3}(x_P x\omega^2) \\ &= y_P - y\omega^3 - \omega\lambda(x_P x\omega^2) \\ &= y_P - \lambda x_P \omega + (\lambda x - y)\omega^3. \end{aligned}$$

Then, to compute $\ell_{T,T}(P)$ we need two multiplications (i.e., $x_P \cdot \lambda$) in $\mathbb{F}_p$ and one multiplication (i.e., $\lambda \cdot x$) in $\mathbb{F}_{p^2}$. Hence, the doubling step requires $3M_2 + 2S_2 + I_2 + 2M \approx 15M + I_2$ using affine coordinates.

Let $T = (x, y) \in G_2'$ and $Q = (x_Q, y_Q) \in G_2'$ with $T \neq \pm Q$. The point addition formula in Section 2.1.1 is given as $T + Q = (x_3, y_3)$ where

$$\begin{cases} \lambda = \dfrac{y - y_Q}{x - x_Q} \\ x_3 = \lambda^2 - x - x_Q \\ y_3 = \lambda(x - x_3) - y. \end{cases}$$

Clearly, we need one inversion (i.e., $(x-x_Q)^{-1}$), two multiplications (i.e., $(y-y_Q)\cdot(x-x_Q)^{-1}$ and $\lambda\cdot(x-x_3)$), and one squaring (i.e., λ^2) in $\mathbb{F}_{p^2}$, for a total cost of $2M_2+S_2+I_2 \approx 8M+I_2$ for computing one point addition in G_2'. Let $T = (x\omega^2, y\omega^3) \in G_2$, $Q = (x_Q\omega^2, y_Q\omega^3) \in G_2$, $P = (x_P, y_P) \in G_1$ and let $T + Q = (x_3, y_3)$. Recall the formula for evaluating the line through T and Q at P in Section 2.3:

$$\begin{aligned} \ell_{T,Q}(P) &= y_P - y_Q\omega^3 - \frac{y_Q\omega^3 - y\omega^3}{x_Q\omega^2 - x\omega^2}(x_P - x_Q\omega^2) \\ &= y_P - y_Q\omega^3 - \lambda\omega(x_P - x_Q\omega^2) \\ &= y_P - \lambda x_P\omega + (\lambda x_Q - y_Q)\omega^3. \end{aligned}$$

Then, we need one multiplication (i.e., $\lambda \cdot x_Q$) in $\mathbb{F}_{p^2}$ and two multiplications (i.e., $\lambda \cdot x_Q$) in $\mathbb{F}_p$, that is $M_2+2M \approx 5M$ for compute $\ell_{T,Q}(P)$. If $Q+T$ is not computed at the same time, we need to compute λ. Then the cost of evaluating $\ell_{Q,T}(P)$ is $2M_2 + 2M + I_2 \approx 8M + I_2$. Both $\ell_{T,T}(P)$ and $\ell_{Q,T}(P) \in \mathbb{F}_{p^{12}}$ have the form $a+b\omega+c\omega^3$ with a, b, $c \in \mathbb{F}_{p^2}$. Therefore, as discussed in Section 5.3.1, the multiplication between line functions $\ell_{T,T}(P)$, $\ell_{Q,T}(P) \in \mathbb{F}_{p^{12}}$ and $f \in \mathbb{F}_{p^{12}}$ requires $39M$. We list the operation counts in Table 6.3 and compare with those using projective coordinates.

Operation	Affine Coordinates	Projective Coordinates
Point Doubling in G_2' $(T + T)$	$10M + I_2$	20M
Evaluating Line $\ell_{T,T}(P)$		
(while computing $(T + T)$)	5M	4M
Point Addition in G_2' $(T + Q)$	$8M + I_2$	34M
Evaluating Line $\ell_{T,Q}(P)$		
(while computing $(T + Q)$)	5M	10M
Evaluating Line $\ell_{T,Q}(P)$	$8M + I_2$	16M
Multiplication between f and ℓ	39M	39M
Doubling Step in Miller's Algorithm	$36M + 39M + 15M + I_2$	36M + 39M + 24M
Addition Step in Miller's Algorithm	$39M + 13M + I_2$	39M + 44M

Table 6.3: Cost comparison: Affine coordinates vs. projective coordinates

Our chosen loop parameter is a 65-bit integer with hamming weight 5. Then we need

$$\begin{aligned} & 64 \cdot (36M + 39M + 15M + I_2) + 4 \cdot (39M + 13M + I_2) \\ = \; & (5760M + 64I_2) + (208M + 4I_2) \end{aligned}$$

operations to compute the Miller loop when using affine coordinates.

Note that the efficiency of using affine coordinates depends on how efficient inversions in $\mathbb{F}_{p^2}$ can be computed. We argued in Section 5.2.1 that one inversion in $\mathbb{F}_{p^2}$ can be computed using two squarings, two multiplications and one inversion in $\mathbb{F}_p$, that is $I_2 \approx 2S + I + 2M \approx 4M + I$, to compute an inversion in $\mathbb{F}_{p^2}$. Hence, the cost of computing the Miller loop is

$$\begin{aligned} & (5760M + 64I_2) + (208M + 4I_2) \\ = \; & 5968M + 68(4M + I) \\ = \; & 6240M + 68I. \end{aligned}$$

Montgomery introduced a sharing-inversions trick in 1987 [22]. Using his idea, n inversions can be computed at the cost of one inversion and $3(n-1)$ multiplications in the same field. The algorithm is given as follows.

Algorithm 6.2.1 ***Algorithm to compute*** n ***inversions in a field:***

Input: n *elements,* $a_1, a_2, \ldots, a_n$*, in a field.*

Output: $a_1^{-1}, a_2^{-1}, \ldots, a_n^{-1}$.

1. $A_1 \leftarrow a_1$.
2. *For* $i = 2$ *to* n *do:* $A_i \leftarrow A_{i-1} \cdot a_i$.
3. $B_n \leftarrow A_n^{-1}$.
4. *For* $i = n - 1$ *to* 1 *do:* $B_i \leftarrow B_{i+1} \cdot a_{i+1}$.
5. $a_1^{-1} \leftarrow B_1$.
6. *For* $i = 2$ *to* n *do:* $a_i^{-1} \leftarrow B_i \cdot a_{i-1}$.
7. *Return* $a_1^{-1}, a_2^{-1}, \ldots, a_n^{-1}$.

According to the algorithm, we need one inversion for Step 3, $n-1$ multiplications for Step 2, $n-1$ multiplications for Step 4, and $n-1$ multiplications for Step 6, for a total of one inversion and $3(n-1)$ multiplications to compute n inversions. We can see that if n is large, then an inversion has effectively the same cost as three multiplications in the same field. However, the algorithm introduced at the beginning of Section 5.3 goes through the binary representation of the loop parameter from left to right. In this order, point doublings and point additions must be computed one after another. In order to use the inversion-sharing thick, Schroeppel and Beaver suggested to go through the binary representation from right to left [26]. Algorithm 6.2.2 is the updated right-to-left algorithm.

Algorithm 6.2.2 ***Right-to-left algorithm to compute the R-ate pairing on the BN curve (5.1):***

Input:

- $P = (x_P, y_P) \in G_1$ *with* x_P, $y_P \in \mathbb{F}_p$,
- $Q = (x_Q, y_Q) \in G_2'$ *with* x_Q, $y_Q \in \mathbb{F}_{p^2}$,
- $b = |6u+2| = \sum_{i=0}^{64} b_i 2^i$.

Output: $R_{A,B}(Q, P)$.

1. $T \leftarrow Q$, $f \leftarrow 1$, $j \leftarrow 0$.
2. *For* $i = 0$ *to 64 do:*
   (a) *if* $b_i = 1$, *then* $T'[j] \leftarrow T$, $f'[j] \leftarrow f$, $j \leftarrow j+1$.
   (b) $f \leftarrow f^2 \cdot \ell_{T,T}(P)$.
   (c) $T \leftarrow 2T$.
3. $T \leftarrow T'[0]$, $f \leftarrow f'[0]$.
4. *For* $j = 1$ *to 4 do:*
   (a) $f \leftarrow f \cdot f'[j] \cdot \ell_{T'[j],Q}(P)$.
   (b) $T \leftarrow T + T'[j]$.
5. $T \leftarrow -T$, $f \leftarrow f^{-1}$.
6. $f \leftarrow f \cdot (f \cdot \ell_{T,Q}(P))^p \cdot \ell_{\phi_p(T+Q),T}(P)$.
7. $f \leftarrow f^{(p^{12}-1)/n}$.

8. *Return* $R_{A,B}(Q,P) = f$.

This algorithm can only apply the inversion-sharing trick on the point addition steps. However, if multiple pairings are computed at the same time, we can complete all steps up to the inversions and then compute the inversions simultaneously. If a large number of pairings are computed at the same time, we can achieve $I \approx 3M$. We count the cost for the Miller loop (i.e., Steps 1 to 4 in the above algorithm) on our chosen curve using affine coordinates. We need $36M+39M+15M+I_2$ to compute Steps 2(b) and 2(c). Note that one more multiplication in $\mathbb{F}_{p^{12}}$ is required for each addition step. Then, $54M+39M+13M+I_2$ is required to compute Steps 4(a) and 4(b). Hence, we need

$$\begin{aligned} & 64 \cdot (36M + 39M + 15M + I_2) + 4 \cdot (54M + 39M + 13M + I_2) \\ = \; & (6016M + 64I) + (440M + 4I) \\ = \; & (6456M + 64I) + I + 3(4-1)M \\ = \; & 6465M + 65I \end{aligned}$$

to compute the Miller loop. We compare the cost of the Miller loop using affine coordinates and projective coordinates in Table 6.4.

Operations Counts for Miller loop	I	$I \approx 41M$	$I \approx 3M$
Affine Coordinates (left-to-right)	6240M + 68I	9028M	6444M
Affine Coordinates (right-to-left)	6465M + 65I	10885M	6660M
Projective Coordinates	6668M	6668M	6668M

Table 6.4: Cost of the Miller loop: Affine coordinates vs. projective coordinates

The pairing computation is even slower with the inversion-sharing trick, since there are two $\mathbb{F}_{p^{12}}$ multiplications in Step 6 of the right-to-left algorithm for each addition step. Therefore, we analyze the efficiency of using affine coordinates without applying the inversion-sharing trick. In Section 5.2, we assumed that

1. $I \approx 41M$;
2. $S \approx M$;
3. The cost of additions and subtractions are negligible;
4. The cost of multiplications by small field elements is negligible.

Using these assumptions, pairings using affine coordinates are much slower than pairings using projective coordinates. However, Lauter, Montgomery and Naehrig [19] used very different cost ratios:

1. $I \approx 13M$, thus $I_2 \approx 4M + I = 17M$;
2. $S \approx M$;
3. The cost of additions and subtractions are not negligible — 3 additions or subtractions in $\mathbb{F}_p$ cost roughly the same as 1 multiplication in $\mathbb{F}_p$. Consequently, 1 multiplication in $\mathbb{F}_{p^2}$ costs roughly 5 multiplications in $\mathbb{F}_p$, and 1 squaring in $\mathbb{F}_{p^2}$ costs roughly 3 multiplications in $\mathbb{F}_p$.

When we derived the point doubling formula using projective coordinates, two multiplications are counted as two squarings using the formula $xy = \frac{1}{2}[(x+y)^2 - x^2 - y^2]$ as x^2 and y^2 are precomputed. However, one squaring, one addition and two subtractions in $\mathbb{F}_{p^2}$ cost roughly as much as one multiplication in $\mathbb{F}_{p^2}$. Thus, we consider the two squarings to be multiplications and count the operations again in Table 6.5 using the ratios from [19].

Operations	Affine Coordinates	Projective Coordinates
Point Doubling in G_2' $(T+T)$	$2M_2 + 2S_2 + I_2 \approx 33M$	$4M_2 + 5S_2 \approx 35M$
Evaluating Line $\ell_{T,T}(P)$ (while computing $(T+T)$)	$M_2 + 2M \approx 7M$	4M
Point Addition in G_2' $(T+Q)$	$2M_2 + S_2 + I_2 \approx 30M$	$10M_2 + 2S_2 \approx 56M$
Evaluating Line $\ell_{T,Q}(P)$ (while computing $(T+Q)$)	$M_2 + 2M \approx 7M$	$2M_2 + 4M \approx 14M$
Multiplication between f and ℓ	39M	39M
Doubling Step	$36M + 39M + 40M$	$36M + 39M + 39M$
Addition Step	$39M + 37M$	$39M + 70M$
Whole Miller loop	$7664M$	$7732M$

Table 6.5: Cost comparison using nonstandard ratios from [19]

Note that there are more additions and subtractions in formulas using projective coordinates. Together with the multiplication counts in Table 6.5, Lauter et al.'s implementation achieves a faster pairing computation using affine coordinates even when a single pairing is computed. However, their multiplication-inversion ratio and multiplication-addition ratio are not considered to be standard (e.g. see [1], [13], [15], [27]). Affine coordinates are less efficient than projective coordinates in general according to the operation counts using more standard ratios.

6.3 Delaying Some Multiplications

In 2010, Costello, Boyd, Nieto and Wong introduced the idea of delaying some expensive multiplications in order to achieve speedups in pairing computations [10]. The idea can be briefly described as combining n consecutive doubling steps together. By "consecutive", we mean that there are no addition steps between the doubling steps. We begin by describing the method for $n = 2$.

First, we look at two consecutive doubling steps without applying the delaying-idea. We list the operations in order as follows.

1. Compute $T' = 2T$ and $\ell_{T,T}$.
2. Evaluate $\ell_{T,T}(P)$.
3. Compute $f' = f^2 \cdot \ell_{T,T}(P)$.
4. Compute $T'' = 2T'$ and $\ell_{T',T'}$
5. Evaluate $\ell_{T',T'}(P)$.
6. Compute $f'' = f'^2 \cdot \ell_{T',T'}(P)$.

Thus, two point doublings (i.e., $T' = 2T$ and $T'' = 2T'$) and two line evaluations (i.e., $\ell_{T,T}(P)$ and $\ell_{T',T'}(P)$) are computed in $\mathbb{F}_{p^2}$. Two squarings (i.e., f^2 and f'^2) and two multiplications by the line functions (i.e., $f^2 \cdot \ell_{T,T}(P)$ and $f'^2 \cdot \ell_{T',T'}(P)$) are computed in $\mathbb{F}_{p^{12}}$. As discussed in Section 6.1, the costs estimated by Costello et al. [10] can be improved as follows:

1. It is better to use Karatsuba's method for multiplication than the Toom-Cook method.
2. The multiplications between the pairing value f and the line functions can be computed at a cost of $39M$.
3. Assuming $s = m$ in all extension fields is not quite accurate. It is better to only assume that $s = m$ in the base field.

The estimated costs are listed in Table 6.6 for both the twisted ate pairing and the optimal ate pairing.

In [10], two consecutive doubling steps are combined as follows.

1. Compute $T' = 2T$ and $\ell_{T,T}$.

Operation	Twisted-ate pairing	Optimal-ate pairing
$T' = 2T$ and $\ell_{T,T}$	$2M + 7S$	$2M_2 + 7S_2$
Evaluate $\ell_{T,T}(P)$	$4M$	$4M$
$f' = f^2 \cdot \ell_{T,T}(P)$	$S_{12} + M_{12}$	$S_{12} + M_{12}$
$T'' = 2T'$ and $\ell_{T,T}$	$2M + 7S$	$2M_2 + 7S_2$
Evaluate $\ell_{T',T'}(P)$	$4M$	$4M$
$f'' = f'^2 \cdot \ell_{T',T'}(P)$	$S_{12} + M_{12}$	$S_{12} + M_{12}$
2 doubling steps	$206M$	$242M$
2 doubling steps (improved estimates)	$176M$	$198M$

Table 6.6: Cost of two doubling steps without the delaying idea

2. Compute $T'' = 2T'$ and $\ell_{T',T'}$.
3. Compute $\ell_{T,T}^2 \cdot \ell_{T',T'}$.
4. Evaluate $\ell_{T,T}^2 \cdot \ell_{T',T'}(P)$.
5. Compute $f'' = f^4 \cdot \ell_{T,T}^2 \cdot \ell_{T',T'}(P)$.

Thus, two point doubling (i.e., $T' = 2T$ and $T'' = 2T'$) and one line evaluation (i.e., $\ell_{T,T}^2 \cdot \ell_{T',T'}(P)$) are computed in $\mathbb{F}_{p^2}$. Two squarings (i.e., f^2 and $(f^2)^2$) and one multiplication by the line functions (i.e., $f^4 \cdot \ell_{T,T}^2 \cdot \ell_{T',T'}(P)$) are computed in $\mathbb{F}_{p^{12}}$. The hard part is the computation of the line function $\ell_{T,T}^2 \cdot \ell_{T',T'}$. Recall that $\ell_{T,T}(P)$ and $\ell_{T',T'}(P)$ have the form

$$\ell_{T,T} = ay_P + bx_p\omega + c\omega^3$$

and

$$\ell_{T',T'} = a'y_P + b'x_p\omega + c'\omega^3,$$

where a, b, c, a', b' and $c' \in \mathbb{F}_{p^2}$. Then, we need at most three squarings (i.e., a^2, b^2 and c^2) and another three squarings (i.e., $2x \cdot y = (x+y)^2 - x^2 - y^2$ where $x \in \{a, b, c\}$, $y \in \{a, b, c\}$ and $x \neq y$) to compute $\ell_{T,T}^2$. Thus, at most 18 multiplications (i.e., $x \cdot y$ where $x \in \{a', b', c'\}$ and $y \in \{a^2, b^2, c^2, ab, ac, bc\}$) are needed to compute $\ell_{T,T}^2 \cdot \ell_{T',T'}$. If y_P^2 is converted to $x_P^3 + b$, the form of $\ell_{T,T}^2$ is

$$\ell_{T,T}^2(P) = (A + Bx_P + Cx_P^2 + Dx_P^3) + (E + Fx_P)y_P$$

where A, B, C, D, E and F can be represented by $\mathbb{F}_{p^2}$ elements. Similarly, $\ell_{T,T}^2 \cdot \ell_{T',T'}$ has the form

$$\ell_{T,T}^2 \cdot \ell_{T',T'}(P) = (A' + B'x_P + C'x_P^2 + D'x_P^3 + E'x_P^4) + (F' + G'x_P + H'x_P^2 + I'x_P^3)y_P$$

where A', B', C', D', E', F', G', H' and I' can be represented by $\mathbb{F}_{p^2}$ elements. Thus, multiplications such as $B' \cdot x_P$ cost more than $\mathbb{F}_{p^2}$ multiplications. Suppose that x_P^2, x_P^3, x_P^4, $x_P y_P$, $x_P^2 y_P$ and $x_P^3 y_P$ are precomputed; we approximate the cost of line evaluation to be eight multiplications in $\mathbb{F}_{p^2}$. Note that $\ell_{T,T}^2 \cdot \ell_{T',T'}(P)$ is no longer of the form $a+b\omega+c\omega^3$ where a, b and $c \in \mathbb{F}_{p^2}$. The multiplication $f^4 \cdot \ell_{T,T}^2 \cdot \ell_{T',T'}(P)$ costs $54M$. The estimated costs are listed in Table 6.7 for both the twisted ate pairing and the optimal ate pairing.

Operations	Twisted-ate pairing	Optimal-ate pairing
$T' = 2T$ and $\ell_{T,T}$	$2M + 7S$	$2M_2 + 7S_2$
$T'' = 2T'$ and $\ell_{T',T'}$	$2M + 7S$	$2M_2 + 7S_2$
Compute $\ell_{T,T}^2 \cdot \ell_{T',T'}$	$18M + 6S$	$18M_2 + 6S_2$
Evaluate $\ell_{T,T}^2 \cdot \ell_{T',T'}(P)$	$8 \cdot 2M$	$8 \cdot 2M$
$f'' = f^4 \cdot \ell_{T,T}^2 \cdot \ell_{T',T'}(P)$	$2S_{12} + M_{12}$	$2S_{12} + M_{12}$
2 doubling steps	$193M$	$257M$
2 doubling steps (improved estimates)	$184M$	$248M$

Table 6.7: Cost of two doubling steps with the delaying idea

Comparing the results in Tables 6.6 and 6.7, we can see that although one multiplication in $\mathbb{F}_{p^{12}}$ is saved, the computation of the line function is more complicated. Moreover, the line function is not sparse, thus cannot be computed using the method introduced in Section 5.3.1. The delaying idea slows down the pairing computation in general. It is argued by Costello et al. [10] that combining more consecutive doubling steps together makes it more difficult to compute the super-line function. For curves with small embedding degree, it is optimal to combine the doubling steps in pairs. Costello et al. summarize their results using a table in Section 6 of [10]. We examine the row with embedding degree $k = 12$ and the column with $s = m$. They count the $\mathbb{F}_p$ multiplications required to compute one doubling step for the twisted ate pairing. According to the results in Tables 6.6 and 6.7, the value in column "$N = 0$" should be 88 instead of 103. The value in column "Optimal N" should be 92 rather than 96.5.

Costello, Boyd, Nieto and Wong [11] also specialized this delaying idea to some special curves. For BN curves with embedding degree $k = 12$ and curve equation $y^2 = x^3 + b$, the curve arithmetic is explicitly given.

Let $T = (X, Y, Z)$ be a point on the curve stored in projective coordinates. Then

$T' = 2T = (X_3, Y_3, Z_3)$ and $L = \ell^2_{T,T} \cdot \ell_{T',T'}(P)$ can be computed using the follow formulas.

$$\begin{cases} X_3 = 4XY(Y^2 - 9bz^2) \\ Y_3 = Y^4 + 18bY^2Z^2 - 27b^2Z^2 \\ Z_3 = 8Y^3Z \\ L = \alpha \cdot (L_0 + L_1 \cdot x_P + L_2 \cdot x_P^2 + L_3 \cdot y_P + L_4 \cdot x_P y_P) \\ \alpha = \dfrac{-Z^3[X(X^3 - 8bZ^3) - 4Z(X^3 + bZ^3)]^2}{64z^7Y^5 \cdot (27X^6 - 36X^3Y^2Z + 8Y^4Z^2)} \\ L_0 = 2X(Y^6 - 75bY^4Z^2 + 27b^2Y^2Z^4 - 81b^3Z^6) \\ L_1 = -4Z(5Y^6 - 75bZ^2Y^4 + 135Y^2b^2Z^4 - 81b^3Z^6) \\ L_2 = -6X^2Z(5Y^4 + 54bY^2Z^2 - 27b^2Z^4) \\ L_3 = 8XYZ(5Y^4 + 27b^2Z^4) \\ L_4 = 8YZ^2(Y^4 + 18bY^2Z^2 - 27b^2Z^4). \end{cases}$$

Note that α is in a proper subfield of $\mathbb{F}_{p^{12}}$, and so will be eliminated after the final exponentiation (Lemma 2.3.1). Thus, L can be computed as

$$L = L_0 + L_1 \cdot x_P + L_2 \cdot x_P^2 + L_3 \cdot y_P + L_4 \cdot x_P y_P.$$

The cost of computing $T' = 2T$ and $L = \ell^2_{T,T} \cdot \ell_{T',T'}$ is $11M_2 + 11S_2$ (or $11M + 11S$) for optimal ate pairings (or twisted ate pairings), and the cost of computing $L(P)$ is $4 \cdot 2M$. (The details are omitted. The interested reader can refer to Section 5.1 and Appendix 2 of [11].) Note that the point doubling $T'' = 2T'$ is computed without the evaluation of $\ell_{T',T'}$. We do not compute X_3^2 and thus have to compute $X_3 \cdot Y_3$ using one multiplication instead of one squaring. Hence, we need $3M_2 + 5S_2$ (or $3M + 5S$) to compute $T'' = 2T'$ for optimal ate pairings (or twisted ate pairings). The estimated costs are listed in Table 6.8 for both the twisted ate pairing and the optimal ate pairing using the faster formulas.

Operation	Twisted-ate pairing	Optimal-ate pairing
$T' = 2T$ and L	$11M + 11S$	$11M_2 + 11S_2$
$T'' = 2T'$	$3M + 5S$	$3M_2 + 5S_2$
Evaluate $L(P)$	$4 \cdot 2M$	$4 \cdot 2M$
$f'' = f^4 \cdot L(P)$	$2S_{12} + M_{12}$	$2S_{12} + M_{12}$
2 doubling steps	$173M$	$233M$
2 doubling steps (improved estimates)	$164M$	$208M$

Table 6.8: Cost of two doubling steps with the delaying idea using faster formulas

The last two rows of Tables 6.6, 6.7 and 6.8 are summarized in Table 6.9. For better comparison with Table 2 of [11], we provide the cost of computing six doubling steps in different scenarios.

	Twisted-ate Pairing	Optimal-ate Pairing
Standard	$618M$	$726M$
Pair-up the doubling steps	$579M$	$771M$
Pair-up with faster formulas	$519M$	$699M$
Standard (improved estimates)	$528M$	$594M$
Pair-up the doubling steps (improved estimates)	$552M$	$744M$
Pair-up with faster formulas (improved estimates)	$492M$	$624M$

Table 6.9: Cost of six doubling steps

We can see that the delaying-idea can improve the twisted ate pairings with the faster formulas. However, it does not speedup the optimal ate pairings with our more accurate cost estimation. Costello et al. [11] also provide formulas for combining three doubling steps. In that case, we need $498M$ (or $720M$) to compute six doubling steps for twisted ate pairings (or optimal ate pairings), which is slower. Table 2 of [11] summarize their results. We examine the row with embedding degree $k = 12$. In their table, $n = i$ means i doubling steps are combined for simultaneous computation. They count the $\mathbb{F}_p$ multiplications required to compute six doubling steps for both twisted ate pairings and optimal ate pairings. According to the results in Tables 6.6 and 6.9, the value in column "Pairings on $G_1 \times G_2$" should be 528 (or 492, 498 respectively) instead of 618 (or 519, 536 respectively) for $n = 1$ (or $n = 2$, $n = 3$ respectively). The value in column "Pairings on $G_2 \times G_1$" should be 594 (or 624, 720 respectively) instead of 726 (or 699, 824 respectively) for $n = 1$ (or $n = 2$, $n = 3$ respectively). The value in column "Best" on the right hand side of column "Pairings on $G_2 \times G_1$" should be $(n = 2, 7\%)$ instead of $(n = 2, 18\%)$. This means that, for twisted ate pairings, the doubling steps can be computed 7% more efficiently if two consecutive doubling steps are combined. The value in column "Best" on the right hand side of column "Pairings on $G_1 \times G_2$" should be $(n = 1, -)$ instead of $(n = 2, 5\%)$. This means that, for optimal ate pairings, the delaying idea cannot speed up the computation of doubling steps.

6.4 Final Exponentiation

Define

$$\Phi_k(x) = x^{2 \cdot 2^{a-1} 3^{b-1}} - x^{2^{a-1} 3^{b-1}} + 1$$

with $k = 2^a 3^b$ to be the k-th cyclotomic polynomial. Define

$$G_{\Phi_k(p)} = \{\alpha \in \mathbb{F}_{p^k} \mid \alpha^{\Phi_k(p)} = 1\}$$

to be the cyclotomic subgroup of $\mathbb{F}_{p^k}$. Recall that we can factor the exponent $(p^{12}-1)/n$ into three parts,

$$\frac{p^{12}-1}{n} = (p^6-1)\cdot(p^2+1)\cdot\frac{p^4-p^2+1}{n}.$$

Note that $p^4-p^2+1 = \Phi_{12}(p)$ is the 12th cyclotomic polynomial evaluated at p. Thus, we have

$$\frac{p^{12}-1}{n} = \frac{p^{12}-1}{\Phi_{12}(p)}\cdot\frac{\Phi_{12}(p)}{n}.$$

As discussed in Section 5.3.3, the first term $\alpha^{(p^{12}-1)/\Phi_{12}(p)}$ can be computed efficiently using the Frobienius map. For any $\alpha \in \mathbb{F}^*_{p^{12}}$, we have $\alpha^{(p^{12}-1)/\Phi_{12}(p)} \in G_{\Phi_{12}(p)}$.

Theorem 6.4.1 *Let $q = p^i \equiv 1 \pmod 6$ and $g \in G_{\Phi_6(q)} \subset \mathbb{F}^*_{q^6}$. Let e be an ℓ-bit exponent with binary representation $e = e_{\ell-1}e_{\ell-2}...e_2e_1e_0$. Let $H_e = \{i : 1 \le i \le \ell-1 \text{ and } e_i = 1\}$, and let $|H_e| = N$. Then, g^e can be computed at a cost dominated by*

$$\min\{4(\ell-1)M_i + (6N-3)M_i + NM_{6i} + 3NS_i + I_i, 6(\ell-1)M_i + NM_{6i}\}. \tag{6.5}$$

This theorem is a combination of Corollary 4.1 in [18] and results of Section 3 in [14]. We omit the proof here. The interested reader can refer to Sections 3 and 4 of [18] and Sections 2 and 3 of [14]. There are two formulas estimating the cost of the exponentiation. The first is faster when ℓ is large while the second is better for small exponents.

Recall the algorithm to compute the final exponentiation in Section 5.3. We do the following to compute $f^{(p^{12}-1)/n}$:

1. $f \leftarrow f^{p^6-1}$.
2. $f \leftarrow f^{p^2+1}$.
3. $a \leftarrow f^{-6u-5}$.
4. $b \leftarrow a^p$.
5. $b \leftarrow a \cdot b$.
6. $f \leftarrow f^{p^3} \cdot [b \cdot (f^p)^2 \cdot f^{p^2}]^{6u^2+1} \cdot b \cdot (f^p \cdot f)^9 \cdot a \cdot f^4$.

The first two exponentiations, Steps 1 and 2, are computed at a cost of $123M$. Exponentiations to powers of p can be efficiently computed using Frobenius. Other exponentiations in terms of u can be computed using the square and multiply method. In our chosen

curve, $-6u-5$ is a 65-bit integer of Hamming weight 5 and $6u^2+1$ is a 127-bit integer of Hamming weight 13. By Theorem 6.4.1, Step 3 can be computed using

$$4 \cdot 64M_2 + (24-3)M_2 + 4M_{12} + 12S_2 + I_2 \approx 1075M + I.$$

Similarly, we need

$$4 \cdot 126M_2 + (72-3)M_2 + 12M_{12} + 36S_2 + I_2 \approx 2443M + I$$

for the $(6u^2+1)$-th powering in Step 6. Step 4 costs 15M. Step 5 costs 54M. We need $45M$ to compute f^p, f^{p^2} and f^{p^3}. By Theorem 6.4.1, we need $6M_2 \approx 18M$ each to compute f^2, $(f^2)^2$ and $(f^p)^2$. Similarly, we need $6 \cdot 3M_2 + M_{12} \approx 108M$ to compute the 9-th powering. Finally, $8M_{12} \approx 432M$ to multiply all terms together. Therefore, the total cost of the whole final exponentiation is $4349M + 2I$ using the squaring methods suggested in [18] and [14]. This estimate is significantly faster than the $6291M$ estimated in Chapter 5.

Recall that the only method in this chapter which actually speeds up the Miller loop computation is to use projective coordinates instead of jacobian coordinates. The cost of the Miller loop decreases from $7164M$ to $6668M$. Therefore, the best operation count we can achieve for R-ate pairing computation on our chosen BN curve is

$$6668M + 323M + 2I + 4349M + 2I = 11340M + 4I,$$

which is 20% faster than the operation count of $13778M + 2I$ in Chapter 5.

Chapter 7

Concluding Remarks

Optimal ate pairings are known to be the fastest pairings for single pairing computation. The implementation of optimal ate pairings on pairing-friendly BN curves currently holds the speed record. Vercauteren gave a method to characterize optimal ate pairings by considering a fixed power of the Tate pairing [30]. According to the discussion at the end of Section 3.4, Vercauteren's method does not characterize all optimal ate pairings.

In this thesis, we evaluated the multiplication costs for computation of a single R-ate pairing with jacobian coordinates, projective coordinates and affine coordinates. With the improved formulas provided by Costello, Lange and Naehrig [12], projective coordinates turn out to be the best for implementing the optimal ate pairings. In contrast, Lauter, Montgomery and Naehrig [19] claimed that affine coordinates are more efficient on their platform. We find that their cost ratios A/M (cost ratio between addition and multiplication) and M/I (cost ratio between multiplication and inversion) are very large compared with the commonly-accepted ratios. Affine coordinates are more efficient if A/M and M/I are large, while projective coordinates are more efficient if A/M and M/I are small. A careful operation count (including the cost of additions) can be applied to determine the break-even point. However, it would appear that projective coordinates are superior on all existing platforms.

The delaying idea introduced by Costello, Boyd, Nieto and Wong in [10] and [11] is not suitable to compute a single optimal ate pairing. However, if multiple pairings are computed in parallel so that the cost of an inversion is as small as the cost of three multiplications in the same field, the delaying idea may be worth using. The efficiency of final exponentiation is significantly improved using the faster squaring method introduced by Granger and Scott [14] and Karabina [18].

Finally, we used the improved formulas for projective coordinates by Costello, Lange and Naehrig [12] to compute the Miller loop and use the faster squaring method intro-

duced in [18] and [14] to compute the final exponentiation. The overall cost of a pairing computation is 20% faster than the operation counts derived in Chapter 5.

References

[1] D. Aranha, K. Karabina, P. Longa, C. Gebotys and J. Lopez, Faster explicit formulas for computing pairings over ordinary curves, Advances in Cryptology - EUROCRYPT 2011, Lecture Notes in Computer Science, 663, 48-68, 2011. 41, 69

[2] R. Balasubramanian and N. Koblitz, The improbability that an elliptic curve has subexponential discrete log problem under the Menezes-Okamoto-Vanstone algorithm, Journal of Cryptology, 11, 141-145, 1998. 9

[3] P. Barreto, B. Lynn and M. Scott, Efficient implementation of pairing-based cryptosystems, Journal of Cryptology, 17, 321-334, 2004. 19

[4] P. Barreto and M. Naehrig, Pairing-friendly elliptic curves of prime order, Selected Areas in Cryptography (SAC 2005), Lecture Notes in Computer Science 3897 (2006), 354-368, 2005. 34

[5] D. Boneh and M. Franklin, Identity-based encryption from the Weil Pairing, Advances in Cryptology - CRYPTO 2001, Lecture Notes in Computer Science, 2139, 213-229, 2001. 23

[6] D. Boneh, C. Gentry, B. Lynn and H. Shacham, Aggregate and verifiably entrypted signatures from bilinear maps, Advances in Cryptology - EUROCRYPT 2004, Lecture Notes in Computer Science, 2656, 416-432, 2003. 21, 22

[7] D. Boneh, B. Lynn and H. Shacham, Short signatures from the Weil pairing, Advances in Cryptology - ASIACRYPT 2001, Lecture Notes in Computer Science, 2248, 514-532, 2001. 21

[8] S. Chatterjee, D. Hankerson, E. Knapp and A. Menezes, Comparing two pairing-based aggregate signature schemes, Designs, Codes and Cryptography, 55, 141-167, 2010. 22

[9] J. Chung and A. Hasan, Asymmetric squaring formula, 18th IEEE Symposium on Computer Arithmetic (ARITH 07), 113-122, 2007 45

[10] C. Costello, C.Boyd, J. Nieto and K. Wong, Delaying mismatched field multiplications in pairing computations, International Workshop on the Arithmetic of Finite Fields - WAIFI 2010, Lecture Notes in Computer Science, 6087, 196-214, 2010. 57, 70, 72, 77

[11] C. Costello, C.Boyd, J. Nieto and K. Wong, Avoiding full field arithmetic in pairing computations, AFRICACRYPT 2010, Lecture Notes in Computer Science, 6055, 203-224, 2010. 57, 72, 73, 74, 77

[12] C. Costello, T. Lange and M. Naehrig, Faster pairing computations on curves with high-degree twists, Public Key Cryptography - PKC 2010, Lecture Notes in Computer Science, 6056, 224-242, 2010. 57, 62, 63, 64, 77

[13] A. Devegili, M. Scott and R. Dahab, Implementing cryptographic pairings over Barreto-Naehrig curves, Pairing-Based Cryptography - Pairing 2007, Lecture Notes in Computer Science, 4575, 197-207, 2007. 54, 63, 69

[14] R. Granger and M. Scott, Faster squaring in the cyclotomic subgroup of sixth degree extensions, Public Key Cryptography - PKC 2010, Lecture Notes in Computer Science, 6056, 209-223, 2010. 75, 76, 77, 78

[15] D. Hankerson, A. Menezes and M. Scott, Software implementation of pairings, M. Joyed, G. Neven (eds) Identity-Based Cryptography, IOS Press, 2008. 42, 69

[16] F. Hess, A note on the Tate pairing of curves over finite fields, Archive for Mathematical Logic, 82, 28-32, 2004. 17

[17] F. Hess, N. Smart and F. Vercauteren, The eta pairing revisited, IEEE Transactions on Information Theory, 52, 4595-4602, 2006. 26

[18] K. Karabina, Squaring in cyclotomic subgroups, Cryptology ePrint Archive, Report 2010/542, 2010. 57, 75, 76, 77, 78

[19] K. Lauter, P. L. Montgomery and M. Naehrig, An analysis of affine coordinates for pairing computation, Pairing Based Cryptography - Pairing 2010, Lecture Notes in Computer Science, 6487, 1-20, 2010. viii, 57, 64, 68, 69, 77

[20] E. Lee, H. Lee and C. Park, Efficient and generalized pairing computation on abelian varieties, IEEE Transactions on Information Theory, 55, 1793-1803, 2009. 28, 29

[21] V. Miller, The Weil pairing and its efficient calculation. Journal of Cryptology, 17, 235-261, 2004. 17

[22] P. Montgomery, Speeding the Pollard and elliptic curve methods of factorization, Mathematics of Computation 48, 243-264, 1987. 66

[23] M. Naehrig, Constractive and computational aspects of cryptographic pairings, PhD thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2009. 36

[24] Y. Nogami, M. Akane, Y. Sakemi, H. Kato and Y. Morikawa, Integer variable χ-based ate pairing, Pairing-Based Cryptography - Pairing 2008, Lecture Notes in Computer Science, 5209, 178-191, 2008.

[25] G. Pereira, M. Simplicio Jr., M. Naehrig and P. Barreto, A family of implementation-friendly BN elliptic curves, Cryptology ePrint Archive, Report 2010/429, 2010. 38, 41

[26] R. Schroeppel and C. Beaver, Accelerating elliptic curve calculations with the reciprocal sharing trick, Mathematics of Public-Key Cryptography, University of Illinois at Chicago, 2003. 67

[27] M. Scott, Implementing cryptographic pairings, Pairing-Based Cryptography - Pairing 2007, Lecture Notes in Computer Science, 4575, 177-196, 2007. 45, 69

[28] A. Shamir, Identity-based cryptosystems and signature schemes, Advances in Cryptology - Proceedings of CRYPTO 84, Lecture Notes in Computer Science, 196, 47-53, 1985. 22

[29] M. Stam and A. Lenstra, Efficient subgroup exponentiation in quadratic and sixth degree extensions, Cryptographic Hardware and Embedded System - CHES 2002, Lecture Notes in Computer Science, 4249, 134-147, 2006. 45

[30] F. Vercauteren, Optimal Pairings, IEEE Transactions on Information Theory, 56, 455-461, 2010. 25, 28, 31, 32, 77

[31] L. Washington, Elliptic curves cumber theory and cryptography, second edition, CRC Press, 2008. 3, 6, 10

[32] C. Zhao, F. Zhang and J. Huang. A Note on the Ate Pairing, Cryptology ePrint Archive: Report 2007/247, 2007. 28