九州大学学術情報リポジトリ
Kyushu University Institutional Repository

Efficient Algorithms for Identity-Based Encryption using Supersingular Elliptic Curves

富田, 琢巳

https://doi.org/10.15017/1441044

出版情報：九州大学, 2013, 博士（機能数理学）, 課程博士
バージョン：
権利関係：全文ファイル公表済

Efficient Algorithms for Identity-Based Encryption using Supersingular Elliptic Curves

Takumi TOMITA

A thesis submitted in fulfillment of the requirements for the Ph.D. degree in Functional Mathematics

Supervisor: Prof. Tsuyoshi TAKAGI

Graduate School of Mathematics Kyushu University

February 2014

ABSTRACT

In this thesis, we report the results of faster computation of Identity-Based Encryption (IBE), which was proposed by D. Boneh and M. Franklin in 2001 (called BF-IBE). This scheme is acknowledged as the first practical IBE and it is standardized by the Internet Engineering Task Force (IETF), which develops and promotes Internet standards. For practical use, improved computation speed is required particularly for embedded device systems such as smart phones and mobile phones, because of limited computing resources (CPU and memory etc.) in embedded devices. We set a restriction in which we minimize the impact on popular applications. That is, we choose a parameter optimization approach, which does not require any changes to be made to the algorithms used in many applications.

In BF-IBE, there is a hash-to-map function called HashToPoint, which maps an identity to a point on an elliptic curve. However, HashToPoint typically requires a modular exponentiation, which is relatively expensive compared with other cryptographic functions in BF-IBE. Therefore, we focus on a speed improvement of HashToPoint, as outlined below.

We observed that there is a parameter, called the cofactor, which can save computation costs in HashToPoint. We propose some cofactors that efficiently compute HashToPoint without losing the speed of other cryptographic functions in IBE. Most of the processing time of HashToPoint depends on a scalar

multiplication on a point of an elliptic curve by the cofactor which is a large integer. To speed up scalar multiplication, we chose a cofactor with low Hamming Weight. From the distribution of primes in the arithmetic progression, we estimate the number of such cofactors for a fixed low Hamming Weight. However, from an efficiency perspective, it is important to list such cofactors. First, we identify them, and then we find those cofactors that have Hamming Weight of 2. Next, we measure the timing of HashToPoint on a desktop PC using one of the cofactors with Hamming Weight of 2 and the pairing library PBC. To fairly compare the improved efficiency, we also choose a random cofactor using the PBC library and measure the timing of HashToPoint using both cofactors. Finally, we find that the timing for our implementation of HashToPoint using the cofactor with Hamming Weight of 2 is reduced by approximately 30% on a desktop PC.

ACKNOWLEDGEMENTS

Firstly, I express my great gratitude to Professor Tsuyoshi Takagi for his continuing support and constant encouragement during my doctoral course. I obtained many valuable advices and constructive feedbacks from him. Without his guidance and persistent help, this dissertation would not have been completed. I would like to express my appreciation to Professor Masanobu Kaneko to give me this opportunity to study the latest technology in Elliptic Curve Cryptography at Kyushu University. I would also like to thank Yuichiro Taguchi for encouraging my research. From 2002 to 2007, I have learned the basis of the theory of elliptic curves with the knowledge of Arithmetic Geometry from him. This study is owed to this basic knowledge of mathematics. I would like to thank all the referees and examiners for careful reading of my manuscript and for giving useful comments. Many thanks go to my supervisors and colleagues at the Hitachi, Ltd for their consideration and encouragement. Finally, I express my heartfelt thanks to my family. I thank my parents, who have always stimulated me to stay open-minded and to keep on doing the best I could. Special thanks to my wife, Fumiko, for her love and patient support, above all, for taking care of our little sons. Thanks to her support, I could devote much more time to my studies and research.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

List of Algorithms

CHAPTER 1

INTRODUCTION

1.1 IDENTITY-BASED ENCRYPTION

Public key cryptography (PKC), also known as asymmetric cryptography, plays an important role in information and system security [94]. The most successful application of public key technology has been SSL, which requires minimal interaction. However, there is a trust issue in PKC with public keys. When Alice wants to send a message to Bob, she uses Bob's public key to encrypt the message. Suppose Eve masquerades as Bob, there is an attacking technique known as man-in-the-middle-attack. To prevent this type of attack, Alice needs to ensure that the public key that is claimed to be Bob's does indeed belong to Bob. This is achieved by using an authority that both Alice and Bob trust. This authority issues certificates for public keys and is called a certifying authority (CA). However, it seems that this is too difficult for many users [112, 96]. Poor usability causes high-support costs for technology users, and this has probably been one of the major factors hindering the widespread adoption of PKC.

The problem associated with the practical deployment of PKC motivated Shamir to introduce the concept of identity-based cryptography (IBC) [95].

Shamir posed a challenge to the crypto-community to create a practical IBC. A satisfactory solution eluded researchers until the turn of the millennium, and then came from three different quarters: Sakai-Ohgishi-Kasahara[89], Boneh-Franklin [16] and Cocks[23]. The identity-based encryption scheme (IBE), except for Cocks' scheme, is one of the most well-known pairing-based cryptosystems, and is a kind of public key encryption scheme where the public key of a user can be any arbitrary string, typically the e-mail address. In IBE, when Alice securely sends a message to Bob, she can encrypt the message with her identity, such as the e-mail address, as Bob's public key. Bob can easily check whether the public key is correct without a certificate authority, since the identity is a known string.

In IBE, the computation of each procedure is the most costly operation in each algorithm. To make IBE feasible in a practical sense, it is important to implement the computations as efficiently as possible. In particular, speed is desirable for embedded devices such as smart phones and mobile phones, because computing resources (CPU and memory etc.) in embedded devices are often limited.

1.2 KNOWN RESULTS

There are three approaches for speeding up as follows:

A) Protocol and Algorithm Optimization
B) Parameter Optimization
C) Hardware Optimization

In approach A), suitable elliptic curves and their base fields are used to speed up Miller's algorithm, which can efficiently calculate pairings [72, 74]. Other algorithms which calculate the pairing value faster, such as [4, 5, 36]

were examined. Duursma and Lee proposed an efficient algorithm for computing the Tate pairing on hyperelliptic curves [27]. Barreto et al. indicated a η_T pairing [2], which is a different version of the Duursma-Lee algorithm, and is approximately twice as fast. In addition, various algorithms to compute pairing, such as Ate pairing [49], Ate_i pairing [113], R-ate pairing [62], and Optimal pairing [103] have also been proposed to date.

In approach B), we achieve the speeding up of computation over finite fields or points operations on an elliptic curve, by selecting the appropriate system parameters, such as the elliptic curve itself or a characteristic of a finite field. In this approach, there is a contribution to improve the cost of computing the pairing related to Miller's algorithm and Montgomery multiplication. In fact, Nakajima et al. proposed efficient primes p which have low Hamming Weight to speed up the computation of a Montgomery multiplication inside the Miller's algorithm, and achieved a speeding up of approximately 22% in computation of the latter [78].

In approach C), hardware is described for implementing the speeding up of the computation of field arithmetic or pairing computations by integrated circuit tools, such as Gate, VLSI, FPGA, and ASIC [47, §5.2]. There are many hardware implementations of pairing accelerators which have been proposed [1, 7, 98].

Because our main purpose is to achieve the speeding up of IBE on the condition that we minimize the impact for applications, we select approach B).

1.3 CONTRIBUTION OF THIS THESIS

The practical IBE which was due to Boneh and Franklin [16] is standardized in RFC5091 [19], which is denoted by BF-IBE. BF-IBE depends on certain properties of pairings on special elliptic curves, which are supersingular elliptic curves. Either the Weil or Tate Pairing can be used, although it is recognized that the latter will always be faster. When encrypting to an identity, there is a requirement to compute $Q_{\mathrm{ID}} = H_1(\mathrm{ID}) \in \mathbb{G}_1^{\times}$ and then $g_{\mathrm{ID}} = \widehat{T}_r^{\mathrm{mod}}(rQ_{\mathrm{ID}}, P_{\mathrm{pub}})$, where Q_{ID} and P_{pub} are points on an elliptic curve of order r. However $H_1(\cdot)$, which is called HashToPoint, in practice is actually a hash-and-map function that must first hash the input identity to a point on the curve, and then map it to a point of order r for the suggested supersingular curve over the prime field $\mathbb{F}_p$ $(p \geq 5)$. From the viewpoint of security, NIST recommends keys of size at least 80 (the size of the key space here is 2^{80}, which is a lot of brute force work for an attacker)— this condition means p is 512 bit and r is 160 bit. For the detail of relation between key size and security level, see §3.4. Under this condition, the mapping requires a point multiplication by a $352(= 512 - 160)$ bit cofactor. This cost is likely to dwarf the cost of calculating the pairing. The main feature of faster computation is that we choose such a cofactor with low Hamming Weight to speed up the scalar multiplication, which saves extra additional operation of points on an elliptic curve.

In this thesis, we propose some system parameters that efficiently compute the HashToPoint without losing the speed of other cryptographic functions in the IBE. From the distribution of primes in the arithmetic progression, we estimate the number of such cofactors for a fixed size. However, it is important from an industrial viewpoint to list such cofactors. First, we explore their ex-

istence, then we find several cofactors with Hamming Weight of 2. Next, we measure the timing of HashToPoint on a desktop PC, using one of the cofactors with Hamming Weight of 2, using the pairing library PBC [66]. To fairly compare improved efficiency, we also choose a random cofactor with Hamming eight of 182 using the PBC library, and measure the timing of HashToPoint using both cofactors. Finally, we find that the timing of our implementation of HashToPoint using the cofactor with Hamming Weight 2 is reduced by approximately 30% on a desktop PC.

1.4 THESIS STRUCTURE

The thesis is organized as follows. Chapter 2 deals with the mathematical theory of the pairing cryptography, while focusing on cryptographic applications. Chapter 3 discusses BF-IBE, which is the first practical and secure IBE scheme. Chapters 4 and 5 comprise the main results of this thesis. In Chapter 4, we propose a search method to find cofactors that realize fast computation for the curves, and list the cofactors. Chapter 5 shows the timing results of the computation used by our proposed cofactors in C. Finally, the thesis is concluded in Chapter 6.

CHAPTER 2

MATHEMATICAL BACKGROUND

This chapter contains a review of all of the necessary definitions needed in the following chapters. For the more detailed description such as theorems and proofs, refer to the books [11, 22, 24, 47, 50, 65, 85].

2.1 FINITE FIELDS

In this section we will recall basic properties of groups, rings and fields.

Group:

Definition 2.1.1. Given a set S, a composition law $\times$ of S into itself is a mapping from the Cartesian product $S \times S$ to S. Common notations for the image of (x, y) under this mapping are $x \times y$, $x * y$ or simply xy. When the law is commutative, it is customary to denote it by $+$.

Definition 2.1.2. A group G is a set with a composition law $\times$ such that

- $\times$ is associative, that is for all $x, y \in G$ we have $(xy)z = x(yz)$.
- $\times$ has a unit element e, that is for all $x \in G$ we have $xe = ex = x$.
- For every $x \in G$ there exists $y \in G$, an inverse of x, such that $xy = yx = e$.

Remark. The unit of a group G is necessarily unique as well as the inverse of an element x that is denoted by x^{-1}. If G is commutative the inverse of x is usually denoted by $-x$.

Definition 2.1.3. Let $x \in G$. The set $\{x^n \mid n \in \mathbb{Z}\}$ is the subgroup of G generated by x. It is denoted by $\langle x \rangle$.

Ring:

Definition 2.1.4. A ring R is a set together with two composition laws $+$ and $\times$ such that

- R is a commutative group with respect to $+$.
- $\times$ is associative and has a unit element 1, which is different from 0, the unit of $+$.
- $\times$ is distributive over $+$, that is for all $x, y, z \in R$, $x(y+z) = xy + xz$ and $(y+z)x = yx + zx$.

Remark. The ring R is said to be commutative, if the law $\times$ is commutative.

We define an important arithmetic invariant called characteristic. Let R be a ring and let ψ be the natural ring homomorphism from $\mathbb{Z}$ to R where $\mathbb{Z}$ is the set of integer.

$$\psi(n) = \begin{cases} 1 + 1 + \cdots + 1 & n \text{ times if } n \geq 0 \\ -(1 + 1 + \cdots + 1) & -n \text{ times otherwise.} \end{cases} \tag{2.1}$$

The kernel of ψ is an ideal of $\mathbb{Z}$ and if the multiples of 1 are all different then $\ker\psi = \{0\}$. Otherwise, for example if R is finite, some multiples of 1 must be zero. In other words, the kernel of ψ is generated by a positive integer m.

Definition 2.1.5. Let R be a ring and ψ defined as above. The kernel of ψ is of the form $m\mathbb{Z}$, for some non-negative integer m, which is called the characteristic of R and is denoted by $\mathrm{char}(R)$.

Field:

Definition 2.1.6. A field K is a commutative ring such that every nonzero element is invertible.

Proposition 2.1.1. The characteristic of a field is either 0 or a prime number p.

Finite Field:

Definition 2.1.7. A finite field is a field whose order is finite. Finite fields are also referred to as Galois fields.

The order of a finite field is the number of elements in the field. There exists a finite field $\mathbb{F}$ of order q if and only if q is a prime power. That is $q = p^m$ where p is a prime number called the characteristic of $\mathbb{F}$ and m is a positive integer. If $m = 1$, then $\mathbb{F}$ is called a prime field. For any prime power q, there is essentially only one finite field of order q. This means that any two finite fields of order q are structurally the same, we say that any two finite fields of order q are isomorphic and denote by $\mathbb{F}_q$. The finite field $\mathbb{F}_{p^m}$ can be viewed as a vector space over its subfield (base field) $\mathbb{F}_p$.

The nonzero elements of a finite field $\mathbb{F}_q$, denoted $\mathbb{F}_q^\times$, form a cyclic group under multiplication. Hence there exist elements $b \in \mathbb{F}_q^\times$ called generators such that

$$\mathbb{F}_q^\times = \{b^i \mid 0 \leq i \leq q-2\}.$$

The order of $a \in \mathbb{F}_q^\times$ is the smallest positive integer t such that $a^t = 1$. Since $\mathbb{F}_q^\times$ is a cyclic group, it follows that t is a divisor of $q-1$.

The fields that are useful in cryptography are large fields with $p \geq 2^{512}$. Some commonly used fields are of the following types:

- $\mathrm{char}(\mathbb{F}) = 2$: This field is called binary field, and implementation is done using the so-called polynomial basis. There are suitable algorithms for performing arithmetic in the prime field $\mathbb{F}_{2^m}$. For the detail, see [47, §2.3].

- $\mathrm{char}(\mathbb{F}) = 3$: This field has been suggested mainly for pairing based cryptography.

- $\mathrm{char}(\mathbb{F}) = p(\neq 2, 3)$ where p is large: This is the field we are going to use in this thesis. There are suitable algorithms for performing arithmetic in the prime field $\mathbb{F}_p$. For the detail, see [47, §2.2].

2.2 ELLIPTIC CURVES

In this thesis, we set the following notation:

K	a perfect field (i.e. every algebraic extension of K is separable.)
$\overline{K}$	a fixed algebraic closure of K
$G_{\overline{K}/K}$	the Galois group of $\overline{K}/K$

We begin our study of algebraic geometry with affine/projective n-space. We begin our review of algebraic geometry with affine and projective space. Let K be a field and $\overline{K}$ be its algebraic closure. An affine n-space over K , which we denote by $\mathbb{A}^n$ (or $\mathbb{A}^n_{/K}$), is the set of n-tuples

$$\mathbb{A}^n = \mathbb{A}^n(\overline{K}) = \{P = (x_1, x_2, \ldots, x_n) \mid x_i \in \overline{K}\}.$$

The set of K-rational points in $\mathbb{A}^n$ is the set

$$\mathbb{A}^n(K) = \{P \in \mathbb{A}^n \mid P^\sigma = P \text{ for all } \sigma \in G_{\overline{K}/K}\}.$$

This is the same as follows:

$$\mathbb{A}^n(K) = \{P = (x_1, x_2, \ldots, x_n) \in \mathbb{A}^n \mid x_i \in K\}.$$

Next we define a projective space. A projective n-space $\mathbb{P}^n$ is the set of lines through the origin in $\mathbb{A}^{n+1}$. In symbols,

$$\mathbb{P}^n = \mathbb{P}^n(\overline{K}) = \frac{\{(x_1, x_2, \ldots, x_n) \in \mathbb{A}^{n+1} \mid \text{some } x_i \neq 0\}}{\sim} = \frac{\mathbb{A}^{n+1} \setminus \{0\}}{\sim},$$

where the equivalence relation $\sim$ is defined by

$$\begin{aligned}&(x_1, x_2, \ldots, x_{n+1}) \sim (y_1, y_2, \ldots, y_{n+1})\\ &\Leftrightarrow (x_1, x_2, \ldots, x_{n+1}) = \lambda(y_1, y_2, \ldots, y_{n+1}) \text{ for some } \lambda \in \overline{K}^\times\end{aligned}$$

An equivalence class of the form

$$\{(\lambda x_1, \lambda x_2, \ldots, \lambda x_{n+1}) \mid \lambda \in \overline{K}\}$$

is represented by $[x_1 : x_2 : \ldots : x_{n+1}]$. The set of K-rational points in $\mathbb{P}^n$ is teh set

$$\begin{aligned}\mathbb{P}^n(K) &= \{P \in \mathbb{P}^n \mid P^\sigma = P \text{ for all } \sigma \in G_{\overline{K}/K}\}\\ &= \{P = [x_1 : x_2 : \ldots : x_n] \in \mathbb{P}^n \mid x_i \in K\}.\end{aligned}$$

2.2.1 WEIERSTRASS EQUATIONS

Definition 2.2.1. An elliptic curve is a pair $(E, \mathcal{O})$, where E is a smooth projective algebraic curve of genus 1 with $\mathcal{O} \in E$. We often just write E for the elliptic curve, the point $\mathcal{O}$ being understood. The elliptic curve E is over a field K, written $E_{/K}$, if E is over a field K as a curve and $\mathcal{O} \in E(K)$.

Every such curves can be written as the locus in $\mathbb{P}^2$ of a cubic equation with only one point on the line at ∞.

Theorem 2.2.1. Let $E_{/K} = (E_{/K}, \mathcal{O})$ be a elliptic curve over a field K. Then there exist constants $a_1, a_2, a_3, a_4, a_6 \in K$ such that $E_{/K}$ is isomorphic over K to the smooth plane cubic given by the equation

$$E : Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3. \tag{2.2}$$

Under this isomorphism, the point $\mathcal{O}$ is mapped to the inflection point $[X : Y : Z] = [0 : 1 : 0] \in E$.

The above equation is called a homogeneous Weierstrass equation. Frequently, these equations are written in affine coordinates (i.e. by setting $Z = 1$), where it is understood that there is one additional point $\mathcal{O} = [0 : 1 : 0]$. Namely, a plane nonsingular affine part E_a of E is given by

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{2.3}$$

where $a_1, a_2, a_3, a_4, a_6 \in K$ and $E \setminus E_a$ consists of one point with homogeneous coordinates $[0 : 1 : 0]$. We call this equation affine Weierstrass equation. In the remainder of this thesis, E will be a standard notation for an elliptic curve given by a homogeneous Weierstrass equation, and we will often abuse nota-

tion and denote by E the affine part E_a.

Definition 2.2.2. Let E be a curve over a field K by (2.3) and let a_1, a_2, a_3, a_4, a_6 be as above. The discriminant of the curve E denoted by Δ_E satisfies

$$\Delta_E = -b_2^2 b_8 - 8b_4^3 - 27b_6^2 + 9b_2 b_4 b_6,$$

where

$$b_2 = a_1^2 + 4a_2, \quad b_4 = a_1 a_3 + 2 + a_4$$
$$b_6 = a_3^2 + 4a_6, \quad b_8 = a_1^2 a_6 - a_1 a_3 a_4 + 4a_2 a_6 + a_2 a_3^2 - a_4^2.$$

The curve E is nonsingular, and thus is an elliptic curve, if and only if Δ_E is nonzero. In this case, we introduce the j-invariant of E as follows:

Definition 2.2.3. If $\Delta_E \neq 0$, we define j-invariant of E by

$$j_E = \frac{(b_2^2 - 24b_4)^3}{\Delta_E}.$$

Theorem 2.2.2. Let K be a field. The isomorphism classes of elliptic curves E over K are, up to twist, uniquely determined by the absolute invariants j_E, and for every $j \in K$ there exists an elliptic curve E with absolute invariant $j_E = j$.

If K is algebraically closed then the isomorphism classes of elliptic curves over K correspond one-to-one to the elements in K via the map $E \mapsto j_E$.

Short Weierstrass Equation:

Let us first describe the transformations that keep the curve in Weierstrass form. Let $E_{/K}$ be an elliptic curve over a field K given by

$$E : y^2 + a_1 xy + a_3 y = x^3 + a_2 x^2 + a_4 x + a_6.$$

The map

$$(x, y) \mapsto (u^2x' + r, u^3y' + u^2sx' + t)$$

with $(u, r, s, t) \in K^\times \times K^3$ is invertible and transforms the curve E into

$$E' : y'^2 + a'_1x'y' + a'_3y' = x'^3 + a'_2x'^2 + a'_4x' + a'_6,$$

where the a_i's belong to K and can be expressed in terms of the a_i's and u, r, s, t. This transformation is called an admissible change of variables. Via the inverse map, we associate to each point of E a point of E' showing that both curves are isomorphic over K. There changes of variables are the only ones leaving the shape of the defining equation invariant and, hence, they are the only admissible change of variables. In case (u, r, s, t) belongs to $\overline{K}^\times \times \overline{K}^3$ whereas the curves E and E', as above, are still defined over K, then E and E' are isomorphic over $\overline{K}$ or twists of each other.

The Weierstrass equation can be simplified considerably by applying admissible changes of variables. We consider separately the cases where the underlying field K has characteristic different from 2 and 3, or has characteristic equal to 2 or 3.

Case char $K \neq 2, 3$**:**

$$(x, y) \mapsto \left(\frac{x - 3q_1^2 - 12a_2}{36}, \frac{y - 3a_1x}{216}, \frac{a_1^3 + 4a_1a_2 - 12a_3}{24}\right)$$

transforms E to the curve

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

where $a, b \in K$. The discriminant and j-invariant are

$$\Delta_E = -16(4a_4^3 + 27a_6^2),\ j_E = 1728a_4^3/\Delta_E.$$

Case char $K = 2$**:**

$$(x, y) \mapsto \left(a_1^2 x + \frac{a_3}{a_1}, a_1^3 y + \frac{a_1^2 a_4 + a_3^2}{a_1^3} \right).$$

The Weierstrass equation is transformed to the following curve by the above admissible change of variables:

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

where $a, b \in K$. The discriminant and j-invariant are

$$\Delta_E = b,\ j_E = 1/b.$$

If $a_1 = 0$, then the curve transforms by the admissible change of variables

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

where $a, b, c \in K$. The discriminant and j-invariant are

$$\Delta_E = c^4,\ j_E = 0.$$

Case char $K = 3$**:**

If the characteristic of K is equal to 3, if $a_1^2 \neq -a_2$, then the admissible change of variables

$$(x, y) \mapsto \left(x + \frac{d_4}{d - 2}, y + a_1 x + a_1 \frac{d_4}{d_2} + a_3 \right)$$

where $d_2 = a_1^2 + a_2, d_4 = a_4 - a_1 a_3$, and the Weierstrass equation is transformed into the curve

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

Table 2.1: Short Weierstrass Equations

Char K	Equation	Discriminant Δ	j-invariant
$\neq 2,3$	$y^2 = x^3 + a_4x + a_6$	$-16(4a_4^3 + 27a_6^2)$	$1728a_4^3/\Delta$
3	$y^2 = x^3 + a_4x + a_6$	$-a_4^3$	0
	$y^2 = x^3 + a_2x^2 + a_6$	$-a_2^3a_6$	$-a_2^3a_6/a_6$
2	$y^2 + a_3y = x^3 + a_4x + a_6$	a_3^4	0
	$y^2 + xy = x^3 + a_2x^2 + a_6$	a_6	$1/a_6$

where $a, b \in K$. The discriminant and j-invariant are

$$\Delta_E = -a_3b,\ j_E = -\frac{a^3}{b}.$$

If $a_1^2 = -a_2$, then the admissible change is

$$(x, y) \mapsto (x, y + a_1x + a_3)\,.$$

For $a, b \in K$, the Weierstrass equation then transforms to the curve

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

where $a, b \in K$. The discriminant and j-invariant are

$$\Delta_E = -a^3,\ j_E = 0.$$

We summarize short Weierstrass equation in Table 2.1.

2.2.2 THE GROUP LAW

Let E be an elliptic curve given by a Weierstrass equation. Remember that $E \subset \mathbb{P}^2$ consists of the points $P = (x, y)$ satisfying the equation together with the point $\mathcal{O} = [0 : 1 : 0]$. Let $L \subset \mathbb{P}^2$ be a line. Then since the equation has degree three, L intersects E at exactly three points, say $P, Q, R \in E$. Define a composition law " $+$ " on E by the following rule:

Definition 2.2.4 (Composition Law)**.** Let $P, Q \in E$, L the line connecting P and Q (L is tangent line to E if $P = Q$), and R the third point of intersection of L with E. Let L' be the line connecting R and $\mathcal{O}$. Then $P + Q$ is the point such that L' intersects E at $R, \mathcal{O}$, and $P + Q$.

We now justify the above Composition Law.

Proposition 2.2.3. The composition law Definition 2.2.4 has the following properties:

(i) If a line L intersects E at the (not necessarily distinct) points P, Q, R, then $(P + Q) + R = \mathcal{O}$.

(ii) $P + \mathcal{O} = P$ for all $P \in E$.

(iii) $P + Q = Q + P$ for all $P, Q \in E$.

(iv) Let $P \in E$. There is a point of E, denoted $-P$, so that $P + (-P) = \mathcal{O}$.

(v) Let $P, Q, R \in E$. Then $(P + Q) + R = P + (Q + R)$.

Proposition 2.2.4. Suppose E is over K. Then K-rational points of E, written $E(K)$, is a subgroup of E.

Case $y^2 = x^3 + a_4x + a_6$ (char $K \neq 2, 3$):

For the elliptic curve $y^2 = x^3 + ax + b$, the formulas of the negative point, the point addition and point doubling are as follows:

Identity $P + \mathcal{O} = \mathcal{O} + P = P$ for all $P \in E(K)$.

Negatives $P = (x, y) \in E(K)$, then $-P = (x, -y)$.

Addition

Let $P = (x_1, y_1), Q = (x_2, y_2)$ be points in $E(K)$ such that $P \neq \pm Q$. Then $R = (x_3, y_3) = P + Q$ is computed by

$$(x_3, y_3) = (\lambda^2 - (x_1 + x_2), \lambda(x_1 - x_3) - y_1),$$

where $\lambda = (y_2 - y_1)/(x_2 - x_1)$.

Doubling

Let $P = (x_1, y_1)$ be a point in $E(K)$ where $P \neq -P$. Then $R = (x_3, y_3) = 2P$ is computed by

$$(x_3, y_3) = (\lambda^2 - 2x_1, \lambda(x_1 - x_3) - y_1),$$

where $\lambda = (3x_1^2 + a)/2y_1$.

Case $y^2 = x^3 + a_4x + a_6$ (char $K = 3$):

In this case, the computation is same as the above one ($y^2 = x^3 + a_4x + a_6$ ($\mathrm{char}\, K \neq 2, 3$)).

Case $y^2 = x^3 + a_2x^2 + a_6$ (char $K = 3$):

Negatives $P = (x, y) \in E(K)$, then $-P = (x, -y)$.

Addition

Let $P = (x_1, y_1), Q = (x_2, y_2)$ be points in $E(K)$ such that $P \neq \pm Q$. Then

$R = (x_3, y_3) = P + Q$ is computed by

$$(x_3, y_3) = (\lambda^2 - (x_1 + x_2 + a), \lambda(x_1 - x_3) - y_1),$$

where $\lambda = (y_2 - y_1)/(x_2 - x_1)$.

Doubling

Let $P = (x_1, y_1)$ be a point in $E(K)$ where $P \neq -P$. Then $R = (x_3, y_3) = 2P$ is computed by

$$(x_3, y_3) = (\lambda^2 - 2x_1, \lambda(x_1 - x_3) - y_1),$$

where $\lambda = ax_1/y_1$.

CASE $y^2 + a_3y = x^3 + a_4x + a_6$ (char $K = 2$):

Identity $P + \mathcal{O} = \mathcal{O} + P = P$ for all $P \in E(K)$.

Negatives $P = (x, y) \in E(K)$, then $-P = (x, y + c)$.

Addition

Let $P = (x_1, y_1), Q = (x_2, y_2)$ be points in $E(K)$ such that $P \neq \pm Q$. Then $R = (x_3, y_3) = P + Q$ is computed by

$$(x_3, y_3) = (\lambda^2 + x_1x_2, \lambda(x_1 + x_3) + y_1 + c),$$

where $\lambda = (y_1 + y_2)/(x_1 + x_2)$.

Doubling

Let $P = (x_1, y_1)$ be a point in $E(K)$ where $P \neq -P$. Then $R = (x_3, y_3) = 2P$

is computed by

$$(x_3, y_3) = (\lambda^2, \lambda(x_1 + x_3) + y_1 + c),$$

where $\lambda = (x_1^2 + a)/c$.

CASE $y^2 + xy = x^3 + a_2x^2 + a_6$ (char $K = 2$):

Identity $P + \mathcal{O} = \mathcal{O} + P = P$ for all $P \in E(K)$.

Negatives $P = (x, y) \in E(K)$, then $-P = (x, x + y)$.

Addition

Let $P = (x_1, y_1), Q = (x_2, y_2)$ be points in $E(K)$ such that $P \neq \pm Q$. Then $R = (x_3, y_3) = P + Q$ is computed by

$$(x_3, y_3) = (\lambda^2 + \lambda + x_1 + x_2, \lambda(x_1 + x_3) + x_3 + y_1),$$

where $\lambda = (y_1 + y_2)/(x_1 + x_2)$.

Doubling

Let $P = (x_1, y_1)$ be a point in $E(K)$ where $P \neq -P$. Then $R = (x_3, y_3) = 2P$ is computed by

$$(x_3, y_3) = (\lambda^2 + \lambda + a, x_1^2 + \lambda x_3 + x_3),$$

where $\lambda = (x_1 + y_1)/x_1$.

Scalar Multiplication:

Take a positive integer $n \in \mathbb{N}$ and let us denote the scalar multiplication by n on E by $[n]$. Namely,

$$[n] : E \to E; \quad P \mapsto [n]P := P + P + \cdots + P. \quad \text{(n times)}$$

This definition extends trivially to all $n \in \mathbb{Z}$, setting $[0]P := \mathcal{O}$ and $[n]P := [-n]([-1]P)$ for $n < 0$. We write $[n]P$ or more simply nP.

2.2.3 TORSION AND CARDINALITY

Definition 2.2.5. Let $E_{/K}$ be an elliptic curve and $n \in \mathbb{Z}$. The kernel of $[n]$, denoted by $E[n]$, satisfies

$$E[n] = \{P \in E(\overline{K}) \mid [n]P = \mathcal{O}\}.$$

An element $P \in E[n]$ is called a n-torsion point.

Theorem 2.2.5. [107, Theorem 3.2] Let E be an elliptic curve over a field K and let n be a positive integer. If $\mathrm{char}(K)$ does not divide n, or is 0, then

$$E[n] \simeq \mathbb{Z}/n\mathbb{Z} \oplus \mathbb{Z}/n\mathbb{Z}.$$

If $\mathrm{char}(K) = p > 0$ and $p|n$, write $n = p^r n'$ with $p \nmid n'$. Then

$$E[n] \simeq \mathbb{Z}/n'\mathbb{Z} \oplus \mathbb{Z}/n'\mathbb{Z} \text{ or } \mathbb{Z}/n\mathbb{Z} \oplus \mathbb{Z}/n'\mathbb{Z}.$$

Let $\mathbb{F}_q$ be a finite field and $\overline{\mathbb{F}_q}$ its algebraic closure. We define the Frobenius map for $\mathbb{F}_q$ by

$$\begin{array}{rccc} \phi_q : & \overline{\mathbb{F}_q} & \to & \overline{\mathbb{F}_q}, \\ & x & \mapsto & x^q. \end{array}$$

Let E be an elliptic curve over $\mathbb{F}_q$. The Frobenius map ϕ_q acts on a point $(x, y) \in E(\overline{\mathbb{F}_q})$ by

$$\phi_q(x, y) \mapsto (x^q, y^q), \phi_q(\mathcal{O}) \mapsto \mathcal{O}.$$

The cardinality of an elliptic curve E over $\mathbb{F}_q$, that is the number of $\mathbb{F}_q$-rational

points, is an important aspect for the security of cryptosystems built on $E(\mathbb{F}_q)$. The theorem of Weil relates the number of points to the field size.

Theorem 2.2.6. Let E be an elliptic curve over $\mathbb{F}_q$. Then

$$|E(\mathbb{F}_q)| = q + 1 - t \text{ and } |t| \leq 2\sqrt{q}.$$

Remark. There are remarks as below:

1. The integer t is called the trace of the Frobenius endomorphism.
2. For any integer $t \in [-2\sqrt{q}, 2\sqrt{q}]$ there is at least one elliptic curve E over $\mathbb{F}_p$ whose cardinality is $p + 1 - t$.

Proposition 2.2.7. Let E be a curve over a field $\mathbb{F}_q$ of characteristic p. The curve E is supersingular if and only if the trace t of the Frobenius satisfies

$$t \equiv 0 \pmod{p}.$$

Note. A curve defined over a prime field $\mathbb{F}_p$ ($p \geq 5$) is supersingular if and only if $|E(\mathbb{F}_p)| = p + 1$. If $ch(K)$ =2 or 3, E is supersingular if and only if its j-invariant is zero.

2.2.4 ALGORITHM OF SCALAR MULTIPLICATION

Scalar point multiplication is the main cryptographic operation in ECC which computes $Q = nP$, a point P is multiplied by an integer n resulting in another point Q on the elliptic curve. Binary method is the traditional scalar multiplication method based on the binary expansion of the scalar $n = \sum_{i=0}^{m-1} n_i 2^i$ where $n_{m-1} = 1$, $n_i \in \{0, 1\}$ for $i = 0, 2, \ldots, m-2$.

Algorithm 1 Left-to-right binary method for point multiplication

Require: Binary representation $n = \sum_{i=0}^{m-1} n_i 2^i$, $n_{m-1} = 1$, $n_i \in \{0, 1\}$ for $i = 0, 2, \ldots, m-2$, and $P \in E(\mathbb{F}_p)$.

Ensure: $nP \in E(\mathbb{F}_p)$

1: $Q \leftarrow P$
2: **for** $i = m - 1$ to 0 **do**
3: $\quad Q \leftarrow 2Q$
4: $\quad$ **if** $n_i = 1$ **then**
5: $\quad\quad Q \leftarrow Q + P$
6: $\quad$ **end if**
7: **end for**
8: **return** Q

Binary Method:

Algorithm 1 is the additive version of the basic left-to-right "square-and-multiply" method for exponentiation, which is called left-to-right "double-and-add" method. This Algorithm 1 requires $m - 1$ point doublings and $w - 1$ point additions where w is Hamming Weight of its binary representation, i.e., the number of nonzero terms in the representation. The expected number of ones in the binary representation of n is approximately $m/2$ point additions (A) and m point doublings (D), denoted $m/2A + mD$.

Example 2.2.1. Let us compute $345P$. One has $(101011001)_2$ and $m = 9$. The next example provides a computation of Algorithm 1.

i	8	7	6	5	4	3	2	1	0
n_i	1	0	1	0	1	1	0	0	1
P		$2P$	$5P$	$10P$	$21P$	$43P$	$86P$	$72P$	$345P$

Signed Binary Method:

For a positive integer n, we can calculate its unique signed binary representation called NAF ([47, §3.3 Algorithm 3.30]).

Definition 2.2.6. A non-adjacent form (NAF) of a positive integer n is an expression $n = \sum_{i=0}^{m-1} n_i 2^i$ where $n_{m-1} = 1$, $n_i \in \{-1, 0, 1\}$ for $i = 0, 2, \ldots, m-2$, and no two consecutive digits n_i are nonzero. The length of the NAF is m.

Theorem 2.2.8. Let n be a positive integer.

- n has a unique NAF denoted NAF(n).
- NAF(n) has the fewest nonzero digits of any signed digit representation of n.
- The length of NAF(n) is at most one more than the length of the binary representation of n.
- If the length of NAF(n) is m, then $2^m/3 < n < 2^{m+1}/3$.
- The average density of nonzero digits among all NAFs of length m is approximately $1/3$.

Let $n = \sum_{i=0}^{m-1} n_i 2^i$ be the NAF of n where m is the length of its signed binary representation. Algorithm 2 modifies the left-to-right binary method for point multiplication (Algorithm 1) by using NAF instead of the binary representation. We see that Algorithm 2 requires $m - 1$ point doublings and $w - 1$ point additions where w is Hamming Weight of its signed binary representation. This means that the expected running time of Algorithm 2 is approximately $m/3A + mD$.

Algorithm 2 Left-to-right signed binary method for point multiplication

Require: Signed binary representation $n = \sum_{i=0}^{m-1} n_i 2^i$, $n_{m-1} = 1$, $n_i \in \{-1, 0, 1\}$ for $i = 0, 2, \ldots, m-2$, and $P \in E(\mathbb{F}_p)$

Ensure: $nP \in E(\mathbb{F}_p)$

1: $Q \leftarrow P$
2: **for** $i = m-1$ to 0 **do**
3: $\quad Q \leftarrow 2Q$
4: $\quad$ **if** $n_i = 1$ **then**
5: $\quad\quad Q \leftarrow Q + P$
6: $\quad$ **end if**
7: $\quad$ **if** $n_i = -1$ **then**
8: $\quad\quad Q \leftarrow Q - P$
9: $\quad$ **end if**
10: **end for**
11: **return** Q

2.3 PAIRINGS

A bilinear pairing suitable for use in cryptography is a non-degenerate, efficient to compute, bilinear map

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$$

where $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are cyclic groups of the same prime order p. The most commonly used pairings arise from the theory of elliptic curves where $\mathbb{G}_1$ and $\mathbb{G}_2$ are subgroups of points on an elliptic curve over a finite field, and $\mathbb{G}_2$ is a subgroup of the multiplicative group of a finite field. Before getting into details, we briefly discuss what is meant by efficient, bilinear and non-

degenerate properties of the map e.

Efficiency is taken to mean that there is a polynomial time algorithm which can compute the map e.

Bilinearity means that the map e is linear in both components which refers to the following two properties.

- $e(P_1 + P_2, Q) = e(P_1, Q)e(P_2, Q)$ for all $P_1, P_2 \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$
- $e(P, Q_1 + Q_2) = e(P, Q_1)e(P, Q_2)$ for all $P \in \mathbb{G}_1$ and $Q_1, Q_2 \in \mathbb{G}_2$

A consequence of these two properties, we obtain

- $e(aP, bQ) = e(bP, aQ) = e(P, Q)^{ab}$ for any integers $a, b \in \mathbb{Z}$.
- $e(P, 0) = e(0, Q) = 1$
- $e(-P, Q) = e(P, -Q) = e(P, Q)^{-1}$

Non-degeneracy means that if $e(P, Q)$ is identity element of G_T, then either P is the identity of $\mathbb{G}_1$ or Q is the identity of $\mathbb{G}_2$.

2.3.1 DIVISORS

Divisors are a crucial part of the pairing on elliptic curves. In this section, we give some results of divisor theory.

Definition 2.3.1. Let E be an elliptic curve over a field K. The divisor class group of E, denoted by $\mathrm{Div}(E)$, is the free abelian group generated by the points of $E(\overline{K})$. Thus any divisor $D \in \mathrm{Div}(E)$ is of the form

$$D = \sum_{p \in E(\overline{K})} n_P(P)$$

where $n_P \in \mathbb{Z}$ and $n_P = 0$ except for finitely many P's.

The degree of a divisor $D \in \mathrm{Div}(E)$ is defined by

$$\deg : \mathrm{Div}(E) \to \mathbb{Z}; D \mapsto \deg(D) := \sum_{p \in E(\overline{K})} n_P.$$

The divisors of degree zero form a subgroup of $\mathrm{Div}(E)$, which we denote by

$$\mathrm{Div}^0(E) = \{D \in \mathrm{Div}(E) \mid \deg(D) = 0\}.$$

Let $G_{\overline{K}/K}$ be a galois group of K. Then $G_{\overline{K}/K}$ act on $\mathrm{Div}(E)$ (and $\mathrm{Div}^0(E)$) in the obvious way,

$$D^\sigma = \sum_{p \in E(\overline{K})} n_P(P^\sigma).$$

Then D is defined over K if $D^\sigma = D$ for all $\sigma \in G_{\overline{K}/K}$. We denote the group of divisors over K by $\mathrm{Div}_K(E)$, and similarly for $\mathrm{Div}^0_K(E)$. Let f be a non-zero rational function on E. Rational function on E can be roughly understood to be the ratio of two polynomials over the E. The divisor of a rational function f on an elliptic curve E, written (f), is represented by

$$(f) = \sum_{P \in E(\overline{K})} \mathrm{ord}_P(f)(P)$$

where $\mathrm{ord}_P(f)$ is the order of the zero/pole that f has at P. A divisor D is said to be principal if $D = (f)$ for a rational function f.

Definition 2.3.2. A divisor $D \in \mathrm{Div}(E)$ is principal if it has the form $D = (f)$ for some $f \in \overline{K}(E)^\times$. Two divisors D_1, D_2 are linearly equivalent, denoted $D_1 \sim D_2$ if $D_1 - D_2$ is principal. The divisor class group (or Picard group) of E, denoted $\mathrm{Pic}(E)$, is the quotient of $\mathrm{Div}(E)$ by the subgroup of principal divisors. We let $\mathrm{Pic}_K(E)$ be the subgroup of $\mathrm{Pic}(E)$ fixed by $G_{\overline{K}/K}$.

The degree zero part of the divisor class group of E, which we denote by $\mathrm{Pic}^0(E)$, is the quotient of $\mathrm{Div}^0(E)$ by the subgroup of principal divisors. Further, $\mathrm{Pic}^0_K(E)$ is the subgroup of $\mathrm{Pic}^0(E)$ fixed by $G_{\overline{K}/K}$.

Proposition 2.3.1. Let E be an elliptic curve and $\mathcal{O} \in E(K)$.

1. For every divisor $D \in \mathrm{Div}^0(E)$, there exists a unique point $P \in E$ so that

$$D \sim (P) - (\mathcal{O}).$$

   Let $\sigma : \mathrm{Div}^0(E) \to E$ be the map given by this association.

2. The map σ is surjective.

3. Let $D_1, D_2 \in \mathrm{Div}^0(E)$. Then

$$\sigma(D_1) = \sigma(D_2) \text{ if and only if } D_1 \sim D_2.$$

   Thus σ induces a bijection of sets (which we also denote by σ)

$$\sigma : \mathrm{Pic}^0(E) \xrightarrow{\sim} E.$$

4. The inverse to σ is the map

$$\tau : E \xrightarrow{\sim} \mathrm{Pic}^0(E); P \mapsto \text{ class of } (P) - (\mathcal{O}).$$

5. If E is given by a Weierstrass equation, then the "geometric group law" on E arising from §2.2.2 and the group law induced from $\mathrm{Pic}^0(E)$ by using σ are the same.

2.3.2 TATE PAIRING

The Tate pairing was introduced by Tate for abelian varieties over local fields [99]. Lichtenbaum gave an interpretation in the case of Jacobians of curves over local fields which permits explicit computation [64]. Frey and Rück considered the Tate pairing over finite fields and introduced it to the cryptographic community [33] [32] [34].

Let E be an elliptic curve over K_0 of characteristic p, and r be a positive integer which is coprime to the characteristic of the field K_0. The set of r-th roots of unity is defined by $\mu_r = \{u \in \overline{K}_0^{\times} \mid u^r = 1\}$. Let $K = K_0(\mu_r)$ be the extension field of K_0 generated by the rth roots of unity. Let

$$E(K)[r] = \{P \in E(K) \mid rP = \mathcal{O}\}$$

and

$$rE(K) = \{rP \mid P \in E(K)\}.$$

Let $P \in E(K)[r], Q \in E(K)/rE(K)$. We denote a divisor $\mathcal{A}_P$ by equivalent to the divisor $(P) - (\mathcal{O})$.

The Tate Pairing over Finite Field:

Let E be an elliptic curve over $\mathbb{F}_q$ and r be coprime to q and $r \mid \#E(\mathbb{F}_q)$. The embedding degree of E with respect to r is defined to be the smallest positive integer k such that $r \mid (q^k - 1)$. Then k is also the least positive integer such that the field $\mathbb{F}_{q^k}$ contains all the rth roots of unity. Let $P \in E(\mathbb{F}_q)[r]$, $Q \in E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})$. The divisor $\mathcal{A}_Q$ is a divisor equivalent to the divisor $(Q) - (\mathcal{O})$. The Tate pairing is defined as follows:

$$T_r : E(\mathbb{F}_q)[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) \to \mathbb{F}_{q^k}/(\mathbb{F}_{q^k})^r$$

is given by

$$T_r(P,Q) := f_{r,P}(\mathcal{A}_Q).$$

Here, $f_{s,P}$ where s is a positive integer is a rational function defined by the following divisor:

$$(f_{s,P}) = s(P) - (sP) - (s-1)(\mathcal{O}).$$

Note that P is from $E(\mathbb{F}_q)$ while Q is from $E(\mathbb{F}_{q^k})$ where $\mathbb{F}_q$ is a finite field and $\mathbb{F}_{q^k}$ is a degree k extension of $\mathbb{F}_q$. Since P is an r-torsion point, it follows that $rP = \mathcal{O}$ and so

$$\begin{aligned}(f_{r,P}) &= r(P) - (rP) - (r-1)(\mathcal{O}) \\ &= r(P) - r(\mathcal{O}).\end{aligned}$$

In this definition, a value of the Tate pairing is an equivalence class in $\mathbb{F}_{q^k}/(\mathbb{F}_{q^k})^r$. For the practical purposes, we would like a unique representative of this class. The natural way to proceed is to raise this value to the power $(q^k-1)/r$. This exponential, called final exponential, kills off all rth powers leaving an exact rth root of unity in $\mathbb{F}_{q^k}$. Hence for the remainder of this chapter, we consider the reduced (normalized) Tate pairing is defined as follows:

$$\widehat{T}_r : E(\mathbb{F}_q)[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) \to \mu_r(\mathbb{F}_{q^k}) \subset \mathbb{F}_{q^k}/(\mathbb{F}_{q^k})^r$$

is given by

$$\widehat{T}_r(P,Q) := f_{r,P}(\mathcal{A}_Q)^{(q^k-1)/r}$$

Weil Pairing:

In this part, we describe the Weil Pairing as given by [85]. Let E be an elliptic curve over K_0 and let r be an integer coprime to the characteristic of K_0. Define $K = K_0(E[r])$ to be the field extension of K_0 generated by the coordinates of all the points in $E(\overline{K})$ of order divisible by r. Let $P, Q \in E[r]$ be points,

$\mathcal{A}_P$ a divisor equivalent to the divisor $(P) - (\mathcal{O})$. Note that $r\mathcal{A}_P$ is a principal divisor, so there exists a function f_P such that $(f_P) = r\mathcal{A}_P$. The Weil pairing is defined as follows:

$$W_r : E[r] \times E[r] \to \mu_r \subseteq K^\times$$

is given by

$$W_r(P, Q) := f_{r,P}(\mathcal{A}_Q)/f_{r,Q}(\mathcal{A}_P).$$

Similarity between the definition of the Weil pairing and the Tate pairing. In the Weil pairing, the term $f_{r,P}(\mathcal{A}_Q)$ is equivalent modulo rth powers to $T_r(P, Q)$ while the term $f_{r,Q}(\mathcal{A}_P)$ is equivalent modulo rth powers to $T_r(Q, P)$. Hence we can write

$$W_r(P, Q) = T_r(P, Q)/T_r(Q, P) \text{ up to } r\text{th powers}.$$

This relation indicates that computation of the Weil pairing takes roughly twice as long as the computation of the Tate pairing. This can be verified by looking at the algorithms for computing the pairings. Considering the algorithm for computing the Weil pairing $W_r(P, Q)$, we see that we need to construct two functions $f_{r,P}$ and $f_{r,Q}$ such that $(f_{r,P}) \sim r(P) - r(\mathcal{O})$ and $(f_{r,Q}) \sim r(Q) - r(\mathcal{O})$. These functions need to be evaluated in divisors $\mathcal{A}_Q \sim (Q) - (\mathcal{O})$ and $\mathcal{A}_P \sim (P) - (\mathcal{O})$, respectively. The Tate pairing $T_r(P, Q)$, on the other hand, only requires a single function $f_{r,P}$. This implies that the computation of the Weil pairing takes at least twice the computation time for the Tate pairing. The possible additional cost for the final exponentiation in the Tate pairing does not offset the difference in computation times. Hence, from a computational point of view, the Tate pairing is superior to the Weil pairing.

2.3.3 MILLER'S ALGORITHM

The Computation of $f_{r,P}$:

The computation of $f_{r,P}$ is using a double-and-add algorithm similar to that of scalar multiplication. Assume that E is given in Weierstrass form. Let P and R be points on E. We define the following rational functions and their divisors.

1. $l_{P,R}(R \neq P)$ is the line passing through P, R and $-(R+P)$.

$$(l_{P,R}) = (P) + (R) + (-(P+R)) - 3(\mathcal{O}).$$

2. $l_{R,R}$ is the line passing through R, $-2R$. We choose the line as the tangent on the curve through the point R.

$$(l_{R,R}) = 2(R) + (-2R) - 3(\mathcal{O}).$$

3. $l_{R,-R}$ is the line passing through R, $-R$ (i.e. a vertical line).

$$(l_{R,-R}) = (R) + (-R) - 2(\mathcal{O}).$$

Using the above notations $l_{*,*}$, we also prepare notations, which are convenient for the remain discussion, as follows:

1. $h_{P,R}(R \neq P)$ is defined to be $h_{P,R} = l_{P,R}/l_{T,-T}$ where $T = R + P$.

$$(h_{P,R}) = (l_{P,R}) - (l_{T,-T}).$$

2. $h_{R,R}$ is defined to be $h_{R,R} = l_{R,R}/l_{T,-T}$ where $T = 2R$.

$$(h_{R,R}) = (l_{R,R}) - (l_{T,-T}).$$

Note that $(f_{1,P}) = (P) - (P) = 0$ and so $f_{1,P} = 1$. A recurrence for $f_{s,P}$ can be obtained as follows.

$$\begin{aligned}
(f_{2m,P}) &= 2m(P) - (2mP) - (2m-1)(\mathcal{O}) \\
&= 2(m(P) - (mP) - (m-1)(\mathcal{O})) + 2(mP) - (2mP) - (\mathcal{O}) \\
&= 2(f_{m,P}) + 2(mP) + (-2mP) - 3(\mathcal{O}) \\
&\quad -((2mP) + (-2mP) - 2(\mathcal{O})) \\
&= 2(f_{m,P}) + (l_{mP,mP}) - (l_{2mP,-2mP}) \\
&= 2(f_{m,P}) + (h_{mP,mP}).
\end{aligned}$$

$$\begin{aligned}
(f_{2m+1,P}) &= (2m+1)(P) - ((2m+1)P) - 2m(\mathcal{O}) \\
&= 2m(P) - (2mP) - (2m-1)(\mathcal{O}) \\
&\quad +(P) + (2mP) - ((2m+1)P) - (\mathcal{O}) \\
&= (f_{2m,P}) + (P) + (2mP) + (-(2m+1)P) - 3(\mathcal{O}) \\
&\quad -(((2m+1)P) + (-(2m+1)P) - 2(\mathcal{O})) \\
&= (f_{2m,P}) + (l_{2mP,P}) - (l_{(2m+1)P,-(2m+1)P}) \\
&= (f_{2m,P}) + (h_{P,2mP}).
\end{aligned}$$

Then we have $(f_{2m,P}) = 2(f_{m,P}) + (h_{mP,mP})$ from which we get

$$f_{2m,P} = f_{m,P}^2 \times h_{mP,mP}.$$

Algorithm 3 Miller's Algotirhm (I)

Require: $P \in E(\mathbb{F}_q)[r], Q, U \in E(\mathbb{F}_{q^k}), r = (1 r_1 r_2 \ldots, r_{n-1})$

Ensure: Reduced Tate pairing $\widehat{T}_r(P, Q)$

1: $f \leftarrow 1$
2: $T \leftarrow P$
3: **for** $i = 1$ to $n - 1$ **do**
4: $\quad f \leftarrow f^2 \cdot h_{T,T}(Q + U)/h_{T,T}(Q)$
5: $\quad T \leftarrow 2T$
6: $\quad$ **if** $r_i = 1$ **then**
7: $\quad\quad f \leftarrow f^2 \cdot h_{T,P}(Q + U)/h_{T,P}(Q)$
8: $\quad\quad T \leftarrow T + P$
9: $\quad$ **end if**
10: **end for**
11: **return** $f^{(q^k-1)/r}$

Similarly $(f_{2m+1,P}) = 2(f_{m,P}) + (h_{P,2mP})$ whows

$$f_{2m+1,P} = f_{2m,P} \times h_{P,2mP}.$$

Computing Tate pairing reduces to the following task.

We show an algorithm to calculate Reduced Tate Pairing at Algorithm 3 [73, 74]. Let k be the embedding degree. In the algorithm, line 4 is calculated by the tangent line at T and the vertical line through between T and $-T$. Moreover, line 7 is calculated by the line through T and P, and the vertical at $T + P$.

This computation is the called Miller's Algorithm. The final T obtained after the full iteration of the loop is raised to the power $(q^k - 1)/r$ to get a unique element in μ_r. This is the final exponentiation part in the Tate pair-

ing. There are numerous enhancements to Miller's algorithm. Unfortunately, most of them apply only to small subsets of elliptic curves, for example to supersingular elliptic curves. Supersingular elliptic curves have rich properties, and easy to use. In fact, BF-IBE is restricted to use the pairing constructed by a family of supersinbular elliptic curves (over a prime field). We explain the definition of supersingular elliptic curves in the next section.

2.4 PAIRINGS OVER SUPSERSINGULAR ELLIPTIC CURVES

2.4.1 SUPERSINGULAR ELLIPTIC CURVES (E_{ss})

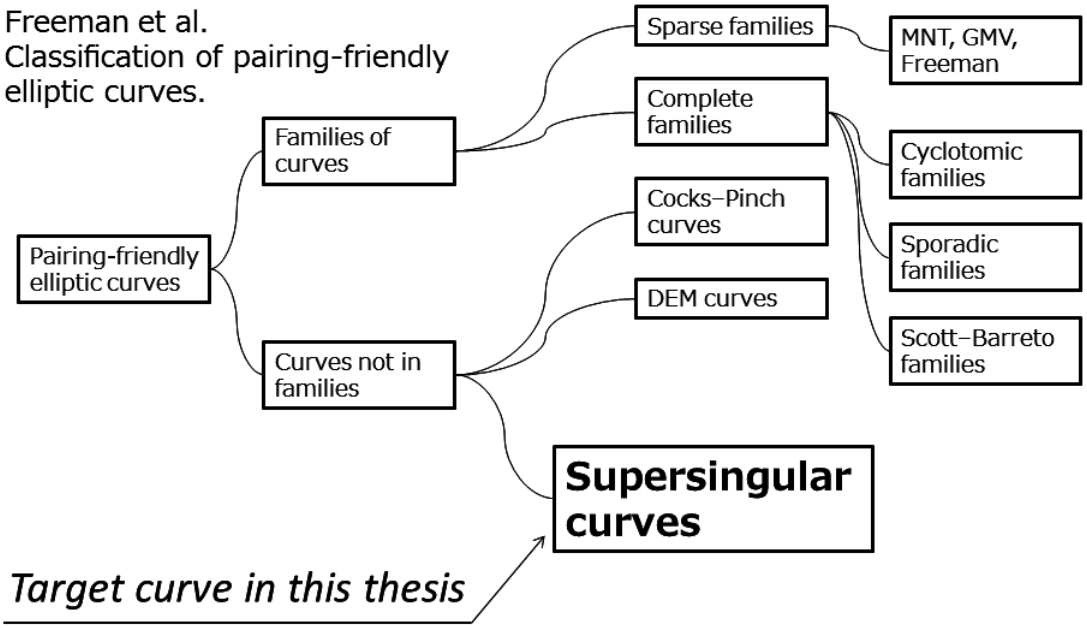
There is much research on the generation of suitable elliptic curves for pairings, namely pairing-friendly curves, which contain the large prime subgroup and the small embedding degree. Refer to the in-depth overview [31] for details. Here, we describe supersingular elliptic curves and their use for pairing-based cryptosystems.

Definition 2.4.1. Let $\mathrm{char}(K) = p$ and let E be an elliptic curve over K. If $E[p^r] = \{\mathcal{O}\}$ for one and in fact for all positive integers r, then the curve is called supersingular. Otherwise the curve is called ordinary.

Note. [107, §3.1] An elliptic curve E whose definition field has characteristic p is called ordinary if $E[p] \simeq \mathbb{Z}/p\mathbb{Z}$. It is called supersingular if $E[p] = \{\mathcal{O}\}$. Note that the terms "supersingular" and "singular" are unrelated. In the complex multiplication, the "supersingular" j-invariants are those corresponding to elliptic curves with the largest possible endomorphism rings, namely, orders in quaternion algebras. The "singular" means that j-invariants are those corresponding to elliptic curves with endomorphism rings larger than $\mathbb{Z}$.

The most important property is that the trace $t = Tr(\phi_q)$ of the Frobenius endomorphism satisfies $t \equiv 0 \pmod{p}$. By the Theorem of Hasse-Weil we have $|t| \leq 2\sqrt{q}$. Hence, over prime fields $\mathbb{F}_p$ with $p \geq 5$, the condition implies $t = 0$ and the cardinality of $E(\mathbb{F}_p)$ satisfies $\#E(\mathbb{F}_p) = p+1$. As r divides $\#E(\mathbb{F}_p) = p+1$, which in turn divides (p^2-1), we see that for supersingular curves over large prime fields the embedding degree is bounded by 2. Menezes, Okamoto, and

Figure 2.1: Pairing-friendly elliptic curves by Freeman et al.



Vanstone prove that the embedding degree for supersingular elliptic curves is always less than or equal to 6 [70]. In fact, we can even say more. It is proved that the upper bound on the embedding degree depends on the characteristic of the base field:

Proposition 2.4.1. [24, Proposition 6.20] Let E be a supersingular elliptic curve over $\mathbb{F}_q$ with $q = p^d$. Assume that E has a $\mathbb{F}_q$-rational point of order r. Let k be the smallest natural number such that $r \mid (q^k - 1)$. Then

- characteristic 2 we have $k \leq 4$,
- characteristic 3 we have $k \leq 6$,
- over prime fields $\mathbb{F}_p$ with $p \geq 5$ we have $k \leq 2$,

and these bounds are attained.

This means that for supersingular elliptic curves over large prime fields

we must work in a larger field than usual for curve based cryptography. For current security parameters we should choose a ground field with around 2^{512} elements to obtain a satisfying level of security, since the finite field $\mathbb{F}_{q^k}$ needs to have at least 2^{1024} elements to adhere to the proposals. But this is to the detriment of efficiency, since we have to compute on some larger field for the same level of security (note that the current proposals assume elliptic curves with 2^{160} elements to offer sufficient security). Therefore, it is preferable to work in characteristic 2 or, even better, in characteristic 3 and on curves with the maximal possible k.

2.4.2 DISTORTION MAP

The typical case in cryptographic applications if $P \in E(\mathbb{F}_q)$ and $k > 1$. In this condition, a valuable technique introduced by Verheul[104], which applies to both Tate and Weil pairings, is to use a non-rational endomorphism.

For a supersingular elliptic curve, there is at least one nice endomorphism, which maps a point from $E(\mathbb{F}_q)$ to $E(\mathbb{F}_{q^k})$ called distortion map. Indeed, the following theorem proves that distortion maps always exist.

Theorem 2.4.2. [105, Theorem 5] Let E be a supersingular elliptic curve over a finite field $\mathbb{F}_q$ with $q = p^d$ and k an embedding degree. Let $K' = \mathbb{F}_{q^k}$ and P be a point on E over K of prime order r relatively prime to p, then $\mathrm{End}_{K'}(E[r])$ is isomorphic to the ring $M_2(\mathbb{Z}/r\mathbb{Z})$ of all 2×2 matrices over $\mathbb{Z}/r\mathbb{Z}$. In particular there is an abundance of distortions maps (defined over K') with respect to P.

Table 2.2: Supersingular Elliptic Curve and Distortion Map

Type	Elliptic Curve Data
Type-1	$E_{\mathbb{F}_p} : y^2 = x^3 + a$ where $p \equiv 2 \pmod 3$ $\#E(\mathbb{F}_p) = p + 1$ Distortion map $(x, y) \mapsto (\zeta_3 x, y)$ where $\zeta_3^3 = 1$.
Type-2	$E_{\mathbb{F}_p} : y^2 = x^3 + x$ where $p \equiv 3 \pmod 4$ $\#E(\mathbb{F}_p) = p + 1$ Distortion map $(x, y) \mapsto (-x, iy)$ where $i^2 = -1$.

Embedding degree $k = 2$:

Supersingular elliptic curves are suitable for pairing-based cryptosystems since it is possible to have $k = 2, 3, 4,$ and 6. Table 2.2 shows the most popular supersingular elliptic curves and its concrete distortion maps where the embedding degree $k = 2$.

2.4.3 TATE PARING FOR E_{ss}

The main drawback of the Weil pairing is that $W_r(P, P)$ is always equal to 1. Hence, the pairing is degenerate if applied to the cyclic subgroup of order r in both arguments. This is not the case with the Tate pairing. Frey, Müller, and Rück state in [32] that if r^2 does not divide the cardinality of the elliptic curve over $\mathbb{F}_{q^k}$, the Tate pairing applied to a point with itself yields a primitive r-th root of unity. Otherwise, $r^2 \mid \#E(\mathbb{F}_{p^k})$, a modification is required to make the

pairing nontrivial on the cyclic subgroup generated by $P \in E(\mathbb{F}_q)$.

A mapping that is nondegenerate and bilinear and is also efficiently computable is called a pairing, and such mappings are the fundamental primitives from which many cryptographic algorithms are constructed. On the other hand, the Tate pairing also has the following property that limits its usefulness because it returns the value 1 in many cases.

Proposition 2.4.3. [35, Galbraith] Let $P \in E(\mathbb{F}_q)[r] \setminus \mathcal{O}$ and r relatively prime to q. Then to have $T_r(P, P) \neq 1$, we must have $k = 1$.

So for an embedding degree $k > 1$ we have $T_r(P, P) = 1$, which also means that $T_r(aP, bP) = T_r(P, P)^{ab} = 1$ for integers a and b, so that the Tate pairing may not seem very useful at first. The following result provides insight into how to overcome this limitation.

Proposition 2.4.4. [104, Verheul] Let r be a prime, $P \in E(\mathbb{F}_q)[r] \setminus \mathcal{O}$, $Q \in E(\mathbb{F}_{q^k})$ be linearly independent from P, and $k > 1$. Then we have that $T_r(P, P)$ is nondegenerate.

So if we have $P \in E(\mathbb{F}_q)[r]$ and a nontrivial embedding degree, that is, we have $k > 1$, then one way to make sure that the Tate pairing $T_r(P, Q)$ is nondegenerate is to make sure that Q is linearly independent of P. One way to do this is to use a distortion map. We compute

$$\widehat{T}_r^{\mathrm{mod}} : E(\mathbb{F}_q)[r] \times E(\mathbb{F}_q)[r] \to \mu_r(\mathbb{F}_{q^k}) \subset \mathbb{F}_{q^k}/(\mathbb{F}_{q^k})^r.$$

is given by

$$\widehat{T}_r^{\mathrm{mod}}(P, Q) := \widehat{T}_r(P, \phi(Q))$$

where $\phi : E(\mathbb{F}_q) \to E(\mathbb{F}_{q^k})$ is an appropriate distortion map. We call such an $\widehat{T}_r^{\mathrm{mod}}$ the modified Tate pairing.

Algorithm 4 Miller's Algotirhm (II)

Require: $P, Q \in E(\mathbb{F}_q)[r]$ where $k > 1$, $r = (1 r_1 r_2 \ldots, r_{n-1})$.

Ensure: Modified Tate pairing $\widehat{T}_r^{\mathrm{mod}}(P, \phi(Q))$

1: $f \leftarrow 1$

2: $T \leftarrow P$

3: $Q \leftarrow \phi(Q)$

4: **for** $i = 1$ to $n - 1$ **do**

5: $\quad f \leftarrow f^2 \cdot l_{T,T}(Q)$

6: $\quad T \leftarrow 2T$

7: $\quad$ **if** $r_i = 1$ **then**

8: $\quad\quad f \leftarrow f^2 \cdot l_{T,P}(Q)$

9: $\quad\quad T \leftarrow T + P$

10: $\quad$ **end if**

11: **end for**

12: **return** $f^{(q^k-1)/r}$

2.4.4 MILLER'S ALGORITHM FOR E_{ss}

Using the Modified Tate Paring, we can get a variant algorithm of the previous Miller's Algorithm as Algorithm 4.

Note. Notice the $\phi(Q) \in E(\mathbb{F}_{q^k}) \setminus E(\mathbb{F}_q)$, the denominator $l_{T,-T}$ of $h_{T,T}, h_{T,P}$ (line 4,7 in Algorithm 3) are eliminated by the final exponential $(q^k - 1)/r$.

CHAPTER 3

BONEH-FRANKLIN IDENTITY BASED ENCRYPTION (BF-IBE)

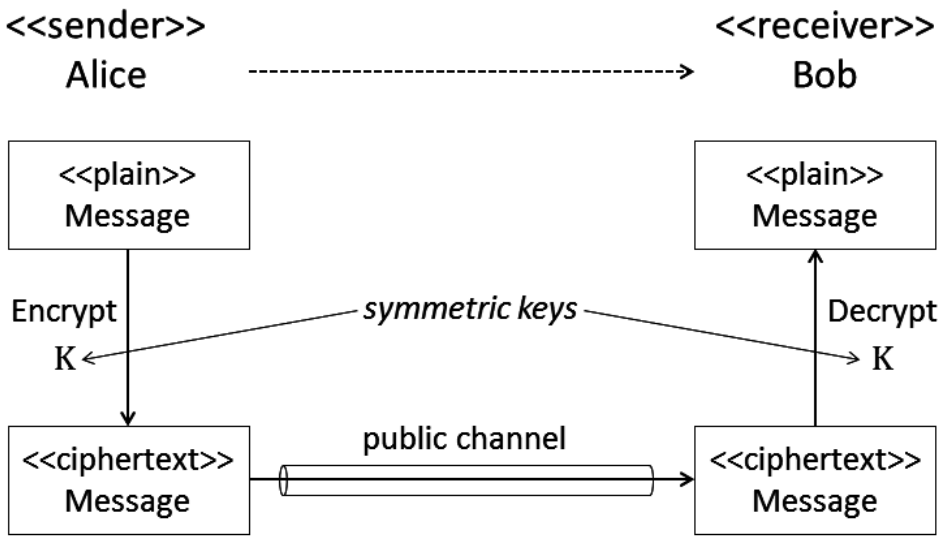
3.1 BACKGROUND OF CRYPTOGRAPHY

The modern field of cryptography can be divided into two areas, the symmetric key encryption (SKE) and public key encryption (PKE).

3.1.1 SYMMETRIC KEY ENCRYPTION

SKE refers to encryption methods in which both the sender and receiver share the same key (secret key K, Figure. 3.1). SKE was the only kind of encryption publicly known until 1976. In SKE, the ciphertext moving across the public channel is a function of the message and the secret key K. An adversary has access to the ciphertext, but without knowledge of K should be unable to obtain the intended message M. The receiver knows K and should be able to recover M from the ciphertext. SEC use the same key for encryption and decryption of a message, though a message may have a different key than others. Thus, arose the problem of ensuring secure communication between

Figure 3.1: Symmetric Key Encription

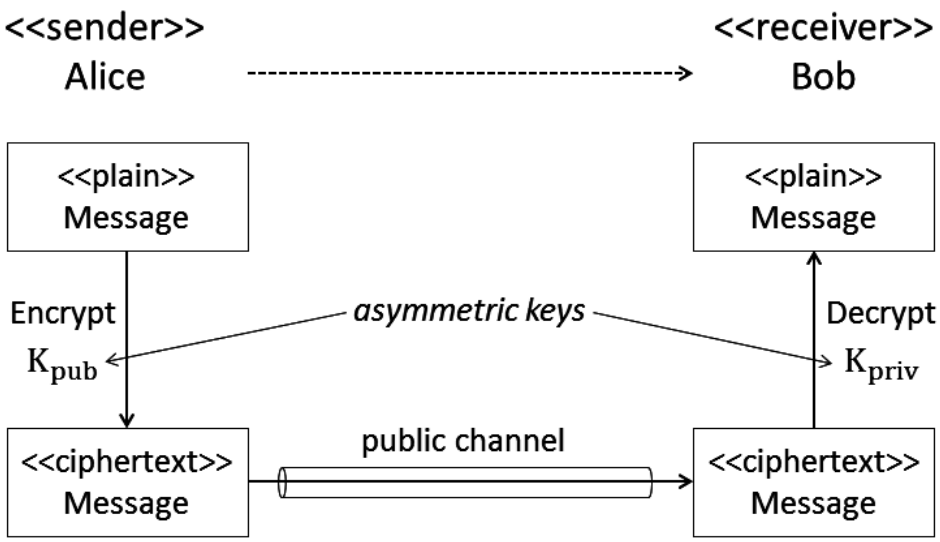


any two of a number of parties. Suppose there are n parties. Using the SKC, secure communication between any two parties requires a secret key per pair of parties. So the total number of secret keys in the system is $\binom{n}{2} = n(n-1)/2$ and each party has to maintain $n-1$ secret keys.

3.1.2 PUBLIC KEY ENCRYPTION

PKE is designed to overcome the problem. The basic idea was simple. Instead of using the same key for encryption and decryption, one may consider two separate keys for each party. The encryption key may be made public, so that any other party (Alice) may send an encrypted message. On the other hand, the decryption key should be kept secret, so that only the intended receiver (Bob) can decrypt the ciphertext. It was first published by Diffie and Hellman in their seminal paper [26] titled “New Directions in Cryptography”. Though

Figure 3.2: Public Key Encription



the concept of PKE was introduced by Diffie and Hellman, they were unable to provide a concrete instantiation of such a scheme.

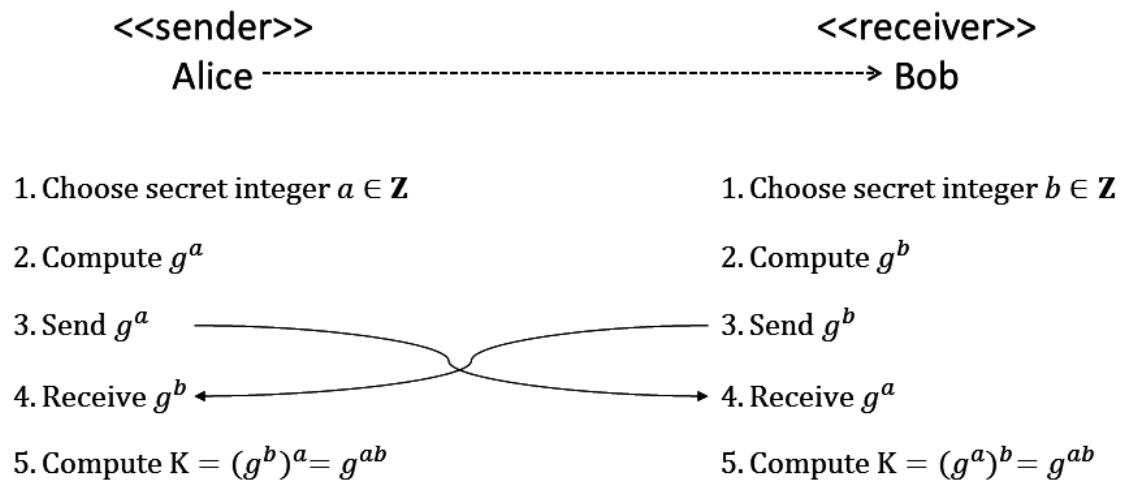
RSA Public Key Encryption:

It was left as an open problem until it was solved by three other researchers, Rivest, Shamir and Adleman [88]. This was called RSA public key encryption.

Diffie Hellman Key Agreement:

Diffie and Hellman had introduced and solved another related and equally important problem. They considered the possibility of two parties performing some private computations and exchanging some message over a public channel to finally arrive at a shared secret key. This is called Diffie-Hellman key agreement (DH-KA).

Figure 3.3: Diffie Hellman Key Agreement



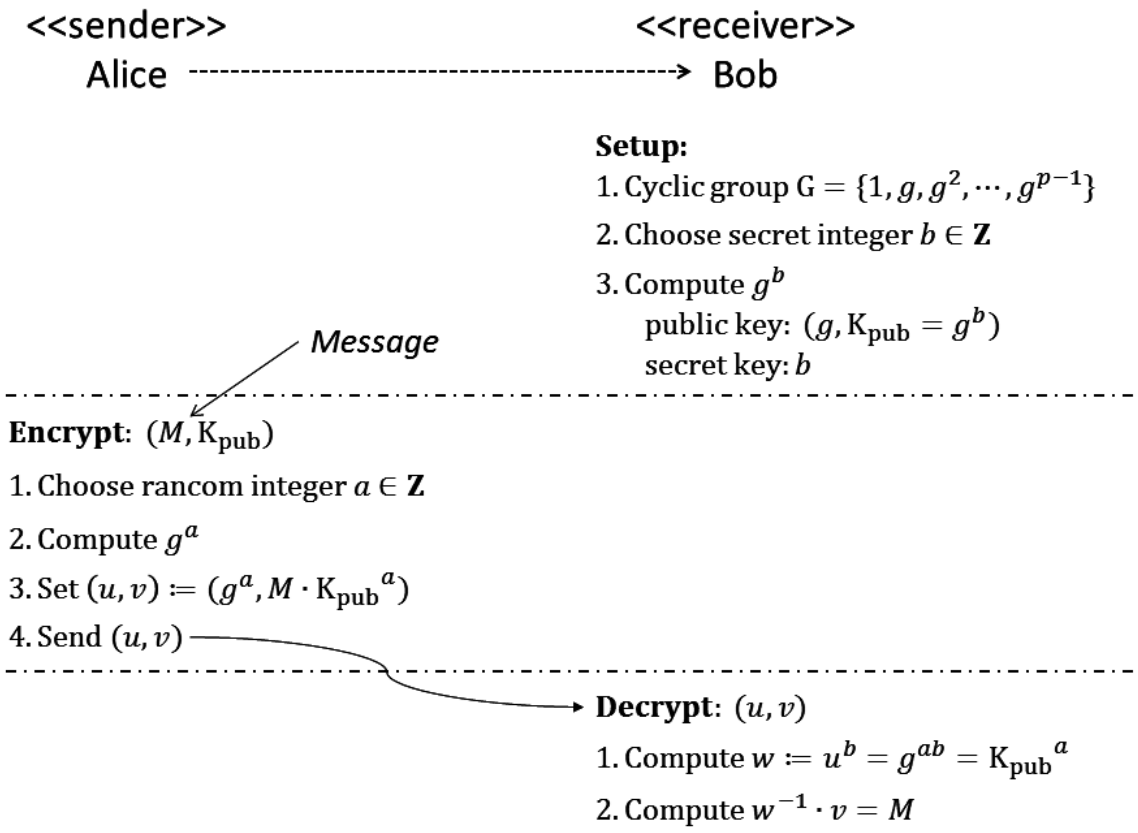
ElGamal Public Key Encryption:

Later a public key encryption scheme was developed by ElGamal [28] which is based on the Diffie-Hellman key agreement (DH-KA).

3.1.3 Digital Signatures

Digital signature is a mechanism by which a message is authenticated. In this primitive, each user has a secret signing key and a public verification key. For Example, suppose that Alice wants to digitally sign a message to Bob. To do so, she uses her private key to encrypt the message, she then sends the message along with her public key. Since Alice's public key is the only key that can decrypt that message, a successful decryption constitutes a Digital Signature Verification, meaning that there is no doubt that it is Alice's private key that encrypted the message. Concrete proposals of signature schemes

Figure 3.4: ElGamal Public Key Encryption

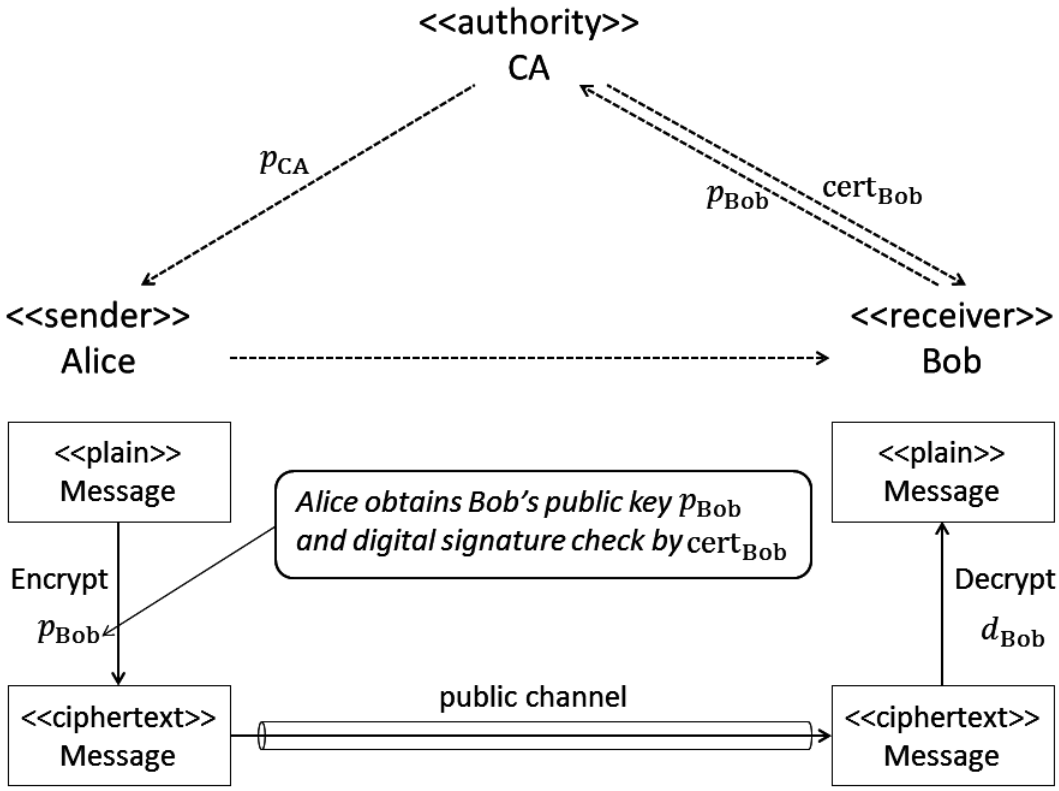


were made using the RSA and the ElGamal schemes.

3.1.4 Public-Key Infrastructure (PKI)

The main concern in a PKC is the authenticity of the public key. If a malevolent can convince other participants that Bob's public key is some key of his choice instead of the Bob's public-key, he can decrypt messages intended for Bob only and forge signatures under Bob's name. This type of threat is known as man-in-the-middle-attack. Therefore, it is importance that participants in

Figure 3.5: Certifying Authority System



a PKC system can verify the authenticity of other user's public keys. The conventional solution to the authentication problem is the use of a public-key infrastructure (PKI). A PKI often works with a party trusted by all users, called Certification Authority (CA), which can guarantee the correctness of the public keys. For the detail of PKI, refer to the book [101].

3.1.5 History of Identity-Based Encryption (IBE)

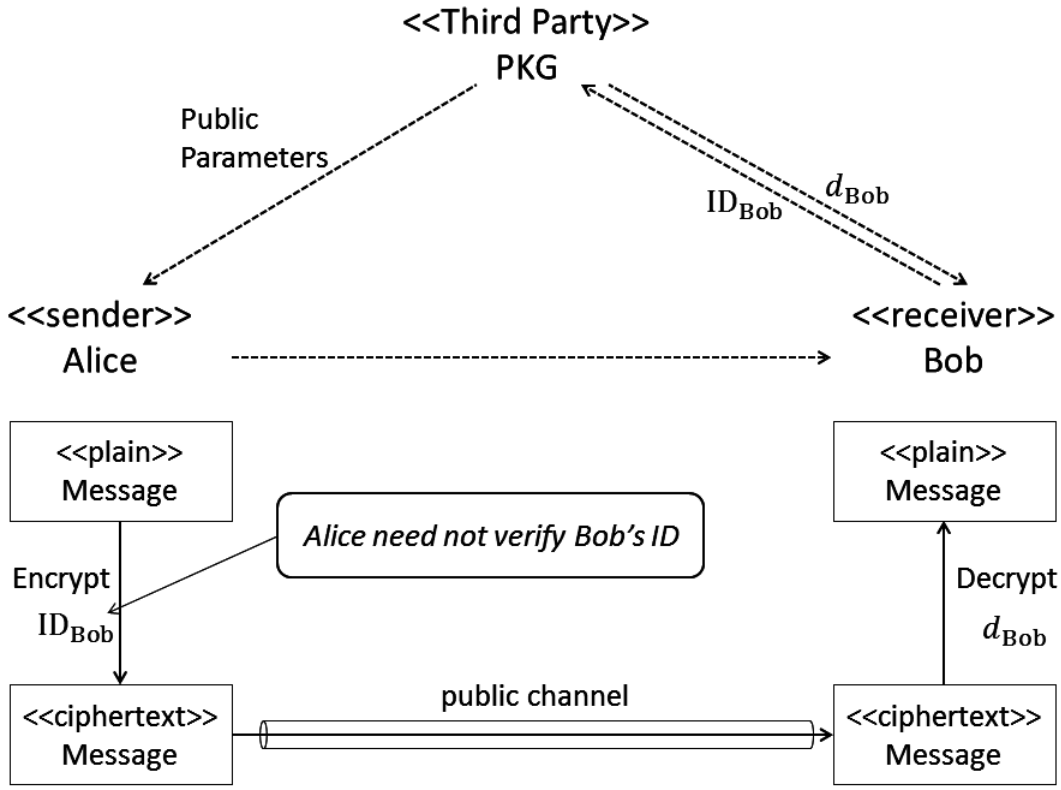
In 1984, Shamir [74] invented the concept of Identity-Based Cryptography, which addresses the authenticity problem of public keys in a different way

(see also [56]). His idea was to avoid the need for authentication altogether, by making sure that the actual value of a user's public key is inherently linked to his identity. More precisely, the public key of a user is derived directly from publicly available information that uniquely and undeniably identifies that user. This information is denoted by a user's (digital) identity. Depending on the application, the identity can range from (combinations of) the user's name, social security number, phone number, email address, and possibly other personal information. In this setting, a user's public key is readily available to anyone who knows his identity, so there is no need to look up the key in some database. Moreover, the fact that there is no doubt about the authenticity of the public key takes away the need for certificates as in a PKI setting. We should note, however, that the realization of the link between users and their digital identities is far from trivial.

3.2 BF-IBE Protocol

The practical encryption scheme of the identity based cryptosystems was proposed by Sakai, Ogishi, Kasahara by using bilinear pairing over elliptic curves [89]. Boneh and Franklin also suggested the identity based cryptosystem and its concrete implementation [16]. Here, we present the Boneh-Franklin Identity-Based Encryption scheme, denoted BF-IBE. They give two versions of IBE, BasicIdent and FullIdent. BasicIndentis developed and shown to be secure in the sense of IND-ID-CPA (indistinguishability under adaptive identity and adaptive Chosen Plaintext Attack) and FullIdent is shown to provide IND-ID-CCA (indistinguishability under adaptive identity and adaptive Chosen Ciphertext Attack). For the detail of IND-ID-CPA and IND-IP-CCA, see [22, §2.3 Security Model for (H)IBE].

Figure 3.6: IBE System



We describe BasicIdent, which has four algorithms, Setup, Extract, Encrypt, and Decrypt as followings:

Setup:

This algorithm is run by the PKG one time for creating the whole IBE environment. The master key is kept secret and used to derive users' private keys, while the system parameters are made public. Given a security parameter $k \in \mathbb{Z}$. The setup phase for BF-IBE executes the following steps.

Setup (1)	Run a randomized algorithm on input k to generate a prime r, tow groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order r generated by random generators $P \in \mathbb{G}_1$, and $g \in \mathbb{G}_2$, and an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.
Setup (2)	Pick a random $s \in (\mathbb{Z}/r\mathbb{Z})^\times$ and set $P_{\text{pub}} \leftarrow sP$.
Setup (3)	Choose cryptographic hash functions $H_1 : \{0,1\}^* \to \mathbb{G}_1^\times$ and $H_2 : \mathbb{G}_2^\times \to \{0,1\}^n$ for some positive integer n.
Setup (4)	Set message space $\mathcal{M} = \{0,1\}^*$ and ciphertext space $\mathcal{C} = \mathbb{G}_1^\times \times \{0,1\}^n$. Then publish parameter = $\{r, \mathbb{G}_1, \mathbb{G}_2, e, n, P, P_{\text{pub}}, H_1, H_2\}$ as system parameters, and conceal $s \in (\mathbb{Z}/r\mathbb{Z})^\times$ as a master key which has a public key generator.

Extract:

This algorithm is run by the PKG when a user requests his private key. Note that the verification of the authenticity of the requestor and the secure transport of d_{ID} are problems with which IBE protocols do not try to deal.

Extract (1)	For a given user identity $\text{ID} \in \{0,1\}^*$, compute $Q_{\text{ID}} \leftarrow H_1(\text{ID}) \in \mathbb{G}_2^\times$.
Extract (2)	Set the private key $d_{\text{ID}} \leftarrow sQ_{\text{ID}}$ where s is the master key.

Encrypt:

Let $M \in \mathcal{M}$ be the message. There are five steps as follows:

Encrypt (1)	Generate a random $r \in \mathbb{Z}/r\mathbb{Z}$ and compute rP.
Encrypt (2)	Calculate $Q_{\mathrm{ID}} = H_1(\mathrm{ID})$ from the recipient's identity $\mathrm{ID} \in \{0,1\}^*$.
Encrypt (3)	Calculate rQ_{ID}.
Encrypt (4)	Calculate pairing $g_{\mathrm{ID}} := \widehat{T}_r^{\mathrm{mod}}(rQ_{\mathrm{ID}}, P_{\mathrm{pub}})$.
Encrypt (5)	Let C be $M \oplus H_2(g_{\mathrm{ID}})$ and return (rP, C) where the symbol $\oplus$ means bitwise XOR.

Decrypt:

Let (rP, C) be the received message, the recipient can decrypt it by the following three steps.

Decrypt (1)	Extract private key $d_{\mathrm{ID}} := sQ_{\mathrm{ID}}$ by Extract (1) if necessary.
Decrypt (2)	Calculate pairing $g'_{\mathrm{ID}} := \widehat{T}_r^{\mathrm{mod}}(rP, d_{\mathrm{ID}})$
Decrypt (3)	Extract message by $M = C_2 \oplus H_2(g'_{\mathrm{ID}})$

Table 3.1: System parameters of the IBE

Notation	Comments
n	Positive integer, length of plaintext (in bits)
s	Integer in $\mathbb{Z}/r\mathbb{Z}$, master secret
$\mathbb{G}_1$	$E(\mathbb{F}_p)[r] = \langle P \rangle$, a cyclic subgroup with order r with its generator P
$\mathbb{G}_2$	$\langle \widehat{T}_r^{\mathrm{mod}}(P,P) \rangle$ where P is a generator of $\mathbb{G}_1$, and the extension field over $\mathbb{F}_p$ with degree 2
$\widehat{T}_r^{\mathrm{mod}}$	$\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, Modified Tate Pairing
P_{pub}	$P_{\mathrm{pub}} := sP \in E(\mathbb{F}_p)$, master public-key
H_1	$H_1 : \{0,1\}^n \to \mathbb{G}_1$, HashToPoint (see §3.3).
H_2	$H_2 : \mathbb{G}_2 \to \{0,1\}^n$, cryptographic hash function

3.3 Algorithm for BF-IBE

To implement BF-IBE, we first need a security parameter that defines the level of bit strength that the encryption will provide. Then we need to define groups $\mathbb{G}_1$ and $\mathbb{G}_2$ and a pairing $\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. To do this, we pick an elliptic curve E over $\mathbb{F}_q$ with embedding degree k, and a prime r with $r \mid \#E(\mathbb{F}_q)$. We also require that $r^2 \nmid \#E(\mathbb{F}_q)$ to ensure that the subgroup of order r that we will hash identities into is unique. The parameter r is the order of the both groups $\mathbb{G}_1$ and $\mathbb{G}_2$, and $\mathbb{G}_2$ is a subgroup of $\mathbb{F}_{q^k}^{\times}$. To attain a particular level of security, these parameters need to be chosen as described in §3.4. We then randomly pick a point $P \in E(\mathbb{F}_q)[r]$ and let $\mathbb{G}_1 := \langle P \rangle$ and $\mathbb{G}_2 := \langle \widehat{T}_r(P,P) \rangle$,

Algorithm 5 Find a point to a given x

Require: An interger x and a elliptic curve $E(\mathbb{F}_p)$.

Ensure: A point $P \in E(\mathbb{F}_p)$.

1: $y \leftarrow -1$.
2: **while** $y = -1$ **do**
3: $y_2 \leftarrow x^3 + x \pmod{p}$.
4: **if** y_2 is a square in $\mathbb{F}_p$ **then**
5: $y \leftarrow y_2$
6: **else**
7: $x \leftarrow x + 1 \pmod{p}$
8: **end if**
9: **end while**
10: **return** $P = (x, y)$

which are cyclic groups of prime order r. In this thesis, we use a supersingular elliptic curve E of the form $y^2 = x^3 + x$ over prime field $\mathbb{F}_p$ where $p \geq 5$ and $p \equiv 3 \pmod{4}$. So we use Modified Tate pairing $\widehat{T}_r^{\mathrm{mod}}$.

We explain how to compute HashToPoint explicitly. First, we use Algorithm 6 for finding r-torsion points from $x \in \mathbb{F}_p$. Let E be a supersingular elliptic curve in Weierstrass form, $y^2 = x^3 + x$. First we define $h_1 : \{0,1\}^n \to E(\mathbb{F}_p)$ as follows. For any $\mathrm{ID} \in \{0,1\}^n$, which is a bit string of n bits, we can embed ID into the x-coordinate of a point $Q = (x, y) \in E(\mathbb{F}_p)$ as an integer modulo r. Then we calculate a y-coordinate of Q by $y = (x^3 + x)^{1/2}$. From Euler's theorem, we have that $a^{p-1} \equiv 1 \pmod{p}$ so that $a^{p-1}a^2 = a^{p+1} \equiv a^2 \pmod{p}$. Then $a^{(p+1)/4} \equiv a^{1/2} \pmod{p}$ whenever we have that $4 \mid (p+1)$. Note that if $x^3 + x$ is not quadratic residue in $\mathbb{F}_p$, we increment x by $x + 1$ (Algorithm 5).

Next, we define $h_2 : E(\mathbb{F}_p) \to E(\mathbb{F}_p)[r]$ as follows. We know that $\#E(\mathbb{F}_p) = p+1 = rc$, so we get $Q_{\mathrm{ID}} := (p+1)/rQ = cQ \in E(\mathbb{F}_p)[r]$. HashToPoint is defined by

Algorithm 6 HashToPoint

Require: A string ID $=: x \in \{0,1\}^*$, $\mathbb{F}_p$-rational points $E(\mathbb{F}_p)$ with $\#E(\mathbb{F}_p) = n = rc$.

Ensure: A point $Q_{\mathrm{ID}} \in E(\mathbb{F}_p)[r]$ corresponding to the string ID.

1: $c \leftarrow n/r \pmod p$.
2: $x \leftarrow h_1(\mathrm{ID})$.
3: Let $P_{\mathrm{ID}} \in E(\mathbb{F}_p)$ be the result from algorithm 5.
4: $P' \leftarrow cP$
5: **while** $P' = \mathcal{O}$ **do**
6: $x \leftarrow x$ coordinate of P
7: $x \leftarrow x + 1 \pmod p$
8: Let $P_{\mathrm{ID}} \in E(\mathbb{F}_p)$ be the result from algorithm 5 with arguments x and $E(\mathbb{F}_p)$.
9: $P' \leftarrow cP$
10: **end while**
11: **if** $rP' \neq \mathcal{O}$ **then**
12: **return** "Wrong group order, no r-torsion point found"
13: **else**
14: **return** P'
15: **end if**

the composition $h_2 \circ h_1$. Focus on h_2, we can use Algorithm 1, 2 for computing the scalar multiplication cQ.

If we choose the cofactor c with low Hamming Weight, then the computational time of cQ becomes faster. We discuss the existence of such cofactor c with low Hamming Weight in the following section.

Table 3.2: A comparison of public-key cryptosystems [102, Table 3]

Public key systems	Example	Mathematical Problem	Best known method for solving math problem (running time)
Integer factorization	RSA, Rabin-Williams	Given a number n, find its prime factors	Number field sieve: exp[$1.923(\log n)^{1/3}(\log\log n)^{2/3}$] (Sub-exponential)
Discrete logarithm	Diffie-Hellman(DH), DSA, ElGamal	Given a prime n, and numbers g and h, find x such that $h = g^x \pmod n$	Number field sieve: exp[$1.923(\log n)^{1/3}(\log\log n)^{2/3}$] (Sub-exponential)
Elliptic curve discrete logarithm	ECDH, ECDSA	Given an elliptic curve E and points P and Q on E, find x such that $Q = xP$	Pollard-rho algorithm: $\sqrt{n}$ (Fully exponential)

3.4 SECURE PARAMETER SIZE

At the foundation of every public-key cryptosystem is a hard mathematical problem that is computationally intractable. The relative difficulty of solving that problem determines the security strength of the corresponding system. Table 3.2 summarizes three types of well known public-key cryptosystems. As shown in the last column, RSA, Diffie-Hellman and DSA can all be attacked using sub-exponential algorithms, but the best known attack on ECC requires exponential time. For this reason, ECC can offer equivalent security

Table 3.3: Prospects of key sizes

Algorithm security lifetimes	SKE	RSA/DH	ECC(ECDH)
Through 2010	80	1024	160
Through 2030	112	2048	224
Beyond 2030	128	3072	256
—	192	7680	384
—	256	15360	512

with substantially smaller key sizes [63]. Public-key schemes are typically used to transport or exchange keys for symmetric-key ciphers. Since the security of a system is only as good as that of its weakest component, the work factor needed to break a symmetric key must match that needed to break the public-key system used for key exchange. Table 3.3 shows NIST guidelines [80] on choosing computationally equivalent symmetric and public key sizes. As shown in Table 3.3, ECC-160 provides the same security as RSA-1024 and ECC-224 matches RSA-2048. NIST provides detailed information required to implement the cryptographic algorithms, including various types of encryption keys, their use, and required key lengths (Table 3.3). As shown in Table 3.3, NIST describes their recommendation of cryptographic algorithms and key lengths by separating cases into those for which they may be used up to the end of 2010, up to the end of 2030, and after 2030.

CHAPTER 4

SPEEDING UP HASHTOPOINT

In this chapter, we focus on BF-IBE which is defined by RFC5091 [19]. RFC5091 is restricted to use the pairing constructed by a family of supersingular elliptic curves over finite fields of large prime characteristic. For a prime $p \geq 5$, there are two type of supersingular elliptic curves represented by the following in Table 2.2 or [31, §3]:

(Type-1) $y^2 = x^3 + 1$ where $p \equiv 2 \pmod 3$
(Type-2) $y^2 = x^3 + x$ where $p \equiv 3 \pmod 4$.

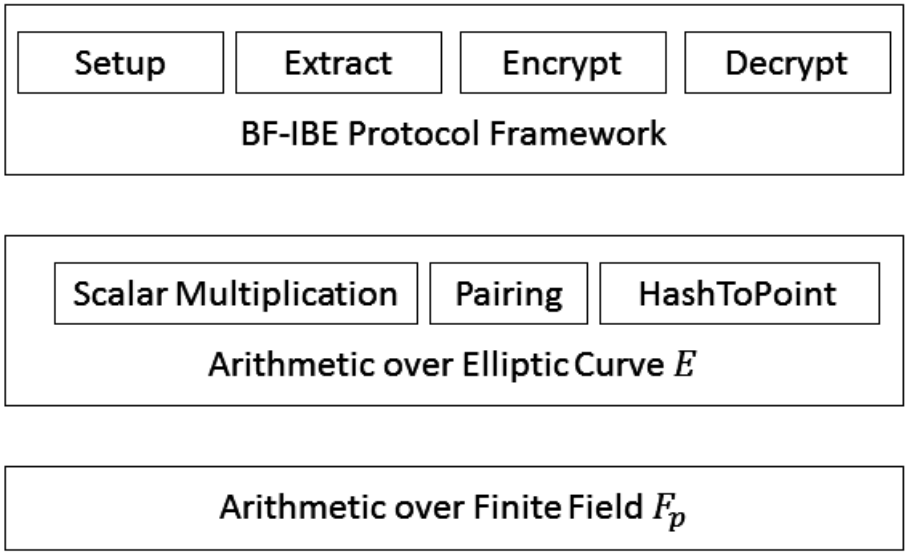
In RFC5091 is adapted (Type-1). However we substitute the curve (Type-2) with curve (Type-1) because the library (PBC) which we use to check the IBE performance only supports the curve (Type-2). The differences between (Type-1) and (Type-2) do not affect the efficiency of the IBE discussed in this paper. We review some notations and summarize basic properties of supersingular elliptic curve here. Let $\widehat{T}_r^{\mathrm{mod}}(P,Q) = \widehat{T}_r(P,\phi(Q))$ be the modified Tate pairing, where $P,Q \in E(\mathbb{F}_p)[r]$ are points of order r on the supersingular elliptic curve where $r \mid \#E(\mathbb{F}_p) = p+1$. Note that this particular curve has the nice property that for any y a unique point (x,y) can be found on the curve, which makes mapping arbitrary values to curve points particularly easy. The func-

tion $\phi(\cdot)$ is a distortion map, an automorphism which maps a point on $E(\mathbb{F}_p)$ to a linearly independent point on $E(\mathbb{F}_{p^2})$. For (Type-2) supersingular elliptic curve an appropriate distortion map is $\phi(x, y) = (-x, iy)$, where $i \in E(\mathbb{F}_{p^2})$ is a square root of unity mod p (Table 2.2). We choose three security types —160 bit, 224 bit, and 256 bit— for r which refers to $\#E(\mathbb{F}_p)[r]$. Recall that these three types are recommended by NIST as enough secure key size for ECC (Table 3.3). In this chapter, we propose efficient parameters for faster computation of HashToPoit—as the result faster computation of BF-IBE— in Table 4.3. First, we explain the arithmetic functions over a elliptic curve which are required by BF-IBE (§4.1) and the profiling result of BF-IBE. In this section, we can find that speeding up HashToPoint is efficient for the faster computation of BF-IBE. For the speeding up HashToPoint, we need suitable parameters called cofactor. Next, we explain how to find such parameters— the cofactor with low Hamming Weight—using Algorithm 7. If we use the cofactor low Hamming Weight, then the computation speed of HashToPoint becomes faster (§3.3). We use PARI/GP to execute the algorithm and we find explicit script in §A.2.1. Next, we list our searched result and propose suitable parameters in each security level. Finally, we estimate a existence probability of such parameters using prime number theory (§4.3).

4.1 HashToPoint and Cofactors

The BF-IBE consists of four steps: setup, extract, encryption, and decryption. These four steps are essentially constructed by arithmetic functions over elliptic curve $E(\mathbb{F}_p)$and finite field $\mathbb{F}_p$. For example, scalar multiplication, pairing, and HashToPoint are defined over a elliptic curve E. Figure 4.1 describes the layer structure of functions used in the IBE and Table 4.1

Figure 4.1: Layer Structure of BF-IBE



shows the relation between BF-IBE protocol framework and arithmetic functions over elliptic curve. In addition, Table 4.1 shows that BF-IBE requires three arithmetic functions over a elliptic curve E, scalar multiplication, pairing, and HashToPoint. For example, Encrypt (1) in Encryption phase require a scalar multiplication. In the similar way, the other procedures, Encrypt (2)–(4) in Encryption phase and Extract (1)–(2), Decrypt (2) in Decryption phase, require each arithmetic functions over a elliptic curve E.

To start our investigation, we measured the execution time of each procedures in Table 4.1. Table 4.2 shows the profiling result of BF-IBE. By the profiling result, we can find that HashToPoint, which correspond to Encrypt (1) or Extract (1), is the dominant part of the other procedures. Then the speeding up HashToPoint turn out to be efficient. Next, we explain the idea of speeding up HashToPoint.

Table 4.1: Relation between BF-IBE protocol and the arithmetic functions

Procedures in BF-IBE					Arithmetic functions
Encryption:	Encrypt	(1)	rP	$\leftrightarrow$	Scalar Multiplication
	Encrypt	(2)	$Q_{\mathrm{ID}} := H_1(\mathrm{ID})$	$\leftrightarrow$	HashToPoint
	Encrypt	(3)	rQ_{ID}	$\leftrightarrow$	Scalar Multiplication
	Encrypt	(4)	$g_{\mathrm{ID}} := \widehat{T}_r^{\mathrm{mod}}(rQ_{\mathrm{ID}}, P_{\mathrm{pub}})$	$\leftrightarrow$	Modified Tate Pairing
Decryption:	Extract	(1)	$Q_{\mathrm{ID}} := H_1(\mathrm{ID})$	$\leftrightarrow$	HashToPoint
	Extract	(2)	sQ_{ID}	$\leftrightarrow$	Scalar Multiplication
	Decrypt	(2)	$g'_{\mathrm{ID}} := \widehat{T}_r^{\mathrm{mod}}(rP, d_{\mathrm{ID}})$	$\leftrightarrow$	Modified Tate Pairing

The idea of faster computation:

Let us recall the procedures in BF-IBE (§3.2). When encrypting to an identity, there is a requirement to compute $Q_{\mathrm{ID}} = H_1(\mathrm{ID}) \in \mathbb{G}_1^{\times}$ and then $g_{\mathrm{ID}} = \widehat{T}_r^{\mathrm{mod}}(rQ_{\mathrm{ID}}, P_{\mathrm{pub}})$, where Q_{ID} and P_{pub} are points on an elliptic curve of order r, and H_1 is HashToPoint. HashToPoint is a hash-and-map function that must first hash the input identity to a point on the curve, and then map it to a point of order r for our supersingular curve over the prime field $\mathbb{F}_p$ ($p \geq 5$). From the viewpoint of security, NIST recommends keys of size at least 80 (the size of the key space here is 2^{80}, which is a lot of brute force work for an attacker)—this condition means p is 512 bit and r is 160 bit. For the detail of relation between key size and security level, see §3.4. Under this condition, the mapping requires a point multiplication by a 352 (=512-160) bit cofactor. This cost

Table 4.2: Performance Profile

Procedures in BF-IBE				Execution Time	Occupation
Enctyption:	Encrypt	(1)	rP	1.89	11.6
	Encrypt	(2)	Q_{ID}	**4.27**	**26.2**
	Encrypt	(3)	rQ_{ID}	1.87	11.5
	Encrypt	(4)	g_{ID}	1.06	6.5
Decryption:	Extract	(1)	Q_{ID}	**4.28**	**26.3**
	Extract	(2)	sQ_{ID}	1.87	11.5
	Decrypt	(2)	g'_{ID}	1.06	6.5

is likely to dwarf the cost of calculating the pairing. The main idea of faster computation is that we choose such a cofactor with low Hamming Weight to speed up the scalar multiplication, which saves extra additional operation of points on an elliptic curve.

4.2 PROPOSED COFACTORS

4.2.1 SEARCHING ALGORITHM OF COFACTORS

Algorithm 7 show the searching method of cofactor with Hamming Weight of less than tree. First, we explain how to find cofactors with Hamming Weight of less than three in the case of p is 512 bit, and r is 160 bit. Recall that RFC 5091 chooses prime r as Solinas primes of the form $2^{159} \pm 2^t \pm 1$ for $t = 1, 2, ..., 158$. There are ten Solinas prime which are listed in the second column of Table 4.3. To find cofactors with Hamming Weight of less than three, we use two type cofactor in the form $c = 2^{352} \pm k2^i$ for $i = 1, 2, \ldots, 351$ and $k \in \{-1, 0, 1\}$. We have the properties $p = rc - 1$, then we try to find prime p where $p \pmod 4 \equiv 3$ We can apply the above scheme to the remain cases; $(\ell(p), \ell(r)) = (1024, 224), (1536, 256)$, similarly.

4.2.2 LIST OF COFACTORS

Table 4.3 lists the complete list of cofactor with Hamming Weight of less than three. The firs column denotes by its counter. The (1-$*$) show Solinas primes under the level of RSA-1024 (ECC-160). The (2-$*$) and (3-$*$) show Solinas primes under the level of RSA-2048 (ECC-224) and RSA-3072 (ECC-256) respectively. The second column is the list of explicit formula of Solinas primes r with length 160 bit, 224 bit, and 256 bit. The third column is the cofactors with Hamming Weight of less than three. These cofactors are results of our exhaustive search using Algorithm 7.

In the remains, we pick up some parameters in each three security level using Table 4.3.

Table 4.3: Cofactor with Hamming Weight less than three

#	Solinas prime r $(2^a \pm 2^t \pm 1)$	proposed cofactor c $(2^u \pm 2^v)$
p : 512 bit r : 160 bit $a = 159$		
(1-1)	$2^{159} + 2^{17} + 1$	NA
(1-2)	$2^{159} + 2^{19} + 1$	$2^{352} - 2^{150}, 2^{352} - 2^{198}, 2^{352} - 2^{208}$
(1-3)	$2^{159} + 2^{59} + 1$	$2^{352} + 2^{127}, 2^{352} - 2^{134}$
(1-4)	$2^{159} + 2^{63} + 1$	$2^{352} - 2^{18}, 2^{352} - 2^{24}, 2^{352} - 2^{88}, 2^{352} - 2^{108}$
(1-5)	$2^{159} + 2^{88} - 1$	$2^{352} - 2^{24}, 2^{352} - 2^{176}$
(1-6)	$2^{159} + 2^{107} + 1$	$2^{352} - 2^{12}, 2^{352} - 2^{156}$
(1-7)	$2^{159} + 2^{110} - 1$	$2^{352} + 2^{33}, 2^{352} - 2^{162}$
(1-8)	$2^{159} + 2^{116} - 1$	$2^{352} + 2^{19}, 2^{352} - 2^{246}, 2^{352} + 2^{335}$
(1-9)	$2^{159} + 2^{135} + 1$	$2^{352} + 2^{31}$
(1-10)	$2^{159} + 2^{138} - 1$	$2^{352} + 2^{13}, 2^{352} + 2^{89}, 2^{352} + 2^{269}, 2^{352} + 2^{321}$
p : 1024 bit, r : 224 bit, $a = 223$		
(2-1)	$2^{223} + 2^{8} - 1$	$2^{800} + 2^{261}, 2^{800} + 2^{741}$
(2-2)	$2^{223} + 2^{10} - 1$	$2^{800} - 2^{80}, 2^{800} + 2^{193}, 2^{800} - 2^{212}, 2^{800} + 2^{475}, 2^{800} - 2^{578}$
(2-3)	$2^{223} + 2^{13} + 1$	$2^{800} - 2^{4}, 2^{800} - 2^{34}, 2^{800} - 2^{206}, 2^{800} - 2^{230}$
(2-4)	$2^{223} + 2^{30} - 1$	$2^{800} + 2^{5}, 2^{800} - 2^{92}$
(2-5)	$2^{223} + 2^{55} + 1$	NA
(2-6)	$2^{223} + 2^{80} - 1$	$2^{800} + 2^{317}$
(2-7)	$2^{223} + 2^{139} + 1$	$2^{800} - 2^{358}, 2^{800} - 2^{490}, 2^{800} - 2^{622}$
(2-8)	$2^{223} + 2^{153} + 1$	$2^{800} + 2^{395}, 2^{800} + 2^{771}$
p : 1536 bit, r : 256 bit, $a = 255$		
(3-1)	$2^{255} + 2^{41} + 1$	$2^{1280} + 2^{173}, 2^{1280} + 2^{633}, 2^{1280} + 2^{753}, 2^{1280} - 2^{1026}$
(3-2)	$2^{255} + 2^{96} - 1$	$2^{1280} + 2^{1225}$
(3-3)	$2^{255} + 2^{166} + 1$	$2^{1280} + 2^{110}, 2^{1280} + 2^{413}, 2^{1280} - 2^{863}, 2^{1280} + 2^{938}, 2^{1280} - 2^{1073}$
(3-4)	$2^{255} + 2^{176} - 1$	$2^{1280} + 2^{43}, 2^{1280} + 2^{893}, 2^{1280} + 2^{1039}$
(3-5)	$2^{255} + 2^{227} + 1$	$2^{1280} + 2^{311}, 2^{1280} - 2^{506}, 2^{1280} - 2^{780}, 2^{1280} - 2^{970}$
(3-6)	$2^{255} + 2^{232} - 1$	$2^{1280} + 2^{109}, 2^{1280} + 2^{693}, 2^{1280} + 2^{853}$
(3-7)	$2^{255} + 2^{243} + 1$	$2^{1280} + 2^{215}, 2^{1280} - 2^{458}, 2^{1280} - 2^{1090}$

Algorithm 7 Searching for cofactor with Hamming Weight of less than three

Require: Let $a = \ell(p)$ be a positive integer that is the length of p, r be a Solinas prime

Ensure: The set of cofactors $C = \{c_1, c_2, \ldots, c_N\}$ where Hamming Weight of each c_i is less than three

1: Let C be the empty set. i.e. $C := \{\}$
2: $u \leftarrow a - \ell(r)$ where $\ell(r)$ is the length of r
3: **for** $i = 1$ to $u - 1$ **do**
4: **for** $k \in \{-1, 0, 1\}$ **do**
5: $c \leftarrow 2^u + k2^i$
6: $p' \leftarrow cr - 1$
7: **if** $\ell(p')$ is equal to a, $p' \equiv 3 \pmod 4$, and p' is a prime **then**
8: c puts C
9: **end if**
10: **end for**
11: **end for**
12: Return (C)

4.2.3 Examples of Proposed Cofactors

Example 1 (p: 512bit, r: 160bit):

- $E_{/\mathbb{F}_p} : y^2 = x^3 + x$
- $r = 2^{159} + 2^{135} + 1$
- $c = 2^{352} + 2^{31}$
- $p = 2^{511} + 2^{487} + 2^{352} + 2^{190} + 2^{166} + 2^{31} + 1$
- Hamming Weight of (c, p) is $(2, 132)$.

Hexadecimal Notation:

r = 0x 80000000 00000000 00000000 08000000 00000001
c = 0x 00000001 00000000 00000000 00000000 00000000
00000000 00000000 00000000 80000000 00000000
00000000 00000000
p = 0x 80000000 00000000 00000000 08000000 00000001
00000000 00000000 40000000 00000000 00000000
04000000 00000000 7fffffff ffffffff ffffffff
ffffffff

Example 2 (p: 1024bit, r: 224bit):

- $E_{/\mathbb{F}_p} : y^2 = x^3 + x$
- $r = 2^{223} + 2^{153} + 1$
- $c = 2^{800} + 2^{395}$
- $p = 2^{1023} + 2^{953} + 2^{800} + 2^{618} + 2^{548} + 2^{395} + 1$
- Hamming Weight of (c, p) is $(2, 400)$.

Hexadecimal Notation:

r = 0x 80000000 00000000 02000000 00000000 00000000
00000000 00000001
c = 0x 00000001 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000800 00000000
00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
00000000
p = 0x 80000000 00000000 02000000 00000000 00000000
00000000 00000001 00000000 00000000 00000000
00000000 00000000 00000400 00000000 00000010
00000000 00000000 00000000 00000000 000007ff
ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff

Example 3 (p: 1536bit, r: 256bit):

- $E_{/\mathbb{F}_p} : y^2 = x^3 + x$
- $r = 2^{255} + 2^{41} + 1$
- $c = 2^{1280} + 2^{173}$
- $p = 2^{1535} + 2^{1321} + 2^{1280} + 2^{428} + 2^{214} + 2^{173} + 1$
- Hamming Weight of (c, p) is $(2, 178)$.

Hexadecimal Notation:

r = 0x 80000000 00000000 00000000 00000000 00000000
00000000 00000200 00000001

c = 0x 00000010 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 0000000
00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
00002000 00000000 00000000 00000000 00000000
00000000

p = 0x 80000000 00000000 00000000 00000000 00000000
00000000 00000200 00000001 00000000 00000000
00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00001000
00000000 00000000 00000000 00000000 00000000
00000000 00400000 00001fff ffffffff ffffffff
ffffffff ffffffff

4.3 DISTRIBUTION OF COFACTORS

Here we estimate the number of cofactor with Hamming Weight of less than three using the prime number theorem. Let $\pi_{a,n}(x)$ be the number of primes in the arithmetic progression $\{a, a+n, a+2n, \dots\}$ less than x, where a and n are some positive integers. The prime number theorem for arithmetic progressions states that $\phi(a)^{-1}\,\mathrm{Li}(x)$ is an approximation to $\pi_{a,n}(x)$, where $\phi(x)$ is the Euler's totient function and $\mathrm{Li}(x)$ is logarithmic integral defined by $\int_2^x (1/\log x)dx$ [108].

In the case of $(\ell(p), \ell(r))$ =(512 bits, 160 bits), the number of primes p where $p = 2^{511} \pm 2^{a_1} \pm 2^{a_2} \cdots$ and $p \equiv 3 \pmod 4$ is nearly equal $\phi(4)^{-1}\,(\mathrm{Li}(2^{512}) - \mathrm{Li}(2^{511}))$. For each $r = 2^{159} + 2^a \pm 1$ where $a \in \{17, 19, 59, 63, 88, 107, 110, 116, 135, 138\}$, we try to find the cofactor c that has the form $c = 2^{352} \pm 2^x (1 \le x \le 351)$. Therefore the total number of cofactor is estimated by

$$\frac{\phi(4)^{-1}\Big(\mathrm{Li}(2^{512}) - \mathrm{Li}(2^{511})\Big)}{2^{(512-1)}} \times 10 \times (351 \times 2) = 9.89.$$

In the same way, we get

$$\frac{\phi(4)^{-1}\Big(\mathrm{Li}(2^{1024}) - \mathrm{Li}(2^{1023})\Big)}{2^{(1024-1)}} \times 8 \times (799 \times 2) = 9.01.$$

$$\frac{\phi(4)^{-1}\Big(\mathrm{Li}(2^{1536}) - \mathrm{Li}(2^{1535})\Big)}{2^{(1536-1)}} \times 7 \times (1279 \times 2) = 8.41.$$

The total number of cofactor found in our experiment in Table 4.3 is 23, 19, and 23 which are the same in the order of 9.89, 9.01, and 8.41 respectively.

CHAPTER 5

IMPLEMENTATION OF BF-IBE USING PROPOSED COFACTORS

In this chapter, we give a timing results to evaluate the efficiency of our proposed cofactors. We compare the timing of BF-IBE using proposed cofactors in each three security level; one is our proposed cofactor and the other is a random cofactor. In §3.2, we showed the scheme of BF-IBE and compare the timing with each procedures in the scheme here—the procedures are Encrypt (1)–(4) and Decrypt (1)–(2). The results of the comparison are summarized in Table 5.1–5.3.

5.1 MACHINE ENVIRONMENT AND LIBRARIES

All tests were running on a desktop PC (Mac mini) with an Intel Core i7 2.6 MHz processor (including four core) and 16 GBytes RAM using Mac OS X 10.9.1 (Mavericks). In C, we measure the running time using GCC 4.4.4 compiler with the -O3 and -fomit-frame-pointer options. The program is implemented without assembly and SSE implementation. For more information, see the followings:

- CPU: Intel Core i7, 2.6 GHz, 1 processor, 4 cores.
  - L1 cache: 64 KByte (32 KByte Instruction/Data) per core
  - L2 cache: 256 KByte per core
  - L3 cache: 6 MByte
- RAM: 16 GByte
- Operating System: Mac OS X 10.9.1 (Mavericks) (13B42)
  - kernel version: Darwin 13.0.0

To implement the IBE algorithm, we write programs in ANCI-C using GNU GCC compiler without specific optimizations. C is a general-purpose programming language, and it is generally used for implementation of program and system. By using C compiler, a very high-speed program is generated from a source code, since we can optimize the implementation such as memory management, and the C compiler outputs the optimized native code for a target platform. Thus it is one of the fastest programming languages. In the efficient implementation of BF-IBE, C and C++, that is the extension of C, are mainly used now. We deploy the Pairing Based Crypto (PBC) library developed at Stanford University by Benn Lynn [66]. In Appendix (§A.1), we provide complete source code.

PBC Library:

The PBC library is a free portable C library allowing the rapid prototyping of pairing-based cryptosystems. It provides an abstract interface to a cyclic group with a bilinear pairing, insulating the programmer from mathematical details. The PBC library is built on top of the GMP library, and the PBC API is strongly influenced by the GMP API. Accordingly, this manual tries to imitate the look and feel of the GMP manual. For more details of the PBC and GMP libraries, see the followings:

PBC library: `ttp://crypto.stanford.edu/pbc/`

GMP library: `http://www.swox.com/gmp/`

PBC Library (Type-A curve):

Type A pairings are constructed on the curve $y^2 = x^3 + x$ over the prime field $\mathbb{F}_p$ for some prime $p = 3 \pmod 4$. The $\mathbb{G}_1$ is the group of points $E(\mathbb{F}_p)$. It turns out $\#E(\mathbb{F}_p) = p + 1$ and $\#E(\mathbb{F}_{p^2}) = (p + 1)^2$. Thus the embedding degree k is 2, and hence $\mathbb{G}_2$ is a subgroup of $\mathbb{F}_{p^2}$. The order r is some prime factor of $p + 1$. Write $p + 1 = r * c$. For efficiency, r is picked to be a Solinas prime, that is, r has the form $2^a \pm 2^b \pm 1$ for some integers $0 < b < a$. Also, we choose $p = -1 \pmod{12}$ so F_{p^2} can be implemented as $F_p[i]$ (where $i = \sqrt{-1}$).

Note. In PBC Library used a slightly different notation. q defined the order of the prime field $\mathbb{F}_q$ and h is the cofactor.

PBC Library (Parameter Struct Fields):

```
exp2, exp1, sign1, sign0, r:
  r = 2^exp2 + sign1 * 2^exp1 + sign0 * 1 (Solinas prime)
q, h:
  r*h=q+1
  q is a prime, h is a multiple of 12 (thus q = -1 mod 12)
```

For the explicit examples, see Appendix A.1.

5.2 OUR PARAMETERS

We compare the timing of BF-IBE using proposed cofactors and random cofactors. For the proposed cofactors, we use the three examples in §4.2.3.

Let the Solinas prime r be $2^{159} + 2^{135} + 1$. Using PBC library, we can get a random cofactor c' where $p'(= rc' - 1)$ is a 512 bit prime. For the explicit value, see the Example 1–3. The prime c' has Hamming Weight 173. In general, the average value of Hamming Weight is equal to half of the bit length; in this case, the half of the bit length is 176 (= 352/2). So our random cofactor c' and p' are thoroughly general for a fair comparison. We compute encryption and decryption which consist of procedures Encrypt (1)-(4) and Decrypt (1)-(2) in Table 5.1– 5.3. Here, Decrypt (1) is decomposed into Extract (1)–(2). Next we choose the following proposed cofactor $c = 2^{352} + 2^{31}$ from Table 4.3 at (1-9). We see that c is represented by

c = 0x 00000001 00000000 00000000 00000000 00000000
00000000 00000000 00000000 80000000 00000000
00000000 00000000,

then the corresponding prime p with Hamming Weight 132 is as follows:

p = 0x 80000000 00000000 00000000 08000000 00000001
00000000 00000000 40000000 00000000 00000000
04000000 00000000 7fffffff ffffffff ffffffff
ffffffff.

In the remain case of $(\ell(p), \ell(r))$ =(1024 bit, 224 bit), (1536bit, 256 bit), we can test it in the same way. Finally, we note that the timings in Table 5.1–5.3 are the average values of 1,000 random functions. Next, we list the random parameters correspond to each security level; security level means that $(\ell(p), \ell(r))$ is equal to (512 bit, 160 bit), (1024 bit, 224 bit), and (1536bit, 256 bit).

Example 1 (p: 512bit, r: 160bit):

Hamming Weight: $(c', p') = (173, 270)$

c' = 0x e10a80c1 c717acae 119024df af9e5d42 065d56ce
b5a3d350 645c0be5 2bc533eb 0146f8f9 84bf6a41
b7d46c7c

p' = 0x 70854060 e38bd657 08c8126f ded782a7 116768cd
ac68eb90 f6c2a58a b7a5a98b a5eef859 4bdfec4e
baec332d bc6643ac f7eb2f3d 0f059c5d 64bf6a41
b7d46c7b

Example 2 (p: 1024bit, r: 224bit):

Hamming Weight: $(c', p') = (397, 524)$

c' = 0x 3e999d38 0fdf8d6a 7f7bd745 05b5fc81 b4a5dfb7
a809fbb0 d5e8d9f0 f2812252 e2ff7418 1f365c45
1361f9ec 9d8fd065 4a507244 31425883 e0d7a602
0e180932 4a475bf4 c7d72470 91a6bf62 53cfccd1
8e148173 995ec079 f91034b7 675c626a c0c773a4

p' = 0x 1f4cce9c 07efc6b5 403b1edc f2fabd5b af51e78a
5e1069d1 6e5db8b8 272a4258 e30b192a 70fc07ac
352cf860 33ac34a2 d758f8c6 c7c525d3 ad819638
726dfd62 4c1bb98b 7b69bc37 4af7bece 5dc806f5
d96fe6bc 722ea5d8 adbc4c90 e5284aab 1c2cfeac
00757827 29515bb8 d6148173 995ec079 f91034b7
675c626a c0c773a3

Example 3 (p: 1536bit, r: 256bit):

Hamming Weight: $(c', p') = (644, 779)$

c' = 0x 77e17fcf 3e549611 2689973f bb9838bc 982c0ff2
ad0a7c98 8f870912 a6700d8d e67bc33a 209b3062
3191f23e 5cf318ef d07e71e6 2c355789 8e6bbeaa
640447fb 73a00958 0c1248e4 9af06156 e7c06d6c
1b5b3189 a7e0ca45 1f33d527 1deda61e afd559c9
73c9cf39 5f127e9c a2b77d84 0cd846f3 10599ff5
d2400143 4f78ca37 514cddd4 6a47d81e e0749b46
97ea737c b8e446aa 783dad06 8f59a93d 5eb45d14

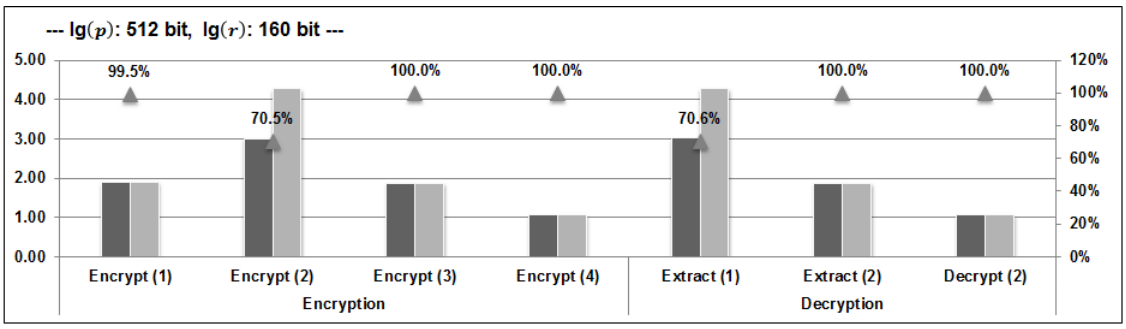
p = 0x 3bf0bfe7 9f2a4b08 9344cb9f ddcc1c5e 4c1607f9
56853e4c 47c38579 1637a544 144b83b9 61d0adb9
6fc4098f 4231aa8e 95647a04 d1374daa 36d80434
cff8a5cc d6ac8c49 4a88d18e 653c028a cdb71bfe
48db1dc7 d7a31174 2695a025 330dcb22 f01c7f72
a6b9de50 cb5478db ef7f3f7d e35bdf41 97b7e87b
e3a01328 705d9e21 ec1a3b71 cdeaf48e 3e47d459
9f3a9d63 1c8a55ec cc7afd18 0b814484 988ea195
e0fd1b95 537e654e b55b94b8 6077c86d 343e53c9
2b9027c3 f813d13d 5eb45d13

5.3 TIMING RESULTS

The results of the comparison are summarized in Table 5.1–5.3. Each Table consists of execution time and occupation. Execution time shows the running time for each procedures in BF-IBE for proposed cofactor c and random one c'. The unit is milliseconds. We can evaluate the improvement seeing the ratio c/c'. In the tables, we emphasize Encrypt (2) and Extract (1) because these procedures are HashToPoint on which we focus. Occupation shows the ratio of each occupation in the corresponding procedures in BF-IBE. We can see how much HashToPoint occupation decrease by our proposed cofactors. Each tables has the visualization of the data. Accordingly, figures help us to recognize the improvement of HashToPoint easily. Moreover, we can see that our cofactors do not affect the other procedures.

Table 5.1: Timing result (p :512bit, r :160bit)

Procedure in BF-IBE (§3.2)				Execution Time			Occupation	
				c	c'	c/c'	c	c'
Enctyption:	Encrypt	(1)	rP	1.88	1.89	99.5	13.7	11.6
	Encrypt	(2)	Q_{ID}	**3.01**	**4.27**	**70.5**	**21.9**	**26.2**
	Encrypt	(3)	rQ_{ID}	1.87	1.87	100	13.6	11.5
	Encrypt	(4)	g_{ID}	1.06	1.06	100	7.7	6.5
Decryption:	Extract	(1)	Q_{ID}	**3.02**	**4.28**	**70.6**	**21.9**	**26.3**
	Extract	(2)	sQ_{ID}	1.87	1.87	100	13.6	11.5
	Decrypt	(2)	g'_{ID}	1.06	1.06	100	7.7	6.5



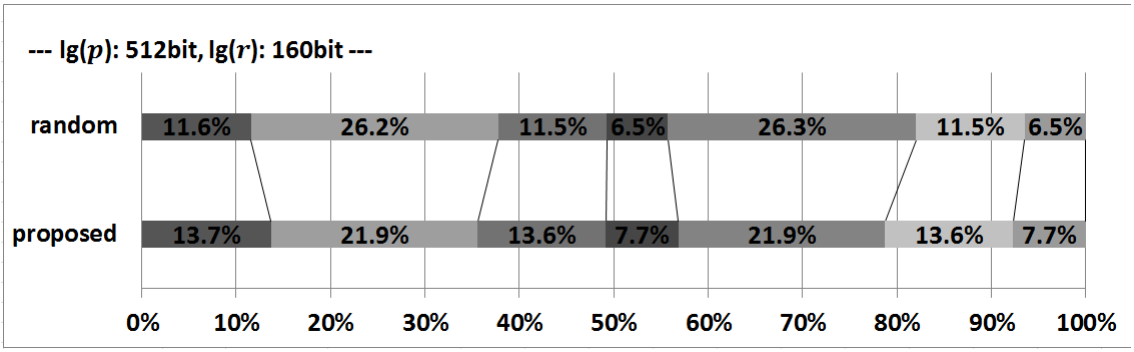
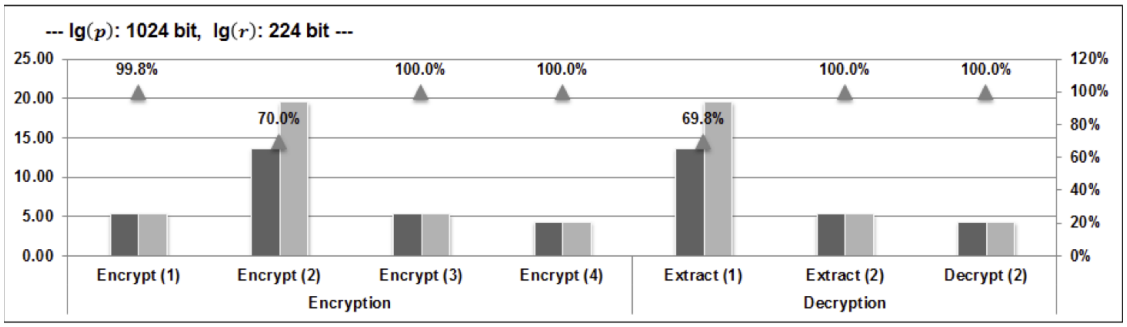


Table 5.1 presents a Execution Time and a Occupation under the security level of RSA-1024 (the same security as ECC-160).

Table 5.2: Timing result (p :1024bit, r :224bit)

Procedure in BF-IBE (§3.2)				Execution Time			Occupation	
				c	c'	c/c'	c	c'
Enctyption:	Encrypt	(1)	rP	5.32	5.33	99.8	10.3	8.4
	Encrypt	(2)	Q_{ID}	**13.7**	**19.5**	**70.0**	**26.4**	**30.7**
	Encrypt	(3)	rQ_{ID}	5.26	5.26	100	10.2	8.3
	Encrypt	(4)	g_{ID}	4.29	4.29	100	8.3	6.8
Decryption:	Extract	(1)	Q_{ID}	**13.7**	**19.6**	**69.8**	**26.4**	**30.8**
	Extract	(2)	sQ_{ID}	5.26	5.26	100	10.2	8.3
	Decrypt	(2)	g'_{ID}	4.27	4.27	100	8.3	6.7



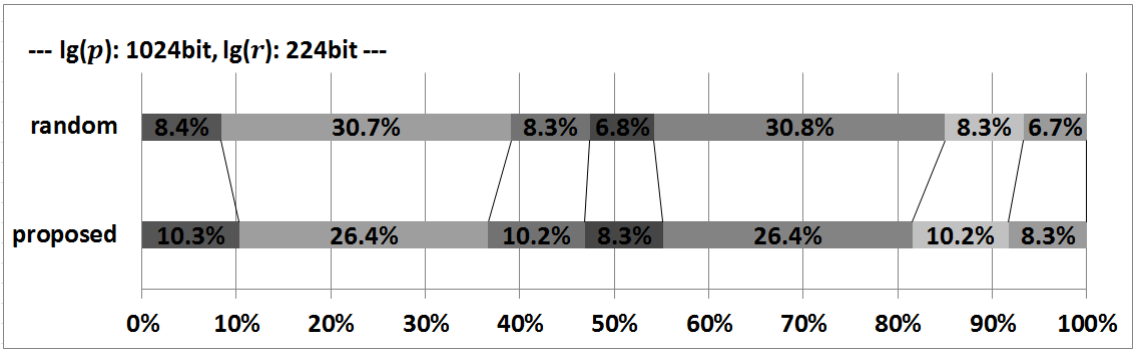
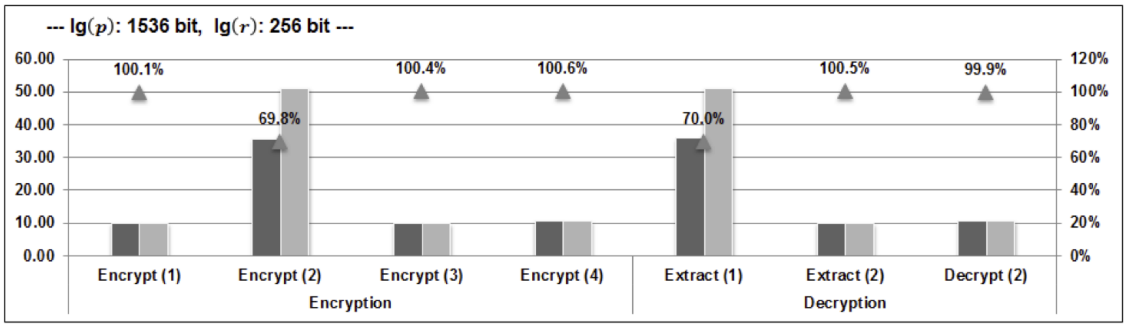


Table 5.2 presents a Execution Time and a Occupation under the security level of RSA-2048 (the same security as ECC-224).

Table 5.3: Timing result (p :1536bit, r :256bit)

Procedure in BF-IBE (§3.2)				Execution Time			Occupation	
				c	c'	c/c'	c	c'
Enctyption:	Encrypt	(1)	rP	9.97	9.86	101c	8.1	6.5
	Encrypt	(2)	Q_{ID}	**35.7**	**51.1**	**69.8**	**29.2**	**33.5**
	Encrypt	(3)	rQ_{ID}	9.80	9.76	100	8.0	6.4
	Encrypt	(4)	g_{ID}	10.6	10.5	101	8.7	6.9
Decryption:	Extract	(1)	Q_{ID}	**35.8**	**51.1**	**70.0**	**29.3**	**33.5**
	Extract	(2)	sQ_{ID}	9.80	9.75	101	8.0	6.4
	Decrypt	(2)	g'_{ID}	10.5	10.5	100	8.6	6.9



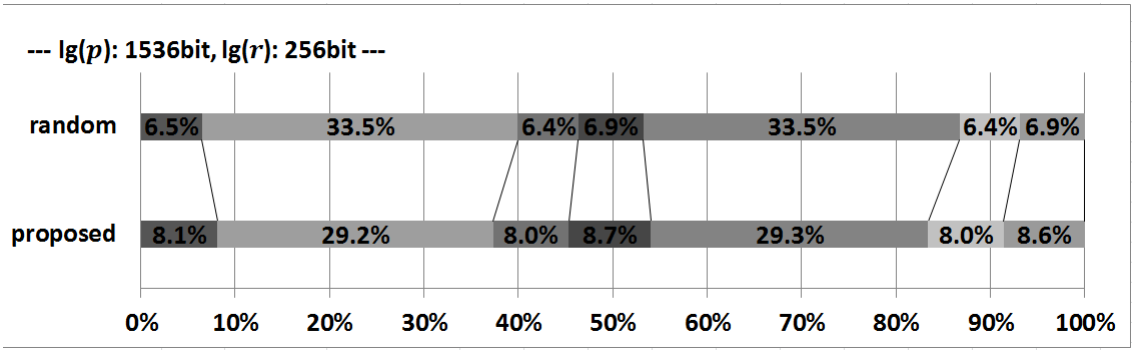


Table 5.3 presents a Execution Time and a Occupation under the security level of RSA-3072 (the same security as ECC-256).

Table 5.4: Comparison of computation costs

$(wt(c), wt(c'), \ell(c)(=\ell(c')))$	$wt(c)(3M+I)+\ell(c)(4M+I)$ where $I = 20M$		
	c	c'	c/c'
$(2, 182, 352)$	$8494M$	$12634M$	0.67
$(2, 397, 800)$	$19246M$	$28331M$	0.68
$(2, 644, 1280)$	$30766M$	$45532M$	0.68

Evaluation:

In Table 5.1–5.3, the timing of HashToPoint (Encrypt (2) and Extract (2)) improves approximately 30% using proposed cofactor c instead of c. Next we evaluate the reason of this improvement. Let M and I be the cost of field multiplication and inversion on $\mathbb{F}_p$, respectively. In [10, Ch. IV.1], the cost of point addition and doubling of an elliptic curve E is $3M + I$ and $4M + I$, respectively. We can calculate the running time of HashToPoint with c as follows. The dominant part of running time of HashToPoint is the scalar multiplication cQ (see §3.3). To calculate cQ using Algorithm 1 (or Algorithm 2 for signed binary representation) shows that the running time of cQ is $wt(c)$ point additions and $\ell(c)$ point doublings where $wt(c)$ denotes by Hamming Weight of c and $\ell(c)$ is the length of c. Therefore the running time of HashToPoint with c is $wt(c)(3M + I) + \ell(c)(4M + I)$. In general, we know that $I = 20M$, we can estimate the ratio c/c' by each three security level. Table 5.4 show the result of the estimate. This is the reason why we improve the running cost of HashToPoint approximately 30%.

CHAPTER 6

CONCLUSION

In this thesis, we proposed efficient system parameters, which are called cofactor, for BF-IBE standardized as RFC5091 by IETF.

First, we searched efficient cofactors whose Hamming weight is 2 using PARI/GP calculator and presented a list of such cofactors. These proposed cofactors can achieve efficient implementation of HashToPoint in BF-IBE. In addition to that, we mentioned the density of such cofactors using prime number theorem.

Next we implemented the cryptographic functions—Setup, Extract, Encrypt, and Decrypt used in BF-IBE— using our proposed cofactors by C language and PBC library. All tests were running on a desktop PC (Mac mini) with an Intel Core i7 2.6 MHz processor (including four core) and 16 GBytes RAM using OS X 10.9.1 (Mavericks). To implement the BF-IBE algorithm, we write programs in ANCI-C using GNU GCC compiler without specific optimizations. We deploy the Pairing Based Crypto (PBC) library. The timing of our implementation of HashToPoint using the proposed system parameters is reduced by approximately 30% on a desktop PC without losing the speed of other cryptographic functions in the IBE.

FUTURE WORKS:

We recommend the following topics for further works.

- In this thesis, we deal with the supersingular elliptic curve with embedding degree two. There remains a work to enhance the algorithm for computing HashToPoint efficiently on ordinary elliptic curves.

- O. Schirokauer has recently proposed a new attack to discrete logarithm problem with prime numbers of low Hamming Weight [91]. T. Nakajima, T. Izu, and T. Takagi proposed parameters p which are efficient computation, as well as security against the attack [78]. In this thesis, we proposed efficient parameter c, which is equal to $(p+1)/r$. To select a suitable c, avoiding the new attack is an open problem for the future.

- To evaluate BF-IBE with our proposed parameters not only on the desktop PC, but also on the other platforms such as smart phone, which is an embedded device and has limited computational resources like CPU and memory. Moreover, we can choose various programming languages other than C.

APPENDIX A

PROGRAMS (SOURCE CODE)

A.1 C PROGRAM (BENCHMARK)

```
#include <pbc.h>
#include <pbc_test.h>

#define getTime pbc_get_time()

int main(int argc, char **argv) {
  int n = atoi(argv[2]);
  double  time_rP, time_Q_ID_1, time_rQ_ID,
          time_pair_enc,
          time_Q_ID_2,
          time_sQ_ID,
          time_pair_dec,
          time_total,
          t0,
          t1,
          t_start;
          time_rP = 0.0;
          time_Q_ID_1 = 0.0;
          time_rQ_ID = 0.0;
          time_pair_enc = 0.0;
          time_Q_ID_2 = 0.0;
          time_sQ_ID = 0.0;
          time_pair_dec = 0.0;
```

```
        time_total = 0.0;
        t0 = 0.0;
        t1 = 0.0;
        t_start = 0.0;
int i;
for (i = 0; i < n; i++) {
  pairing_t pairing;
  element_t s,
            P,
            sP,
            r,
            rP,
            Q_ID_1,
            rQ_ID,
            pairing_enc,
            Q_ID_2,
            sQ_ID,
            pairing_dec;
  pbc_demo_pairing_init(pairing, argc, argv);

  element_init_Zr(s, pairing);
  element_init_G1(P, pairing);
  element_init_G1(sP, pairing);

  element_init_Zr(r, pairing);
  element_init_G1(rP, pairing);
  element_init_G1(Q_ID_1, pairing);
  element_init_G1(rQ_ID, pairing);
  element_init_GT(pairing_enc, pairing);

  element_init_G1(Q_ID_2, pairing);
  element_init_G1(sQ_ID, pairing);
  element_init_GT(pairing_dec, pairing);

  //generate s
  element_random(s);

  //generate P
  element_random(P);
```

```
 //calc sP
 element_pow_zn(sP, P, s);

 //generate r
 element_random(r);

 //calc rP
 t_start = getTime;
 t0 = getTime;
 element_pow_zn(rP, P, r);
 t1 = getTime;
 time_rP += t1 - t0;

 //calc Q_ID_1 (MapToPoint)
 t0 = getTime;
 element_from_hash(Q_ID_1,
                   "t-tomita@math.kyushu-u.ac.jp",28);
 t1 = getTime;
 time_Q_ID_1 += t1 - t0;

//calc rQ_ID
 t0 = getTime;
 element_pow_zn(rQ_ID, Q_ID_1, r);
 t1 = getTime;
 time_rQ_ID += t1 - t0;

//calc pairing_enc
 t0 = getTime;
 element_pairing(pairing_enc, rQ_ID, sP);
 t1 = getTime;
 time_pair_enc += t1 - t0;

//calc Q_ID_2 (MapToPoint)
 t0 = getTime;
element_from_hash(Q_ID_2, "t-tomita@math.kyushu-u.ac.jp",28);
t1 = getTime;
time_Q_ID_2 += t1 - t0;
```

```
    //calc sQ_ID
    t0 = getTime;
    element_pow_zn(sQ_ID, Q_ID_2, s);
    t1 = getTime;
    time_sQ_ID += t1 - t0;

    t0 = getTime;
    element_pairing(pairing_dec, sQ_ID, rP);
    t1 = getTime;
    time_pair_dec += t1 - t0;

    t1 = getTime;
    time_total += t1 - t_start;
    if (element_cmp(pairing_enc, pairing_dec)){
        printf("BUG!\n");
        exit(1);
    }
    element_clear(s);
    element_clear(P);
    element_clear(sP);
    element_clear(r);
    element_clear(rP);
    element_clear(Q_ID_1);
    element_clear(rQ_ID);
    element_clear(pairing_enc);
    element_clear(Q_ID_2);
    element_clear(sQ_ID);
    element_clear(pairing_dec);
    pairing_clear(pairing);
}

    printf("test count,%d\n", n);
    printf("time_rP,%f\n", time_rP*1000 / n);
    printf("time_Q_ID_1,%f\n", time_Q_ID_1*1000 / n);
    printf("time_rQ_ID,%f\n", time_rQ_ID*1000 / n);
    printf("time_pair_enc,%f\n", time_pair_enc*1000 / n);
    printf("time_Q_ID_2,%f\n", time_Q_ID_2*1000 / n);
    printf("time_sQ_ID,%f\n", time_sQ_ID*1000 / n);
    printf("time_pair_dec,%f\n", time_pair_dec*1000 / n);
```

```
    printf("time_total,%f\n", time_total*1000 / n);
    return 0;
}
```

Shell Script:

```
#!/bin/sh

rm ../pbc_perf_test/src/pbc-0.5.14/.libs/libpbc.dylib
gcc -o main5 -I ../pbc_perf_test/src/pbc-0.5.14/include/ \
    -L ../pbc_perf_test/src/pbc-0.5.14/.libs \
    main5.c -lpbc -lgmp

testCount=1000

param_general=l_159_59_general.param
param_proposal=l_159_59_proposal.param
echo start $param_proposal
 ./main5 params/$param_proposal $testCount
echo end $param_proposal
echo start $param_general
 ./main5 params/$param_general $testCount
echo end $param_general

param_general=l_223_153_general.param
param_proposal=l_223_153_proposal.param
echo start $param_proposal
 ./main5 params/$param_proposal $testCount
echo end $param_proposal
echo start $param_general
 ./main5 params/$param_general $testCount
echo end $param_general

param_general=l_255_41_general.param
param_proposal=l_255_41_proposal.param
echo start $param_proposal
 ./main5 params/$param_proposal $testCount
echo end $param_proposal
echo start $param_general
 ./main5 params/$param_general $testCount
```

```
echo end $param_general
```

Parameters:

l_159_59_proposal.param:

```
type a
q 670390396497129854978701249910821151149000489909698600 5\
  790003176226076866890798532578939444964356887959596504 6\
  719786726195019170002768620427181830942228 47
r 730750818665451459101842416358717970580269694977
h 917399446396028604644328358120834776318625995667312449 4\
  950355357547861645537399701511805899744218630324224
exp1 59
exp2 159
sign0 1
sign1 1
```

l_159_59_general.param:

```
type a
q 589317747796890611291950167824502589260033131883508267 4\
  559558143524030183080918514136100417712417217521802684 3\
  058434563855947178526526675574207391384485 07
r 730750818665451459101842416358717970580269694977
h 806455131823381506115416041801269602858156850262936541 2\
  007285065465055856885866131208062141611775532231804
exp1 59
exp2 159
sign0 1
sign1 1
```

l_223_153_proposal.param:

```
type a
q 898846567431157953865413948051652991737141097470206O3\
  545393761101197966913487016939825294674641101972282 71\
  431596021579214107660748631896772254182018039497175 73\
  845978323194506213817892571041812253567010290708847 41\
  048998609928182481527794062724418900719003097576587 92\
  168669403882726808076260402988422124693094 3
r 134799733335753198973449255250514630158670384990258 82\
  201642867097601
h 666801443287985427407985179072125779714475832231590 81\
  603962578117640372378176320715214322008715542907429 29\
  910593433240445969496962809580973790103526455924167 16\
  796088553992779323903639080370926291699130346806301 92\
  716950552698484354153122365 44
exp1 153
exp2 223
sign0 1
sign1 1
```

l_223_153_general.param:

```
type a
q 219796268581215278879172922437918138278863463390606 67\
  032118711936326947372906080961711326093912647814264 58\
  723878463195796789652135855886984535794303077875585 81\
  911529453763576454500303978597923813193930797362256 36\
  864174824850834052982706980606418553839147706437255 33\
  892628262463443458331367964052647356238736 3
```

```
r 134799733335753198973449255250514630158670384990258825\
  201642867097601
h 163053934263917628483733284523089431514799379137852485\
  102467859307592512381245834523523587458362002818729265\
  909878289782289382072708788477397978774493551560978495\
  195162433591914211951868140240719767192249548002472415\
  765046193701458355804003009645
exp1 153
exp2 223
sign0 1
sign1 1
```

l_255_41_proposal.param:

```
type a
q 120515621346051629429005830301415705645604662397284445\
  756798375195784032656724731743453498447409572145362375\
  386105478570547642135513070902940285491006589413429325\
  434022758157044002130222986845284606548580402969184905\
  560264964023604215020820535717829739674932858206278695\
  774549219259528701893373872631445734736326743870072325\
  455072332823501963078198764362006535153457842725386885\
  256687276699958089927826700503242530234763364284239345\
  391245665491296516462113208868880252927
r 578960446186580977117854925043439539266349923328202825\
  019728792006155588075521
h 208158643893287981638504806547281710772305244945334095\
  610638224700807216119346720596024478883464648369684845\
  322790856201558276713249664692981627981321135464152585\
```

```
  482590187784406915463666993231671009459188410953796 22\
  423387354295096957733925002768876520583464697770622 32\
  165707683317005651120933244966378183760369413644440 62\
  810420533968709894385374707708584453969599510412319 18\
  132805729517568
exp1 41
exp2 255
sign0 1
sign1 1
```

l_255_41_general.param:

```
type a
q 564356090032987850790261181746353824800313191032089 73\
  319255242938528825203467066806642000349490027323011 01\
  837381274628400265513197625831655462398411879169963 53\
  546539320033566162186991059152873133664380723211177 50\
  027107476926953622407065519439041171479525684303725 01\
  308950211417218832996932088815900448506366689483664 14\
  798991465172301541850991846018551791690480686544512 89\
  812816611340679900947436434844486572152159887109020 54\
  061850441237653847105487931339180843 71
r 578960446186580977117854925043439539266349923328202 82\
  019728792006155588075521
h 974774863723787797839911463979887585720367421882809 06\
  420419987158292587445181435922594327305944173222634 8643\
  496848995819447623819277233518276131401705784742666 7232\
  466734970828174007380962222235835911437387110104369 3391\
  509727782518504083007506312332326210338053527058892 5386\
```

```
  615049055992227547410072205651505107072280164239389541 7\
  109198558397714457041350641509634097529572712684071443 7\
  32
exp1 41
exp2 255
sign0 1
sign1 1
```

Output Samples:

```
start l_159_59_proposal.param
test count,1000
time_rP,1.881750
time_Q_ID_1,3.006005
time_rQ_ID,1.869219
time_pair_enc,1.061821
time_Q_ID_2,3.015178
time_sQ_ID,1.867805
time_pair_dec,1.061464
time_total,13.763664
end l_159_59_proposal.param
start l_159_59_general.param
test count,1000
time_rP,1.888596
time_Q_ID_1,4.274279
time_rQ_ID,1.869418
time_pair_enc,1.060167
time_Q_ID_2,4.277790
time_sQ_ID,1.870109
```

```
time_pair_dec,1.060170
time_total,16.300943
end l_159_59_general.param
start l_223_153_proposal.param
test count,1000
time_rP,5.324740
time_Q_ID_1,13.652358
time_rQ_ID,5.256011
time_pair_enc,4.287045
time_Q_ID_2,13.650458
time_sQ_ID,5.255549
time_pair_dec,4.274219
time_total,51.700798
end l_223_153_proposal.param
start l_223_153_general.param
test count,1000
time_rP,5.333480
time_Q_ID_1,19.514330
time_rQ_ID,5.257851
time_pair_enc,4.294408
time_Q_ID_2,19.549102
time_sQ_ID,5.260588
time_pair_dec,4.268909
time_total,63.479118
end l_223_153_general.param
start l_255_41_proposal.param
test count,1000
```

```
time_rP,9.867092
time_Q_ID_1,35.679617
time_rQ_ID,9.804549
time_pair_enc,10.575425
time_Q_ID_2,35.792747
time_sQ_ID,9.804978
time_pair_dec,10.523498
time_total,122.048429
end l_255_41_proposal.param
start l_255_41_general.param
test count,1000
time_rP,9.864049
time_Q_ID_1,51.134755
time_rQ_ID,9.761029
time_pair_enc,10.519335
time_Q_ID_2,51.143395
time_sQ_ID,9.750550
time_pair_dec,10.527038
time_total,152.700653
end l_255_41_general.param
```

A.2 PARI/GP Script

A.2.1 Find Cofactors

The following script shows how to find cofactors with using algorithm 7 by PARI/GP, which is free software with the main aim of facilitating number

theory computations.

```
/**
 * Find a cofactor c which is equal to (p-1)/r
 * where c's Hamming Weight of less than three and
 * p is a prime and p (mod 4) = 3.
 * @param pBitLen: Bit length of prime p.
 * @param rBitLen:  Bit length of prime r.
 * @param r:       Solinas prime.
 */
find_cofactor(pBitLen, rBitLen, r)=
{
a=pBitLen - rBitLen;
b=2^a;
sign_str=["+","-"];

// Find a cofactor with Hamming Weight of 1.
c = b;
p = r*c - 1;
if(length(binary(p)) == pBitLen && p%4 == 3 && isprime(p),
  printf("2^%d\n",a);
);

// Find a cofactor with Hamming Weight of 2.
for(i=1, a-1,
  c = [b + 2^i, b - 2^i];
  for(j=1, 2,
    p = r*c[j] - 1;
```

```
    if(length(binary(p)) == pBitLen && p%4 == 3 && isprime(p),
       printf("2^%d %s 2^%d\n",a,sign_str[j],i);
    );
  );
);
}

/**
 * Entry point of this searching procedure. (main function)
 */
Main =
{
\\ Prepare the list of solinas primes for
\\ (p:512bit r:160bit),
\\ (p:1024bit r:224bit),
\\ (p:1536bit r:256bit).
rList=
[
  [
    [512,160],
    2^159 + 2^17  + 1, 2^159 + 2^19  + 1, 2^159 + 2^59  + 1,
    2^159 + 2^63  + 1, 2^159 + 2^88  - 1, 2^159 + 2^107 + 1,
    2^159 + 2^110 - 1, 2^159 + 2^116 - 1, 2^159 + 2^135 + 1,
    2^159 + 2^138 - 1
  ],
  [
    [1024,224],
```

```
      2^223 + 2^8   - 1, 2^223 + 2^10  - 1, 2^223 + 2^13  + 1,
      2^223 + 2^30  - 1, 2^223 + 2^55  + 1, 2^223 + 2^80  - 1,
      2^223 + 2^139 + 1, 2^223 + 2^153 + 1
    ],
    [
      [1536,256],
      2^255 + 2^41  + 1, 2^255 + 2^96  - 1, 2^255 + 2^166 + 1,
      2^255 + 2^176 - 1, 2^255 + 2^227 + 1, 2^255 + 2^232 - 1,
      2^255 + 2^243 + 1
    ],
 ];

for(i=1,length(rList),
  printf("=== p:%d bit, r:%d bit ===\n",
           rList[i][1][1],rList[i][1][2]);
  for(j=2,length(rList[i]),
    find_cofactor(rList[i][1][1],
                  rList[i][1][2], rList[i][j]);
    print();
  );
);
return(0);
}
```

A.2.2 FIND SOLINAS PRIME

```
find_solinas_primes(bitLength)=
```

```
{
a = 2^(bitLength-1);
signs = [[1,1],[1,-1],[-1,1],[-1,-1]];
signs_str = [["+","+"],["+","-"],["-","+"],["-","-"]];

for(i=1, level-2,
  for(j=1,length(signs),
    p = a + 2^i*signs[j][1] + 1*signs[j][2];
    if(length(binary(p))==level && isprime(p),
      printf("2^%d %s2^%d %s1\n",
        bitLength-1,
        signs_str[j][1],
        i,
        signs_str[j][2]
      )
    );
  )
)
}

my_main=
{
\\ (+/-) 2^159 (+/-) 2^a (+/-) 2^b
security_level=[160,224,256,384,512];
for(i=1,length(security_level),
  level=security_level[i];
  printf("=== find solinas l(%d bit) ===\n", level);
```

```
  find_solinas_primes(level);
  printf("\n");
);
return(0);
}
```

Output Sample:

```
=== find solinas l(160 bit) ===
2^159 +2^17 +1
2^159 +2^19 +1
2^159 +2^59 +1
2^159 +2^63 +1
2^159 +2^88 -1
2^159 +2^107 +1
2^159 +2^110 -1
2^159 +2^116 -1
2^159 +2^135 +1
2^159 +2^138 -1

=== find solinas l(224 bit) ===
2^223 +2^8 -1
2^223 +2^10 -1
2^223 +2^13 +1
2^223 +2^30 -1
2^223 +2^55 +1
2^223 +2^80 -1
2^223 +2^139 +1
2^223 +2^153 +1
```

```
=== find solinas l(256 bit) ===
2^255 +2^41 +1
2^255 +2^96 -1
2^255 +2^166 +1
2^255 +2^176 -1
2^255 +2^227 +1
2^255 +2^232 -1
2^255 +2^243 +1

=== find solinas l(384 bit) ===
2^383 +2^155 +1
2^383 +2^233 +1
2^383 +2^270 +1

=== find solinas l(512 bit) ===
2^511 +2^19 +1
2^511 +2^34 +1
2^511 +2^87 +1
2^511 +2^102 -1
2^511 +2^156 -1
2^511 +2^322 +1
2^511 +2^334 +1
2^511 +2^344 -1
2^511 +2^462 -1
2^511 +2^466 -1
```

A.2.3 CALCULATE HAMMING WEIGHT

```
check_hamming_weight(vec)=
{
counter=0;
for(i=1,length(vec),
  if(vec[i] != 0,counter++)
);
return(counter);
}
```

APPENDIX B

RATIONAL POINTS IN ABELIAN VARIETY

B.1 JACOBIAN OF CURVES

Let X be a projective non-singular algebraic curve defined over $\mathbf{Q}$ whose genus is g. If given such a curve, we can construct a pair $(\mathrm{Jac}(X), j)$ such that

1. $\mathrm{Jac}(X)$ is an abelian variety whose dimension is g

2. an injection $j : X \hookrightarrow \mathrm{Jac}(X)$.

$\mathrm{Jac}(X)$ is called the *Jacobian* (or *Jacobian variety*) *of* X, and j is called the *Jacobian embedding* of X. It is known that such a curve X determines the pair $(\mathrm{Jac}(X), j)$ up to a natural isomorphism. The fundamental tool for the algebraic construction of the Jacobian is the Riemann-Roch theorem for curves, see [110], [111].

Remark. The Jacobian $\mathrm{Jac}(X)$ of a curve X is naturally isomorphic to $\mathrm{Pic}^0(X)$. Suppose that X is defined over a field K. Then its Jacobian $\mathrm{Jac}(X)$ is also defined over K. However it may not be possible to define the injection $j : X \hookrightarrow \mathrm{Jac}(X)$ over K. More precisely, the map j is defined by choosing a divisor D of

degree 1 and then setting

$$j : X \hookrightarrow \mathrm{Pic}^0(X) \simeq \mathrm{Jac}(X) \quad (P) \mapsto [(P) - D].$$

In particular, if there is a K-rational point $Q \in X(K)$, then we can take $D = (Q)$ to get a map j that is defined over K.

Abelian varieties of GL_2-type:

In this subsection, we introduce the concept of abelian varieties of "GL_2-type" introduced by K. Ribet in [84]. Roughly speaking, this is an abelian variety defined over $\mathbf{Q}$ whose algebra of $\mathbf{Q}$-endomorphisms is a number field of degree equal to the dimension of the abelian variety. This is a subcategory of the category of abelian varieties defined over $\mathbf{Q}$ as follows:

Definition B.1.1. An abelian variety A defined over $\mathbf{Q}$ is said to be GL_2-type if the algebra of $\mathbf{Q}$-endomorphisms of A defined over $\mathbf{Q}$ ($\mathrm{End}_{\mathbf{Q}}(A) \otimes \mathbf{Q}$) is a number field of degree equal to the dimension of A. That is the number field $\mathrm{End}_{\mathbf{Q}}(A) \otimes \mathbf{Q}$ satisfies

$$[\mathrm{End}_{\mathbf{Q}}(A) \otimes \mathbf{Q} : \mathbf{Q}] = \dim_{\mathbf{Q}}(A).$$

The reason why this concept is called GL_2-type is that if A is an abelian variety such that $F := \mathrm{End}_{\mathbf{Q}}(A) \otimes \mathbf{Q}$ is a number field with $[F : \mathbf{Q}] = \dim(A)$, then it is known that for any prime l, the Tate module $V_l(A)$ is free $F \otimes_{\mathbf{Q}} \mathbf{Q}_l$-module of rank two (for the definition of the Tate module, see §3.4). Thus the action of the absolute Galois group $G_{\mathbf{Q}}$ on $V_l(A)$ defines a representation $G_{\mathbf{Q}} \to \mathrm{GL}_2(F \otimes_{\mathbf{Q}} \mathbf{Q}_l)$. In general, it is known the following result:

Proposition B.1.1. If A is an abelian variety over $\mathbf{Q}$ and a field $K \subset \mathrm{End}_{\mathbf{Q}}(A) \otimes \mathbf{Q}$, then $[K : \mathbf{Q}]$ divides $\dim(A)$

(Sketch of proof). As discussed in [84, §2], $F = \mathrm{End}_{\mathbf{Q}}\mathbf{Q}$ acts on the tangent space $\mathrm{Lie}(A)$ over $\mathbf{Q}$ to A at 0 which is a $\mathbf{Q}$-vector space of dimension $\dim(A)$ by functoriality [77, §11]. Thus $\mathrm{Lie}(A)$ is a vector space over K, hence has $\mathbf{Q}$-dimension a multiple of $[K : \mathbf{Q}]$. □

From this Proposition, an abelian variety A of GL_2-type has a number field contained in $\mathrm{End}_{\mathbf{Q}}(A) \otimes \mathbf{Q}$ whose degree is the maximal dimension $\dim(A)$.

Example B.1.1. Every elliptic curve E defined over $\mathbf{Q}$ is of GL_2-type (since $\mathbf{Q} \subset \mathrm{End}_{\mathbf{Q}}(E) \otimes \mathbf{Q}$). Moreover we see that if E has a complex multiplication ($\mathbf{Z} \neq \mathrm{End}_{\bar{\mathbf{Q}}}(E)$), then extra endomorphisms are never defined over $\mathbf{Q}$ by the above Prop. B.1.1.

Example B.1.2. All modular abelian varieties are of GL_2-type. Let A_f be a modular abelian variety (the construction of A_f, see §2.3). The $\mathbf{Q}$-endomorphisms defined over $\mathbf{Q}$ ($\mathrm{End}_{\mathbf{Q}}(A_f) \otimes \mathbf{Q}$) contains a number field F_f which is constructed of Hecke algebras (Hecke correspondences) of $X_0(N)$, and we see that F_f satisfies $[F_f : \mathbf{Q}] = \dim(A_f)$ by its construction.

Algebraic and analytic categories:

If X is an algebraic variety defined over $\mathbf{C}$, then we can regard X as a complex manifold X^{an}. That is there exists a functor h from the category of schemes of finite type over $\mathbf{C}$ (an object of this category is often called algebraic scheme) to the category of complex analytic spaces

$$(\mathrm{Var})/\mathbf{C} \xrightarrow{\ h\ } (\mathrm{An}) \tag{B.1}$$

$$(X, \mathscr{O}_X) \longmapsto (X^{\mathrm{an}}, \mathscr{O}_X^{\mathrm{an}})$$

(for the explicit construction, see [48, Appendix B, pp. 438]). In this sense, we can obtain an analytic space from arbitrary algebraic varieties defined over

C. If X is a projective scheme, there is a famous theorem as follows:

Theorem B.1.2. [82, "GAGA principle", Prop. 15, Thm. 1–3]Let X be a projective scheme over C. Then the functor h induces an equivalence of categories from the category of coherent sheaves on X to the category of coherent analytic sheaves on X^{an}. Furthermore, for every coherent sheaf $\mathscr{F}$ on X, the natural maps

$$\alpha_i : H^i(X, \mathscr{F}) \to H^i(X^{\mathrm{an}}, \mathscr{F}^{\mathrm{an}})$$

are isomorphisms, for all i.

Lattice index:

Let V be a finite-dimensional vector space defined over $\mathbf{R}$. A *lattice* $L \subset V$ is a free abelian group of rank equal to the dimension of V such that $L \otimes_{\mathbf{Z}} \mathbf{R} = V$.

If $L, M \subset V$ are lattices, the *lattice index* $[L : M] \in \mathbf{R}$ is the absolute value of the determinant of an automorphism of V taking L isomorphically onto M.

Our settings:

Let X be a projective non-singular algebraic curve defined over $\mathbf{Q}$ whose genus is g and has at least one rational point $Q \in X(\mathbf{Q})$, and J be its Jacobian (with Jacobian embedding $i_Q : X \hookrightarrow J$).

Remark. Since X is defined over $\mathbf{Q}$ and has at least one rational point the pair (J, j) are both defined over $\mathbf{Q}$ (see Remark B.1).

Moreover we assume

(A0) We can find an isogeny decomposition

$$J \sim A_1 \times \cdots \times A_r$$

each A_i is simple and not isogenous to A_j for $i \neq j$.

(A1) Each A_i is an abelian variety of GL_2-type.

Let F be the $\mathrm{End}_{\mathbf{Q}}(J)\otimes\mathbf{Q}$ and F_i be the $\mathrm{End}_{\mathbf{Q}}(A_i)\otimes\mathbf{Q}$. Then we have a canonical decomposition as rings

$$F \simeq \prod_{i=1}^{r} F_i. \tag{B.2}$$

Let Ω_{A_i} be the sheaf of regular differentials on A_i, and $\Gamma(A_i, \Omega_{A_i})$ be the space of its global sections. Then the space $\Gamma(A_i, \Omega_{A_i})$ is a $\mathbf{Q}$-vector space of dimension $d_i := \dim(A_i)$. By functoriality, we see that each F_i acts on $\Gamma(A_i, \Omega_{A_i})$. Since $[F_i : \mathbf{Q}] = \dim_{\mathbf{Q}}(\Gamma(A_i, \Omega_{A_i})) = d_i$, the space $\Gamma(A_i, \Omega_{A_i})$ can be regarded as a F_i-vector space of dimension one. These induce the action of F on the space $\Gamma(J, \Omega_J)$, and we see that the space $\Gamma(J, \Omega_J)$ is a free F-module of rank 1.

B.2 Perfect Parings

Next we consider X as an analytic space X^{an} ($X \xrightarrow{h} X^{\mathrm{an}}$ by (B.1)) by the embedding $\mathbf{Q} \hookrightarrow \mathbf{C}$. Since X is projective and non-singular, X^{an} is a compact one-dimensional complex manifold (i.e. compact Riemann surface). So we can consider a system of coordinate charts (U_α, ϕ_α) where each $\phi_\alpha : U_\alpha \to \mathbf{C}$ is a homeomorphism of U_α onto an open subset of $\mathbf{C}$, such that the change of coordinate maps are analytic isomorphisms. A 1-form ω on X^{an} (i.e. a global section on X^{an} of the sheaf of holomorphic differentials $\Omega^{\mathrm{an}}_{X^{\mathrm{an}}}$ in the sense of complex analysis, we denote $\Gamma(X^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X^{\mathrm{an}}})$ this space) is a choice of two holomorphic functions f and g to each local coordinate $z = x + iy$ on $U_\alpha \subset X^{\mathrm{an}}$ such that $fdx + gdy$ is invariant under the change of coordinates. If $\gamma : [0, 1] \to X^{\mathrm{an}}$

is a path and $\omega = fdx + gdy$ is such a 1-form, then we define the integral

$$\langle\ ,\ \rangle_{X^{\mathrm{an}}} := \int_\gamma \omega := \int_0^1 \left(f(x(t), y(t))\frac{dx}{dt} + g(x(t), y(t))\frac{dy}{dt} \right) dt \in \mathbf{C}.$$

We introduce the following perfect paring.

Proposition B.2.1. There exists a R-linear perfect paring as follows:

$$H_1(X^{\mathrm{an}}, \mathbf{R}) \times \mathrm{Res}_{\mathbf{C}/\mathbf{R}}\left(\Gamma(X^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X^{\mathrm{an}}})\right) \xrightarrow{\langle\ ,\ \rangle_{X^{\mathrm{an}}}} \mathbf{C}. \tag{B.3}$$

Here $H_1(X^{\mathrm{an}}, \mathbf{R})$ is regarded as $H_1(X^{\mathrm{an}}, \mathbf{Z}) \otimes_{\mathbf{Z}} \mathbf{R}$, and $H_1(X^{\mathrm{an}}, \mathbf{Z})$ denotes the singular homology group with coefficient in $\mathbf{Z}$.

This perfect paring plays a important key part in our construction of the winding element.

(sketch of proof). Let $a_1, \ldots, a_{2g}$ be the fundamental cycles relative to a polygonal decomposition of the Riemann surface. We may view $H_1(X^{\mathrm{an}}, \mathbf{R})$ as the space of formal linear combinations $\gamma = \sum x_i a_i$ with real coefficients $x_i \in \mathbf{R}$. Then we construct the paring between $H_1(X^{\mathrm{an}}, \mathbf{R})$ and $\Gamma(X^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X^{\mathrm{an}}})$

$$\langle \gamma, \omega \rangle_{X^{\mathrm{an}}} := \sum x_i \int_{a_i} \omega.$$

Next we can construct the dual basis in $\Gamma(X^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X^{\mathrm{an}}})$ to the basis $a_1, \ldots, a_{2g}$ with respect to the real part of the above paring (an explicit construction, see [59, Lem. 4, III]). We can show that the natural map

$$\begin{array}{ccc} H_1(X^{\mathrm{an}}, \mathbf{R}) & \to & \mathrm{Hom}_{\mathbf{R}}(\mathrm{Res}_{\mathbf{C}/\mathbf{R}}\left(\Gamma(X^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X^{\mathrm{an}}})\right), \mathbf{R}) \\ (\gamma & \mapsto & \omega \mapsto \langle \gamma, \omega \rangle_{X^{\mathrm{an}}}) \end{array}$$

is an injection. Since the two spaces have the same real dimension, this injection must be an isomorphism. □

To interpret this paring as a duality over $\mathbf{C}$, we can give $H_1(X^{\mathrm{an}}, \mathbf{R})$ the structure of a vector space over $\mathbf{C}$ (of dimension g) as follows. Given $\gamma \in H_1(X^{\mathrm{an}}, \mathbf{R})$ and $\alpha \in \mathbf{C}$, we can define $\alpha\gamma$ to be that element of $H_1(X^{\mathrm{an}}, \mathbf{R})$ which satisfies $\langle \alpha\gamma, \omega \rangle_{X^{\mathrm{an}}} = \langle \gamma, \alpha\omega \rangle_{X^{\mathrm{an}}}$ for all $\omega \in \Gamma(X^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X^{\mathrm{an}}})$. In other words, $\alpha\gamma$ is the element corresponding to the functional $\omega \mapsto \alpha\langle \gamma, \omega \rangle_{X^{\mathrm{an}}}$. Now the map $(\gamma, \omega) \mapsto \langle \gamma, \omega \rangle_{X^{\mathrm{an}}}$ is $\mathbf{C}$-bilinear, and we have a $\mathbf{C}$-linear perfect paring

$$H_1(X^{\mathrm{an}}, \mathbf{R}) \times \Gamma(X^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X^{\mathrm{an}}}) \xrightarrow{\langle\ ,\ \rangle_{X^{\mathrm{an}}}} \mathbf{C}. \tag{B.4}$$

Next we show that this paring induces a perfect paring on the $H_1(J^{\mathrm{an}}, \mathbf{R})$ and $\Gamma(J^{\mathrm{an}}, \Omega^{\mathrm{an}}_{J^{\mathrm{an}}})$ as follows:

Proposition B.2.2. The above perfect paring $(B.3)$ induces a $\mathbf{C}$-linear perfect paring:

$$H_1(J^{\mathrm{an}}, \mathbf{R}) \times \Gamma(J^{\mathrm{an}}, \Omega^{\mathrm{an}}_{J^{\mathrm{an}}}) \xrightarrow{\langle\ ,\ \rangle_{J^{\mathrm{an}}}} \mathbf{C}. \tag{B.5}$$

Proof. First we prepare a lemma:

Lemma B.2.3. There exists an isomorphism $\Gamma(J^{\mathrm{an}}, \Omega^{\mathrm{an}}_{J^{\mathrm{an}}}) \xrightarrow{\sim} \Gamma(X^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X^{\mathrm{an}}})$ as $\mathbf{C}$-vector space.

Proof. It is known that the pull back $i_Q^* : \Gamma(J, \Omega_J) \to \Gamma(X, \Omega_X)$ which is induced from the Jacobian embedding $i_Q : X \hookrightarrow J$ induces an isomorphism as $\mathbf{Q}$-vector spaces by the method of algebraic geometry [75, Prop. 2.2]. Since X^{an} and J^{an} are analytic objects arising from non-singular algebraic varieties over $\mathbf{C}$ (theses spaces are written $X(\mathbf{C})$, $J(\mathbf{C})$ by the embedding $\mathbf{Q} \hookrightarrow \mathbf{C}$), the GAGA principle is applied. □

Next we show that

$$H_1(J^{\mathrm{an}}, \mathbf{Z}) \text{ is essentially identified with } H_1(X^{\mathrm{an}}, \mathbf{Z}). \tag{B.6}$$

If this is shown, we can construct a perfect paring (B.5) by combining (B.3), Lem. B.2.3, and (B.6). First we recall the analytic construction of Jacobian variety J^{an}. $\Gamma(X^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X^{\mathrm{an}}})$ is a complex vector space of dimension g. From the theory of abelian integrals, we see that the map $\gamma \mapsto (\omega \mapsto \int_\gamma \omega)$ embeds $H_1(X^{\mathrm{an}}, \mathbf{Z})$ as a lattice into the dual space $\Gamma(X^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X^{\mathrm{an}}})^\vee$, and we define the Jacobian J^{an} by the quotient $J^{\mathrm{an}} := \Gamma(X^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X^{\mathrm{an}}})^{vee}/H_1(X^{\mathrm{an}}, \mathbf{Z})$ (This is a complex torus, and the paring $H_1(X^{\mathrm{an}}, \mathbf{Z}) \times H_1(X^{\mathrm{an}}, \mathbf{Z}) \to \mathbf{Z}$ defined by Poincaré duality gives a non-degenerate Riemann form on J^{an}, therefore J^{an} is an abelian variety over C). From the above construction, we have following exact sequence:

$$0 \to H_1(X^{\mathrm{an}}, \mathbf{Z}) \to \Gamma(X^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X^{\mathrm{an}}})^\vee \to J^{\mathrm{an}} \to 0.$$

We denote $H_1(X^{\mathrm{an}}, \mathbf{Z})$ by G. This group G acts on the space $\Gamma(X^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X^{\mathrm{an}}})^\vee$ freely and discontinuously, so that $\Gamma(X^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X^{\mathrm{an}}})^\vee$ is the principal G-space. Moreover $\Gamma(X^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X^{\mathrm{an}}})^\vee \simeq \mathbf{C}^g$ is contractible (i.e. simply connected), then the natural projection map $\pi : \Gamma(X^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X^{\mathrm{an}}}) \to J^{\mathrm{an}}$ is a universal covering of J^{an}. Therefore $\pi_1(J^{\mathrm{an}}) \simeq G$. Since G is an abelian group, $\pi_1(J^{\mathrm{an}}) \simeq H_1(J^{\mathrm{an}}, \mathbf{Z})$ so we see that $H_1(J^{\mathrm{an}}, \mathbf{Z}) \simeq G$. □

Now we consider the relation between algebraic and analytic spaces. For each F_i, we identify $F_i \otimes_{\mathbf{Q}} \mathbf{C}$ with $\prod_{\sigma \in \mathrm{Hom}(F_i, \mathbf{C})} \mathbf{C}$ for the set $\mathrm{Hom}(F_i, \mathbf{C})$ of embeddings of F_i into C. The number of elements in $\mathrm{Hom}(F_i, \mathbf{C})$ is $d_i := \dim(A_i)$. It is known that the space $\Gamma(A_i^{\mathrm{an}}, \Omega^{\mathrm{an}}_{A_i^{\mathrm{an}}})$ is a C-vector space of dimension d_i, and each $\Gamma(A_i^{\mathrm{an}}, \Omega^{\mathrm{an}}_{A_i^{\mathrm{an}}})$ is a free $F_i \otimes_{\mathbf{Q}} \mathbf{C}$-module of rank one, then $\Gamma(J^{\mathrm{an}}, \Omega^{\mathrm{an}}_{J^{\mathrm{an}}})$ is also a free $F \otimes_{\mathbf{Q}} \mathbf{C}$-module of rank 1. By using the perfect paring (B.5), we can

define an action of $F \otimes_{\mathbf{Q}} \mathbf{C}$ on the homology group $H_1(J^{\mathrm{an}}, \mathbf{R})$ (and it is also a free $F \otimes_{\mathbf{Q}} \mathbf{C}$-module of rank 1).

Proposition B.2.4. There exists a canonical decomposition:

$$\Gamma(J^{\mathrm{an}}, \Omega^{\mathrm{an}}_{J^{\mathrm{an}}}) \simeq \bigoplus_{i=1}^{r} \Gamma(A_i^{\mathrm{an}}, \Omega^{\mathrm{an}}_{A_i^{\mathrm{an}}}) \tag{B.7}$$

$$H_1(J^{\mathrm{an}}, \mathbf{R}) \simeq \bigoplus_{i=1}^{r} H_1(A_i^{\mathrm{an}}, \mathbf{R}) \tag{B.8}$$

where each $H_1(A_i^{\mathrm{an}}, \mathbf{R})$ and $\Gamma(A_i^{\mathrm{an}}, \Omega^{\mathrm{an}}_{A_i^{\mathrm{an}}})$ are both free $F_i \otimes_{\mathbf{Q}} \mathbf{C}$-modules of rank one.

Proof. To decompose the left hand side, we use a basic fact in linear algebra:

Lemma B.2.5. Let M be an F-module and r be an integer ≥ 1. For each $i = 1, \ldots, r$ let $e_i : M \to M$ be an F-homomorphism such that

$$\sum_{i=1}^{r} e_i = \mathrm{id}_M \quad \textit{and} \quad e_i \circ e_j = 0 \quad \textit{if } i \neq j. \tag{B.9}$$

Then $e_i^2 = e_i$ for all i. Let $M_i := e_i(M)$. Then the map

$$M \to \prod_{i=1}^{r} M_i \quad \varphi(x) \mapsto (e_1(x), \ldots, e_r(x)).$$

is an F-isomorphism of M onto the direct product $\prod M_i$.

Proof. See [61, III, Prop. 3.1, pp. 128]. □

For each i, let

$$e_i \in F \otimes_{\mathbf{Q}} \mathbf{C} \leftrightarrow (0, \ldots, 0, 1, 0, \ldots, 0) \in \prod_{i=1}^{r} F_i \otimes_{\mathbf{Q}} \mathbf{C}$$

be projections onto the factor $F_i \otimes_{\mathbf{Q}} \mathbf{C}$ on the product. It is easy to see that these e_i satisfy the assumptions (B.9), we have the decomposition (B.7) and (B.8) by applying Lemma B.2.5 to these e_i. □

From (B.7), (B.8), we have a C-linear perfect paring

$$H_1(A_i^{\mathrm{an}}, \mathbf{R}) \times \Gamma(A_i^{\mathrm{an}}, \Omega^{\mathrm{an}}_{A_i^{\mathrm{an}}}) \xrightarrow{\langle\ ,\ \rangle_{A_i^{\mathrm{an}}}} \mathbf{C}, \tag{B.10}$$

and summarize the above discussions, we have the following diagram:

$$\begin{array}{ccccc}
H_1(X^{\mathrm{an}}, \mathbf{R}) & \times & \Gamma(X^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X^{\mathrm{an}}}) & \longrightarrow & \mathbf{C} \cdots (B.3) \\
{\scriptstyle (B.6)}\| & & {\scriptstyle \text{Lem. B.2.3}}\uparrow \simeq & & \\
H_1(J^{\mathrm{an}}, \mathbf{R}) & \times & \Gamma(J^{\mathrm{an}}, \Omega^{\mathrm{an}}_{J^{\mathrm{an}}}) & \longrightarrow & \mathbf{C} \cdots (B.5) \\
\downarrow e_i & & \downarrow e_i & & \\
H_1(A_i^{\mathrm{an}}, \mathbf{R}) & \times & \Gamma(A_i^{\mathrm{an}}, \Omega^{\mathrm{an}}_{A_i^{\mathrm{an}}}) & \longrightarrow & \mathbf{C} \cdots (B.10)
\end{array}$$

From the above diagram, we have the following diagram:

$$\begin{array}{ccccc}
H_1(X^{\mathrm{an}}, \mathbf{R}) & \xrightarrow{\simeq} & \Gamma(X^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X^{\mathrm{an}}})^\vee & & \\
\| & & \simeq\uparrow & & \\
H_1(J^{\mathrm{an}}, \mathbf{R}) & \xrightarrow{\simeq} & \Gamma(J^{\mathrm{an}}, \Omega^{\mathrm{an}}_{J^{\mathrm{an}}})^\vee & \prec\cdots & \text{free } F \otimes_{\mathbf{Q}} \mathbf{C}\text{- module of rank one.} \\
\downarrow e_i & & \downarrow e_i & & \\
H_1(A_i^{\mathrm{an}}, \mathbf{R}) & \xrightarrow{\simeq} & \Gamma(A_i^{\mathrm{an}}, \Omega^{\mathrm{an}}_{A_i^{\mathrm{an}}})^\vee & \prec\cdots & \text{free } F_i \otimes_{\mathbf{Q}} \mathbf{C}\text{- module of rank one.}
\end{array} \tag{B.11}$$

where $\Gamma(X^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X^{\mathrm{an}}})^\vee$ denotes the dual space $\mathrm{Hom}_{\mathbf{C}}(\Gamma(X^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X^{\mathrm{an}}}), \mathbf{C})$. The key point of the above diagram (B.11) is that the middle row is free $F \otimes_{\mathbf{Q}} \mathbf{C} \simeq \prod_i F_i \otimes_{\mathbf{Q}} \mathbf{C}$-module of rank one, and the lower row is free $F_i \otimes_{\mathbf{Q}} \mathbf{C} \simeq \prod_{\sigma \in \mathrm{Hom}(F_i, \mathbf{C})} \mathbf{C} \simeq \mathbf{C}^{d_i}$-module of rank one.

B.3 SUMMARY OF THE WORK OF A. AGASHE AND W. STEIN

Let $X_0(N)$ be the modular curve defined over $\mathbf{Q}$ associated with the problem of classifying elliptic curves E together with cyclic subgroups of E having order N (as a Riemann surface, $X_0^{\mathrm{an}}(N)$ is the quotient $\Gamma_0(N)\backslash\mathfrak{h}^*$ of the extended complex upper halfplane $\mathfrak{h}^*$) and $J_0(N)$ denotes its Jacobian. Let $S_2(\Gamma_0(N))_{\mathbf{C}}$ is a space of cuspforms of weight 2 on $\Gamma_0(N)$ with an action of the Hecke algebra $\mathbf{T}$ (cf. [25]). A new form $f = \sum_{n\geq 1} a_n q^n \in S_2(\Gamma_0(N))_{\mathbf{C}}^{\mathrm{new}}$ is an eigenvector for $\mathbf{T}$ that is normalized $a_1 = 1$. Consider the ring homomorphism

$$\lambda_f : \mathbf{T} \to \mathbf{C} \quad (T \mapsto \lambda_f(T) \text{ s.t. } Tf = \lambda_f(T)f \quad \text{for } T \in \mathbf{T})$$

and $\mathbf{I}_f$ denotes its kernel (i.e. annihilator ideal $\mathrm{Ann}_{\mathbf{T}}(f)$ of f). Following Shimura [97], attach to $\mathbf{I}_f$ the quotient

$$\begin{cases} A_f := J_0(N)/\mathbf{I}_f J_0(N) \\ \mathbf{T}_f := \mathbf{T}/\mathbf{I}_f \end{cases}$$

and it is known that A_f is an abelian variety over $\mathbf{Q}$ of dimension $d_f := [\mathbf{Q}(\cdots, a_n, \cdots) : \mathbf{Q}]$ which is equipped with a faithful action of $\mathbf{T}_f$. Since the field $F_f := \mathbf{T}_f \otimes_{\mathbf{Z}} \mathbf{Q}$ is contained in $\mathrm{End}_{\mathbf{Q}}(A_f) \otimes \mathbf{Q}$ and $[F_f : \mathbf{Q}] = d_f$ [25, Prop. 21, §3.7], A_f is an abelian variety of GL_2-type. If we deal with the modular

abelian varieties, there exists two important canonical isomorphisms

$$\begin{cases} \phi_2 : \Gamma(X_0(N)^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X_0(N)^{\mathrm{an}}}) \simeq S_2(\Gamma_0(N))_{\mathbf{C}} \\ \\ \phi_3 : S_2(\Gamma_0(N))^{\vee}_R \simeq \mathbf{T} \otimes_R R \end{cases} \tag{B.12}$$

For the first isomorphism is compatible the action of $\mathbf{T}$ (see [76, Prop. 25.3, §25] and [25, Prop. 19–20, §3.6]) and the second isomorphism holds for any ring R. We also have the similar isomorphisms

$$\begin{cases} \psi_2 : \Gamma(A_f^{\mathrm{an}}, \Omega^{\mathrm{an}}_{A_f^{\mathrm{an}}}) \simeq S_2(\Gamma_0(N))_{\mathbf{C}}[\mathbf{I}_f] \\ \\ \psi_3 : S_2(\Gamma_0(N))_R[\mathbf{I}_f]^{\vee} \simeq \mathbf{T}_f \otimes_R R \end{cases} \tag{B.13}$$

by the restriction of scalars. Then by using (B.11), (B.12), (B.13), we have a diagram as follows:

$$\begin{array}{ccccccc}
H_1(X_0(N)^{\mathrm{an}}, \mathbf{R}) & \xrightarrow[\phi_1]{\simeq} & \Gamma(X_0(N)^{\mathrm{an}}, \Omega^{\mathrm{an}}_{X_0(N)^{\mathrm{an}}})^{\vee} & \xrightarrow[\phi_2]{\simeq} & S_2(\Gamma_0(N))^{\vee}_{\mathbf{C}} & \xrightarrow[\phi_3]{\simeq} & \mathbf{T} \otimes_{\mathbf{Z}} \mathbf{C} \simeq \mathbf{C}^g \\
\| & & \uparrow \simeq & & & & \\
H_1(J_0(N)^{\mathrm{an}}, \mathbf{R}) & \xrightarrow{\simeq} & \Gamma(J_0(N)^{\mathrm{an}}, \Omega^{\mathrm{an}}_{J_0(N)^{\mathrm{an}}})^{\vee} & & \downarrow \mathrm{res.} & & \\
\downarrow e_i & & \downarrow e_i & & & & \\
H_1(A_f^{\mathrm{an}}, \mathbf{R}) & \xrightarrow[\psi_1]{\simeq} & \Gamma(A_f^{\mathrm{an}}, \Omega^{\mathrm{an}}_{A_f^{\mathrm{an}}})^{\vee} & \xrightarrow[\psi_2]{\simeq} & S_2(\Gamma_0(N))_{\mathbf{C}}[\mathbf{I}_f]^{\vee} & \xrightarrow[\psi_3]{\simeq} & \mathbf{T}_f \otimes_{\mathbf{Z}} \mathbf{C} \simeq \mathbf{C}^d
\end{array}$$

In the above diagram, g denotes the dimension of $X_0(N)^{\mathrm{an}}$ (in the same way denotes the dimension of $J_0(N)^{\mathrm{an}}$) and d denotes the dimension of A_f^{an}. Let e be the element of $H_1(X_0(N)^{\mathrm{an}}, \mathbf{R})$ such that it corresponds to the map $\omega \mapsto -\int_0^{\infty} \omega$

under the isomorphism ϕ_1, then we see that

$$\phi_2 \circ \phi_1(e)(f) = L(f, 1) \quad \text{for all } f \in S_2(\Gamma_0(N))_{\mathbf{C}} \tag{B.14}$$

follows from the definition of $L(f, s)$ as a Mellin transform. It is easy to see that if $e_f \in H_1(A_f^{\mathrm{an}}, \mathbf{R})$ is the image of e, we see

$$\psi_2 \circ \psi_1(e_f)(f) = L(f, 1) \quad \text{for all } f \in S_2(\Gamma_0(N))_{\mathbf{C}}[\mathbf{I}_f]. \tag{B.15}$$

This element e is called the "winding element" introduced by B. Mazur in [68, §II.18, pp. 136], and this winding element is crucial to many algorithms for computing with modular abelian varieties. The theory of modular symbols were introduced by B. J. Birch [9] and studied by many others (J. I. Manin, B. Mazur, L. Merel, J. E. Cremona etc. see the References) and recent years, A. Agashe and W. Stein established a beautiful formula:

Theorem B.3.1 (A. Agashe and W. Stein)**.** Assume that $L(A_f, 1) \neq 0$. Let Φ be the composition of $\psi_3 \circ \psi_2 \circ \psi_1$. Then the images $\Phi(H_1(A_f^{\mathrm{an}}, \mathbf{Z})^+)$ and $\Phi(\mathbf{T}_f e_f)$ are lattices in $\mathbf{R}^d \subset \mathbf{C}^d$, and the following formula holds:

$$\frac{L(A_f, 1)}{\Omega_{A_f}} = (\text{constant}) \times [\Phi(H_1(A_f^{\mathrm{an}}, \mathbf{Z})^+) : \Phi(e_f \mathbf{T}_f)]$$

where e_f denotes the image of e in $H_1(A_f^{\mathrm{an}}, \mathbf{R})$.

This formula is the motivation for the author's research. We try to explain the rough proof here according to their papers because the author's purpose is the generalization on this formula. In the later explanation, the following

diagram may be helpful in understanding it.

$$\begin{array}{ccccccc}
H_1(A_f^{\mathrm{an}},\mathbf{R}) & \xrightarrow[\psi_1]{\simeq} & \Gamma(A_f^{\mathrm{an}},\Omega^{\mathrm{an}}_{A_f^{\mathrm{an}}})^\vee & \xrightarrow[\psi_2]{\simeq} & S_2(\Gamma_0(N))_{\mathbf{C}}[\mathbf{I}_f]^\vee & \xrightarrow[\psi_3]{\simeq} & \mathbf{T}_f\otimes_{\mathbf{Z}}\mathbf{C}\simeq\mathbf{C}^d \\
\uparrow & & \uparrow & & \uparrow & & \uparrow \\
H_1(A_f^{\mathrm{an}},\mathbf{R})^+ & \xrightarrow[\psi_1]{\simeq} & \Gamma(A_f^{\mathrm{an}}(\mathbf{R}),\Omega^{\mathrm{an}}_{A_f^{\mathrm{an}}(\mathbf{R})})^\vee & \xrightarrow[\psi_2]{\simeq} & S_2(\Gamma_0(N))_{\mathbf{R}}[\mathbf{I}_f]^\vee & \xrightarrow[\psi_3]{\simeq} & \mathbf{T}_f\otimes_{\mathbf{Z}}\mathbf{R}\simeq\mathbf{R}^d \\
\| & & & & & & \\
H_1(A_f^{\mathrm{an}}(\mathbf{R}),\mathbf{R}) & & & & & &
\end{array}$$

First we define the real period Ω_{A_f} of $A_f^{\mathrm{an}}(\mathbf{R})$ (this is a real Lie group). Let $\mathscr{A}_f$ be the Néron model of A_f defined over $\mathbf{Z}$. It is known that the space $\Gamma(A_f^{\mathrm{an}}(\mathbf{R}),\Omega^{\mathrm{an}}_{A_f^{\mathrm{an}}(\mathbf{R})})$ is regarded as the cotanjent space of $A_f(\mathbf{R})$, and $\Gamma(\mathscr{A}_f,\Omega_{\mathscr{A}_f/\mathbf{Z}})$ and $S_2(\Gamma_0(N))_{\mathbf{Z}}[\mathbf{I}_f]$ are both lattices in $\Gamma(A_f^{\mathrm{an}}(\mathbf{R}),\Omega^{\mathrm{an}}_{A_f^{\mathrm{an}}(\mathbf{R})})$. The space $\Gamma(\mathscr{A}_f,\Omega_{\mathscr{A}_f/\mathbf{Z}})$ is called "the space of Néron differentials" and strictly speaking, the space $S_2(\Gamma_0(N))_{\mathbf{Z}}[\mathbf{I}_f]$ is the inverse image $\psi_2^{-1}(S_2(\Gamma_0(N))_{\mathbf{Z}}[\mathbf{I}_f])$. Then this two lattices define lattices $\Lambda,\tilde{\Lambda}$ in $\mathrm{Lie}(A_f^{\mathrm{an}}(\mathbf{R}))$ respectively. In general, if given a lattice L in $\mathrm{Lie}(A_f^{\mathrm{an}}(\mathbf{R}))$, we have a Haar measure μ_L such that $\mu_L(\mathrm{Lie}(A_f^{\mathrm{an}}(\mathbf{R}))/L)=1$. Note that $H_1(A_f^{\mathrm{an}}(\mathbf{R}),\mathbf{Z})$ is a lattice and it is known that $A_f^{\mathrm{an}}(\mathbf{R})^0$ is expressed as a quotient $\mathrm{Lie}(A_f^{\mathrm{an}}(\mathbf{R}))/H_1(A_f^{\mathrm{an}}(\mathbf{R}),\mathbf{Z})$ where $A_f^{\mathrm{an}}(\mathbf{R})^0$ denotes the connected components of $A_f^{\mathrm{an}}(\mathbf{R})$ containing the identity. Then we have the following diagram:

$$\begin{array}{ccccccc}
H_1(A_f^{\mathrm{an}}(\mathbf{R}),\mathbf{Z}) & \xhookrightarrow[\text{lattice}]{} & \mathrm{Lie}(A_f^{\mathrm{an}}(\mathbf{R})) & \longrightarrow & A_f^{\mathrm{an}}(\mathbf{R})^0 & \longrightarrow & 0 \\
& \nwarrow & \uparrow{\scriptstyle \text{lattice}} & & & & \\
& & L & & & &
\end{array}$$

From this, we see that μ_L induces the Haar measure on $A_f^{\mathrm{an}}(\mathbf{R})^0$ such that

$$\mu_L(A_f^{\mathrm{an}}(\mathbf{R})^0)=\mu_L(\mathrm{Lie}(A_f^{\mathrm{an}}(\mathbf{R}))/L)\times[L:H_1(A_f^{\mathrm{an}}(\mathbf{R}),\mathbf{Z})].$$

Then we see

$$\mu_L(A_f^{\mathrm{an}}(\mathbf{R})) = c_\infty(A_f^{\mathrm{an}}(\mathbf{R})) \times [L : H_1(A_f^{\mathrm{an}}(\mathbf{R}), \mathbf{Z})]$$

where $c_\infty(A_f^{\mathrm{an}}(\mathbf{R}))$ denotes the number of elements of $A_f^{\mathrm{an}}(\mathbf{R})/A_f^{\mathrm{an}}(\mathbf{R})^0$. Now we define the real period $\Omega_{A_f^{\mathrm{an}}}$ as $\Omega_{A_f^{\mathrm{an}}} := \mu_\Lambda(A_f^{\mathrm{an}}(\mathbf{R}))$. Let c_{A_f} be the number of the elements of $\tilde{\Lambda}/\Lambda$ (they call this Manin constant). Then we have the formula:

$$\begin{aligned}
\Omega_{A_f^{\mathrm{an}}} := \mu_\Lambda(A_f^{\mathrm{an}}(\mathbf{R})) &= c_\infty(A_f^{\mathrm{an}}(\mathbf{R})) \times [\Lambda : H_1(A_f^{\mathrm{an}}(\mathbf{R}), \mathbf{Z})] \\
&= c_\infty(A_f^{\mathrm{an}}(\mathbf{R})) \times c_{A_f} \times [\tilde{\Lambda} : H_1(A_f^{\mathrm{an}}(\mathbf{R}), \mathbf{Z})] \\
&= c_\infty(A_f^{\mathrm{an}}(\mathbf{R})) \times c_{A_f} \times [\mathbf{T}_f : H_1(A_f^{\mathrm{an}}(\mathbf{R}), \mathbf{Z})]
\end{aligned}$$

where we get the third equality by the duality $\tilde{\Lambda} := S_2(\Gamma_0(N))_{\mathbf{Z}}[\mathbf{I}_f]^\vee \simeq \mathbf{T}_f$ (apply (B.13) for $R = \mathbf{Z}$). On the other hand, we see that

$$L(A_f, 1) = [\mathbf{T}_f : \Phi(e_f)\mathbf{T}_f]$$

We view $\Phi(e_f)$ as the operator of the left multiplication. Then we have

$$\begin{aligned}
\frac{L(A_f, 1)}{\Omega_{A_f}} &= \frac{[\mathbf{T}_f : \Phi(e_f)\mathbf{T}_f]}{c_\infty(A_f^{\mathrm{an}}(\mathbf{R})) \times c_{A_f} \times [\mathbf{T}_f : H_1(A_f^{\mathrm{an}}(\mathbf{R}), \mathbf{Z})]} \\
&= (\text{constant}) \times [\Phi(H_1(A_f^{\mathrm{an}}(\mathbf{R}), \mathbf{Z})) : \Phi(e_f\mathbf{T}_f)]
\end{aligned}$$

It remains to show $H_1(A_f^{\mathrm{an}}(\mathbf{R}), \mathbf{Z}) \simeq H_1(A_f^{\mathrm{an}}, \mathbf{Z})^+$. This seems to be natural, but

the complete proof is not easy.

APPENDIX C

BSD CONJECTURE

In 1965 B. J. Birch and H. P. F. Swinnerton-Dyer suggested a conjecture about the arithmetic of elliptic curves defied over Q [8]. During the next few years various people gradually extended this conjecture to general settings, J. Tate extended this conjecture to abelian varieties of any dimension defined over any global fields [100], and P. Deligne, A. Beilinson, S. Bloch and K. Kato et al. extended to Grothedieck motives (it is called the "Tamagawa number conjecture" in [12]). As Cassels remarks [21], a fundamental problem of number theory is to find all of rational solutions of a set of polynomial equations with rational coefficients, and moreover, investigate their structure. The conjecture of Birch and Swinnerton-Dyer (the BSD conjecture for short) describes their structure without actually finding the solutions. Thus the BSD conjecture addresses some basic questions in number theory. The BSD conjecture also implies the existence of mysterious relations between the L-functions and various arithmetic groups (Mordell-Weil group, Tate-Shafarevich group) associated to the motives $H^1(A)$ where A is an abelian variety defined over a

global field.

$$L\text{-function} \longleftrightarrow \begin{cases} \text{Mordell-Weil group}, \\ \text{Tate-Shafarevich group}. \end{cases} \tag{C.1}$$

Notation:

Let A be an abelian variety of dimension d defined over a number field K (i.e. a finite extension of the rational number field $\mathbf{Q}$). Let M_K be the set of all places of K

$$\begin{aligned} M_K :&= \{\text{non-archimedean places (finite places)}\} \cup \{\text{archimedean places}\} \\ &= M_K^0 \cup M_K^\infty \\ &= (U \cup S) \cup M_K^\infty \end{aligned}$$

where U denote the non-archimedean places such that A has good reduction in U and S is its complement (S is a finite set). $\bar{K}$ denotes an algebraic closure of K (since a number field K has $\mathrm{char}(\mathbf{K}) = 0$, this field is perfect, then all algebraic extension of K is separable, so $\bar{K}$ is a separable closure). If $v \in M_K$, K_v denotes the completion at v of K. If $v \in M_K^0$, O_v denote the valuation ring of K_v, $\mathfrak{m}_v$ denotes the maximal ideal of O_v, $\kappa(v)$ denotes the residue field of O_v and $N(v)$ denotes the cardinal of $\kappa(v)$. For any field K, G_K denotes the absolute Galois group $\mathrm{Gal}(\bar{K}/K)$ of K.

Let $\mathscr{A}$ be the Néron model of A over O_K, and $\mathscr{A}_{\kappa(v)}$ be the closed fiber $\mathscr{A} \times \kappa(v)$ of $\mathscr{A}$ at v. Let $\mathscr{A}^0$ be the open subgroup of $\mathscr{A}$ whose closed fibers $\mathscr{A}^0_{\kappa(v)}$ at each non-archimedean places v which coincides with the connected component of $\mathscr{A}$ containing the origin [13] (cf. Figure C.1).

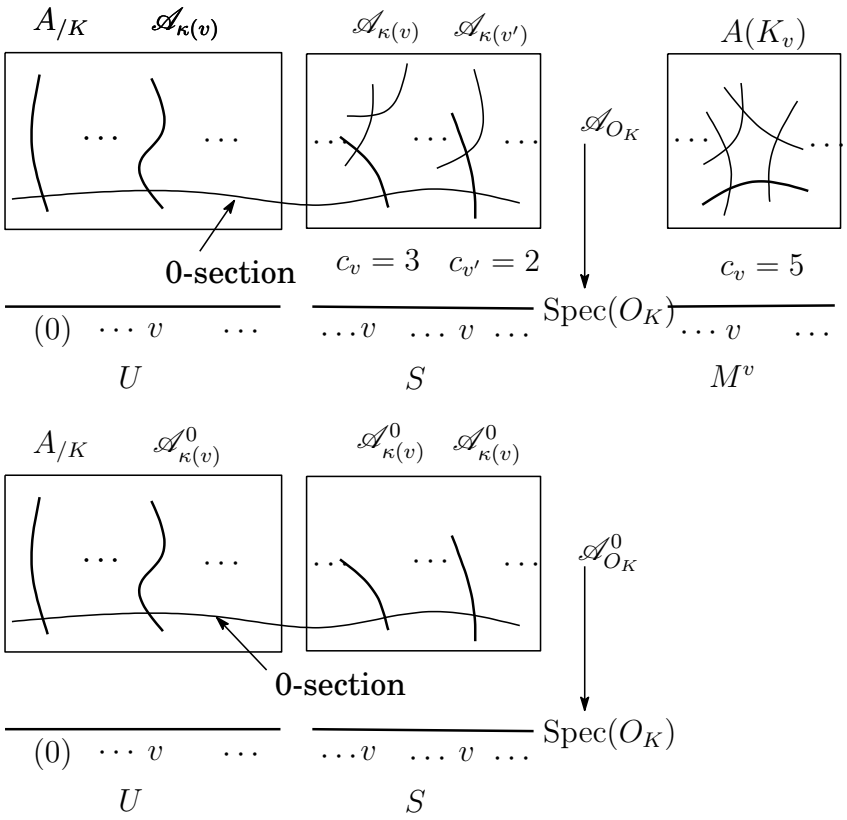


Figure C.1: Néron model.

The local Tamagawa numbers:

At each non-archimedean place $v \in M_K^0$, we define the component group $\Phi_{A,v}$ as follows:

Definition C.0.1. The component group of $\mathscr{A}$ at v is

$$\Phi_{A,v} := \mathscr{A}_{\kappa(v)}/\mathscr{A}^0_{\kappa(v)}.$$

Remark. Since $\mathscr{A}_{\kappa(v)}$ is a smooth commutative group scheme over $\kappa(v)$, it is a disjoint union of one or more connected components, and it is easy to see that $\mathscr{A}^0_{\kappa(v)}$ is a subgroup of $\mathscr{A}_{\kappa(v)}$ (by definition, the connected component $\mathscr{A}^0_{\kappa(v)}$ contains the identity element).

Remark. The $\Phi_{A,v}$ is a finite flat group scheme over $\kappa(v)$ (a finite abelian group equipped with an action of the absolute Galois group $G_{\kappa(v)}$ of $\kappa(v)$), and there is an exact sequence as group schemes

$$0 \to \mathscr{A}^0_{\kappa(v)} \to \mathscr{A}_{\kappa(v)} \to \Phi_{A,v} \to 0.$$

Definition C.0.2 (local Tamagawa numbers)**.** The local Tamagawa number of A at v is

$$c_v := \#\Phi_{A,v}(\kappa(v)).$$

ARITHMETIC INVARIANTS

Mordell-Weil group:

By the Mordell-Weil Theorem, the group $A(K)$ of K-rational points of A is finitely generated. This theorem can be generalized to fields which are finitely generated over their prime field [60, Ch. 6,Thm. 1]. Using elementary group theory, we rephrase this theorem by saying that there are points $e_1, \ldots, e_r$ such that

$$A(K) \simeq A(K)_{\mathrm{tors}} \oplus \mathbf{Z}e_1 \oplus \cdots \oplus \mathbf{Z}e_r.$$

The integer r is called the rank of the abelian variety A, and $A(K)$ is called the *Mordell-Weil group* of A. The torsion subgroup $A(K)_{\mathrm{tors}}$ is a finite abelian group, it can be written as

$$A(K)_{\mathrm{tors}} \simeq \bigoplus_{i=1}^{s} (\mathbf{Z}/m_i\mathbf{Z}),$$

where $m_1, \ldots, m_r$ are integer satisfying $m_i | m_{i+1}$ and $s \leq 2\dim(A)$.

Regulator:

Let $A^\vee$ be the dual abelian variety of A. It is known that $A^\vee(K)$ has the same rank as $A(K)$ ($r := \mathrm{rank}A(K) = \mathrm{rank}A^\vee(K)$) and there is a canonical **Z**-bilinear form

$$h(\ ,\ ) : A(K) \times A^\vee(K) \to \mathbf{R}$$

called the (Néron-Tate) height paring. Let $e_1, \ldots, e_r$ be a **Z**-basis of $A(K)/A(K)_{\mathrm{tors}}$, and let $e_1^*, \ldots, e_r^*$ be a **Z**-bases of $A^\vee(K)/A^\vee(K)_{\mathrm{tors}}$. Then

$$R(A, K) := \left| \det \left(h(e_i, e_j^*) \right) \right| \in \mathbf{R}$$

is independent of the choices of the basis and we call it the *regulator* of A.

Tate-Shafarevich group:

The Tate-Shafarevich group $Ш(A, K)$ is defined by

$$Ш(A, K) := \ker \left(H^1(K, A(K)) \to \prod_v H^1(K_v, A_v(K_v)) \right)$$

where $H^1(K, A(K))$ denotes the Galois cohomology group $H^1(G_K, A(K))$.

Remark. We can regard the group $Ш(A, K)$ as the set of everywhere locally trivial principal homogeneous spaces for A.

L-FUNCTION OF ABELIAN VARIETIES

The l-adic representation on $H^1_{\mathrm{et}}(\bar{A}, \mathbf{Q}_l)$:

A. Grothendieck, M. Artin et al. constructed the l-adic étale cohomology group $H^1_{\mathrm{et}}(\bar{A}, \mathbf{Q}_l)$ [45], [46]. $H^1_{\mathrm{et}}(\bar{A}, \mathbf{Q}_l)$ is $\mathbf{Q}_l$-vector space of dimension $2d$ where $\bar{A}$ denotes the fiber product $A \times_K \bar{K}$. The absolute Galois group G_K of K acts

on it (for the detail of the action, see [**?**, §1]). Then $H^1_{\text{et}}(\bar{A}, \mathbf{Q}_l)$ determines an l-adic representation of G_K

$$\rho_l : G_K \to \text{Aut}(H^1_{\text{et}}(\bar{A}, \mathbf{Q}_l)) \simeq \text{GL}_{2d}(\mathbf{Q}_l). \tag{C.2}$$

The theory of the étale cohomology may be unfamiliar with non-experts, we can construct more explicitly the above representation by using a $\mathbf{Z}_l$-free module $T_l(A)$ called the "Tate module of A" substitutions of the étale cohomology group when A is an abelian variety. It is known that

$$A_l(\bar{K}) := \ker\big(l : A(\bar{K}) \to A(\bar{K})\big)$$

has order l^{2d} for any l' dividing l, so that $A_l(\bar{K})$ is a free $\mathbf{Z}/l\mathbf{Z}$-module of rank $2d$. Moreover the absolute Galois group G_K acts on $A_l(\bar{K})$ (since if $lP = 0$, then $lP^\sigma = (lP)^\sigma = 0$ for any $P \in A_l(\bar{K})$ and $\sigma \in \text{Gal}(\bar{K}/K)$). If we chose a basis for $A_l(\bar{K})$, we obtain a representation

$$G_K \to \text{Aut}(A_l(\bar{K})) \simeq \text{GL}_{2d}(\mathbf{Z}/l\mathbf{Z}).$$

This representation has coefficients of ring of positive characteristic. We lift this coefficients to a ring of characteristic 0.

Definition C.0.3 (Tate module)**.** Fix a prime $l \neq \text{char}(K)$. We define the Tate module $T_l(A)$ of A by

$$T_l(A) := \varprojlim_n A_{l^n}(\bar{K}),$$

the inverse limit being taken with respect to the natural maps

$$A_{l^{n+1}}(\bar{K}) \xrightarrow{l} A_{l^n}(\bar{K}).$$

$T_l(A)$ is a free $\mathbf{Z}_l$-module of rank $2d$ with an action of G_K. If we chose a $\mathbf{Z}_l$-basis for $T_l(A)$, we obtain a representation

$$G_K \to \mathrm{Aut}(T_l(A)) \simeq \mathrm{GL}_{2d}(\mathbf{Z}_l),$$

and two-dimensional representation of G_K acting on $V_l(A) := \mathbf{Q}_l \otimes_{\mathbf{Z}_l} T_l(A)$

$$G_K \to \mathrm{Aut}(V_l(A)) \simeq \mathrm{GL}_{2d}(\mathbf{Q}_l) \tag{C.3}$$

by the extension of the scalar (take the natural inclusion $\mathbf{Z}_l \hookrightarrow \mathbf{Q}_l$). What is the relation between two representations (C.2) and (C.3)? It is known that $\mathbf{Z}_l$-module $H^1_{\mathrm{et}}(\bar{A}, \mathbf{Z}_l)$ is isomorphic to the dual module of $T_l(A)$, and more important, the isomorphism is compatible with the action of G_K, that is there is

$$H^1_{\mathrm{et}}(\bar{A}, \mathbf{Z}_l) \simeq \mathrm{Hom}_{\mathbf{Z}_l}(T_l(A), \mathbf{Z}_l) \quad \text{as } \mathbf{Z}_l[G_K]\text{-modules.} \tag{C.4}$$

By using (C.4), we see that the representation (C.2) is a contragredient representation of (C.3) on the dual space $V_l(A)^\vee$ of $V_l(A)$.

L-function of $H^1_{\mathrm{et}}(\bar{A}, \mathbf{Q}_l)$:

Given a non-archimedean place $v \in M_K^0$, one defines a characteristic polynomial

$$P_v(A, T) := \det\left(1 - T\rho_l(\sigma_{\mathfrak{p}}^{-1}) \mid H^1_{\mathrm{et}}(\bar{A}, \mathbf{Q}_l)^{I_{\mathfrak{p}}}\right), \tag{C.5}$$

where $I_{\mathfrak{p}}$ and $\sigma_{\mathfrak{p}}$ denote respectively the inertia group and a Frobenius element ($\rho_l(\sigma_{\mathfrak{p}}^{-1})$ is usually called the geometric Frobenius) of some prime ideal $\mathfrak{p}$ of $\bar{K}$ lying over v, and

$$H^1_{\mathrm{et}}(\bar{A}, \mathbf{Q}_l)^{I_{\mathfrak{p}}} := \left\{v \in H^1_{\mathrm{et}}(\bar{A}, \mathbf{Q}_l) \mid \rho_l(g)v = v \quad \text{for all } g \in I_{\mathfrak{p}}\right\}.$$

Remark. It is easy to see that the characteristic polynomial $P_v(A,T)$ is independent of the choice of $\mathfrak{p}$ and $\sigma_{\mathfrak{p}}$ follows by a straightforward verification from the conjugacy under G_K of the prime ideals lying over v.

Remark. If $v \in U$, the action of the inertia $I_{\mathfrak{p}}$ on $H^1_{\text{et}}(\bar{A}, \mathbf{Q}_l)$ is trivial, and the characteristic polynomial satisfies

$$\begin{aligned} P_v(A,T) =& \det\left(1 - T\rho_l(\sigma_{\mathfrak{p}}^{-1}) \mid H^1_{\text{et}}(\bar{A}, \mathbf{Q}_l)\right) \\ =& \det\left(1 - T\rho_l(\sigma_{\mathfrak{p}}^{-1}) \mid H^1_{\text{et}}(\overline{A_{\kappa(v)}}, \mathbf{Q}_l)\right), \end{aligned}$$

where $\overline{A_{\kappa(v)}} = A_{\kappa(v)} \times \overline{\kappa(v)}$. It is known that $P_v(A,T)$ belongs to $\mathbf{Z}[T]$ and is independent of the choice of $v \nmid l$ by the Weil conjecture ([53]).

Now we define the (Hasse-Weil) L-function of $H^1_{\text{et}}(\bar{A}, \mathbf{Q}_l)$:

Definition C.0.4 (Hasse-Weil L-function)**.**

$$L(H^1_{\text{et}}(\bar{A}, \mathbf{Q}_l), s) := \prod_{v \in M_K^0} \frac{1}{P_v(A, N(v)^{-s})}.$$

We denote $L(H^1_{\text{et}}(\bar{A}, \mathbf{Q}_l), s)$ by $L(A,s)$ for short.

MEASURES ON LOCALLY COMPACT GROUPS

For any $v \in M_K$, since $\mathrm{Lie}(A)(K_v)$ is a locally compact group, there exists a Haar measure μ_v on $\mathrm{Lie}(A)(K_v)$ up to $K^\times$. If $v \in U$, we normalize the measure by $\mu_v(\mathrm{Lie}(A)(O_v)) = 1$. That is we determine the unique product measure $\mu := \prod_{v \in M_K} \mu_v$ on $\mathrm{Lie}(A)(\mathbf{A}_K)$ where $\mathrm{Lie}(A)(\mathbb{A}_K) := \prod_{v \in M_K} \mathrm{Lie}(A)(K_v)$ is the adele group

$$\left\{ (a_v) \in \prod_{v \in M_K} \mathrm{Lie}(A)(K_v) \;\middle|\; a_v \in \mathrm{Lie}(\mathscr{A})(O_v) \quad \text{for almost all } v \in M_K \smallsetminus M_K^\infty \right\}.$$

The Haar measure μ_v on $\mathrm{Lie}(A)(K_v)$ induces a Haar measure on the compact group $A(K_v)$ (we denote this Haar measure on $A(K_v)$ by μ'_v) which is characterized as follows. Since $\mathrm{char}(K_v) = 0$, we have an injective exponential map $\exp : W \to A(K_v)$ for a sufficiently small compact neighborhood W of 0 in $\mathrm{Lie}(A)(K_v)$. We have the Haar measure μ'_v on $A(K_v)$ which is characterized by

$$\mu_v(W) = \mu'_v(\exp(W)). \tag{C.6}$$

In particular, if $v \in M_K^0$, the Haar measure μ'_v on $A(K_v)$ is characterized by

$$\mu_v(\mathrm{Lie}(\mathscr{A})(\mathfrak{m}_v^n)) = \mu'_v(\mathscr{A}(\mathfrak{m}_v^n)) \tag{C.7}$$

for $n \geq 1$, where

$$\begin{cases} \mathrm{Lie}(\mathscr{A})(\mathfrak{m}_v^n) := \ker\left(\mathrm{Lie}(\mathscr{A})(O_v) \to \mathrm{Lie}(\mathscr{A})(O_v/\mathfrak{m}_v^n)\right) \\ \mathscr{A}(\mathfrak{m}_v^n) := \ker\left(\mathscr{A}(O_v) \to \mathscr{A}(O_v/\mathfrak{m}_v^n)\right). \end{cases}$$

We denote the Haar measure μ'_v on $A(K_v)$ by μ_v in the same symbol.

THE CONJECTURE

Definition C.0.5 (Variant of the L-function)**.**

$$L_{\mathrm{BSD}}(A, s) := \prod_{v \in M_K^\infty} \frac{1}{\mu_v(A(K_v))} \times \prod_{v \in S} \frac{1}{c_v} \times L(A, s)$$

Conjecture C.0.1 (The conjecture of Birch and Swinnerton-Dyer)**.** Let

$$L_{\mathrm{BSD}}(A, s) = I_A(s-1)^{\rho(A)} + O(|s-1|^{\rho(A)+1})$$

be the Laurent expansion of $L_{\mathrm{BSD}}(A, s)$ at $s = 1$, then

1. $\rho(A) = \mathrm{rank}(A(K))$,

2. $|Ш(A, K)| < \infty$,

3. $I_A = \dfrac{|Ш(A, K)|}{|A(\mathbf{K})_{\mathrm{tors}}||A^{\vee}(\mathbf{K})_{\mathrm{tors}}\}|} R(A, K)$.

C.1 BSD Conjecture for Elliptic Curves

In this Appendix, we explain Birch and Swinnerton-Dyerwhat conjecture using a short Weierstrass equation. Let E be an elliptic curve defined over $\mathbf{Q}$ given by the minimal Weierstrass equation

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (a_i \in \mathbf{Q}). \tag{C.8}$$

Let $\mathscr{E}$ be the global Néron model of E over $\mathbf{Z}$. The natural reduction map induces a morphism

$$\varphi_p : E(\mathbf{Q}_p) \to \mathscr{E}_{\mathbf{F}_p}(\mathbf{F}_p),$$

where $\mathscr{E}_{\mathbf{F}_p}$ denotes the closed fiber of $\mathscr{E}$ at p. Next we define two subgroups (filtration) of $E(\mathbf{Q}_p)$ as follows:

$$E^0(\mathbf{Q}_p) := \left\{ P \in E(\mathbf{Q}_p) \;\middle|\; \varphi_p(P) \in \mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p}(\mathbf{F}_p) \right\}$$

where $\mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p} := \mathscr{E}_{\mathbf{F}_p} \smallsetminus \{\text{any singular points}\}$ and

$$E^1(\mathbf{Q}_p) := \ker \left(\varphi_p|_{E^0(\mathbf{Q}_p)} \;\middle|\; E^0(\mathbf{Q}_p) \to \mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p}(\mathbf{F}_p) \right).$$

Proposition C.1.1. There is an exact sequence of abelian groups as follows:

$$0 \to E^1(\mathbf{Q}_p) \to E^0(\mathbf{Q}_p) \to \mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p}(\mathbf{F}_p) \to 0.$$

Proof. Rough Sketch. Since we defined $E^1(\mathbf{Q}_p)$ to be the kernel of $\varphi_p|_{E^0(\mathbf{Q}_p)}$, it remains to show that the restricted reduction map $\varphi_p|_{E^0(\mathbf{Q}_p)}$ is surjective. This will follow from Hensel's lemma and the completeness of $\mathbf{Q}_p$. For the detail see [85, VII, Prop. 2.1]. □

Definition C.1.1 (local Tamagawa number)**.**

$$c_p := (E(\mathbf{Q}) : E^0(\mathbf{Q}_p))$$

Remark. We have $c_p = 1$ if $p \in U$.

Next we define the real period $\mu_\infty(E(\mathbf{R}))$. We choose a generator $\omega_E \pmod{\mathbf{Z}^\times}$ of $\Gamma(\mathscr{E}, \Omega_{\mathscr{E}/\mathbf{Z}})$ which is a free $\mathbf{Z}$-module of rank 1. This is called the "Néron differential". The ω_E defines a holomorphic differential on the complex torus E^{an} (E^{an} denotes an analytic space defined by E) and $E^{\mathrm{an}}(\mathbf{R})$ is regarded as the 1-homology cycle $[E^{\mathrm{an}}(\mathbf{R})] \in H_1(E^{\mathrm{an}}, \mathbf{Z})$. We define the "real period" of E by the period integral:

Definition C.1.2.

$$\mu_\infty(E(\mathbf{R})) := \int_{[E(\mathbf{R})]} \omega_E. \qquad \text{we call this "Real period"}$$

Proposition C.1.2. At each non-archimedean place $p \in M^0_{\mathbf{Q}}$

$$\mu_p(E(\mathbf{Q}_p)) = c_p \times \frac{\#\mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p}(\mathbf{F}_p)}{p}.$$

Proof. Decompose

$$\begin{aligned}\mu_p(E(\mathbf{Q}_p)) &= (E(\mathbf{Q}_p) : E^1(\mathbf{Q}_p)) \times \mu_p(E^1(\mathbf{Q}_p)) \\ &= (E(\mathbf{Q}_p) : E^0(\mathbf{Q}_p)) \times (E^0(\mathbf{Q}_p) : E^1(\mathbf{Q}_p)) \times \mu_p(E^1(\mathbf{Q}_p)),\end{aligned}$$

By Def. 3.2, the first quantity satisfies

$$(E(\mathbf{Q}_p) : E^0(\mathbf{Q}_p)) = c_p,$$

and by Prop. C.1.1, the second quantity satisfies

$$(E^0(\mathbf{Q}_p) : E^1(\mathbf{Q}_p)) = \#\mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p}(\mathbf{F}_p).$$

So it remains to show that $\mu_p(E^1(\mathbf{Q}_p)) = 1/p$, we use a lemma from the formal group theory:

Lemma C.1.3. There exists an isomorphism of additive groups:

$$p\mathbf{Z}_p \xrightarrow{\sim} E^1(\mathbf{Q}_p). \tag{C.9}$$

Proof. If an elliptic curve E is given by a Weierstrass equation with coefficients in $\mathbf{Z}_p$, we can construct the formal group associated to E, denoted the formal group by $\hat{E}$, and there are power series $1/w(z) \in \mathbf{Z}[\![z]\!]$ such that the map

$$\hat{E}(p\mathbf{Z}_p) \to E^1(\mathbf{Q}_p) \quad \left(z \mapsto \left(\frac{z}{w(z)}, -\frac{1}{w(z)}\right)\right)$$

is an isomorphism of $p\mathbf{Z}_p$ onto $E^1(\mathbf{Q}_p)$. (for the explicit construction of $w(z)$, see [85, §1, IV]). By formal logarithm map, we have an isomorphism $\hat{E}(p\mathbf{Z}_p) \xrightarrow{\sim} p\mathbf{Z}_p$ [85, Thm. 6.4 (b), IV]. □

From above lemma, we see $\mu_p(E^1(\mathbf{Q}_p)) = \mu_p(p\mathbf{Z}_p)$, and μ_p is a Haar measure

on $\mathbf{Z}_p$ for which $\mathbf{Z}_p$ has measure 1 and therefore $p\mathbf{Z}_p$ has measure $1/(\mathbf{Z}_p : p\mathbf{Z}_p) = 1/p$. □

Remark. Since $\mathbf{Z}_p$ is complete, the power series $F(x, y)$ and $i(x)$ (these power series called the formal group law of $\hat{E}$ [85, Def. A, IV]) converge in $\mathbf{Z}_p$ for $x, y \in p\mathbf{Z}_p$ and $\hat{E}$ makes $p\mathbf{Z}_p$ into a group.

Remark. If $p \in U$, then $c_p = 1$ and $\mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p} = \mathscr{E}_{\mathbf{F}_p}$ so we have a simple formula

$$\mu_p(E(\mathbf{Q}_p)) = \frac{\#\mathscr{E}_{\mathbf{F}_p}(\mathbf{F}_p)}{p}. \tag{C.10}$$

Proposition C.1.4. For each $p \in M_K^0$, let $P_p(E, T) \in \mathbf{Z}[T]$ be the characteristic polynomial. Then

$$P_p(E, p^{-1}) = \frac{\#\mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p}(\mathbf{F}_p)}{p}.$$

Proof. Let $\varphi : \mathscr{E}^{\bar{\mathrm{sm}}}_{\mathbf{F}_p} \to \mathscr{E}^{\bar{\mathrm{sm}}}_{\mathbf{F}_p}$ be the p-th Frobenius map where $\mathscr{E}^{\bar{\mathrm{sm}}}_{\mathbf{F}_p}$ denotes $\mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p} \times \bar{\mathbf{F}_p}$. This map induces

$$\varphi^* : H^i_{\mathrm{et}}(\mathscr{E}^{\bar{\mathrm{sm}}}_{\mathbf{F}_p}, \mathbf{Q}_l) \to H^i_{\mathrm{et}}(\mathscr{E}^{\bar{\mathrm{sm}}}_{\mathbf{F}_p}, \mathbf{Q}_l) \quad (0 \le i \le 2)$$

where $H^i_{\mathrm{et}}(\mathscr{E}^{\bar{\mathrm{sm}}}_{\mathbf{F}_p}, \mathbf{Q}_l)$ is $\mathbf{Q}_l$-vector space of dimension two. By using the étale cohomology theory (Lefshetz fixed point theorem), we see that

$$\#\mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p} = \mathrm{Tr}(\varphi^* | H^0_{\mathrm{et}}(\mathscr{E}^{\bar{\mathrm{sm}}}_{\mathbf{F}_p}, \mathbf{Q}_l)) - \mathrm{Tr}(\varphi^* | H^1_{\mathrm{et}}(\mathscr{E}^{\bar{\mathrm{sm}}}_{\mathbf{F}_p}, \mathbf{Q}_l)) + \mathrm{Tr}(\varphi^* | H^2_{\mathrm{et}}(\mathscr{E}^{\bar{\mathrm{sm}}}_{\mathbf{F}_p}, \mathbf{Q}_l)). \tag{C.11}$$

It is easy to see that

$$\begin{cases} \det\left(1 - T\varphi^* | H^0_{\mathrm{et}}(\mathscr{E}^{\bar{\mathrm{sm}}}_{\mathbf{F}_p}, \mathbf{Q}_l)\right) = 1 - T \\ \det\left(1 - T\varphi^* | H^2_{\mathrm{et}}(\mathscr{E}^{\bar{\mathrm{sm}}}_{\mathbf{F}_p}, \mathbf{Q}_l)\right) = 1 - pT, \end{cases}$$

Therefore we see that

$$\begin{cases} \mathrm{Tr}\left(\varphi^* | H^0_{\mathrm{et}}(\bar{\mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p}}, \mathbf{Q}_l)\right) = 1 \\ \mathrm{Tr}\left(\varphi^* | H^2_{\mathrm{et}}(\bar{\mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p}}, \mathbf{Q}_l)\right) = \det\left(\varphi^* | H^2_{\mathrm{et}}(\bar{\mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p}}, \mathbf{Q}_l)\right) = p. \end{cases} \tag{C.12}$$

From (C.11) and (C.12), we obtain

$$\mathrm{Tr}\left(\varphi^* | H^1_{\mathrm{et}}(\bar{\mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p}}, \mathbf{Q}_l)\right) = 1 - \#\mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p}(\mathbf{F}_p) + p. \tag{C.13}$$

Next, we calculate

$$\begin{aligned} P_p(E,T) =& \det\left(1 - T\varphi^* | H^1_{\mathrm{et}}(\bar{\mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p}}, \mathbf{Q}_l)\right) \\ =& 1 - \mathrm{Tr}\left(\varphi^* | H^1_{\mathrm{et}}(\bar{\mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p}}, \mathbf{Q}_l)\right) T + \det\left(\varphi^* | H^2_{\mathrm{et}}(\bar{\mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p}}, \mathbf{Q}_l)\right) T^2 \\ =& 1 - (1 - \#\mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p}(\mathbf{F}_p) + p)T + pT^2. \end{aligned}$$

Therefore we see that

$$\begin{aligned} P_p(E, p^{-1}) &= 1 - (\frac{1 - \#\mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p}(\mathbf{F}_p) + p}{p}) + \frac{1}{p} \\ &= \frac{\#\mathscr{E}^{\mathrm{sm}}{}_{\mathbf{F}_p}(\mathbf{F}_p)}{p}. \end{aligned}$$

□

Remark. If $p \in S$, we can determine

$$P_p(E,T) := \begin{cases} 1 - T & E \text{ has split multiplicative reduction} \\ 1 + T & E \text{ has non-split multiplicative reduction} \\ 1 & E \text{ has additive reduction.} \end{cases}$$

Then we see that

$$P_p(E, p^{-1}) = \frac{\#\mathscr{E}^{\mathrm{sm}}_{\mathbf{F}_p}(\mathbf{F}_p)}{p}.$$

Indeed, it is known that $\#\mathscr{E}^{\mathrm{sm}}{}_{\mathbf{F}_p}(\mathbf{F}_p)/p = p-1, p+1, p$ respectively.

By the Wiles and Breuil, Conrad, Diamond, Taylor's theorem, the Hasse-Weil L-function $L(E,s)$ has an analytic continuation to an entire function of order one satisfying a functional equation, so we can consider the Laurent expansion of $L(E,s)$ (resp. $L_{\mathrm{BSD}}(E,s)$) at $s=1$. In the these situations, the conjecture of Birch and Swinnerton-Dyer is as follows:

Conjecture C.1.1. Let

$$L_{\mathrm{BSD}}(E,s) = I_E(s-1)^{\rho(E)} + O(|s-1|^{\rho(E)+1})$$

be the Laurent expansion of $L_{\mathrm{BSD}}(E,s)$ at $s=1$, then

1. $\rho(E) = \mathrm{rank}(E(\mathbf{Q}))$

2. $Ш(E) < \infty$

3. $I_E = \dfrac{|Ш(E)|}{|E(\mathbf{Q})_{\mathrm{tors}}|^2} R(E)$.

Remark. The quantity I_E is regarded as follows:

$$\begin{aligned} I_E &= \prod_{p \in M_{\mathbf{Q}}} \frac{1}{\mu_p(E(\mathbf{Q}_p))} \\ &= \frac{1}{\mu_\infty(E(\mathbf{R}))} \times \prod_{p \in S} \frac{1}{\mu_p(E(\mathbf{Q}_q))} \times \prod_{p \in U} \frac{1}{\mu_p(E(\mathbf{Q}_p))}. \end{aligned}$$

C.2 COMPUTATION USING PARI/GP

In the original article [8], B. J. Birch and H. P. F. Swinnerton-Dyer investigated a behavior of an infinite product of a "p-adic density"

$$f(x) := \prod_{p \leq x} \frac{\#\mathscr{E}_{\mathbf{F}_p}(\mathbf{F}_p)}{p}$$

(the definition of $\mathscr{E}$ in §3) of special kinds of elliptic curves ($y^2 = x^3 - Dx$ ($D \in \mathbf{Q}$)) with their computer (EDSAC II), and they conjectured:

Conjecture C.2.1. [8, pp. 79] If $r := \mathrm{rank}(E(\mathbf{Q}))$, then there are non-zero constant C, C' such that

$$f(x) \sim C(\log(x))^r \quad (\text{as } x \to \infty), \tag{C.14}$$

and

$$L(E, s) \sim C'(s-1)^r \quad (\text{as } s \to 1). \tag{C.15}$$

Remark. It seems that the precise mathematical relation between the above two statements ((C.14) and (C.15)) is not known.

In this section, we check this statement numerically (C.14) to some special kind of elliptic curves using the computer program Pari which is specifically designed for computation in algebraic number theory. We consider the elliptic curves

$$E_a : y^2 = x^3 + ax \quad \text{where } a = -3, +3, +14, -82$$

over $\mathbf{Q}$. It is known that the Mordell-Weil rank of these elliptic curves E_a (denote by $r_a := \mathrm{rank}(E_a(\mathbf{Q}))$) is zero, one, two and three respectively [54, pp. 17]. We will compute $f(x)$ using the Pari (GP calculator) and make a table. (cf. Figure C.2.)

EXAMPLE

Step 1:

In this example, we deal with the elliptic curve $E_3 : y^2 = x^3 + 3x$. First we prepare a script (i.e. a program written in a language of GP) in order to make a table, and save as a name "ell.gp" in the working directory.

```
{
e = ellinit([0,0,0,3,0]);


x = 1.0;
forprime(i = 5,500000,
N_p = i+1-ellap(e,i);
x = x*(N_p/i);
write("ell.dat",i," "N_p," "x);
)
}
```

The first line

```
e = ellinit([0,0,0,3,0])
```

defines the elliptic curve $y^2 = x^3 + 3x$. If one write

```
e = ellinit([2,3,4,5,6])
```

then this defines the elliptic curve $y^2 + 2xy + 4y = x^3 + 3x^2 + 5x + 6$. The third line's command

```
forprime(i = a, b, seq)
```

means that the formal variable i ranging over the prime numbers between a to b (including a and b if they are prime), and the seq is evaluated. For example, one write

```
forprime(i = 2, 12,
print(i);
if (p == 3, p = 6);
)
}
```

then the GP calculator output

```
2
3
7
11.
```

Step2:

To start Pari on your computer, type

```
> gp
```

Next to read the above file “ell.gp”, type

```
> read("ell.gp")
```

(This “read” command reads in the file whose name is “ell.gp”) and GP calculator does the script “ell.gp” and make a file “ell.dat” in the working directory. (cf. Figure C.2.)

Remark. One can check the correctness of the table in [54, pp. 17].

x	$N(x)$	$\prod'_{p\leq x}\frac{N(p)}{p}$	x	$N(x)$	$\prod'_{p\leq x}\frac{N(p)}{p}$
5	10	$2.0000000000\cdots$	499879	499880	$22.0861713223\cdots$
7	8	$2.2857142857\cdots$	499883	499884	$22.0862155050\cdots$
11	12	$2.4935064935\cdots$	499897	500256	$22.1020766751\cdots$
13	20	$3.8361638361\cdots$	499903	499904	$22.1021208878\cdots$
17	26	$5.8670741023\cdots$	499927	499928	$22.1021650985\cdots$
19	20	$6.1758674761\cdots$	499943	499944	$22.1022093079\cdots$
23	24	$6.4443834534\cdots$	499957	500036	$22.1057017573\cdots$
29	26	$5.7777230961\cdots$	499969	501044	$22.1532319630\cdots$
31	32	$5.9641012605\cdots$	499973	498578	$22.0914211080\cdots$
37	40	$6.4476770384\cdots$	499979	499980	$22.0914652927\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

Figure C.2: This table is “ell.dat” file, the symbol $\prod'$ means the product of exceptional primes. (2 and those dividing the discriminant)

Figure C.3: Rank 0 (upper) and Rank 1 (lower)

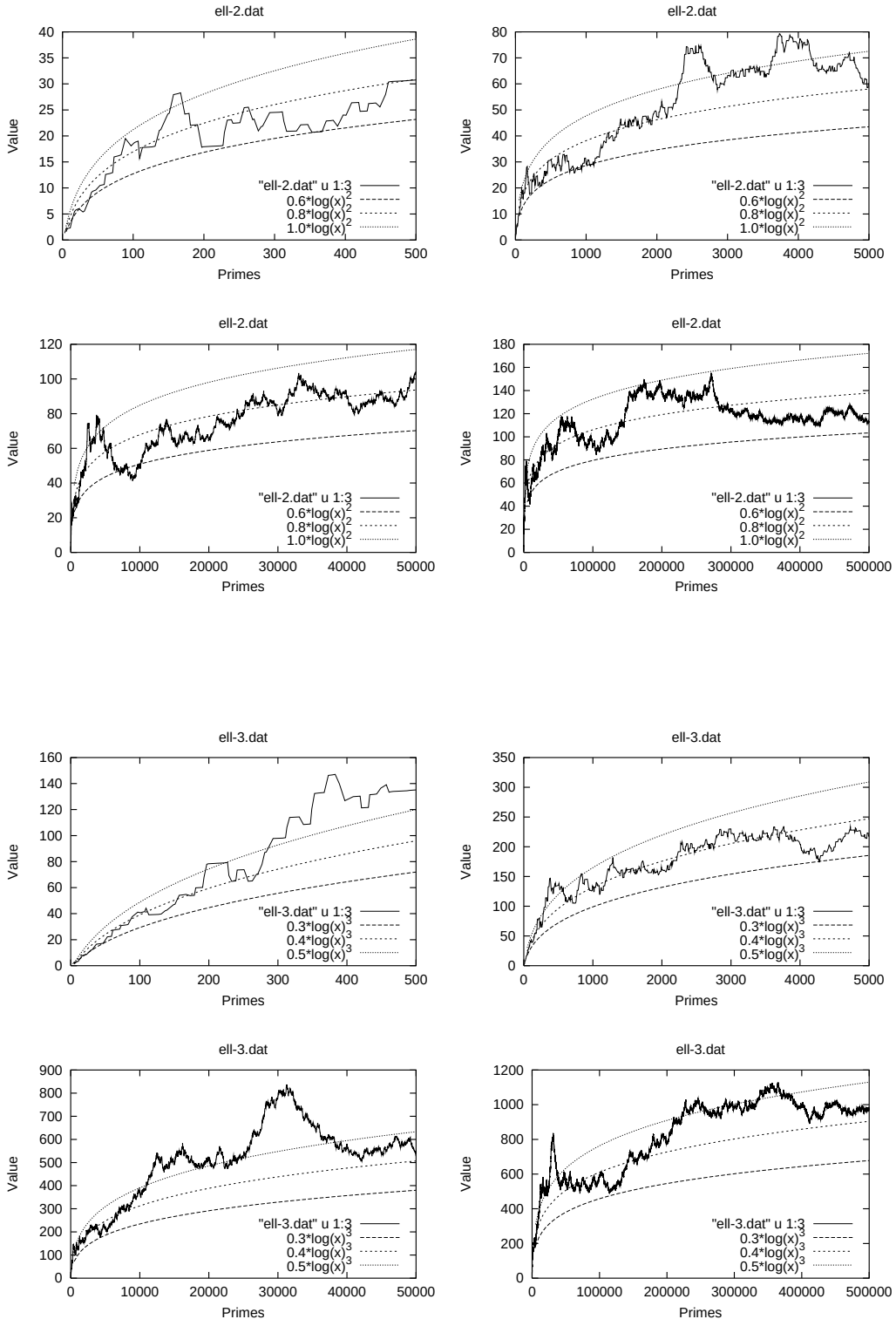
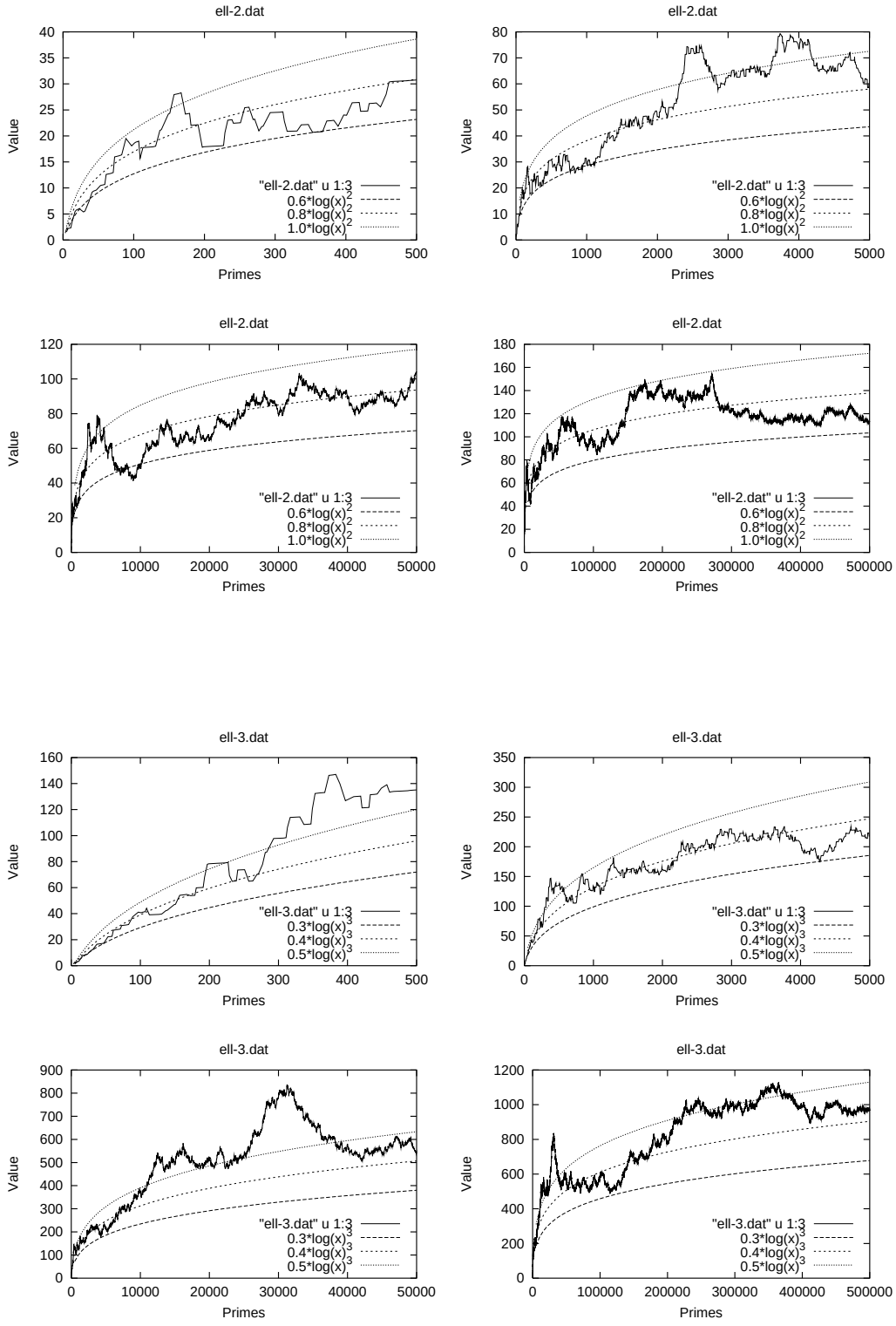


Figure C.4: Rank 3 (upper) and Rank 4 (lower)



BIBLIOGRAPHY

[1] A. Barenghi, G. Bertoni, L. Breveglieri and G. Pelosi, "A FPGA Coprocessor for the Cryptographic Tate Pairing over Fp", Information Technology: New Generations, Fifth International Conference on. IEEE, pp. 112–119, 2008.

[2] P. S. L. M. Barreto, S. D. Galbraith, C. ÓhÉigeartaigh and M. Scott, "Efficient pairing computation on supersingular Abelian varieties", Designs, Codes and Cryptography, Volume 42(3), pp. 239–271, 2007.

[3] P. S. L. M. Barreto and H. Y. Kim, "Fast hashing onto elliptic curves over fields of characteristic 3", Cryptology ePrint Archive, Report 2001/098, 2001.

[4] P. S. L. M. Barreto, H. Y. Kim, B. Lynn and M. Scott, "Efficient Algorithms for Pairing-Based Cryptosystems", in M. Yung (ed.): *CRYPTO 2002*, Lecture Notes in Computer Science Volume 2442, Springer-Verlag, pp. 354–369, 2002.

[5] P. S. L. M. Barreto, B. Lynn and M. Scott, "Efficient Implementation of Pairing-Based Cryptosystems", Journal of Cryptology, Volume 17(4), pp. 321–334, 2004.

[6] P. S. L. M. Barreto and J. F. Voloch, "Efficient Computation of Roots in Finite Fields", *Designs, Codes and Cryptography*, Volume 39(2), pp. 275–280, 2006.

[7] J. L. Beuchat, H. Doi, K. Fujita, A. Inomata, P. Ith, A. Kanaoka, M. Katouno, M. Mambo, E. Okamoto, T. Okamoto, T. Shiga, M. Shirase, R. Soga, T. Takagi, A. Vithanage and H. Yamamoto, "FPGA and ASIC Implementations of the η_T Pairing in Characteristic Three", Computers and Electrical Engineering, Volume 36(1), pp. 73–87, 2010.

[8] B. J. Birch and H. P. F. Swinnerton-Dyer, "Notes on elliptic curves. II", Journal für die reine und angewandte Mathematik, Volume 165(218), pp. 79–108, 1965.

[9] B. J. Birch, "it Elliptic curves over Q", A progress report, 1969 Number Theory Institute (Proc. Sympos. Pure Math., Volume **20**, State Univ. New York, Stony Brook, N.Y., 1969), Amer. Math. Soc., Providence, R.I., pp. 396–400, 1971.

[10] I. Blake, G. Seroussi and N. Smart, *Elliptic Curves in Cryptography*, London Mathematical Society Lecture Note Series 265, Cambridge University Press, 1999.

[11] I. Blake, G. Seroussi and N. Smart, *Advances in Elliptic Curve Cryptography*, London Mathematical Society Lecture Note Series 317, Cambridge University Press, 2005.

[12] S. Bloch and K. Kato, "L-functions and Tamagawa numbers of motives", The Grothendieck Festschrift, Volume I, Birkháuser Boston, Boston, MA, pp. 333–400, 1990..

[13] S. Bosch, W. Lutkebohmert, M. Raynaud, *Néron models*, Springer-Verlag, 1990.

[14] D. Boneh and X. Boyen, "Efficient Selective-ID Secure Identity-Based EncryptionWithout Random Oracles" in C. Cachin and J. Camenisch (eds.): *Advances in Cryptology — EUROCRYPT 2004*, Volume 3027 of Lecture Notes in Computer Science, pp. 223–238, Springer, 2004.

[15] D. Boneh, G. Di Crescenzo, R. Ostrovsky and G. Persiano, "Public key encryption with keyword search", in C. Cachin and J. L. Camenisch (eds.): *EUROCRYPT 2004*, LNCS 3027, Springer-Verlag, pp. 506–522, 2004.

[16] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing", in J. Kilian (ed.): *CRYPTO 2001*, LNCS 2139, Springer-Verlag, pp. 213–229, 2001.

[17] D. Boneh, C. Gentry and B.Waters, "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys", in V. Shoup (ed.): CRYPTO 2005, LNCS 3621, Springer-Verlag, pp. 258–275, 2005.

[18] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing", in C. Boyd (ed.): Proceedings of Asiacrypt 2001, LNCS 2248, Springer-Verlag, pp. 514–532, 2001.

[19] X. Boyen and L. Martin, Identity-Based Cryptography Standard (IBCS) #1: Supersingular Curve Implementations of the BF and BB1 Cryptosystems, Internet Engineering Task Force, RFC5091 (Informational), http://www.ietf.org/rfc/rfc5091.txt, 2007.

[20] J. A. Buchmann, *INTRODUCTION TO CRYPTOGRAPHY*, Springer-Verlag, 2004.

[21] J. W. S. Cassels, "Arithmetic on an elliptic curve", Proc. Internat. Congr. Mathematicians (Stockholm, 1962), Inst. Mittag-Leffler, Djursholm, pp. 234–246, 1963.

[22] S. Chatterjee and P. Sanjit, *Identity-based encryption*, Springer, 2011.

[23] C. Cocks, "An identity based encryption scheme based on quadratic residues" in B. Honary (ed.): Cryptography and Coding, Lecture Notes in Computer Science, Volume 2260, pp. 360–363, 2001.

[24] H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen and F. Vercauteren, *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, Chapman and Hall/CRC, 2006.

[25] D. E. Rohrlich, "Modular curves, Hecke correspondences, and L-functions", Modular forms and Fermat's last theorem (Boston, MA, 1995), Springer, NewYork, pp. 41–100, 1997.

[26] W. Diffie and M. E. Hellman, "New directions in cryptography", Information Theory, IEEE Transactions, Volume 22, pp./ 644–654, 1976.

[27] I. M. Duursma and H.-S. Lee, "Tate Pairing Implementation for Hyperelliptic Curves $y^2 = x^p - x + d$", in C.S. Laih (ed.): Advances in Cryptology— ASIACRYPT 2003, Volume 2894 of Lecture Notes in Computer Science, pp. 111–123, Springer, 2003.

[28] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", Information Theory, IEEE Transactions, Volume 31(4), pp. 469–472, 1985.

[29] N. Estibals, "Compact Hardware for Computing the Tate Pairing over 128-Bit-Security Supersingular Curves" in M. Joye, A. Miyaji, and A.

Otsuka (eds.): *Pairing-Based Cryptography — Pairing 2010*, Volume 6487 of Lecture Notes in Computer Science, pp. 397–416, Springer, 2010.

[30] K. Fong, D. Hankerson, J. López and A. Menezes, "Field Inversion and Point Halving Revisited", *IEEE Transactions on Computers*, Volume 53(8), pp. 1047–1059, 2004.

[31] D. Freeman, M. Scott and E. Teske, "A Taxonomy of Pairing-Friendly Elliptic Curves", Journal of Cryptology, Volume 23, pp. 224–280, 2010.

[32] G. Frey, M. Müller and H. G. Rück, "The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems", Information Theory, IEEE Transactions, Volume 45, pp. 1717–1719, 1999.

[33] G. Frey and H. G. Rück, "A Remark Concerning m-Divisibility and the Discrete Logarithm in the Divisor Class Group of Curves", Mathematics of Computation, Volume 62, pp. 865–874, 1994.

[34] G. Frey, "Applications of Arithmetical Geometry to Cryptographic Constructions", in D. Jungnickel and H. Niederreiter (eds.): Finite Fields and Applications, Springer-Verlag, pp. 128–161 2001.

[35] S. D. Galbraith, "Supersingular Curves in Cryptography", in C. Boyd(ed.): Advances in Cryptology — ASIACRYPT 2001, Lecture Notes in Computer Science Volume 2248, Springer-Verlag, pp. 495–513, 2001.

[36] S. D. Galbraith, K. Harrison and D. Soldera, "Implementing the Tate Pairing" in C. Fieker and D. R. Kohel (eds.): Algorithmic Number Theory —ANTS-V, Volume 2369 of Lecture Notes in Computer Science, pp. 324–337, Springer, 2002.

[37] S. D. Galbraith and X. Lin, “Computing pairings using x-coordinates only”, *Designs, Codes and Cryptography*, Volume 50(3), pp. 305–324, 2009.

[38] S. D. Galbraith, K. G. Paterson and N. P. Smart, “Pairings for cryptographers”, *Discrete Applied Mathematics*, Volume 156(16), pp. 3113–3121, 2008.

[39] R. P. Gallant, R. J. Lambert and S. A. Vanstone, “Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms” in J. Kilian (ed.): *Advances in Cryptology — CRYPTO 2001*, Volume 2139 of Lecture Notes in Computer Science, pp. 190–200, Springer, 2001.

[40] D. Geer, “Risk management is still where the money is”, Computer, Volume 36(12), pp. 129–131, 2003.

[41] C. Gentry, “Practical Identity-Based Encryption Without Random Oracles” in S. Vaudenay (ed.): *Advances in Cryptology — EUROCRYPT 2006*, Volume 4004 of Lecture Notes in Computer Science, pp. 445–464, Springer, 2006.

[42] C. Gentry and A. Silverberg, “Hierarchical ID-Based Cryptography” in Y. Zheng (ed.): *Advances in Cryptology — ASIACRYPT 2002*, Volume 2501 of Lecture Notes in Computer Science, pp. 548–566, Springer, 2002.

[43] D. M. Gordon, “A Survey of Fast Exponentiation Methods”, Journal of Algorithms, Volume 27(1), pp. 129–146, 1998.

[44] M. Green and G. Ateniese, “Identity-Based Proxy Re-encryption”, in J. Katz and M. Yung (eds.): ACNS 2007, LNCS 4521, Springer-Verlag, pp. 288–306, 2007.

[45] A. Grothendieck, (with M. Artin and J. L. Verdier), "Théorie des Topos et Cohomologie Étale des Schémas", SGA4, Lecture Notes in Math. **269 270**, **305**, Springer-Verlag, Heidelberg (1972–1973).

[46] A. Grothendieck, (by P. Deligne, with J. F. Boutot, L. Illusie, and J. L. Verdier), "Cohomologie Etale", SGA4$\frac{1}{2}$, Lecture Notes in Math. **569**, Springer-Verlag, Heidelberg (1977).

[47] D. Hankerson, A. Menezes and S. Vanstone, *Guide to Elliptic Curve Cryptography*, Springer, 2004.

[48] R. Hartshorne, *Algebraic Geometry*, Graduate Texts in Math. **5**2, New York, Springer-Verlag, 1977.

[49] F. Hess, N. P. Smart and F. Vercauteren, "The Eta Pairing Revisited", IEEE Transactions on Information Theory, Volume 52(10), pp. 4595–4602, 2006.

[50] M. Hindry and J. H. Silverman, *Diophantine Geometry (An Introduction)*, Graduate Texts in Mathematics, Volume 201, Springer, 2000.

[51] A. Joux, "A One Round Protocol for Tripartite Diffie-Hellman", in W. Bosma (Ed.): ANTS-IV, LNCS 1838, pp. 385–393, 2000.

[52] K. Kawahara, "Efficient Implementation of η_T Pairing on Supersingular Elliptic Curves in Characteristic 3", Graduate School of Mathematics, Kyushu University, Thesis, 2012.

[53] S. L. Kleiman, "The Standard Conjecture", in Motives, Part 1, Proc. Sympo. Pure Math. **55**, Amer. Math. Soc., 3–20, 1994.

[54] A. Knap, *Elliptic Curves*, Mathematical Notes (Book 40), Princeton University Press, 1992.

[55] N. Koblitz, “Elliptic Curve Cryptosystems”, Mathematics of Computation, Volume 48(177), pp. 203–209, 1987.

[56] N. Koblitz, *Algebraic aspects of cryptography*, Volume=3, Springer-Verlag, 1998.

[57] N. Koblitz, “An Elliptic Curve Implementation of the Finite Field Digital Signature Algorithm” in H. Krawczyk (ed.): *Advances in Cryptology — CRYPTO ’98*, Volume 1462 of Lecture Notes in Computer Science, pp. 327–337, Springer, 1998.

[58] N. Koblitz and A. Menezes, “Pairing-Based Cryptography at High Security Levels” in N. P. Smart (ed.): *Cryptography and Coding — C&C 2005*, Volume 3796 of Lecture Notes in Computer Science, pp. 13–36, Springer, 2005.

[59] S. Lang, *Introduction to algebraic and abelian functions*, Addison Wesley, 1972.

[60] S. Lang, *Fundamentals of Diophantine Geometry*, Springer-Verlag, 1983.

[61] S. Lang, *Algebra*, Graduate Text in Math. **2**11, third edition, Springer-Verlag, New York, 2002.

[62] E. Lee, H. S. Lee and C. M. Park, “Efficient and Generalized Pairing Computation on Abelian Varieties”, IEEE Transactions on Information Theory, Volume 55(4), pp. 1793–1803, 2009.

[63] A. Lenstra and E. Verheul, “Selecting Cryptographic Key Sizes”, Journal of Cryptology, Volume 14, pp. 255–293, 2001.

[64] S. Lichtenbaum, “Duality theorems for curves overP-adic fields”, Inventiones mathematicae, Volume 7, pp. 120–136, 1969.

[65] R. Lidl and H. Niederreiter, *Finite Fields*, Encyclopedia of Mathematics and its Applications, Volume 20, Cambridge University Press, 2nd edition, 1997.

[66] B. Lynn, The Pairing-Based Cryptography Library, Stanford University, `http://crypto.stanford.edu/pbc/`.

[67] L. Martin, *Introduction to Identity-Based Encryption*, Artech House Publishers, 2008.

[68] B. Mazur, “Modular curves and the Eisenstein ideal”, Inst. Hautes Études Sci. Publ. Math. no. **47**, pp. 33–186, 1977.

[69] A. J. Menezes, *Elliptic Curve Public Key Cryptosystems*, Kluwer Academic Publishers, 1993.

[70] A. J. Menezes, T. Okamoto, and S. A. Vanstone, “Reducing elliptic curve logarithms to logarithms in a finite field”, Information Theory, IEEE Transactions on , Volume 39(5), pp. 1639–1646, 1993.

[71] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, *Handbook of Applied Cryptography*, Discrete Mathematics and Its Applications, CRC Press, 1997.

[72] V. S. Miller, “Use of Elliptic Curves in Cryptography” in H. C. Williams (ed.), Advances in Cryptology — CRYPTO ’85, Volume 218 of Lecture Notes in Computer Science, pp. 417–426, Springer, 1986.

[73] V. S. Miller, “Short Programs for functions on Curves”, unpublished manuscript, 1986.

[74] V. S. Miller, “The Weil Pairing, and Its Efficient Calculation”, Journal of Cryptology, Volume 17(4), pp. 235–261, 2004.

[75] J.S. Milne, “Jacobian Varieties”, In Arithmetic geometry (Cornell, Silverman (eds.)) Springer, New York, pp. 167–212, 1986.

[76] J.S. Milne, “Elliptic Curves”, Course Notes in his web site:, `http://www.jmilne.org/math/`.

[77] D. Mumford, *Abelian varieties*, Second edition, Oxford University Press, London, 1974.

[78] T. Nakajima, T. Izu and T. Takagi, “An Efficient Algorithm for Pairing Cryptography with Supersingular Elliptic Curves over Prime Fields”, IPSJ Journal, Volume 50(7), pp. 1745–1756, 2009.

[79] T. Nakajima , “Efficient Algorithms for Pairing-Based Cryptography using Supersingular Elliptic Curves”, Graduate School of Systems Information Science, Future University Hakodate, Thesis, 2010.

[80] E. Barker, W. Barker, W. Burr, W. Polk and M. Smid, Recommendation for Key Management - Part 1: General (Revised), NIST Special Publication 800-57, NIST, 2007.

[81] R. Sakai, K. Ohgishi and M. Kasahara, “Cryptosystems based on pairing”, in *Proceedings of the 2000 Symposium on Cryptography and Information Security*, SCIC 2000-C70, 2000.

[82] J. -P. Serre, *Géométrie algébrique et géométrie analytique*, Ann. Inst. Fourier 6, pp. 1–42, 1956.

[83] T. Okamoto and K. Takashima, “Hierarchical Predicate Encryption for Inner-Products” in M. Matsui (ed.): Advances in Cryptology — ASIACRYPT 2009, Volume 5912 of Lecture Notes in Computer Science, pp. 214–231, Springer, 2009.

[84] K. A. Ribet, “Abelian varieties over Q and modular forms”, Proceedings of KAIST Mathematics Workshop, Korea Advanced Institute of Science and Technology, Taejon, pp. 53–79, 1992.

[85] J. H. Silverman, *The Arithmetic of Elliptic Curves*, Graduate Texts in Mathematics, Volume 106, Springer, 1986.

[86] J. A. Solinas, *Generalized Mersenne Numbers*, University of Waterloo Research Report, CORR 99–39, 1999.

[87] M. Stögbauer, “Efficient Algorithms for Pairing-Based Cryptosystems”, Department of Mathematics, Darmstadt University of Technology, Thesis, 2004.

[88] R. L. Rivest, A. Shamir and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems”, Communications of the ACM, Volume 21, pp. 120–126, ACM New York, NY, USA, 1978.

[89] R. Sakai, K. Ohgishi and M. Kasahara, “Cryptosystems based on pairing”, in Proceedings of the 2000 Symposium on Cryptography and Information Security, SCIS2000-C70, 2000.

[90] R. Sakai, K. Ohgishi and M. Kasahara, “Cryptosystem Based on Pairing over Elliptic Curve” in Proceedings of Symposium on Cryptography and Information Security — SCIS 2001, 7B-2, 2001.

[91] O. Schirokauer, "The number field sieve for integers of low weight", Mathematics of Computation, Volume 79, pp. 583–602, 2010.

[92] M. Scott, "Computing the Tate pairing", Topics in Cryptology - CT-RSA 2005, Lecture Notes in Computer Science, Volume 3376, Springer-Verlag, pp. 293-304, 2005.

[93] M. Scott, "On the Efficient Implementation of Pairing-Based Protocols", Cryptology ePrint Archive, Report 2011/334, 2011.

[94] A. Salomaa, *Public-key cryptography*, Volume 23, Springer, 1996.

[95] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes" in G. R. Blakley and D. Chaum (eds.): Advances in Cryptology, Lecture Notes in Computer Science, Volume 196, pp. 47–53, 1984.

[96] S. Sheng, L. Broderick, C. A. Koranda and J. J. Hyland, "Why Johnny still can't encrypt: evaluating the usability of email encryption software", Proceedings of the 2006 Symposium On Usable Privacy and Security, Pittsburgh, PA, pp. 12–14, 2006.

[97] G. Shimura, "On the factors of the jacobian variety of a modular function field", J. Math. Soc. Japan **25**, pp. 523–544, 1973.

[98] C. Shu, S. Kwon and K. Gaj, "Reconfigurable Computing Approach for Tate Pairing Cryptosystems over Binary Fields", Computers, IEEE Transactions, Volume 58(9), pp. 1221–1237, 2009.

[99] J. Tate, "WC-groups over p-adic fields", Séminaire Bourbaki; 10e année: 1957/1958. Textes des conférences; Exposés 152 a 168; 2e éd. corrigée, Exposé 156, vol. 13, Secrétariat mathématique, Paris, 1958.

[100] J. Tate, “On the conjecture of Birch and Swinnerton-Dyer and a geometric analog”, Sém. Bourbaki, no. 306, pp. 1–26, 1966.

[101] J. R. Vacca, *Public Key Infrastructure: Building Trusted Applications and Web Services*, CRC Press, 2004.

[102] S. A. Vanstone, “Next generation security for wireless: elliptic curve cryptography”, Computers and Security, Volume 22(5), pp. 412–415, 2003.

[103] F. Vercauteren, “Optimal pairings”, IEEE Transactions on Information Theory, Volume 56(1), pp. 455–461, 2010.

[104] E. R. Verheul, “Evidence that XTR Is More Secure than Supersingular Elliptic Curve Cryptosystems”, in B. Pfitzmann(ed.): Advances in Cryptology — EUROCRYPT 2001, Lecture Notes in Computer Science Volume 2045, pp. 195–210, 2001.

[105] E. R. Verheul, “Evidence that XTR Is More Secure than Supersingular Elliptic Curve Cryptosystems”, Journal of Cryptology, Volume 17(4), pp. 277–296, 2004.

[106] A. Walfisz, “Zur additiven Zahlentheorie. II.”, Mathematische Zeitschrift, Volume 40, pp. 592–607, 1936.

[107] L. C. Washington, *Elliptic Curves: Number Theory and Cryptography, Second Edition*, Chapman and Hall/CRC, 2008.

[108] A. Walfisz, “Zur additiven zahlentheorie. II”, Mathematische Zeitschrift, Volume 40, pp. 592–607, 1936.

[109] B. Waters, “Efficient Identity-Based Encryption Without Random Oracles”, in R. Cramer (ed.): Advances in Cryptology — EUROCRYPT

2005, Volume 3494 of Lecture Notes in Computer Science, pp. 114–127, Springer, 2005

[110] A. Weil, *Sur les courbes algébriques et les variétés qui s'endéduisent*, Hermann et Cie., Paris, 1948.

[111] A. Weil, *Variétés abéliennes et courbes algébriques*, Hermann et Cie., Paris, 1948.

[112] A. Whitten and J. D. Tygar, "Why Johnny can't encrypt: A usability evaluation of PGP 5.0", Proceedings of the 8th USENIX Security Symposium, Volume 99, Washington, D. C., pp. 169–184, 1999.

[113] C. A. Zhao, F. Zhang and J. Huang, "A note on the Ate pairing", International Journal of Information Security, Volume 7(6), pp. 379–382, 2008.

CURRICULUM VITAE

EDUCATION:

- Doctor's programs, Functional Mathematics course, Kyushu University (2011–2014).
- Doctor's programs, Mathematics course, Kyushu University (2004–2007).
- M.S., Mathematics, Kyushu University (2004).
- B.S., Mathematics, Kyushu University (2002).

EXPERIENCE:

- IT Engineer (Current Position: Chief staff), Hitachi, Ltd. IT Platform Division Group (2007–present).
- Part-time Lecturer, Faculty of Information Science, Kyushu Sangyo University (2004–2007).

PAPER:

1. Takumi Tomita and Tsuyoshi Takagi, "Efficient system parameters for Identity-Based Encryption using supersingular elliptic curves" (accepted), to be published: JSIAM Letters, 2013.

2. Takumi Tomita, "Winding element of abelian varieties of GL2-type and

special values of the L-functions", Master Thesis, Kyushu University, March 2004.

TALKS:

1. Takumi Tomita and Tsuyoshi Takagi. Efficient algorithm of HashTo-Point using supersingular elliptic curves over $\mathbb{F}_p$. Japan Algorithmic Number Theory(JANT), The Japan Society for Industrial and Applied Mathematics, March 2013.

2. Counting certain imaginary quadratic fields with prescribed 2-class order. The 2nd COE Conference for Young Researchers, Hokkaido University, February 2006.

3. Distribution of certain imaginary quadratic fields with prescribed 2-class order. Kyushu Branch, The Mathematical Society of Japan, October 2005.

4. Modular symbol and special value of L-function. Seminar on Algebra, Tohoku University, Japan, December 2003.

5. Winding element of abelian varieties of GL_2-type and special values of the L-functions. Seminar on Number Theory, Kyushu University, Japan, June 2003.

6. Goncharov's approach to Zagier's conjecture. Seminar on Number Theory, Kyushu University, Japan, October 2002.