ISOGENIES AND CRYPTOGRAPHY

Raza Ali Kazmi

A thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of **Master of Computer Science**

Concordia University

Montréal, Québec, Canada

August 2008

Concordia University

School of Graduate Studies

This is to certify that the thesis prepared

By: Raza Ali Kazmi

Entitled: Isogenies and Cryptography

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

____________________ Chair
Sudhir Mudur

____________________ Examiner
Unknown

____________________ Examiner
Unknown

____________________ Supervisor
Claude Crépeau

____________________Supervisor
David Ford

Approved ____________________
Chair of Department or Graduate Program Director

__________ 20 _____ ____________________

Robin A.L. Drew, Dean

Faculty of Engineering and Computer Science

Abstract

Isogenies and Cryptography

Raza Ali Kazmi

This thesis explores the notion of isogenies and its applications to cryptography. Elliptic curve cryptography (ECC) is an efficient public cryptosystem with a short key size. For this reason it is suitable for implementing on memory-constraint devices such as smart cards, mobile devices, etc. However, these devices leak information about their private key through side channels (power consumption, electromagnetic radiation, timing etc) during cryptographic processing. In this thesis we have examined countermeasures against a specific side channel attack (power consumption) using isogeny, (a rational homomorphism between elliptic curves) and elliptic curve isomorphism. We found that these methods are an efficient way of securing cryptographic devices using ECC against power analysis attacks. We have also investigated the security and efficiency of implementation of a public key cryptosystem based on isogenies. We found that in order to implement the proposed cryptosystem one has to compute a root of the Hilbert polynomial $H_D(X)$ over $\mathbf{F}_p$. Since there is no known efficient way of achieving this calculation, the proposed cryptosystem cannot be used in practice.

Acknowledgments

I would like to thanks to Professor Claude Crépeau (McGill University) for introducing me to the wonderful field of cryptography. I would also like to thank him for all the help he has given me over the past. Without his help, love and encouragement I would have not finished this thesis.

I would also like to thanks Professor David Ford (Concordia University) for having agreed to take me on as a M.Sc. student and for providing me financial support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Prior to 1976, cryptography was a black art, understood and practiced by government and military personnel. It all changed in 1976, when Whitfield Diffie and Martin Hellman in the paper entitled "New Directions in Cryptography" proposed a protocol that allowed two parties with no prior knowledge of each other to jointly share a private key over an insecure channel. [1] That was the beginning of Public Key Cryptography. Since 1976 many public key cryptosystems and signature schemes have been proposed and are currently in use in the public domain. Some of the most popular ones are **RSA, Elgamal,** etc. Today cryptography has become a well established *academic discipline* that is taught in many universities. It is widely available for use by companies and individuals. For example, smart cards such as debit and credit cards use cryptography to prevent the duplication and fraudulent use of information saved to chips on such cards. While there is no doubt that these cryptographic devices have become very popular, their security has become a major concern.

The advantage of elliptic curve based cryptosystems over other cryptosystems (**RSA**, etc.) is their short key size. For this reason it is suitable for implementing on memory-constrained devices such as smart cards. Akishita and Takagi [AT05] and their predecessors, Coron [Cor99], Goubin [Gou03], Okeya and Sakurai [Os00] proposed power analysis attacks on elliptic curve cryptosystems that will allow an adversary to recover the private key by monitoring the power consumption of cryptographic devices such as smart cards.

[1] Diffie-Hellman protocol was discovered a few years prior at the **Government Communications Headquarters**, British intelligence agency, by Malcolm J. Williamson, but was kept classified [Wil74]

In this thesis we assess the application of **isogenies** (a rational homomorphism between elliptic curves) and elliptic curve isomorphisms for defense against the various power analysis attacks proposed above. We also study and analyse a new public cryptosystem where the security is based on the isogeny problem[RS06]. The isogeny problem can be stated as follows: Given two isogenious elliptic curves E_i and E_j over F_p, compute an isogeny between E_i and E_j.

1.1 Contribution of the Thesis

The contributions of this thesis are the following:

- We show that a certain class of standard curves (**SECG**[sec00] and **NIST** [nis]) over $\mathbf{F}_{2^m}$ can be defended very efficiently against power analysis attacks proposed in [AT05] using elliptic curve isomorphisms. We show that the remaining classes of curves over $\mathbf{F}_{2^m}$ can be defended efficiently using isogenies. We also show that all standard curves except *P-521* (a curve in **NIST** standard) over $\mathbf{F}_p$ can be defended efficiently through isogeny against all bad points (**ZVP** points) suggested in [AT05]:
- We calculate the additional cost of the isogeny and an elliptic curve isomorphism defense for standard curves.
- We also examine the security and efficiency of a public key cryptosystem based on isogeny. We found that this cryptosytem has many drawbacks and in practice it can not be used unless we can efficiently compute a root of Hilbert polynomial over $\mathbf{F}_p$ [IBS06].

1.2 Organization of the Thesis

The thesis consists of 5 chapters. Chapter 2 provides the mathematical background required to understand the remaining chapters. Chapter 3 describes in detail the power analysis attacks as well as the countermeasures against these attacks. Chapter 4 describes the new public key cryptosystem based on the isogeny problem and its drawbacks, while chapter 5 is the conclusion.

Chapter 2

Mathematical Background

2.1 Groups

A **group** is a nonempty set G together with a binary operation $*$ on G which satisfies the following first three axioms.

1. **Associativity:** $a * (b * c) = (a * b) * c$ *for all* $a, b, c \in G$.

2. **Identity:** There exist a *unique* element $e \in G$ such that $a * e = e * a = a$ *for all* $a \in G$.

3. **Inversibility :** For every $a \in G$ there exist an element $a^{-1} \in G$ such that $a * a^{-1}{=}a^{-1} * a{=}e$.

4. **Commutativity:** A group is called **abelian** if $a * b = b * a$ *for all* $a, b \in G$.

The *order* of a group is the number of elements in it. A group is called finite if it has a finite order. One example of a group is $\mathbf{SL_2}(\mathbb{Z})$ = *Set of all* 2×2 matrices over $\mathbb{Z}$ with determinant 1.

2.1.1 Group Homomorphism and Automorphism

Let G and H be two groups. The group homomorphism is a function $f : G \longrightarrow H$ such that

$$f(ab) = f(a)f(b) \text{ for all } a, b \in G$$

An isomorphism $f : G \longrightarrow G$ is called an automorphism.

2.2 Rings

A **ring** is a nonempty set R together with two binary operations (usually denoted as addition $(+)$ and multiplication $(\times)$) such that R satisfies at least the following first three axioms.

1. **(R,+)** is an abelian group.
2. **Associative multiplication:** $(a \times b) \times c = a \times (b \times c)$ *for all* $a, b, c \in R$.
3. **Distributive laws:** $a \times (b + c) = a \times b + a \times c$ and $(a + b) \times c = a \times c + b \times c$.
4. In addition *If* $a \times b = b \times a$ *for all* $a, b \in R$, then R is called a **commutative** ring.
5. If R contains an element 1_R such that $1_R \times a = a \times 1_R = a$ *for all* $a \in R$, then R is said to be a **ring with identity**.

2.2.1 Fields

A **field** is a commutative ring R with multiplicative identity 1_R such that *for all* $a \in R$ and $a \neq 0$, the equation $ax = 1_R$ has a solution.

2.2.2 Subring

Let R be a ring and S be a nonempty subset of R that is closed under the operations of addition and multiplication in R. If S is itself a ring under these operations then S is called a subring of R.

2.2.3 Ideals

A **subring I** of a ring R is an **ideal** provided *for all* $r \in R$ and $a \in I \Rightarrow ar \in I$ and $ra \in I$.

2.3 Imaginary Quadratic Field

Definition: An *integer* $D < 0$ is called a **fundamental discriminant** If *either* $D \equiv 1 \pmod 4$ and D is *square free*, *or* $D \equiv 0 \pmod 4$ *and* $D/4$ *is square free and* $D/4 \equiv 2, 3 \pmod 4$.

Let D be a fundamental discriminant and let

$$K = \mathbb{Q}(\sqrt{D}) = \{\alpha + \beta\sqrt{D} | \alpha, \beta \in \mathbb{Q}\}$$

Then K is called an **imaginary quadratic field**. Let

$$O_D = \mathbb{Z}[\delta]=\{\text{a + b}\delta|a, b \in \mathbb{Z}\}$$

Where $\delta = \frac{1+\sqrt{D}}{2}$ if $D \equiv 1 \pmod 4$ or $\delta = \sqrt{D/4}$ if $D \equiv 0 \pmod 4$. It is easy to see that O_D is a subring of $\mathbf{K}$. An **order** in K is any ring R such that $\mathbb{Z} \subset R \subseteq O_D$ and $\mathbb{Z} \neq R$. Any order has the form

$$R = \{a + bf\delta|a, b \in \mathbb{Z}\}$$

Where f is a positive integer called **conductor** of R. The discriminant of R is f^2D. Every order R of K has discriminant of form f^2D and if A is any non-square integer such that $A \equiv 0, 1 \pmod 4$ then A is uniquely of the form $A = f^2D$, where D is a fundamental discriminant and there exists a unique order R of the discriminant A [Coh93].

Any **ideal** of quadratic imaginary order R has the form $a\mathbb{Z} + (b + f\delta)\mathbb{Z}$, where a, b, c are any integers such that $b^2 - 4ac = f^2D$ and $gcd(a, b, c) = 1$ [Bucon]. If I and J be two ideals of R, then we say that I and J are equivalent if there exist a nonzero element $\alpha \in \mathbf{K}$ such that $I = \alpha J$.

2.4 Binary Quadratic Integral Forms

A binary quadratic integral form is $f(X, Y) = aX^2 + bXY + cY^2$ with $a, b, c \in \mathbb{Z}$ and $a = b = c \neq 0$. We will write binary quadratic integrals form as $f = (a, b, c)$ and called f a *form*. The discriminant of f is $\triangle(f) = b^2 - 4ac$. A integral form $f = (a, b, c)$ is called primitive if $gcd(a, b, c) = 1$ and called positive definite if $a > 0$ and $\triangle(f) < 0$. *We say that the two forms f and g are equivalent* if there exist $U \in \mathbf{SL(2}, \mathbb{Z})$ such that $g = f(U(X, Y))$, where we define:

$$U(X, Y) = (uX + vY, wX + zY) = \begin{pmatrix} u & v \\ w & z \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix}$$

Note that two equivalent forms have the same discriminant but the converse is not true. For example $f = (2, -1, 3)$ and $g = (2, 1, 3)$ have discriminant -23 but they are not equivalent.

Let $\triangle < 0$ be a fixed integer and $\triangle \equiv 0 \pmod 4$ or $\triangle \equiv 1 \pmod 4$. We define an equivalence class of integral form a set $[f]$

$$[f] = \{g|g = f(U(X, Y)) \text{ and } U \in \mathbf{SL_2}(\mathbb{Z})\}$$

2.5 Class Number

The Class Number $h_{\triangle}$ is the number of equivalence classes of integral forms of discriminant $\triangle$.

It is shown in [EKZ03] that every proper equivalence class has a unique representation $(a, b, c) \in \mathbb{Z}$ which satisfies the following conditions:

1. $b^2 - 4ac = \triangle$ and $gcd(a, b, c) = 1$
2. $|b| \leq a \leq \sqrt{| \triangle / 3|}, a \leq c$ and if $|b| = a$ or $a = c$ then $b \geq 0$.

We call these triplets *reduced* and write these as $[a, b, c]$ in order to emphasize the fact that each triplet represents a class. From above it follows that the number of proper equivalence classes of a fixed discriminant is finite, since $\frac{b^2 - \triangle}{4a} = c$ and b, a are bounded, therefore c is bounded. The fastest algorithm known, for computing class numbers and reduced forms is by Shank whose running time is $O(|D|^{\frac{1}{5}+\epsilon})$. See [Coh93] for more details.

2.6 The Relationship Between Forms and Imaginary Quadratic Fields

The imaginary quadratic fields and binary quadratic integral forms are closely related. Let $D < 0$ be a fundamental discriminant and $K = \mathbb{Q}(\sqrt{D})$ and $O_{f^2 D}$ be an order in K. Let I and J be two ideals of $O_{f^2 D}$, we say that I and J are equivalent if there exists a non-zero element $\alpha \in \mathbf{K}$ such that $I = \alpha J$. Let $Cl_{f^2 D}$ be the set of proper equivalence classes of integral forms of discriminant $f^2 D$. Let $[a, b, c]$ be a representative of a class in $Cl_{f^2 D}$. If $[a, b, c]$ represent a class in $Cl_{f^2 D}$, then the set $a\mathbb{Z} + (b + f\delta\mathbb{Z})$ is an ideal of $O_{f^2 D}$ (where δ is the same as defined in section 2.3). If $[a_1, b_1, c_1]$ and $[a_2, b_2, c_2]$ are two representations of a same class, then the corresponding ideals are also in the same class and are properly equivalent. Similarly, if $a\mathbb{Z} + (b + f\delta\mathbb{Z})$ is an ideal of $O_{f^2 D}$, then $[a, b, \frac{b^2 - f^2 D}{4a}]$ represents a class in $Cl_{f^2 D}$ and if I and J are two properly equivalent ideals, then the corresponding forms are also in the same class. [Bucon]

Example 1.

$K = \mathbb{Q}(\sqrt{-3})$, $g = [1, 1, 1]$ and $f = [195751, 37615, 1807] = 195751X^2 + 37615XY + 1807Y^2$ are two proper equivalent forms with matrix $U \in \mathbf{SL_2}(\mathbb{Z})$ define below:

$$U = \begin{pmatrix} -22 & -49 \\ 229 & 510 \end{pmatrix}$$

$$f(U(X,Y)) = f(uX + vY, wX + zY) = f(-22X - 49Y, 229X + 510Y) = X^2 + XY + Y^2 = g(X,Y)$$

Hence the two forms are properly equal. The corresponding ideals $I_g = \mathbb{Z} + \frac{(1+\sqrt{-3})}{2}\mathbb{Z}$, $I_f = 195751\mathbb{Z} + 37615\frac{(1+\sqrt{-3})}{2}$ are also properly equivalent for $\alpha = -22 + 229\frac{1-\sqrt{-3}}{2}$

$$\mathbb{Z} + \frac{(1+\sqrt{-3})}{2}\mathbb{Z} = (-22 + 229\frac{(1-\sqrt{-3})}{2})(195751\mathbb{Z} + \frac{37615(1+\sqrt{-3})}{2}\mathbb{Z})$$

In general, if $f = [a,b,c], g = [a^{'}, b^{'}, c']$ are two equivalent forms with matrix $U \in \mathbf{SL_2}(\mathbb{Z})$ and I_f and I_g are corresponding ideals then $\alpha = u + w\frac{b-\sqrt{\delta}}{2a}$ such that $I_g = \alpha I_f$. Note u, w are the $(1,1)$ and $(2,1)$ entries of matrix U.

2.7 Class Group

From the above discussion it follows that Cl_D can also be viewed as the set of equivalence classes of ideals of O_D. In fact Cl_D is an abelian group of finite order with respect to the group operation defined below [Coh93, Bucon]. We assume that each class Cl_D is represented by its unique reduced form (see section 2.5):

1. **Inverse:** If $[a,b,c] \in Cl_D$ then $[a,b,c]^{-1} = [a,-b,c]$
2. **Identity:** If $D \equiv 0 \pmod 4$ then $e = [1, 1, \frac{|D|}{4}]$ and if $D \equiv 1 \pmod 4$ then $e = [1, 1, \frac{|D|+1}{4}]$
3. **Binary Operation:** Let $[a_1, b_1, c_1]$ and $[a_2, b_2, c_2]$ be two unique representations in Cl_D, then their product is given by

$$[a_1, b_1, c_1] * [a_2, b_2, c_2] = [a_3, b_3, c_3] = \left[d_0\frac{a_1 a_2}{d^2}, b_2 + \frac{2a_2}{d}(v(s - b_2) - wc_2), \frac{b_3^2 - D}{4a_3}\right]$$

Where $s = (b_1 + b_2)/2$, $n = (b_1 - b_2)/2$, $d = gcd(a_1, a_2, s)$, $d_0 = gcd(d, c_1, c_2, n)$ and u, v, w, d are integers such that $ua_1 + va_2 + ws = d$

2.8 Algebraic Numbers, Integers and Algebraic Closure

Let α be a complex number. If there exist a non constant polynomial $g(x) \in \mathbb{Q}[x]$ with a property that $g(\alpha) = 0$, then we say α is an **algebraic number**, similarly if there exist a non constant polynomial $g(x) \in \mathbb{Z}[x]$ with property that $g(\alpha) = 0$ then we say α is an **algebraic integer**. In general let $\mathbf{F}_1$ and $\mathbf{F}_2$ be two fields with $\mathbf{F_1} \subseteq \mathbf{F_2}$ and $s \in \mathbf{F}_2$. We say s is **algebraic** over $\mathbf{F}_1$ if there exist a non constant polynomial $g(x) \in \mathbf{F}_1[x]$ with the property that $g(s) = 0$. If every element of F_2 is **algebraic** over $\mathbf{F}_1$, then we say that $\mathbf{F}_2$ is an **algebraic** extension of $\mathbf{F_1}$. An **algebraic Closure** of a field $\mathbf{K}$ is a field $\overline{\mathbf{K}}$ such that:

1. $\overline{\mathbf{K}}$ *is an algebraic extension of* $\mathbf{K}$.

2. *Every non-constant polynomial* $g(x) \in \overline{\mathbf{K}}$ *has a root in* $\overline{\mathbf{K}}$.

If we let $\mathbf{F_1} = \mathbb{Q}$ and $\mathbf{F}_2 = \mathbb{Q}(\sqrt{D})$, then set of algebraic integers in $\mathbb{Q}(\sqrt{D})$ is precisely the ring O_D.

2.9 Elliptic Curves over an Arbitrary Field K

An elliptic curve E over $\mathbf{K}$ is the set of solutions of an equation of the form

$$E(\mathbf{K}) : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \quad \textit{where } a_1, a_2, a_3, a_4, a_5 \in \mathbf{K}$$

and let

$$b_2 = a_1^2 + 4a_2, \quad b_4 = 2a_4 + a_1a_3, \quad b_6 = a_3^2 + 4a_6, \quad b_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2{a_3}^2 - {a_4}^2,$$
$$c_4 = b_2^2 - 24b_4, \;\; c_6 = -b_2^3 + 36b_2b_4 - 216b_6 \quad \textit{and} \quad \Delta = -b_2^2b_8 - 8b_4^3 - 27b_6^2 + 9b_2b_4b_6$$

Δ is the discriminant of $E(\mathbf{K})$. We assume $\Delta \neq 0$ and $a_1 \neq 0$. The *j-invariant* of $E(\mathbf{K})$ is

$$j(E) = c_4^3/\Delta$$

For $char(\mathbf{K}) \neq 2, 3$ the elliptic curve equation is reduced to

$$E := y_1^2 = x_1^3 + Ax_1 + B \textit{ for some } A, B \in \mathbf{K}$$

Where $A = -c_4/12$, $B = -c_6/216$

And *j-invariant* of E is [1]

$$j(E) = 1728\frac{4A^3}{4A^3 + 27B^2}$$

For field of characteristic 2 the elliptic curve equation is reduced to

$$E : y^2 + xy = x^3 + A^{'}x + B^{'} \quad for \quad A^{'}, B^{'} \in \mathbf{K}$$

And *j-invariant* is given by

$$j(E) = \frac{a_1^{12}}{\Delta} = \frac{1}{B'}$$

2.10 Group Law in Characteristic $\neq 2, 3$

The set of points on $E(\mathbf{K})$ form an additive abelian group with P_∞ (point at infinity) as the neutral element. In section 2.12 we will explain P_∞ in more detail. Let $\mathbf{K}$ be a field such that $char(\mathbf{K}) \neq 2, 3$ and $E(\mathbf{K}) : y^2 = 4x^3 - Ax - B$ be an elliptic curve. Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be two points on E. We add two points $P_3 = P_1 + P_2$ by the following rules:

1. If $x_1 \neq x_2$, then $P_3 = P_1 + P_2$ is computed by the following rule

$$x_3 = m^2 - x_1 - x_2, \quad y_3 = m(x_1 - x_3) - y_1, \quad where \quad m = \frac{y_2 - y_1}{x_2 - x_1}$$

2. If $x_1 = x_2$ but $y_1 \neq y_2$, then $P_1 + P_2 = P_\infty$

3. If $P_1 = P_2$ and $y_1 = 0$, then $P_1 + P_2 = P_\infty$

4. If $P_1 = P_2$ and $y_1 \neq 0$, then $P_3 = P_1 + P_2$ is computed by the following rule

$$x_3 = m^2 - 2x_1, \quad y_3 = m(x_1 - x_3) - y_1, \quad where \quad m = \frac{3x_1^2 + A}{2y_1}$$

2.11 Group Law in Character 2

Let $\mathbf{K}$ be a field such that $char(\mathbf{K}) = 2$ and $E(\mathbf{K}) : y^2 + xy = x^3 + Ax + B$ be an elliptic curve. Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be points on E. We add two points $P_3 = P_1 + P_2$ by the following

[1] we always assume that $x_1^3 + Ax_1 + B$ has no multiple roots

rule:

1. $P_1 = (x_1, y_1)$, then $-P_1 = (x_1, x_1 + y_1)$

2. If $P_1 \neq P_\infty$, $P_2 \neq P_\infty$ and $P_1 \neq -P_2$ then $P_3 = P_1 + P_2$

$$x_3 = m^2 + m + x_1 + x_2 + A, \quad y_3 = m(x_1 + x_3) + x_3 + y_1, \quad where \quad m = \frac{y_2 + y_1}{x_2 + x_1}$$

3. If $P_1 = P_2$ and $P_1 \neq P_\infty$, then $P_3 = P_1 + P_2$ is computed by the following rule

$$x_3 = m^2 + m + A, \quad y_3 = x_1^2 + (m+1)x_3, \quad where \quad m = x_1 + \frac{y_1}{x_1}$$

2.12 Projective Plane

Let $\mathbf{K}$ be a field. A two dimensional projective plane $\mathbf{P^2_K}$ over $\mathbf{K}$ is a set of equivalence classes of triples $[(X, Y, Z)]$ with $X, Y, Z \in \mathbf{K}$ and not all X, Y, Z are zero. Two triples (X_1, Y_1, Z_1) and (X_2, Y_2, Z_2) are said to be equivalent if there exist a nonzero $\alpha \in \mathbf{K}$ such that

$$(X_1, Y_1, Z_1) = (\alpha^2 X_2, \alpha^3 Y_2, \alpha Z_2)$$

If $[(X, Y, Z)]$ be any equivalence class (point) in $\mathbf{P^2_K}$ with $Z \neq 0$, then this class can uniquely be representated by the class $[(x, y, 1)]$, by setting $x = X/Z^2$ and $y = Y/Z^3$. The points for which $Z = 0$ are called points at infinity. A two dimensional **affine plane** over $\mathbf{K}$ is define to be

$$\mathbf{A^2_K} = \{(x, y) \in \mathbf{K} \times \mathbf{K}\}$$

There is a map from affine to projective plane

$$\mathbf{A^2_K} \longrightarrow \mathbf{P^2_K}$$

$$(x, y) \longmapsto (x, y, 1)$$

The point P_∞ on an elliptic curve is identified with an equivalence class $[(\alpha^2, \alpha^3, 0)]$ in $\mathbf{P^2_K}$. To see this note that any curve $f(x, y) = 0$ in $\mathbf{A^2_K}$ corresponds to $F(X, Y, Z)$ in $\mathbf{P^2_K}$ by simply setting x by X/Z^2 and y by Y/Z^3. If $char(\mathbf{K}) > 3$, then the equation of an elliptic curve in $\mathbf{A^2_K}$ is given by

$$E : y^2 = x^3 + Ax + B$$

By setting $x = X/Z^2$ and $y = Y/Z^3$ we obtain the equation of an elliptic curve in $\mathbf{P^2_K}$

$$E : ZY^2 = X^3 + AXZ^2 + BZ^3$$

Now to see what points of form $[(X, Y, 0)]$ lie on $Y^2 = X^3 + AXZ^4 + BZ^6$, we set $Z = 0$ and obtained $Y^2 = X^3$. Hence point at infinity P_∞ is represented by any triples of form $((\alpha^2, \alpha^3, 0)$ in $\mathbf{P^2_K}$.

Similarly, if char($\mathbf{K}$=2), then equation of an elliptic curve in $\mathbf{A^2_K}$ is

$$E : y^2 + xy = x^3 + Ax + B$$

By setting $x = X/Z^2$ and $y = Y/Z^3$ we obtain the equation of elliptic curve in $\mathbf{P^2_K}$

$$E : Y^2 + XYZ = X^3 + AX^2Z^2 + BZ^6$$

We set $Z = 0$ and obtain $Y^2 = X^3$. Hence, the point at infinity P_∞ is represented by any triplet of form $(\alpha^2, \alpha^3, 0)$ in $\mathbf{P^2_K}$.

2.13 Group Law in Projective Coordinates for Char($\mathbf{K} \neq 2, 3$)

Let $\mathbf{K}$ be a field such that char($\mathbf{K}$) $\neq 2, 3$ and $E : ZY^2 = X^3 + AXZ^2 + BZ^3$ be an elliptic curve over projective plane $\mathbf{P^2_K}$. Let $P_1 = (X_1, Y_1, Z_1)$ and $P_2 = (X_2, Y_2, Z_2)$ be two points on E. We add two points $P_3 = P_1 + P_2$ by the following rule.

1. If $P_1 = (X_1, Y_1, Z_1)$, then $-P_1 = (X_1, -Y_1, Z_1)$ and $P_1 + (-P_1) = P_\infty$.

2. $P_1 + P_\infty = P_1$.

3. If $P_1 \neq \pm P_2$ and $P_1 \neq P_\infty$, then $P_3 = P_1 + P_2$ is computed by the following rule

$$X_3 = -H^3 - 2U_1H^2 + R^2, \quad Y_3 = -S_1H^3 + R(U_1H^2 - X_3), \quad Z_3 = Z_1Z_2H$$

where,

$U_1 = X_1 Z_2^2$, $U_2 = X_2 Z_1^2$, $S_1 = Y_1 Z_2^3$, $S_2 = Y_2 Z_1^3$, $H = U_2 - U_1$, $R = S_2 - S_1$

4. If $P_1 = P_2$ and $P_1 \neq P_\infty$, then $P_3 = P_1 + P_2$ is computed by the following rule

$$X_3 = T, \quad Y_3 = -8Y_1^4 + M(S - T), \quad Z_3 = 2Y_1 Z_1$$

where,

$S = 4X_1 Y_1^2$, $M = 3X_1^2 + AZ_1$, $T = -2S + M^2$

2.14 Group Law in Projective Coordinates for Char(K=2)

Let $\mathbf{K}$ be a field such that char($\mathbf{K}$) = 2 and $Y^2 + XYZ = X^3 + AX^2Z^2 + BZ^6$ be an elliptic curve over projective plane $\mathbf{P}^2_{\mathbf{K}}$. Let $P_1 = X_1, Y_1, Z_1)$ and $P_2 = (X_2, Y_2, Z_2)$ be points on E. We add two points in $\mathbf{P}^2_{\mathbf{K}}$ by the following rules:

1. If $P_1 = (X_1, Y_1, Z_1)$, then $-P_1 = (X_1, X_1 + Y_1, Z_1)$ and $P_1 + (-P_1) = P_\infty$
2. $P_1 + P_\infty = P_1$.
3. If $P_1 \neq \pm P_2$ and $P_1 \neq P_\infty$, then $P_3 = P_1 + P_2$ is computed by the following rule

$$X_3 = C^2 + H + G, \quad Y_3 = HI + Z_3 J, \quad Z_3 = F^2$$

where,

$A_1 = Y_1 Z_2^2$, $A_2 = Y_2 Z_1^2$, $B_1 = X_1 Z_2$, $B_2 = X_2 Z_1$, $C = A_1 + A_2$, $D = B_1 + B_2$,
$E = Z_1 Z_2$, $F = DE$, $G = D^2(F + AE^2)$, $H = CF$, $I = D^2 B_1 E + X_3$,
$J = D^2 A_1 + X_3$

4. If $P_1 = P_2$ and $P_1 \neq P_\infty$, then $P_3 = 2P_1$ is computed by the following rule

$$X_3 = X_1^4 + BZ_1^4, \quad Y_3 = BZ_1^4 + X_3(AZ_3 + Y_1^2 + BZ_4), \quad Z_3 = X_1^2 Z_1^2$$

2.15 Lattices and Elliptic Curves over $\mathbb{C}$

Let ω_1, ω_2 be complex numbers that are linearly independent over $\mathbb{R}$. Then

$$\mathbf{L} = \mathbb{Z}\omega_1 + \mathbb{Z}\omega_2 = \{n\omega_1 + m\omega_2 | n, m \in \mathbb{Z}\}$$

is called a lattice in $\mathbb{C}$. Let $\tau = \frac{w_1}{w_2}$. Since ω_1, ω_2 are linearly independent, τ cannot be real. By switching ω_1, ω_2 if necessary we may assume that

$$\Im(\tau) > 0$$

which means $\tau \in H$, where

$$H = \{x + iy \in \mathbb{C} | y > 0\}$$

Two lattices L_1, L_2 are equivalent if there exist a nonzero complex number α such that $L_1 = \alpha L_2$. The lattice

$$L_\tau = \mathbb{Z}_\tau + \mathbb{Z}$$

is equivalent to L, since $L_\tau = \alpha L$ for $\alpha = w_2$. Let L be a lattice. Define the **Eisentein series** for any fixed $k \geq 2$

$$G_K(L) = \sum_{i \in L, i \neq 0} i^{-k}$$

A j-function for a lattice L is

$$j(L) = 1728 \frac{60G_4(L)^3}{60G_4^3 - 3780G_6(L)2}$$

and $j(L) = j(L_\tau) = j(\tau)$ where

$$j(\tau) = 1728 \frac{(1 + 240 \sum_{j=1}^{\infty} \frac{j^3 q^j}{1-q^j})^3}{(1 + 240 \sum_{j=1}^{\infty} \frac{j^3 q^j}{1-q^j})^3 - (1 - 504 \sum_{j=1}^{\infty} \frac{j^5 q^j}{1-q^j})^2}$$

and $q = e^{2\pi i \tau}$. Note that $\mathbb{C}$ is an additive group and any lattice L in $\mathbb{C}$ is its subgroup. For any $a \in \mathbb{C}$ define the set

$$L + a = \{b \in \mathbb{C} | b - a \in L\}$$

This is called a **right coset** of L in $\mathbb{C}$. Any two right cosets of L are either disjoint or identical. The set

$$\mathbb{C}/L \textit{ denotes the set of all right cosets of } L \textit{ in } \mathbb{C}$$

$\mathbb{C}/\mathbb{L}$ is isomorphic to an elliptic curves $E := y^2 = 4x^3 - Ax - B$ over $\mathbb{C}$, where

$$A = 60G_4(L) \textit{ and } B = 140G_6(L)$$

Similarly, an elliptic curve $E^{'} : y^2 = 4x^3 - A^{'}x - B^{'}$ over $\mathbb{C}$ is isomorphic to $\mathbb{C}/L_\tau$, where

$$j(E^{'}) = 1728\frac{4A'^3}{4A'^3 + 27B'2} = j(\tau) = 1728\frac{(1 + 240\sum_{j=1}^{\infty}\frac{j^3q^j}{1-q^j})^3}{(1 + 240\sum_{j=1}^{\infty}\frac{j^3q^j}{1-q^j})^3 - (1 - 504\sum_{j=1}^{\infty}\frac{j^5q^j}{1-q^j})^2}$$

The set of lattices over $\mathbb{C}$ bijectively corresponds to the set of elliptic curves over $\mathbb{C}$.

2.16 Unramified Primes

In simple words, a prime number over a ring R is called unramified if it splits over R. For example consider **ring of gaussian integers**

$$\mathbb{Z}[i] = \{a + bi | a, b \in \mathbb{Z}\}$$

2 is unramified in $\mathbb{Z}[i]$, because $(1 + i)(1 - i) = 2$. Let $D < 0$ *and* $D \equiv 0, 1 \pmod 4$, the primes p that are unramified in the quadratic imaginary field with discriminant D have form
$p = x^2 + |D|y^2$, where $x, y \in \mathbb{Z}$. There is roughly a fraction $1/2h_D$ of the primes that have this form [IGN06]. This means that we are expected to try $2h_D$ primes before a suitable one is found.

2.17 Hilbert Polynomials

Let $D < 0$ and $D \equiv 0, 1 \pmod 4$. Let Cl_D be the corresponding class group of order h_D. We assume that each element in Cl_D is represented by a unique reduced triplet (see section 2.5). Define

$$\tau_i = \frac{b_i + \sqrt{D}}{2a_i}, \quad 1 \le i \le h_D$$

We can assume that $\tau_i \in H$. To see this, note that $a_i c_i = \frac{b^2-D}{4}$ and $D < 0$ which implies $a_i > 0$ and $c_i > 0$ *or* $a_i < 0$ and $c_i < 0$, but $a_i c_i = (-a_i)(-c_i)$. Hence we can assume $a_i > 0$ and $b_i > 0$ which means $\Im(\tau_i) > 0 \Leftrightarrow \tau_i \in H$

The Hilbert polynomial is

$$H_D(X) = \prod_{i=1}^{h_D} (X - j(\tau_i)) \in \mathbb{Z}[X]$$

Let p be an unramified prime in $\mathbb{Q}(\sqrt{D})$. The roots of the Hilbert polynomial $H_D(X)$ over $\mathbf{F}_p$ will give j-invariants of elliptic curves with an equal number of points. We will denote U_D as the set of all roots of $H_D(X)$ over $\mathbf{F}_p$.

2.18 Torsion Points

Let E be an elliptic curve over a field K. Let l be a positive integer. We define

$$E[l] = \{(x, y) \in E(\overline{\mathbf{K}}) | [l](x, y) = P_\infty\}$$

It is easy to verify that $E[l]$ is a subgroup of $E(\overline{\mathbf{K}})$.

Theorem 1. *Let E be an elliptic curve over a field $\mathbf{K}$ and let l be a positive integer. If the characteristic of $\mathbf{K}$ does not divide l or characteristic of $\mathbf{K}$ is 0, then*

$$E[l] \simeq \mathbb{Z}_l \oplus \mathbb{Z}_l$$

If characteristic of $\mathbf{K}$ is some prime p and $p|l$, write $l = p^r l^{'}$ then

$$E[l] \simeq \mathbb{Z}_{l'} \oplus \mathbb{Z}_{l'} \text{ or } \mathbb{Z}_l \oplus \mathbb{Z}_{l'}$$

2.19 Endomorphism of an Elliptic Curve

Let $E : y^2 = x^3 + Ax + B$ be an elliptic curve over some field $\mathbf{K}$. An endomorphism of E is a homomorphism that is given by rational functions:

$$\alpha : E(\overline{\mathbf{K}}) \longrightarrow E(\overline{\mathbf{K}})$$

$$\alpha(x,y) = (R_1(x,y), R_2(x,y))$$

where, $R_1(x,y), R_2(x,y) \in \overline{\mathbf{K}}(x,y)$. Any endomorphism α of E can be written as

$$\alpha(x,y) = (r_1(x), y\ r_2(x))$$

where, $r_i(x) = p_i(x)/q_i(x) \in \overline{\mathbf{K}}(x)$ for $i = 1, 2$.

An endomorphism $\alpha \neq 0$ is defined to be **separable** if the derivative $r_1^{'}(x)$ is not identically zero. Note that an endomorphism of an elliptic curve E is nothing but an isogeny from E to itself. The set of all isogenies from an elliptic curve E to itself, together with a zero map forms a ring called *ring of endomorphisms* E and denoted by $\mathbf{End(E)}$. If E is not super singular [2], then $\mathbf{End(E)}$ is either equal to $\mathbb{Z}$ or an order in an imaginary quadratic field. If $\mathbf{End(E)}$ is equal to an **order** R in an imaginary quadratic field, then the curve E is said to have **Complex Multiplication**.

2.20 Frobenius Endomorphism

Let D be a fundamental discriminant and p be a prime that splits in $\mathbb{Q}(\sqrt{D})$ i.e. $p = x^2 + |D|y^2$ and $E(\overline{\mathbf{F}}_p)$ be an elliptic curve with $\#E(\mathbf{F}_p) = p + 1 - a$ for some integer $|a| \leq 2\sqrt{p}$. The frobenius endomorphism:

$$\phi_p : E(\overline{\mathbf{F}}_p) \longrightarrow E(\overline{\mathbf{F}}_p)$$

$$(x,y) \longmapsto (x^p, y^p)$$

satisfy its characteristic equation

$$\phi_p(x,y)^2 - a\phi_p(x,y) + p = P_\infty \quad \textit{for all } (x,y) \in E(\overline{\mathbf{F}}_p)$$

The discriminant D_ϕ of the characteristic equation is related to the discriminant of $\mathbb{Q}(\sqrt{D})$ in the following way:

$$D_\phi = a^2 - 4p = 4Dy^2 \Longrightarrow a = \pm 2x$$

[2] An elliptic curve over $\mathbf{F}_p$ is called supersingular if $E[p] \cong 0$

It is a very important relationship. It shows that the possible orders of any elliptic curve with complex multiplication by the ring of integers of $\mathbb{Q}(\sqrt{D})$ in $\mathbf{F}_p$ are $(p+1 \pm a)$ and each *j-invariant* will give rise to two elliptic curves with one having the order $p+1-a$ and the other $p+1+a$ [IBS06]. Let $\mathbb{Q}(\sqrt{D})$ be a quadratic imaginary field and p be a prime that splits in $\mathbb{Q}(\sqrt{D})$. Then the roots (*j-invariant*) of $H_D(X)$ (Hilbert polynomial) over $\mathbf{F}_P$ will correspond to two elliptic curves E and $E^{'}$, one having the order $p+1-a$ and the other $p+1+a$ and the equations of E and $E^{'}$ can easily be obtained from the *j-invariant*. Let j be any root of $H_D(X)$, then

$$E : y^2 = x^3 + 3kx + 2k \quad \textit{and} \quad E^{'} : y^2 = x^3 + 3kcx + 2kc$$

Where $k = j/(1728-j)$ and c is any non-quadratic residue in $\mathbf{F}_P$. The curves E and $E^{'}$ are called a quadratic twist of each other. Suppose we are given E and $E^{'}$ and we want to know which curve is of order $p+1-a$. We can choose a random point P on (say) E and check if $[p+1-a]P = P_\infty$. If $[p+1-a]P \neq P_\infty$, then $E^{'}$ has the order $p+1-a$. This is a probabilistic algorithm, which is very efficient and works well in practice.

2.21 Isogeny

The field of rational functions in two variables is

$$\overline{\mathbf{F}}_p(x,y) = \left\{ \frac{f(x,y)}{g(x,y)} \Big| f(x,y), g(x,y) \in \overline{\mathbf{F}}_p[x,y] \ \textit{and} \ g(x,y) \neq 0 \right\}$$

Let $E_1(\overline{\mathbf{F}}_p)$ and $E_2(\overline{\mathbf{F}}_p)$ be two elliptic curves. An **isogeny** I is a homomorphism that is given by rational functions

$$I(x,y) : E_1(\overline{\mathbf{F}}_p) \longrightarrow E_2(\overline{\mathbf{F}}_p)$$

Where $I(x,y) = (R_1(x,y), R_2(x,y))$ and $R_i(x,y) \in \overline{\mathbf{F}}_p(x,y)$ for *i= 1, 2.* Two elliptic curves are called isogenious if there exists an isogeny between them.

2.22 Degree of an Isogenies

Let

$$I(x,y) : E_1(\overline{\mathbf{F}}_p) \longrightarrow E_2(\overline{\mathbf{F}}_p)$$

be an isogeny, then kernel of I is a set

$$ker(I) = \{(x,y) \in E_1(\overline{\mathbf{F}}_p) | I(x,y) = P_\infty\}$$

The degree of $I(x,y)$ denoted $deg(I)$ is the number of elements in $ker(I)$. An isogeny is fully determined by its kernel. Suppose we are given

$$E_1(\mathbf{K}) := y_1 = x_1^3 + A_1 x + B_1 \text{ and } ker(I) \; with \; \#ker(I) = l$$

If $(x,y) \in ker(I)$ then so $(x,-y)$ because $ker(I)$ is a cyclic subgroup of $E[l]$ [ABaS96]. Let $d = \frac{l-1}{2}$ and $F = \{x_i | (x_i, y_j) \in E[l] \; and \; x_i \neq P_\infty\}$. Note each pair (x,y) and $(x,-y)$ in $E[l]$ will contribute one element (x coordinate) to F. So there are d elements in F. Define the kernel polynomial:

$$K(x) = \prod_{x_i \in F} (x - x_i) = x^d - p_1 x^{d-1} + p_2 x^{d-2} - ... + (-1)^d p_0$$

Our task is to compute $I_l(x,y)$ and a curve $E_2(\mathbf{K})$ from $K(x)$ such that

$$I_l(x,y) : E_1(\mathbf{K}) \longrightarrow E_2(\mathbf{K})$$

Define recursively for $1 \leq i \leq d-1$

$$h_i = (4i+2)p'_{i+1} + (4i-2)Ap'_{i-1} + (4i-4)Bp'_{i-2} \tag{1}$$

where $p'_0 = d$ and $p'_1 = p_1$. Then the equation of $E_2(\mathbf{K})$ is

$$E_2(\mathbf{K}) := y_2^2 = x^3 + A_2 x + B_2$$

where $A_2 = A - 5h_1$ and $B_2 = B - 7h_2$ and the isogeny I_l is given by

$$I_l(x,y) = \left(\frac{G(x)}{K(x)^2}, y \left(\frac{G(x)}{K(x)^2} \right)' \right)$$

where,

$$\frac{G(x)}{K(x)^2} = lx - 2p_1 - 2(3x^2 + A)\left(\frac{K(x)'}{K(x)}\right) - 4(x^3 + Ax + B)\left(\frac{K(x)'}{K(x)}\right)'$$

In section 2.29 we will show how to compute kernel polynomial $K(x)$ in polynomial time.

2.23 Composition of Isogenies

Let $I_1(x, y) : E_1(\overline{\mathbf{F}}_p) \longrightarrow E_2(\overline{\mathbf{F}}_p)$ and $I_2(x, y) : E_2(\overline{\mathbf{F}}_p) \longrightarrow E_3(\overline{\mathbf{F}}_p)$ be two isogenies. Then their composition is an isogeny

$$I_2(I_1(x, y)) : E_1(\overline{\mathbf{F}}_p) \longrightarrow E_3(\overline{\mathbf{F}}_p)$$

The degree of $I_2(I_1(x, y)) = deg(I_1)deg(I_2)$.

Theorem 2. *Elliptic curves are isogenous over $\mathbf{F}_p$ if and only if they have an equal number of points.*

Proof. See [Tat66]

Theorem 2 shows that given two elliptic curves it is very easy to decide if there exists an isogeny between them. However, it is believed to be difficult to determine the isogeny degree. In fact, the security of the cryptosystem proposed in [RS06] is based on this assumption.

2.24 Legendre and Kronecker Symbols

Suppose p is an odd prime. For any integer $a \geq 0$, we define the **Legendre symbol** $\left(\frac{a}{p}\right)$ as follows:

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{if } a \equiv 0 \pmod{p} \\ 1 & \text{if } a \text{ is a quadratic residue modulo } p \\ \text{-1} & \text{if } a \text{ is a non-quadratic residue modulo } p. \end{cases}$$

Let p be an odd prime. Then the congruence

$$x^2 \equiv a \pmod{p}$$

can have one solution if $a \equiv 0 \pmod{p}$, two solutions (we then say that a is a quadratic residue $\pmod{p}$) or no solutions (we then say that a is a non-quadratic residue $\pmod{p}$)

We define **Kronecker symbol** $\left(\frac{a}{b}\right)$ for any $a, b \in \mathbb{Z}$ in the following way.

1. If $b = 0$, then $\left(\frac{a}{0}\right) = 1$ if $a = \pm 1$, and is equal to 0 otherwise.
2. For $b \neq 0$, write $b = \prod p$, where the p are not necessarily distinct primes (including $p = 2$) or $p = -1$ to take care of the sign

$$\left(\frac{a}{b}\right) = \prod \left(\frac{a}{p}\right)$$

where $\left(\frac{a}{p}\right)$ is the *Legendre symbol* defined above for $p > 2$ and we define

$$\left(\frac{a}{2}\right) = \begin{cases} 0 & \text{if } a \text{ is even} \\ (-1)^{\frac{a^2-1}{8}} & \text{if } a \text{ is odd} \end{cases}$$

and also

$$\left(\frac{a}{-1}\right) = \begin{cases} 1 & \text{if } a \geq 0 \\ -1 & \text{if } a < 0 \end{cases}$$

Theorem 3. *Let E be an elliptic curve over $\mathbf{F}_p$ such that char$(\mathbf{F}_p) > 3$ and D_ϕ is the discriminant defined in section 2.20. Let l be a positive integer and $(\frac{D_\phi}{l})$ be a Kronecker symbol. If $(\frac{D_\phi}{l}) = -1$, then there is no elliptic curve that is l-degree isogenious to E; if $(\frac{D_\phi}{l}) = 1$, then there are exactly two elliptic curves over $\mathbf{F}_p$ that are l-degree isogenious to E; and if $(\frac{D_\phi}{l}) = 0$, then there is either 1 or $l + 1$ elliptic curves over $\mathbf{F}_p$ that are isogenious to E.*

Proof. See [Koh96]

2.25 Isogeny Cycle

If $(\frac{D_\phi}{l}) = 1$, then l is called an **Elkies isogeny degree**. Suppose D and D_ϕ are as above and let l be an *elkies isogeny degree*, let h_D be the corresponding class number and U_D denote the set of all roots of $H_D(X)$ over $\mathbf{F}_p$. The elements of U_D are connected to each other through isogeny I_l. **If h_D is a prime, then the elements of U_D form a single isogeny cycle** [aa06]. We will explain this by an example.

Example 2.

$D = -47$, $h_D = 5$, $p = 197 = 3^2 + (47)2^2$, $D_\phi = -752$ and Elkies isogeny degrees are $3, 7$. The roots (*j-invariants*) of Hilbert polynomial over $\mathbf{F}_p$ are $U_{-47} = \{17, 137, 105, 31, 118\}$

$$\begin{tikzcd} E_{17} \arrow[r, leftrightarrow, "3"] & E_{105} \arrow[r, leftrightarrow, "3"] & E_{118} \arrow[r, leftrightarrow, "3"] & E_{31} \arrow[r, leftrightarrow, "3"] & E_{137} \arrow[r, leftrightarrow, "3"] & E_{17} \end{tikzcd}$$

Figure 1: The isogeny cycle for degree 3

$$\begin{tikzcd} E_{17} \arrow[r, leftrightarrow, "7"] & E_{118} \arrow[r, leftrightarrow, "7"] & E_{137} \arrow[r, leftrightarrow, "7"] & E_{105} \arrow[r, leftrightarrow, "7"] & E_{31} \arrow[r, leftrightarrow, "7"] & E_{17} \end{tikzcd}$$

Figure 2: The isogeny cycle for degree 7

The $x \overset{l}{\longleftrightarrow} y$ means that there exist isogenies of degree l from x to y and y to x, and E_j means an elliptic curve with invariant j.

2.26 Route on Isogeny Cycles

Let $L = \{l_i | 1 \leq i \leq d \text{ and } l_i, d \in \mathbb{Z}^+\}$ be a set of Elkies isogeny degrees. A route is a set $R = \{r_i | 1 \leq i \leq d, r_i \in \mathbb{Z}\}$, where each r_i is the number of steps by the isogeny of degree l_i in the direction specified by the set $F = \{\phi_i\}$. We will explain how to specify the direction later, but before consider the example above. Let $R = \{2, 1\}$ and the direction we take is left starting from E_{118}. Then $R(118, left) = 105$.

$$\begin{tikzcd} E_{118} \arrow[r, "3"] & E_{31} \arrow[r, "3"] & E_{137} \arrow[r, "7"] & E_{105} \end{tikzcd}$$

Figure 3: Route $R = \{2, 1\}$ starting from E_{118}

Note that $R(118) = 105$ imply there exists an isogeny of degree $3^2 \times 7$ between elliptic curves E_{118} and E_{105}

2.27 Direction on Isogeny Cycle

In the example above we specify direction left on route $R = \{2, 1\}$ which is ambiguous because we could have written the same isogeny cycle in a different way. Consider the isogeny cycle above for degree 3.

$$\begin{tikzcd}E_{17} \arrow[r, leftrightarrow, "3"] & E_{137} \arrow[r, leftrightarrow, "3"] & E_{31} \arrow[r, leftrightarrow, "3"] & E_{118} \arrow[r, leftrightarrow, "3"] & E_{105} \arrow[r, leftrightarrow, "3"] & E_{17}\end{tikzcd}$$

Figure 4: The isogeny cycle for degree 3

It is exactly the same cycle, but $R(118, left) = 31 \neq 105$. To remedy this, we will explain how to specify direction, which is well defined, but before we need few definitions.

Let s, v be positive integers such that

$$s = \frac{12}{gcd(l-1, 12)} \tag{2}$$

$$v = \frac{s(l-1)}{12}$$

For isogeny degree l, define the **Müller′s modular polynomials**

$$G_l(x,y) = \textstyle\sum_{r=0}^{l+1} \sum_{k=0}^{v} a_{r,k} x^r y^k \in \mathbb{Z}\ [x,y]$$

Consider the characteristic equation $\phi_p(x,y)^2 - a\phi_p(x,y) + p = P_\infty$ over $\mathbb{Z}_l$. If $(\frac{D_\phi}{l}) = 1$, then it has two roots $\pi_1, \pi_2 \in \mathbb{Z}_l$ and π_1 gives one direction (say clockwise) and π_2 gives the other direction (say counterclockwise) on the isogeny cycle [CJM]. Suppose $E(\mathbf{F}_p) : y^2 = x^3 + Ax + B$, $(\frac{D_\phi}{l}) = 1$ and $p > 3$. Let π_1, π_2 be two roots of Frobenious charateristic equation. The direction is given by the following algorithm

$$\begin{tikzcd}E_2 & E \arrow[l, "\pi 2"'] \arrow[r, "\pi_1"] & E_1\end{tikzcd}$$

Algorithm 1.

```
• Input ($j(E)$, $l$, $\pi_i$, $p$).
    1. Compute polynomial $G_l(x,y)$.
    2. Factor polynomial $G_l(x, j(E))$ over $\mathbf{F}_p$. It will have two roots $F_1, F_2$ over $\mathbf{F}_p$. One corresponds to $\pi_1$ and the other to $\pi_2$.
    3. Choose $F_i$ that corresponds to $\pi_i$ (for details see [CJM]).
    4. Compute j-invariant of isogenious curve $j(E_i)$ using algorithm 2
    5. Output $j(E_i)$ .
```

Consider the example above. We are given the directions $F = \{\pi_3 = 1, \pi_7 = 2\}$ and route $R = \{2, 1\}$ starting from E_{17}. The $\ddot{M}uller's$ *modular polynomials* for degree 3 and 7 are

$G_3(x, y) = x^4 + 36x^3 + 270x^2 - xy + 756x + 729$
$G_7(x, y) = x^8 + 28x^7 + 322x^6 + 1904x^5 + 5915x^4 + 8624x^3 + 4018x^2 - xy + 748x + 49$

The roots of $G_3(x, 17)$ are $F_1 = 9$ and $F_2 = 40$. We choose F_1 because it corresponds to direction π_3 and computes *j-invariant*= $G_3(3^6/9, y) = 105$. In a similar way we compute ($F_{11} = 81$ and $F_{22} = 112$) the roots of $G_3(x, 105)$. Note that one root will take us back to E_{17} and the other to E_{118}. We note that $\frac{3^6}{81} = 9 = F_1$, therefore, we must disregard this one since we would go back to E_{118}; so we chose $F_{22} = 112$ and computed $G_3(3^6/112, y) = 118$. In a similar way we computed ($F_{33} = 101$ and $F_{44} = 132$) the roots of $G_7(x, 118)$ and choose $F_{33} = 101$ and found $G_7(7^2/101, y) = 137$. Hence, *R(E_{17}, F)=E_{137}*

2.28 Composition and Commutativity of Routes

Let $R_1 = \{r_i | 1 \leq i \leq d, r_i \in \mathbb{Z}\}$ and $S_1 = \{s_i | 1 \leq i \leq d, s_i \in \mathbb{Z}\}$ be two roots. The composition of two routes is defined as

$$R_1 S_1 = \{r_i + s_i | 1 \leq i \leq d\}$$

The routes are commutative

$$R_1 S_1 = S_1 R_1$$

Since,

$$R_1 S_1 = \{r_i + s_i | 1 \leq i \leq d\} = \{s_i + r_i | 1 \leq i \leq d\} = S_1 R_1$$

2.29 Computation of Isogeny

In this section we will give an algorithm which will take an Elliptic curve *E=(A, B)*, Elkies isogeny degree l and $G_l(x, y)$ as an input and output $E_2 = (A_2, B_2)$ (an isogenious elliptic curve) and p_1 (the coefficient of the second highest power of kernel polynomial defined in section 2.22). Recall that in section 2.22 we defined h's recursively using the coefficient of $K(x)$ (see equation 1). These h's can also satisfy the recurrence relation below:

$$h_k = \frac{3}{(k-2)(2k+3)}\sum_{i=1}^{k-2} h_i h_{k-1-i} - \frac{2k-3}{2k+3}Ah_{k-2} - \frac{2(k-3)}{(2k+3)}Bh_{k-3},\ 3 \le k \le d-1 \qquad (3)$$

With initial conditions:

$$h_1 = \frac{A - A_2}{5} \quad \textit{and} \quad h_2 = \frac{B - B_2}{7}$$

If we assume that p_1 is known and $l < 4p$ (p is the field characteristic), then we can compute $h_3, ..., h_d$ and $p_2, ..., p_d$ in $O(l \log l)$ operations in $\mathbf{F}_p$ using algorithm fastElkies from [ABaS96]. Please note that the $p_i's$ are the coefficient of kernel polynomial $\mathbf{K}(x)$ defined in section 2.22.

Algorithm 2. Isogeny Computation

- **Input** $((A, B), l, p, G_l(x, y))$.

1. Factor polynomial $G_l(x, j(E))$ over $\mathbf{F}_p$. It will have two roots $g, g^{'}$ over $\mathbf{F}_p$.

2. Let g be the root according to the direction (see section 2.27).

3. Set $\overline{E}_4$, $\overline{E}_6$ and Δ
$$\overline{E}_4 = \frac{-A}{3},\ \overline{E}_6 = \frac{-B}{2} \ \textit{and}\ \Delta = \frac{\overline{E}_4^3 - \overline{E}_6^2}{1728}$$

4. Set $j = j(E)$ and compute D_g and D_j as follows. Please note that notation indicates that the derivatives are to be evaluated at (g, j)
$$D_g = g\left(\frac{\partial}{\partial x}G_l(x, y)\right)(g, j)$$
$$D_j = j\left(\frac{\partial}{\partial y}G_l(x, y)\right)(g, j)$$

5. Set $\Delta^{(l)} = l^{-12}\Delta g^{12/s}$, where $s = 12/gcd(l-1, 12)$.

6. If $D_j \neq 0$, then set
$$\overline{E}_2^* = \frac{-12\overline{E}_6 D_j}{s\overline{E}_4}D_g,\ g^{'} = -(s/12)\overline{E}_2^* g,\ j^{'} = -\overline{E}_4^2\overline{E}_6\Delta^{-1},\ \overline{E}_0 = \overline{E}_6(\overline{E}_4\overline{E}_2^*)^{-1}$$

7. Compute the quantities

$$D_g^{'} = g^{'}\left(\frac{\partial}{\partial x}G_l(x,y)\right)(g,j) + g\left[g^{'}\left(\frac{\partial^2}{\partial x^2}G_l(x,y)\right)(g,j) + j^{'}\left(\frac{\partial^2}{\partial x \partial y}G_l(x,y)\right)(g,j)\right]$$

$$D_j^{'} = j^{'}\left(\frac{\partial}{\partial y}G_l(x,y)\right)(g,j) + j\left[j^{'}\left(\frac{\partial^2}{\partial y^2}G_l(x,y)\right)(g,j) + g^{'}\left(\frac{\partial^2}{\partial y \partial x}G_l(x,y)\right)(g,j)\right]$$

8. Set

$$\overline{E}_0^{'} = \frac{1}{D_j}\left(\frac{-s}{12}D_g^{'} - \overline{E}_0 D_g^{'}\right)$$

9. Set $\overline{E}_4^{(l)} = \frac{1}{l^2}\left(\overline{E}_4 - \overline{E}_2^{*}\left[12\frac{\overline{E}_0^{'}}{\overline{E}_0} + 6\frac{\overline{E}_4^{\;2}}{\overline{E}_6} - 4\frac{\overline{E}_6}{\overline{E}_4}\right] + \overline{E}_2^{*\,2}\right)$

10. Set $j(E^{'}) = \overline{E}_4^{(l)^3}/\Delta^{(l)}$ **(j-invariant of isogenious curve)**.

11. Set $f = l^s g^{-1}$ and $f^{'} = \overline{E}_2^{*} s f/12$

12. Compute

$$D_g^{*} = \left(\frac{\partial}{\partial x}G_l(x,y)\right)(f, j(E^{'})),\ D_j^{*} = \left(\frac{\partial}{\partial y}G_l(x,y)\right)(f, j(E^{'}))\ and\ j^{l} = -f^{'}\frac{D_g^{*}}{lD_j^{*}}$$

13. Set

$$\overline{E}_6^{l} = \frac{\overline{E}_4^{(l)} j^{l}}{j(E^{'})}$$

14. Compute A_2, B_2 *and* p_1 as follows.

$$A_2 = -3l^4\overline{E}_4^{(l)}, \quad B_2 = -2l^6\overline{E}_6^{(l)}, \quad and \quad p_1 = -\frac{l\overline{E}_2^{*}}{2}$$

15. **Output** (p_1, A_2, B_2).

The computational complexity of Algorithm 2 is $O(l^2)$ operations in $\mathbf{F}_p$ [IBS06]. Now once we have computed (A_2, B_2, p_1), we can compute $h_1, h_2, ..., h_d$ using recurrence 3 in $O(l^2)$ operation in $\mathbf{F}_p$ or in $O(l \log l)$ using the method described in [ABaS96]. Once the h_i's are computed, we can compute $p_0^{'}, p_1^{'}, ..., p_d^{'}$ using the recurrence 1 in $O(l)$ operations in $\mathbf{F}_p$. Then we compute the polynomial

$$K^{'}(x) = (-1) \times \frac{p_1^{'}}{1}x^1 + \frac{p_2^{'}}{2}x^2 + ... + \frac{p_d^{'}}{d}x^5 \mod p$$

We have

$$x^d K(1/x) = \exp_{d+1}(K^{'}(x))$$

but $\exp_{d+1}(K^{'}(x))$ itself is a polynomial of degree d. Let $\exp_{n+1}(K^{'}(x)) = a_0 + a_1 x + a_2 x^2 ... + a_d x^d$. Therefore we can write

$$x^d K(1/x) = a_0 + a_1 x^1 + a_2 x^2 + ... + a_d x^d$$

$$\Rightarrow K(1/x) = a_0 x^{-d} + a_1 x^{2-d} + a_1 x^{2-d} + ... + a_d$$

$$\Rightarrow K(x) = a_0 x^d + a_1 x^{d-1} + a_1 x^{d-2} + ... + a_d$$

The $\exp_{n+1}(K^{'}(x))$ can be computed in $O(l \log l)$ operations using the method described in [ABaS96].

Chapter 3

Side Channel Attacks on Elliptic Curve Cryptosystems

The advantage of elliptic curve based cryptosystems over other cryptosystems (**RSA**, etc.) is their short key size. For this purpose it is suitable for implementing on memory-constrained devices such as smart cards, mobile devices, etc. A cryptographic device such as a smart card uses a private key to process input information. The designer of the cryptosystem assumes that the attacker has pairs of plaintext/ciphertext, key sizes, but other secrets will be manipulated in closed and safe computing environments. However these devices leak information about the private key through side channels (power consumption, electromagnetic radiation, timing, etc.) during cryptographic processing. The term “side channel” is used to describe the leakage of unintended information during cryptographic processing from a supposedly tamper-resistant device, such as smart cards. Hence, the side channel attacks are practical attacks as opposed to theoretical attacks (e.g. differential cryptanalysis attack on **DES**, etc). There are many kinds of side channel attacks such as timing attacks, power analysis attacks, electromagnetic attacks, etc. The side channel attack we are interested in is power analysis. Note that we will assume throughout this chapter that we are working on elliptic curves defined over prime fields $\mathbf{F}_p$, with $p > 3$ or binary fields $\mathbf{F}_{2^m}$, $m \geq 1$.

3.1 Elliptic Curve Cryptosystems

Common Parameters

- An elliptic curve E over $\mathbf{F}_p$ or $\mathbf{F}_{2^m}$.
- The order of $\#E$ must be divisible by a large prime q.
- $P \in E$.

Private Key

- $d \in [1, q-1]$ chosen randomly.

Public Key

- $Q = [d]P$.

Encryption Of message m

- Pick a random $n \in [1, q-1]$.
- Compute the points $(x_1, y_1) = [n]P$ and $(x_2, y_2) = [n]Q$.
- Compute $c = x_2 + m$.
- Output ciphertext (x_1, y_1, c).

Decryption

- Receive ciphertext (x_1, y_1, c).
- Compute $(x, y) = [d](x_1, y_1)$ and $m = c - x$.

3.2 Power Analysis Attack

In a power analysis attack the side channel is the device's power consumption. A power analysis attack works by exploiting the fact that a temper-resistant device such as a smart card consumes different amount of power if it is processing 0 or 1. The reason for this power variation is that integrated circuits are built out of individual transistors, which act as voltage-controlled switches. The current flows across the transistor substrate when a charge is applied to (or removed from) the gate. This current then delivers a charge to the gates of other transistors, interconnect wires and other circuit loads. The motion of the electric charge consumes power and produces electromagnetic radiation, both of which are externally detectable. Therefore, individual transistors produce externally observable electrical behavior. Because microprocessor logic units exhibit regular transistor switching patterns, it is possible to easily identify microprocessor activity by simply monitoring the power consumption [PKJ98].

There are two types of power analysis attacks: one is called **Simple Power Analysis** (*SPA*) and the other is **Differential Power Attack** (*DPA*). We will first describe the *SPA* attack on implementation of elliptic curve cryptosystem and then the *DPA* attack.

3.3 Simple Power Analysis

A simple power analysis attack consists of observing the power consumption of one single execution of a cryptographic algorithm. Let E be an elliptic curve and P be a point on it. The operation of adding a point P to itself d times is called scalar multiplication by d and denoted by $[d]P$. The simplest and oldest efficient method for scalar multiplication is called **binary method**.

Algorithm 3. *(Binary Method)*

- **Input** P, $d = \sum_{j=0}^{l-1} d_j 2^j$, where $d_j \in \{0, 1\}$.
- $Q \leftarrow P$.

```
1. for $i$ from $l-2$ to 0 do
2.     $Q \leftarrow [2]Q$.
3.     if $d_i = 1$ then $Q \leftarrow Q + P$.
4. **Output** $Q$.
```

Let **A** and **D** denote the cost of elliptic point addition and doubling in finite fields. The algorithm 3 requires in total $[(l-1)]\mathbf{A} + [(W-1)]\mathbf{D}$ field operations, where W is the weight (number of 1s) in the binary representation of d.

Suppose an attacker knows P, then by monitoring the power consumption during the computation of $Q = [d]P$, he/she can recover the private key d, since we perform step 3 only if $d_i = 1$, the power consumption will be more when $d_i = 1$ thus revealing the bits of the private key d. Algorithm 3 can easily be modified so that step 3 is performed no matter what the secret bit is.

Algorithm 4. *(Always-add-double)*

- **Input** P, $d = \sum_{j=0}^{l-1} d_j 2^j$, where $d_i \in \{0, 1\}$.

```
• $Q[0] \leftarrow P$.
    1. for $i$ from $l-2$ to 0 do
    2.     $Q[0] \leftarrow [2]Q[0]$.
    3.     $Q[1] \leftarrow Q[0] + P$.
    4.     $Q[0] \leftarrow Q[d_j]$.
    5. Output $Q[0]$.
```

The computational cost of this countermeasure is $(l-1)\mathbf{A} + (l-1)\mathbf{D}$. The algorithm 4 is secure against the *SPA* attack defined above. We assume that algorithm 3 is performed in constant time (i.e. The time for each *i-th* loop is the same). Otherwise the implementation can be subject to timing attacks [Koc96]. We further assume that the power consumption cost of the assignment (step 4) is constant.

3.4 Differential Power Analysis DPA

A DPA is a more powerful attack. It consists of performing statistical analysis of several execution of the same algorithm with many different inputs. We will show in this section that algorithm 4 is insecure against *DPA* attack [Cor99]. Let the binary expansion of the private key be $d = d_{l-1}, ..., d_0$ where d_{l-1} is the most significant bit of d. Suppose algorithm 4 is used in computing $Q = [d]P$. We notice that if $d_{l-2} = 0$ then in the next loop ($i = l-3$) algorithm 4 will compute the values $[4]P$ and $[5]P$ and if $d_{l-2} = 1$ algorithm 4 will compute the values $[6]P$ and $[7]P$. In general we notice that at step j of algorithm 4 the value of point Q depends only on the most significant bits $d_{l-1}, ..., d_{j+1}$ of d. Suppose the attacker knows the bits $(d_{l-1}\ ,\ ...\ d_{j+1})$ of d. He/She will guess $d_j = 1$ (or $d_j = 0$). Then the attacker will pick random points $P_1,\ ...\ ,\ P_n$ and compute

$$Q_r = \left[\sum_{j=i}^{l-1} d_j 2^j\right] P_r, \quad 1 \leq r \leq n$$

Let s any fixed bit of points Q_r, $1 \leq r \leq n$. The attacker divides points Q_r in two sets: one set contains points for which $s = 1$ and the other set contains points for which $s = 0$. Let $C(r)$ be the power consumption function associated with *r-th* execution of algorithm 4. If the guess bit d_{l-1} is incorrect, then

$$G(r) = \langle C(r)\rangle_{r=1,2\ldots,n|s=1} - \langle C(r)\rangle_{r=1,2\ldots,n|s=0}$$

$G(r) \approx 0$ as the two sets appear uncorrelated, otherwise the guess key is correct. Once d_j is known, the remaining bits can be found recursively using the same method.

3.5 Countermeasures Against DPA

In [Cor99] Coron proposed three countermeasures against the above *DPA* attack. They are

1. Randomization of the private key d.

2. Adding a random point R to the base point P.

3. Using randomized projective coordinates.

3.6 Coron's First Countermeasure:Randomization of the Private Key d

Let $\#E$ be the number of points on the elliptic curve. The computation of $Q = [d]P$ is done as follows

1. Choose a random number n of size k bits . In practice, one can take $k = 20$ bits.

2. Compute $d^{'} = d + n\#E$.

3. Compute the point $Q = [d^{'}]P$ (using algorithm 4).

4. Output Q (Note that $[d^{'}]P = [d]P$, since $[n\#E]P = P_\infty$).

This countermeasure makes the previous attack infeasible since exponent $d^{'}$ changes at each new execution. Let $\mathbf{R}$ denote the cost of random number generation in finite fields. The computational cost of this countermeasure is

$$[(l-1)+(k-1)]\mathbf{A} + [(l-1)+(k-1)]\mathbf{D} + \mathbf{R}$$

If we take $k = 20$, then the cost is

$$[(l-1)+19]\mathbf{A}+[(l-1)+19]\mathbf{D}+\mathbf{R}$$

3.7 Coron's Second Countermeasure:Blinding the Base Point

Let R and S be two points such that $S=[d]R$. The points R and S are stored initially inside the device and refreshed after each new execution. The computation of $Q=[d]P$ is done as follows.

1. Compute $Q^{'}=[d](R+P)$ (using algorithm 4).
2. Compute the point $Q=Q^{'}+(-S)$ using algorithm 4.
3. Set $R\longleftarrow(-1)^r 2R$ and $S\longleftarrow(-1)^r 2S$, where r is a random bit.
4. Output Q.

This countermeasure makes the previous attack infeasible since the point $P+R$ used to compute the scalar product remain unknown to the attacker. The computational cost of this countermeasure is $[(l-1)]\mathbf{A}+[(l-1)]\mathbf{D}$ plus 2 addition, 2 point doubling and 1 random number generation $\mathbf{R}$. Hence, the total cost is

$$[(l-1)]\mathbf{A}+[(l-1)]\mathbf{D}+2\mathbf{A}+2\mathbf{D}+\mathbf{R}=[(l+1)]\mathbf{A}+[(l+1)]\mathbf{D}+\mathbf{R}$$

3.8 Third countermeasure: Randomization in Projective Coordinates

Recall that in section 2.12 we defined a two dimensional projective plane $\mathbf{P}^{\mathbf{2}}_{\mathbf{K}}$ over $\mathbf{K}$ as a set of equivalence classes of triples $[(X,Y,Z)]$ with $X,Y,Z\in\mathbf{K}$ and not all X,Y,Z are zero. Two triples (X_1,Y_1,Z_1) and (X_2,Y_2,Z_2) are said to be equivalent if there exists a nonzero $\alpha\in\mathbf{K}$ such that

$$(X_1,Y_1,Z_1)=(\alpha^2X_2,\alpha^3Y_2,\alpha Z_2) \tag{4}$$

Let $P=(x,y)$, be the base point. The computation of $Q=[d]P$ is done as follows.

- Map affine coordinates (x,y) to projective coordinates.

$$X \leftarrow x, Y \leftarrow y, Z \leftarrow 1$$

- Choose a random $\alpha \in \mathbf{K}^*$ and using relation 4 we randomize the point P as

$$P = (\alpha^2 x, \alpha^3 y, \alpha)$$

- Compute $Q = [d]P = (X, Y, Z)$ using algorithm 4.
- Compute Q in affine coordinates by setting

$$x \longleftarrow X/Z^2 \ , \ y \longleftarrow X/Z^3$$

- **Output** $Q = (x, y)$.

This countermeasure makes the DPA attack infeasible since the representation of point P in projective coordinates remains unknown to the attacker. Let M and R denote the computational cost of multiplication and random number generation in finite fields. This countermeasure is the most efficient among Coron's countermeasures as it requires 3 additional multiplications (3 **M**) and one random number generation (**R**) in **K**. The computational cost of this countermeasure is

$$[(l-1)]\mathbf{A} + [(l-1)]\mathbf{D} + 3\mathbf{M} + \mathbf{R}$$

Therefore, the cost of this measure is almost the same as for algorithm 4. The projective coordinates are also more efficient than affine coordinates, since we can avoid computing field inversions during the computation of $Q = [d]P$.

3.9 Algebraic Countermeasures

Joye and Tymen [JT01] also proposed two countermeasures against Coron's DPA. In the first countermeasure, they chose a random isomorphic elliptic curve and computed the scalar multiplication over that curve. In the second countermeasure they computed the scalar multiplication in an isomorphic field which is also chosen randomly.

3.10 Fourth Countermeasure: Randomizing the Base Point Through a Random Isomorphic Elliptic Curve

Let $\mathbf{K}$ be any field whose characteristic is neither 2 nor 3, then any two elliptic curves $E_1(\mathbf{K}) = x^3 + a_1x + b_1$ and $E_2(\mathbf{K}) = x^3 + a_2x + b_2$ are isomorphic over $\mathbf{K}$; if and only if there exists $u \in \mathbf{K}^*$ such that $a_1 = u^4a_2$ and $b_1 = u^6b_2$. Moreover, if $E_1 \cong E_2$ over $\mathbf{K}$, then the isomorphism is given by

$$\Psi : E_1(\mathbf{K}) \longrightarrow E_2(\mathbf{K}), \quad \Psi : (x, y) \longmapsto (u^{-2}x, u^{-3}y)$$

or equivalently

$$\Psi^{-1} : E_2(\mathbf{K}) \longrightarrow E_1(\mathbf{K}), \quad \Psi^{-1} : (x, y) \longmapsto (u^2x, u^3y)$$

Let $E(\mathbf{K}) = x^3 + a_1x + b_1$ be an elliptic curve such that the characteristic of field $\mathbf{K} \neq 2, 3$. The computation of $Q = [d]P$ is done as follows:

- **Input** $(P = (x, y),\ d,\ E_1(\mathbf{K}) = (a, b))$.

  1. Randomly choose an element $u \in \mathbf{K}^*$.
  2. Set $a_2 \longleftarrow u^{-4}a_1$ (note we do have to compute $b_2 = u^{-6}b_1$).
  3. Set $P_2 \longleftarrow \Psi(P_1) = (u^{-2}x, u^{-3}y)$.
  4. Compute $Q_2 = [d]P_2$ in $E_2(\mathbf{K}) = x^3 + a_2x + b_2$ using algorithm 4.
  5. If $(Q_2 = P_\infty)$, then set $Q_1 \longleftarrow P_\infty$
     Else set $Q_1 \longleftarrow \Psi^{-1}(Q_2) = (u^2x, u^3y)$.
  6. **Output** Q_1.

Let $\mathbf{M}$ and $\mathbf{R}$ denote the cost of multiplication and random number generation in finite fields. The computational cost of this countermeasure is

$$[(l-1)]\mathbf{A} + [(l-1)]\mathbf{D} + 11\mathbf{M} + \mathbf{R}$$

It has been suggested that in step 3 of the above countermeasure, one can map P_2 to projective point $P_2^{'} = [(u^{-2}x, u^{-3}y, 1)]$ [Cor99]. This will make this method more efficient since computation of $[d]P$ in projective coordinates is faster than in affine coordinates.

3.11 Fifth Countermeasure:Randomizing the Representation of Base Point Through a Random Field Isomorphism

Let $\mathbf{K}$ be a finite field of characteristic p, and $f(x) \in \mathbf{K}[\mathbf{x}]$ be a polynomial of degree greater than zero. We say that $f(x)$ is irreducible over $\mathbf{K}[\mathbf{x}]$ if it cannot be written as product of two polynomials in $\mathbf{K}[\mathbf{x}]$ both of degree greater than zero. Let $f(x) \in \mathbf{K}[\mathbf{x}]$ be an irreducible polynomial. Let $h(x) \in \mathbf{K}[\mathbf{x}]$. The congruence class of $h(x)$ modulo $f(x)$ is denoted $[h(x)]$ and consists of all polynomials in $\mathbf{K}[\mathbf{x}]$ that are congruent to $h(x)$ modulo $f(x)$.

$$[h(x)] = \{g(x)|g(x) \in \mathbf{K}[\mathbf{x}] \; and \; f(x)|(g(x) - h(x))\}$$

We denote a set of all equivalence classes modulo $f(x)$ is denoted

$$\mathbf{K}[\mathbf{x}]/f(x)$$

where $\mathbf{K}[\mathbf{x}]/f(x)$ is a finite field of order p^n for some $n \geq 1$ [Hun96]. Every finite field $\mathbf{K}$ has characteristic p and order p^n for some prime p and integer $n \geq 1$ [Hun96]. If $\mathbf{K}$ has order p, then $\mathbf{K} \cong \mathbf{F}_p$, and if order of $\mathbf{K}$ is (p^n) $n \geq 1$ then $\mathbf{K} \cong \mathbf{F}_p[x]/(f(x))$ [Hun96]. It follows that if $g(x) \neq f(x)$ and $g(x) \in \mathbf{F}_p[x]$ is any irreducible monic polynomial of degree n, then $\mathbf{F}_p[x]/(f(x)) \cong \mathbf{F}_p[x]/(g(x))$. Furthermore $\mathbf{F}_p[x]/(f(x))$ contains a root α of $g(x)$ and $\mathbf{F}_p[x]/(g(x))$ contains a root β of $f(x)$. The ϕ isomorphism between two fields is given by [JT01].

$$\phi : \mathbf{F}_p[x]/(f(x) \longrightarrow \mathbf{F}_p[x]/(g(x), \quad \phi : h(x) \longmapsto h(\alpha)$$

and

$$\phi^{-1} : \mathbf{F}_p[x]/(g(x)) \longrightarrow \mathbf{F}_p[x]/(f(x)), \quad \phi^{-1} : q(x) \longmapsto q(\alpha)$$

Let $\mathbf{K} = \mathbf{F}_p[x]/(f(x))$ and $\mathbf{K}^{'} = \mathbf{F}_p[x]/(g(x))$. The isomorphism ϕ extends to $\mathbf{K} \times \mathbf{K}$ as

$$\phi^{'} : \mathbf{K} \times \mathbf{K} \longrightarrow \mathbf{K}^{'} \times \mathbf{K}^{'}, \quad \phi^{'}(x, y) \longmapsto (\phi(x), \phi(y))$$

If $E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ is an elliptic curve over $\mathbf{K}$, then the equation of elliptic curve $E^{'}$ over $\mathbf{K}^{'}$ is given by $E^{'} : y^2 + \phi(a_1)xy + \phi(a_3)y = x^3 + \phi(a_2)x^2 + \phi(a_4)x + \phi(a_6)$

The computation $Q = [d]P$ using this countermeasure is done as follows.

- **Input** ($\mathbf{K} = \mathbf{F}_{2^m},\ d,\ P,\ E$).

  1. Choose a field $\mathbf{K}^{'}$ isomorphic to $\mathbf{K}$ through ϕ.
  2. Set $P^{'} \leftarrow \phi(P)$.
  3. Compute $Q^{'} = [d]P^{'}$ in $E^{'}$, using algorithm 4 .
  4. Set $Q \leftarrow \phi^{-1}(Q^{'})$.
  5. **Output** Q.

3.12 Weakness of First Countermeasure:Randomization of the Private Key d

The drawback of this countermeasure is that it does not properly randomize the secret key d. Hence the bits of d are correlated to $d^{'}$. We will explain this by an example. For simplicity, we assume that n is a two bit number and the least significant bits of $\#E$ are 001. Since $d^{'} = d + n\#E$, the possible values of $d^{'}_2d^{'}_1d^{'}_0$ are given in table 1. An attacker can find $d^{'}$ by applying DPA attack described in section 3.4. Since $d^{'}$ is different for every input, the attacker can compute several $d^{'}$ using DPA and find the frequency of $d^{'}_2$, from which they will be able to estimate d. For example, if the attacker finds that the probability $d^{'}_2 = 1$ in $d^{'}$ is 100%, then by using the statistical table above he/she is able to estimate that $d_2d_1d_0 = 100$. In general, least significant bits of $\#E$ may not be 001, however an attacker knows the value of $\#E$ since it is public. If we take n to be a 20-bit number (which is about 1 million) as recommended in [Cor99], then one can easily write a statistical table as above and deduce some information about the private key. We recall from section 3.6 that the computational cost of this countermeasure is

$d_2d_1d_0$	000	001	010	011	100	101	110	111
$d^{'}_2d^{'}_1d^{'}_0$, $(n = 00)$	000	001	010	011	100	101	110	111
$d^{'}_2d^{'}_1d^{'}_0$, $(n = 01)$	001	010	011	100	101	110	111	000
$d^{'}_2d^{'}_1d^{'}_0$, $(n = 10)$	010	011	100	101	110	111	000	001
$d^{'}_2d^{'}_1d^{'}_0$, $(n = 11)$	011	100	101	110	111	000	001	010
probability $d^{'}_2 = 1$	0%	25%	50%	75%	100%	75%	50%	25%

Table 1:

$$[(l-1)+19]\mathbf{A}+[(l-1)+19]\mathbf{D}+\mathbf{R}$$

In order to thwart this attack, d should be randomized properly, in other words $\log_2 n = \log_2 d$. The computational cost of which comes to

$$[2(l-1)]\mathbf{A}+[2(l-1)]\mathbf{D}$$

3.13 Weakness of Second Countermeasure:Blinding the Base Point

Okeya and Sakurai present an attack on this method [Os00]. We recall from section 3.7 that in this countermeasure we initially stored two points R and S in the cryptographic device and after each new execution we refreshed R and S. Let R_j and S_j denote the values at the *j-th* execution, then

$$R_j = (-1)^r R \; and \; S_j = (-1)^r S$$

The attack is as follows. Let P, $2P$,..., $2^k P$ be points on an elliptic curve and d be the secret scalar. Let t_0 be the time required for each round. We assume t_0 is constant. Let $C_j(t)$ be the power consumption function associated with each execution of $[d](2^j P)$. We define a correlation function $g(t)$.

$$g(t) = \frac{1}{k}\sum_{j=0}^{k-1} min\left\{\frac{1}{(C_j(t+t_0-C_{j+1}(t)))^2}, MAXVAL\right\}$$

where $MAXVAL$ is some big number in case $g(t)$ vanishes. Let t_1 be a time where $g(t)$ does not vanish and n_1 be an integer satisfying

$$(n_1-1)t_0 < t_1 < n_1 t_0$$

Then we find $d_{l-1-n_1} = 0$. Similarly, if the function vanishes for $t^{'}$ then we find $n^{'}$ as above and conclude $d_{l-1-n'} = 1$.

The drawback of this countermeasure is that the number of possibilities of R after many executions are few and the number of possible values for the next R is only two. It has been suggested that in order to thwart the above attack, the refreshing method of R and S described in section 3.7

should be modified to the following:

1. Compute $Q^{'} = [d](R + P)$ (using algorithm 4).
2. Compute the point $Q = Q^{'} + (-S)$ using algorithm 4.
3. Pick a random n bit integer k, (for example $n = 20$).
4. Set $R \longleftarrow [k]R$ and $S \longleftarrow [k]S$. Note that computations of $[k]R$ and $[k]S$ is done using algorithm 4[1]
5. Output Q.

The computational cost of this countermeasure is

$$[(l+1)]\mathbf{A} + [(l+1)]\mathbf{D} + 2 \times [(n-1)]\mathbf{A} + [(n-1)]\mathbf{D}] + \mathbf{R}$$

$$= [(l+2n-1)]\mathbf{A} + [(l+2n-1)]\mathbf{D} + \mathbf{R}$$

If we take $n = 20$, then the computational cost will be

$$[(l-39)]\mathbf{A} + [(l+39)]\mathbf{D} + \mathbf{R}$$

Which is computationally less costly then the modified countermeasure 1.

3.14 Special Points and Countermeasure 3, 4 and 5

We saw that the problem with Coron's first two countermeasures is that they do not properly randomize the secret scalar d or base point P which allows the attacker to retrieve information about the scalar using DPA attack. The last three countermeasures (randomized projective coordinates, random elliptic curve isomorphism and random field isomorphism) seem to randomize base point P. However Goubin [Gou03] observed that points $(0, y)$ and $(x, 0)$(called special points) cannot be properly randomized by any of these anti-DPA methods. Hence if these special points lie on the elliptic curve then they can be used to launch a DPA attack on the elliptic curve scalar multiplication.

[1] Note in this step this measure differs from countermeasure 3.7

3.15 Random Projective Coordinates and Special Points

Consider the points $(x, 0)$ and $(0, y)$ in affine coordinates over a finite field $\mathbf{K}$. The corresponding points in projective coordinates are $(x, 0, 1)$ and $(0, y, 1)$. We randomized these points by multiplying each coordinate with some random $r \in \mathbf{K}^*$

$$(r^2x, 0, r) \quad and \quad (0, r^3y, r)$$

These points still remain of form $(X, 0, Z)$, $(0, Y, Z)$ and are not properly randomized.

3.16 Random Elliptic Curve Isomorphism and Special Points

Let $\mathbf{K}$ be a field with $char(\mathbf{K}) \neq \mathbf{2}, \mathbf{3}$. Recall from section 3.10 that, if $E_1(\mathbf{K})$ and $E_2(\mathbf{K})$ are two isomorphism elliptic curves, then their isomorphism is given by

$$\Psi : E_1(\mathbf{K}) \longrightarrow E_2(\mathbf{K}), \quad \Psi : (x, y) \mapsto (u^{-2}x, u^{-3}y), \quad u \in \mathbf{K}^*$$

if $(x, 0) \in E_1(\mathbf{K})$ or $(0, y) \in E_1(\mathbf{K})$, then clearly $\Psi(x, 0) = (u^{-2}x, 0) \in E_2(\mathbf{K})$ or $\Psi(0, y) = (0, u^{-3}y) \in E_2(\mathbf{K})$. Hence, special points remain of form $(x, 0)$ or $(0, y)$. It is easy to see that even if we use projective coordinates in this method the special points will remain of form $(X, 0, Z)$ or $(0, Y, Z)$

3.17 Random Field Isomorphism and Special Points

Recall that (section 3.11) if ϕ is an isomorphism between finite fields, then ϕ is given by

$$\phi : \mathbf{F}_p[x]/f(x) \longrightarrow \mathbf{F}_p[x]/g(x), \quad \phi : h(x) \longmapsto h(\alpha)$$

where α is a root of $f(x)$ in $\mathbf{F}_p[x]/g(x)$. The isomorphism ϕ extends to $\mathbf{F}_p[x]/g(x) \times \mathbf{F}_p[x]/g(x)$ as

$$\phi' : \mathbf{F}_p[x]/(f(x)) \times \mathbf{F}_p[x]/(f(x)) \longrightarrow \mathbf{K}' \times \mathbf{K}', \quad \phi'(x, y) \longmapsto (\phi(h_1(x)), \phi(h_2(y))$$

If E is an elliptic curve over the field $\mathbf{F}_p[x]/f(x)$ and if $(h_1(x), 0)$ or $(0, h_2(x))$ lies on the elliptic curve and E' is the corresponding curve over $\mathbf{F}_p[x]/g(x)$, then the map ϕ' will take

$$\phi^{'} : E \longrightarrow E^{'}$$

$$\phi^{'}(h_1(x), 0) \longmapsto (\phi(h_1(x), 0)) \quad and \quad \phi^{'}(0, h_2(x)) \longmapsto (0, \phi(h_2(x)))$$

hence special point remains of form $(x, 0)$ and $(0, y)$.

3.18 Refined Power Analysis Attack Using Special Points (Goubin Attack)

Let E be an elliptic curve over a finite field $\mathbf{K}$ and suppose we use algorithm 4 for scalar multiplication $Q = [d]P$ together with any one of the three anti-DPA methods (randomized projective coordinates, random elliptic curve isomorphism and random field isomorphism). Let d be a l bit private key and suppose $P_0 \in E(\mathbf{K})$ be a point such that $P_0 \neq P_\infty$ and $P_0 = (x, 0)$ or $P_0 = (0, y)$. The attack is recursive therefore we assume that the attacker knows the most significant $i - 1$ bits $(d_{l-1}, ..., d_{i+1})$ of d . First we note that the value Q_i we obtain at the end of the *i-th* loop is

$$Q_i = \left(\sum_{j=i+1}^{l-1} d_j 2^{j-i} + d_i \right) P$$

If the bit $d_i = 0$, then during the $(i+1) - th$ loop of the algorithm the values appear in variable $Q[0]$ and $Q[1]$ are

$$Q[0] \leftarrow \left(\sum_{j=i+1}^{l-1} d_j 2^{j-i+1} \right) P \quad and \quad Q[1] \leftarrow \left(\sum_{j=i+1}^{l-1} d_j 2^{j-i+1} + 1 \right) P$$

If the bit $d_i = 1$, then during the $(i+1) - th$ loop of the algorithm the values appear in variable $Q[0]$ and $Q[1]$ are

$$Q[0] \leftarrow \left(\sum_{j=i+1}^{l-1} d_j 2^{j-i+1} + 2 \right) P \quad and \quad Q[1] \leftarrow \left(\sum_{j=i+1}^{l-1} d_j 2^{j-i+1} + 3 \right) P$$

Now the attacker will guess the secret bit d_i as follows. He chooses a base point P_1

- *if* $\gcd\left(\left(\sum_{j=i+1}^{l-1} d_j 2^{j-i+1} + 1\right), \#E\right) = 1$, then set

$$P_1 \longleftarrow \left(\sum_{j=i+1}^{l-1} d_j 2^{j-i+1} + 1 \right)^{-1} P_0$$

$$OR$$

- *if* gcd $\left(\left(\sum_{j=i+1}^{l-1} d_j 2^{j-i+1} + 3 \right), \#E \right) = 1$, then set

$$P_1 \longleftarrow \left(\sum_{j=i+1}^{l-1} d_j 2^{j-i+1} + 3 \right)^{-1} P_0$$

An attacker will feed the same point P_1 to the cryptographic device R times. Let $\mathbf{C}_r$, $1 \leq r \leq R$ be the power consumption curve associated with each distinct scalar computation of $[d]P_1$, because of the randomization performed before each computation, the power consumption can be different between any two computation of $[d]P_1$. Therefore we will consider the mean curve M_{P_1} for computation of $[d]P_1$

$$M_{P_1} = \frac{1}{R} \sum_{r=1}^{R} \mathbf{C}_R$$

If we guess the secret bit d_i incorrectly, then $M_{P_1} \approx 0$, since the values appear in the $(i+1)st$ loop of algorithm 4 are correctly randomized. However, if we guess the secret bit d_i correctly, then the curve M_{P_1} will show considerable consumption (as opposed to the mean function of random points). Once d_i is known we can find the remaining secret bits $d_{i-1}, ..., d_0$ using the above method. One condition for this attack to work is that at least one of the $\left(\sum_{j=i+1}^{l-1} d_j 2^{j-i+1} + 3 \right)$ or $\left(\sum_{j=i+1}^{l-1} d_j 2^{j-i+1} + 1 \right)$ is co-prime to $\#E$. In cryptography, elliptic curves are chosen to have order

$$\#E = h \times q$$

where q is a large prime and h is a small positive integer called the cofactor. For example in both standards: *The Standards for Efficient Cryptography Group* **(SECG)** [sec00] and *National Institute of Standards and Technology* **NIST** [nis] h is chosen from the set $\{1, 2, 4\}$. Hence the conditions for above attack are easily met.

3.19 Special Points on Curves Over Prime Field

$E : y^2 = x^3 + ax + b$ be an elliptic curve over a prime field $\mathbf{F}_p$. A point $(0, y)$ lies on the E if and only if b is a quadratic residue modulo p.

$$y^2 \equiv b \pmod{p}$$

The point $(x, 0)$ lies on the curve if and only if

$$x^3 + ax + b \equiv 0 \pmod{p},\ for\ some\ x \in \mathbf{F}_p$$

the Standards for Efficient Cryptography Group **SECG** [sec00] have curves of order

$$\#E = h \times q$$

Over $\mathbf{F}_p$ all **NIST** curves and most **SECG** curves (except *sec112r2* and *sec128*) have prime orders. It is easy to see that point $(x, 0)$ has order 2 therefore it cannot lie on any curve except *sec112r2* and *sec128*. The point $(0, y)$ lies on most standard curves, for example 8 out of 15 **SECG** curves and 4 out of 5 **NIST** curves contain point $(0, y)$.

3.20 Special Points on Curves over Binary Field

Let $E : y^2 + xy = x^3 + ax^2 + b$ be a non-singular elliptic curve over the binary field $\mathbf{F}_{2^m}$. A special point of form $(0, b^{2^{m-1}})$ always lies on E

$$(b^{2^{m-1}})^2 = b^{2^m} = b$$

The $(x, 0)$ lies on the curve if and only if $\mathbf{F}_{2^m}$

$$x^3 + ax + b = 0 \quad for\ some\ x \in \mathbf{F}_{2^m}$$

It is easy to see that over binary field point $(0, y)$ has order 2.

3.21 Countermeasure Against Goubin's Attack

N. Smart proposed a countermeasure against Goubin's power analysis attack [Sma03]. He pointed out that special points that have small orders can be dealt with by careful implementation of the scalar multiplication algorithm. The countermeasure works as follows. We recall that in standard elliptic curve cryptography

$$\#E(\mathbf{K}) = h \times q$$

where q is a large prime and h is a small integer called *cofactor*. Let d be the private key and h be a cofactor. Note that we will use algorithm 4 together with any one of the three anti-DPA methods (randomized projective coordinates, random elliptic curve isomorphism and random field isomorphism) for scalar multiplication $Q = [d]P$.

1. Compute $Q \leftarrow [h]P$.
2. If $Q \neq P_\infty$, then compute $[d]P$.

This method will thwart Goubin's attack if the special point has a small order because no point of small order will ever enter the scalar multiplication algorithm with scalar d. As mentioned in the previous section, over prime fields $(x, 0)$ has order 2 and over binary fields $(0, y)$ also has order 2. We are left with points $(x, 0)$ on curves over $\mathbf{F}_{2^m}$ and $(0, y)$ on curves over $\mathbf{F}_p$. These points can have large orders. Smart observed that in $\mathbf{F}_{2^m}$ the point $(x, 0)$ of a large order can easily be defended against Goubin's attack if we use the Montgomery method for scalar multiplication [LD99]. For prime fields $\mathbf{F}_p$ Smart proposed a defense against the Goubin's attack using *isogeny.*

3.22 Isogeny Revision

We will quickly review some basic facts about isogeny (for details see chapter 2). Let E_1 and E_2 be two elliptic curves over the finite field $\mathbf{F}_p$ of characteristic $p > 3$. An **isogeny** I is a homomorphism that is given by rational functions. Two elliptic curves are called isogenious if there exists an isogeny between them. E_1 and E_2 are isogenious if and only if $\#E_1 = \#E_2$. The degree of $I(x, y)$ denoted as $deg(I)$ is the number of elements in $ker(I)$.

$$I(x, y) : E_1 \longrightarrow E_2, \quad (x, y) \longmapsto \left(\frac{G(x)}{K(x)^2}, y\left(\frac{f(x)}{K(x)^3}\right)\right)$$

where $K(x)$, $G(x)$, $f(x) \in \mathbf{F}_p[x]$ are polynomials of degrees $2d+1$, d, $3d+1$ and $d = \frac{deg(I)-1}{2}$. Moreover, if $I_l : I(x,y) : E_1 \longrightarrow E_2$ is an isogeny of degree $l^{'}$, then there exists a unique isogeny $I^{-1} : E_2 \longrightarrow E_1$ of degree $l^{'}$ such that for all points $P_1 \in E_1$ and $P_2 \in E_2$ [IBS06].

$$I^{-1}(I(P_1)) = P_1 \quad and \quad I(I^{-1}(P_2) = P_2)$$

3.23 Isogeny Defense Against Goubin's Attack

Let $E : y^2 = x^3 + ax + b$ be an elliptic curve over $\mathbf{F}_p$ such that point $(0, y) \in E$. Let $E^{'}$ be a curve isogenious to E. If $E^{'}$ has no point of form $(0, y)$, then we can map a base point $P \in E$ to point $P^{'} \in E^{'}$ by computing isogeny

$$I_{l'} : E \longrightarrow E^{'}, \quad P \longmapsto P^{'}$$

Instead of computing $Q = [d]P$ on E, we will compute scalar multiplication $Q^{'} = [d]P^{'}$ on $E^{'}$ and then we will map $Q^{'}$ back to Q using

$$I_{l'}^{-1} : E^{'} \longrightarrow E, \quad Q^{'} \longmapsto Q$$

Recall from section 3.19 that out of 15 curves recommended by **SECG** over $\mathbf{F}_p$, 8 of them have the special point $(0, y)$. Smart applied the isogeny defense to these curves and found that all 8 curves can be mapped efficiently to isogenious curves that have no special points (see table 2). In particular he shows **SECG** curves of form

$$E : y^2 = x^3 - 3x + b$$

can be mapped to isogenious curves of form

$$E^{'} : y^2 = x^3 - 3x + b^{'}$$

Curves of this form $(a = -3)$ are efficient for computational purposes [AT05].

We have found that 3 out of 5 **NIST** [nis] curves over $\mathbf{F}_p$ have the special point $(0, y)$. We have applied the isogeny defense to these curves and found that all 3 curves can efficiently be mapped to curves which have no special points (see table 3). Please note that all **NIST** curves over $\mathbf{F}_p$ have prime orders and are of form

$$E : y^2 = x^3 - 3x + b$$

For each curve from the standards we list the *Minimal Isogeny Degree* (the minimum isogeny degree required to map E to $E^{'}$ such that $E^{'}$ does not have a special point) and the *Preferred Minimal Isogeny Degree* (the minimum isogeny degree required to map E to $E^{'}$ such that $E^{'}$ does not have a special point and $E^{'}$ is of above special form). If the original curve does not have a special point, then we specify the isogeny degrees as 1. If the original curve is not of form $y^2 = x^3 - 3x + b$ then we do not compute *Preferred Minimal Isogeny Degree* for it. Note that in some cases the *Preferred Minimal Isogeny Degree* is higher than the *Minimal Isogeny Degree* (secp160r2, secp192r1, secp256r1).

Name of Curve	Minimal Isogeny Degree	Preferred Minimal Isogeny Degree
secp112r1	1	1
secp112r2	11	11
secp128r1	7	7
secp128r2	1	-
secp160k1	1	-
secp160r1	13	13
secp160r2	19	41
secp192k1	1	-
secp192r1	23	73
secp224k1	1	1
secp224r1	1	-
secp256k1	1	-
secp256r1	3	11
secp384r1	19	19
secp521r1	5	5

Table 2: *SECG Curves* over $\mathbf{F}_p$

Hence, all that the smart card needs to do to protect against special points of large order is to store along with the original curve from the standard curves, the isogenious curve and the equations of isogenies I and I^{-1}.

Name of Curve	Minimal Isogeny Degree	Preferred Minimal Isogeny Degree
P-192	23	73
P-224	1	1
P-256	3	11
P-384	19	19
P-521	1	1

Table 3: *NIST Curves* over $\mathbf{F}_p$

3.24 Computational Cost of Isogeny Defense

Let $l^{'}$ be the isogeny degree. Then we need to evaluate polynomials $K(x)$, $G(x)$, $f(x)$ of degrees $2d+1$, d, $3d+1$, where $d = (l^{'} - 1)/2$. If we use Horner's rule, then the total number of field multiplications required for evaluating these polynomials are

$$(2d+1) + (3d+1) + d = 6d + 2 \approx 3l^{'}$$

Looking at the values of $l^{'}$ for standard curves in the tables above we see that in the worst case ($l^{'} = 73$) one need to perform 219 field multiplications. Let's compare this countermeasure against Coron's countermeasures (*randomization of the secret exponent*, *blinding the base point*).

Randomization of the private key. In this method we set $d^{'} = d + n\#E$ for some random integer n and in order to thwart the attack presented in [Os00] require $\log_2 n = \log_2 d$. The additional computational cost of which comes to

$$[(l-1)]\mathbf{A} + [(l-1)]\mathbf{D} + \mathbf{R}$$

where l is the number of bits of d, and $\mathbf{A}$, $\mathbf{D}$, $\mathbf{R}$ denote the cost of the elliptic curve point addition and doubling and random number generartion in finite fields. Over $\mathbf{F}_p$ point addition required 16 field multiplications and point doubling required 11 field multiplications (8 if $a = 3$) (section 3.25). All **NIST** curves and most **SECG** curves (except secp 112r2 and secp 128r2) satisfy $a = -3$. Hence one requires an additional $24 \times (l-1)$ field multiplications for curves satisfying $a = -3$ and $26 \times (l-1)$ field multiplications otherwise. Let $l^{'}$ denote the isogeny degree, then for
$l^{'} < 8 \times (l-1)$, the isogeny method is faster than randomization method.

Blinding the Base Point. The additional computational cost of this countermeasure comes to $39 \times \mathbf{A} + 39 \times \mathbf{D}$ (section 3.13). If we are working over $\mathbf{F}_p$, then this method requires an additional

936 field multiplications if $a = -3$ and 1014 additional field multiplications otherwise. Let $l^{'}$ denote the isogeny degree. The isogeny method is more efficient than this method for $l^{'} < 312$.

Finite Field	Isogeny Defense	Randomization of Secret Exponent	Blinding the Base Point
# of multiplication in $\mathbf{F}_p$	$3l^{'}$	$24(l-1)$	936

Table 4: $l^{'}$ is the isogeny degree and l is the size of secret scalar d

However, one difficulty with Smart's countermeasure is that memory-constraint devices need to store the coefficients of polynomials $K(x)$, $G(x)$, $f(x)$ as well as the isogenious elliptic curve. Hence the best solution will be to completely replace *SECG Curves* and *NIST Curves* with the isogenious that are given by degrees in tables 2 and 3.

3.25 Zero Value Point Attack

For computational reasons, scalar multiplication is done using projective coordinates in standard cryptography (see section 2.12). Before we explain Zero Value Point Attack over prime fields, we will describe an implementation of elliptic curve point doubling and point addition in projective coordinates over prime fields. Let $\mathbf{K}$ be a prime field and $E : ZY^2 = X^3 + AXZ^2 + BZ^3$ be an elliptic curve over projective plane $\mathbf{P^2_K}$ and $P_1 = (X_1, Y_1, Z_1) \in E$, $P_2 = (X_2, Y_2, Z_2) \in E$.

3.26 Prime Field

- *Implementation Of Elliptic Curve Point Double* (**ECDBL**) *Over* $\mathbf{P^2_K}$
- **Input** $(P_1 \neq P_\infty, A)$.
- **Output** $(2P_1)$.

```
1. $T_4 \longleftarrow X_1$ , $T_5 \longleftarrow Y_1$ , $T_6 \longleftarrow Z_1$
2. $T_1 \longleftarrow T_4 \times T_4 :$     $(= X_1^2)$
3. $T_2 \longleftarrow T_5 \times T_5 :$     $(= Y_1^2)$
4. $T_2 \longleftarrow T_2 + T_2 :$     $(= 2Y_1^2)$
5. $T_4 \longleftarrow T_4 \times T_2 :$     $(= 2X_1Y_1^2)$
6. $T_4 \longleftarrow T_4 + T_4 :$     $(= 4X_1Y_1^2 = S)$
```

7. $T_2 \longleftarrow T_2 \times T_2 :\quad (= 4Y_1^4)$

8. $T_2 \longleftarrow T_2 + T_2 :\quad (= 8Y_1^4)$

9. $T_3 \longleftarrow T_6 \times T_6 :\quad (= Z_1^2)$

10. $T_3 \longleftarrow T_3 \times T_3 :\quad (= Z_1^4)$

11. $T_6 \longleftarrow T_5 \times T_6 :\quad (= Y_1 Z_1)$

12. $T_6 \longleftarrow T_6 + T_6 :\quad (= 2Y_1 Z_1)$

13. $T_5 \longleftarrow T_1 + T_1 :\quad (= 2X_1^2)$

14. $T_1 \longleftarrow T_1 + T_5 :\quad (= 3X_1^2)$

15. $T_3 \longleftarrow A \times T_3 :\quad (= AZ_1^4)$

16. $T_1 \longleftarrow T_1 + T_3 :\quad (= 3X_1^2 + AZ_1^4 = M)$

17. $T_3 \longleftarrow T_1 \times T_1 :\quad (= M^2)$

18. $T_3 \longleftarrow T_3 - T_4 :\quad (= M^2 - S)$

19. $T_3 \longleftarrow T_3 - T_4 :\quad (X_3 = M^2 - 2S = T)$

20. $T_4 \longleftarrow T_4 - T_3 :\quad (= S - T)$

21. $T_1 \longleftarrow T_1 \times T_4 :\quad (= M(S - T))$

22. $T_4 \longleftarrow T_1 - T_2 :\quad (= 8Y_1^4 - M(S - T))$

23. $X_3 \longleftarrow T_3$, $Y_3 \longleftarrow T_4$, $Z_3 \longleftarrow T_6$

- *Implementation Of Elliptic Curve Point Addition* **ECADD** *Over* $\mathbf{P}^{\mathbf{2}}_{\mathbf{K}}$

- **Input** $(P_1 \neq P_\infty, P_2 \neq P_\infty)$.

- **Output** (P_3).

1. $T_2 \longleftarrow X_1$, $T_3 \longleftarrow Y_1$, $T_4 \longleftarrow Z_1$

2. $T_5 \longleftarrow X_2$, $T_6 \longleftarrow Y_2$, $T_7 \longleftarrow Z_2$

3. $T_1 \longleftarrow T_7 \times T_7 :\quad (= Z_2^2)$

4. $T_2 \longleftarrow T_2 \times T_1 :\quad (= X_1 Z_2^2 = U_1)$

5. $T_3 \longleftarrow T_3 \times T_7 :\quad (= Y_1 Z_2)$

6. $T_3 \longleftarrow T_3 \times T_1 :\quad (= Y_1 Z_2^3 = S_1)$

7. $T_1 \longleftarrow T_4 \times T_4 : \quad (= Z_1^2)$

8. $T_5 \longleftarrow T_5 \times T_1 : \quad (= X_2 Z_1^2 = U_2)$

9. $T_6 \longleftarrow T_6 \times T_4 : \quad (= Y_2 Z_1)$

10. $T_6 \longleftarrow T_6 \times T_1 : \quad (= Y_2 Z_1^3 = S_2)$

11. $T_5 \longleftarrow T_5 - T_2 : \quad (= U_2 - U_1 = H)$

12. $T_7 \longleftarrow T_4 \times T_7 : \quad (= Z_1 Z_2)$

13. $T_7 \longleftarrow T_5 \times T_7 : \quad (= Z_1 Z_2 H = Z_3)$

14. $T_6 \longleftarrow T_6 - T_3 : \quad (= S_2 - S_1 = R)$

15. $T_1 \longleftarrow T_5 \times T_5 : \quad (= H^2)$

16. $T_4 \longleftarrow T_6 \times T_6 : \quad (= R^2)$

17. $T_2 \longleftarrow T_2 \times T_1 : \quad (= U_1 H^2)$

18. $T_5 \longleftarrow T_5 \times T_1 : \quad (= H^3)$

19. $T_4 \longleftarrow T_4 - T_5 : \quad (= R^2 - H^3)$

20. $T_1 \longleftarrow T_2 + T_2 : \quad (= 2U_1 H^2)$

21. $T_4 \longleftarrow T_4 - T_1 : \quad (= -H^3 - 2U_1 H^2 + R^2 = X_3)$

22. $T_2 \longleftarrow T_2 - T_4 : \quad (= U_1 H^2 - X_3)$

23. $T_6 \longleftarrow T_6 \times T_2 : \quad (= R(U_1 H^2 - X_3))$

24. $T_1 \longleftarrow T_3 \times T_5 : \quad (= S_1 H^3)$

25. $T_1 \longleftarrow T_6 - T_1 : \quad (= S_1 H^3 + R(U_1 H^2 - X_3))$

26. $X_3 \longleftarrow T_4$, $Y_3 \longleftarrow T_1$, $Z_3 \longleftarrow T_7$

In [AT03] Toru Akishita and Tsuyoshi Takagi proposed an attack called **Zero Value Point Attack**, which is a generalization of Goubin's attack [Gou03]. They show that even if elliptic curves have no *special points* $(x, 0)$ and $(0, y)$, they can still have points called *zero value points* (**ZVP**), for which auxiliary register [2] takes zero value and these points cannot be randomized by the countermeasures (randomized projective coordinates, random elliptic curve isomorphism and random field isomorphism). If the **ZVP** points lie on the elliptic curves, then they can be used to launch

[2] In the above implementation of elliptic curve point doubling and point addition in projective coordinates T_i, $1 \leq i \leq 7$, denote auxiliary registers

a DPA attack on elliptic curve scalar multiplication. They showed that standard curves (**SECG** curves) have these **ZVP** points. However, unlike Goubin's attack, the **ZVP** attack depends strongly on the implementation of a scalar multiplication algorithm. For example if $E : y^2 = x^3 + Ax + B$ is an elliptic curve over a prime field $\mathbf{K}$ and (x, y) is a point such that $5x^4 + 2Ax^2 - 4Bx + A^2 = 0$ in $\mathbf{K}$, then (x, y) is a **ZVP** point (equivalently the corresponding point $(r^2x, r^3y, r) \in \mathbf{P}^{\mathbf{2}}_{\mathbf{K}}$ is the **ZVP** point in **ECDBL**. We note that in step 19 of **ECDBL** we computed $(X_3$ by first computing $(M^2 - S)$ (see step 18) and then $T = (M^2 - S)$. However if we compute $2S$ in step 18 and then compute $(M^2 - 2S)$ in step 19, then no auxiliary register in **ECDBL** will take a zero value for this point. In this thesis the **ZVP** attack is launched on implementations described in the previous section.

3.27 Zero Value Points from ECDBL over Prime Fields

Please note that $x(P)$ denotes the *x-coordinate* and $y(P)$ denote the *y-coordinate* of point $P = (x, y)$.

Theorem 4. *Let $E : y^2 = x^3 + Ax + B$ be an elliptic curve over prime field $\mathbf{F}_p$. The elliptic curve E has a zero value point $P = (x, y)$ of* **ECDBL** *if and only if the following conditions are satisfied:*

1. $3x^2 + A = 0.$
2. $5x^4 + 2Ax^2 - 4Bx + A^2 = 0.$
3. $[3]P = P_\infty.$
4. $x(P) = 0$ *or* $x([2]P) = 0.$
5. $y(P) = 0$ *or* $y([2]P) = 0.$

Moreover, the zero-value points are not randomized by these two countermeasures (randomized projective coordinates, random elliptic curve isomorphism).

Proof. Let $P_1 = (X_1, Y_1, Z_1) \neq P_\infty$ be the corresponding point in the projective coordinates. Let $P_3 = \mathbf{ECDBL}(P_1)$. **ECDBL** has a zero value register if and only if one of the following values are zero:

$$X_1,\ Y_1,\ Z_1,\ X_3,\ Y_3,\ M,\ -S + M^2,\ S - T$$

Note that in projective coordinate $Z_1 = 0 \Longrightarrow P = P_\infty$ will never be an input to **ECDBL**.

- $M = 0 = 3X_1^2 + AZ_1^4 = 0 \Longrightarrow 3x^2Z_1^4 + AZ_1^4 = 0 \Longrightarrow 3x^2 + A = 0$ which is condition (1).

  This point cannot be randomized by the above randomization methods. Let $p = (x, y)$ be a point such that $x^2 + A = 0$ then the corresponding randomized point in the projective coordinates is $P = (X, Y, Z) = (\alpha^2 x, \alpha^3 y, \alpha)$, we have $M = 3X^2 + AZ^4 = 0 = \alpha^4(3x^2 + A) = 0$. Similarly Joye-Tymen second counter (section 3.10) will map point $P = (x, y)$ to point $(u^{-2}x, u^{-3}y, 1)$ on some isomorphic curve, we have $M = 3u^{-4}(x^2 + A) = 0$.

- $M^2 - S = 0 = 3X_1^2 + AZ_1^{4^2} - 4X_1Y_2 = 0 \Longleftrightarrow (3x^2 + A)^2 - 4xy = 0$ in affine coordinates , but $(3x^2 + A)^2 - 4xy = 5x^4 + 2Ax^2 - 4Bx + A^2 = 0$, which is condition (2). This point cannot be randomized by the anti-DPA methods of Coron and Joye-Tymen. Let $p = (x, y)$ be a point such that $5x^4 + 2Ax^2 - 4Bx + A^2 = 0$ then the corresponding randomized point in projective coordinates is $P = (X, Y, Z) = (\alpha^2 x, \alpha^3 y, \alpha)$, we have

  $$M^2 - S = 3X^2 + AZ^{4^2} - 4XY^2 = \alpha^8(5x^4 + 2Ax^2 - 4Bx + A^2 = 0)$$

  a similar argument show that Joye-Tymen second counter cannot randomize this point.

- We have $X_3 = T = M^2 - 2S$, $Y_3 = -8Y_1^4 - M(S - T)$ and $Z_3 = 2Y_1Z_1$. If $S - T = 0$ then $X_3 = 4X_1Y_1^2$, $Y_3 = -8Y_1^4$ and $Z_3 = 2Y_1$. Hence $P_3 = ((2Y_1)^2X_1, (2Y_1)^3Y_1, 2Y_1Z_1)$. Using relation 4 we note that $P_3 = (X_1, -Y_1, Z_1)$ for $\alpha = 2Y_1$. But $P_3 = [2]P_1 = P_1^{-1}$ which means $[3]P_1 = P_\infty$. Which is condition (3).

- $X_1 = 0$ or $X_3 = 0$ clearly implies condition (5). Similarly if $Y_1 = 0$ or $Y_3 = 0$ implies condition (4). We have seen in section 3.14 (Goubin's attack) these point cannot be randomized by Coron's or Joye-Tymen countermeasures.

3.27.1 Finding ZVP from ECDBL

In this section we will discuss how to find **ZVP** points from **ECDBL**. Let $E : y^2 = x^3 + ax + b$ be an elliptic curve over $\mathbf{F}_p$. We recall from see section 3.21 that points of small order can be dealt with by careful implementation of scalar multiplication algorithm. Hence, we don't have to worry about condition (3), and condition (4) [3]. To find **ZVP** point from the remaining conditions we have to solve the following polynomials over $\mathbf{F}_p$.

[3] In char> 3 $(x, 0)$ has order 2 and (x, y), such that $x([2](x, y) = 0$ has order 4

For condition (1) we have to solve the polynomial $3x^2 + A = 0$ and for condition (2) we have to solve the polynomial $5x^4 + 2Ax^2 - 4Bx + A^2 = 0$.

For condition (4) we have to solve the polynomial $y^2 - B = 0$.

The solutions for these polynomials over finite fields can be easily computed in polynomial time, for details see [Coh93].

3.28 Zero Value Points from ECADD over Prime Fields

Theorem 5. *Let $E : y^2 = x^3 + Ax + B$ be an elliptic curve over prime field $\mathbf{F}_p$. The elliptic curve E has a zero value point $P = (x, y)$ in $\mathbf{ECADD}(cP, P)$ if and only if one of the following six conditions are satisfied:*

1. *$P = (x, y)$ is a y-coordinate self collision point i.e. $\exists n \in \mathbb{Z}^+$ such that $y([c]P) = y(P)$*
2. *$x(P) + x([c]P) = 0$*
3. *$x([c]P) - x(P) = m^2$, where*

$$m = \begin{cases} \frac{y(P)-y([c]P)}{x(P)-x([c]P)} & \text{if } P \neq [c]P \\ \frac{3x^2+A}{2y} & \text{if } P = [c]P \end{cases}$$

4. *$2x(P) + x[c]P = m^2$*
5. *$x(P) = 0$, $x([c]P) = 0$ or $x([c+1]P) = 0$*
6. *$y(P) = 0$, $y([c]P) = 0$ or $y([c+1]P) = 0$*

Moreover, the zero-value points are not randomized by the following anti-DPA methods (randomized projective coordinates, random elliptic curve isomorphism).

Proof. Let $P_1 = (X_1, Y_1, Z_1)$ and $P_2 = (X_2, Y_2, Z_2)$ be two points such that $P_1 = [c]P_2$ for some $c \in \mathbb{Z}$ and $P_3 = \mathbf{ECADD}(P_1, P_2)$. **ECADD** has a zero value register if and only if one of the following values are zero

$$X_1,\ Y_1,\ Z_1\ , X_2,\ Y_2,\ Z_2,\ X_3,\ Y_3,\ H,\ R,\ R_2 - H^3, U_1H^2 - X_3$$

- Condition (1). $R = 0 = Y_2Z_1^3 - Y_1Z_2^3 = 0 \Longrightarrow y_2 = y_1$ in affine coordinates.
- Condition (2). $H = X_2Z_1^2 - X_1Z_2^2 = 0 \Longrightarrow x_1 = x_2$ in affine coordinates.
- $R^2 - H^3 = \left(Y_2Z_1^3 - Y_1Z_2^3\right)^2 - \left(X_2Z_1^2 - X_1Z_2^2\right)^3 = 0$ which in affine coordinate is $(y_2 - y_1)^2 - (x_2 - x_1)^3 = 0$.

  $$(y_2 - y_1)^2 - (x_2 - x_1)^2 = 0 \Longleftrightarrow \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 = x_2 - x_1 = m^2 = x_2 - x_1$$

  which is condition (3).
- $U_1H^2 - X_3 = U_1H^2 - (-H^3 - 2U_1H^2 + R^2) = 0 \Longleftrightarrow 3U_1H^2 + H^3 + R^2 = 0 \Longrightarrow 3(x_1)(x_2 - x_1)^2 + (x_2 - x_1)^3 - (y_2 - y_1)^2 = 0$ in affine coordinates which implies $2x_1 + x_2 = m^2$. Which is condition (4)
- Condition (5). $X_1 = 0 \Longrightarrow x(P_1) = 0$, $X_2 = 0 \Longrightarrow x(cP_2) = 0$ and $X_3 = 0 \Longrightarrow x([c+1]P) = 0$.
- Condition (6). $Y_1 = 0 \Longrightarrow y(P_1) = 0$, $Y_2 = 0 \Longrightarrow y(cP_1) = 0$ and $Y_3 = 0 \Longrightarrow x([c+1]P_1) = 0$.

3.28.1 Finding ZVP in ECADD

Let $E : y^2 = x^3 + ax + b$ be an elliptic curve over $\mathbf{F}_p$ and $P = (x, y)$ be a point on E. Let c be a positive integer. Then point $[c]P$ can be given as follows [Was03]:

$$[c]P = \left(\frac{\phi_c(P)}{\psi_c^2(P)}, \frac{\omega_c(P)}{\psi_c^3(P)}\right)$$

where polynomials

$$\phi_c = x\psi_c^2 - \psi_{c+1}\psi_{c-1}$$

$$\omega_c = (4y)^{-1}(\psi_{c+2}\psi_{c-1}^2 - \psi_{c-2}\psi_{c+1}^2)$$

and ψ_c, called **division polynomial** is defined recursively as follows

$$\psi_0 = 0$$

$$\psi_1 = 1$$

$$\psi_2 = 2y$$

$$\psi_3 = 3x^4 + 6ax^2 + 12bx - 12bx - a$$

$$\psi_4 = 4y(x^6 + 5ax^4 + 20bx^3 - 5ax^2 - 4abx - 8b^2 - a^3)$$

$$\psi_{2k+1} = \psi_{k+2}\psi_k^3 - \psi_{k-1}\psi_{k+1}^3 \ , \ for \ k \geq 2$$

$$\psi_{2k} = (2y)^{-1}(\psi_k)(\psi_{k+2}\psi_{k-1}^2 - \psi_{k-2}\psi_{k+1}^3) \ , \ for \ k > 2$$

- Condition (1). Let $P = (x, y) \in E$ be a point such that

$$y([c]P) = y \Longleftrightarrow \omega_c(P) = y\psi_c^3(P) \Longleftrightarrow \omega_c(P) + y\psi_c^3(P) = 0$$

- Condition (2). Let $P = (x, y) \in E$ be a point such that

$$x(P) + x([c]P) = 0 \Longleftrightarrow \phi_c(P) + x(p)\psi_c^2(P) = 0$$

- Condition (3). Let $P = (x, y) \in E$ be a point such that

$$x(P) + x([c]P) = m^2 \Longleftrightarrow (x(P)\psi_c^2(P) - \phi_P(P))^3 = (x(P)\psi_c^3(P) - \omega_c(P))^2$$

- Condition (4). Let $P = (x, y) \in E$ be a point such that

$$2x(P)+x[c]P = m^2 \Longleftrightarrow \left[2x(P)\psi_c^2(P) + \phi_c(P)\right]\left[\phi_c(P) - x(p)\psi_c^2(P)\right]^2 = \left[y(P)\psi_c^3(P) + \omega_c(P)\right]^2$$

- Condition (5). Let $P = (x, y) \in E$ be a point such that

$$x([c]P) = 0 \Longleftrightarrow \phi_c(P) = 0$$

- Condition (6). Let $P = (x, y) \in E$ be a point such that

$$y([c]P) = 0 \Longleftrightarrow \omega_c(P) = 0$$

The polynomials ψ_c, ϕ_c and ω_c have degrees of order $O(c^2)$ which increase exponentially in size of c [Was03]. Therefore, it is a hard problem to find a solution for these polynomials for a large c. There is no other efficient way known for computing **ZVP** points from **ECADD** which makes the

ZVP attack from **ECADD** infeasible.

3.29 Isogeny Defense Against ZVP Attack Over F_p

In [Sma03], Smart proposes a defense against Goubin's attack for curves over $\mathbf{F}_p$ using isogeny. Akishita and Takagi in [AT05] examined the isogeny defense against **ZVP** attack over $\mathbf{F}_p$. They pointed out that most **SECG** curves have **ZVP** points from **ECDBL** (see table 5). We have found that all **NIST** curves have **ZVP** from **ECDBL** (see table 7).

Name of Curve	$(0, y)$	$3x^2 + a = 0$	$5x^4 + 2Ax^2 - 4Bx + A^2 = 0$	*Order*
secp112r1	no	yes	yes	*prime*
secp112r2	yes	no	no	$4 \times prime$
secp128r1	yes	no	no	*prime*
secp128r2	yes	no	no	$4 \times prime$
secp160k1	no	no	no	*prime*
secp160r1	yes	no	no	*prime*
secp160r2	yes	no	yes	*prime*
secp192k1	no	no	no	*prime*
secp192r1	yes	yes	yes	*prime*
secp224k1	no	no	no	*prime*
secp224r1	no	no	yes	*prime*
secp256k1	no	no	no	*prime*
secp256r1	yes	no	yes	*prime*
secp384r1	yes	yes	no	*prime*
secp521r1	yes	yes	no	*prime*

Table 5: list of all **SECG** over prime field

They showed that some standard curves require a higher isogeny degree than shown in table 2 (Smart's defense). Moreover, they proved that the class of curves that satisfy $\left(\frac{-3}{p}\right) = -1$ and whose order is odd cannot be mapped by isogeny to curves with $a = -3$ and are secure against the **ZVP** attack (*these curves will always have point* (x, y) *such that* $3x^2 + a = 0$ *if* $a = -3$). They further point out that three **SECG** curves are in this class (secp112r1, secp192r1, secp384r1). As described in section 3.28.1, to find **ZVP** from **ECADD** is a hard problem, so we will only examine the isogeny defense against **ZVP** from (**ECDBL**). We recall that in **ECDBL** the only **ZVP** points we have to worry about are $P = (x, y) \in E$ such that $5x^4 + 2Ax^2 - 4Bx + A^2 = 0$ or $3x^2 + a = 0$ or $x(P) = 0$ (section 3.27.1). In [AT05], they examine *Isogeny Defense Against* **ZVP** *attack.* However, they did not discuss any defense against the **ZVP** attack for curves over $\mathbf{F}_{2^m}$. They only examined **ZVP** points $3x^2 + a = 0$ and $(0, y)$ and did not consider the point $5x^4 + 2Ax^2 - 4Bx + A^2 = 0$. They also did not discuss isogeny defense against all standard curves over prime fields (compare table 5 with table 6). We found that 5 **SECG** curves and 2 **NIST** curves contain **ZVP** points $P = (x, y)$ for which $5x^4 + 2Ax^2 - 4Bx + A^2 = 0$. We also found that if we use an isogeny defense against all three **ZVP** points, then in some curves the isogeny degree increases dramatically. For example, for curve secp224r1, if we apply the isogeny defense against **ZVP** points for which $x(P) = 0$ or $3x^2 + a = 0$ then l_m (minimal isogeny degree to a curve which has neither $(0, y)$ *nor* $3x^2 + a = 0$) is 1 and l_p (minimal isogeny degree to a curve which has neither $(0, y)$ *nor* $3x^2 + a = 0$ *and* $a = -3$) is also 1. But if we apply the isogeny defense against all three **ZVP** points then we find $l_m = 3$ and $l_p = 163$. In this thesis we will examine the isogeny defense against all possible **ZVP** points over $\mathbf{F}_p$ and $\mathbf{F}_{2^m}$. Please note for curves which have odd order, $a = -3$ and $\left(\frac{a}{p}\right) = -1$ we list l_p $\nexists$(does not exist).

Name of Curve	l_m	l_p
secp112r1	7	$\nexists$
secp128r1	7	7
secp160r1	13	13
secp160r2	19	41
secp192r1	23	$\nexists$
secp224r1	1	1
secp256r1	3	23
secp384r1	31	$\nexists$
secp521r1	5	5

Table 6: Isogeny defense for *SECG Curves* over $\mathbf{F}_p$ for point $(0, y)$ and $3x^2 + a = 0$

Name of Curve	$(0, y)$	$3x^2 + a = 0$	$5x^4 + 2Ax^2 - 4Bx + A^2 = 0$	*Order*
P-192	yes	yes	yes	*prime*
P-224	no	no	yes	*prime*
P-256	yes	no	yes	*prime*
P-384	yes	yes	no	*prime*
P-521	no	yes	yes	*prime*

Table 7: list of all **NIST** over $\mathbf{F}_p$

We essentially use the same algorithm as in [AT05] with slight modifications. For each curve from the standards, we search the minimal isogeny degree l_m to a curve which has no point $P = (x, y)$ such that $x = 0$ or $3x^2 + a = 0$ or $5x^4 + 2ax^2 - 4bx + a^2 = 0$. If the original curve has no such point, we specify its degree 1. We also search the preferred minimal isogeny degree l_p to a curve $E^{'}$ for which $a = -3$. As mentioned before, these curves are computationally more efficient. In section 3.24 we saw that the isogeny method is more efficient than Coron's modified countermeasure provided that the isogeny *degree* < 312. Therefore, in algorithm 5 we search for suitable curves for isogeny degrees less than 312.

Algorithm 5.

- **Input** ($E = (a, b)$, j_E, #E, p)

```
1. Set $l_m \longleftarrow 0$ and $l_p \longleftarrow 0$
2. Set $flag \longleftarrow CheckZvp(E,\ a,\ p)$.
3. If $flag = 1$ then
       $l_m \longleftarrow 1$.
       $flag \longleftarrow PreferredCurve(a',\ p)$
4. If $flag = 1$ then
       $l_p \longleftarrow 1$.
       Output ($l_m$, $l_p$).
5. Set $l \longleftarrow 3$.
6. Find the roots of polynomial $G_l(x, j_E)$ and store in list $L$
       If $L = Null$ then go to step 7
         For each root $r \in L$ do
            Set $E'$=($a'$, $b'$)$\longleftarrow$ ConstructCurve($r$, $P$, $\#E$)
             $L \longleftarrow L - \{r\}$
             Set $flag \longleftarrow CheckZvp(E',\ a',\ p)$
             If $flag = 1$, then
               If $l_m = 0$, then
                 $l_m = l$
               Set $flag \longleftarrow PreferredCurve(a',\ p\ )$
               If $flag = 1$ then
                 $l_p \longleftarrow l$
                 Output ($l_m$, $l_p$).
7. If $l > 312$ then stop.
8. $l \longleftarrow NextPrime(l)$ and go to step 6.
```

- The function *CheckZvp*(E, P) will return 1 if E has no zero value point over $\mathbf{F}_p$ and return 0 otherwise.

- The function *ConstructCurve*(r, p, $\#E$) constructs an isogenious curve from the root r.
- The function *PreferredCurve*($a^{'}$, p) return 1 if $a^{'} = -3 \pmod{p}$ or if there exists a curve $E^{''}$ isomorphic to $E^{'}$ such that $a^{''} = -3 \pmod{p}$ and returns 0 otherwise.
- The polynomial $G_l(x, y)$ are called $M\ddot{u}ller's$ *modular polynomials* which were defined in section 2.27.

Tables 8 and 9 show isogeny degrees l_m and l_p for **SECG** curves and **NIST** curves. If the original curve is not of form $y^2 = x^3 - 3x + b$, then we list $l_p = -$. The number in the parenthesis ( ) is the isogeny degree listed in tables 6 and 7, which considers only **ZVP** points $(0, y)$ and $3x^2 + a = 0$. Looking at the entries in tables 6 and 7, we see that out of 15 **SECG** curves 5 require a higher isogeny degree if we consider all three **ZVP** points $(0, y)$, $3x^2+a = 0$ and $5x^4+2Ax^2-4Bx+A^2 = 0$. Further, we can see that for some curves the l_p is considerably high, e.g. secp160r2.

Name of *SECG* Curve over $\mathbf{F}_p$	l_m	l_p
secp112r1	7(7)	∄
secp112r2	13(11)	23(11)
secp128r1	7(7)	181(7)
secp128r2	37(37)	−
secp160k1	1(1)	−
secp160r1	13(13)	13(13)
secp160r2	19(19)	227(41)
secp192k1	1(1)	−
secp192r1	23(23)	∄
secp224k1	1(1)	−
secp224r1	3(1)	163(1)
secp256k1	1(1)	−
secp256r1	3(3)	23(23)
secp384r1	31(31)	∄
secp521r1	5(5)	5(5)

Table 8: Isogeny defense against points $(0, y)$, $3x^2 + a = 0$ and $5x^4 + 2Ax^2 - 4Bx + A^2 = 0$

Name of *NIST* Curve over $\mathbf{F}_p$	l_m	l_p
P-192	23(23)	∄
P-224	3(3)	163(107)
P-256	3(3)	23(23)
P-384	31(31)	∄
P-521	29(29)	∄

Table 9: Isogeny defense against points $(0, y)$, $3x^2 + a = 0$ and $5x^4 + 2Ax^2 - 4Bx + A^2 = 0$

Let E be an elliptic curve over $\mathbf{F}_p$ and d be an *l-bit* private key. Let $l^{'}$ denote the isogeny degree. In section 3.24 we saw that the total number of field multiplications required for an isogeny defense is $3 \times l^{'}$. Furthermore the scalar multiplication $[d]P$ required $24 \times (l-1)$ field multiplications if $a = -3$ and $26 \times (l-1)$ field multiplications if $a \neq -3$. We further recall that the modified Coron's first countermeasure (randomization of the secret exponent and blinding the base point) requires $24 \times (l-1)$ additional field multiplications if $a = -3$ and $26 \times (l-1)$ field multiplications if $a \neq -3$. Coron's second countermeasure (blinding the base point) requires an additional 936 field multiplications if $a = -3$ and 1014 field multiplications if $a \neq -3$. Hence if we obtained the curve through isogeny degree l_m and $l_m \neq l_p$, then one requires additional $3 \times l_m + 2 \times (l-1)$ field multiplication, since elliptic curves obtained through l_m do not satisfy $a = -3$ except when $l_m = l_p$.

Curve	#multi for l_m	#multi for l_p	#multi randomization	#multi blinding
secp112r1	243	N/A	2664	936
secp112r2	261	39	2886	1014
secp128r1	275	543	3048	936
secp128r2	367	N/A	3302	1014
secp160r1	39	39	3816	936
secp160r2	375	681	3816	936
secp192r1	451	N/A	4584	936
secp224r1	455	489	5352	936
secp256r1	519	69	6120	936
secp384r1	857	N/A	9192	936
secp521r1	15	15	12480	936

Table 10: Comparison of computational cost for **SECG** curves

Curve	#multi for l_m	#multi for l_p	#multi randomization	#multi blinding
P-192	451	N/A	4584	936
P-224	455	489	5352	936
P-256	519	69	6120	936
P-384	775	N/A	9192	936
P-521	**1127**	N/A	12480	**936**

Table 11: Comparison of computational cost for **NIST** curves

In tables 10 and 11 we have compared the computational cost of an isogeny defense with the modified Coron's countermeasure against the **ZVP** attack. We can see that for all curves in **SECG** the computational cost for isogeny defense against the **ZVP** attack is less than Coron's modified countermeasure. Interestingly, there is one curve (P-521) in **NIST** standards (see table 11) where

Coron's second countermeasure (blinding the base point) is less costly than isogeny defense. Hence, in order to protect from the **ZVP** attack, all cryptographic devices need to store along with the original curve from the standard the isogenious curve as well as the equation of the isogeny of the isogeny and its inverse. Then input points can be mapped over to the isogenious curve for scalar multiplication and then mapped back to the original curve. This method is described formally on next page. Please note that E_s denotes the standard curve, E_i denotes the isogenious curve, $l^{'}$ denotes the isogeny between them, and d is an *l-bit* private key and $P_s \in E_s$ is the base point.

Algorithm 6. Isogeny Defense Against ZVP Point Attack in $\mathbf{F}_P$.

```
• Input($P_s = (X_s, Y_s,\ d,\ E_s = (a_s, b_s),\ l^{'},\ E_i = (a_i, b_i),\ \#E = h \times q)$

    1. Compute Isogeny $I_{l'}$ between $E_i$ and $E_s$ (see section 2.29).
    2. Compute a corresponding point $(X_i, Y_i) \longleftarrow I_{l'}(X_s, Y_s)$ on $E_i$.
    3. Compute a corresponding point in projective coordinates $P_i^J$

           $P_i^J \longleftarrow (\alpha^2 X_i, \alpha^3 Y_i, \alpha),$   for some random $\alpha \in \mathbf{F}_p^*$

    4. Compute $Q_i^J \longleftarrow$ Always-add-double ($P_i^J$, $a_i$, $h$).
    5. If $Q_i^J \neq P_\infty$ then $Q_i^J \longleftarrow$Always-add-double ($P_i^J$, $a_i$, $d$).
    6. Compute Isogeny $I_{l'}^{-1}$ between $E_s$ and $E_i$ (see section 2.29).
    7. Compute $Q_s^J = (X_s^{'}, Y_s^{'}, Z_s^{'}) \longleftarrow I_{l'}^{-1}(Q_j)$.
    8. Compute corresponding point $Q_s = (X, Y)$ in affine coordinates by setting

           $X \longleftarrow \alpha^{-2} X_s^{'} \quad Y \longleftarrow \alpha^{-3} Y_s^{'}$

• Output $Q_s$.
```

Algorithm 7. *Always-add-double*

```
• Input ($P_1$, $a_1$, $d = \sum_{j=0}^{l-1} d_j 2^j$).

    1. $Q[0] \longleftarrow P$.
    2. for $j$ from $l-2$ to 0 do
```

3. $Q[0] \longleftarrow$ **ECDBL** $(Q[0], a^{'})$.

4. $Q[1] \longleftarrow$ **ECADD** $(Q[0], P)$.

5. $Q[0] \longleftarrow Q[d_j]$.

- **Output** $Q[0]$.

3.30 Zero Value Attack Over Binary Fields

If we use algorithm 4 for scalar multiplication then for over $\mathbf{F}_{2^m}$ the most efficient method of point doubling and point addition was proposed in [LD99]. In their, paper affine coordinates (x, y) were mapped to projective coordinate (X, Y, Z) by setting $x = X/Z$ and $y = Y/Z^2$. The equation of the elliptic curve over these projective coordinates is given by $E : Y^2 + XYZ = X^3Z + A(XZ)^2 + BZ^4$. Let $P_1 = (X_1, Y_1, Z_1)$ and $P_2 = (X_2, Y_2, Z_2)$ be two points on E. The elliptic curve point doubling and addition is done as follows [4].

Inverse and Point of infinity
$-P_1 = (X_1, XZ + Y, Z)$ and $P_\infty = [(\alpha, 0, 0)]$ for any $\alpha \in \mathbf{F}_{2^m}$ and $\alpha \neq 0$

ECDBL-F$_{2^m}$

$$X_2 = X_1^4 + BZ_1^4, \qquad Y_2 = BZ_2Z_1^4 + X_2(AZ_2 + Y_1^2 + BZ_1^4), \qquad Z_2 = X_1^2Z_1^2$$

ECADD-F$_{2^m}$ $\quad X_3 = C^2 + H + G, \qquad Y_3 = HI + Z_3J, \qquad Z_3 = F^2$

where,

$A_0 = Y_2Z_1^2$, $A_1 = Y_1Z_2^2$, $B_0 = X_2Z_1$, $B_1 = X_1Z_2$, $C = A_0 + A_1$, $D = B_0 + B_1$, $E = Z_1Z_2$, $F = DE$, $G = D^2(F + AE^2)$, $H = CF$, $I = D^2B_0E + X_3$, $J = D^2A_0 + X_3$.

In section 3.25 we saw that **ZVP** attack depends strongly on the explicit implementation of scalar multiplication algorithm. In this thesis we assume that **ECDBL-F**$_{2^m}$ and **ECADD-F**$_{2^m}$ is implemented in the following way.

- *Implementation of Elliptic Curve Point Doublimg* (**ECDBL-F**$_{2^m}$) *in* $\mathbf{F}_{2^m}$

- **Input** $(P_1 \neq P_\infty, A, c = B^{2^{m-1}})$

[4] These projective coordinates are slightly different than the one we defined in chapter 2

- **Output** $(2P_1)$

1. $T_1 \longleftarrow X_1$, $T_2 \longleftarrow Y_1$, $T_3 \longleftarrow Z_1$
2. $T_4 \longleftarrow c$
3. $T_3 \longleftarrow T_3 \times T_3 :$ $\quad (= Z_1^2)$
4. $T_4 \longleftarrow T_3 \times T_4 :$ $\quad (= cZ_1^2)$
5. $T_4 \longleftarrow T_4 \times T_4 :$ $\quad (= BZ_1^4)$
6. $T_1 \longleftarrow T_1 \times T_1 :$ $\quad (= X_1^2)$
7. $T_3 \longleftarrow T_1 \times T_3 :$ $\quad (= X_1^2 Z_1^2 = Z_2)$
8. $T_1 \longleftarrow T_1 \times T_1 :$ $\quad (= X_1^4)$
9. $T_1 \longleftarrow T_1 + T_4 :$ $\quad (= X_1^4 + BZ_1^4 = X_2)$
10. $T_2 \longleftarrow T_2 \times T_2 :$ $\quad (= Y_1^2)$
11. If $A \neq 0$

    $T_5 \longleftarrow A :$

    $T_5 \longleftarrow T_3 \times T_5 :$

    $T_2 \longleftarrow T_5 + T_2 :$ $\quad (= AZ_2 + Y_1^2)$
12. $T_2 \longleftarrow T_2 + T_4 :$ $\quad (= AZ_2 + Y_1^2 + BZ_1^4)$ *or* $(= Y_1^2 + BZ_1^4)$
13. $T_2 \longleftarrow T_1 \times T_2 :$ $\quad (= X_2(AZ_2 + Y_1^2 + BZ_1^4))$ *or* $(= X_2(Y_1^2 + BZ_1^4))$
14. $T_4 \longleftarrow T_3 \times T_4 :$ $\quad (= BZ_2 Z_1^4)$
15. $T_2 \longleftarrow T_2 + T_4 :$ $\quad (= BZ_2 Z_1^4 + X_2(Y_1^2 + BZ_1^4) = Y_2)$ *or* $(= BZ_2 Z_1^4 + X_2(AZ_2 + Y_1^2 + BZ_1^4) = Y_2)$
16. $X_2 \longleftarrow T_1$
17. $Y_2 \longleftarrow T_2$
18. $Z_2 \longleftarrow T_3$

- *Implementation of Elliptic Curve Point Addition* (**ECADD-F**$_{2^m}$) *in* $\mathbf{F_{2^m}}$

- **Input** $(P_1 \neq P_\infty,\ P_2 \neq P_\infty,\ A,\ B)$

- **Output** $(P_1 + P_2)$

1. $T_1 \longleftarrow X_1$, $T_2 \longleftarrow Y_1$, $T_3 \longleftarrow Z_1$

2. $T_4 \longleftarrow X_2$, $T_5 \longleftarrow Y_2$, $T_6 \longleftarrow Z_2$

3. $T_7 \longleftarrow T_4 \times T_3 : \quad (= X_2 Z_1 = B_0)$

4. $T_1 \longleftarrow T_6 \times T_1 : \quad (= X_1 Z_2 = B_1)$

5. $T_8 \longleftarrow T_3 \times T_6 : \quad (= Z_1 Z_2 = E)$

6. $T_3 \longleftarrow T_5 \times T_7 : \quad (= Y_2 Z_1^2 = A_0)$

7. $T_6 \longleftarrow T_6 \times T_6 : \quad (= Z_2^2)$

8. $T_6 \longleftarrow T_2 \times T_6 : \quad (= Y_1 Z_2^2 = A_1)$

9. $T_2 \longleftarrow T_3 + T_6 :: \quad (= A_0 + A_1 = Y_2 Z_1^2 + Y_1 Z_2^2 = C)$

10. $T_4 \longleftarrow T_1 + T_7 : \quad (= B_0 + B_1 = X_2 Z_1 + X_1 Z_2 = D)$

11. $T_5 \longleftarrow T_4 \times T_8 : \quad (= D(Z_1 Z_2) = F)$

12. $T_6 \longleftarrow T_5 \times T_5 : \quad (= F^2 = Z_3)$

13. $T_4 \longleftarrow T_4 \times T_4 : \quad (= D^2)$

14. $T_9 \longleftarrow T_8 \times T_8 : \quad (= E^2)$

15. $T_9 \longleftarrow A \times T_9 : \quad (= AE^2)$

16. $T_9 \longleftarrow T_5 + T_9 : \quad (= F + AE^2)$

17. $T_9 \longleftarrow T_4 \times T_9 : \quad (= (D^2)(F + AE^2) = G)$

18. $T_1 \longleftarrow T_2 \times T_2 : \quad (= (Y_2 Z_1)^2 + (Y_1 Z_2)^2 = C^2)$

19. $T_2 \longleftarrow T_2 \times T_5 : \quad (= CF = H)$

20. $T_1 \longleftarrow T_1 + T_2 : \quad (= C^2 + H)$

21. $T_1 \longleftarrow T_1 + T_9 : \quad (C^2 + H + G = X_3)$

22. $T_5 \longleftarrow T_4 \times T_7 : \quad (= D^2 B_0)$

23. $T_5 \longleftarrow T_5 \times T_8 : \quad (= B_0 D^2 E)$

24. $T_5 \longleftarrow T_5 + X_2 : \quad (= B_0 D^2 E + X_3 = I)$

25. $T_8 \longleftarrow T_4 + T_3 : \quad (= A_0 D^2)$

26. $T_8 \longleftarrow T_8 + T_2 : \quad (= A_0 D^2 + X_3 = J)$

27. $T_8 \longleftarrow T_6 \times T_8 : \quad (= Z_3 J)$

28. $T_2 \longleftarrow T_2 \times T_5 : \quad (= HI)$

29. $T_2 \longleftarrow T_2 + T_8 : \quad (= HI + Z_3 J = Y_3)$

ECDBL-F$_{2^m}$ required 5 field multiplications and 5 squarings in general. If $a = 0$ or 1, then it required 4 multiplications and 5 squarings. Please note that in $\mathbf{F}_{2^m}$, the cost of squaring is much lower than cost of field multiplication [IBS06].

ECADD-F$_{2^m}$ required 14 field multiplications and 4 squarings in general and if $a = 0$ or 1 and $Z = 1$, then **ECADD-F**$_{2^m}$ required 9 multiplications and 4 squarings.

3.31 Zero Value Point from ECDBL-F$_{2^m}$

Theorem 6. *Let $E : y^2 + xy = x^3 + Ax^2 + B$ be an elliptic curve over $\mathbf{F}_{2^m}$ such that $B \neq 0$. The elliptic curve E has a zero value point $P = (x, y)$ from* **ECDBL-F**$_{2^m}$ *if and only if the following conditions are satisfied:*

1. $x^2 + y = 0$.

2. $Ax^2 + y^2 = 0$.

3. $y^2 + B = 0$.

4. $y(P) = 0$ *or* $y([2]P) = 0$.

5. $x(P) = 0$ *or* $x([2]P) = 0$.

Moreover, the zero-value points are not randomized by Coron's third countermeasure (randomized projective coordinates).

Proof. Let $P_1 = (X_1, Y_1, Z_1) \neq P_\infty$ be the corresponding point in projective coordinates. Let $P_2 =$**ECDBL-F**$_{2^m}(P_1)$. The algorithm **ECDBL-F**$_{2^m}$ has a zero value register if and only if one of the following values are zero

$$X_1,\ Y_1,\ X_2,\ Y_2,\ Z_2,\ AZ_2 + Y_1^2,\ Y_1^2 + BZ_1^4,\ AZ_2 + BZ_1^4,\ AZ_2 + Y_1^2 + BZ_1^4$$

- $AZ_2 + Y_1^2 + BZ_1^4 = 0 \Longleftrightarrow AX_1^2 Z_1^2 + Y_1^2 + BZ_1^4 = 0$, which is in affine coordinate ($X_1 = xZ_1$ and $Y_1 = yZ_1^2$) is

$$Ax^2 Z_1^4 + y^2 Z_1^4 + BZ_1^4 = 0$$

$$Ax^2 + y^2 + B = 0$$

$$Ax^2 + y^2 + (y^2 + xy + x^3 + Ax^2) = 0$$

$$2Ax^2 + 2y^2 + xy + x^3 = 0$$

$$xy + x^3 = x(x^2 + y) = 0 \Longrightarrow x = 0 \; or \; (x^2 + y) = 0$$

$(x^2 + y) = 0$ is condition (1) and $x = 0$ condition (5) $x(P) = 0$

- $AZ_2 + Y_1^2 = 0 \Longleftrightarrow AX_1^2 Z_1^2 + Y_1^2 \Longrightarrow Ax^2 + y^2 = 0$ in affine coordinate which is condition (2). Similarly $Y_1^2 + BZ_1^4 \Longrightarrow y^2 + B = 0$ in affine coordinate which is condition (3) and
- $X_1 = 0 \Longrightarrow x(P) = 0$ and $X_2 = 0 \Longrightarrow x(2P) = 0$ which is condition (4).
- $Y_1 = 0 \Longrightarrow x(P) = 0$ and $Y_2 = 0 \Longrightarrow x(2P) = 0$ which is condition (5).

These points cannot be randomized by Coron's third countermeasure. Let $P = (x, y)$ such that $x^2 + y = 0$, then the corresponding point in randomized projective coordinates is $(\alpha x, \alpha^2 y, \alpha)$, for some non-zero $\alpha \in \mathbf{F}_{2^m}$. We have $(\alpha x)^2 + \alpha^2 y = \alpha^2(x^2 + y) = 0$. If P satisfies $Ax^2 + y^2 = 0$, then in corresponding projective coordinates we have $A\alpha^2(\alpha x)^2 + (\alpha^2 y)^2 = \alpha^4(Ax^2 + y^2) = 0$. A similar argument shows that other **ZVP** points cannot be randomized by Coron'n third countermeasure.

3.31.1 Finding ZVP in ECDBL-$\mathbf{F}_{2^m}$

In this section we will discuss how to find **ZVP** in **ECDBL-$\mathbf{F}_{2^m}$**. Let $E : y^2 + xy = x^3 + Ax^2 + B$ be an elliptic curve over $\mathbf{F}_{2^m}$.

- Condition(1) $x^2 + y = 0$. Let $P \in E$ be such that $x^2 = y \Longrightarrow x^4 + Ax^2 + B = 0$. The solution of this polynomial can easily be found efficiently [Coh93].
- Condition(2) $Ax^2 + y^2 = 0$. Let $P \in E$ such that $y^2 = Ax^2 \Longrightarrow x^3 + \sqrt{A}x^2 + B = 0$. As mentioned above solving this polynomial is easy. Please note that the equation $y^2 = A$ is trivially solved in $\mathbf{F}_{2^m}$

$$y = A^{2^{m-1}} \Longrightarrow y^2 = A$$

- Condition(3) $y^2 + B = 0$. Therefore point $(0, \sqrt{B})$ will always lie on the curve. But point $(0, \sqrt{y})$ has order 2. We saw in section 3.21, **ZVP** points of small order can easily be dealt with by careful implementation of the scalar multiplication algorithm. However, if point $(x, \sqrt{y})$ for any $x \in \mathbf{F}_{2^m}^*$ lies on the curve, then this point can have a large order. Such a point lies on the curve if and only if $x^2 + ax + \sqrt{B} = 0$ has a solution in $\mathbf{F}_{2^m}$.

- Condition(4) $y(P) = 0$ requires to solve polynomial $x^3 + Ax^2 + B = 0$ which can be easily solved in polynomial time [Coh93]. For condition $y([2]P) = 0$ requires to solve $\psi_3(x^2 + x + y) - (x^2 + xy) = 0$, where ψ_3 is a division polynomial defined in section 3.32.1. Note that if $x^3 + Ax^2 + B$ has no roots, then there can be no point (x, y) on the curve such that $y([2]P) = 0$.

3.32 Zero Value Points from ECADD-$\mathbf{F}_{2^m}$

Theorem 7. *Let $E : y^2 + xy = x^3 + Ax^2 + B$ be an elliptic curve over $\mathbf{F}_{2^m}$. The elliptic curve E has a zero value point $P = (x, y)$ of* **ECADD-F**$_{2^m}(cP, P)$ *if and only if one of the following conditions are satisfied:*

1. $P = (x, y)$ is a y-coordinate self collision point i.e. $\exists c \in \mathbb{Z}^+$ such that $y([c]P) = y(P)$

2. $x(P) + x([c]P) + A = 0$

3. $m = 1$, where

$$m = \begin{cases} \frac{y(P)-y([c]P)}{x(P)-x([c]P)} & \text{if } P \neq [c]P \\ \frac{3x^2+A}{2y} & \text{if } P = [c]P \end{cases}$$

4. $x(P) = 0$ or $x([c]P) = 0$ or $x([c+1]P) = 0$

5. $y(P) = 0$ or $y([c]P) = 0$ or $y([c+1]P) = 0$

6. $x(P) + x([c+1]P) = 0$

7. $y([c]P) + x([c+1]P) = 0$

Proof. Let $P_1 = (X_1, Y_1, Z_1)$ and $P_2 = (X_2, Y_2, Z_2)$ be two points such that $P_1 = [c]P_2$ for some $c \in \mathbb{Z}$ and $P_3 =$**ECADD-F**$_{2^m}(P_1, P_2)$. **ECADD-F**$_{2^m}$ has a zero value register if and only if one of the following values is zero

$$X_1,\ Y_1,\ X_2,\ Y_2,\ X_3,\ Y_3,\ Z_3,\ C,\ D,\ F + AE^2,\ C^2 + H,\ I,\ J$$

- $C = Y_2Z_1^2 + Y_1Z_2^2 = 0 \Longrightarrow y_2 + y_1 = 0$ (in affine coordinates) which is condition (1).
- $F + AE^2 = 0 \iff (X_2Z_1 + X_1Z_2)(Z_1Z_2) + A(Z_1Z_2)^2 = 0 \Longrightarrow x_2(Z_2Z_1)^2 + x_1(Z_1Z_2)^2 + A(Z_1Z_2)^2 = 0$ which implies $x_2 + x_1 + A = 0$ which is condition (2).
- $C^2 + H = 0 \iff C = F \iff y_2(Z_1Z_2)^2 + y_1(Z_1Z_2)^2 = x_2(Z_1Z_2)^2 + x_1(Z_1Z_2)^2 \Longrightarrow y_2 + y_1 = x_2 + x_1 \iff \frac{y_2+y_1}{x_2+x_1} = 1$ which is condition (3).
- $X_1 = 0 \Longrightarrow x(P) = 0$ and if $X_2 = 0 \Longrightarrow x([c]P)$ or $X_3 \Longrightarrow x([c+1]P)$ which is condition (4).
- $Y_1 = 0 \Longrightarrow y(P) = 0$ and if $y_2 = 0 \Longrightarrow y([c]P)$ or $Y_3 \Longrightarrow y([c+1]P)$ which is condition (5).
- $I = B_0D^2E + X_3 = 0 \Longrightarrow x + (x[c+1]P = 0)$ which is condition (6)
- $J = A_0D^2 + X_3 = 0 \Longrightarrow y([c]P) + (x[c+1]P = 0)$ which is condition (7)

3.32.1 Finding ZVP in ECADD-$\mathbf{F}_{2^m}$

Let $E : y^2 + xy = x^3 + Ax^2 + B$ be an elliptic curve over $\mathbf{F}_{2^m}$ and $P = (x, y)$ be a point on E. Let c be a positive integer. Then the point $[c]P$ can be given as follows [Was03]:

$$[c]P = \left(x + \frac{\psi_{c-1}(P)\psi_{c+1}(P)}{\psi_c^2(P)}, x + y + \frac{(x^2 + x + y)\psi_{c-1}(P)\psi_c(P)\psi_{c+1}(P) + \psi_{c-2}(P)\psi_{c+1}^2(P)}{x\psi_c^3(P)} \right)$$

where polynomials ψ_c called **division polynomial** is defined recursively as follows

$$\psi_0 = 0$$

$$\psi_1 = 1$$

$$\psi_2 = x$$

$$\psi_3 = x^4 + x^3 + B$$

$$\psi_4 = x^6 + Bx^2)$$

$$\psi_{2k+1} = \psi_{k+2}\psi_k^3 + \psi_{k-1}\psi_{k+1}^3 \ , \ for\ k \geq 2$$

$$\psi_{2k} = \frac{\left(\psi_{k+2}\psi_{k-1}^2 + \psi_{k-2}\psi_{k+1}^3\right)\psi_k}{x} \ , \ for\ k > 3$$

It is easy to see that all 6 conditions in **ECADD-F$_{2^m}$** require to solve ψ_c, for example let $P = (x, y)$ which satisfies condition (2) i.e.

$$x(P) + x([c]P) + A = 0 \Longleftrightarrow xP + x + \frac{\psi_{c-1}(P)\psi_{c+1}(P)}{\psi_c^2(P)} + A = 0$$

$$= \psi_c^2(P)(A) + \psi_{c-1}(P)\psi_{c+1}(P) = 0$$

The polynomial ψ_c has degrees of order $O(c^2)$ which increase exponentially in size of c. Therefore, it is believed to be a hard problem to find solution of these polynomials for a large c [Was03].

3.33 Defense Against ZVP Attack over $\mathbf{F}_{2^m}$

Preventing **ZVP** point attack or existence of **ZVP** points on standard curves (**SECG, NIST**) over $\mathbf{F}_{2^m}$ was not discussed in [AT05] or [AT03]. We have found that 15 out of 18 **SECG** curves and 7 out of 10 **NIST** have **ZVP** points from **ECDBL-F$_{2^m}$** (see tables 12 and 13). We recall that **ZVP** points of small order can easily be dealt with by using the method described in section 3.21, in parenthesis ( ) we list whether the **ZVP** points are of large or small order. Please also note that we will only discuss defense against **ZVP** points from **ECDBL-F$_{2^m}$**, since finding **ZVP** point from **ECADD-F$_{2^m}$** is believed to be a hard problem.

We notice that Koblitz curves [5] (*in* **SECG** *names ending with* k1 *e.g.* sectk1, *and in* **NIST** *starting with* K *e.g. K-571*) have **ZVP** points of small order only. Hence, these can be protected easily by implementing the scalar multiplication as in algorithm 8. For all other curves we will apply isogeny defense if $a = 1$, and elliptic curve isomorphism, if $a \neq 1$.

Algorithm 8. (Defense against ZVP attack for Koblitz curves over $\mathbf{F}_{2^m}$)

- **Input** $(p = (x, y),\ d,\ E = (a, b),\ \#E = h \times q)$
- Set $P \longleftarrow (rx, r^2, r)$, *for some random non-zero* $r \in \mathbf{F}_{2^m}$.

    1. Compute $Q \longleftarrow BMR\ (P, a, b, h)$.
    2. If $Q \neq P_\infty$ then Compute $BMR\ (P, a, b, A, d)$.

Algorithm 9. *BMR*

- **Input** P, a, b, $d = \sum_{j=0}^{l-1} d_j 2^j$

[5] Koblitz curves refer to binary curves over $\mathbf{F}_{2^m}$ which have a, $b \in \{0, 1\}$

Name of Curve	$x^2 + y = 0$	$Ax^2 + y^2 = 0$	$y^2 + B = 0$	$(x, 0)$	Curve Order
sect113r1	no	no	yes(large)	no	$2 \times Prime$
sect113r2	no	yes(large)	no	yes(large)	$2 \times Prime$
sect131r1	yes(large)	no	yes(large)	yes(large)	$2 \times Prime$
sect131r2	no	no	no	yes(large)	$2 \times Prime$
sect163k1	no	no	no	no	$4 \times Prime$
sect163r1	yes(large)	no	yes(large)	no	$2 \times Prime$
sect163r2	no	yes(large)	no	yes(large)	$2 \times Prime$
sect193r1	yes(large)	no	no	yes(large)	$2 \times Prime$
sect193r2	yes(large)	yes(large)	no	no	$2 \times Prime$
sect233k1	yes(small)	yes(small)	yes(small)	yes(small)	$4 \times Prime$
sect233r1	no	no	no	no	$2 \times Prime$
sect239k1	yes(small)	yes(small)	yes(small)	yes(small)	$4 \times Prime$
sect283k1	yes(small)	yes(small)	yes(small)	yes(small)	$4 \times Prime$
sect283r1	no	yes(large)	no	yes(large)	$2 \times Prime$
sect409k1	yes(small)	yes(small)	yes(small)	yes(small)	$4 \times Prime$
sect409r1	no	no	no	no	$2 \times Prime$
sect571k1	yes(small)	yes(small)	yes(small)	yes(small)	$4 \times Prime$
sect571r1	no	yes(large)	no	yes(large)	$2 \times Prime$

Table 12: **SECG** curves over $\mathbf{F}_{2^m}$ and **ZVP** points from **ECDBL-$\mathbf{F}_{2^m}$**

1. $Q[0] \longleftarrow P$.
2. For j from $l-2$ to 0 do
3. $Q[0] \longleftarrow$ **ECDBL-$\mathbf{F}_{2^m}$** ($Q[0]$, a, $b^{2^{m-1}}$).
4. $Q[1] \longleftarrow$ **ECADD-$\mathbf{F}_{2^m}$** ($Q[0]$, P, a, b).
5. $Q[0] \longleftarrow Q[d_j]$.

- **Output** $Q[0]$.

3.34 Elliptic Curve Isomorphism over $\mathbf{F}_{2^m}$

We will quickly recap few properties of elliptic curves over binary fields, for details see [Men93]. Let $E_1 : y_1^2 + x_1y_1 = x_1^3 + a_1x_1^2 + b_1$ and $E_2 : y_2^2 + x_2y_2 = x_2^3 + a_2x_2^2 + b_2$ be two elliptic curves over $\mathbf{F}_{2^m}$ with $b_i \neq 0$ for $i = 1, 2$. Then E_1 and E_2 are isomorphic over $\mathbf{F}_{2^m}$ if and only if there exists $s \in \mathbf{F}_{2^m}$ such that $a_2 = a_1 + s + s^2$ and $b_2 = b_1$. The j-invariants are given by $j_i = b_i^{-1}$, for $i = 1, 2$. Furthermore, isomorphisms φ and φ^{-1} are given by.

$$\varphi : E_1 \longrightarrow E_2, \quad \varphi\,(x, y) \longmapsto (x, y + sx)$$

Curve	$x^2+y=0$	$Ax^2+y^2=0$	$y^2+B=0$	$(x,0)$	Curve Order
B-163	no	yes(large)	no	yes(large)	$2 \times Prime$
K-163	no	no	no	no	$2 \times Prime$
B-233	no	no	no	no	$2 \times Prime$
K-233	yes(small)	yes(small)	yes(small)	yes(small)	$4 \times Prime$
B-283	no	yes(large)	no	yes(large)	$2 \times Prime$
K-283	yes(small)	yes(small)	yes(small)	yes(small)	$4 \times Prime$
B-409	no	no	no	no	$2 \times Prime$
K-409	yes(small)	yes(small)	yes(small)	yes(small)	$4 \times Prime$
B-571	no	yes(large)	no	yes(large)	$2 \times Prime$
K-571	yes(small)	yes(small)	yes(small)	yes(small)	$4 \times Prime$

Table 13: **NIST** curves over $\mathbf{F}_{2^m}$ and **ZVP** points from **ECDBL-$\mathbf{F}_{2^m}$**

and

$$\varphi^{-1} : E_2 \longrightarrow E_1, \quad \varphi^{-1}(x,y) \longmapsto (x, y+sx)$$

3.35 Defense Against ZVP Attack Through Isomorphism For Binary Curves with $a \neq 1$

In this section we present a countermeasure against **ZVP** attack using elliptic curve isomorphism. In order to thwart the **ZVP** attack we have to choose a curve which has no **ZVP** points of large orders. Our focus will be on non-Koblitz binary curves for which $a \neq 1$. For each curve we pick a random s in $\mathbf{F}^*_{2^m}$ and using isomorpishm φ we compute the corresponding curve $E^{'}$. If $E^{'}$ has no **ZVP** we will return s, otherwise we will pick another random s. It took me on average less than 30 tries to find a suitable curve. Below in the table 14 we list s for each **SECG** curve with $a \neq 1$. Furthermore, we will represent s in hexadecimal for convenience. The conversion from hexadecimal to a field element is done by converting it to binary number and each coefficient of the binary string represents coefficients of s.

Therefore, to protect against the **ZVP** attack, cryptographic devices need to store the original curve, as well as the isomorphic curve and isomorphisms φ and φ^{-1}. Input points can then be mapped to the isomorphic curve for scalar multiplication(using algorithm 8) and then mapped back again to the original curve. The additional computational cost of the isomorphism defense is negligible (2 field multiplications).

Name Of Curve	s (Isomorphic Curves)
sect113r1	18FAA414E74440750490C01BB277D
sect113r2	1D15B022CC73D9E966F8A0ABFA26F
sect131r1	5EF4C6145AA39FFACD6C3296E49AB1246
sect131r2	71C674FDCAE7A4BDE497F58E833EDF9F2
sect163r1	5277F5AB7FFFF42D506904A46AE18086F317DFD86
sect193r1	163E42DF9E7D5373A9C1610E3758E626CC784110B676111AD
sect193r2	1D1A0FD2202E04C7E7EF572304AE231CCBDE817720884068F

Table 14: Isomorphism defense for **SECG** curves over $\mathbf{F}_{2^m}$ with $a \neq 1$

3.36 Isogeny Defense Against ZVP Attack for Binary Curves with $a = 1$

First of all we should stress the fact that if an elliptic curve $E := y^2 + xy = x^3 + ax^2 + b$ over $\mathbf{F}_{2^m}$ satisfies $a = 1$ and $b \neq 0, 1$, then it cannot be mapped to an isomorphic curve $E^{'} : y^2 + xy = x^3 + a^{'}x^2 + b$ that satisfies $a^{'} = 1$. Since by isomorphism we have $a^{'} = a + s + s^2$ for some $s \in \mathbf{F}^*_{2^m}$, if $a = a^{'} = 1$, then $s(1 + s) = 0$, is only possible when $s = 0$. Hence, for these curves we have to apply isogeny defense in order to obtain curves that are safe against **ZVP** attack and satisfy $a = 1$. In table 15 we list the minimal degree of isogeny (l_m) to a curve that is secure against **ZVP** attack.

Curve	l_m
sect163r2	37
sect283r1	23
sect571r1	7
B-163	37
B-283	23
B-571	7

Table 15: Isogeny Defense against **ZVP** points

In table 16 we compare the computational cost of the isogeny defense to Coron's third countermeasure (blinding the base point, see section 3.13). The isogeny defense costs us an additional $3 \times l_m$ field multiplications and Coron's countermeasure costs us $39 \times 4 + 39 \times 9 = 507$ field multiplications. Therefore, for isogeny degree < 169, the isogeny defense is faster than Coron's third countermeasure. Please note that if we use an isomorphism to protect these curves then the additional computational cost would be $5 \times l + 5 \times l$, where l is the number of bits in scalar (see section 3.30). It follows that the isogeny defense requires an additional 1630, 2830 and 5710 field multiplications for curves sect163r2(B-163), sect283r1(B-283), sect571r1(B-571), respectively.

Curve	#multiplication Isogeny defense	#multiplication blinding
sect163r2	111	507
sect283r1	69	507
sect571r1	21	507
B-163	111	507
B-283	69	507
B-571	21	507

Table 16: Comparison of computational cost of isogeny defense with Coron's Countermeasure

In order to protect these curves from **ZVP** attack, cryptographic devices such as Smart cards etc, need to store along with the original curve from the standard, the isogenious curve and the equation of isogeny and its inverse. The input points can then be mapped to the isogenious curve for scalar multiplication and then mapped back to the original curve.

Chapter 4

A Public Key Crytosystem Based on Isogenies

In 1997 Peter Shor proposed an algorithm that can factor a composite number and solve discrete logarithm problem in a cyclic group in polynomial time on a quantum computer [RS06]. Since security of many cryptosystems is based on either factoring a composite number, or solving discrete logarithm in a cyclic group, there have been attempts to come up with cryptosystems whose security is based on problems other than factoring and discrete log. In this spirit Rostovtsev and Stolbunov proposed a new public key cryptosystem whose security is based on computing an isogeny between two given elliptic curves [RS06]. In this chapter we will study this cryptosystem. The chapter is organized in the following way. In section 4.1 we will describe the public key Cryptosystem based on isogenies. In section 4.2 we will describe the cryptosystem security. In section 4.3 we will discusses cryptosystem parameter selection. In section 4.5 we will estimate the computational complexity of the encryption and decryption algorithms. In section 4.6 we will describe the drawbacks of this cryptosystem.

4.1 Cryptosystem

4.1.1 Common Parameters (*Public Information*)

- $D < 0$ such $D \equiv 1 \pmod 4$ or $D \equiv 0 \pmod 4$ such that *class number* h_D is a prime (see section 2.5).

- Choose prime p such that $p = x^2 + |D|y^2$ for $x, y \in \mathbb{Z}$.
- j_{init} a root of Hilber polynomial H_D over $\mathbf{F}_p$ (see section 2.17).
- d is the number of isogeny degrees.
- $L = l_1, ..., l_d$ is the set of Elkies isogeny degrees (see section 2.25).
- $F = \{\pi_1, .., \pi_d\}$ set of Frobenius eigen values, which specify the direction for every isogeny degree l_i (see section 2.25).
- k a limit for the number of steps by any isogeny degree in L.
- **Private Key** is route $R_{priv} = \{r_1, r_2, ..., r_d| - k \leq r_i \leq k\}$
- **Public Key** is a $j_{pub} = R_{priv}(j_{init}, F)$
- **Plaintext** is a set $\mathbf{P} = \{m | m \in \mathbf{F}_p{}^*\}$.
- **Ciphertext** is a set $\mathbf{C} = \{(c_1, c_2) | c_1, c_2 \in \mathbf{F}_p{}^*\}$

4.1.2 Encryption Algorithm

- **Input** (*Common parameters*, j_{pub}, m)

1. Choose a random route R_{enc} subject to constraints above. If $R_{enc} = \{0, 0, ..., 0\}$, then repeat this step.
2. Compute $j_{enc} = R_{enc}(j_{pub}, F)$.
3. Compute $c = m \times j_{enc} \pmod{p}$.
4. Compute $j_{add} = R_{enc}(j_{init})$.
5. **Output** ciphertext (c, j_{add}).

Note that steps 2 and 4 are series of steps themselves for details see section 2.26, 2.25 and 2.27.

4.1.3 Decryption Algorithm

- **Input**(*Common parameters*, R_{priv}, c, j_{add})

1. Compute $j_{enc} = R_{priv}(j_{add})$.

   $R_{priv}(j_{add}) = R_{priv}(R_{enc}(j_{init})) = R_{enc}(R_{priv}(j_{init})) = R_{enc}(j_{pub}) = j_{enc}$

2. Compute $m = \frac{c}{j_{enc}} \mod p$.

4.2 Crytosystem Security

First we note that any route from j_{init} to j_{pub} or from j_{init} to j_{add} will break the cryptosystem 4.1. To see this let R be any route from j_{init} to j_{pub} and (c, j_{add}) be the ciphertext.

$$R(j_{add}) = R(R_{enc}(j_{init})) = R_{enc}(R(j_{init})) = R_{enc}(j_{pub}) = j_{enc}$$

$$m = \frac{c}{j_{enc}} \pmod p$$

The strength of Crytosystem 4.1 is based on the assumption that finding a route between two elliptic curves is hard. This problem is equivalent to the problem of computing an isogeny between two elliptic curves. To see this let $R = \{r_1, ..., r_d\}$ be a route from j_{init} to j_{pub}, then there exists an isogeny E_{ini} to E_{pub} of degree l where $l = l_1^{r_1} \times ... \times l_d^{r_d}$. Similarly if we know the isogeny degree between j_{init} to j_{pub}, then we can easily find a route by dividing the isogeny degree from the elements of $L = \{l_1, ..., l_d\}$.

To find a route from j_{init} to j_{pub} the following techniques can be used.

- **Brute Force**

  Using one isogeny degree, move from j_{init} in one direction until we reach to j_{pub}. The complexity of this attack is $O(n)$ (where $n = h_D$). A similar attack is to generate all possible routes from j_{init} to j_{pub}, according to the restrictions of L, d, k, until we reach j_{init}. The complexity of this attack is also $O(n)$

- **Meet-in-Middle**

  Let $n = h_D$. For a single isogeny degree the average route length between j_{init} and j_{pub} is $O(n)$. For two isogeny degrees the average route length is $O(\sqrt{n})$. For m isogeny degrees the average route length is $R_m \approx O(m \sqrt[m]{n})$. The minimum of function $R_m(m)$ is $O(\ln n)$ when $m \approx O(\ln n)$. For the meet-in-middle-attack, the attacker selects $O(\ln n)$ elkies isogeny degrees (see section 2.5), than in this case the average route length from j_{ini} to j_{pub} does not exceed

R_m. The attacker than constructs all routes from E_{ini}, not longer than $\frac{R_m}{2}$ and stores them in a database. The attacker than selects a random route $R^{'}$ of length no greater than $\frac{R_m}{2}$ and applies to j_{pub} and looks to see if there is a route R in database such that

$$R^{'}(j_{pub}) = R(j_{init})$$

It should succeed with high probability according to the birthday paradox. The complexity of this attack is $O(\sqrt{n})$ isogeny computations.

- **Compute an Isogeny**

  In [Gal99] a probabilistic algorithm to compute an isogeny between two elliptic curves over $\mathbf{F}_p$ has been proposed whose running time in worst case is $O\left(\sqrt{p^3}\ln p\right)$ and in most cases is $O\left(\sqrt[4]{p}\ln p\right)$.

4.3 Parameter Selection

For a fixed discriminant D the corresponding class number asymptotically equals $h_D = O\left(\sqrt{D}\right)$ [Coh93]. In [RS06] it has been suggested that for minimizing computational complexity the number of isogeny degrees should equal to $O\left(\log_2(h_D)\right)$ and the number of k steps should not exceed 2. To provide 2^{80} secrecy we should choose

- prime $p \approx 2^{320}$.
- $D \approx 2^{180}$.
- $h_D \approx 2^{180}$.
- $d \approx 40$

4.4 Prime Class Number

The best known algorithm for computing the class number h_D runs in sub-exponential time, to obtain a prime class number is quite impractical. The requirement that h_D has to be a prime can be replaced by the requirement that h_D is divisible by a large prime. In this case the cryptosystem strength will be estimated at $O(\sqrt{r})$, where r is the largest prime divisor. Below we provide an example with

$D = 239, h_D = 15$. The divisors of h_D are $3, 5, 15$. The roots of H_D(Hilbert polynomial) form three disjoint cycles of size 5 for elkies isogeny degree $l = 3$ and one cycle of size 15 for $l = 5$ see tables 18 and 17. Note 1 is also a divisor of 15, but theorem 3 guarantees that there can be no cycle of size 1 for elkies isogeny degrees.

Example 3.

$D = -239$, $p = 925094306563489092151157097$, $D_\phi = -369557620453916349649648064$,
$h_D = 15 = 5 \times 3$ (Frobenius discriminant), and $l = \{3, 5\}$ (*Set of elkies isogeny degrees*). All 15 roots (*j-invariants*) of the Hilbert polynomials over $\mathbf{F}_p$ are given below

{34664314880184897854066447, 91759613407260720335767156, 48056966310244979865459329, 35785819301790353121799408, 36686821072310455235960695, 45222600282096952558309960, 45544569353491886282107080, 63606223206356386605749452, 86213767687379248106559639, 22650120787929679679909462, 69387360793240473889501349, 12808126089717932373847650, 56341770482404363108802548, 84825386470083372492769962, 6520747052416256803132143 }

Cycle
34664314880184897854066447
36686821072310455235960695
45544569353491886282107080
69387360793240473889501349
56341770482404363108802548
12808126089717932373847650
63606223206356386605749452
6520747052416256803132143
45222600282096952558309960
35785819301790353121799408
86213767687379248106559639
22650120787929679679909462
48056966310244979865459329
91759613407260720335767156
84825386470083372492769962

Table 17: *Single Isogeny cycle of size 15 for degree l=5*

Cycle 1	Cycle 2	Cycle 3
346643148801848978540664447	917596134072607203357671566	452226002820969525583099600
357858193017903531217994088	65207470524162568031321433	455445693534918862821070800
693873607932404738895013499	366868210723104552359606955	226501207879296796799094622
480569663102449798654593299	862137676873792481065596399	128081260897179323738476500
636062232063563866057494522	563417704824043631088025488	848253864700833724927699622

Table 18: *Three Disjoint isogeny cycles of size 5 for degree l=3*

4.4.1 Discriminant and Class Number Selection

Theorem 8 provides an efficient way of obtaining a class number which has a large prime factor.

Theorem 8. *Let $D < 0$ be a fundamental discriminant as describe in section 2.3 and $D_\pi = Df^2$ be an order in $Q(\sqrt{D})$, then*

$$\frac{h_{D_\pi}}{w_{D_\pi}} = \frac{h_D}{w_D} f \prod_{q|f} \left(1 - \frac{\left(\frac{D}{q}\right)}{q} \right)$$

where the product is over all prime divisors q of f and $\left(\frac{D}{q}\right)$ is kronecker-jacobi symbol and w_D is the number of invertible elements in the quadratic order O_D

$$w_D = \begin{cases} 2 & \textit{if } D < -4 \\ 4 & \textit{if } D = -4 \\ 6 & \textit{if } D = -3 \end{cases}$$

Proof. See [Coh93]

Suppose h_D is small, (e.g. $D = -3\ h_D = 1$). Let p be a prime in $Q(\sqrt{D})$ such that $p = x^2 + |D|y^2$ and y is prime. Let D_ϕ be the discriminant of frobenius characteristic equation (see section 2.20), then O_{D_ϕ} is an order with discriminant $D_\phi = D(2y)^2$. From theorem 8 it follows that

$$h_{D_\phi} \geq \begin{cases} 2yh_D & \text{if } D < -4 \\ yh_D & \text{if } D = -4 \\ \frac{2}{3}yh_D & \text{if } D = -3 \end{cases}$$

I have found that in practice to find a prime of form $p = x^2 + |D|y^2$ such that y is also a large prime takes less than a minute (see table 19 and 20). Another way to obtain a class number with a large prime divisor is to choose D such that $-D$ is a large prime, then set $f = -D$, from theorem 8 we have

$$h_{D_\pi} = f \times h_D$$

D	h_D	Average Time	bits x	bits y
-3	1	31 seconds	≈ 16	≈ 160
-7	1	22.5 seconds	≈ 16	≈ 160
-8	1	40.7 seconds	≈ 16	≈ 160
-11	1	36.8 seconds	≈ 16	≈ 160
-19	1	39.2 seconds	≈ 16	≈ 160
-43	1	23.91 seconds	≈ 16	≈ 160
-67	1	28.35 seconds	≈ 16	≈ 160
-163	1	9.50 seconds	≈ 16	≈ 160

Table 19: *Computation of a random prime with y is also prime for all D with $h_D = 1$*

D	h_D	Average Time	bits x	bits y
-15	2	28.33 seconds	≈ 16	≈ 160
-20	2	31.20 seconds	≈ 16	≈ 160
-24	2	27.06 seconds	≈ 16	≈ 160
-35	2	17.12 seconds	≈ 16	≈ 160
-40	2	29.70 seconds	≈ 16	≈ 160
-51	2	23.21 seconds	≈ 16	≈ 160
-52	2	33.99 seconds	≈ 16	≈ 160
-88	2	30.78 seconds	≈ 16	≈ 160
-91	2	36.03 seconds	≈ 16	≈ 160
-115	2	45.38 seconds	≈ 16	≈ 160
-123	2	21.40 seconds	≈ 16	≈ 160
-148	2	21.50 seconds	≈ 16	≈ 160
-187	2	29.35 seconds	≈ 16	≈ 160
-232	2	24.98 seconds	≈ 16	≈ 160
-235	2	32.0 seconds	≈ 16	≈ 160
-267	2	24.83 seconds	≈ 16	≈ 160
-403	2	23.43 seconds	≈ 16	≈ 160
-427	2	29.17 seconds	≈ 16	≈ 160

Table 20: *Computation of a random prime with y is also prime for all D with $h_D = 2$*

4.5 Running Time of Crytosystem 4.1

We assume that *Müller's modular polynomials* G_l are precomputed and stored. The most costly steps in the encryption algorithm are steps 2 and 4 which require factoring the polynomial G_l. The overall complexity of factoring G_l over $\mathbf{F}_p$ is $O\left(l^{1.815}\log p\right)$ [KS98]. Let $R_{enc} = \{r_1, r_2, ..., r_d| -k \leq r_i \leq k\}$ be a route, then the computational complexity of encryption algorithms is

$$2k \times \sum_{i=1}^{d} O\left(l_i^{1.815}\log p\right) = O\left(k \times \sum_{i=1}^{d}(l_i^{1.815}\log p\right)$$

Let $l = max\{l_1, l_2, ..., l_d\}$, then substituting l for each l_i in the above equation we get.

$$O\left(k \times d \times l^{1.815}\log p\right)$$

If we take $k = 2$ and $d = O\left(\log h_D\right)$ as suggested in [RS06], then the computational complexity of encryption algorithms is bounded by

$$O\left(l^{1.815}\log h_D \log p\right) = O\left(l^{1.815}(\log p)^2\right)$$

Note that $d = O\left(\log h_D\right) \Rightarrow d = O\left(\log\sqrt{p}\right)$. Since $d = O\left(\log h_D\right) = O\left(\log\sqrt{D}\right)$ [Coh93] and $p = x^2 + |D|y^2 \Rightarrow |D| \leq p$ therefore, we can write $d = O\left(\log\sqrt{p}\right)$.

This is also the complexity of decryption algorithm. In [RS06] it was also never mentioned explicitly that *Müller′s modular polynomials* should be precomputed or should be computed during running time. Storing these polynomials in memory-constrained devices can be a problem. For example for $l = 197$ G_l has 4754 coefficients with the largest coefficient of 516 digits. The overall complexity of computing G_l over $\mathbf{F}_p$ is $O\left(l^{3+\epsilon}\log p \log\log p\right)$ [IFBS99]. It can take several minutes for computing G_l for $l \approx 200$. Hence, computing these polynomial during running time is also very costly.

4.6 DrawBacks of Isogeny Based Cryptosystem

In this section, we will describe the drawbacks of this Cryptosystem. The security of all public key cryptosystems are based on certain assumptions, for example the security of **RSA** public key cryptosystem based on the assumption that factorization of a large composite number into its prime factors is computationally infeasible and its function $f_d(m) = m^e \mod N$ is one way. In Isogeny based cryptosystem we not only assume that the isogeny problem is hard and its encryption is one way, but in addition to this we make an additional assumption. Recall from section 2.25 we have stated that

If the class number h_D is prime, then the roots of Hilbert polynomial will form a single isogeny cycle and if class number is not a prime, then the roots of the Hilbert polynomial will form disjoint cycles of divisors of h_D.

This statement is merely conjectured by the authors of [RS06] based on computational experiments, there is no proof known to this date. If this conjecture is false, the cryptosystem is vulnerable to attacks whenever the elements of U_D form small disjoint cycles. The second problem which is more crucial than the first one is that there is no known efficient way of computing the j_{init} (a root of Hilbert polynomial $H_D(X)$ over $\mathbf{F}_p$).

$$H_D(X) = \prod_{i=1}^{h_D} (X - j(\tau_i)) \in \mathbb{Z}[\mathrm{X}]$$

where $\tau_i = \frac{-b_i + \sqrt{D}}{2a_i}$ and $Cl_D = \{(a_i, b_i, c_i) | 1 \leq i \leq h_D\}$ is the set of reduced triplets described in section 2.5 and the function $j(\tau)$ is defined in section 2.15. The Hilbert polynomials are first computed over integers and then reduced modulo a suitable prime. These polynomials have two drawbacks. First they have coefficients of astronomical size even for small discriminant e.g.

$$H_{-71}(X) = x^7 + 313645809715x^6 - 3091990138604570x^5 + 98394038810047812049302x^4 -$$

$$823534263439730779968091389x^3 + 5138800366453976780323726329446x^2 -$$

$$425319473946139603274605151187659x + 737707086760731113357714241006081263$$

Secondly, very high precision is required to compute the Hilbert polynomials [IBS06]. The precision needed will be round

$$P_H = 10 + \binom{h_D}{\lfloor h_D/2 \rfloor} \frac{\pi\sqrt{D}}{\log 10} \sum_{(a_i, b_i, c_i) \in Cl_D} \frac{1}{a_i}$$

where $\binom{h_D}{\lfloor h_D/2 \rfloor}$ are possible combinations. For $h_D = 100$ the precision required to compute $H_D(X)$ is greater than 729826235521340289034212447957 (a 30 digit number), for $h_D = 10000$ the precision is greater than 5×10^{3009} (a 3000 digit number), for $h_D > 10^5$, calculation of a Hilbert polynomial is practically infeasible. Therefore, we cannot compute j_{init} using Hilbert polynomials. There is an alternate polynomial called the Weber polynomial W_D ([YZ97]) which has much smaller coefficients and require much smaller precision, from the roots of Weber polynomials W_D we can easily compute the roots of Hilbert polynomial $H_D(X)$. In the next section we will describe the Weber polynomials and we will see that even Weber polynomials have their limitations.

4.7 Weber Polynomials

Let $\tau \in H$ (upper half-plane) and $q = e^{2\pi i \tau}$. The classical Weber functions are defined by

$$f(\tau) = q^{-\frac{1}{48}} \prod_{i=1}^{\infty} (1 + q^{n-\frac{1}{2}})$$

$$f_1(\tau) = q^{-\frac{1}{48}} \prod_{i=1}^{\infty} (1 - q^{n-\frac{1}{2}})$$

$$f_2(\tau) = \sqrt{2} q^{-\frac{1}{24}} \prod_{i=1}^{\infty} (1 + q^{n-\frac{1}{2}})$$

Let D be a negative integer such that $D \equiv 1 \pmod 4$ and $D \not\equiv 0 \pmod 3$, for other discriminant see [IBS06]. Let h_D be the corresponding class number and $Cl_D = \{[a_i, b_i, c_i] | 1 \leq i \leq h_D\}$ be the set of equivalence classes of ideals of O_D as defined in (section 2.7). Set $\tau_i = \frac{b_i + \sqrt{D}}{2a_i}, \textit{ for }$ each $[a_i, b_i, c_i] \in Cl_D$.

$$g([a_i, b_i, c_i]) = \begin{cases} \zeta^{b(a-c-c^2)} \times f(\tau) & \text{if } 2|a_i, 2|c_i \\ (-1)^{\frac{D-1}{8}} \times \zeta^{b(a-c-c^2)} \times f(\tau) & \text{if } 2|a_i, 2 \nmid c_i \\ (-1)^{\frac{D-1}{8}} \times \zeta^{b(a-c-a^2c)} \times f(\tau) & \text{if } 2 \nmid a_i, 2|c_i \end{cases}$$

where $\zeta = \exp 2\pi i/48$. The Weber polynomial $W_D(X)$ is

$$W_D(X) = \prod_{\tau_i} (X - g(\tau_i)) \in \mathbb{Z}[\mathrm{X}]$$

Let j_w be the root of $W_D(X)$ over $\mathbf{F}_p$, then the root (j_h) of Hilbert polynomial $H_D(X)$ over $\mathbf{F}_p$ is given by

$$j_h = (j_w^{24} - 16)^3 / j_w^{24}$$

The Weber polynomials have much smaller coefficients, e.g.

$$W_{-71}(X) = x^7 - x^6 - x^5 + x^4 - x^3 - x^2 + 2x + 1$$

and the precision required to compute these polynomials is

$$P_W = \frac{G + \frac{h_D}{4} + 5}{45} + 1$$

Where,

$$G = \frac{\pi\sqrt{|D|}}{\log 10} \times \sum_{(a_i,b_i,c_i)\in Cl_D} \frac{1}{a_i}$$

Table 21 shows that the Weber polynomials are much more computationally friendly, then Hilbert polynomials. Recall that in section 4.3 we have shown that in order to provide 2^{80} we should choose $D \approx 2^{320}$ and $h_D \approx 2^{180}$. The precision required to compute Weber polynomials for discriminant of this size is $> 5.5 \times 10^{53}$. The total running time it took to compute Weber polynomial with $D = -10000031$, $h_D = 5426$ and $P_W = 1986$ was around 1.5 days. These computations carried out on Solaris 2.6 at 333 MHz and having 512 MB of main memory[BB01]. Hence to compute a Weber polynomials for $D \approx 2^{320}$ and $h_D \approx 2^{180}$ is also impractical.

D	h_D	P_W	P_H	Digits in P_H
-7991	100	24	1019002579910198352681606222228570	33
-21311	200	48	$\approx 1.9 \times 10^{62}$	63
-412079	1000	294	$\approx 3.70 \times 10^{302}$	303
-10000031	5426	1986	$\approx 2.50 \times 10^{1636}$	1637

Table 21: *Precision comparison for Weber and Hilbert polynomials*

Theorem 9.

- Every element in $\mathbf{F}_p$ is the j-invariant of an elliptic curves over $\mathbf{F}_p$.
- If $|D| > 4$, then all elliptic curves with the given j-invariants, $j \neq 0, 1728$ over $\mathbf{F}_p$ are given by

  $$y^2 = x^3 + 3kc^2x + 2kc^3$$

  where $k = j/(1728 - j)$ and c is any element in $\mathbf{F}_p$.
- Suppose E and $E^{'}$ have the same j-invariant but are not isomorphic over the field $\mathbf{F}_p$. If $j \neq 0, 1728$, then $E^{'}$ is the quadratic twist if $\#E = p + 1 - t$ then $\#E^{'} = p + 1 + t$.

- Assume that $j \neq 0, 1728$. If E is given by

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

  then $E^{'}$ is given by

  $$y^2 = x^3 + ac^2x + bc^3$$

  where c is any quadratic non-residue in $\mathbf{F}_p$

Proof. (See [IBS06])

Theorem 10. *The number of isomorphism classes of elliptic curves over finite field $\mathbf{F}_p$, $p > 3$ is $2p + 6$, $2p + 2$, $2p + 4$, $2p$, for $p \equiv 1, 5, 7, 11 \pmod{12}$ respectively.*

Proof. (See [Men93])

4.8 Finding a Root (j_{init}) of Hilbert Polynomials over $\mathbf{F}_p$

Let $D < 0$ be any non-square integer such that $D \equiv 0$ or $1 \pmod 4$ and suppose there exists a prime p for which the diophantine equation

$$p = x^2 + |D|y^2$$

can be solved. Recall from section 2.17 that the roots of Hilbert polynomial $H_D(X)$ over $\mathbf{F}_p$ will give *j-invariants* of elliptic curves. Each *j-invariant* will correspond to two elliptic curves E and $E^{'}$ one has order $p + 1 - x$ and the other has order $p + 1 + x$. The equations of E and $E^{'}$ can easily be obtained from the *j-invariant* using theorem 9. Similarly, if an elliptic curve E over $\mathbf{F}_p$ has order $p + 1 - x$ or $p + 1 + x$ then the *j-invariant* of E is a root of Hilbert polynomial. According to [Coh93], over $\mathbf{F}_p$ the number of isomorphic classes of elliptic curves whose orders are $p + 1 + 2x$ or $p + 1 + 2x$ asymptotically equals to $O(\sqrt{D})$. Therefore, if we choose D close to p, then we can randomly choose an elliptic curve E over $\mathbf{F}_p$ and check if $\#E = p + 1 + 2x$ or $\#E = p + 1 + 2x$. The fastest known deterministic algorithm known to count the number of points on an elliptic curve over a finite field is the **Schoof-Elkies-Atkin algorithm** (SEA). The running time complexity of this algorithm is $O((\log p)^{4+\epsilon})$[Sat02]. Hence to find an initial elliptic curve using the above method is also computationally infeasible, however a more efficient way to find an initial elliptic curve is to use a combination of (SEA) and scalar multiplication algorithm 3 (see section 3.3) and theorem 9. We will approach as follows, we will first choose a curve E over F_p using theorem 9 and then will choose a random point Q on E and check the condition $[p+1-2x]Q = P_\infty$ or $[p+1+2x]Q = P_\infty$, if neither of the conditions hold, then $p+1+2x$ or $p+1-2x$ cannot be the group order. If one of the condition hold, then there is a high likelihood that this is the group order. The probability can be increased by drawing and checking further random points. I have find that in practice if we found a point P on E whose order is either $[p+1+2x]$ or $[p+1-2x]$ then it is almost always the case that the order E is the order of P. The method is formally describe below.

Algorithm 10. Find Initial Curve.

- **Input** (p, x).

```
1. For $i$ from 1 to $Max$ do
2. Choose $j$ randomly in $\mathbf{F}_p - \{0, 1728\}$.
3. $A = j/(1728-j) \mod p$ and $B = j/(1728-j) \mod p$ (Theorem 9).
4. Randomly choose a point $Q \in y^2 = x^3 + Ax + B$.
5. If $[p+1-2x]Q = P_\infty$ or $[p+1-2x]Q = P_\infty$ then
6.   If $SEA(A, B, p)$ or $SEA(A, B, p)$ then
7.       Output $j$.
```

Clearly, this is a much more efficient way of finding a root, since we will invoke the SEA algorithm only if $p+1-2x(P) = P_\infty$ or $p+1+2x(P) = P_\infty$. Once we have found such a curve, we can use SEA algorithm to compute the number of points on E and if the number of points on E is either $p+1-2x$ or $p+1+2x$, then we are done otherwise we can choose another random j-invariant and repeat the process. How many times are we are expected to try before we can find the right curve. Let N_E denote the number of isomorphism classes of elliptic curves over $\mathbf{F}_p$. The number of times we expect to try, is

$$\frac{O(\sqrt{D})}{N_E}$$

Curves, before a suitable curve can be found. According to theorem 10 $N_E = O(p)$. Let $D \approx p$, the probability that we choose a random curve $E(\mathbf{F}_p)$ such that $\#E = p+1-2x$ or $p+1+2x$ is

$$\frac{O(\sqrt{p})}{O(p)} \approx \frac{1}{O(\sqrt{p})}$$

Therefore, we are expected to try $\sqrt{p}$ times before we can find an elliptic curve of a desired order. Recall from chapter 3 that a prime p will provide $\sqrt[4]{p}$ secrecy. In order to provide 2^{80} secrecy we are expected to try 2^{160} times before we can find a root of Hilbert polynomial over $\mathbf{F}_p$. This is a huge number and it is quite impractical to find a root like this. I have implemented algorithm 10 in Maple 9 and found that it takes on average 0.5 seconds for algorithm 10. Hence it will take approximately 0.5×2^{160} seconds ($\approx 10^{41}$ years) to find a root of Hilbert polynomial over a prime field of size 2^{320}. Until we can find an efficient way of computing a root of a Hilbert polynomial over prime fields, this cryptosystem cannot be used in practice.

Chapter 5

Conclusion

In this thesis we studied and analysed the application of isogenies and elliptic curve isomorphisms for defense against various power analysis attacks. Our focus was on elliptic curve cryptosystems. We saw that these attacks can easily be thwarted by the help of isogenies and elliptic curve isomorphisms (for curves over $\mathbf{F}_{2^n}$). These side channel attacks also point to the fact that traditional assumptions in cryptography need to be reevaluated. Traditionally the designer of a cryptosystem assumes that an adversary knows everything about the cryptosystem being used except the key and he/she has pairs of plaintext/ciphertext. This is referred to as **Kerchoff's principle**. However in practice more information is often available to the adversary. For example in chapter 3 we saw that cryptographic devices leak information about private key through side channels (power consumption etc). Therefore, it is important that the cryptosystem should be designed with the assumption that unintended information is leaked by these devices. Although it's worth noting that researchers have developed hardware that leak significantly less information, so far no feasible alternatives to transistors are available. However, alternate computation technologies such as pure optical computing [1] may exist in the future [PKJ98].

In 1997 Peter Shor proposed an algorithm that can factor a composite number and solve discrete logarithm problem in a cyclic group in polynomial time on a quantum computer [RS06]. Since security of many cryptosystems is based on either factoring a composite number, or solving discrete logarithm in a cyclic group, there have been attempts to come up with cryptosystems whose security is based on problems other than factoring and discrete log. In this spirit Rostovtsev and Stolbunov

[1] An optical computer is a computer that uses light instead of electricity to manipulate, store and transmit data

proposed a new public key cryptosystem whose security is based on computing an isogeny between two given elliptic curves [RS06]. In this thesis we also studied this cryptosystem. We found that in order to use this cryptosystem in practice one has to compute a root of the Hilbert polynomial H_D over $\mathbf{F}_p$, which becomes infeasible when D is large.

Bibliography

[ABaS96] B. Salvy A. Bostan, F. Morain and É. Schost. Fast algorithms for computing isogenies between elliptic curves. 1996.

[AT03] Toru Akishita and Tsuyoshi Takagi. Zero-value point attacks on elliptic curve cryptosystem. In *ISC 2003, vol 2851, Lecture Notes in Computer Science (LNCS)*, pages 218–233. Springer-Verlag, 2003.

[AT05] Toru Akishita and Tsuyoshi Takagi. On the optimal parameter choice for elliptic curve cryptosystems using isogeny. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E88-A:140–146, 2005.

[BB01] Harald Baier and Johannes Buchmann. Efficient construction of cryptographically strong elliptic curves. Technical report, Darmstadt University of Technology, 2001.

[Bucon] Johannes Buchmann. *Algorithms for binary quadratic forms.* Springer-Verlag, in preparation.

[CJM] Dewaghe L. Couveiges J.M and F. Morain. Isogeny cycles and the schoof-elkies-atkin algorithm. http://citeseer.ist.psu.edu/couveignes96isogeny.html.

[Coh93] Henri Cohen. *A Course in Computational Algebraic Number Theory.* Springer-Verlag, 1993.

[Cor99] Jean-Sébastien Coron. Resistance against differential power analysis for elliptic curve cryptosystems. In *Cryptographic Hardware and Embedded Systems-CHESS(99)*, volume 1717, pages 292–302. Springer-Verlag, 1999.

[EKZ03] Yannis C. Stamatiou Elisavet Konstantinou and Christos Zaroliagis. On the construction of prime order elliptic curves. In *Progress in Cryptology - INDOCRYPT 2003*, volume 2904/2003, pages 309–322. Springer Berlin / Heidelberg, 2003.

[Gal99] Steven Galbraith. Constructing isogenies between elliptic curves over finite fields. *Journal of Computational Mathematics*, 2:118–138, 1999.

[Gou03] Louis Goubin. A refined power-analysis attack on elliptic curve cryptosystems. In *Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography: Public Key Cryptography*, volume Lecture Notes in Computer Science 2567, pages 199–210. Springer-Verlag, 2003.

[Hun96] Thomas W. Hungerford. *Abstract Algebra: An Introduction.* Thomson Brooks/Cole, 1996.

[IBS06] Gadiel Seroussi Ian Blake and Nigel Smart. *Elliptic Curves in Cryptography.* London Mathematical Society Lecture Note Series. Cambridge University press, 2006.

[IFBS99] Michael Rubinstein Ian F. Blake, $J\acute{A}nos$ A. Csirik and Gadiel Seroussi. On the computation of modular polynomials for elliptic curves. Technical report, University of Texas at Austin, 1999.

[JT01] Marc Joye and Christophe Tymen. Protections against differential analysis for elliptic curve cryptography. In *Cryptographic Hardware and Embedded Systems-CHESS (2001)*, volume Lecture Notes in Computer Science 2162, pages 377–390. Springer-Verlag, 2001.

[Koc96] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Advances in Cryptology*, volume 1109, pages 104–113. Springer-Verlag, 1996.

[Koh96] David R. Kohel. *Endomorphism rings of elliptic curves over finite fields.* PhD thesis, University of California, Berkeley, 1996.

[KS98] Erich Kaltofen and Victor Shoup. Subquadratic-time factoring of polynomials over finite fields. *Math. Comp*, 67:398–406, 1998.

[LD99] J. Lopez and R. Dahab. Fast multiplication on elliptic curves over $gf(2^m)$ without precomputation. In *Cryptographic Hardware and Embedded Systems-CHESS(99)*, volume 1717, pages 292–302. Springer-Verlag, 1999.

[Men93] Alfred J. Menezes. *Elliptic Curve Public Key Cryptosystems.* Springer, 1993.

[nis] Recommended Elliptic Curves For Federal Goverment Use. http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf.

[Os00] Katsuyuki Okeya and Kouichi sakurai. Power analysis breaks elliptic curve cryptosystems even secure against the timing attack. In *Progress in Cryptology*, volume 1977, pages 178–190. Springer-Verlag, 2000.

[PKJ98] Joshua Jaffe Paul Kocher and Benjamin Jun. Introduction to differential power analysis and related attacks. Technical report, Cryptography Research, 1998.

[RS06] Alexander Rostovtsev and Anton Stolbunov. Public key cryptosystems based on isogenies. 2006.

[Sat02] Takakazu Satoh. On p-adic point counting algorithms for elliptic curves over finite fields. In *Algorithmic Number Theory*, volume 3076, pages 44–66. Springer Berlin / Heidelberg, 2002.

[sec00] The standards for efficient cryptography group (secg), 2000. http://www.secg.org/.

[Sma03] Nigel P. Smart. An analysis of goubin's refined power analysis attack. In *Cryptographic Hardware and Embedded Systems-CHESS (2003)*, volume Lecture Notes in Computer Science 2779, pages 281–290. Springer Berlin / Heidelberg, 2003.

[Tat66] John Tate. Endomorphisms of abelian varieties over finite fields. *Inventiones Mathematicae*, 2:134–144, 1966.

[Was03] Lawrence C. Washington. *Elliptic curves Number theory and Cryptography.* CHAPMAN AND HALL, 2003.

[Wil74] Malcolm J. Williamson. Non-secret encryption using a finite field. 1974.

[YZ97] Noriko Yui and Don Zagier. On the singular values of weber modular functions. *Mathematics of Computation*, 66:1645–1662, 1997.