UiO : Department of Mathematics
University of Oslo

On the Generation of Strong Elliptic Curves

For Cryptographic Applications

Ståle Zerener Haugnæss

Master's Thesis, Spring 2015

Preface

Introduction

The history of cryptography dates back to ancient times. The first known use of cryptography was found in Egypt where ciphertext had been carved into stone approximately 2000 years BC. It is also believed that the ancient Greeks had knowledge of ciphers, and simple transposition ciphers are presumed to have been used by the Spartan military. In the 16th century, Mary, Queen of Scots used ciphers to communicate with her allies while being held captive by her cousin, Queen Elizabeth I.

Mary is an early example of the importance of using *secure* ciphers. Elizabeth's codebreakers were able to crack Mary's cipher, and after deciphering Mary's messages to her allies, she was charged with treason and conspiracy against the crown. Mary was convicted, and was sentenced to be *hanged, drawn and quartered* (look it up – it is even worse than it sounds).

A fundamental building block in traditional cryptography was that of a *key*, a secret that only the sender and the recipient possessed. The key would then be used by the sender to encrypt the *plaintext* to *ciphertext*, and the same key would be used by the recipient to decrypt the ciphertext back to the original plaintext.

During the 1970s, a remarkable idea of *public key cryptography* emerged. In public key cryptography, different keys are used for encryption and decryption. If Alice and Bob wants to communicate using public key cryptography, they would each generate a key pair consisting of a public key accessible to everyone, and a private key which is kept secret. Alice would then encrypt her message to Bob using Bob's public key, and Bob would decrypt the message from Alice using his private key, and vice versa.

Behind the scenes of public key cryptography lies a substantial amount of mathematics, and new areas of mathematics continue to find its way to applications in cryptography. An example of this is the use of elliptic curves in public key cryptography, which is the subject of this thesis. It was first suggested used in the mid 1980s, and approximately two decades later, it was in widespread use in modern information systems. Today, elliptic curve cryptography is used by governments, military and corporations worldwide.

In elliptic curve cryptography, it turns out that the choice of the elliptic

curve is important for security. This is because there exist elliptic curves where certain "shortcuts" can be made that breaks the security of elliptic curve cryptography. For this reason, standards such as [9, 21, 25] have been proposed to provide government institutions and the public with a set of secure elliptic curves for use in cryptography.

However, following the Snowden revelations, reports published by the New York Times [22] claims knowledge of an internal memo in the NSA (National Security Agency) describing their involvement in one of the NIST (National Institute of Standards and Technology) standards. In the news article, they quote the alleged memo on the involvement of the NSA: "Eventually, N.S.A. became the sole editor.". This has severely weakened the cryptographic community's trust in the NIST curves, and places higher demands for a provable random generation of proposed elliptic curves in current and future standards.

It is the purpose of this thesis to consider requirements that elliptic curves should satisfy in order to be suitable for cryptographic applications. We shall give a mathematical description of why these requirements affect the security and/or technical aspects (e.g. properties that allow for faster implementations in software) of elliptic curve cryptography. Based on these requirements, we develop a tool for generating secure elliptic curves suitable for cryptographic applications.

Preliminaries and Notation

Although most definitions and results will be stated, basic familiarity with algebraic geometry is advisable, if not necessary. Basic commutative algebra and number theory is also assumed to be familiar to the reader. Readers unfamiliar with these topics are advised to consult [1, 13, 31, 16].

In Chapter 1, most of the definitions and results are trivially generalized to algebraic curves and/or algebraic varieties. For the sake of concreteness, we have nevertheless sticked to the case where our object of study is an elliptic curve. We fix the following notation which will be used throughout this thesis:

$\mathrm{char}(K)$	*the characteristic of a field* K
$\mathbb{F}_p$	*the prime field of* p *elements*
$\mathbb{F}_q$	*the finite field* q *elements*
$\bar{K}$	*the algebraic closure of a field* K
$Gal(\bar{K}/K)$	*the Galois group of* $\bar{K}$ *over* K
$\mathbb{P}^2$	*the projective 2-space*
R^*	*the group of units in a ring* R
P^σ	*the action of* $\sigma \in Gal(\bar{K}/K)$ *on a point* P
ord_P	*the order valuation at the point* P

In this thesis, all rings are assumed to be with unity. Furthermore, E will always denote an elliptic curve, and in cases where there are several elliptic curves in play, they will usually be denoted $E_1, E_2, \ldots$, and so on.

Acknowledgments

First and foremost, I would like to thank my supervisors Professor Kristian Ranestad and Professor Leif Nilsen. Their advice and guidance have been truly invaluable while writing this thesis. I would also like to thank Professor Geir Ellingsrud for lecturing the course MAT4250 during the fall semester of 2014. Professor Ellingsrud customized the course curriculum to accommodate my fellow student Mats Myhr Hansen's and my own thesis, which I am sincerely grateful for.

I would like to thank the University of Oslo, and the Department of Mathematics in particular. During my first years at the University of Oslo, the courses offered by the Department of Mathematics sparked an interest and a curiosity for mathematics, which eventually led me to pursue a master's degree in mathematics.

I would like to thank my parents, my brother Stig Zerener Haugnæss and Ida Jeanette Victoria Valstad for their support while writing this thesis. I would like to thank my good friend Håkon Robbestad Gylterud for inspiring me to study mathematics, and I would also like to direct a thank you to the friends I made during my five years at the University of Oslo. You know who you are.

Contents

Chapter 1

Elliptic Curves

In this chapter we will review the basic theory of elliptic curves that is relevant for this thesis. We begin by defining an elliptic curve, and basic properties and quantities associated to an elliptic curve. Then we shall examine an addition law on the group of points on an elliptic curve, before we proceed by looking at maps between elliptic curves. In the end of this chapter, we will consider alternative forms of representing elliptic curves.

1.1 Definition and Basics

Definition 1.1. *An elliptic curve E over a field K is the set of points in $\mathbb{P}^2(\bar{K})$ satisfying a homogeneous equation of the form:*

$$y^2z + a_1xyz + a_3yz^2 = x^3 + a_2x^2z + a_4xz^2 + a_6z^3 \tag{1.1}$$

where $a_1, \ldots, a_6 \in \bar{K}$. Furthermore, we require that the discriminant

$$\Delta = -b_2^2b_8 - 8b_4^3 - 27b_6^2 + 9b_2b_4b6$$

satisfies $\Delta \neq 0$, where $b_2 = a_1^2 + 4a_2$, $b_4 = 2a_4 + a_1a_3$, $b_6 = a_3^2 + 4a_6$ and $b_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2$.

Equation 1.1 is known as the *Weierstrass form* of an elliptic curve. When the characteristic of the field K is (strictly) greater than 3, E can be written as the set of points in $\mathbb{P}^2(\bar{K})$ satisfying

$$y^2z = x^3 + axz^2 + bz^3 \tag{1.2}$$

for some constants $a, b \in \bar{K}$. This is known as the *simple Weierstrass form* of an elliptic curve. In this case the discriminant Δ has the form $\Delta = -16(4a^3 + 27b^2)$. There are other equivalent definitions and representations of elliptic curves. For example, when $K = \mathbb{C}$ one can show that every elliptic curve over K can be written uniquely as a quotient

$$\mathbb{C}/\Lambda, \text{ where } \Lambda = \{n_1w_1 + n_2w_2 \,|\, n_1, n_2 \in \mathbb{Z}\}^1$$

for some $w_1, w_2 \in \mathbb{C}$. An elliptic curve over the complex numbers is thus uniquely determined by this Λ. In Section 1.12, we will look at a representation that can sometimes give us certain arithmetic advantages over the Weierstass form. If not explicitly stated otherwise, we assume that the characteristic of a field K is greater than 3, and use the simple Weierstrass form of an elliptic curve.

Now we will show that an elliptic curve has exactly one point on the line at infinity. In the next section, we will see that this point plays an important role when defining a group law on E. In fact, it will be the identity element.

Proposition 1.1. *An elliptic curve E has $\mathcal{O} = (0, 1, 0)$ as its only point on the line at infinity.*

Proof. We obtain the points at infinity by setting $z = 0$. Then from Definition 1.1 we get $0 = x^3$. The only point (up to scaling) satisfying this equation is the point $(0, 1, 0)$. □

In our definition of an elliptic curve, the defining (homogeneous) polynomial has coefficients in the algebraic closure of K. Sometimes, for example when representing an elliptic curve on a computer, we are interested in curves that can be defined by a polynomial over K.[2]This prompts the following definition:

Definition 1.2. *An elliptic curve E is said to be defined over K, and we write E/K, if it satisfies a smooth homogeneous equation on Weierstrass form where the coefficients $a_i \in K$. A point $P \in E$ is called K-rational if there exists $x_0, y_0, z_0 \in K$ such that $P = (x_0, y_0, z_0)$.*

For an elliptic curve E over a field K, we shall now consider functions $E \to K$. It turns out that we are particularly interested in functions $E \to K$ that arise as quotients of polynomials, i.e "rational functions". Since a point $P \in E$ is only unique up to scaling with a constant in K, these polynomials must be homogeneous and of the same degree for the quotients to be well defined as functions on E.

Definition 1.3. *Let E be an elliptic curve, and let I denote the ideal of polynomials in $\bar{K}[x, y, z]$ that vanish on all of E. We define the function field of E to be the set of quotients f/g of polynomials in $\bar{K}[x, y, z]$ such that*

1. *g is not everywhere zero on E, i.e $g \notin I$.*
2. *f and g are homogeneous and of the same degree.*

[1] We call Λ a *lattice*. A lattice in $\mathbb{C}$ is a discrete subgroup of dimension 2 as an $\mathbb{Z}$-module.

[2]Assuming that we have enough memory, any polynomial in $\bar{K}[x, y, z]$ can represented on a computer by working over a finite extension of K. However, when the field extension becomes large, this is certainly inconvenient, and may not be feasible.

3. f/g and f'/g' are identified, and we write $f/g \sim f'/g'$ if $fg' - f'g \in I$.

We denote the function field of E by $\bar{K}(E)$, and an element in $\bar{K}(E)$ is called a rational function.

Just as we were interested in representing the defining polynomial of an elliptic curve using polynomials over K, we are also interested in when a rational function can be written as a quotient of polynomials over K. This prompts the next definition.

Definition 1.4. *We define $K(E)$ to be the subfield of $\bar{K}(E)$ consisting of rational functions that can be written as a quotient g/h of homogeneous polynomials of the same degree in $K[x, y, z]$.*

Now we shall define a map $\mathrm{ord}_P \colon \bar{K}(E) \to \mathbb{Z}^+ \cup \{\infty\}$ which will be central in our study of divisors in Section 1.7. We will do this in two steps; first we will define the map on a local subring of $\bar{K}(E)$, and then we will extend it to all of $\bar{K}(E)$.

Definition 1.5. *Let $f/g \in \bar{K}(E)$. If there exists $f'/g' \in \bar{K}(E)$ such that $f/g \sim f'/g'$ and where $g'(P) \neq 0$, we say that f/g is defined at P. We define the local ring at P, denoted $\bar{K}(E)_P$ to be subring of $\bar{K}(E)$ consisting of rational functions f/g that are defined at P.*

It is easily checked that $\bar{K}(E)_P$ is indeed a local ring with the maximal ideal being all $f \in \bar{K}(E)_P$ such that $f(P) = 0$. Notice that if $f \notin \bar{K}(E)_P$, then the "denominator" of f (when considered as a quotient of homogeneous polynomials) must vanish at P, so it does not make sense to evaluate f at P. Now we will define a map on $\bar{K}(E)_P$:

Definition 1.6. *We define $ord_P \colon \bar{K}(E)_P \to \mathbb{Z}^+ \cup \{\infty\}$ to be the map*

$$f \mapsto \sup \left\{ n \in \mathbb{Z}^+ \cup \{\infty\} : f \in M_P^n \right\}$$

where $M_P = \{f \in \bar{K}(E)_P : f(P) = 0\}$.

Let $P \in E$ consider the rational function $f/g \in \bar{K}(E)$. Clearly we can find a homogeneous $h \in \bar{K}[x, y, z]$ of the same degree as f and g such that $h(P) \neq 0$. Then f/h and g/h are in $\bar{K}(E)_P$. We shall extend the map ord_P to $\bar{K}(E)$ by setting $\mathrm{ord}_P(f/g) = \mathrm{ord}_P(f/h) - \mathrm{ord}_P(g/h)$.

Proposition 1.2. *Let $P \in E$, and let $f/g \in \bar{K}(E)$ be a rational function. Let $h \in \bar{K}[x, y, z]$ be a polynomial of the same degree as f and g, and with $h(P) \neq 0$. Then the map $ord_P \colon \bar{K}(E) \to \mathbb{Z} \cup \{\infty\}$ defined by*

$$f/g \mapsto ord_P(f/h) - ord_P(g/h)$$

is well-defined.

Proof. Clearly f/h and g/h are in $\bar{K}(E)_P$. We need to show that $\mathrm{ord}_P(f/g)$ is independent of the choice of $h \in \bar{K}[x, y, z]$. Since $h \notin M_P$, then $f/h \in M_P^n \Leftrightarrow f \in M_P^n$. This is obviously also true for g/h and g, so the map is well-defined. □

Proposition 1.3. *The map* $ord_P \colon \bar{K}(E) \to \mathbb{Z} \cup \{\infty\}$ *defined above is a discrete valuation on* $\bar{K}(E)$.

Proof. See Section II.1 in [31], and/or Proposition 9.2 in [1]. □

1.2 Group Law and Torsion Points

In this section we define a group law on the points on an elliptic curve. Then we will review torsion points on an elliptic curve, and state a proposition about the structure of the m-torsion subgroups of an elliptic curve.

We define the following binary operation on the set of points on an elliptic curve:

Definition 1.7 (Addition Law on E)**.** *Let* $P, Q \in E$. *Let* L *be the line in* $\mathbb{P}^2$ *determined by* P *and* Q *(and the tangent line to* P *if* $P = Q$*).* L *intersects* E *in a third point* $R \in E$. *Let* L' *be the line determined by* R *and* $\mathcal{O}$. *Then* L' *intersects* E *in a third point which we denote* $P + Q$.

Geometrically, the group law can be illustrated as following:

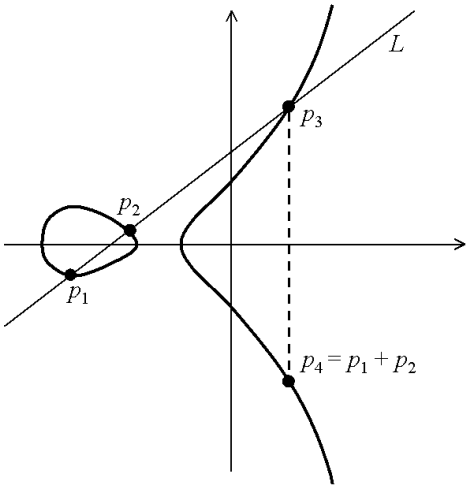

Figure 1.1: The geometric group law on an elliptic curve, illustrated by [17].

Proposition 1.4. *The binary operation* $+ : E \times E \to E$ *defines an abelian group law on* E *with* $\mathcal{O}$ *as identity element.*

Proof. Geometrically, it is easy to convince oneself that all the group axioms except associativity holds. A solid proof that $+: E \times E \to E$ does indeed define an abelian group law can be done by deriving explicit formulas for the group law, and then algebraically verifying that the group axioms are satisfied. □

By abuse of language, we shall sometimes refer to E as a group. It is then understood that we then mean the set of points in E equipped with the group law from Definition 1.7. For the sake of completeness, we give explicit formulas for the group law on E.

Proposition 1.5. *Let E be an elliptic curve over K with char$(K) > 3$. Assume E is given on simple Weierstrass form* (1.3)*. Let $P = (x_1, y_1, z_1)$ and $Q = (x_2, y_2, z_2)$ be points on E, and let $P + Q = (x_3, y_3, z_3)$. Then*

(i) If $P = Q$, then

$$\begin{aligned} x_1 &= 2y_1z_1(3(3x_1^2 + az_1^2)^2 - 8y_1^2x_1z_1) \\ y_2 &= (3x_1^2 + az_1^2)(12y_1^2x_1z_1 - (3x_1^2 + az_1^2)^2) - 8y_1^4z_1^2 \\ z_2 &= 8y_1^3z_1^3 \end{aligned}$$

(ii) If $P \neq Q$, then

$$\begin{aligned} x_3 &= (x_1z_1 - z_1x_2)(z_1z_2(y_1z_2 - z_1y_2)^2 - (x_1z_2 + z_1x_2)(x_1z_2 - z_1x_2)^2) \\ y_3 &= (y_1z_2 - z_1y_2)((2x_1z_2 + z_1x_1)(x_1z_2 - z_1x_2)^2 - z_1z_2(y_1z_2 - z_1y_2)^2) \\ &\quad - y_1z_2(x_1z_2 - z_1x_2)^3 \\ z_3 &= z_1z_2(x_1z_2 - z_1x_2)^3 \end{aligned}$$

Proof. See the discussions prior to the Group Law Algorithm in Section II.2 in [31], and repeat the arguments using homogeneous coordinates. □

Of particular interest are the points on E of finite order, i.e points $P \in E$ such that $P + \ldots + P = \mathcal{O}$. These points are called *torsion points*, and they are easily seen to form a subgroup E_{tors} of E. The set of K-rational points on E is also a subgroup of E, and when K is a finite field, this subgroup is necessarily finite. We have in this case that $E(K) \subseteq E_{tors}$. Hence, torsion points are intrinsic when working with elliptic curves over finite fields.

Definition 1.8. *Let $P \in E$. We call P an m-torsion point if $P + \ldots + P$ (m times) equals $\mathcal{O}$. We denote the subgroup of m-torsion points of E by $E[m]$.*

Now we will state a proposition which gives us complete description of the group structure of the set of $E[m]$.

Proposition 1.6. *Let E be an elliptic curve and let $m \in \mathbb{Z}$. Then:*

(a) If $m \neq 0$ in K, $E[m] \cong \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$

(b) If $char(K) = p > 0$, we have either

(i) $E[p^e] \cong \{\mathcal{O}\}$ for all $e \in \mathbb{N}$ or

(ii) $E[p^e] \cong \mathbb{Z}/p^e\mathbb{Z}$ for all $e \in \mathbb{N}$

Proof. See Corollary III.6.4 in [31] □

1.3 Maps Between Elliptic Curves and the Frobenius Morphism

In this section we will review the basic theory of maps between elliptic curves. We begin by defining rational maps and morphisms of elliptic curves. Then we consider two different criteria for elliptic curves to be *isomorphic*. [3] In the end of this section we will look at the *Frobenius morphism*.

Definition 1.9. *Let E_1 and E_2 be elliptic curves defined over K. A rational map is a map $\phi\colon E_1 \to E_2$ defined by*

$$\phi(P) = (f_1(P), f_2(P), f_2(P))$$

where $f_1, f_2, f_3 \in K(E)$, and $\phi(P) \in E_2$ at all P where f_1, f_2, f_3 is defined (i.e where all $f_i \in K(E)_P$).

Suppose $\phi\colon E_1 \to E_2$ is a rational map defined by $\phi = (f_1, f_2, f_3)$, and let $g \in \bar{K}(E)$. Assume that ϕ is defined at a point $P \in E_1$, and suppose we can find $g \in \bar{K}(E)$ such that $h_i = f_i g$ is defined at P for all i. Let $\phi'\colon E_1 \to E_2$ be the rational map defined by $\phi' = (h_1, h_2, h_3)$. Since projective coordinates are only unique up to scaling, we have that $\phi(P) = \phi'(P)$ for all points $P \in E_1$ where they are both defined. Now suppose instead that ϕ is *not* defined at P, but that ϕ' *is* defined at P. Then it still makes sense to evaluate ϕ at P by setting $\phi(P) = \phi'(P)$. This justifies the definition:

Definition 1.10. *Let $\phi : E_1 \to E_2$ be a rational map of elliptic curves, and let $\phi = (f_1, f_2, f_3)$ for some rational functions $f_i \in \bar{K}(E_1)$. We say that ϕ is defined at P if either:*

1. *$f_i \in \bar{K}(E_1)_P$ for all i, and $f_i(P) \neq 0$ for some i, or*
2. *There exists $g \in \bar{K}(E_1)$ such that $f_i g \in \bar{K}(E_1)_P$ for all i, and $(f_i g)(P) \neq 0$ for some i. In this case we set $\phi(P) = ((f_1 g)(P), (f_2 g)(P), (f_3 g)(P))$.*

[3] A priori, it is not clear exactly what "isomorphic" means in this context, but we will define it shortly.

If ϕ is defined at all points of E_1, we say that ϕ is a morphism. Furthermore, if there exists a morphism $\phi^{-1}\colon E_2 \to E_1$ such that $\phi \circ \phi^{-1} = id_{E_2}$ and $\phi^{-1} \circ \phi = id_{E_1}$ then ϕ is said to be an isomorphism, and we say that E_1 and E_2 are isomorphic.

Definition 1.11. *Let $\phi\colon E_1 \to E_2$ be a rational map of elliptic curves, and assume $\phi = (f_1, f_2, f_3)$ for some rational functions $f_1, f_2, f_3 \in \bar{K}(E_1)$. We say that ϕ is defined over K if there exists $f_i \in K(E_1)$ such that $\phi = (f_1, f_2, f_3)$. If ϕ is an isomorphism defined over K, we say that E_1 and E_2 are isomorphic over K.*

Since a rational map $\phi\colon E_1 \to E_2$ is defined in terms of rational functions, the composition of a ϕ and a rational function on E_2 gives a rational function on E_1. Thus, a rational map between induces a map of the the corresponding function fields.

Definition 1.12. *Let $\phi\colon E_1 \to E_2$ be a rational map. We define $\phi^*\colon \bar{K}(E_2) \to \bar{K}(E_1)$ to be the map given by $f \mapsto f \circ \phi$.*

As the next proposition shows, this map is in fact an injection of function fields and consequently also gives rise to a field extension.

Proposition 1.7. *The map $\phi^*\colon \bar{K}(E_2) \to \bar{K}(E_1)$ is an injection of function fields, and $\bar{K}(E_2)/\phi^*\bar{K}(E_1)$ is a field extension of finite degree.*

Proof. Composing a rational function with a rational map clearly gives a rational function since a rational map is defined in terms of rational functions, and the composition of two rational functions is a rational function. Hence, the map ϕ^* induces a map of function fields, and it is easily seen to be injective. Since the field extensions $\bar{K}(E_2)/\bar{K}$ and $\phi^*\bar{K}(E_1)/\bar{K}$ is of finite degree, then $\bar{K}(E_2)/\phi^*\bar{K}(E_1)$ must be of finite degree as well. □

We will say that a rational map ϕ is *separable* if the corresponding field extension is separable, and we define the *degree* of a rational map to be the degree of the induced field extension. We similarly define the separability degree $\deg_s(\phi)$ and the inseparability degree $\deg_i(\phi)$ of ϕ. Note that if ϕ is an isomorphism, then ϕ^* is an isomorphism of function fields, and so the degree of ϕ is 1. The next proposition gives us information about the fibers of separable rational maps.

Proposition 1.8. *Let $\phi\colon E_1 \to E_2$ be a map. Then $\#\phi^{-1}(P) = deg_s(\phi)$ for all but finitely many $P \in E_2$. In particular, if ϕ is separable, then $\#\phi^{-1}(P) = deg(\phi)$ for all but finitely many $P \in E_2$.*

Proof. See Proposition II.6.9 in [13]. □

Now we will turn our attention to questions regarding the Weierstrass form of an elliptic curve. To what extent is the Weierstrass form of an elliptic curve unique, and when are two elliptic curves isomorphic?

Proposition 1.9. *Let $E_1/K\colon y^2z = x^3 + axz^2 + bz^3$ and $E_2/K\colon y'^2z' = x'^3 + a'x'z'^2 + b'z'^3$ be elliptic curves that are isomorphic over K. Then E_1 and E_2 are related by a linear change of variables of the form $x = u^2x' + r$ and $y = u^3y' + su^2x' + t$ for some $u \in K^*$, $r, s, t \in K$.*

Proof. See Proposition III.3.1 in [31]. □

Every elliptic curve has an associated quantity called the *j-invariant*. The j-invariant will give us a condition for determining when two elliptic curves are isomorphic over $\bar{K}$.

Definition 1.13. *Let $E\colon y^2z = x^3 + axz^2 + bz^3$ be an elliptic curve over K. Then E has an associated quantity called the j-invariant, which is a quotient of polynomials in $\mathbb{Z}[a, b]$. It is given by*

$$j(E) = \frac{2^8 3^3 a^3}{4a^3 + 27b^2}$$

One can show that any admissible change of variables (as given by Proposition 1.9) leaves the j-invariant unchanged. Hence, the j-invariant does indeed live up to its name. Note that in general we have $j(E) \in \bar{K}$, but when E is defined over K, we also have that $j(E) \in K$.

Proposition 1.10. *Let E_1 and E_2 be elliptic curves over K. Then E_1 and E_2 are isomorphic over $\bar{K}$ if and only if $j(E_1) = j(E_2)$.*

Proof. We need only show that $j(E_1) = j(E_2)$ implies that E_1 and E_2 are isomorphic over $\bar{K}$. We outline a rough sketch of a proof of this: If the E_1 and E_2 are isomorphic over $\bar{K}$, then one can use explicit formulas for the j-invariant and the admissible change of variables from Proposition 1.9 to show that the j-invariant is invariant under this change of variables. If $j(E_1) = j(E_2)$, then we can again use explicit formulas for the j-invariant to deduce a change of variables of the form given in Proposition 1.9. A detailed proof can be found in Proposition III.1.4 in [31]. □

Consider now the field $\mathbb{F}_q$ where $q = p^n$, and the map $\bar{\mathbb{F}}_q \to \bar{\mathbb{F}}_q$ defined by $x \mapsto x^q$. This map is called the *q-power Frobenius map*. It is a well known result from algebra that $x^q = x$ if and only if $x \in \mathbb{F}_q$, so this map is in $Gal(\bar{\mathbb{F}}_q/\mathbb{F}_q)$. The q-power Frobenius map acts on a point $P \in E(\mathbb{F}_q)$ by raising the coefficients of P to the q-th power. This action induces the following map of curves:

Definition 1.14. *Let E be an elliptic curve over a field K of characteristic $p > 0$, and let $q = p^r$ for some $r \geq 1$. We define the q-power Frobenius morphism to be the map*

$$\phi\colon E \to E^{(q)}, \quad \phi((x_0, y_0, z_0)) \mapsto (x_0^q, y_0^q, z_0^q)$$

where $E^{(q)}$ is the curve obtained by raising the coefficients of the defining polynomial of E to the q-th power.

If an elliptic curve E is *defined* over $\mathbb{F}_q$, then the defining polynomial for E can be written using coefficients in $\mathbb{F}_q$. Since the q-power Frobenius map leaves elements in $\mathbb{F}_q$ fixed, this implies that $E = E^{(q)}$. Hence, when $E/\mathbb{F}_q$, the Frobenius map induces an *endomorphism* $\phi_q \in End(E)$ of E. Furthermore, we see that $\phi_q(P) = P$ if and only if $P \in E(\mathbb{F}_q)$. This turns out to be a crucial observation in the proof of Hasse's theorem (see Theorem 1.2). We conclude this section with a proposition stating a few properties of the Frobenius morphism.

Proposition 1.11. *Let E be an elliptic curve over a field K of characteristic $p > 0$, and let $q = p^r$. The q-th power Frobenius morphism $E \to E^{(q)}$ has the following properties:*

1. *ϕ is purely inseparable.*
2. *$deg(\phi) = q$.*

Proof. See Proposition II.2.11 in [31]. □

1.4 The Reduction Map

In this section we will consider an elliptic curve E defined over a field K. We will suppose that K has a discrete valuation $\nu\colon K \to \mathbb{Z} \cup \{\infty\}$, and let $R \subseteq K$ be the associated discrete valuation ring. Then $R = \{x \in K : \nu(x) \geq 0\}$, and R has a maximal ideal $\mathfrak{m}$ given by $\mathfrak{m} = \{x \in K : \nu(x) > 0\}$. We let $k = R/m$ denote the corresponding residue field.

Now consider the curve obtained by reducing the coefficients of E modulo the maximal ideal $\mathfrak{m}$. This only makes sense when the defining homogeneous polynomial can written using coefficients such that $\nu(a_i) \geq 0$, i.e if all the coefficients are in R. It is easily seen using the transformation formulas from Proposition 1.9 that such a representation always exists.

Definition 1.15. *Let E/K be an elliptic curve over K. We define a minimal Weierstrass equation for E to be a Weierstrass equation for E such that $\nu(\Delta)$ is minimal over all Weierstrass equations for E.*

Since the discrete valuation ν on K is a function $K \to \mathbb{Z} \cup \{\infty\}$, the existence of such a minimal discriminant is clear. This is due to the trivial observation that all non-empty subsets of $\mathbb{Z} \cup \{\infty\}$ have a minimum. Choosing a minimal Weierstrass equation for E/K allows us to *reduce* E modulo $\mathfrak{m} \subseteq R$ to a curve $\tilde{E}/k$ simply by reducing the coefficients modulo $\mathfrak{m}$. The reduced curve $\tilde{E}/k$ may not be an elliptic curve, since we require the reduced discriminant $\tilde{\Delta}$ to be non-zero. In the case where $\tilde{\Delta} \neq 0$, we say that E has *good reduction* at ν, and we define the following reduction map:

Definition 1.16. *Let K be a field with a discrete valuation ν. Let E/K be an elliptic curve, and suppose E has good reduction at ν. We define the reduction map π_ν to be the map*

$$\pi_\nu \colon E \to \tilde{E}, \quad P \mapsto \tilde{P}$$

where $\tilde{P}$ is the point on $\tilde{E}$ obtained by reducing the coefficients of E modulo $\mathfrak{m}$. We denote the kernel of π_ν by E_1.

Proposition 1.12. *Let E/K be an elliptic curve over K, and assume E has good reduction at ν. Then the reduction map $\pi_\nu \colon E \to \tilde{E}$ defines a group homomorphism.*

Proof. The reduction map sends a line in $\mathbb{P}^2$ to another line in $\mathbb{P}^2$. Since the group law is defined in terms of intersections between rational lines, it follows that the group law is a homomorphism. □

1.5 The Endomorphism Ring

On an elliptic curve E over a field K there is a natural ring associated to E, namely the *endomorphism ring* of E. It is the ring of all morphisms $E \to E$, and is denoted $End(E)$. The arch example of an endomorphism of E is the *multiplication-by-m* map. It is the map $E \to E$ defined by $P \mapsto P + \ldots + P$ (m times). We review a few properties of the multiplication-by-m map:

Proposition 1.13. *Let E be an elliptic curve and let $m \in \mathbb{Z}$ with $m \neq 0$ in K. Then the multiplication-by-m map $[m] \colon E \to E$ has the following properties:*

1. *$[m]$ is a morphism.*
2. *$[m]$ is separable map.*
3. *$deg\,[m] = m^2$.*

Proof. We will only give a very rough sketch of the proof of 1. The addition map $+ \colon E \times E \to E$ is defined in terms of rational functions, so it is easily seen to be a rational map. Since we can add any two points on an elliptic curve, it is defined for all pairs of points, so it is a morphism. Then it is immediate that the multiplication-by-m map is a morphism too, as the composition of two morphisms maps is a morphism. A detailed proof can be found in the proof of Theorem III.3.6 in [31]. For 2 and 3, see the proofs of Corollary III.5.6 and Theorem III.6.2 in [31]. □

Notice that since $ker[m] = E[m]$, Proposition 1.6 gives us a description of the kernel of the multiplication-by-m map. We shall now state a theorem that will give us information on the structure of $End(E)$. Before stating the theorem, we need the following two definitions from [31].

Definition 1.17. *Let $\mathcal{K}$ be a $\mathbb{Q}$-algebra that is finitely generated over $\mathbb{Q}$. We define an order $\mathcal{R}$ of $\mathcal{K}$ to be a subring of $\mathcal{K}$ that is finitely generated as a $\mathbb{Z}$-module a, and with $\mathcal{R} \otimes \mathbb{Q} = \mathcal{K}$.*

Definition 1.18. *A quaternion algebra is an algebra of the form*

$$\mathcal{K} = \mathbb{Q} + \mathbb{Q}\alpha + \mathbb{Q}\beta + \mathbb{Q}\alpha\beta$$

whose multiplication satisfies

$$\alpha^2, \beta^2 \in \mathbb{Q} \quad \alpha^2 < 0, \quad \beta^2 < 0 \quad \beta\alpha = -\alpha\beta.$$

Theorem 1.1. *The endomorphism ring of an elliptic curve E/K is either $\mathbb{Z}$, an order in an imaginary quadratic field, or an order in a quaternion algebra. If $char(K) = 0$, then only the first two are possible.*

Proof. See Corollary III.9.4 in [31]. □

Corollary 1.1. *As a $\mathbb{Z}$-module, $End(E)$ can only have rank 1, 2 or 4.*

Proof. This is immediate from Theorem 1.1, and the fact that an order $\mathcal{R}$ of $\mathcal{K}$ is finitely generated as a $\mathbb{Z}$-module, and satisfies $\mathcal{R} \otimes \mathbb{Q} = \mathcal{K}$. □

In the beginning of this chapter, we noted that elliptic curves over the complex numbers can be written as a quotient $\mathbb{C}/\Lambda$ where Λ is a discrete subgroup of $\mathbb{C}$ of dimension 2 as a $\mathbb{Z}$-module. Now if $E_1 = \mathbb{C}/\Lambda_1$ and $E_2 = \mathbb{C}/\Lambda_2$ are two elliptic curves over $\mathbb{C}$, then one can show that every morphism between E_1 and E_2 can be realized as multiplication by a complex number $z \in \mathbb{C}$ such that $z\Lambda_1 \subseteq \Lambda_2$.

When $E_1 = E_2$, then usually the only $z \in \mathbb{C}$ that satisfies this requirement are precisely those z that are in $\mathbb{Z}$, in which case $End(E) \cong \mathbb{Z}$. In the occasion that $End(E)$ has an endomorphism that is not a multiplication-by-m map for $m \in \mathbb{Z}$, we make the following definition:

Definition 1.19. *If $End(E)$ has $\mathbb{Z}$-rank 2 or 4, then E is said to have complex multiplication.*

In some sense, the endomorphism ring of E gives a rough measure on the amount of structure that the curve possesses. In cryptography, we are interested in curves with as little structure as possible. This intuition comes from the suspicion that more structure could perhaps provide an attacker with dirty tricks for solving the ECDLP. For example, elliptic curves with endomorphism rings with $\mathbb{Z}$-rank 4 are called supersingular elliptic curves, and as it turns out, they are indeed vulnerable to known attacks on elliptic curves (see Section 3.1.2 for a description of such an attack).

1.6 Isogenies

We now turn our attention to a special type of morphism between elliptic curves. These are the morphisms that respect the identity elements of the groups of points on an elliptic curve.

Definition 1.20. *We define an isogeny to be a non-zero morphism $\phi\colon E_1 \to E_2$ such that $\phi(\mathcal{O}_{E_1}) = \mathcal{O}_{E_2}$, where $\mathcal{O}_{E_1}$ and $\mathcal{O}_{E_2}$ are the identity elements of the group of points on E_1 and E_2 respectively.*

We have already seen two examples of isogenies, namely the multiplication-by-m map, and the q-power Frobenius morphism. Both are easily verified to be isogenies. The next proposition is somewhat peculiar. It states that a morphism that decides to respect the identity element of elliptic curves will automatically also respect their group structure.

Proposition 1.14. *Let $\phi\colon E_1 \to E_2$ be an isogeny. Then ϕ is a homomorphism of the additive groups of points on the elliptic curves.*

Proof. This can be proved with straightforward but tedious calculations using explicit formulas for the group law on elliptic curves. Alternatively, using the theory of divisors which we introduce in the next section, one can consider the diagram

$$\begin{array}{ccc} E_1 & \xrightarrow{\tau} & Pic_0(E_1) \\ {\scriptstyle\phi}\downarrow & & \downarrow{\scriptstyle\phi^*} \\ E_2 & \xrightarrow[\tau]{} & Pic_0(E_2) \end{array}$$

with $\tau = \sigma^{-1}$ where σ is the isomorphism defined in Corollary 1.2. One can check that the diagram commutes. Since σ^{-1} is an isomorphism and ϕ^* is a homomorphism, ϕ must be a homomorphism. □

Proposition 1.15. *Let $\phi\colon E_1 \to E_2$ be an isogeny, and let $m = deg(\phi)$. Then there exists a unique dual isogeny $\hat{\phi}\colon E_2 \to E_1$ such that $\hat{\phi} \circ \phi = [m]$. Assume $\phi, \psi\colon E_1 \to E_2$ are isogenies, then the following properties hold:*

1. $\hat{\hat{\phi}} = \phi$
2. $deg(\phi) = deg(\hat{\phi})$
3. $\widehat{\phi + \psi} = \hat{\phi} + \hat{\psi}$

Proof. See Theorem III.6.1 and Theorem III.6.2 in [31]. □

1.7 Divisors

In this section we will introduce *divisors*, and prove basic results about them. Divisors are a powerful tool when studying curves in general, but we will restrict our attention to only cover what we need to define the Weil pairing in Section 1.8. Our goal in this section will be to give necessary and sufficient conditions for a divisor to be *principal*.

Definition 1.21. *We define the divisor group of E, denoted $Div(E)$ to be the group of formal sums $\sum_{P\in E} n_P P$ with $n_P \in \mathbb{Z}$, where only finitely many n_P are non-zero, i.e. it is the free abelian group generated by the points on E. An element of $Div(E)$ is called a divisor of E.*

Next we will define an important quantity associated to every divisor.

Definition 1.22. *Let $D \in Div(E)$. We define the degree of D to be $\sum_{P\in E} n_P$. We denote the degree of D by $deg(D)$.*

There are certain divisors that are of particular interest to us, and they will play a central role when we define the aforementioned Weil e_m-pairing in Section 1.8. These are the divisors that arise from rational functions on E:

Definition 1.23. *For a rational function $f \in \bar{K}(E)$, we define the divisor $div(f)$ of f to be $\sum_{P\in E} ord_P(f)P$. A divisor $D \in Div(E)$ such that $D = div(f)$ for some $f \in \bar{K}(E)$ is called a principal divisor.*

Now we define an equivalence relation on $Div(E)$ by identifying two divisors if their difference is a principal divisor.

Definition 1.24. *Two divisors $D_1, D_2 \in Div(E)$ are said to be linearly equivalent and we write $D_1 \sim D_2$ if $D_1 - D_2$ is a principal divisor. That is, $D_1 - D_2 = div(f)$ for some $f \in \bar{K}(E)$.*

It is readily checked that $\sim$ does indeed define an equivalence relation on $Div(E)$.

Definition 1.25 (Picard Group)**.** *We define the Picard group, denoted $Pic(E)$ to be $Div(E)$ under the equivalence relation $\sim$.*

The group $Pic(E)$ is called the *Picard group* of E. We will now state and prove a lemma which we will use to define a map between divisors of degree zero and points on the elliptic curve. We shall denote the subgroup of $Div(E)$ consisting of divisors of degree zero by $Div_0(E)$.

Lemma 1.1. *Let $D \in Div_0(E)$. Then there exists a unique $P \in E$ satisfying $D \sim P - \mathcal{O}$. Consequently, the map $\sigma\colon Div_0(E) \to E$ defined by $D \mapsto P$ is well-defined, and $ker(\sigma) = \{div(f) \mid f \in \bar{K}(E)\}$.*

Proof. In this proof we will use the Riemann-Roch theorem. For a divisor $D \in Div(E)$ and the vector space $\mathcal{L}(D) = \{f \in \bar{K}(E) : div(f) \geq -D\}$, the Riemann-Roch theorem says that $\dim_{\bar{K}} \mathcal{L}(D) = \deg(D)$.

We begin by proving uniqueness by showing that for $P, Q \in E$ we have $P \sim Q \Leftrightarrow P = Q$. If $P \sim Q$, then $div(g) = P - Q$ for some $g \in \bar{K}(E)$, so $div(g) - P = -Q$. Then we have $g \in \mathcal{L}(Q)$. Now $\bar{K} \subseteq \mathcal{L}(Q)$ since $div(k) = 0 \geq -Q$ for any $k \in \bar{K}$. However, by the Riemann-Roch theorem $\dim_{\bar{K}} \mathcal{L}(Q) = 1$, so $g \in \bar{K}$ and consequently $P = Q$.

To prove existence, consider the $\bar{K}$-vector space $\mathcal{L}(D+\mathcal{O})$. By Riemann-Roch, $\dim_{\bar{K}} \mathcal{L}(D + \mathcal{O}) = 1$, so there exists a non-zero $f \in \bar{K}(E)$ such that $div(f) \geq -D - \mathcal{O}$. Then f is a basis for $\mathcal{L}(D)$. Taking degrees we see that $deg(div(f)) = 0$ and $deg(-D-\mathcal{O}) = -1$. It follows that $div(f) = -D-\mathcal{O}+P$ for some unique $P \in E$, as the only divisors of E that are positive and of degree 1 are precisely the points of E. This proves that $P \sim D + \mathcal{O}$.

To conclude the proof of this lemma, we observe that for a divisor $D \in Div_0(E)$, we have $D \sim \mathcal{O}$ if and only if $D = div(f)$ for some $f \in \bar{K}(E)$. Then it follows immediately that $ker(\sigma) = \{div(f) \mid f \in \bar{K}(E)\}$. □

The previous lemma showed that there exists a well-defined map σ sending a divisor of degree zero to a point on an elliptic curve.

Proposition 1.16. *The map $\sigma\colon Div_0(E) \to E$ defined in Lemma 1.1 is a group homomorphism.*

Proof. See [31], Proposition III.3.4. □

Corollary 1.2. *Let $D \in Pic_0(E)$ and let $P \in E$ be the unique point satisfying $D \sim P - \mathcal{O}$. Then the map $\sigma\colon Pic_0(E) \to E$ defined by $D \mapsto P$ is an isomorphism of groups.*

Proof. Let $\sigma\colon Div_0(E) \to E$ be the map from Proposition 1.16. It is easily verified that σ is a surjective group homomorphism, and that $ker(\sigma) = \{D \in Div_0(E) : D \sim 0 \Leftrightarrow D = div(f) \text{ for some } f \in \bar{K}(E)\}$, so $Div_0(E)/ker(\sigma) \cong E$ and σ induces an isomorphism $\sigma\colon Pic_0(E) \to E$. □

Now we have the tools we need to determine necessary and sufficient conditions for a divisor to be principal.

Proposition 1.17. *A divisor $D = \sum_{P \in E} n_P(P) \in Div(E)$ is principal if and only if*

$$\sum_{P \in E} n_P = 0 \qquad \text{and} \qquad \sum_{P \in E} [n_P]P = \mathcal{O}$$

Proof. Assume $D \in Div_0(E)$. $D \sim 0$ if and only if $\sigma(D) = \mathcal{O}$ where $\sigma\colon Div_0(E) \to E$ is the isomorphism from Corollary 1.2. Clearly we have $\sigma((P) - (\mathcal{O})) = P$, so $\sigma(D) = \sigma(\sum_{P \in E} n_P(P)) = \sum_{P \in E} [n_P]\sigma((P) - (\mathcal{O}))$ □

1.8 The Weil Pairing

In this section we will define the *Weil e_m-pairing* on the group of m-torsion points on an elliptic curve E over K. Unlike the determinant pairing (which we can define on any free module), the Weil e_m-pairing is *Galois invariant*. It will be a key tool when we study the MOV attack in Section 3.1.2.

Lemma 1.2. *Let $T \in E[m]$, and assume $T = [m]T'$ for some $T' \in E$. For any $Q \in E[m]$, we set*

$$[m]^*(Q) = \sum_{P \in [m]^{-1}(Q)} (P)$$

Then the divisor $D = [m]^*(T) - [m]^*(\mathcal{O})$ is principal.

Proof. Consider the set $E[m] + T' = \{R + T' : R \in E[m]\}$. Clearly $E[m] + T' \subseteq [m]^{-1}(T')$. By Proposition 1.13, we have that $\#[m]^{-1}(T') = \#[m]^{-1}(\mathcal{O})$, so $E[m] + T' = [m]^{-1}(T)$. Hence

$$\begin{aligned} D = [m]^*(T) - [m]^*(\mathcal{O}) &= \sum_{P \in [m]^{-1}(T)} (P) - \sum_{P \in E[m]} (P) \\ &= \sum_{R \in E[m]} (R + T') - (R) \end{aligned}$$

It is obvious that $deg(D) = 0$. Now $\#E[m] = m^2$, so we see that the divisor sums to $\mathcal{O}$ since $[m]([m]T') = [m]T = \mathcal{O}$. By Proposition 1.17, D is then principal. □

Assume $T \in E[m]$. By Proposition 1.17, we can find $f \in \bar{K}(E)$ such that $div(f) = m(P) - m(O)$, and by Lemma 1.2 we can find $g \in \bar{K}(E)$ such that $div(g) = [m]^*(T) - [m]^*(\mathcal{O})$. We will use this to construct two rational functions having the same divisor.

Lemma 1.3. *Let $T \in E[m]$ with $T = [m]T'$ for some $T' \in E$. Let $f, g \in \bar{K}(E)$ be rational functions satisfying $div(f) = m(T) - m(\mathcal{O})$ and $div(g) = [m]^*(T) - [m]^*(\mathcal{O})$. Then $div(f \circ [m]) = div(g^m)$.*

Proof. This follows from the straightforward calculations:

$$\begin{aligned} div(g^m) = m\, div(g) &= \sum_{P \in [m]^{-1}(T)} m(P) - \sum_{P \in E[m]} m(P) \\ &= \sum_{P \in [m]^{-1}(T)} (T) - \sum_{P \in E[m]} \mathcal{O} \\ &= m^2(T) - m^2(\mathcal{O}) \\ &= m([m]T) - m([m]\mathcal{O}) \\ &= div(f \circ [m]) \end{aligned}$$

where the fourth equality holds because $[m]$ is a separable map by Proposition 1.13, so $\#[m]^{-1}(T) = deg([m]) = m^2$. □

Now we have what we need to define the Weil e_m-pairing on $E[m]$:

Proposition 1.18. *The map $e_m \colon E[m] \times E[m] \to \mu_m$ defined by*

$$e_m(S,T) = \frac{g(X+S)}{g(S)}$$

where g is determined by T and $X \in E$ is arbitrary, is a well defined map.

Proof. Let $S \in E[m]$ and $X \in E$. Then $g(X+S)^m = f([m]X + [m]S) = f([m]X) = g(X)^m$. Hence $g(X+S)/g(S)$ is an m-th root of unity. To show that it is independent of the choice of X, we observe that the map $E \to \mathbb{P}^1$ given by $X \mapsto g(X+S)/g(X)$ takes on finitely many values (since $g(X+S)/g(X)$ is an m-th root of unity), so the map must be constant by Proposition II.2.3 in [31]. □

Now we will look at some important properties about the Weil e_m-pairing, which will be central in the MOV-attack on an elliptic curve (see Section 3.1.2).

Proposition 1.19. *The Weil e_m-pairing has the following properties:*

1. *It is bilinear*
2. *$e_m(T,T) = 1$ (alternating)*
3. *If $e_m(S,T) = 1$ for all $S \in E[m]$, then $T = \mathcal{O}$ (non-degenerate)*
4. *$e_m(S,T)^\sigma = e_m(S^\sigma, T^\sigma)$ for all $\sigma \in Gal(\bar{\mathbb{F}}_q/\mathbb{F}_q)$ (Galois invariant)*
5. *$e_{mm'}(S,T) = e_m([m']S,T)$ for all $S \in E[mm']$ and $T \in E[m]$ (compatible)*

Proof. See Proposition III.8.1 in [31]. □

Remark 1.1. *If two m-torsion points $Q, R \in E$ is in the same cyclic subgroup generated by a point $P \in E$ of order m, then $e_m(Q,R) = e_m([k_1]P, [k_2]P)$ for some $k_1, k_2 \in \mathbb{Z}$, so $e_m(Q,R) = e_m(P,P)^{k_1 k_2} = 1$ since e_m is bilinear and alternating.*

Remark 1.2. *If $E[m] \subseteq E(\mathbb{F}_q)$, then the Galois invariance of the Weil e_m-pairing gives that $e_m(P,Q) = e_m(P^\sigma, Q^\sigma) = e_m(P,Q)^\sigma$ for all $P, Q \in E[N]$ and all $\sigma \in Gal(\bar{K}/K)$, so $e_m(P,Q) \in K^*$.*

1.9 Elliptic Curves over Finite Fields

So far we have looked at elliptic curves over an arbitrary field K. Elliptic curves over finite fields are of particular interest in cryptography since they can be represented on a computer. In this section, we will restrict our attention to elliptic curves over finite fields. Our primary interest will be the number of rational points on the curve. This quantity is interesting in its own right, but it turns out to be of particular importance for elliptic curves in cryptography. In the end of this section, we will briefly discuss the endomorphism ring of elliptic curves over finite fields.

We begin with stating and proving a famous theorem by Hasse. The theorem gives a bound on the number of rational points on an elliptic curve.

Theorem 1.2 (Hasse)**.** *Let $E/\mathbb{F}_q$ be an elliptic curve defined over a finite field. Then*

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

Proof. Let ϕ be the q-th power Frobenius morphism. Then $P \in E(\mathbb{F}_q)$ if and only if $\phi(P) = P$. This implies $E(\mathbb{F}_q) = ker(1-\phi)$. $1-\phi$ is a separable map by Corollary 5.5 in [31], so $\#E(\mathbb{F}_q) = deg(1-\phi)$. The degree map is a quadratic form on $End(E)$, so by the Cauchy-Schwartz inequality we get $|deg(1-\phi)-deg(\phi)-deg(1)| \leq 2\sqrt{deg(\phi)deg(1)}$ since $deg(\phi) = q$ (being the q-th power Frobenius map), and $deg(1) = 1$ we get $|\#E(\mathbb{F}_q) - q - 1| \leq 2\sqrt{q}$, which completes the proof. □

The theorem of Hasse essentially says that the number of $\mathbb{F}_q$-rational points on an elliptic curve $E/\mathbb{F}_q$ is close to $q+1$. The difference between this and the actual number of $\mathbb{F}_q$-rational points is called the *trace of Frobenius*.

Definition 1.26. *Let $E/\mathbb{F}_q$ be an elliptic curve defined over $\mathbb{F}_q$. We define the trace of Frobenius of $E/\mathbb{F}_q$ be the quantity $a_q = \#E(\mathbb{F}_q) - (q+1)$.*

The next lemma gives an alternative description of the number of rational points on an elliptic curve. Unlike Hasse's theorem, it does not immediately give us any estimate of this quantity. However, we will use it in Section 1.11 to determine the number of rational points on the so called *quadratic twist* of an elliptic curve.

Lemma 1.4. *Let $E/\mathbb{F}_q\colon y^2z = f(x,z)$ be an elliptic curve over $\mathbb{F}_q$, and let $\chi\colon \mathbb{F}_q \to \{1,-1\}$ be the map*

$$\chi(x) = \begin{cases} 1 & \text{if } x \text{ is a square in } \mathbb{F}_q \\ -1 & \text{otherwise} \end{cases}$$

Then

$$\#E(\mathbb{F}_q) = \sum_{x\in\mathbb{F}_q}(1+\chi(x)) = q + \sum_{x\in\mathbb{F}_q}\chi(x)$$

Proof. Assume $P = (x_0, y_0, z_0) \in E$ and $z_0 \neq 0$. Since we are allowed to scale, we can take P to be the point $P = (x_0, y_0, 1)$. Then it is clear that $(x_0, \pm\sqrt{f(x_0, 1)}), 1) \in E(\mathbb{F}_q)$ if and only if $f(x_0, 1)$ is a square in $\mathbb{F}_q$. □

We will now turn our attention to the endomorphism ring of elliptic curves over finite fields. From Corollary 1.1, the endomorphism ring of E is of $\mathbb{Z}$-rank 1, 2 or 4. In our search for elliptic curves that are suitable for setting up a DLP, we would look for elliptic curves over finite fields with an endomorphism ring of $\mathbb{Z}$-rank 1, but the following result says that such curves do not exist:

Proposition 1.20. *Let E/K be an elliptic curve defined over a finite field K. Then $End(E)$ has $\mathbb{Z}$-rank 2 or 4.*

Proof. See the proof of Theorem V.3.1 in [31]. The theorem states that when K is a field of positive characteristic, then $End(E)$ is either an order in a quaternion algebra or an order in a quadratic imaginary field. In these cases, the $\mathbb{Z}$-rank of $End(E)$ must be 2 or 4, respectively. □

The next best would then be to find elliptic curves E over a finite field K with $\text{rank}_{\mathbb{Z}}\, End(E) = 2$. Such elliptic curves do exist, and in fact, most elliptic curves over a finite field K will have $\mathbb{Z}$-rank 2.

1.10 The Formal Group of an Elliptic Curve

The motivation for introducing formal groups is the attack on anomalous elliptic curves (see Section 3.1.1). The attack uses the formal logarithm to reduce the elliptic curve discrete logarithm problem to almost a trivial computation of an additive logarithm.

Definition 1.27. *A formal group $\mathcal{F}$ over a ring R is a power series $F(x, y) \in R[[x, y]]$ satisfying*

1. *$F(x, y) = x + y +$ higher degree terms*
2. *$F(x, F(y, z)) = F(F(x, y), z)$*

We call $F(x, y)$ the formal group law on $\mathcal{F}$.

Next we define a notion of homomorphisms and isomorphisms between formal groups:

Definition 1.28. *Let $\mathcal{F}/R$ and $\mathcal{G}/R$ be formal groups over R with formal group laws $F(x, y)$ and $G(x, y)$ in $R[[x, y]]$ respectively. A homomorphism $f\colon \mathcal{F}/R \to \mathcal{G}/R$ is a power series $f \in R[[t]]$ satisfying $f(F(x, y)) = G(f(x), f(y))$. If there also exists a homomorphism $g\colon \mathcal{G}/R \to \mathcal{F}/R$ such that $f(g(t)) = g(f(t)) = t$, then f and g are said to be isomorphisms.*

Example 1: Let R be a ring and let $F(x,y) \in R[[x,y]]$ be the power series $F(x,y) = x+y$. This is called the *additive formal group* and is denoted $\mathbb{G}_a$. It is arguably the simplest formal group, and the group operation on $\mathbb{G}_a$ coincides with addition in R.

For arbitrary $x, y \in R$, it is in general no reason to believe that the power series $F(x,y)$ will converge. However, if R is a complete local ring with maximal ideal $\mathcal{M}$ and $x, y \in \mathcal{M}$, then the series will converge. It is then readily checked that $F(x,y)$ induces a group operation on the set $\mathcal{M}$.

Definition 1.29. *Let $\mathcal{F}/R$ be a formal group where R is a complete local ring with maximal ideal $\mathcal{M}$. We denote by $\mathcal{F}(\mathcal{M})$ the group $(\mathcal{M}, +)$ where $x + y = F(x,y)$ for $x, y \in \mathcal{M}$.*

1.10.1 Formal Groups and Differential Forms

We defined the formal group law $F(x,y)$ of $\mathcal{F}/R$ in terms of the group law on E. In this section we shall introduce *formal differential forms* which we will eventually use to linearize the formal group law, and is analogous to how differentiation is used as a linearization tool in elementary calculus. Formal differential forms will play a key role in defining the *formal logarithm*.

Definition 1.30. *The space of formal differential forms over a ring R, denoted Ω_R is the R-module generated by symbols $dP(T)$ with $P(T) \in R[[T]]$ subject to the conditions:*

1. $d(P(T) + Q(T)) = dP(T) + dQ(T)$
2. $d(P(T)Q(T)) = Q(T)dP(T) + P(T)dQ(T)$
3. $da = 0$ *for all* $a \in R$.

Proposition 1.21. *Ω_R is generated by dT as an $R[[T]]$-module.*

Proof. Let $P(T) = c_0 + c_1T + c_2T^2 + c_3T^3 + \ldots \in R[[T]]$. Then $d(P(T)) = c_1dT + 2c_2TdT + 3c_3T^2dT + \ldots$ so $d(P(T)) = (c_1 + 2c_2T + 3c_3T^2 + \ldots)dT \in R[[T]]dT$. □

The proposition asserts that a formal differential form is just an expression $P(T)dT$ for some $P(T) \in R[[T]]$.

Definition 1.31. *An invariant differential on a formal group $\mathcal{F}/R$ is a differential form $w(T) = P(T)dT \in R[[T]]dT$ such that $w(F(T,S)) = w(T)$, or equivalently, $P(F(T,S))dF(T,S) = P(T)dT$.*

An invariant differential is therefore a differential form that honors the group structure of $\mathcal{F}/R$. This is in some sense similar to the result from elementary differential calculus where we have $\frac{d}{dx}(x+y) = dx$.

Definition 1.32. *We say that an invariant differential $\omega(T) = P(T)dT \in \Omega_{\mathcal{F}}/R$ is normalized if it satisfies $P(0) = 1$.*

Proposition 1.22. *Let $\mathcal{F}/R$ be a formal group. There exists a unique normalized invariant differential on $\mathcal{F}/R$ given by*

$$\omega(T) = \frac{d}{dx}F(0,T)^{-1}dT$$

where F is the power series in $R[[x,y]]$ giving the formal group law on $\mathcal{F}/R$.

Proof. See [31], Proposition IV.4.2. □

1.10.2 The Formal Logarithm

An invariant differential is a differential form that respects the group structure of $\mathcal{F}/R$, i.e a "derivative" that honors the group law. We then naturally wonder if integrating an invariant differential might yield a homomorphism from $\mathcal{F}/R$ to the additive group $\mathbb{G}_a$. It turns out that it does yield a homomorphism, but in general, it is not a homomorphism over the ring R.

Definition 1.33. *Let R be a torsion-free ring, $\mathcal{F}/R$ be a formal group, and*

$$\omega(T) = (1 + c_1T + c_2T^2 + \cdots)dT$$

be the normalized invariant differential on $\mathcal{F}$. We define the formal logarithm of $\mathcal{F}/R$ to be the power series

$$log_{\mathcal{F}}(T) = \int \omega(T) = T + \frac{c_1}{2}T^2 + \frac{c_2}{3}T^3 + \cdots \in K[[T]]$$

where $K = R \otimes \mathbb{Q}$.

Proposition 1.23. *Let R be a torsion-free ring and let $\mathcal{F}/R$ be a formal group. Then*

$$\log_{\mathcal{F}} \colon \mathcal{F} \to \hat{\mathbb{G}}_a$$

is an isomorphism of formal groups over $K = R \otimes \mathbb{Q}$.

Proof. Take $\omega(T)$ to be a normalized invariant differential on $\mathcal{F}/R$. Then $\omega(F(T,S)) = \omega(T)$, and integrating this with respect to T gives (from the definition of the formal logarithm) $\log_{\mathcal{F}} F(T,S) = \log_{\mathcal{F}}(T) + C(S)$. Now setting $T = 0$, we see that $\omega(F(0,S)) = \omega(S) = C(S)$. Then $\log_{\mathcal{F}}$ is a homomorphism, and since integration introduces denominators, it is a homomorphism over $R \otimes \mathbb{Q}$. By Lemma III.2.4 in [31], there exists a unique power series $\exp_{\mathcal{F}}(t) \in R[[t]]$ satisfying $\log_{\mathcal{F}} \circ \exp_{\mathcal{F}} = \exp_{\mathcal{F}} \circ \log_{\mathcal{F}} = t$, so $\log_{\mathcal{F}}$ is an isomorphism of formal groups. □

1.10.3 The Formal Group Associated to an Elliptic Curve

We will now define the formal group of an elliptic curve E over a field K. This will be a key tool when we describe the attack on anomalous elliptic curves in Section 3.1.1. We shall assume in this section that K is a local field complete with respect to a discrete valuation ν, and we let R denote the associated discrete valuation ring of K. For simplicity, we also assume $\mathrm{char}(K) > 3$, so E is given by an equation

$$E\colon y^2z = x^3 + axz^2 + bz^3 \tag{1.3}$$

for some constants $a, b \in R$. Note that the assumption that $a, b \in R$ can be made without loss of generality (see the discussion in 1.4 before Definition 1.15). We will study an elliptic curve E in a neighborhood of $\mathcal{O}$. When doing so, we will make the following change of variables

$$(x, y, z) \mapsto (-x, -z, y) = (s, t, u)$$

"Near" the origin we can assume $u = y$ is different from 0, so we can divide by u. In (s, t, u)- coordinates, we get $(s, t, u) = (\frac{s}{u}, \frac{t}{u}, 1)$. Near (but not at!) the origin, we can also assume that $z = -t \neq 0$, so since we are allowed to scale, we can assume $z = 1$. Then the change of variables gives a transformation $(x, y, 1) \to (-\frac{x}{y}, -\frac{1}{y}, 1)$. Now let z, w be variables, and let $z = \frac{s}{u} = -\frac{x}{y}$ and $w = -\frac{1}{u} = -\frac{1}{y}$. Then we get $x = \frac{z}{w}$ and $y = -\frac{1}{w}$. Substituting the expressions for x and y back into (1.3) gives:

$$\frac{1}{w^2} = \frac{z^3}{w^3} + a\frac{z}{w} + b \qquad \Rightarrow \qquad w = z^3 + azw^2 + bw^3$$

Now the idea is to repeatedly substitute this equation into itself and hopefully get a power series in one variable. Let $f(u, v) = z^3 + azw^2 + bw^3$. We construct a sequence of polynomials $\{f_i\}$ by letting $f_1(z, w) = f(z, w)$ and $f_{i+1}(z, w) = f_i(z, f(z, w))$.

Proposition 1.24. *Taking the limit $w(z) = \lim_{i\to\infty} f_i(z, f(z, w))$ produces a unique power series in $\mathbb{Z}[a, b]\,[[z]]$ satisfying $w(z) = f(z, w(z))$.*

Proof. We apply Hensel's lemma, which we state and prove in Lemma 3.2 in Chapter 3. Set $R = \mathbb{Z}[a, b]\,[[z]]$. Then $I = (z)$ is an ideal, complete with respect to R. Now let $f(z, w) = z^3+azw^2+bw^3$, and set $F(w) = f(z, w)-w$. Choosing $a = 0$, we have have that $F(a) \in I^3$, and $F'(a) = -1 \in R^*$. Then Hensel's lemma asserts the existence and uniqueness (since R is an integral domain) of a power series $w(z) \in R$ such that $f(z, w(z)) - w(z) = 0$, which is what we want to prove. □

Using the relation $x = \frac{z}{w}$ and $y = -\frac{1}{w}$, we can express x and y as the power series $x(z) = \frac{z}{w(z)}$ and $y(z) = -\frac{1}{w(z)}$. Then $(x(z), y(z))$ is a formal solution to the (affine) equation $y^2 = x^3 + ax + b$. Next we define the *formal group associated to E.*

Proposition 1.25. *Let z_1 and z_2 be independent indeterminates. Then there exists a formal group $\hat{E}/R$ with formal group law $F(z_1, z_2) \in \mathbb{Z}[a, b][[z_1, z_2]]$ formally giving the group law on E. We call this group the formal group associated to E.*

Proof. See [31], Chapter IV.1. □

Since K is complete with respect to the discrete valuation ν, the subring R of K is complete with respect to the maximal ideal $\mathcal{M} = \{x \in R : \nu(x) > 0\}$. Thus, if the coefficients a and b from equation (1.3) are in $\mathcal{M}$, then the power series $F(z_1, z_2)$ will converge for $z_1, z_2 \in \mathcal{M}$ and induce a group $\hat{E}(\mathcal{M})$.

Remark 1.3. *Note also that when $a, b \in \mathcal{M}$, the power series $x(z)$ and $y(z)$ will converge for $z \in \mathcal{M}$, so we get a map $\hat{E}(\mathcal{M}) \to E(K)$ defined by $z \mapsto (x(z), y(z), 1)$.*

1.11 The Quadratic Twist of an Elliptic Curve

In software implementations of elliptic curve cryptosystems, a common family of side-channel attack (see Section 3.1.4) are invalid-curve attacks. A popular counter-measure is using so called x-coordinate ladders. An x-coordinate ladder is a point multiplication algorithm that only depends on the x-coordinate of a point. In this case, the only possibility for an invalid-curve attack is on the *quadratic twist*.

Invalid-curve attacks are very easy to protect against in implementations, as it it simply a matter of verifying that a point satisfies the given curve equation. However, a paranoid curve designer devoid of any trust in the security engineers responsible for implementing the elliptic curves may take further precaution by enforcing *twist security*. A *twist secure* elliptic curve is an elliptic curve where the quadratic twist satisfies the same security requirements as the curve itself.

Definition 1.34. *Let $E/\mathbb{F}_q\colon y^2z = x^3 + axz^2 + bz^3$ be an elliptic curve. Then the quadratic twist of E, denoted E^d is the curve $E^d/\mathbb{F}_q\colon dy^2z = x^3 + axz^2 + bz^3$ where $d \in \mathbb{F}_p^*$ is a non-square.*

As implied in the definition (by the use of the singular form "the quadratic twist"), the quadratic twist of an elliptic curve is in fact unique. When proving this, we will make use of the following lemma.

Lemma 1.5. *Let $\mathbb{F}_q$ be a field of odd characteristic, and let $d_1, d_2 \in \mathbb{F}_q^*$ be quadratic non-residues. Then there exists $u \in \mathbb{F}_q^*$ such that $d_1 = u^2 d_2$.*

Proof. This follows from the fact that the (normal) subgroup $(\mathbb{F}_q^*)^2 \subseteq \mathbb{F}_q^*$ has index two in $\mathbb{F}_q^*$ when $\mathbb{F}_q$ is a field of odd characteristic. For convenience,

we set $G = \mathbb{F}_q^*$ and $H = (\mathbb{F}_q^*)^2$. Now let $\phi\colon G \to G/H$ be the canonical homomorphism. Then $\phi(x) \neq 1$ if and only if $x \notin H$. Since H has index two in G, it is clear that $G/H \cong \mu_2$. Then for $x, y \in G \setminus H$ we have $\phi(xy) = \phi(x)\phi(y) = 1$, so $xy \in H$. It follows immediately that for quadratic non-residues $d_1, d_2 \in \mathbb{F}_q^*$, we have that $d_1/d_2 = u^2$ for some $u \in \mathbb{F}_q^*$, so $d_1 = u^2 d_2$. □

Proposition 1.26. *Let $E/\mathbb{F}_q$ be an elliptic curve and assume $\mathbb{F}_q$ has odd characteristic. Let d be a non-square in $\mathbb{F}_q$. Then the quadratic twist $E^d/\mathbb{F}_q$ is unique up to isomorphism.*

Proof. Let $E\colon y^2z = x^3 + axz^2 + bz^3$ be an elliptic curve, and let $d_1, d_2 \in \mathbb{F}_q^*$ be quadratic non-residues. We assume $z \neq 0$, so we can scale and assume $z = 1$, so we have the elliptic curve $E\colon y^2 = x^3 + ax + b$. Consider the quadratic twist $E_{d_1}\colon d_1y^2 = x^3 + ax + b$. Dividing by d_1^3 and making the change of variables $x/d_1 \mapsto x$ and $y/d_1 \mapsto y$ gives

$$y^2 = x^3 + \frac{a}{d_1^2}x + \frac{b}{d_1^3} \quad \Leftrightarrow \quad d_1^3y^2 = d_1^3x^3 + ad_1x + b$$

Since d_1 and d_2 are both quadratic non-residues, we can find $u \in \mathbb{F}_q^*$ such that $d_1 = d_2u^2$ by Lemma 1.5. Substituting this into the above equation gives:

$$d_2^3u^6y^2 = d_2^3u^6x^3 + ad_2u^2x + b$$

By Proposition 1.9, this elliptic curve is the same as the elliptic curve we get when making the substitution $u^3y \mapsto y$ and $u^2x \mapsto x$, so we get

$$d_2^3y^2 = d_2^3x^3 + ad_2x + b$$

Now we make a final change of variables $d_2x \mapsto x$ and $d_2y \mapsto y$, and we get

$$d_2y^2 = x^3 + ax + b$$

which is the desired result. □

Let $E/\mathbb{F}_q$ be an elliptic curve, and suppose there does not exist a point P on E such that $x(P) = x_0$ for some $x_0 \in \mathbb{F}_q$. Then the next lemma guarantees that such a point will exist on the quadratic twist of E.

Lemma 1.6. *Let $E/\mathbb{F}_q\colon y^2z = f(x,z)$ be an elliptic curve and let $E^d/\mathbb{F}_q\colon dy^2z = f(x,z)$ be it's quadratic twist. Assume $\mathbb{F}_q$ has odd characteristic. If $f(x_0, 1)$ is a non-square in $\mathbb{F}_q$, so there does not exist a point $P \in E$ where the x-coordinate is equal to x_0, then such a point will exist on $E^d(\mathbb{F}_p)$.*

Proof. We assume $z \neq 0$ since the only point on an elliptic curve with $z = 0$ is the origin, which is readily checked to be on both E and on it's quadratic twist. Hence we can scale, and assume $z = 1$. If $f(x_0, 1)$ is a non-square in $\mathbb{F}_q$, then by Lemma 1.5 we can write $f(x_0, 1) = y_0^2 d$ for some $y_0, d \in \mathbb{F}_q^*$ where d is a quadratic non-residue. But then $(x_0, y_0, 1)$ satisfies $dy^2 z = f(x, z)$, and since d is a quadratic non-residue, this equation describes the quadratic twist of E (which is unique, by Proposition 1.26). □

The next proposition and the subsequent corollary shows that there is a strong correspondence between the number of points on an elliptic curve and its quadratic twist.

Proposition 1.27. *Let $E/\mathbb{F}_q : y^2 z = f(x, z)$ be an elliptic curve and $E^d/\mathbb{F}_q$ it's quadratic twist. Then $\#E(\mathbb{F}_q) + \#E^d(\mathbb{F}_q) = 2q + 2$.*

Proof. By Lemma 1.4, we have:

$$\begin{aligned}
\#E(\mathbb{F}_q) + \#E^d(\mathbb{F}_q) &= q + 1 + \sum_{x \in \mathbb{F}_q} \chi(f(x, 1)) + q + 1 + \sum_{x \in \mathbb{F}_q} \chi(df(x, 1)) \\
&= 2q + 2 + \sum_{x \in \mathbb{F}_q} \chi(f(x, 1)) + \chi(df(x, 1)) \\
&= 2q + 2
\end{aligned}$$

where the map χ is defined as in 1.4, and the last equality holds since $\chi(f(x, 1)) = -\chi(df(x, 1))$ when $d \in \mathbb{F}_p^*$ is a non-square. □

Corollary 1.3. *Let $E/\mathbb{F}_q$ and $E^d/\mathbb{F}_q$ be an elliptic curve and its quadratic twist. Assume $\#E(\mathbb{F}_q) = q + 1 + t$, then $\#E^d(\mathbb{F}_q) = q + 1 - t$.*

Proof. This follows immediately from the previous proposition. □

1.12 The Twisted Edwards Form of an Elliptic Curves

We previously defined an elliptic curve as a smooth curve on Weierstrass form. In cryptographic applications, the existence of efficient point addition and multiplication algorithms are essential. As it turns out, a carefully chosen representation of a curve can reduce the number of operations required to perform point addition, and thus speed up implementations.

Another reason for looking into elliptic curves on forms different from the usual Weierstrass form, is that the group law in the Weierstrass form is rather complex. Consequently, the addition formula can be difficult to get right when implementing it in cryptosystems. This can render the system vulnerable to side-channel attacks (see Section 3.1.4).

In this section we will look at the Twisted Edwards curves. In 2007, [11] introduced a family of curves known as Edwards curves, and also defined an addition law for these curves. An Edwards curve is a curve of the form $(x^2 + y^2)z^2 = c^2(z^4 + x^2y^2)$, and was shown to be birationally equivalent to a certain type of elliptic curves known as Montgomery curves. The Twisted Edwards curve is a generalization by [3] of the Edwards curve and the Edwards addition law, and it was shown that every Montgomery curve is birationally equivalent to a Twisted Edwards curve.

Definition 1.35. *A Twisted Edwards curve over a field K is a curve in $\mathbb{P}^2(\bar{K})$ given by an equation of the form*

$$E_{a,d}\colon ax^2z^2 + y^2z^2 = z^4 + dx^2y^2$$

with $a, d \in K^*$ and $a \neq d$.

Proposition 1.28 (Addition Law on Edwards Curves)**.** *The map $+\colon E_{a,d} \times E_{a,d} \to E_{a,d}$ given by $(x_1, y_1, z_1) + (x_2, y_2, z_2) \to (x_3, y_3, z_3)$, where*

$$\begin{aligned} x_3 &= \lambda(x_1y_2 + x_2y_1)(\lambda^2 - \mu) \\ y_3 &= \lambda(y_1y_2 - ax_1x_2)(\lambda^2 + \mu) \\ z_3 &= (\lambda^2 - \mu)(\lambda^2 + \mu) \end{aligned}$$

where $\lambda = z_1z_2$ and $\mu = dx_1x_2y_1y_2$ defines a group law on $E_{a,d}$. If d is a non-square, then the addition law is compelte in the sense that there are no exceptional input where the group law fails to give a correct result.

Proof. As for the Weierstrass addition law, the Twisted Edwards addition law can be verified using cumbersome but relatively straightforward algebra. Details can be found in [11] and [4].

The only way for the addition law to be undefined at some points $P = (x_1, y_1, z_1)$ and $Q = (x_2, y_2, z_2)$, is if the corresponding x_3, y_3 and z_3 (as given by the addition law) vanish simultaneously. This happens if and only if $\lambda \in \{-1, 1\}$. We will closely follow the proof of Theorem 3.3 in [7] which states that the addition law on Edwards curves is complete when d is a non-square. We will do a completely analogous proof for *Twisted* Edwards curve.

To ease notation, we will assume $z_1, z_2 \neq 0$ and scale P and Q so that we get $z_1 = z_2 = 1$. Then x_1, y_1 and x_2, y_2 satisfy the equations $ax_1^2 + y_1^2 = 1 + dx_1^2y_1^2$ and $ax_2^2 + y_2^2 = 1 + dx_2^2y_2^2$ respectively. Now we assume for contradiction that $\lambda \in \{-1, 1\}$. Then

$$\begin{aligned} dx_1^2y_1^2(ax_2^2 + y_2^2) &= dx_1^2y_1^2(1 + dx_2^2y_2^2) \\ &= dx_1^2y_1^2 + d^2x_1^2y_1^2x_2^2y_2^2 \\ &= dx_1^2y_1^2 + \lambda^2 = 1 + dx_1^2 + y_1^2 \\ &= ax_1^2 + y_1^2 \end{aligned} \tag{1.4}$$

To derive our desired contradiction, we proceed with looking at the squares $(ax_1 + \lambda y_1)^2$ and $(ax_1 - \lambda y_1)^2$ in turn.

$$\begin{aligned}(ax_1 + \lambda y_1)^2 &= a^2x_1^2 + 2a\lambda x_1y_1 + \lambda^2 y_1^2 &&= a^2x_1^2 + 2a\lambda x_1y_1 + y_1^2\\ &= dx_1^2y_1^2(a^2x_2^2 + y_2^2) + 2a\lambda x_1y_1 &&= dx_1^2y_1^2(a^2x_2^2 + y_2^2 + 2ax_2y_2)\\ &= dx_1^2y_1^2(ax_2 + y_2)^2\end{aligned}$$

where the second equality follows from (1.4). If $ax_2 + y_2 \neq 0$, then $d = (ax_1 + \lambda y_1)^2/(x_1y_1(ax_2 + y_2))^2$, which contradicts the assumption that d is a non-square. Similarly we get

$$\begin{aligned}(ax_1 - \lambda y_1)^2 &= a^2x_1^2 - 2a\lambda x_1y_1 + \lambda^2 y_1^2 &&= a^2x_1^2 - 2a\lambda x_1y_1 + y_1^2\\ &= dx_1^2y_1^2(a^2x_2^2 + y_2^2) - 2a\lambda x_1y_1 &&= dx_1^2y_1^2(a^2x_2^2 + y_2^2 - 2ax_2y_2)\\ &= dx_1^2y_1^2(ax_2 - y_2)^2\end{aligned}$$

If $ax_2 - y_2 \neq 0$, then $d = (ax_1 - \lambda y_1)^2/(x_1y_1(ax_2 - y_2))^2$, which again contradicts our assumption that d is a non-square. If $ax_2+y_2 = 0 = ax_2-y_2$, then $x_2 = y_2 = 0$, which implies that $\lambda = 0$. This is also a contradiction. □

Proposition 1.29. *Let $E_{a,d}$ be the Edwards curve defined by $ax^2z^2+y^2z^2 = z^4 + dx^2y^2$. Then the map*

$$\phi\colon \begin{cases} x \mapsto (a-d)(z+y)x \\ y \mapsto 2(a-d)(z^2+yz) \\ z \mapsto zx(z-y) \end{cases}$$

is a birational map $E_{a,d} \to E$, and where E is the elliptic curve defined by $y^2 = x^3 + 2(a+d)x^2z + (a-d)^2x$. It has an inverse $\phi^{-1}\colon E \to E_{a,d}$ defined by

$$\phi^{-1}\colon \begin{cases} x \mapsto 2x(x + (a-d)z) \\ y \mapsto (x - (a-d)z)y \\ z \mapsto y(x + (a-d)z) \end{cases}$$

Proof. This is can be proved using straightforward but tedious algebra by making the substitution defined by ϕ and plugging it in the equation defining the Edwards curve. In [3], this is verified using the Sage computer algebra system. □

Remark 1.4. *A Twisted Edwards curve is pretty close to being an elliptic curve (being birationally equivalent to one). However, it fails to be isomorphic to an elliptic curve because it is singular at the points $(1, 0, 0)$ and $(0, 1, 0)$. Blowing up the curve at these points will give an elliptic curve.*

We conclude this section by looking at an example.

Example 2: Let us look at the Edwards curve defined over the field $\mathbb{F}_{13}$ defined by:

$$E_{a,d}\colon ax^2z^2 + y^2z^2 = z^4 + 2x^2y^2$$

This curve is birationally equivalent (over some finite extension of $\mathbb{F}_{13}$) to the elliptic curve defined by:

$$E\colon y^2z = x^3 + 6x^2z + xz^2$$

Now we pick a point $Q = (12, 2, 1) \in E$, and we let $f\colon E \to E_{a,d}$ be a birational map with inverse $f^{-1}\colon E_{a,d} \to E$ such that f is defined at Q and f^{-1} is defined at $f(Q)$. The elliptic curve E and the maps f and f^{-1} can easily be computed using the *TwistedEdwardsCurve*-class that we have implemented in Section B.1.1.

We shall now compare point multiplication of Q on E with the point multiplication of $f(Q)$ on $E_{a,d}$. Benchmarking the point multiplications (see Section B.1.2) shows that the point addition on the Edwards curve is roughly 30% faster than the point addition on the elliptic curve.

Chapter 2

The Discrete Logarithm Problem

In 1976 Whitfield Diffie and Martin Hellman proposed a key exchange scheme based on the assumed hardness of the *discrete logarithm problem*. It allows two parties to exchange keys over an insecure communication channel, and today it is known today as the Diffie-Hellman key exchange scheme. Almost a decade later, Taher Elgamal described a public-key cipher called ElGamal which is also based on the assumed hardness of the discrete logarithm problem.

In this chapter we shall review the discrete logarithm problem (DLP) and the Diffie-Hellman key exchange scheme. We then consider three algorithms for solving the discrete logarithm problem. Readers already familiar with the basics of the discrete logarithm problem may want to skip to Chapter 3. We define the discrete logarithm problem:

Definition 2.1 (Discrete Logarithm Problem (DLP))**.** *Let G be a cyclic group generated by g. For $y \in G$, find $n \in \mathbb{Z}$ such that $g^n = y$.*

It is an unsolved problem whether the discrete logarithm problem can be solved in polynomial time, but it is assumed to be difficult in general. There are examples of groups where the DLP is easy. In the additive group $\mathbb{Z}/n\mathbb{Z}$ of integers modulo n, the DLP is trivial. Note that any finite abelian group is isomorphic to a direct product of groups $\mathbb{Z}/n\mathbb{Z}$, in which the DLP is trivial. Finding this isomorphism is on the other hand non-trivial, and is equivalent to solving the DLP.

2.1 Diffie-Hellman Key Exchange Scheme

The Diffie-Hellman key exchange scheme uses the discrete logarithm problem to allow two entities to securely exchange keys over an insecure communication channel. We assume Alice and Bob wants to establish a shared key

over an insecure channel where the malicious Eve is eavesdropping. The key exchange scheme works as following:

1. Alice and Bob fix the following *system parameters*; a finite cyclic group G of large order, and a generator g for G.
2. Alice generates a secret number $x \in \mathbb{N}$, and sends g^x to Bob.
3. Bob generates a secret number $y \in \mathbb{N}$, and sends g^y to Alice.
4. Alice computes $(g^y)^x = g^{xy}$.
5. Bob computes $(g^x)^y = g^{xy}$.

Now Alice and Bob have established a shared secret key $g^{xy} = g^{yx}$. Although Eve is eavesdropping the messages between Alice and Bob, Eve does not know neither Alice's private key, x, or Bob's private key, y. Despite Eve having knowledge of both g^x and g^y, she must know either x or y to compute the shared secret g^{xy}. Finding x or y is the discrete logarithm problem, and is assumed to be difficult to solve.

A weakness of the Diffie-Hellman key exchange scheme is the lack of authentication. If Eve is able to intercept and forge messages between Alice and Bob, she can perform two separate key exchanges with Alice and Bob without them knowing. Alice and Bob may then believe they have successfully made a key exchange with each other, when in reality they have both exchanged keys with Eve.

2.2 General Attacks on the DLP

Next follows a description of two general attacks on the DLP, namely the Pollard-ρ algorithm and the Pholig-Hellman algorithm. They are general in the sense that they will solve the DLP in any cyclic group. In particular, these attacks apply to the ECDLP.

2.2.1 The Pollard-ρ Algorithm

Now we will describe an algorithm due to Pollard [24]. The algorithm takes advantage of a phenomenon known as the birthday paradox in probability theory. The birthday paradox is not really a paradox, it is merely an observation that in a group of only 23 people, the probability that two or more people in the group share birthday is over 50%. [1]

Translated to our discrete logarithm problem, this means that the likelihood of having a *collision* in a small, randomly chosen subset of G, is relatively (or for some, "paradoxically") high, considering the difficulty of finding the discrete logarithm of a random element in G when $ord(G)$ is

[1] This was apparently counter to some people's intuition, so they called it a paradox.

large. By collision, we mean in this case that two or more elements have the same discrete logarithm. The next lemma shows that we can exploit such collisions to compute the discrete logarithm.

Lemma 2.1. *Let G be a cyclic group of order n generated by g. Let $y = g^m$ for some $m \in \mathbb{Z}/n\mathbb{Z}$ which we want to find. Assume we can find $s, t, u, v \in \mathbb{Z}/n\mathbb{Z}$ such $g^s y^t = g^u y^v$, and assume $gcd(v - t, n) = 1$. Then $m = \frac{s-u}{v-t}$.*

Proof. Let $g^s y^t = g^u y^v$. Then $1 = g^{s-u} y^{t-v} = g^{s-u} g^{m(t-v)} = g^{s-u-m(v-t)}$ and it follows that $s - u = m(v - t) \pmod{n}$, so $m = \frac{s-u}{v-t}$. □

We formalize our initial discussion about the birthday paradox by stating a theorem which gives us the expected number of times we must shuffle an element around before getting a collision. Before stating the theorem, we state the following definition.

Definition 2.2. *Let $f : G \to G$ be a surjective function, and let $G = A_1 \cup B \cdots \cup A_n C$ where $A_1, \cdots, A_i$ are disjoint, non-empty sets. We say that f is a shuffling function (with respect to $\{A_1, \ldots, A_n\}$) if $f(x)$ has equal probability of being in each of the n sets for every $x \in G$.*

Theorem 2.1. *Let G be a finite set, and let $f : G \to G$ be a function. Define a sequence of points $x_i = f(x_{i-1})$. Let T denote the largest integer such that x_{T-1} occurs only once in the sequence $\{x_i\}$, and let L denote the smallest integer such that $x_{T+L} = x_T$. Then*

(a) There exists an integer $i \in [1, T + L]$ with $x_i = x_{2i}$.

(b) If $f : G \to G$ is a shuffling function, then the expected value of $T + L$ is $\sqrt{\pi \#G/2}$, where $\#G$ denotes the order of G.

Proof. See Theorem XI.5.3 in [31]. □

Pollard's ρ-algorithm for computing the discrete logarithm is based on finding collisions, and then computing the discrete logarithm using Lemma 2.1. The previous theorem is thus at the very heart of Pollard's algorithm since it gives us the expected number of steps required to find the discrete logarithm.

Algorithm 2.1 (Pollard-ρ Algorithm)**.** *Let G be a cyclic group of order n generated by g, and assume we want to solve the DLP $g^m = y$ for some $y \in G$. Then the following algorithm solves the DLP in G:*

1. *Partition the set G into distjoint sets: $G = A \cup B \cup C$.*
2. *Define the function*

$$f(z) = \begin{cases} gz & \text{if } z \in A \\ z^2 & \text{if } z \in B \\ yz & \text{if } z \in C \end{cases}$$

3. *Compute the sequences* $z_i = f(z_{i-1})$, $w_i = f(f(w_{i-1}))$ *and the two sequences corresponding to* z_i*:*

$$\alpha_i = \begin{cases} \alpha_{i-1} + 1 & \text{if } z_i \in A \\ 2\alpha_{i-1} & \text{if } z_i \in B \\ \alpha_{i-1} & \text{if } z_i \in C \end{cases}, \text{ and } \beta_i = \begin{cases} \beta_{i-1} & \text{if } z_i \in A \\ 2\beta_{i-1} & \text{if } z_i \in B \\ \beta_{i-1} + 1 & \text{if } z_i \in C \end{cases}$$

until we get $z_i = w_i$*.*

Proof. By construction of the sequences $\{z_i\}$ and $\{w_i\}$, we see that $w_i = z_{2i}$. We assume that f is a shuffling function. Strictly speaking, it is not a shuffling function, but it is "close to being so" and it turns out it works pretty good in practice. By construction of f, we see that $z_i = g^{\alpha_i} y^{\beta_i}$, and by Theorem 2.1 we expect a collision after $\sqrt{\pi \#G/2}$ iterations. □

Pollard's ρ-algorithm one of the fastest publicly known algorithms for computing the discrete logarithm in arbitrary groups.

2.2.2 Pholig-Hellman Algorithm

We will now look at an algorithm commonly referred to as the Pholig-Hellman algorithm. The Pholig-Hellman algorithm is particularly efficient if the group order is smooth (that is, it factors complete into small prime numbers). The basic idea is to solve the DLP in the subgroups H_i of the group G, and use these solutions to formulate a congruence equation which can be solved using the Chinese Remainder Theorem (CRT).

Theorem 2.2. *Let* G *be a cyclic group of order* n *generated by* g*. Let* $H \subseteq G$ *be the largest subgroup of order* m*. Then the following procedure solves the discrete logarithm problem can be solved in approximately* $\mathcal{O}(\sqrt{m})$ *steps:*

1. *Let* $\#G = n = p_1 \cdots p_r$ *be a factorization of the order of* G*.*
2. *Let* $g_i = g^{n/p_i}$*, and find the discrete logarithms* $g_i^{x_i} = y$ *for each* i*.*
3. *Use the CRT to solve the congruence equations* $x_i = x \pmod{p_i}$ *modulo* $p_1 \cdots p_r = n$*.*

Proof. Let G be a cyclic group generated by g, and let $|G| = n = p_1 p_2 \cdots p_r$. For a given $y \in G$, we want to find $x \in \mathbb{Z}/n\mathbb{Z}$ such that $g^x = y$. For every prime p_i, let $g_i = g^{n/p_i}$ (note that g_i has order p_i). If we can find $x \in \mathbb{Z}/n\mathbb{Z}$ such that $g^x = y$, then clearly we have that $g_i^{x_i} = y$ where $x_i = x \pmod{p_i}$ for all i. Thus we can formulate a system of congruences given by $x_i = x \pmod{p_i}$. By the CRT, there is a unique $n \in \mathbb{Z}/n\mathbb{Z}$ such that $x = x_1 x_2 \cdots x_r \in \mathbb{Z}/n\mathbb{Z}$. This solves the DLP in G. □

2.2.3 Small Subgroup Confinement Attack

Let G be a large group of non-prime order n. A small subgroup confinement attack is when an attacker is able to confine a DLP to a small subgroup in which the problem can be solved, either by the use of an efficient algorithm or by an exhaustive search. There are various ways, depending on the circumstances, in which an attacker can confine a DLP to a small subgroup. We consider the attack on the Diffie-Hellman key exchange scheme proposed by [18].

Assume Alice and Eve is doing a Diffie-Hellman key exchange and that they use the encryption function $E_K(m)$ to encrypt a message m with a key K. If Alice does not do proper checking, [18] demonstrated that Eve can do the following procedure to reveal information about Alice's private key:

1. Alice generates a public key $y_A = g^{x_A}$ and sends it to Eve.
2. Eve generates a public key $y_E = \beta g^{x_E}$ and where β is an element of G of small order
3. Alice computes the session key $K = y_E^{x_A} = \beta^{x_A} g^{x_E x_A}$ and sends a message $c = E_K(m)$ to Eve encrypted with the key K.
4. Eve exhaustively search for x_{β_i} such that $E_{S_i}(m) = c$ which can be done in $ord(\beta)$ steps.

When Eve finds x_{β_i} such that $E_{S_i}(m) = c$, she has revealed x_A modulo $ord(\beta)$. Eve can then repeat this process several times with a different β each time, and formulate a system of congrurence equations which can be solved using the Chinese Remainder Theorem.

2.3 The Index Calculus Algorithm

Now we will briefely describe the main steps involed in the Index Calculus algorithm. A thorough description and analysis of the Index Calculus algorithm is beyond the scope of this thesis, so our description will be brief, and is only meant to give the reader an idea of how the algorithm works. An in-depth description can be found in [14].

Assume we have a DLP in the group $G = \mathbb{F}_p^*$ with g a generator. Then $g^x = y$ for some $g, y \in \mathbb{F}_p$ and $x \in \mathbb{N}$. In the index calculus algorithm, one starts with selecting a *factor base*. The factor base is a relatively small subset of $\mathbb{F}_p^*$ for which we will find relations that we will hopefully be able to use to solve our DLP.

Algorithm 2.2. *The following probabilistic algorithm solves the DLP in approximately* $\mathcal{O}(e^{c\sqrt[3]{(\log q)(\log\log q)^2}})$ *steps, where* c *is a small constant [14]:*

1. *Choose an index base* $\mathcal{B} = p_1, \cdots, p_n$ *consisting of primes smaller than some given bound* b.

2. *Pick a random* $\alpha \in \mathbb{Z}/(p-1)\mathbb{Z}$ *and compute* g^α. *If* g^α *factors completely in* $\mathcal{B}$ *we find the factorization* $g^\alpha = p_1^{m_1} \cdots p_n^{m_n}$.
3. *We get the relation* $\alpha = m_1 \log_g(p_1) + \cdots + m_n \log_g(p_n) \pmod{p-1}$.
4. *Continue until we get* m *linearly independent relations. Then we can find* $log_g(p_i)$ *for each* i *using basic linear algebra.*
5. *Find* $\xi \in \mathbb{Z}/(p-1)\mathbb{Z}$ *such that* $g^\xi y$ *factors completely in* $\mathcal{B}$.
6. *Then we factor* $g^\xi y = p_1^{r_1} \cdots p_n^{r_n}$, *and taking logarithms yields* $\xi + log_g(y) = r_1 \log_g(p_1) + \ldots + r_n \log_g(p_n)$. *Then we solve the DLP by computing* $\log_g(y) = r_1 \log_g(p_1) + \ldots + r_n \log_g(p_n) - \xi$.

When implementing the Index Calculus algorithm, a trade-off must be made between the size of the factor base and the probability that you can find ξ such that $g^\xi y$ factors completely in β. Intensive research has been conducted on this trade-off, but it is beyond the scope of this thesis.

Chapter 3

Elliptic Curves in Cryptography

The security of the cryptosystems which we briefly mentioned in the previous chapter is completely determined by the hardness of the DLP. The group $\mathbb{F}_p^*$ of units modulo p, has been a popular choice of group for setting up a DLP due to its ease of implementation. However, because of the Index Calculus algorithm, relatively large keys are required to achieve an acceptable level of security. In many cases, for example on modern personal computers, this need not be problematic since memory and computational power is vast. On embedded devices, where bandwidth and/or computational power may be limited, it can be a major problem.

This has led researchers to consider alternative groups for setting up the DLP in the hope of finding groups where Index Calculus-like algorithms do not exist. This would allow for smaller keys for the same level of security. The group of points on an elliptic curve is believed to be precisely this, and was independently suggested for use in cryptography by Neal Koblitz and Victor S. Miller in 1985. Since 2005, it has seen widespread use in cryptographic software. Now we define the elliptic curve discrete logarithm problem.

Definition 3.1 (The Ellipic Curve Discrete Logarithm Problem)**.** *Let E/K be an elliptic curve defined over K. Let $P \in E(K)$, and let $Q \in \langle P \rangle$ (i.e Q is in the subgroup generated by all multiples of P). The discrete logarithm problem is to find m such that $Q = [m]P$.*

As with the discrete logarithm problem, the elliptic curve discrete logarithm problem (ECDLP) has been the subject of extensive research over the past decades. In 2009, a decentralized digital currency named Bitcoin was released as open-source software. At the time of writing, the total market value of all bitcoins is an estimated 3.5 billion USD. The security of Bitcoin relies on the assumed hardness of the ECDLP, so anyone capable of solving

the ECDLP would essentially also be capable of stealing 3.5 billion USD worth of bitcoins. Solving the ECDLP would in other words allow for one of the greatest heist in history to take place. Hence, there is certainly not a lack of incentives for solving the ECDLP.

In this chapter we will begin with describing attacks on the elliptic curve discrete logarithm problem (ECDLP). Then we propose a set of *security requirements* that elliptic curves should satisfy in order to be secure against these attacks. In the end of this section, we will consider *technical requirements* that the elliptic curves should satisfy in order to facilitate efficient implementations on a computer.

3.1 Attacking the Elliptic Curve Discrete Logarithm Problem

Although there are no publicly known sub-exponential algorithms for solving the ECDLP for arbitrary elliptic curves, there are some classes of elliptic curves in which the ECDLP can be solved efficiently. For an elliptic curve E to be suitable for use in cryptography, we require E to satisfy certain conditions to ensure that these attacks either do not apply, or are inefficient.

In the next sections, we will give a mathematical description of known attacks on the ECDLP with the intention of deducing security requirements that E must satisfy in order to be suitable for use in cryptography. Our main focus will be on the attacks that affect the ECDLP directly. Attacks that exploit weak implementations (commonly referred to as *side-channel attacks*), will be covered in less detail. In the end of this section, we will consider a recently proposed attack on the ECDLP and a corresponding security requirement made by [33].

3.1.1 The Anomalous Attack

Consider an elliptic curve $E/\mathbb{F}_p$ satisfying $\#E(\mathbb{F}_p) = p$. In other words, these are elliptic curves over $\mathbb{F}_p$ where the trace of Frobenius is equal to 1. Such elliptic curves are called *anomalous*, and an attack on anomalous elliptic curves was proposed independently by Smart [32], Semaev [28] and Satoh-Araki [26]. For this reason, the attack is sometimes called the Smart-ASS attack. The attack on anomalous elliptic curves is based on moving the ECDLP to a formal group. Once we are in a formal group, we can use the formal logarithm to almost trivially compute the discrete logarithm.

Lemma 3.1. *Let K be a field complete with respect to a discrete valuation ν. Suppose E/K is an elliptic curve over a field K, and assume E has good reduction at ν. Let E_1 denote the kernel of the reduction map π_ν, and let $\hat{E}(\mathcal{M})$ denote the formal group associated to E/K. Then the map*

$E_1(K) \to \hat{E}(\mathcal{M})$ given by $\mathcal{O} \mapsto 0$, $(x, y, 1) \mapsto -\frac{x}{y}$ is an injective group homomorphism.

Proof. Assume E/K is given by a minimal Weierstrass equation. Now assume a point $P = (x, y, 1) \in E_1(K)$. Since P is reduced to $(0, 1, 0)$, we must have that $\nu(x) < 0$ or $\nu(y) < 0$. From the curve equation and since ν is a discrete valuation, we have that $2\nu(y) = 3\nu(x)$, so $\nu(x) < 0$ if and only if $\nu(y) < 0$. Moreover, since $2\nu(y) = 3\nu(x)$ and 2 and 3 are co-prime, we must have that $2\nu(y) = 3\nu(x) = -6r$ for some $r \in \mathbb{N}$. This gives $\nu(x) = -2r$ and $\nu(y) = -3r$, and it follows that $\frac{x}{y} \in \mathcal{M}$. Consequently, the map $E_1(K) \to \hat{E}(\mathcal{M})$ as given above is well defined. It is a homomorphism since the group law on $\hat{E}(\mathcal{M})$ is defined from the group law on E, and it is clearly injective (it admits an inverse $-x/y \mapsto (x, y, 1)$). □

Remark 3.1. *One can show that the map given in the preceding lemma is actually an isomorphism of groups.*

Let K be a field complete with respect to a discrete valuation ν, and let R be the discrete valuation ring associated to ν. Suppose $\mathfrak{m} \subseteq R$ is the maximal ideal in R, and that $R/\mathfrak{m} \cong \mathbb{F}_p$. If E/K and $\tilde{E}/\mathbb{F}_p$ are elliptic curves such that $\pi_\nu(E) = \tilde{E}$, we say that E is a *lift* of $\tilde{E}$ modulo $\mathfrak{m}$. Similary, if $P \in E(K)$ is a point such that $\pi_\nu(P) = \tilde{P}$, we say that P is a lift of $\tilde{P}$ (modulo $\mathfrak{m}$).

Now for points $\tilde{P}, \tilde{Q} \in \tilde{E}(\mathbb{F}_p)$ satisfying $\tilde{Q} = [m]\tilde{P}$, the idea of the anomalous attack is to lift $\tilde{P}, \tilde{Q}$ to points P, Q on a lifted curve E/K while maintaining the relation $Q = [m]P$. If this can be accomplished, then we can usually "move" the points into the formal group where we can recover m by applying the formal logarithm.

Lifting $\tilde{E}$ and the points $\tilde{P}, \tilde{Q} \in E(\mathbb{F}_p)$ is easy, and can be done using Hensel's lemma, which we will state shortly. Maintaining the relation that $Q = [m]P$ is however non-trivial. In fact, a key factor in the attack on anomalous elliptic curves is that this relation is automatically preserved when lifting $\tilde{P}$ and $\tilde{Q}$.

Lemma 3.2 (Hensel's Lemma)**.** *Let R be a ring that is complete with respect to some ideal $I \subseteq R$, and let $F(w) \in R[w]$ be a polynomial. Suppose that there is an integer $n \geq 1$ and an element $a \in R$ satisfying*

$$F(a) \in I^n \qquad \text{and} \qquad F'(a) \in R^*$$

Then for any $\alpha \in R$ satisfying $a \equiv F'(a) \pmod{I}$, the sequence

$$w_0 = a, \qquad w_{m+1} = w_m - \frac{F(w_m)}{\alpha}$$

converges to an element $b \in R$ satisfying

$$F(b) = 0 \qquad \text{and} \qquad b = a \pmod{I^n}$$

If R is an integral domain, then these conditions determine b uniquely.

Proof. In this proof, we follow the exposition given in [31]. We will consider the sequence $\{x_n\} = \{w_n - a\}$ by replacing $F(w_n)$ with $F(x_n + a)/\alpha$. Then we get the recurrence

$$x_{m+1} = x_m - F(x_m)$$

with $x_0 = 0$, $F(0) \in I^n$ and $F'(0) = 1 \pmod{I}$. Now we want to show that $x_{m+1} = x_m \pmod{I^{m+n}}$ for all $m \geq 0$. First, we observe that $x_m \in I^n$ implies that $x_m - F(x_m) \in I^n$. This is true because if $F(x_0) = F(0) \in I^n$, then the constant term of the polynomial F must be in I^n, which implies that $F(x_m) \in I^n$ when $x_m \in I^n$ (since all the terms of $F(x_m)$ is in I^n).

Now we show by induction that $x_m = x_{m+1} \pmod{I^{m+n}}$ for all $m \geq 0$. Assume $x_m = x_{m+1} \pmod{I^{m+n}}$ holds for all $m < n$. For variables x and y, the expression $F(x) - F(y)$ will not have any constant terms, so we can factor

$$F(x) - F(y) = (x - y)(F'(0) + xG(x, y) + yH(x, y)) \tag{3.1}$$

where $G, H \in R[x, y]$. Now we use the factorization

$$\begin{aligned} x_{m+1} - x_m &= (x_m - F(x_m)) - (x_{m-1} - F(x_{m-1})) \\ &= (x_m - x_{m-1}) - (F(x_m) - F(x_{m-1})) \\ &= (x_m - x_{m-1}) - ((x_m - x_{m-1})(F'(0) \\ &\qquad + x_m G(x_m, x_{m-1}) + x_{m-1} H(x_m, x_{m-1})) \\ &= (x_m - x_{m-1})(1 - F'(0) - x_m G(x_m, x_{m-1}) \\ &\qquad - x_{m-1} H(x_m, x_{m-1})) \end{aligned}$$

By the induction hypothesis $x_m - x_{m-1} \in I^{n+m-1}$. Since $F'(0) = 1 \pmod{I}$, we have $1 - F'(0) \in I$, and since $x_m, x_{m-1} \in I^n \subseteq I$, we have that $x_m G(x_m, x_{m-1}) \in I$ and $x_{m-1} H(x_m, x_{m-1}) \in I$. It follows that the product $(x_m - x_{m-1})(1 - F'(0) - x_m G(x_m, x_{m-1}) - x_{m-1} H(x_m, x_{m-1})) = x_{m+1} - x_m \in I^{m+n}$ which is what we wanted to show.

By assumption, R is complete with respect to I, so the sequence $\{x_m\}$ converges to an element $b \in I^n$ since $w_m \in I^n$ for all $m \geq 0$. Taking the limit as $m \to \infty$ on both sides of $x_{m+1} = x_m - F(x_m)$ gives us the relation $b = b - F(b)$, so $F(b) = 0$.

Now we prove uniqueness of b under the assumption that R is an integral domain. Assume there exists $c \in I^n$ with $F(c) = 0$. Using the factorization (3.1), we get $F(b) - F(c) = 0 = (b - c)(F'(0) + bG(b, c) + cH(b, c))$. If $b \neq c$, then (now we use the assumption that R is an integral domain) $F'(0) = -bG(b, c) - cH(b, c) \in I$ which contradicts $F'(0) = 1 \pmod{I}$. Hence we must have that $b = c$. □

Hensel's lemma is sometimes also called Hensel's lifting lemma, and not surprisingly, liftings are precisely what we shall use Hensel's lemma for. Let

$\tilde{E}/\mathbb{F}_p\colon y^2z = x^3+axz^2+bz^3$ and assume $\tilde{P} \in \tilde{E}(\mathbb{F}_p)$. Then $\tilde{P}$ is merely a root of the polynomial $f(x,y,z) = y^2z - x^3 - axz^2 + bz^3$ modulo p, i.e $f(\tilde{P}) = 0 \pmod p$, so $f(\tilde{P}) \in (p)$. Now we can canonically lift the elliptic curve $\tilde{E}$ to an elliptic curve $E'/\mathbb{Z}_p\colon y^2z = x^3 + axz^2 + bz^3$ simply by considering the coefficients of $\tilde{E}$ as elements in $\mathbb{Z}_p$. The local ring $\mathbb{Z}_p$ is complete with respect to the p-adic valuation, and has the maximal ideal $\mathfrak{p} = (p)$. It is then easy to see that $E'/\mathbb{Z}_p$ reduces to $\tilde{E}/\mathbb{F}_p$ modulo $\mathfrak{p}$.

Since the ring $\mathbb{Z}_p$ is a complete local ring with maximal ideal $\mathfrak{p} = (p)$, we can use Hensel's lemma to find a point $P' \in E'/\mathbb{Z}_p$ with $f(P) = 0$ which also satisfies $P' \pmod{\mathfrak{p}} = \tilde{P}$. Hence, we can use Hensel's lemma to "lift" the point $\tilde{P} \in \tilde{E}/\mathbb{F}_p$ to a point $P' \in E'/\mathbb{Z}_p$ which reduces to $\tilde{P}$ modulo $\mathfrak{p}$. Just as we canonically lifted the elliptic curve $\tilde{E}$ and the point $\tilde{P}$ to $\mathbb{Z}_p$, $E'/\mathbb{Z}_p$ and $P' \in E'(\mathbb{Z}_p)$ can both be canonically lifted to an elliptic curve $E/\mathbb{Q}_p$ and a point $P \in E(\mathbb{Q}_p)$. This is exactly what we are going to do in the following attack on the ECDLP for anomalous elliptic curves.

Theorem 3.1 (Anomalous Attack)**.** *Let $\tilde{E}$ be an elliptic curve defined over $\mathbb{F}_p$ with $\#E(\mathbb{F}_p) = p$. Let $\tilde{P}, \tilde{Q} \in \tilde{E}$ and $\tilde{Q} = [m]\tilde{P}$ for some $m \in \mathbb{N}$. Then there exists a linear time algorithm for solving the ECDLP.*

Proof. Use Hensel's lemma to lift the points $\tilde{P}, \tilde{Q} \in \tilde{E}(\mathbb{F}_p)$ to points $P, Q \in E(\mathbb{Q}_p)$. There is no reason to believe that the lifted points P and Q will satisfy the relation $Q = [m]P$ (in fact, one can show that only one of the lifts will satisfy this relation). Let

$$R = Q - [m]P \tag{3.2}$$

Clearly $R \in E_1(\mathbb{Q}_p)$ so by Lemma 3.1, it is in a formal group associated to E. The points Q and P are not in the formal group, but $[p]Q$ and $[p]P$ are, since $\#E(\mathbb{F}_p) = p$ and the reduction map is a group homomorphism. Taking the multiplication-by-p map on both sides of (3.2) gives

$$[p]R = [p](Q - [m]P) = [p]Q - [m][p]P$$

Now all the terms in this equation is in the formal group. In this formal group, we have the formal logarithm map which induces a map on the formal group associated to E:

$$\log_E\colon E(p\mathbb{Z}_p) \to p\mathbb{Z}_p$$

Since $R \in E_1(\mathbb{Q}_p)$ we have $\log_E(R) \in p\mathbb{Z}_p$. But then $\log_E([p]R) = plog_E(R) \in p^2\mathbb{Z}_p$. Consequently, $\log_E([p]Q) - m\log_E([p]P) = 0 \pmod{p^2}$, so $m = \frac{\log_E([p]Q)}{\log_E([p]P)} \pmod p$. □

Assuming that we have an efficient way of lifting points and computing the formal logarithm, we can solve the ECDLP for anomalous elliptic curves

over $\mathbb{F}_p$. In the proof of the theorem, we saw that we only need to lift modulo p^2, and since we only need to know m modulo p, it is also easy to compute the formal logarithm (we compute it modulo p^2, and reduce).

Remark 3.2. *It is tempting to try to generalize this attack to any curve satisfying $ord(P) = q$. By the theorem of Lagrange, $\#E(\mathbb{F}_q)$ must be divisible by p, but by Hasse's theorem, $\#E(\mathbb{F}_q) \in [q + 1 - 2\sqrt{q}, q + 1 + 2\sqrt{q}]$, so this is impossible. Thus $ord(P) = q \Rightarrow \#E(\mathbb{F}_q) = q$.*

3.1.2 The MOV Attack

For an elliptic curve E defined over a finite field $\mathbb{F}_p$ and points $P \in E$ and $Q \in \langle P \rangle$, solving the discrete logarithm problem on E is to find $m \in \mathbb{Z}$ such that $[m]P = Q$. Recall that the Weil e_N-pairing is a bilinear, alternating, non-degenerate and Galois invariant pairing $E[N] \times E[N] \to \mu_N$.

Assume $P \in E(\mathbb{F}_p)[N]$ with $P \neq \mathcal{O}$. Then by Proposition 1.6, the $\mathbb{Z}$-rank of $E[N]$ is either 1 or 2. If $\operatorname{rank}_{\mathbb{Z}}(E[N]) = 1$, then the next proposition shows that the elliptic curve is anomalous, and we can use the technique described in Section 3.1.1 to solve the ECDLP in linear time.

Proposition 3.1. *Assume $rank_{\mathbb{Z}}(E[N]) = 1$ and let $P \in E(\mathbb{F}_p)[N]$ and $P \neq \mathcal{O}$. Then $N = \#E(\mathbb{F}_p) = p$.*

Proof. By Proposition 1.6, P must generate a $\mathbb{F}_p$-rational subgroup of order N where p divides N. Then since $p + 1 - 2\sqrt{p} \leq \#E(\mathbb{F}_p) \leq p + 1 + 2\sqrt{p}$ by the theorem of Hasse we must have that $N = p$ since the order of P must divide the order of $\#E(\mathbb{F}_p)$. By a similar argument (or by Remark 3.2), we get $\#E(\mathbb{F}_p) = N = p$. □

If $\operatorname{rank}_{\mathbb{Z}}(E[N]) = 2$, then we can find linearly independent points on E. This is necessary if we want to make use of the Weil e_N-pairing, since the pairing is trivial on elements in the same cyclic subgroup (see Remark 1.1).

Theorem 3.2 (The MOV Attack)**.** *Let $P \in E(\mathbb{F}_q)[N]$ and $T \in E[N]$ be linearly independent points on E. Then the ECDLP $Q = [m]P$ can be reduced to the DLP*

$$e_N(Q, T) = e_N([m]P, T) = e_N(P, T)^m \subseteq \mathbb{F}_{q^d}$$

in some finite extension $\mathbb{F}_{q^d}$ of $\mathbb{F}_q$ of degree d.

Proof. In general, we have that $e_N(Q, T) \in \bar{F}_q^*$ since the rational function we used when defining the Weil e_m-pairing is in $\bar{K}(E)$. However, since $\#E[N]$ is finite, the image of e_N is contained in a finite field extension of $\mathbb{F}_q$. In particular, the image of e_N under the set where T is fixed and Q ranges over $E[N]$ is contained in some finite field extension $\mathbb{F}_{p^d}$ of $\mathbb{F}_p$ of degree d.

Now we will show that $e_N(P,T)$ is a primitive N-th root of unity. Assume the contrary. Then $e_N(P,T)$ generates a subgroup of some order r. By the bilinearity of the Weil e_N-pairing, this implies that $e_N(P,T)^r = e_N(P,[r]T) = 1$ for all $T \in E[N]$. By the non-degeneracy of the Weil e_N-pairing, this implies that $[r]T = \mathcal{O}$. Since T is chosen arbitrarily in $E[N]$, we must have $r = N$. Hence $e_N(P,T)$ is an N-th root of unity, and we have reduced the ECDLP on E to the DLP $e_N(P,T) = e_N(P,T)^m$ in the field $\mathbb{F}_{q^d}$. □

For the MOV-attack we are particularly interested in cases where the degree d of the field extension of $\mathbb{F}_q$ is low. For example, if $E[N] \subseteq E(\mathbb{F}_q)$, then by Remark 1.2 it follows immediately that $d = 1$. As we will see in the Section 3.1.5, d will usually be large, and so the reduced DLP will typically be much harder than the original ECDLP. Now we shall give an explicit expression for d, but first it is convenient to state the following definition.

Definition 3.2. *Let G be a finite cyclic group. We define the embedding degree of G in $\mathbb{F}_q$ to be the smallest integer d such that $\mathbb{F}_{q^d}^*$ contains a subgroup isomorphic to G.*

Proposition 3.2. *Let G be a cyclic group of order N, and let d be the embedding degree of G in the finite field $\mathbb{F}_q$. Then $d = ord(q) \pmod N$.*

Proof. The group $\mathbb{F}_{q^d}^*$ has order $q^d - 1$. Since it is a finite abelian group, then if $N \mid q^d - 1$ implies that $\mathbb{F}_{q^d}^*$ has a subgroup of order N. Hence, the embedding degree of G in F_q is the smallest integer d such that N divides $q^d - 1$. Then $q^d - 1 = 0 \pmod N \Leftrightarrow q^d = 1 \pmod N$, so $d = ord(q)$. Since all finite cyclic groups of equal order are isomorphic, the subgroup of order N is isomorphic to G. Furthermore, d divides $q^d - 1$ by the little theorem of Fermat. □

The MOV attack as we described it requires the computation of a linearly independent point $T \in E[N]$. This point may very well not be $\mathbb{F}_q$-rational (for example, if $\#E(\mathbb{F}_q)$ is prime, it most certainly is not), and it may not even be $\mathbb{F}_{q^d}$-rational. Hence, we risk having to work over a field extension of $\mathbb{F}_q$ of high degree. The next result shows that adding the requirement that $gcd(q-1, N) = 1$ ensures that all torsion points are in a manageable extension field of $\mathbb{F}_q$.

Proposition 3.3. *Let $E/\mathbb{F}_q$ be an elliptic curve defined over $\mathbb{F}_q$. Assume $gcd(q-1, N) = 1$, and that $N \neq 0$ in $\mathbb{F}_q$. Let d be the embedding degree of μ_N in $\mathbb{F}_q$. Then $E[N] \subseteq E(\mathbb{F}_{q^d})$*

Proof. Let $P \in E(\mathbb{F}_q)$ be of exact order N, and choose $T \in E[N]$ such that P and T are linearly independent, so $\{P, T\}$ is a basis for $E[N]$. Let $\phi \in Gal(\overline{\mathbb{F}}_q/\mathbb{F}_q)$ be the q-power Frobenius map (which we defined in Section

1.3). We recall that $Gal(\bar{\mathbb{F}}_q/\mathbb{F}_q)$ acts on a point in $E(\mathbb{F}_q)$ by acting on its coordinates, and that $P \in E(\mathbb{F}_q)$ if and only if $P^\phi = P$. Since P and T generate $E[N]$, we have $T^\phi = [a]P + [b]T$ for some $a, b \in \mathbb{Z}/N\mathbb{Z}$. From the properties of the Weil e_N-pairing, we get that

$$\begin{aligned} e_N(P,T)^q &= e_N(P,T)^\phi = e_N(P^\phi, T^\phi) \\ &= e_N(P, [a]P + [b]T) \\ &= e_N(P,[a]P)e_N(P,[b]T) \\ &= e_N(P,T)^b \end{aligned}$$

Since P and T are linearly independent, $e_N(P,T)$ is a primitive N-th root of unity. Then $e_N(P,T)^q = e_N(P,T)^b$ implies that $q = b \pmod N$. Now we repeatedly apply ϕ to the point T, and we keep in mind that $b = q \pmod N$. This gives

$$\begin{aligned} T^\phi &= [a]P + [q]T \\ T^{\phi^2} &= [a]P + [q]([a]P + [q]T) = [a + aq]P + [q^2]T \\ T^{\phi^3} &= [a]P + [q]([a]P + [q]([a]P + [q]T) = [a + aq + aq^2]P + [q^3]T \\ &\vdots \\ T^{\phi^d} &= [a(1 + q + q^2 + \ldots + q^{d-1})]P + [q^d]T \end{aligned}$$

Since $q^d = 1 \pmod N$ by assumption, and $gcd(q-1, N) = 1$, we have $\frac{q^d-1}{q-1} = 0 \pmod N$. It is easily proved using induction on d that $\frac{q^d-1}{q-1} = 1+q+q^2+\ldots+q^{d-1}$. Then we must have $1+q+q^2+\ldots+q^d = 0 \pmod N$. Since $E[N] \cong \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$, every element in $E[N]$ has order at most N. It follows that $T^{\phi^d} = [0]P + [1]T = T$, so T is left fixed by the q^d-power Frobenius endomorphism. This happens if and only if $T \in E(\mathbb{F}_{q^d})$, and since T was chosen arbitrarily, it is immediate that $E[N] \subseteq E(\mathbb{F}_{q^d})$. □

In cases where $gcd(q-1, N) \neq 1$, an alternative to searching for a point T linearly independent of P, is using a pairing that returns a primitive N-th root of unity even for points where there is a linear dependence. The Tate-Lichtenbaum pairing is an example of such a pairing, and it was suggested used by Frey and Rück in [12]. Another advantage of the Tate-Lichtenbaum pairing is that it only requires the evaluation of one rational function (the Weil-pairing requires the evaluation of two rational functions), so it is twice as fast as the Weil-pairing. For this reason, the Tate-Lichtenbaum pairing is usually preferred in software implementations.

For a curve to be resistant against the MOV-attack, we need the reduced DLP in $\mathbb{F}_{p^d}$ to be at least as hard as solving the ECDLP. The best known algorithms for solving the DLP in an arbitrary group G of order n can compute the discrete logarithm in approximately $O(2^{n_b/2})$ steps, where n_b

is the number of bits needed to represent n. The Index Calculus algorithm solves the DLP in $\mathbb{F}_{p^d}$ in approximately $\mathcal{O}(e^{c\sqrt[3]{(\log p^d)(\log\log p^d)^2}})$ steps. A rough estimate then tells us that the embedding degree d of μ_N in $\mathbb{F}_p$ should at least be greater than xx to prevent the reduced DLP from being easier than the ECDLP.

3.1.3 Lifting Attacks and the Class Number Condition

The Index Calculus algorithm is in essence based on lifting a finite number field to the ring of integers, $\mathbb{Z}$, establishing a set of relations there, and then using these relations to deduce the discrete logarithm. It is natural to attempt a similar strategy when attacking the ECDLP, and this approach turns out to be somewhat fruitful; if we can lift the points $\tilde{P}, \tilde{Q} \in \tilde{E}(\mathbb{F}_q)$ where $\tilde{Q} = [m]\tilde{P}$ to points $P, Q \in E(K)$ where K is an algebraic number field and the relation $Q = [m]P$ is preserved, then it is relatively easy to find m.

Example 3: Let $E/\mathbb{Q} : y^2 = x^3 - 3x - 1$ be an elliptic curve over $\mathbb{Q}$. Let $P, Q \in E(\mathbb{Q})$ be non-torsion points with $Q = [m]P$ for some m. Let P_p and Q_p denote the reduction of P and Q respectively at a prime p. We can solve the ECDLP by solving the reduced ECDLP $Q_{p_i} = m_i P_{p_i}$ for many i. Then we will have $m_i = m \pmod{ord(P_{p_i})}$, and we apply the Chinese Remainder Theorem to find m modulo $\prod_i ord(P_{p_i})$. We consider the discrete logarithm problem above with the given P and Q:

$$P = (2, 1, 1),$$
$$Q = (96730 \cdots 13056, -93868 \cdots 51296, 1)^1$$

Reducing the points Q and P modulo 5 gives two points $P_5 = (2, 1, 1)$ and $Q_5 = (4, 4, 1)$. The order of P_5 is 7 and an exhaustive search reveals that $4P_5 = Q_5$. We continue this procedure with primes $7, 11, 13$ and organize the results in the following table, where P_p and Q_p again denotes the reduction of P and Q respectively modulo p.

Prime	P_p	Q_p	$ord(P_p)$	m (mod $ord(P_p)$)
5	(2, 1, 1)	(4, 4, 1)	7	4
7	(2, 1, 1)	(4, 4, 1)	11	2
11	(2, 1, 1)	(10, 10, 1)	7	4
13	(2, 1, 1)	(12, 12, 1)	19	10

Using the Chinese Remainder Theorem we combine the residues $4, 2, 4, 10$ and the moduli $7, 11, 7, 19$ and obtain $m = 200$ which is easily verified to be the correct answer.

[1] 36728 and 55093 digits are needed for a base-10 representation of the x- and y-coordinate of Q respectively, so for the convenience of everyone, all but the 5 leading and trailing digits of the x- and y-coordinate of Q have been omitted.

When lifting the points $\tilde{P}, \tilde{Q} \in \tilde{E}(\mathbb{F}_q)$, we have the choice between lifting to torsion points or non-torsion points. The next theorem, which is due to Serre, shows that lifting the points $\tilde{P}$ and $\tilde{Q}$ to torsion points $P, Q \in E(K)$ appears unfeasible since it requires the degree of the algebraic number field K over the rationals to be large, thus making it difficult to efficiently represent field elements on a computer.

Theorem 3.3 (Serre)**.** *Let $E/\mathbb{Q}$ be an elliptic curve, and let K be an algebraic number field. Assume $E(K)$ has a torsion point of exact order n. Then there exists a constant $c > 0$ such that $[K : \mathbb{Q}] \geq c\#GL_2(\mathbb{Z}/n\mathbb{Z}) = cn^4$.*

Proof. See Chapter IV in [29] for a proof. □

Example 4: Assume we want to set up a ECDLP, and that we require *128-bits security*. In other words, we want the ECDLP to require approximately 2^{128} steps to solve when using the best algorithms known. Assuming that the elliptic curve is not vulnerable to known attacks on the ECDLP, the fastest algorithm for solving the ECDLP have a complexity of of $O(\sqrt{n})$ where n is the group order. To protect against small subgroup attacks (see Section 2.2.3), we choose an elliptic curve E such that there exists a point $\tilde{P} \in \tilde{E}(\mathbb{F}_q)$ of prime order and $ord(\tilde{P}) \approx 2^{256}$. Then by Theorem 3.3, lifting $\tilde{P}$ and $\tilde{Q} = [m]\tilde{P}$ to torsion points $P, Q \in E(K)$ over an algebraic number field K, would require the degree of K over the rationals to satisfy $[K : \mathbb{Q}] \geq c \cdot 2^{1024}$ for some constant $c > 0$. This makes it unlikely that lifting the points $\tilde{P}$ and $\tilde{Q}$ to torsion points over an algebraic number field will lead to a practical attack on the ECDLP.

One might then hope to lift $\tilde{P}, \tilde{Q} \in \tilde{E}(\mathbb{F}_q)$ to non-torsion points on a curve E/K, where the degree of K over the rationals is relatively small. We shall see that in this case, we run into two separate problems. First, it is difficult to lift to non-torsion points while preserving the relation $Q = [m]P$. Second, the class number of K (which we shall define shortly) imposes a lower bound on the degree of K over the rational which again makes it difficult to even represent field elements on a computer unless the class number is reasonably small. We will proceed with defining the class number of K, but prior to this we need a couple of definitions:

Definition 3.3. *Let K be an algebraic number field, and let $R \subseteq K$ be its ring of integers. We define a fractional ideal $\mathfrak{m}$ of K to be a finitely generated R-submodule of K. We define $\mathfrak{m}^{-1}$ to be the fractional ideal $\{x \in K : x\mathfrak{m} \subseteq R\}$.*

The claim that $\mathfrak{m}^{-1}$ is a fractional ideal is somewhat bold, as it is not immediately clear that it satisfies the requirement of being finitely generated as an R-submodule of K. To see that it is, let $x \in \mathfrak{m}$. Then $\mathfrak{m}^{-1}x \subseteq R$, so $\mathfrak{m}^{-1} \subseteq Rx^{-1}$. Since R is Noetherian, Rx^{-1} is finitely generated as a R-submodule of K, so $\mathfrak{m}^{-1}$ must be too.

Definition 3.4. *For two fractional ideals $\mathfrak{m}, \mathfrak{n}$ of K, we define the product $\mathfrak{mn}$ to be the R-submodule of K generated by the set $\{mn : m \in \mathfrak{m}, n \in \mathfrak{n}\}$.*

One can check that this operation, together with the notion of inverses which we defined, give rise to a group operation on the set of all fractional ideals of K. This prompts the following definition:

Definition 3.5. *We define the ideal class group of an algebraic number field K to be the group of fractional ideal of K. We define the class number of K to be the order of the ideal class group of K.*

In general, the class number of a field K need not be finite, but when K is an algebraic number field, it can be proved to be finite. The interested reader is referred to Theorem I.13.8 in [16] for a proof.

Let $\tilde{E}/\mathbb{F}_q$ be an elliptic curve, and suppose we want to lift $\tilde{E}$ to an elliptic curve over the rationals. When doing so, we want to preserve as much of the structure of of $\tilde{E}$ as possible, so we require that $End(\tilde{E}) \cong End(E)$. However, such a lift may very well not exist. In these cases, the next best thing would be to lift $\tilde{E}$ to an elliptic curve over an algebraic number field K. In the event that such a lift really does exist, the following theorem gives a result that we will use to establish a lower bound on the degree of this number field K over the rationals.

Theorem 3.4. *Let E be an elliptic curve representing an isomorphism class over $\mathbb{C}$ with $End(E) \subseteq \mathcal{K}$, where $\mathcal{K}$ is an imaginary quadratic field. Then $[\mathbb{Q}(j(E)) : \mathbb{Q}] = [\mathcal{K}(j(E)) : \mathcal{K}] = h_{\mathcal{K}}$ where $h_{\mathcal{K}}$ denotes the class number of $\mathcal{K}$.*

Proof. See Theorem II.4.2 in [30]. □

Corollary 3.1. *Let $\tilde{E}/\mathbb{F}_q$ be an elliptic curve, and let E/K be a lift of $\tilde{E}$ to a number field K. Assume $End(\tilde{E})$ is an order in an imaginary quadratic field $\tilde{\mathcal{K}}$, and suppose $End(\tilde{E}) \cong End(E)$. Then $[K : \mathbb{Q}] \geq h_{\tilde{\mathcal{K}}}$ where $h_{\tilde{\mathcal{K}}}$ denotes the class number of $\tilde{\mathcal{K}}$.*

Proof. Let Λ denote a representative for an isomorphism class of elliptic curves over $\mathbb{C}$ with $j(\Lambda) = j(E)$ and $End(\Lambda) \cong End(E)$. By Theorem 1.1, $End(E)$ is an order in some imaginary quadratic field $\mathcal{K}$. By Theorem 3.4, we have that $[\mathbb{Q}(j(\Lambda)) : \mathbb{Q}] = h_{\mathcal{K}}$, where $h_{\mathcal{K}}$ denotes the class number of $\mathcal{K}$. Since $j(\Lambda) = j(E) \in K$, we must have that $[K : \mathbb{Q}] \geq [\mathbb{Q}(j(\Lambda)) : \mathbb{Q}] = h_{\mathcal{K}}$.

Furthermore, $End(\tilde{E})$ is also an order in some imaginary quadratic field $\tilde{\mathcal{K}}$. Since $End(E) \cong End(\tilde{E})$, we have that $End(E) \otimes \mathbb{Q} \cong End(\tilde{E}) \otimes \mathbb{Q}$, so $\mathcal{K} \cong \tilde{\mathcal{K}}$. But then $h_{\mathcal{K}} = h_{\tilde{\mathcal{K}}}$, so $[K : \mathbb{Q}] \geq h_{\tilde{\mathcal{K}}}$ which is the desired result. □

There are no publicly known attacks on the ECDLP that exploits a small class number. Regardless, the German Information Security Agency (GISA) require the class number of the field in which $End(E)$ is an order to

be greater than 200 [2]. In Brainpool Standard Curves and Curve Generation [9], they require the class number to be greater than 10000000. They claim that the paper [15] can be seen as an argument for the class number condition. In this paper, an Index Calculus based attack on the ECDLP is described for elliptic curves over an algebraic number field.

3.1.4 Side-Channel Attacks

Although elliptic curve cryptography is based on the hardness of the ECDLP, caution must be taken to prevent attackers from exploiting weak implementations. There exists a plethora of various side-channel attacks on elliptic curve cryptosystems, and we will only give a very brief description of two classes of attacks.

Branching Attacks

Assume that an implementation is using the standard Weierstrass addition law where point doubling and addition of distinct points is different. Now consider the following implementation of the famous Double-And-Add algorithm:

```
def double_and_add(P, n):
    R = P
    for b in map(int, bin(n)[2:]):
        R = 2*R
        if b == 1: R = R + P

    return R
```

If an attacker can observe the power consumption, he can hope to determine the individual bits of n from a power trace of the algorithm since in each iteration, the power consumption in reality is a function of the binary value b. Similar but different attacks can be carried out by timing an algorithm, or observing memory page faults.

Common to many side-channel attacks like this, is that they are often able to use branching or special cases in the algorithm to deduce information about some internal state or value. To counter side-channel attacks when implementing point addition algorithms, a complete addition law, meaning that it gives the correct result for any pair of points on the curve, is therefore desirable. In [7], Bernstein and Lange give fast explicit formulas for an addition law for points on Twisted Edwards curves which we described in Section 1.12.

Invalid Curve Attacks

Assume we have a curve given on short Weierstrass form $E/\mathbb{F}_p : y^2 = x^3 + ax + b$. Since the standard Weierstrass addition law does not involve the constant b, the same addition law will work for any elliptic curve $E'/\mathbb{F}_p : y^2 = x^3 + ax + c$ with $c \in \mathbb{F}_p$. Assume $\#E(\mathbb{F}_p)$ is of prime order. Then it is resistant to small subgroup attacks, and should be safe to use for example in a Diffie-Hellman key exchange scheme (see Section 2.1). However, if the implementation does not check that a point satisfies the curve equation, i.e that a point is actually on the curve E, an attacker can provide a point P on a *different* curve $E'/\mathbb{F}_p : y^2 = x^3 + ax + c$ where P is of small order. This is called an *invalid curve attacks*.

In some implementations, one uses so called x-coordinate ladders (e.g the Montgomery ladder or the Brier-Joye ladder) to do point multiplication. As the name suggests, they only depend on the x-coordinate of a point when doing a point multiplication. This greatly reduces an attackers possibilities for carrying out an invalid curve attack. In this case, the only possibility for an invalid curve attack is on the quadratic twist (this follows from Lemma 1.6).

Assuming now that an implementation fails to check that a given point satisfies the curve equation, and that a malicious participant in a Diffie-Hellman key exchange scheme sends a point on the quadratic twist instead of the intended elliptic curve. If this point is of small order, then the attacker can solve the ECDLP by using a small subgroup attack as described in Section 2.2.3. For this reason, SafeCurves [6] requires the quadratic twist of an elliptic curve E to satisfy the same security requirements of E to protect against invalid curve attacks by moving to the quadratic twist.

3.1.5 Curve Manipulation Attacks

Assume Alice and Bob wants to use an elliptic curve cryptosystem to communicate securely. Then they would typically use an elliptic curve proposed by a third-party, which we will call Snake. Snake may be a government agency or institution such as National Institute of Standards and Technology (NIST). In the 2014 paper [5], Bernstein and others describe various ways in which curve standards may be manipulated by Snake so that he can generate seemingly secure curves that are vulnerable to attacks unknown to the public.

We will use ideas from [5] to estimate how many $\mathbb{F}_p$-isomorphism classes that satisfies our security requirements. Except from the class number condition in Section 3.1.3, our requirements on an elliptic curve $E/\mathbb{F}_p$ are given by restrictions on 1) $\#E(\mathbb{F}_p)$, the number of $\mathbb{F}_p$-rational points on the elliptic curve, or equivalently, the trace of Frobenius and 2) the factorization of $\#E(\mathbb{F}_p)$. Hence, we need estimates for the probability that a randomly

chosen elliptic curve satisfies these requirements.

The next theorem, which is due to Birch, gives an asymptotic estimate of the number of $\mathbb{F}_p$-isomorphism classes of elliptic curves over $\mathbb{F}_p$ where the trace of Frobenius is in a given interval.

Theorem 3.5. *Let p be a prime, and let E_p denote the set of $\mathbb{F}_p$-isomorphism classes of elliptic curves defined over $\mathbb{F}_p$. For $E \in E_p$, let $a_p(E) = p + 1 - \#E(\mathbb{F}_p)$ and set $cos\,\theta_p(E) = \frac{a_p(E)}{2\sqrt{p}}$, then for all $0 \leq \alpha \leq \beta \leq \pi$ we have*

$$\lim_{p\to\infty} \frac{\#\{E \in E_p : \alpha \leq \theta_p(E) \leq \beta\}}{\#E_p} = \frac{2}{\pi}\int_\alpha^\beta sin^2\theta d\theta$$

Proof. See [8] for a proof. □

In the previous sections we showed that elliptic curves over $\mathbb{F}_p$ with trace of Frobenius $a_p \in \{0, 1, 2\}$ are vulnerable to attacks on the ECDLP. We will estimate how many $\mathbb{F}_p$-isomorphism classes of elliptic curves that are subject to these attacks. Consider the following example:

Example 5: We will estimate the number of $\mathbb{F}_p$-isomorphism classes of curves E with $a_p(E) \in \{0, 1, 2\}$ for a couple of prime numbers. Hence, we let $\alpha = cos^{-1}(\frac{2}{2\sqrt{p}})$ and $\beta = cos^{-1}(\frac{1}{2\sqrt{p}})$. We use the previous theorem to compute the (asymptotic) estimates:

$$\frac{\#\{E \in E_p : \alpha \leq \theta_p(E) \leq \beta\}}{\#E_p} \approx \int_\alpha^\beta sin^2(\theta)d\theta$$

We computed the asymptotic estimates for three consecutive 20-bit primes and one 100-bit prime. Then we ran numerical experiments by generating 10000 random elliptic curves over $\mathbb{F}_p$ for the 20-bit prime numbers, and computed an estimated probability that $a_p(E) \in \{0, 1, 2\}$. The results are listed in the table:

p	Asymptotic	Experimental
1048583	$3.108484 \cdot 10^{-4}$	$6.0 \cdot 10^{-4}$
1048589	$3.108475 \cdot 10^{-4}$	$4.0 \cdot 10^{-4}$
1048601	$3.108457 \cdot 10^{-4}$	$7.0 \cdot 10^{-4}$
1267650600228229401496703205653	$2.827159 \cdot 10^{-16}$	N/A [2]

[2] In light of the asymptotic estimate for our 100-bit prime, we would have to test approximately 10^{16} random elliptic curves for a numerical experiment to begin making any sense. This is beyond what was feasible for us.

So for elliptic curves defined over $\mathbb{F}_p$ with p given as above, very few of the $\mathbb{F}_p$-isomorphism classes satisfy $a_p(E) \in \{0, 1, 2\}$. It is clear that as p increases, then the fraction of $\mathbb{F}_p$-isomorphism classes with $a_p(E) \in \{0, 1, 2\}$ will be even lower.

In the previous example, we looked at the frequency of elliptic curves $E/\mathbb{F}_p$ with $a_p(E) \in \{0, 1, 2\}$. These elliptic curves are certainly vulnerable to embedding attacks, but they are not the only elliptic curves that are vulnerable to embedding attacks. Hence, a priori there could still be a substantial fraction of curves that are vulnerable to such attacks. The next theorem asserts that the (asymptotic) fraction of elliptic curves over $\mathbb{F}_p$ with a low embedding degree is indeed very low:

Theorem 3.6. *Let p be a sufficiently large prime number, and let K be a positive integer with $\log(K) = O(\log_2 p)$. Let $E \in E_p$ be randomly chosen, and let $N = \#E(\mathbb{F}_p)$. The probability that $N \mid (p^k - 1)$ for some positive integer $k \leq K$ is at most $p^{-1/(4\kappa+6)+o(1)}$ where $\kappa = \log(K)/\log_2(K)$.*

Proof. See Theorem 3.1 in [19]. □

In [9] one of the requirements for an elliptic curve $E/\mathbb{F}_p$ is that $\#E(\mathbb{F}_p)$ is a prime number. We will follow a similar strategy as in [5] to estimate how likely an elliptic curve is to satisfy this requirement. To do this, we shall use the very famous Prime Number Theorem which we state here for the convenience of the reader:

Theorem 3.7 (The Prime Number Theorem)**.** *Let $\pi(x)$ be the number of primes less than or equal to x. Then $\lim_{x\to\infty} = \frac{\pi(x)}{x/ln(x)} = 1$ or, equivalently, $\pi(x) \sim \frac{x}{ln(x)}$.*

Proof. See Chapter VII in [23]. □

Example 6: We continue from where we left of Example 3.1.5. Let $E/\mathbb{F}_p$ and let p be the prime from this example. In addition to the requirement that $a_p(E) \notin \{0, 1, 2\}$, we want to estimate the probability that a random elliptic curve satisfy the Brainpool requirement that $\#E(\mathbb{F}_p)$ is prime.

From Hasse's theorem (see Theorem 1.2) we know that $p + 1 - 2\sqrt{p} \leq \#E(\mathbb{F}_p) \leq p + 1 + 2\sqrt{p}$. Following the notation from 3.7, we let $\pi(x)$ be the prime counting function, so $\pi(x)$ is the number of primes less than or equal to x. Then by the prime number theorem we have

$$\begin{aligned}\frac{\#\{E \in E_p : \#E(\mathbb{F}_p)\text{is prime}\}}{\#E_p} &\approx \pi(p + 1 + 2\sqrt{p}) - \pi(p + 1 - 2\sqrt{p}) \\ &\sim \frac{p + 1 + 2\sqrt{p}}{\log(p + 1 + 2\sqrt{p}} - \frac{p + 1 - 2\sqrt{p}}{\log(p + 1 - 2\sqrt{p}}\end{aligned}$$

For the three 20-bit primes and the 100-bit prime from Example 3.1.5, we get the following asymptotic and experimental estimates:

p	Asymptotic	Experimental
1048583	0.0669312988	0.0305
1048589	0.0669312733	0.0360
1048601	0.0669312224	0.0324
1267650600228229401496703205653	0.0146484375	0.005

For the numerical experiments, we generated 10000 random elliptic curves over $\mathbb{F}_p$ for the 20-bit primes, and 1000 random elliptic curves for the 100-bit prime. We then counted how many of the elliptic curves that satisfied the requirement that $\#E(\mathbb{F}_p)$ was prime. These experiments indicate that

$$\begin{aligned} \pi_E(p) &= \frac{\#\{E \in E_p : \#E(\mathbb{F}_p) \text{ is prime } \}}{\#E_p} \\ &\approx \frac{1}{2}\left(\frac{p+1+2\sqrt{p}}{\log(p+1+2\sqrt{p})} - \frac{p+1-2\sqrt{p}}{\log(p+1-2\sqrt{p})}\right) \end{aligned} \tag{3.3}$$

is a good estimate for the number of $\mathbb{F}_p$-isomorphism classes E where $\#E(\mathbb{F}_p)$ is prime. Assuming that this is indeed a fair estimate, then approximately 0.3% of elliptic curve defined over a 256-bit prime fields satisfy the Brainpool security requirement that $\#E(\mathbb{F}_p)$ is prime.

The previous results and examples indicated that the class of weak elliptic curves is relatively small. The requirement that the number of $\mathbb{F}_p$-rational points should be of prime order was by far the most difficult requirement to satisfy. It was satisfied by roughly 3% of the elliptic curves over $\mathbb{F}_p$ when p was 20 bits long, and roughly 0.5% of when p was 100 bits long. We used these experiments to estimate that for elliptic curves defined over a 256-bit prime field, the requirement that $E(\mathbb{F}_p)$ is prime is satisfied by approximately 0.3% of the elliptic curves.

Remark 3.3. *Note that the Brainpool requirement that $\#E(\mathbb{F}_p)$ is prime is much stronger than our requirement that $E(\mathbb{F}_p)$ should have a point P of large prime order. It is then reasonable to assume that there will be considerably more isomorphism classes of elliptic curves that satisfy our requirement.*

Continuing now our discussion in the beginning of this section, Snake has a lot of curves to choose from when proposing elliptic curves for public use. Although we estimated that only roughly 0.3% of the $\mathbb{F}_p$-isomorphism classes of elliptic curves over $\mathbb{F}_p$ satisfy the requirement that $\#E(\mathbb{F}_p)$ is prime, it is not unreasonable to assume that Snake has access to vast amounts of

computational power. By Theorem 3.6, only a small fraction of elliptic curves over $\mathbb{F}_p$ have a small embedding degree, so we only need to check this condition when we have found a curve with $\#E(\mathbb{F}_p)$ prime.

On a 64-bit Intel Core i5 1.60 GHz processor, we were able to compute $\#E(\mathbb{F}_p)$ for 10 different elliptic curves over a 256-bit field in less than 71 seconds. This gives an average of approximately 7 seconds to compute $\#E(\mathbb{F}_p)$ for each curve. Curve generation is an embarrassingly parallel problem, so multiple cores would give a linear speedup of this computation time. For less than $600 USD, you can get a 16-core CPU from AMD that according to the manufacturer can be overclocked [3] up to 3.10 GHz. This would give an average time to compute $\#E(\mathbb{F}_p)$ of $\frac{7}{2 \cdot 16} \approx 0.22$ seconds.

If Snake knows a secret vulnerability that applies to an ϵ fraction of all $\mathbb{F}_p$-isomorpism classes of elliptic curves over $\mathbb{F}_p$ and he is cheap enough to only purchase this 16-core CPU from AMD, then Snake can test approximately 40000 curves in 24 hours. Assume that Snake's curves must satisfy the Brainpool requirement that $\#E(\mathbb{F}_p)$ is prime. Within 24 hours, he can expect to find a vulnerable curve that passes this security requirement if ϵ is bigger than approximately $1/(0.003 \cdot 40000) \approx 0.0083$. In a realistic scenario, one would suspect Snake to have a much higher budget, and also more time at hand.

Consequently, if Snake does not have malicious intents, he should provide evidence that the elliptic curve is chosen randomly amongst the curves that satisfy the publicly known security requirements. This can for example be done by giving a detailed description of the curve generation process and should also include a justification of any seeds used for random number generation. Additionally, any such seeds used should also be hashed using a presumed secure hash function like SHA2 or Keccak. This is done to complicate the task of selecting "special" seeds to manipulate the random generation of curves.

Several elliptic curve standards today (e.g [9, 21, 25]) claim that their proposed curves are chosen pseudo-randomly. As an example, we will outline the main steps involved in Brainpool's curve generation process for an n-bit elliptic curve over a prime field $\mathbb{F}_p$ [9]:

1. Use the n-first bits of Euler's number $e \approx 2.71828\ldots$ as the seed, s.
2. Compute $h = \text{SHA1}(s)$, and use this to deterministically choose a (i.e, a is a function of h). Then check if $-3 = au^4$ has a solution in $\mathbb{F}_p$. If not, update the seed s, and start over from 1.
3. Deterministically choose b from h, and check if the elliptic curve $y^2z = x^3 + axz^2 + bz^3$ satisfy the security requirements. If not, update the seed and start over from 1.

[3] Overclocking is a way of making a processor run at a higher clock frequency than intended, usually at the expense of the processor's life span.

Even in this case, we see that there are several steps in the curve generation process that are in no means obvious. For example, how should one deterministically choose the curve constants a and b from the hashed seed s? How should the seed be updated when one of the tests involved in the curve generation process fails? There are also several mathematical constants aside from e (e.g π, $cos(1)$, the golden ratio, Khinchin's constant, etc) that a curve generator just as easily could have chosen as "just a random mathematical constant". Accordingly, Snake is still left with considerable choices when generating elliptic curves. This flexibility was demonstrated in [5] where they, under similar constraints as the Brainpool curve generation process, generated a seemingly random elliptic curve having prescribed properties.

3.1.6 A Proposed Binary Division Attack

In a fairly recent article, Verkhovsky and Polyakov [33] recommend avoiding elliptic curves with $\#E(\mathbb{F}_q) = 2 \pmod 4$. The reason for this being a proposed binary division algorithm which they devised in the article. The algorithm is as following:

Algorithm 3.1. *Let $Q = mP$ and assume P is not two-divisible, i.e there does not exist a point $R \in E(\mathbb{F}_q)$ satisfying $2R = P$. Let $m = b_n b_{n_1} \dots b_0$ be a binary expansion of m which we seek.*

```
R := 0
i := 0
while R != 0 and R != P
    if R is two-divisible
        b_i := 0
    else
        b_i := 1
        R := R - P

    R := R / 2
    i := i + 1

if R = 0 b_i := 1 else b_i := 0
```

In essence, this is just a reversed Double-And-Add-algorithm. It depends on an efficient way to determine two-divisibility and to do point-halving. Note that since the algorithm requires the point P to not be two-divisible, the algorithm does not apply to curves where $\#E(\mathbb{F}_p)$ is odd, since in this case $P = 2A$ for all $P \in E(\mathbb{F}_p)$ with $A = [\frac{q+1}{2}]P$.

We state and prove the following result about the existence of two-divisible points in $E(\mathbb{F}_p)$:

Proposition 3.4. *Let $E/\mathbb{F}_p$ be an elliptic curve, and assume $P \in E(\mathbb{F}_p)$ is a point of odd order. Let $D = \{Q \in E(\mathbb{F}_p) : 2Q = P\}$. Then $\#D \in \{1, 2, 4\}$. If $\#E(\mathbb{F}_p) = 2 \pmod 4$, then $\#D = 2$ and if $\#E(\mathbb{F}_p) = 1 \pmod 2$ then $\#D = 1$.*

Proof. Consider the map $\phi : E \to E$ defined by $Q \mapsto [2]Q - P$. Then $\phi = \tau_{-P} \circ [2]$ where τ_P is the translation-by-P map. Then $D = ker(\phi) \cap E(\mathbb{F}_p) = ker(\tau_{-P} \circ [2]) \cap E(\mathbb{F}_p)$. Since τ_{-P} is an isomorphism (with inverse τ_P), we must have that $\#ker(\tau_{-P} \circ [2]) = \#ker([2]) = 4$. Consequently $\#D \leq 4$.

Let E_2 denote the set of 2-torsion points of E, and let $E_2(\mathbb{F}_p) = E_2 \cap E(\mathbb{F}_p)$. Then $D = A + E_2(\mathbb{F}_p)$ where $A = [\frac{ord(P)+1}{2}]P$. It follows that $\#D = \#E_2(\mathbb{F}_p)$. Since $E_2 = ker\,[2]$ we have $\#E_2 = 4$. $E_2(\mathbb{F}_p)$ is a subgroup of both $E(\mathbb{F}_p)$ and E_2, so $\#E_2(\mathbb{F}_p)$ must divide $\#E(\mathbb{F}_p))$ and $\#E_2 = 4$. The only possibilities are $\#E_2(\mathbb{F}_p) = \#D \in \{1, 2, 4\}$.

If $\#E(\mathbb{F}_p) = 2 \pmod 4$, then $\#E(\mathbb{F}_p)$ does not divide 4, so we must have $\#D \in \{1, 2\}$. But since $\#E(\mathbb{F}_p)$ is even, it has a point of order 2, so $\#D = 2$. By a similar argument, if $\#E(\mathbb{F}_p) = 1 \pmod 2$, then $\#D = 1$. □

Now we will state and prove a lemma which we will use to prove Conjecture 1 in [33].

Lemma 3.3. *Let $P \in E(\mathbb{F}_p)$ with $ord(P)$ odd. Then P is two-divisible by the point $A = [\frac{ord(P)+1}{2}]P$. If $\#E(\mathbb{F}_p) = 2 \pmod 4$, then P is two-divisible if and only if $ord(P)$ is odd.*

Proof. Set $A = [\frac{r+1}{2}]P$. Then $2A = P$, so P is two-divisible.

Suppose $\#E(\mathbb{F}_p) = 2p_1^{r_1} \cdots p_n^{r_n}$ with p_i odd primes. Then any $P \in E(\mathbb{F}_p)$ can be written as a sum $P = B + P_1 + P_2 + \ldots + P_n$, where $B \in \{\mathcal{O}, Z\}$ and Z is the point of order 2, and P_i in a p_i-group. Then $2P = 2(P_1 + P_2 + \ldots + P_n)$, and $ord(2P) = lcm(ord(2P_1), \ldots, ord(2P_n))$ which is odd since each $ord(2P_1)$ is odd. Consequently, if a point is two-divisible, then the order of the point must be odd. □

Verkhovsky and Polyakov state the following conjecture which is easily proved:

Conjecture 1. *Let an elliptic curve $E/\mathbb{F}_p$ be given on simple Weierstrass form, and assume $q = \#E(\mathbb{F}_p) = 2 \pmod 4$. Let $P \in E(\mathbb{F}_p)$ be given. If $(q/2)P = \mathcal{O}$, then P is divisible by two and the two points $A \in E(\mathbb{F}_p)$ satisfying $2A = P$ can be computed as*

$$A_1 = [\frac{q+2}{4}]P \qquad and \qquad A_2 = [\frac{q+2}{4}]P + Z$$

where Z is the point with y-coordinate 0. If $(q/2)P = Z$, then P is not divisible by two.

Proof. Since $\#E(\mathbb{F}_p) = q$ with $q = 2 \pmod 4$, we have $q = 4n + 2$ for some $n \in \mathbb{N}$. But then $\frac{q}{2} = 2n + 1$ which is odd, so $ord(P)$ must be odd. By lemma 3.3, P is two-divisible. Assume $[\frac{q}{2}]P \neq \mathcal{O}$, then P is of even order $ord(P) = r$ so by lemma 3.3, it cannot be two-divisible. In this case we must have that $\frac{r}{2} \mid \frac{q}{2}$, so $gcd(q/2, r) = r/2$. But then $ord([\frac{q}{2}]P) = \frac{r}{gcd(q/2,r)} = 2$, so $[\frac{q}{2}]P = Z$ since Z is the only point in $E(\mathbb{F}_p)$ of order 2. □

From Proposition 3.4, it is clear that there will always be exactly two possibilities for each point-halving when $\#E(\mathbb{F}_p) = 2 \pmod 4$. A rough estimate gives a complexity of at least $O(2^n)$ where n is the number of bits needed to represent p. Verkhovsky and Polyakov claim in [33] that their proposed binary division algorithm is efficient in some cases. Very little justification is given to this claim except from a few test runs of the algorithm on elliptic curves defined over $\mathbb{F}_{23}$. Consequently, we will not take the suggested security requirement that $\#E(\mathbb{F}_p) = 2 \pmod 4$ into consideration when generating elliptic curves for cryptographic applications.

3.2 Security Requirements

In light of the attacks presented in the previous section, we shall in this section list requirements that elliptic curves should satisfy to be secure for use in cryptographic applications. Throughout this section, E will always denote an elliptic curve defined over a prime field $\mathbb{F}_p$, and we assume we have the ECDLP $Q = [m]P$ for $P, Q \in E(\mathbb{F}_q)$. We make the following security requirements:

1. **E must not be anomalous.** Anomalous elliptic curves are subject to the attack described in Section 3.1.1. An anomalous elliptic curve satisfies is a curve where $\#E(\mathbb{F}_p) = q$, or equivalently, curves where the trace of Frobenius is 1. Anomalous elliptic curves are easily avoided by checking that $\#E(\mathbb{F}_p) \neq p$. The number $\#E(\mathbb{F}_p)$ of $\mathbb{F}_p$-rational points on E can be computed in polynomial time using the algorithm described in Section A.1.
2. **P must be of large prime order.** To counter small-subgroups attacks as described in Section 2.2.3, the order of P should be prime. This requirements also prevents the Pholig-Hellman algorithm (see Section 2.2.2) from efficiently computing the discrete logarithm.
3. **E must be resistant to Weil/Tate-embedding attacks.** Let $N = ord(P)$. Then the ECDLP $Q = [m]P$ can be embedded in a finite field extension $\mathbb{F}_{p^d}$ of $\mathbb{F}_p$ of degree d as described in Section 3.1.2. Since the Index Calculus algorithm (see Section 2.3) can solve the induced DLP in $\mathbb{F}_{p^d}$ in sub-exponential time, we need d to be sufficiently large.

4. **Large class number.** As demonstrated in Example 3.1.3, it is reasonably easy to solve the ECDLP for an elliptic curve over $\mathbb{Q}$. The same attack applies to elliptic curves over any algebraic number field K. No general algorithm for lifting E and the points P, Q to non-torsion points on an elliptic curve over K while maintaining the relation $Q = [m]P$ is publicly known today. From the results in Section 3.1.3, the requirement that the class number of the field in which $End(E)$ is an order is large makes it unlikely that E will be rendered vulnerable if a future discovery of such an algorithm is made. It can be seen as a way of "locking down" the ECDLP to the field $\mathbb{F}_p$.
5. **E should be pseudo-randomly selected.** In Section 3.1.5 we saw that it is relatively easy to generate elliptic curves satisfying most of the security requirements. Consequently, a curve standard should provide evidence that the proposed curves are randomly selected (amongst the secure curves) to reduce the probability that the proposed curves are vulnerable to secret attacks unknown to the public.
6. **E should be birationally equivalent to a Twisted Edwards curve $\mathbf{E}_{a,d}$ where d is a non-square.** To counter branching attacks (see Section 3.1.4), we require the addition law to be complete in the sense that it should be defined for all pairs of points on the curve. This limits an attackers possibilities for carrying out branching attacks, e.g by conducting a power trace of the point multiplication algorithm.
7. **E should be twist secure.** The quadratic twist of E (see Section 1.11) should satisfy the same security requirements as E to counter invalid-curve attacks, as described in Section 3.1.4.

Since we have no intentions of proposing a new curve standard, the formulation in 2, 3 and 4 are intentionally vague when we say that the certain quantities should be "large", but without specifying what this really means. Note that 5 is usually safe to neglect if you are generating elliptic curves for your own use, for instance by using the software described in Chapter 6. In this case, you have complete control over the curve generation process so evidence of random generation is typically not needed.

3.3 Accelerating Elliptic Curve Cryptography

The group of points on an elliptic curve is attractive for setting up a discrete logarithm problem since the absence of an Index Calculus-like algorithm allow for smaller keys and less computational power for the same level of security. Efficient algorithms for doing point multiplication are important as they facilitate the use of even larger keys in exchange for little or no extra cost.

In this section we will look at elliptic curves and algorithms that allow for particularly efficient implementations of elliptic curve cryptosystems. We consider different ways of speeding up point multiplication on elliptic curves, and requirements for enabling efficient point compression. Based on this, we will recommending a set of technical requirements for elliptic curves in cryptography.

3.3.1 Fast Point Multiplication

We shall consider two ways in which an elliptic curve can support fast point multiplication. Both are based on the idea of moving the point multiplication to curves in which it can be carried out faster than on arbitrary elliptic curves.

Birationally Equivalent to a Twisted Edwards Curve

Assume E is an elliptic curve that is birationally equivalent to some Twisted Edwards curve $E_{a,d}$. As we saw in Example 1.12, moving the point addition to $E_{a,d}$ can significantly speed up point multiplication. In the example, we used the birational equivalence from Proposition 1.29 to move the point multiplication from the elliptic curve and to the Twisted Edwards curve where it can be done faster.

Isogenous to an Elliptic Curve with Fast Point Multiplication

Brier and Joye [10] suggested the use of isogenies to speed up point multiplication on elliptic curves. Assume we have an elliptic curve E where we want to do point multiplication, and suppose E is isogenous to an elliptic curve E' where we can do fast point multiplication. The idea is then to move the point multiplication to the curve E' where the point multiplication can be done faster, do the multiplications there, and then pull it back. We proceed with formalizing this idea.

Lemma 3.4. *Let $m \in \mathbb{Z}$, and let $\phi \in End(E)$ be an isogeny. Then $\phi \circ [m] = [m] \circ \phi$.*

Proof. We have $(\phi \circ [m])(P) = \phi(P + P + \cdots P) = [m]\phi(P) = ([m] \circ \phi)(P)$, since an isogeny automatically honors the group structure on E. □

Let $\phi : E \to E'$ be an isogeny of degree $d = deg(\phi)$. Then there exists a dual isogeny $\hat{\phi} : E' \to E$ such that $\phi \circ \hat{\phi} = [d] \in End(E)$. Assume now that we want to compute $[dm]P$ for some $m \in \mathbb{N}$ and $P \in E$. It is clear that $[dm] = [d] \circ [m] = (\phi \circ \hat{\phi}) \circ [m]$. By Lemma 3.4, $[m]$ commutes with ϕ and $\hat{\phi}$, so $[dm] = \phi \circ [m] \circ \hat{\phi}$. Thus we obtain the following factorization of the multiplication map:

$$\begin{array}{ccc} E & \xrightarrow{[\mathrm{dm}]} & E \\ \phi \downarrow & & \downarrow \hat{\phi} \\ E' & \xrightarrow[[m]]{} & E' \end{array}$$

The success of this approach is dependent upon a few factors:

1. The existence of curves satisfying our security criteria which are isogenous to curves that allow for fast multiplication.
2. There must be enough of these curves so that there is a reasonable chance that a curve is isogenous to such a curve.
3. The degree of the isogeny must not be too high (preferably 1).

Now we shall see that curves $E\colon y^2z = x^3 + axz^2 + bz^3$ satisfying $a = -3 \pmod 4$ are precisely a family of curves which allow for faster implementations of point multiplication. For $P = (x_1, y_1, z_1) \in E$ the duplications formula gives

$$2P = (x_2, y_2, z_2), \qquad \text{with} \qquad \begin{cases} x_2 = (3x_1^2 + az_1^4)^2 - 8x_1y_1^2 \\ y_2 = (3x_1^2 + az_1^4)(4x_1y_1^2 - x_2) - 8y_1^4 \\ z_2 = 2y_1z_1 \end{cases}$$

This requires 22 field multiplications. The following observation, due to Brier and Joye [10], shows that the choice $a = -3$ reduces the number of field multiplications required when doubling a point.

Observation 3.1. *If $a = -3$ then $x_2 = (3x_1^2 - 3z_1^4) - 8x_1y_1^2 = 3(x_1 + z_1^2)(x_1 - z_1^2) - 8x_1y^2$ which reduces the number of field multiplications for a point doubling to 17.*

Assume $\phi : E \to E'$ is an isogeny between E and an elliptic curve E' where we can do scalar multiplication fast. For this isogeny to be useful in practice, we need the embedding degree of ϕ to be low.

Proposition 3.5. *Assume $p = -3 \pmod 4$. Then approximately half of the isomorphism classes of elliptic curves are $\mathbb{F}_p$-isomorphic to an elliptic curve $E/\mathbb{F}_p\colon y^2z = x^3 + -3xz^2 + bz^3$.*

Proof. Let $E'/\mathbb{F}_p\colon y^2z = x^3 + axz^2 + cz^3$ be an elliptic curve. If E and E' are isomorphic over $\mathbb{F}_p$, then by Proposition 1.9, E and E' are related by a change of variables where $x \mapsto u^2x$ with $u \in \mathbb{F}_p^*$. Expanding this and equating the linear terms gives $au^{-4} = -3$, so $u^{-4} = -\frac{3}{a}$. Thus E and E' are isogenous if and only if $-\frac{3}{a}$ is a quartic residue in $\mathbb{F}_p$.

We now show that $p = -3 \pmod 4$ implies that $x \in \mathbb{F}_p$ is a quartic residue if and only if it is a quadratic residue. This is a well-known result

in number theory, but give a proof for convenience. It is clear that if $x \in \mathbb{F}_p$ is a quartic residue, then it is also a quadratic residue. To show the other implication, we first make the following observation:

$$\left(\frac{-1}{p}\right) = (-1)^{(p-1)/2} = (-1)^{(4n+2)/2} = (-1)^{2n+1} = -1$$

Hence, -1 is not a quadratic residue in $\mathbb{F}_p$. Since the Legendre symbol is a multiplicative function, it follows that $\left(\frac{x}{p}\right) = -\left(\frac{x}{p}\right)$ for all $x \in \mathbb{F}_p^*$, so either x or $-x$ (but not both) is a quadratic residue in $\mathbb{F}_p$. Assume now that $x \in \mathbb{F}_q$ is a quadratic residue. Then $x = y^2$ for some y. Then y or $-y$ is a quadratic residue, so $z^2 = y$ or $z^2 = -y$ for some $z \in \mathbb{F}_p$. In any case, we have $x = y^2 = z^4$, so x is a quartic residue in $\mathbb{F}_p$. For a randomly chosen a, the probability that $-\frac{3}{a}$ is a quadratic residue (and consequently also a quartic residue) is approximately $1/2$. □

3.3.2 Efficient point compression

In implementations, for example in an elliptic curve based public-key cryptosystem or a key exchange algorithm, there is often a need for transmitting points across a network. On systems where bandwidth is limited, point compression can be important for increasing performance.

Proposition 3.6. *Let $E/\mathbb{F}_p : y^2z = f(x, z)$ be an elliptic curve on Weierstrass form. Assume $p > 3$ and $p = 3 \pmod 4$. Let $P = (x_0, y_0, z_0) \in E$ with $z_0 \neq 0$ (i.e P is not the point at infinity). Then y_0 is uniquely determined by $f(x_0, z_0)/z_0$ up to sign.*

Proof. $P = (x_0, y_0, z_0)$ satisfies $y_0^2 z_0 = f(x_0, z_0)$, so $y_0^2 = f(x_0, z_0)/z_0$. Now $(y_0^2)^{(p+1)/4} = y_0^{(p+1)/2}$ since $p = 3 \pmod 4$, and $y_0^{(p+1)/2} = y_0^{(p-1)/2} y_0 = \text{sign}(y_0) f(x_0, z_0)/z_0$ where the second equality follows from the little theorem of Fermat. □

Assume now that Alice wants to send the point $P = (x_0, y_0, z_0)$ to Bob. By the preceding lemma, it is enough for Alice to send x_0, z_0 and a single bit denoting the sign of y_0. Bob can then easily compute y_0 if he knows the curve equation (which he typically does). Thus, the requirement that $p = 3 \pmod 4$ allows for efficient point compression.

Note that in practical implementations one commonly uses affine coordinates, and the point at infinity is handled as a special case. In these cases, Proposition 3.6 reduces the number of bits needed to represent a point by approximately $1/2$.

3.4 Technical Requirements

In light of the previous discussions and results, we shall recommend technical requirements for elliptic curves for use in ECDLP-based cryptosystems. We assume the elliptic curve is written on simple Weierstrass form

$$E : y^2z = x^3 + axz^2 + bz^3 \tag{3.4}$$

We make the following technical requirements:

1. **Fast point multiplication.** The elliptic curve E can support fast point multiplication by being isogenous to an elliptic curve $E' : y^2z = x^3 + a'xz^2 + b'z^3$ where $a' = -3$, or by being birationally equivalent to a Twisted Edwards curve. See Section 3.3.1.
2. **Efficient point compression.** Efficient point compression allow systems which for example have limited storage capacity or bandwidth to efficiently store and/or transfer point on the elliptic curve. As we describe in Section 3.3.2, choosing a base field $\mathbb{F}_p$ where $p = 3 \pmod 4$ allows for efficient point compression.

We have excluded a few technical requirements that some elliptic curve standards make. The main reason for this is that we want the elliptic curves to be "as randomly generated as possible", so technical requirements where there is little or no documentation of the alleged performance or implementation benefits have been excluded. [4]

An example of this is the Brainpool requirement that require $\#E(\mathbb{F}_q) < q$. In [9], they argue that this requirement is for the convenience of the curve implementors, as in some cases the bit length of $\#E(\mathbb{F}_q)$ may exceed the bit length of p. However, by the theorem of Hasse (see Theorem 1.2), the bit length of $\#E(\mathbb{F}_q)$ can never exceed the bit length of q by more than a single bit, and the actual bit length of $\#E(\mathbb{F}_q)$ is easily checked by a curve implementor.

[4] To be completely honest, there is also an element of distrust in the picture. A paranoid (?) person may suspect that a poorly justified technical requirement in reality is designed to trick the public into using an elliptic curve vulnerable to a "secret" attack. That is, the "secret" attack might apply to elliptic curves satisfying this alleged security requirement.

Chapter 4

Examples of Weak Elliptic Curves

In this chapter we will look at concrete examples of weak elliptic curves that are known to be weak in cryptography. The attacks on elliptic curves are primarily attacks that are based on liftings to an elliptic curve over a field where we have a notion of a logarithm, or an embedding attack, where the group used to set up the ECDLP are embedded in a finite field in which subexponential attakcs exist.

4.1 Elliptic Curves with $\#E(\mathbb{F}_p) = p - 1$

Proposition 4.1. *Let E be an elliptic curve with $\#E(\mathbb{F}_p) = p - 1$. Then any cyclic subgroup $\mu_N \subseteq E(\mathbb{F}_p)$ has embedding degree 1 in $\mathbb{F}_p$.*

Proof. $\#\mu_N = N$ divides $\#E(\mathbb{F}_p)$, so $0 = p - 1 \pmod N$. Then $p = 1 \pmod N$, so $ord(p) = 1 \pmod N$. By Proposition 3.2, the embedding degree is 1. □

Example 7: Consider the elliptic curve defined by the equation

$$E/\mathbb{F}_{107} : y^2z = x^346xz^2 + 72z^3 \tag{4.1}$$

It is readily checked that $\#E(\mathbb{F}_{107}) = 106$. We choose a point $P = (66, 45, 1)$ and a point $Q = (90, 72, 1) \in (P)$. Since $gcd(\#E(\mathbb{F}_{107}), p - 1) = p - 1$, we have little control over the torsion points of E, so instead of potentially having to work over a big field extension of $\mathbb{F}_p$, we will instead apply the Tate-Lichtenbaum pairing, denoted τ_N. First, we randomly choose a point R different from P and Q. We picked the point $R = (63, 47, 1)$. Then we compute $\tau_{106}(P, R) = 56 \in \mathbb{F}_{107}$ and $\tau_{106}(Q, R) = 3 \in \mathbb{F}_{107}$ (this can be done using computer algebra software like Sage), and we get the DLP $56^m = 3$ in $\mathbb{F}_{107}$. We find the solution to be $m = 43$.

4.2 Supersingular Elliptic Curves

Supersingular elliptic curves are a family of elliptic curves characterized by having a large endomorphism ring. More precisely, the endomorphism ring has $\mathbb{Z}$-rank 4. The next result says that supersingular elliptic curves over prime fields are precisely the curves where the trace of Frobenius is zero.

Proposition 4.2. *Let E be an elliptic curve over a prime field $\mathbb{F}_p$. Then E is supersingular if and only if $\#E(\mathbb{F}_p) = p + 1$.*

Proof. Let $\#E(\mathbb{F}_p) = p + 1 - a$, so a is the trace of Frobenius. Clearly, we have that $a = p+1-\#E(\mathbb{F}_p) = p+1-deg(1-\phi)$ since $\#E(\mathbb{F}_p) = deg(1-\phi)$ by 1.15. But then we have

$$\begin{aligned}
[a] &= [p + 1 - deg(1 - \phi)] = [p + 1] - [deg(1 - \phi)] \\
&= [p + 1] - \widehat{(1 - \phi)}(1 - \phi) = [p + 1] - (1 - \hat{\phi})(1 - \phi) \\
&= [p + 1] - ([1] - \hat{\phi} - \phi + \hat{\phi}\phi) = [p + 1] - [1] + \hat{\phi} + \phi - [deg(\phi)] \\
&= \hat{\phi} + \phi
\end{aligned}$$

Then we get $\hat{\phi} = [a] - \phi$ which is separable if and only if a does not divide p by Corollary III.5.5 in [31]. Hence, $\hat{\phi}$ is inseparable if and only if $a = 0 \pmod{p}$. Now one can show that $\hat{\phi}$ being inseparable is a neccessary and also sufficient condition for E to be supersingular. Proving this is more involved, and requires some theory that we have not covered. Details can be found in [31], Theorem V.2.2. □

Corollary 4.1. *Let E be a supersingular elliptic curve defined over $\mathbb{F}_p$. Then any cyclic subgroup $\mu_N \subseteq E(\mathbb{F}_p)$ has embedding degree 2 in $\mathbb{F}_p$.*

Proof. $\#\mu_N = N$ divides $\#E(\mathbb{F}_p)$, so $0 = p + 1 \pmod{N}$. Then $p^2 = 1 \pmod{N}$, so $ord(p) = 2 \pmod{N}$ and consequently the embedding degree is 2 by Proposition 3.2. □

The previous proposition showed that supersingular elliptic curves over prime fields $\mathbb{F}_p$ are vulnerable to the MOV-attack described in Section 3.1.2. The next proposition shows that this is also true for supersingular elliptic curves over $\mathbb{F}_q$.

Proposition 4.3. *Let $E/\mathbb{F}_q$ be a supersingular elliptic curve, and let $P \in E(\mathbb{F}_q)$ be an N-torsion point. Then the embedding degree of N in F_q is always less than or equal to 6.*

Proof. See Section 4 in [20]. They give an exhaustive list of the embedding degree of supersingular elliptic curves, and it was shown that it must be either $1, 2, 3, 4$ or 6. □

Example 8: Consider the elliptic curve given by the equation

$$E/\mathbb{F}_{103} : y^2z = x^3 + 61xz^2 + 65z^3 \tag{4.2}$$

One can check that $\#E(\mathbb{F}_{103}) = 104$, so E is supersingular by Proposition 4.2. Take $P = (48, 39, 1) \in E(\mathbb{F}_{103})$ and $Q = (46, 20, 1) \in (P)$ By Corollary 4.1, $e_{104}(E[104] \times E[104]) \subseteq \mathbb{F}_{103^2}$. We canonically lift the curve E and the points P, Q to $E(\mathbb{F}_{103^2})$ where $\mathbb{F}_{103^2} = \mathbb{F}_{103}[x]/(f)$ for some irreducible polynomial $f \in \mathbb{F}_{104}[x]$ of degree 2. We find $R = (78x + 37, 76x + 16, 1)$ to be a 104-torsion point linearly independent of P. Now we compute the Weil pairings $e_{104}(P, R) = 12x + 42$ and $e_{104}(Q, R) = 61x + 79$, and get the DLP $61x + 79 = (12x + 42)^m$ in $\mathbb{F}_{103^2}$. Solving the reduced DLP gives $m = 23$.

4.3 Anomalous Elliptic Curves

Anomalous elliptic curves $E/\mathbb{F}_p$ satisfying $\#E(\mathbb{F}_p) = p$. In other words, they are the elliptic curves with trace of Frobenius equal to 1. The attack on anomalous elliptic curves (described in Section 3.1.1) involved lifting points to an elliptic curve defined over a complete local field $\mathbb{Q}_p$. This may seem impractical at first, but in the proof of Theorem 3.1, we saw that we only have to lift modulo p^2. We demonstrate the attack in the following example.

Example 9: Consider the elliptic curve

$$E/\mathbb{F}_{101} : y^2z = x^3 + 12xz^2 + 83z^3 \tag{4.3}$$

One can check that $\#E(\mathbb{F}_{101}) = 101$, hence E is anomalous. Let $P = (1, 46, 1)$, and $Q = (10, 71, 1)$ be points on E. We want to find m such that $Q = [m]P$. We take the canonical lifting of $E/\mathbb{F}_{101}$ to $E'/\mathbb{Q}_p$. Now we want to lift the points P, Q to some points $P', Q' \in E'(\mathbb{Q}_p)$ while maintaining the relation $Q' = [m]P'$. From the proof of theorem 3.1 we saw that it is enough to lift modulo p^2.

We begin by lifting P modulo p^2 using Hensel's lemma. $P' = (1+pu, 46+pv, 1)$, with $u, v \in \mathbb{F}_{101}$. Now we want P' to be on the curve E', it must satisfy (4.3), so we get

$$\begin{aligned}
(46 + pv)^2 &= (1 + pu)^3 + 12(1 + pu) + 83 \pmod{101^2} \\
46^2 + 2 \cdot 46pv &= 1 + 3pu + 12 + 12pu + 83 \pmod{101^2} \\
92pv &= 15pu + 96 - 46^2 \pmod{101^2} \\
92pv &= 15pu + 81p \pmod{101^2} \\
92v &= 32u + 81 \pmod{101} \\
v &= 32u + 92 \pmod{101}
\end{aligned}$$

Next we lift Q modulo p^2 in a similar way, so we let $Q' = (10 + ps, 71 + pt, 1)$ with $s, t \in \mathbb{F}_{101}$. Obviously we also need Q' to satisfy (4.3), so we get

$$\begin{aligned}
(71 + pt)^2 &= (10 + ps)^3 + 12(10 + ps) + 83 \pmod{101^2} \\
71^2 + 2 \cdot 71pt &= 10^3 + 300ps + 120 + 12ps + 83 \pmod{101^2} \\
142pt &= 312ps + 10^3 + 120 + 83 - 71^2 \pmod{101^2} \\
142pt &= 312ps + 63p \pmod{101^2} \\
142t &= 312s + 63 \pmod{101} \\
t &= 15s + 4 \pmod{101}
\end{aligned}$$

Now by choosing some u and s, for example $u = s = 1$, we can find the lifted points $P' = (1 + 101, 46 + 101 \cdot (32 + 92), 1) = (111, 2369, 1)$ and $Q' = (10 + p, 71 + p(15 + 4), 1) = (111, 1990, 1)$. Now we compute the first terms of the power series giving the formal logrithm of the group E_0':

$$log_E(T) = T + 25T^5 + 50T^7 + 96T^9 + 48T^{11} + 80T^{13} + \dots \tag{4.4}$$

Since $[p]P'$ and $[p]Q'$ are in the kernel of the reduction-modulo-p map, we know that $v_p(-\frac{x}{y}) > 0$ where v_p denotes the p-adic valuation. Hence, we can neglect all but the first term in (4.4), and we get:

$$\begin{aligned}
\frac{log_E([p]Q')}{log_E([p]P')} &= \frac{p \cdot 35}{p \cdot 87} \pmod{p^2} \\
&= \frac{35}{87} = 48 \pmod{p}
\end{aligned}$$

It is readily checked that this is the correct answer.

Chapter 5

Brainpool Standard Curves and Curve Generation

In this chapter we take a closer look at the Brainpool curves [9]. We will consider Brainpool's security and technical requirements in light of the requirements in Section 3.2 and 3.3. In Brainpool, all proposed elliptic curves are over a prime field $\mathbb{F}_p$.

Security Requirements Brainpool describes 6 security requirements that an elliptic curve E over a prime field $\mathbb{F}_p$ must satisfy:

1. **$\#E(\mathbb{F}_p)$ should prime.** This is stronger than our Requirement 2 in Section 3.2. We only require $E(\mathbb{F}_p)$ to contain a subgroup of large prime order. The requirement that $\#E(\mathbb{F}_p)$ is prime implies that there are no points in $E(\mathbb{F}_p)$ of order 4. It was shown in [4] that E cannot be birationally equivalent to a Twisted Edwards curve in this case.
2. **Immunity to Weil-/Tate-pairing.** Requirement 3 in Section 3.2. By Proposition 3.2, the embedding degree l of $q = \#E(\mathbb{F}_p)$ in $\mathbb{F}_p$ equals the order of q modulo p. Brainpool requires $(q-1)/l < 100$, which is a strong requirement. By the little theorem of Fermat, l must then divide $q-1$, and Brainpool verifies the requirement by factoring $q-1$. This is a time consuming operation as $q-1$ is a big number (between 160 and 512 bits).
3. **Trace not equal to one.** Requirement 1 in Section 3.2.
4. **Large class number.** Requirement 4 in Section 3.2. Brainpool requires the class number of the field in which $End(E)$ is an order to be larger than 10000000. Assume that an efficient algorithm for lifting E to a number field K while preserving the relations in the ECDLP was found. If $\#\mathbb{F}_p \approx 2^{256}$, then by Corollary 3.1, representing a single element in K would require approximately $256 \cdot 1000000$ bits.

5. **Verifiably pseudo-random.** Requirement 5 in Section 3.2. Brainpool gives a detailed and verifiable description of the steps involved in the curve generation process.
6. **Proof of Security.** Brainpool requires a curve designer to provide evidence that their proposed curve is not in the class of elliptic curves that are vulnerable to known attacks on elliptic curves. One requirement is that the factorization of $\#E(\mathbb{F}_p) - 1$ is provided.

Technical Requirements In addition to the security requirements listed above, the Brainpool standard also requires their curves to satisfy certain technical requirements. Some of the requirements concern legal issues such as Requirement 4 in [9]. This requirement states that primes used for constructing the base fields should not be a special form. This is to avoid violations of patented fast arithmetic on the base field.

Other requirements (7 and 8) concern details on how elliptic curves should be presented in curve standards to comply with industry standards. We will not consider these requirements, as we regard them as beyond the scope of this thesis. We shall however consider the following technical requirements:

1. **E should be $\mathbb{F}_p$-isomorphic to a curve $E' : y^2z = x^3 + axz^2 + bz^3$ with $a = -3$.** This choice is stricter than the first alternative of Requirement 1 in 3.3, where it is suggested to use an isogeny $\phi : E \to E'$ of low degree. Requiring instead that ϕ is an isomorphism defined over $\mathbb{F}_p$, is equivalent to requiring the degree of this isogeny to be 1.
2. **The base field $\mathbb{F}_p$ should satisfy $p = 3 \pmod 4$.** This requirement is to allow for efficient point compression. We discussed this in 3.3.2, and also stated this as Requirement 2 in Section 3.3.
3. **$\#E(\mathbb{F}_p) < p$.** Brainpool requires this to make it convenient for curve implementers. In some cases, the bit length of $\#E(\mathbb{F}_p)$ exceed the bit-length of p, which can be inconvenient for implementers [9].
4. **For each of the bit lengths 160, 192, 224, 256, 320, 384 and 512 one curve shall be proposed.** Brainpool requires this since there is a need for curves providing different levels of security. Our software implementation (see Chapter 6) enables users to generate a curve with an arbitrary bit length.

Chapter 6

Implementing A Curve Generation Software

In this chapter we present a program we have developed for generating strong elliptic curves for use in cryptography. The program is written using the free open source mathematics software system Sage, and generates strong elliptic curves for use in cryptography. All curves are generated over a prime field $\mathbb{F}_p$, and all generated curves are birationally equivalent to a Twisted Edwards curve (see Section 1.12).

Motivation

In practice, asymmetric cryptography is usually used in combination with symmetric cryptography. Subsequently, it is desirable that the symmetric and asymmetric cryptosystem offers roughly the same level of security. With our software implementation, a user can generate an elliptic curve of any bit length depending on the desired level of security.

Another advantage of generating your own elliptic curves as opposed to using elliptic curves proposed in curve standards, is arguably that of trust. As we discussed in Section 3.1.5, a malicious curve generator may propose curves that are vulnerable to secret attacks that are unknown to the public. Even in cases where the curve generator provide evidence that the curves are generated pseudo-randomly, skepticism is not unfounded.

A third advantage of using this software is that you can generate curves that only satisfy the requirements that you actually need. This is beneficial since it relaxes unnecessary restrictions on the pseudo-random selection of curves.

Curve Generation

The curve generation process is very simple. We randomly generate Twisted Edwards curves, and check to see if the elliptic curve birationally equivalent

with the generated Twiste Edwards curve satisfies a set of security and technical requirements. For example, we require that the trace of Frobenius should be different from 1 to protect against the attack described in Section 3.1.1.

Supported Security Requirements
We support the followings security requirements.

Requirement	Description
Prime order.	Large subgroup of prime order.
Non-Anomalous.	Discard anomalous curves.
Embedding Resistance.	Discard curves with low embedding degree.
Large Class Number	Require the class number of the field in which $End(E)$ is an order to be large.

Supported Technical Requirements
The technical requirements are optional, and can be enabled and disabled using flags when running the curve generation program. Since all the generated curves are birationally equivalent to a Twisted Edwards curve, they automatically support fast multiplication as described in Section 3.3.1.

Requirement	Description	Flag
Point compression	Support point compression	`--point-compression`
Require $\#E(\mathbb{F}_p) < p$	Overrun protection	`--overrun-protection`
d is a non-square	Complete addition law	N/A

Generated Curves

We will now list a few curves that we generated using the software implementation. We generated Twisted Edwards curves over 128-bit, 160-bit, 256-bit and 384-bit prime fields. We shall list them in order. We use a hexadecimal representation of the curve coefficients and the prime modulus.

	Twisted Edwards curve $ax^2z^2 + y^2z^2 = z^4 + dx^2y^2$ over a 128-bit $\mathbb{F}_p$
p	100000000000000000000000000000033
a	82c0cac816af7cb0872110811cbc3365
d	-eee20941efa062b82118e632e4543a79

	Twisted Edwards curve $ax^2z^2 + y^2z^2 = z^4 + dx^2y^2$ over a 160-bit $\mathbb{F}_p$
p	1000000000000000000000000000000000000000007
a	fec118ec92e92cd162a45d16b877d68651760df3
d	-8bdcf869995980da7b923616cd76b72ba156e434

	Twisted Edwards curve $ax^2z^2 + y^2z^2 = z^4 + dx^2y^2$ over a 256-bit $\mathbb{F}_p$
p	10000000000000000000000000000000000000000000000000000000000001e7
a	f100cda978eb6c4d25c841df9dc7405fbfae3369a113c370d206055d208c3c6e
d	-48dfd9b2ac6e3be80fe4d8cd3553617bb83765f6ed408ae6e0c2e8aa5b985575

	Twisted Edwards curve $ax^2z^2 + y^2z^2 = z^4 + dx^2y^2$ over a 384-bit $\mathbb{F}_p$
p	10000000000000000000000000000000000000000000000000000000000000000000000000000000e7
a	-4fee1a963586ef8e10fc046ad6bc10e812542f1efd7adfc1524d577ab9f194500c3e8359389fd36a1979e0c213a0aea5
d	b62e18f5b2c90050cc1804d71a247ec9bbe368156202a1b1be164a9abb2ccb05c72f83ac382cec349fa8c526f46b8692

The prime moduli in all these curves look rather similar in that their hexadecimal representation is made up of many zeros. The reason for this is that the prime moduli for an n-bit prime was chosen to be the first prime bigger than 2^n.

Source Code

The source code is listed in the appendix. See Section B.2.

Appendix A

Algorithms

In this section we will give a brief description of a few important algorithms that we have used in the software implementation. In the implementation we have used the Sage software system where the algorithms for doing field arithmetic, these algorithms are already implemented, sometimes with various tweaks and optimizations.

A.1 Counting Points on an Elliptic Curve

An important characteristic of an elliptic curve over a finite field is the number of rational points on the curve. Counting the number of $\mathbb{F}_q$-rational points on an elliptic curve is something that has been studied intensively the last decades.

Consider the following naive algorithm for counting points on $E(\mathbb{F}_q)$ for an elliptic curve $E/\mathbb{F}_q$ defined by $y^2z = f(x, z)$:

```
def count_points(f):
    # We start at 1 to account for the point at infinity
    count = 1

    for x in range(q):
        if is_square(f(x, 1)):
            count += 2

    return count
```

It requires at least $O(q)$ operations, so it is exponential in the number of bits in q. We will now look at Schoof's algorithm [27] which computes the number of points on an elliptic curve in polynomial time. At the very heart of Schoof's algorithm lies a few results and observations which we will now state.

Proposition A.1. *Let $\tau : E(\bar{\mathbb{F}}_q) \to E(\bar{\mathbb{F}}_q)$ be the Frobenius map $(x, y, z) \mapsto (x^q, y^q, z^q)$. Then τ satisfies the relation $\tau^2 - [a_q]\tau + q = 0 \in End(E)$, where a_q is the trace of Frobenius.*

Now if l is a prime number and $P \in E(\mathbb{F}_q)[l]$ we have

$$(x^{q^2}, y^{q^2}, z^{q^2}) - [n_l](x^q, y^q, z^q) + [q](x, y, z) = 0 \tag{A.1}$$

where $n_l = a_q \pmod{l}$. The basic idea is then to exhaustively search for $n_l \in \mathbb{Z}/l\mathbb{Z}$ and check whether equation (A.1) is satisfied. Then we can formulate a system of congruence equations that can be solved using the Chinese remainder theorem. Since the individual points in $E[l]$ will generally be defined over rather large extension fields of $\mathbb{F}_q$, we will instead make use of what is known as the l-th *division polynomial*:

Definition A.1. *We define the l-th division polynomial to be the polynomial $\psi_l(x)$ whose roots are the x-coordinates of all non-zero l-torsion points of E.*

Instead of working with individual l-torsion points like we did in (A.1), we will formulate the congruence equations with elements in the quotient ring $R_l = \mathbb{F}_q[x, y]/(\psi_l(x), y^2 - f(x))$. Since all l-torsion points will vanish on the polynomials $\psi_l(x)$ and all points on E satisfy $y^2 - f(x)$, we can surely work in this quotient ring.

Although this significantly speeds up computations, Schoof, Elkies and Atkin further developed this idea by substituting ψ_l with a polynomial $f_l \in \mathbb{F}_q[x]$ that divides ψ_l. For a given l, this need not exist but in the cases where it does exist, it allows us to work with polynomials of lower degree and speeds up the arithmetic in R_l. This improvement of Schoof's algorithm is known as the Schoof-Elkies-Atkin algorithm, or simply the SEA algorithm.

A.2 Point Multiplication

A fundamental operation in elliptic curve cryptography is point multiplication, i.e for $P \in E$, compute $[m]P = P + \ldots + P$ (m times). A naive algorithm would be:

```
def add(P, n):
    R = P
    for i in range(n):
        R = R + P

    return R
```

Like our naive attempt at calculating the number of $\mathbb{F}_q$-rational points on an elliptic curve $E/\mathbb{F}_q$, this algorithm is exponential in the number of bits in n and thus useless in cryptography.

```
def double_and_add(P, n):
    R = P
    for b in map(int, bin(n)[2:]):
        R = 2*R
        if b == 1: R = R + P

    return R
```

For each iteration, the point R is doubled and if $b = 1$ we also do a regular point addition. Thus the algorithm will do exactly $\lceil log_2(n) \rceil$ doublings and at most $log_2(n)$ point additions. This algorithm is then linear in the number of bits of n.

Appendix B

Code Listing

B.1 Edwards Curves

B.1.1 Edwards curves and Sage

We have written the following Python/Sage class for handling construction, point addition and finding maps to/from an edwards curve and an elliptic curve.

edwards_curve.py

```
import collections

from sage.schemes.generic.scheme import Scheme, is_Scheme
from sage.schemes.plane_curves.projective_curve \
    import ProjectiveCurve_generic
from sage.schemes.elliptic_curves.weierstrass_transform \
    import WeierstrassTransformation
from sage.schemes.projective.projective_point \
    import SchemeMorphism_point_abelian_variety_field
from sage.schemes.projective.projective_homset \
    import SchemeHomset_points_abelian_variety_field

from sage.all import ProjectiveSpace, EllipticCurve

class TwistedEdwardsCurvePoint(SchemeMorphism_point_abelian_variety_field):
    def __init__(self, curve, v, check = True):
        self.curve = curve

        if v == 0:
            v = (0, 1, 1)
        elif not isinstance(v, collections.Iterable):
```

```
            raise TypeError("Invalid point type")

        if check:
            a, d = self.curve.a, self.curve.d
            x, y, z = v
            if a*x**2*z**2 + y**2*z**2 != z**4 + d*x**2 * y**2:
                raise TypeError("Coordinates %s do not define a point on %s" %
                        (v, curve))

        point_homset = curve.point_homset()
        SchemeMorphism_point_abelian_variety_field.__init__(
                self, point_homset, v, check = False)


    def __add__(self, rhs):
        x1, y1, z1 = self
        x2, y2, z2 = rhs

        a, d = self.curve.ainvs()

        A = z1*z2
        B = A**2
        D = d*x1*x2*y1*y2

        x3 = A*(x1*y2 + x2*y1)*(B - D)
        y3 = A*(y1*y2 - a*x1*x2)*(B + D)
        z3 = (B - D)*(B + D)

        return self.curve.point((x3, y3, z3), check = True)

    def __mul__(self, n):
        if n == 0: return self.curve.point(0)
        if n == 1: return self

        # Dumb point multiplication
        if n < 80:
            P = 0
            for i in xrange(n):
                P = P + self

            return P

        r = floor(log(n, 2))
        d = n - 2**r
```

```
            P = self.double(r)

            return P + self * d

        def double(self, n):
            x1, y1, z1 = self
            x2, y2, z2 = self

            a, d = self.curve.ainvs()

            for i in range(n):
                A = z1*z2
                B = A**2
                D = d*x1*x2*y1*y2

                x3 = A*(x1*y2 + x2*y1)*(B - D)
                y3 = A*(y1*y2 - a*x1*x2)*(B + D)
                z3 = (B - D)*(B + D)

                x1, y1, z1 = x3, y3, z3
                x2, y2, z2 = x3, y3, z3

            return self.curve.point((x3, y3, z3), check = False)

    class TwistedEdwardsCurve(ProjectiveCurve_generic):
        _point = TwistedEdwardsCurvePoint

        def __init__(self, K, ainvs):
            self.__base_ring = K
            self.__ainvs = tuple(map(K, ainvs))
            self.a, self.d = self.__ainvs

            P2 = ProjectiveSpace(2, K, names='xyz')
            x, y, z = P2.coordinate_ring().gens()

            a, d = self.ainvs()
            f = a*x**2*z**2 + y**2*z**2 - z**4 - d*x**2*y**2
            ProjectiveCurve_generic.__init__(self, P2, f)

        def __str__(self):
            a, d = self.ainvs()
            return "Twisted Edwards Curve defined by %dx^2 + y^2 = 1 - %dx^2y^2 " \
                   "over %s" % (a, d, self.base_ring())
```

```
def to_weierstrass_map(self):
    """
    Returns a morphism from this Edwards curve to the associated
    elliptic curve on Weierstrass form.
    """

    P2 = ProjectiveSpace(2, self.__base_ring, names='xyz')
    x, y, z = P2.coordinate_ring().gens()

    a, d = self.ainvs()

    E = self.associated_ec()
    C = P2.subscheme(a*x**2*z**2 + y**2*z**2 - z**4 - d*x**2*y**2)
    f = WeierstrassTransformation(C, E, [
        (a - d)*(z + y)*x,       # <-- x
        (a - d)*2*(z**2 + y*z),  # <-- y
        z*x*(z - y)              # <-- z
        ], 1)

    return f

def from_weierstrass_map(self):
    """
    Returns a morphism from the associated elliptic curve on
    Weierstrass form to this Edwards curve.
    """

    P2 = ProjectiveSpace(2, self.__base_ring, names='xyz')
    x, y, z = P2.coordinate_ring().gens()

    a, d = self.ainvs()

    E = self.associated_ec()
    C = P2.subscheme(a*x**2*z**2 + y**2*z**2 - z**4 - d*x**2*y**2)
    f = WeierstrassTransformation(E, C, [
        2*x*(x + (a - d)*z),  # <-- x
        (x - (a - d)*z)*y,    # <-- y
        y * (x + (a - d)*z)    # <-- z
        ], 1)

    return f

def associated_ec(self):
```

```
        """
        Returns the elliptic curve that is birationally equivalent to
        this Edwards curve. That is, the elliptic curve given by
        E: y^2 = x^3 + 2(a + d)x^2 + (a - d)^2x
        """

        a, d = self.ainvs()
        K = self.__base_ring
        return EllipticCurve(K, [0, 2*(a + d), 0, (a - d)**2, 0])

    def ainvs(self):
        return self.__ainvs

    def __call__(self, *args):
        if len(args) == 1 and args[0] == 0:
            R = self.base_ring()
            return self.point([R(0), R(1), R(1)], check=False)

        return self.point(*args, check=True)

    def _point_homset(self, *args, **kwds):
        return SchemeHomset_points_abelian_variety_field(*args, **kwds)
```

B.1.2 Benchmarking the Point Addition Algorithm

```
load('edwards_curve.py')

from sage.all import *
import random

K = GF(13)
ed = TwistedEdwardsCurve(K, [1, 2])
ec = ed.associated_ec()

Q = ec(12, 2, 1)
f = ed.from_weierstrass_map()
P = TwistedEdwardsCurvePoint(ed, f(Q))

timeit("P + P", number = 10000)
timeit("Q + Q", number = 10000)
```

```
$ sage edwards_test.sage
1000000 loops, best of 3: 19 microseconds per loop
1000000 loops, best of 3: 31.6 microseconds per loop
```

B.2 The Software Implementation

main.py

```
#!/usr/bin/env sage

import sys
import time

from argparse import ArgumentParser
from sage.all import GF, is_prime, next_prime

from curve_generator import CurveGenerator
from trait_set       import TraitSet
from traits          import SecurityTrait, TechnicalTrait

class Main:
    def __init__(self):
        self.args = self.make_parser().parse_args()
        self.curve_traits = self.make_curve_traits()
        self.field_traits = self.make_field_traits()

    def run(self):
        start = time.time()

        F = self.make_base_field()

        curve = CurveGenerator(F, self.curve_traits).run()

        # Save generated curve to file
        with open(self.args.outfile, 'w') as outfile:
            outfile.write(str(curve) + '\n')
            outfile.write(str(curve.associated_ec()) + '\n')
            outfile.write("Elapsed time: " + str(time.time() - start))


    def make_base_field(self):
        if self.args.base_field != None:
            F = GF(self.args.base_field, 'F')
            self.field_traits.check(F)
            return F

        p = next_prime(2**self.args.num_bits)
        while not self.field_traits.check(GF(p, 'F'), nothrow = True):
```

```
            p = next_prime(p)

        return GF(p, 'F')

    def make_curve_traits(self):
        """ Security and technical requirements for the elliptic curve """
        traits = TraitSet()

        # Security requirements
        traits.add_trait(SecurityTrait.PrimeOrderSubgroup)
        traits.add_trait(SecurityTrait.NonTraceOne)
        traits.add_trait(SecurityTrait.EmbeddingResistance)

        # Technical requirements
        if self.args.overrun_protection:
            traits.add_trait(TechnicalTrait.OverrunProtection)

        return traits

    def make_field_traits(self):
        """ Security and technical requirements for the base field"""
        traits = TraitSet()

        # Technical requirements
        if self.args.point_compression:
            traits.add_trait(TechnicalTrait.PointCompression)

        return traits

    def make_parser(self):
        parser = ArgumentParser()

        parser = ArgumentParser(add_help = False)
        parser.add_argument('--num-proc',           type=int)
        parser.add_argument('--point-compression',  action='store_true')
        parser.add_argument('--overrun-protection', action='store_true')
        parser.add_argument('outfile', type=str,
                help='A path to a file where the generated curve will be stored')

        # Add subparser for base field options
        field_opts = parser.add_mutually_exclusive_group(required = True)

        field_opts.add_argument(
                '--base-field',
```

```
                type=str,
                help='Base field for the curve operations (i.e: GF(11)')

        field_opts.add_argument(
                '--num-bits',
                type=int,
                help='Select a random field with the given number of bits')

        return parser


if __name__ == "__main__":
    main = Main()
    main.run()
```

curve_generator.py

```
from sage.all import EllipticCurve, parallel

from random_curve_iterator import RandomCurveIterator

class CurveGenerator:
    """
    Generate a curve over a field F of characteristic different from 2 and
    3, that passes a set of tests on the curve and base field.
    """

    def __init__(self, F, curve_traits):
        self.F = F
        self.curve_traits = curve_traits

    def run(self):
        print("Starting curve generation...")
        for tec in RandomCurveIterator(self.F):
            ec = tec.associated_ec()
            if self.check_curve(ec):
                return tec

    def check_curve(self, curve):
        return self.curve_traits.check(curve, nothrow = True)
```

random_curve_iterator.py

```
from sage.all import EllipticCurve
from edwards_curve import TwistedEdwardsCurve
```

```
class RandomCurveIterator:
    def __init__(self, F):
        self.F = F

    def __iter__(self):
        return self

    def next(self):
        if self.F.characteristic() < 3:
            raise ArithmeticError("Only fields of characteristic > 3 is supported")

        while True:
            try:
                a = self.F.random_element()
                d = self._random_nonsquare()

                tec = TwistedEdwardsCurve(self.F, [a, d])
                aec = tec.associated_ec()
                return tec
            except ArithmeticError:
                # The associated elliptic curve is singular. No biggie.
                pass

    def _random_nonsquare(self):
        """ We want the addition law to be complete """
        d = 1
        while self.F(d).is_square():
            d = self.F.random_element()

        return d
```

trait_set.py

```
class TraitError(RuntimeError):
    pass

class TraitSet(object):
    def __init__(self):
        self.traits = []

    def add_trait(self, trait):
        self.traits += [trait]
```

```
    def check(self, ec, nothrow = False):
        for pred, msg in self.traits:
            if not pred(ec):
                if nothrow: return False
                else:       raise TraitError(msg)

        return True
```

traits.py

```
from sage.all import *

def embedding_degree(E):
    """ Compute the embedding degree of E in F_p """
    p = E.base_field().characteristic()
    q = E.cardinality()
    return Zmod(q)(p).multiplicative_order()

def cm_discriminant(E):
    """ Compute the CM-discriminant of the field K in which
    End(E) is an order """
    p = E.base_field().order()
    t = E.trace_of_frobenius()

    a = t**2 - 4*p

    s = 1
    for f, m in factor(a):
        if m % 2 == 0:
            s *= f**m

    D = a / s
    if D % 4 == 1: return abs(D)

    return 4*abs(D)

def has_prime_order_subgroup(E):
    q = E.cardinality()
    return q % 4 == 0 and is_prime(Integer(q / 4))

class SecurityTrait(object):
    PrimeOrderSubgroup = (
            lambda E: has_prime_order_subgroup(E),
            "The subgroup of rational points is of non-prime order")
```

```
    NonTraceOne = (
            lambda E: E.cardinality() != E.base_field().order(),
            "The curve is of trace one (anomalous)")

    EmbeddingResistance = (
            lambda E: embedding_degree(E) > 5,
            "The curve is of low embedding degree")

class TechnicalTrait(object):
    PointCompression = (
            lambda F: F.order() % 4 == 3,
            "Base field is not congrurent 3 mod 4")

    OverrunProtection = (
            lambda E: E.cardinality() < E.base_field().order(),
            "Number of rational points exceed field order")
```

Bibliography

[1] M. F. Atiyah and I.G. MacDonald. *Commutative algebra*. Addison-Wesley, 1969.

[2] H. Baier. Elliptic curves of prime order over optimal extension fields for use in cryptography. In *Progress in cryptology - indocrypt 2001*, page 101, 2001.

[3] Daniel J Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. Twisted edwards curves. In *Progress in Cryptology–AFRICACRYPT 2008*, pages 389–405. Springer, 2008.

[4] Daniel J Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. Twisted edwards curves. In *Progress in Cryptology–AFRICACRYPT 2008*, pages 389–405. Springer, 2008.

[5] Daniel J. Bernstein, Tung Chou, Chitchanok Chuengsatiansup, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, and Christine van Vredendaal. How to manipulate curve standards: a white paper for the black hat. *Cryptology ePrint Archive, Report 2014/571*, 2014. `https://eprint.iacr.org/2014/571.pdf`.

[6] Daniel J. Bernstein and Tanja Lange. Safecurves: choosing safe curves for elliptic-curve cryptography. `http://safecurves.cr.yp.to`. Accessed 2015-01-28.

[7] Daniel J Bernstein and Tanja Lange. Faster addition and doubling on elliptic curves. In *Advances in cryptology–ASIACRYPT 2007*, pages 29–50. Springer, 2007.

[8] Bryan J Birch. How the number of points of an elliptic curve over a fixed prime field varies. *Journal of the London Mathematical Society*, 1(1):57–60, 1968. Oxford University Press.

[9] ECC Brainpool. Brainpool standard curves and curve generation. Technical report, 2005. `www.ecc-brainpool.org/download/Domain-parameters.pdf`.

[10] Eric Brier and Marc Joye. Fast point multiplication on elliptic curves through isogenies. In *Applied algebra, algebraic algorithms and error-correcting codes*, pages 43–50. Springer, 2003.

[11] Harold Edwards. A normal form for elliptic curves. *Bulletin of the American Mathematical Society*, 44(3):393–422, 2007.

[12] Gerhard Frey and Hans-Georg Rück. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Mathematics of computation*, 62(206):865–874, 1994.

[13] Robin Hartshorne. *Algebraic geometry*, volume 52. Springer Science & Business Media, 1977.

[14] Jeffrey Hoffstein, Jill Catherine Pipher, Joseph H Silverman, and Joseph H Silverman. *An introduction to mathematical cryptography*. Springer, 2008.

[15] Ming-Deh Huang and Wayne Raskind. Signature calculus and discrete logarithm problems. In *Algorithmic Number Theory*, pages 558–572. Springer, 2006.

[16] Gerald J Janusz. *Algebraic Number Fields*, volume 7 of *Graduate Studies in Mathematics*. American Mathematical Society, 2. edition, 1996.

[17] RSA Laboratories. What are elliptic curves? `http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/what-are-elliptic-curves.htm`. Accessed: 2010-09-30.

[18] Chae Hoon Lim and Pil Joong Lee. A key recovery attack on discrete log-based schemes using a prime order subgroup. In *Advances in Cryptology—CRYPTO'97*, pages 249–263. Springer, 1997.

[19] Florian Luca, David Jose Mireles, Igor E Shparlinski, et al. Mov attack in various subgroups on elliptic curves. *Illinois Journal of Mathematics*, 48(3):1041–1052, 2004.

[20] Alfred J Menezes, Tatsuaki Okamoto, and Scott A Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *Information Theory, IEEE Transactions on*, 39(5):1639–1646, 1993. IEEE.

[21] National Institute of Standards and Technology. FIPS PUB 186-4: Digital Signature Standard (DSS). Technical report, Gaithersburg, MD, USA, 2013. `http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf`.

[22] Nicole Perlroth, Jeff Larson, and Scott Shane. N.s.a. able to foil basic safeguards of privacy on web. *New York*

Times, September 2013. http://www.nytimes.com/2013/09/06/us/nsa-foils-much-internet-encryption.html.

[23] Paul Pollack. *Not always buried deep: a second course in elementary number theory*. American Mathematical Soc., 2009.

[24] John M Pollard. A monte carlo method for factorization. *BIT Numerical Mathematics*, 15(3):331–334, 1975. Springer.

[25] Certicom Research. Sec 2: Recommended elliptic curve domain parameters. Technical report, January 2010. http://www.secg.org/sec2-v2.pdf.

[26] Takakazu Satoh. Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves. *Commentarii Math. Univ. Sancti Pauli*, 47(1):81–92, 1998.

[27] René Schoof. Counting points on elliptic curves over finite fields. *Journal de théorie des nombres de Bordeaux*, 7(1):219–254, 1995. Université Bordeaux I.

[28] Igor Semaev. Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p. *Mathematics of Computation of the American Mathematical Society*, 67(221):353–356, 1998.

[29] Jean-Pierre Serre, Willem Kuyk, and John Labute. *Abelian l-adic representations and elliptic curves*, volume 2. WA benjamin New York, 1968.

[30] Joseph H Silverman. *Advanced topics in the arithmetic of elliptic curves*, volume 151. Springer Science & Business Media, 1994.

[31] Joseph H Silverman. *The Arithmetic of Elliptic Curves*. Springer, New York, US, 2009.

[32] Nigel P Smart. The discrete logarithm problem on elliptic curves of trace one. *Journal of cryptology*, 12(3):193–196, 1999.

[33] Boris S. Verkhovsky and Yuriy S. Polyakov. Binary division attack for elliptic curve discrete logarithm problem. *Transactions on Networks and Communications*, 2014.