Evaluating Large Degree Isogenies between Elliptic Curves

by

Vladimir Soukharev

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

Waterloo, Ontario, Canada, 2010

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

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

Abstract

An isogeny between elliptic curves is an algebraic morphism which is a group homomorphism. Many applications in cryptography require evaluating large degree isogenies between elliptic curves efficiently. For ordinary curves of the same endomorphism ring, the previous fastest algorithm known has a worst case running time which is exponential in the length of the input. In this thesis we solve this problem in subexponential time under reasonable heuristics. We give two versions of our algorithm, a slower version assuming GRH and a faster version assuming stronger heuristics. Our approach is based on factoring the ideal corresponding to the kernel of the isogeny, modulo principal ideals, into a product of smaller prime ideals for which the isogenies can be computed directly. Combined with previous work of Bostan et al., our algorithm yields equations for large degree isogenies in quasi-optimal time given only the starting curve and the kernel.

Acknowledgements

I would like to thank my supervisor David Jao. Also, I would like to thank Alfred Menezes and Edlyn Teske.

Dedication

This is dedicated to my parents Guennadi and Lioubov and my brother Pavel.

Table of Contents

List of Figures

Chapter 1

Introduction

An isogeny between a pair of elliptic curves is an algebraic morphism that maps the identity point of the first curve to the identity point of the second curve. The degree of the isogeny is its degree as an algebraic map. A well known theorem of Tate [Tat66] states that two elliptic curves defined over the same finite field $\mathbb{F}_q$ are isogenous (i.e. admit an isogeny between them) if and only if they have the same number of points over $\mathbb{F}_q$. Using fast point counting algorithms such as Schoof's algorithm and others [CFA$^+$06, Sch95], it is very easy to check whether this condition holds, and thus whether or not the curves are isogenous. However, Tate's theorem is non-constructive, and constructing the actual isogeny itself is believed to be a hard problem. Indeed, given an ordinary curve $E/\mathbb{F}_q$ and a degree n, the fastest previously known algorithm for constructing an isogeny of degree n has a running time of $O(n^{3+\varepsilon})$, except in a certain very small number of special cases [BCL08, Gal99, GHS02]. In this thesis, we present a new probabilistic algorithm for evaluating such isogenies, which in the vast majority of cases runs (heuristically) in subexponential time. Specifically, we show that for ordinary curves, one can evaluate isogenies of degree n between curves of nearly equal endomorphism ring over $\mathbb{F}_q$ in time less than $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2}) \log(n)$, provided n has no large prime divisors in common with the endomorphism ring discriminant. Although this running time is not polynomial in the input length, our algorithm is still much faster than the (exponential) previous fastest algorithm known, and in practice allows for the evaluation of isogenies of cryptographically sized degrees, some examples of which we present here.

Isogenies have a great number of applications in cryptography. To give some context for our work, we present a few of these applications here. Some of the examples of applications that we describe include the use of isogenies in counting the number of points on a given elliptic curve defined over some finite field, and the use of isogenies to transfer the discrete logarithm problem in cryptography.

We then provide a summary of previous techniques for evaluating large degree isogenies.

Most of these techniques make use of Vélu's formulas. Vélu's formulas are used to obtain the explicit isogeny and the image curve given the original curve and the kernel of the isogeny. One of the previous fastest known techniques for evaluating large degree isogenies is the so-called Elkies-Atkin technique. It uses modular polynomials and j-invariants of the curves. However, in order to evaluate an isogeny of degree n, the technique requires modular polynomials of level n, which take $O(n^{3+\epsilon})$ time to compute. Although these methods have exponential running time, our subexponential algorithm is based on these techniques, and hence we present them in some detail.

The Bröker-Charles-Lauter algorithm [BCL08] is a more efficient algorithm for constructing isogenies when the discriminant of the endomorphism ring is small. We describe their algorithm in Chapter 4. The major idea is to factor the large prime degree isogeny into a product of small prime degree isogenies and a scalar isogeny. The factorization is obtained for ideals, but for efficiency the factorization is computed in the ideal class group of the endomorphism ring of the given input curve E. Once the factorization is obtained, one can use old techniques recursively to evaluate the isogenies corresponding to each of the elements in the factorization to evaluate the isogeny.

Our algorithm uses the same approach as the Bröker-Charles-Lauter algorithm, but we speed up the process of obtaining the factorization of the ideal that corresponds to the isogeny. In order to achieve that, we use the ideas of Haffner and McCurley's index calculus algorithm [HM89], originally used for computing the structure of the class group of the imaginary quadratic order. We assume a few reasonable heuristics, which include the Generalized Riemann Hypothesis. Our heuristic assumptions are a subset of the heuristic assumptions that were used by Bisson and Sutherland [BS09] in their independent work where they use similar techniques to compute the endomorphism ring of an elliptic curve. We also provide a number of examples to show how our algorithm performs.

Our algorithm has a running time of complexity of $L_{|\Delta|}(\frac{1}{2}, \frac{\sqrt{3}}{2})\log(n)$, where n is the degree of the isogeny and Δ is the discriminant of the quadratic order isomorphic to the endomorphism ring of E. That is, it is polynomial in the degree of the isogeny and subexponential in the magnitude of the discriminant of the endomorphism ring. If we let q be the size of the finite field over which the elliptic curve E is defined, then we can also express the running time as $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2})\log(n)$. Our algorithm in combination with the work by Bostan et al. [BMSS08] yields a quasi-optimal (in the degree of the isogeny) algorithm for finding the explicit equation of the isogeny between the given pair of isogenous curves defined over the same finite field. More details can be found at the end of Chapter 5.

We also describe a variant of our algorithm in Chapter 6. The major difference is that the variant algorithm only requires the GRH assumption. Although that algorithm is slower in practice, the differences can be absorbed into the implied constants, and hence its asymptotic running time is no different than algorithm presented in Chapter 5.

Chapter 2

Isogenies and Applications to Cryptography

In this chapter we present isogenies in depth. We start with background material and definitions. Then, we provide some examples of major applications of isogenies to cryptography, including counting points on elliptic curves over a finite field and the transfer of discrete logarithms.

2.1 Algebraic Curves

The goal in this section to briefly present the material needed to be able to define the notion of isogenies between elliptic curves. The material in this section and the following two sections is contained in [Sil92] (in particular, the first 3 chapters). However our presentation will be more brief and simplified in many respects. In many cases, definitions and propositions, theorems, etc. will be used and the proofs omitted. The reader who is interested in more detail and proofs may refer to that book.

We let K be a perfect field, $\bar{K}$ a fixed algebraic closure of K and $G_{\bar{K}/K}$ the Galois group of $\bar{K}/K$.

We first begin with background on affine varieties.

Definition 2.1.1. *Affine n-space (over K)* is the set of n-tuples

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

Also, the *set of K-rational points in* $\mathbb{A}^n$ is defined by

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

Note that in this work we will mainly focus on $\mathbb{A}^2$ and $\mathbb{A}^3$.

Let $I \subset \bar{K}[X_1, \ldots, X_n]$ be an ideal. Then we associate to I the following subset of $\mathbb{A}^n$:

$$V_I = \{P \in \mathbb{A}^n : f(P) = 0 \text{ for all } f \in I\}.$$

We thus obtain the following definitions:

Definition 2.1.2. An *(affine) algebraic set* is any set of the form V_I. Also, if V is an algebraic set, the *ideal of* V is given by

$$I(V) = \{f \in \bar{K}[X_1, \ldots, X_n] : f(P) = 0 \text{ for all } P \in V\}.$$

We say that V *is defined over* K, denoted by V/K, if $I(V)$ can be generated by polynomials in $K[X_1, \ldots, X_n]$. If V is defined over K, the *set of* K*-rational points of* V is the set

$$V(K) = V \cap \mathbb{A}^n(K)$$

We also define $I(V/K) = I(V) \cap K[X_1, \ldots, X_n]$. If we refer to Hilbert's basis theorem, we see that all such ideals are finitely generated. In this work we will mainly be concerned with the case where $I(V)$ is principal (i.e. generated by one polynomial). Also, note that if $f(X_1, \ldots, X_n) \in K[X_1, \ldots, X_n]$ and $P \in \mathbb{A}^n$, then for any $\sigma \in G_{\bar{K}/K}$, $f(P^\sigma) = f(P)^\sigma$.

Definition 2.1.3. V is called an *(affine) variety* if it is an affine algebraic set such that $I(V)$ is a prime ideal in $\bar{K}[X_1, \ldots, X_n]$. If V/K is a variety, then the *affine coordinate ring of* V/K is defined by

$$K[V] = \frac{K[X_1, \ldots, X_n]}{I(V/K)}$$

Observe that $K[V]$ is an integral domain, and its quotient field, denoted by $K(V)$, is called the *function field of* V/K. (We define $\bar{K}[V]$ and $\bar{K}(V)$ in a similar manner by replacing K with $\bar{K}$.)

We need a few more definitions related to the dimension of V.

Definition 2.1.4. Let V be a variety. The *dimension of* V, denoted by $\dim(V)$, is the transcendence degree of $\bar{K}(V)$ over K.

We will deal primarily with varieties $V \subset \mathbb{A}^n$ given by a single non-constant polynomial; in this case $\dim(V) = n - 1$.

Definition 2.1.5. Let V be a variety, $P \in V$, and $f_1, \ldots, f_m \in \bar{K}[X_1, \ldots, X_n]$ a set of generators for $I(V)$. Then we say that V is *non-singular (or smooth) at* P if the $m \times n$ matrix

$$(\partial f_i / \partial X_j(P))_{1 \le i \le m, 1 \le j \le n}$$

has rank $n - \dim(V)$. If V is non-singular at every point, then we say that V is *non-singular (or smooth)*.

When $m = 1$, a point $P \in V$ is a singular point if and only if

$$\partial f/\partial X_1(P) = \cdots = \partial f/\partial X_n(P) = 0.$$

We now move to discussing projective varieties. Projective spaces arose through the process of adding "points at infinity" to affine spaces.

Definition 2.1.6. *Projective n-space (over K)*, denoted $\mathbb{P}^n$ or $\mathbb{P}^n(\bar{K})$, is the set of all $(n+1)$-tuples

$$(x_0, \ldots, x_n) \in \mathbb{A}^{n+1}$$

such that at least one x_i is non-zero, modulo the equivalence relation given by

$$(x_0, \ldots, x_n) \sim (y_0, \ldots, y_n)$$

if there exists a $\lambda \in \bar{K}^*$ with $x_i = \lambda y_i$ for all i. We denote the equivalence class $\{(\lambda x_0, \ldots, \lambda x_n) : \lambda \in \bar{K}^*\}$ by $[x_0, x_1, \ldots, x_n]$, and we call $x_0, \ldots, x_n$ *homogeneous coordinates* for the corresponding point in $\mathbb{P}^n$. As usual, the *set of K-rational points in* $\mathbb{P}^n$ is given by

$$\mathbb{P}^n(K) = \{[x_0, x_1, \ldots, x_n] \in \mathbb{P}^n : \text{ all } x_i \in K\}.$$

Notice that if $P = [x_0, \ldots, x_n] \in \mathbb{P}^n(K)$, it does not mean that each $x_i \in K$; however, it does mean that choosing some i so that $x_i \neq 0$, we get that each $x_j/x_i \in K$.

Definition 2.1.7. A polynomial $f \in K[X_1, \ldots, X_n]$ is *homogeneous of degree d* if

$$f(\lambda X_0, \ldots, \lambda X_n) = \lambda^d f(X_0, \ldots, X_n)$$

for all $\lambda \in \bar{K}$. An ideal $I \subset \bar{K}[X_1, \ldots, X_n]$ is *homogeneous* if it is generated by homogeneous polynomials.

Given a homogeneous ideal I, we associate a subset of $\mathbb{P}^n$,

$$V_I = \{P \in \mathbb{P}^n : f(P) = 0 \text{ for all homogeneous } f \in I\}.$$

Definition 2.1.8. A *(projective) algebraic set* is any set of the form V_I. If V is a projective algebraic set, the *(homogeneous) ideal of* V, denoted by $I(V)$, is the ideal in $\bar{K}[X_1, \ldots, X_n]$ generated by

$$\{f \in \bar{K}[X_1, \ldots, X_n] : f \text{ is homogeneous and } f(P) = 0 \text{ for all } P \in V\}.$$

We say that such a V is *defined over* K, denoted by V/K, if its ideal $I(V)$ can be generated by homogeneous polynomials in $K[X_1, \ldots, X_n]$. As usual, if V is defined over K, the *set of K-rational points of* V is the set $V(K) = V \cap \mathbb{P}^n(K)$.

Definition 2.1.9. A projective algebraic set V is called a *(projective) variety* if its homogeneous ideal $I(V)$ is a prime ideal in $\bar{K}[X_1, \ldots, X_n]$.

Note that $\mathbb{P}^n$ contains many copies of $\mathbb{A}^n$. For each $0 \leq i \leq n$, we have an inclusion

$$\phi_i \colon \mathbb{A}^n \to \mathbb{P}^n$$

$$(y_1, \ldots, y_n) \to [y_1, y_2, \ldots, y_{i-1}, 1, y_i, \ldots, y_n].$$

We define:

$$U_i = \{P = [x_0, \ldots, x_n] \in \mathbb{P}^n : x_i \neq 0\}$$

(Notice that $U_0, \ldots, U_n$ cover all of $\mathbb{P}^n$.) Hence, we get a natural bijection

$$\phi_i^{-1} \colon U_i \to \mathbb{A}^n$$

$$[x_0, \ldots, x_n] \to (x_0/x_i, x_1/x_i, \ldots, x_{i-1}/x_i, x_{i+1}/x_i, \ldots, x_n/x_i).$$

Thus, fixing i, we will identify $\mathbb{A}^n$ with the set U_i in $\mathbb{P}^n$ via ϕ_i. So, given a projective algebraic set V with homogeneous ideal $I(V) \subset \bar{K}[X_1, \ldots, X_n]$, we will write $V \cap \mathbb{A}^n$ to denote $\phi_i^{-1}(V \cap U_i)$, which is the affine algebraic set with ideal $I(V \cap \mathbb{A}^n) \subset \bar{K}[Y_1, \ldots, Y_n]$ given by

$$I(V \cap \mathbb{A}^n) = \{f(Y_1, \ldots, Y_{i-1}, 1, Y_i, \ldots, Y_n) : f(X_0, \ldots, X_n) \in I(V)\}.$$

This process of replacing $f(X_0, \ldots, X_n)$ by $f(Y_1, \ldots, Y_{i-1}, 1, Y_i, \ldots, Y_n)$ is called *dehomogenization with respect to X_i*. We can also reverse the process—namely, given $f(Y_1, \ldots, Y_n) \in \bar{K}[Y_1, \ldots, Y_n]$, let

$$f^*(X_0, \ldots, X_n) = X_i^d f(X_0/X_i, X_1/X_i, \ldots, X_{i-1}/X_i, X_{i+1}/X_i, \ldots, X_n/X_i)$$

where $d = \deg(f)$ is the smallest integer for which f^* is a polynomial. (We call f^* the *homogenization of f with respect to X_i*.)

Definition 2.1.10. Let V be an affine algebraic set with ideal $I(V)$, and consider V as a subset of $\mathbb{P}^n$ via the map

$$V \subset \mathbb{A}^n \xrightarrow{\phi_i} \mathbb{P}^n.$$

The *projective closure of V*, denoted by $\bar{V}$, is the algebraic set whose homogeneous ideal $I(\bar{V})$ is generated by

$$\{f^*(X_1, \ldots, X_n) : f \in I(V)\}.$$

In this way, each affine variety can be identified with a unique projective variety. Since notationally it is easier to deal with affine coordinates, often, we will write down a non-homogeneous equation for a projective variety V, with the understanding that V is the projective closure of the given affine variety W. The points $V - W$ are called *points at infinity on V*.

Example 2.1.11. Define V to be the *projective* variety given by the equation

$$V : Y^2 = X^3 + 17.$$

In this case we really mean the variety in $\mathbb{P}^2$ given by homogeneous equation

$$\bar{Y}^2\bar{Z} = \bar{X}^3 + 17\bar{Z}^3.$$

This variety has one point at infinity, $[0, 1, 0]$ (we obtain it by setting $\bar{Z} = 0$).

Certain properties of a projective variety V are defined in terms of the affine (sub)variety $V \cap \mathbb{A}^n$.

Definition 2.1.12. Let V/K be a projective variety. Choose $\mathbb{A}^n \subset \mathbb{P}^n$ so that $V \cap \mathbb{A}^n \neq \emptyset$. The *dimension of* V is the dimension of $V \cap \mathbb{A}^n$. The *function field of* V, denoted $K(V)$, is the function field of $V \cap \mathbb{A}^n$; similarly for $\bar{K}(V)$.

Definition 2.1.13. Let V be a projective variety with $P \in V$. Choose $\mathbb{A}^n \subset \mathbb{P}^n$ so that $P \in \mathbb{A}^n$. Then V is *non-singular (or smooth) at P* if $V \cap \mathbb{A}^n$ is non-singular at P.

We now move on to algebraic maps between projective varieties, which are the maps defined by rational functions.

Definition 2.1.14. Let V_1 and $V_2 \subset \mathbb{P}^n$ be projective varieties. A *rational map from V_1 to V_2* is a map of the form

$$\phi \colon V_1 \to V_2$$

$$\phi = [f_0, \ldots, f_n],$$

where all $f_i \in \bar{K}(V_1)$ have the property that for every point $P \in V_1$ at which f_i's are all defined,

$$\phi(P) = [f_0(P), \ldots, f_n(P)] \in V_2.$$

If V_1 and V_2 are defined over K, then $G_{\bar{K}/K}$ acts on ϕ in the following way:

$$\phi^\sigma(P) = [f_0^\sigma(P), \ldots, f_n^\sigma(P)].$$

If there is some $\lambda \in \bar{K}^*$ so that $\lambda f_0, \ldots, \lambda f_n \in K(V_1)$, then ϕ is said to be *defined over K*.

Definition 2.1.15. A rational map

$$\phi = [f_0, \dots, f_n] \colon V_1 \to V_2$$

is *regular (or defined) at* $P \in V_1$ if there is a function $g \in \bar{K}(V_1)$ such that each gf_i is regular at P and for some i, $(gf_i)(P) \neq 0$. If such g exists, we set

$$\phi(P) = [(gf_0)(P), \dots, (gf_n)(P)].$$

A rational map which is regular at every point is called a *morphism.*

We now move on to *curves*, which are projective varieties of dimension 1. We will mostly focus on smooth curves.

Proposition 2.1.16. *Let C be a curve, $V \subset \mathbb{P}^N$ a variety, $P \in C$ a smooth point, and $\phi \colon C \to V$ a rational map. Then ϕ is regular at P. In particular, if C is smooth, then ϕ is a morphism.*

Proof. [Sil92, II.2.1]. □

Theorem 2.1.17. *Let $\phi \colon C_1 \to C_2$ be a morphism of curves. Then ϕ is either constant or surjective.*

Proof. [Sil92, II.2.3]. □

We remark that by definition of $\mathbb{P}^n$ (Definition 2.1.6), the surjectivity of ϕ in Theorem 2.1.17 refers to points over $\bar{K}$, not over K.

Let C_1 and C_2 be curves defined over a field K and let $\phi \colon C_1 \to C_2$ be a non-constant rational map defined over K. The composition with ϕ induces an injection of function fields that fixes K:

$$\phi^* \colon K(C_2) \to K(C_1)$$
$$\phi^*(f) = f \circ \phi.$$

We are now ready to define the degree of ϕ.

Definition 2.1.18. Let $\phi \colon C_1 \to C_2$ be a map of curves defined over K. If ϕ is constant, we define the *degree of* ϕ to be 0. Otherwise, we say that ϕ is *finite*, and define its *degree* by

$$\deg \phi = [K(C_1) : \phi^*(K(C_2))].$$

We say that ϕ is *separable (inseparable)* if the extension $K(C_1)/\phi^*(K(C_2))$ is *separable (inseparable).*

It is a known fact that if ϕ is a non-constant map from curve C_1 to curve C_2 defined over K, then $[K(C_1) : \phi^*(K(C_2))]$ is finite [Sil92, II.2.4(a)]; hence the definition makes sense.

Now we need to define the notion of differentials.

Definition 2.1.19. Let C be a curve. The *space of (meromorphic) differential forms* on C, denoted Ω_C, is the $\bar{K}(C)$-vector space generated by symbols of the form dx for $x \in \bar{K}(C)$, subject to the usual relations:

- $d(x + y) = dx + dy$ for all $x, y \in \bar{K}(C)$;
- $d(xy) = xdy + ydx$ for all $x, y \in \bar{K}(C)$;
- $da = 0$ for all $a \in \bar{K}$.

Once again, if we let $\phi\colon C_1 \to C_2$ be a non-constant map of curves, then the map $\phi^*\colon \bar{K}(C_2) \to \bar{K}(C_1)$ induces a map on differentials

$$\phi^*\colon \Omega_{C_2} \to \Omega_{C_1}$$

$$\phi^*(\sum f_i dx_i) = \sum (\phi^* f_i) d(\phi^* x_i).$$

2.2 Elliptic Curves

An elliptic curve is a curve given by a Weierstrass equation over some field $\mathbb{F}$ (as shown below). An elliptic curve admits an addition operation, which we will define shortly, making the set of points on the curve into an abelian group. We focus on the case where the characteristic of the field is different from 2 and 3; the general case may be found in [Sil92, App. A].

We define the *Weierstrass equation* to be the locus in $\mathbb{P}^2$ of the curve

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3,$$

where $a_1, \ldots, a_6 \in \bar{K}$. For ease of notation, we use non-homogeneous coordinates to express the Weierstrass equation in the following way:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6.$$

We must remember that there is one point at infinity, $[0, 1, 0]$, which we will denote by ∞. If C is the curve represented by the above equation and $a_1, \ldots, a_6 \in K$, then we say that

C is defined over K. If we assume that $\mathrm{char}(K) \neq 2, 3$, then using a change of variables, we can simplify the equation to

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

There are a few associated values with this curve:

- *discriminant* $\Delta = -16(4a^3 + 27b^2)$.
- *j-invariant* $j = -1728(4a)^3/\Delta$.
- *invariant differential* $\omega = dx/(2y) = dy/(3x^2 + b)$.

The curve represented by the above equation is smooth if and only if $\Delta \neq 0$.

Definition 2.2.1. Let F be a field such that $\mathrm{char}\, F \neq 2, 3$. Let $a, b \in F$. An *elliptic curve* E, defined over the field F, is a set

$$\{(x, y) \in F \times F : y^2 = x^3 + ax + b\} \cup \{\infty\}$$

where $4a^3 + 27b^2 \neq 0$.

We will usually denote the elliptic curve by $E(F)$, $E : y^2 = x^3 + ax + b$, or simply by E when the field and equation are known.

As already mentioned, E forms an abelian group under the *group law*, where the point at infinity, ∞, is the identity of the group. We define the *group law* here. Let $P = (x_1, y_1), Q = (x_2, y_2) \in E$. Then we define:

- $P + \infty = \infty + P = P$
- $-P = (x_1, -y_1)$ (assuming $P \neq \infty$)
- $P + (-P) = \infty$
- $P + Q = R = (x_3, y_3) =$

  $$\left(\frac{x_1^4 - 2ax_1^2 - 8bx_1 + a^2}{4(x_1^3 + ax_1 + b)}, \frac{(x_1^6 + 5ax_1^4 + 20bx_1^3 - 5a^2x_1^2 - 4abx_1 - 8b - a)y_1}{8(x_1^3 + ax_1 + b)^2} \right),$$

  if $P = Q$ and $P \neq \infty$
- $P + Q = R = (x_3, y_3) =$

  $$\left(\frac{y_1^2 - 2y_1y_2 + y_2^2 - x_1^3 + x_1^2x_2 + x_1x_2^2 - x_2^3}{x_1^2 - 2x_1x_2 + x_2^2}, \frac{x_1y_2 - x_2y_1 + x_3y_1 - x_3y_2}{x_2 - x_1} \right),$$

  if $P \neq Q$ and $P, Q \neq \infty$

The group law admits a geometric interpretation [Sil92, III.2], but we do not give that here since our emphasis is on the algebra.

When char F is 2 or 3, then the Weierstrass equation simplifies to different forms, with the discriminant, j-invariant, invariant differential, and the group law modified accordingly. For details see [Sil92, III.1, III.2, A].

2.3 Isogenies

We are now ready to define an *isogeny.* We give the definition of isogenies and examine some of their properties. We then present some examples of families of isogenies.

Definition 2.3.1. Let E and E' be elliptic curves defined over some field F. An *isogeny* $\phi\colon E \to E'$ is an algebraic morphism of the form

$$\phi(x,y) = \left(\frac{f_1(x,y)}{g_1(x,y)}, \frac{f_2(x,y)}{g_2(x,y)}\right),$$

satisfying $\phi(\infty) = \infty$ (where $f_i's$ and $g_i's$ are polynomials in x and y). We say that E_1 and E_2 are *isogenous* if there is an isogeny either from E_1 to E_2 or E_2 to E_1.

One can show that every isogeny is in fact a group homomorphism [Sil92, III.4.8].

There is only one constant isogeny, namely $\phi(P) = \infty$ for all $P \in E_1$. This constant isogeny is usually denoted by $[0]$, and by convention we let $\deg[0] = 0$. All other isogenies are non-constant, hence surjective (Theorem 2.1.17), that is $\phi(E_1) = E_2$. For all such non-constant isogenies, we define the degree to be the degree as an algebraic map (i.e. $[F(E_1) : \phi^*(F(E_2))]$); and we classify the isogeny to be separable (inseparable) if the extension $F(E_1)/\phi^*(F(E_2))$ is separable (inseparable).

Let $\phi\colon E_1 \to E_2$ be a non-constant isogeny. We define $\ker\phi = \phi^{-1}(\infty)$. It is known that $\ker\phi$ is a finite subgroup of E_1 [Sil92, III.4.9].

Theorem 2.3.2. *Let E_1, E_2 be elliptic curves defined over field F. Let $\phi\colon E_1 \to E_2$ be a non-constant separable isogeny. Then $\#\ker\phi = \deg\phi$.*

Proof. [Sil92, III.4.10(c)]. □

Proposition 2.3.3. *Let E be an elliptic curve over some field F. Let Φ be a finite subgroup of E defined over F. Then there exists a unique elliptic curve E' (over F) and a separable isogeny*

$$\phi\colon E \to E'$$

such that

$$\ker\phi = \Phi.$$

Proof. [Sil92, III.4.12]. □

We now look at a few examples of isogenies.

Example 2.3.4. *Scalar multiplication*

Let F be a field of characteristic different from 2 and 3 and $E(F) : y^2 = x^3 + ax + b$ be an elliptic curve. For $n \in \mathbb{Z}$, define $[n]\colon E \to E$ by $[n](P) = nP$ (we usually call this *multiplication by n-map*). Then $[n]$ is a separable isogeny. We can give an explicit algebraic morphism for each such n by using the group law for elliptic curves; for instance when $n = 2$,

$$[2](x, y) = \bigg(\frac{x^4 - 2ax^2 - 8bx + a^2}{4(x^3 + ax + b)}, \\ \frac{(x^6 + 5ax^4 + 20bx^3 - 5a^2x^2 - 4abx - 8b - a)y}{8(x^3 + ax + b)^2}\bigg)$$

The degree of $[n]$ is n^2. One can show this by constructing the appropriate field extension and computing the degree of the extension, but this computation is tedious and we do not give it here. The cardinality of $\ker([n])$ is also n^2. Note that $\#\ker([n]) = \deg[n]$, which agrees with Theorem 2.3.2.

Example 2.3.5. *Frobenius map*

Let $F = \mathbb{F}_q$ be a finite field of size q (where q is a prime power). Let E be an elliptic curve defined over $\mathbb{F}_q$. Define $\pi\colon E \to E$ by

$$\pi(x, y) = (x^q, y^q).$$

Then π is an algebraic map and a group homomorphism, hence an isogeny. In fact, π is an inseparable isogeny. Observe that $\deg(\pi) = q$, but $\#\ker(\pi) = 1$. In this case $\deg(\pi) \neq \#\ker(\pi)$ because π is *inseparable.*

Example 2.3.6. *Complex multiplication* Let F be a field such that $i = \sqrt{-1} \in F$. Let $E\colon y^2 = x^3 - x$ be defined over F. Define

$$\phi(x, y) = (-x, iy).$$

Then $\phi \circ \phi = [-1]$. This isogeny can be viewed as an extension of scalar multiplication isogenies to complex numbers.

Notice that in the definition of isogeny, we stated that elliptic curves are isogenous if there exists an isogeny from E_1 to E_2 *or* from E_2 to E_1. In fact, these two conditions are equivalent, as the following result shows.

Theorem 2.3.7. *Let E_1, E_2 be elliptic curves and $\phi\colon E_1 \to E_2$ be an isogeny defined over field F. Let $m = \deg \phi$. Then there exists a unique isogeny*

$$\hat{\phi}\colon E_2 \to E_1$$

which satisfies

$$\hat{\phi} \circ \phi = [m] \text{ (on } E_1\text{) and } \phi \circ \hat{\phi} = [m] \text{ (on } E_2\text{).}$$

Proof. [Sil92, III.6.1(a) and III.6.2(a)]. □

Definition 2.3.8. Let E_1, E_2 be elliptic curves and $\phi\colon E_1 \to E_2$ be an isogeny defined over field F. The *dual isogeny* to ϕ is the isogeny

$$\hat{\phi}\colon E_2 \to E_1$$

given by 2.3.7. (Note that here we assume that $\phi \neq [0]$. If $\phi = [0]$, then we set $\hat{\phi} = [0]$.)

It follows that the relation of being isogenous is an equivalence relation.

We need a few more facts about dual isogenies, which are summarized in the following theorem.

Theorem 2.3.9. *Let E_1, E_2, E_3 be elliptic curves and let $\phi\colon E_1 \to E_2$, $\varphi\colon E_1 \to E_2$, and $\psi\colon E_2 \to E_3$ be isogenies defined over field F. Then:*

- $\widehat{\psi \circ \phi} = \hat{\phi} \circ \hat{\psi}$.
- $\widehat{\phi + \varphi} = \hat{\phi} + \hat{\varphi}$.
- *For all $m \in \mathbb{Z}$, $\widehat{[m]} = [m]$ and $\deg[m] = m^2$.*
- $\deg \hat{\phi} = \deg \phi$.
- $\hat{\hat{\phi}} = \phi$.

Proof. [Sil92, III.6.2]. □

Example 2.3.10. *Dual isogenies*

- Let $F = \mathbb{F}_{109}$.
- Let $E_1\colon y^2 = x^3 + 2x + 2$ and $E_2\colon y^2 = x^3 + 34x + 45$. An isogeny $\phi\colon E_1 \to E_2$ (of degree 3) is given by

$$\phi(x, y) = \left(\frac{x^3 + 20x^2 + 50x + 6}{x^2 + 20x + 100}, \frac{(x^3 + 30x^2 + 23x + 52)y}{x^3 + 30x^2 + 82x + 19}\right).$$

- There exists an isogeny $\hat{\phi}\colon E_2 \to E_1$, given by

  $$\hat{\phi}(x,y) = \left(\frac{x^3 + 49x^2 + 46x + 104}{9x^2 + 5x + 34}, \frac{(x^3 + 19x^2 + 66x + 47)y}{27x^3 + 77x^2 + 88x + 101} \right),$$

  satisfying $\phi \circ \hat{\phi} = [3]$ and $\hat{\phi} \circ \phi = [3]$.

- $\hat{\phi}$ is the *dual isogeny* of ϕ and vice-versa.
- Note that this implies that $\deg(\phi \circ \hat{\phi}) = \deg(\hat{\phi} \circ \phi) = 3^2 = 9$.

There is a very useful theorem by Tate which provides us with an efficient method for determining whether two curves are isogenous or not.

Theorem 2.3.11. *For any two curves E_1 and E_2 defined over $\mathbb{F}_q$, there exists an isogeny from E_1 to E_2 over $\mathbb{F}_q$ if and only if $\#E_1(\mathbb{F}_q) = \#E_2(\mathbb{F}_q)$.*

Proof. [Tat66, §3]. □

Note that using techniques of Schoof in [Sch95], we can compute the number of points on a given elliptic curve in polynomial time. Hence, we obtain an efficient way to check whether two curves are isogenous or not. However, Tate's theorem does not tell us what that isogeny is or how to compute it.

2.4 The Endomorphism Ring of an Elliptic Curve

We now define and give some of the properties of the endomorphism ring of an elliptic curve E. Given elliptic curves E_1 and E_2 defined over some field F, we set

$$\mathrm{Hom}(E_1, E_2) = \{\phi : \phi\colon E_1 \to E_2 \text{ is an isogeny over } \bar{F}\}.$$

Definition 2.4.1. Let E be an elliptic curve defined over a field F. Then the *endomorphism ring* of E is

$$\mathrm{End}(E) = \mathrm{Hom}(E, E).$$

Notice how in the definition, we have used the term *ring.* Besides being the set of all isogenies that map from $E(\bar{F})$ to itself, $\mathrm{End}(E)$ is a ring under pointwise addition (i.e. if $P \in E(\bar{F})$ and $\phi_1, \phi_2 \in \mathrm{End}(E)$, then $(\phi_1 + \phi_2)(P) = \phi_1(P) + \phi_2(P)$) with the multiplication operation being composition of isogenies (i.e. $(\phi_1\phi_2)(P) = (\phi_1 \circ \phi_2)(P) = \phi_1(\phi_2(P))$).

We now specialize to the case of elliptic curves defined over finite fields.

Theorem 2.4.2. *Let E be an elliptic curve defined over a finite field. As a $\mathbb{Z}$-module, $\dim_{\mathbb{Z}} \operatorname{End}(E)$ is equal to either 2 or 4.*

Proof. [Sil92, V.3.1]. □

We formulate a definition to distinguish between the two cases.

Definition 2.4.3. An elliptic curve E over a finite field is *supersingular* if $\dim_{\mathbb{Z}} \operatorname{End}(E) = 4$, and *ordinary* if $\dim_{\mathbb{Z}} \operatorname{End}(E) = 2$.

Two isogenous elliptic curves E_1 and E_2 are either both ordinary, or both supersingular. Thus, there will never be an isogeny between an ordinary and a supersingular elliptic curve. In cryptography, it is more common to work with ordinary curves as they are more secure. The reason is that Menezes et al. [MOV91] have shown that the discrete logarithm problem on a supersingular elliptic curve can be reduced to a discrete logarithm problem in a finite field; this reduction is referred to as the "MOV reduction." One of the main applications of isogenies is discrete logarithm reductions between elliptic curves (Section 2.6). Hence for the rest of this thesis we will only consider ordinary elliptic curves.

Before continuing our discussion of endomorphism rings, we need to briefly discuss the topic of orders in quadratic fields. This material appears in [Cox89, p. 133].

Definition 2.4.4. Let K be a quadratic field (that is, a number field of degree 2). An *order* $\mathcal{O}$ in K is a subset $\mathcal{O} \subset K$ such that:

- $\mathcal{O}$ is a subring of K (containing 1).
- $\mathcal{O}$ is a finitely generated $\mathbb{Z}$-module.
- $\mathcal{O}$ contains a $\mathbb{Q}$-basis of K.

Note that it follows from the definition that $\mathcal{O}$ is a free $\mathbb{Z}$-module of rank 2.

When K is a quadratic field, let $\mathcal{O}_K$ be the ring of integers of K. Then $\mathcal{O}_K$ is an order in K. Moreover, if we let $\mathcal{O}$ be any order of K, then $\mathcal{O} \subset \mathcal{O}_K$. The order $\mathcal{O}_K$ is called the *maximal order* of K.

We can describe these orders more explicitly. Let Δ_K be the discriminant of K and let

$$w_K = \frac{\Delta_K + \sqrt{\Delta_K}}{2}.$$

Then

$$\mathcal{O}_K = \mathbb{Z}[w_K].$$

We can also give a more explicit description of an arbitrary order $\mathcal{O}$ in K.

Lemma 2.4.5. *Let $\mathcal{O}$ be an order in a quadratic field K of discriminant Δ_K. Then $\mathcal{O}$ has finite index in $\mathcal{O}_K$. Letting $c = [\mathcal{O}_K : \mathcal{O}]$, we have*

$$\mathcal{O} = \mathbb{Z} + c\mathcal{O}_K = \mathbb{Z}[cw_K],$$

where w_K is defined as above.

Proof. [Cox89, §7]. □

Note: The index value c in Lemma 2.4.5 is called the *conductor* of $\mathcal{O}$. Also note that if we are given an order $\mathcal{O}$ of discriminant Δ, then the discriminant of the maximal order $\mathcal{O}_K$ is the largest square-free part of Δ, i.e. $\Delta = c^2\Delta_K$, where Δ_K is the discriminant of $\mathcal{O}_K$ and c is a conductor. We say that $\mathcal{O}$ is an *imaginary quadratic order* if $\Delta < 0$, and a *real quadratic order* otherwise.

We now return to the description of the endomorphism ring.

Theorem 2.4.6. *Let E be an ordinary elliptic curve defined over the finite field $\mathbb{F}_q$. Then*

$$\operatorname{End}(E) \cong \mathcal{O}_\Delta,$$

where $\Delta < 0$. That is, the endomorphism ring of E is isomorphic to an imaginary quadratic order of discriminant Δ.

Proof. [Sil92, V.3.1]. □

(Note: This Δ is unrelated to the Δ that we defined previously as the discriminant of the elliptic curve. From now on, we will use Δ to refer only to the discriminant of an imaginary quadratic order.)

Let E be an elliptic curve defined over $\mathbb{F}_q$, let π_q be the Frobenius map, and let $t = \operatorname{Trace}(\pi_q)$ be the trace of π_q as an element of $\operatorname{End}(E)$. The integer t is called the trace of E. We have a relation $t = q + 1 - \#E(\mathbb{F}_q)$ [Sil92, p. 142] and $\pi_q^2 - t\pi_q + q = 0$.

Let K denote the imaginary quadratic field containing $\operatorname{End}(E)$, with maximal order $\mathcal{O}_K$. The field K is called the CM field of E. We write c_E for the conductor of $\operatorname{End}(E)$ and c_π for the conductor of $\mathbb{Z}[\pi_q]$. It follows from Lemma 2.4.5 and [Cox89, §7] that $\operatorname{End}(E) \cong \mathbb{Z} + c_E\mathcal{O}_K$ and $\Delta = c_E^2\Delta_K$, where Δ (respectively, Δ_K) is the discriminant of the imaginary quadratic order $\operatorname{End}(E)$ (respectively, $\mathcal{O}_K$). Furthermore, the characteristic polynomial $x^2 - tx + q$ of π_q has discriminant $\Delta_\pi = t^2 - 4q = \operatorname{disc}(\mathbb{Z}[\pi_q]) = c_\pi^2\Delta_K$, with $c_\pi = c_E \cdot [\operatorname{End}(E) : \mathbb{Z}[\pi_q]]$.

Following [FM02] and [Gal99], we say that an isogeny $\phi\colon E \to E'$ of prime degree ℓ defined over $\mathbb{F}_q$ is *"down"* if $[\operatorname{End}(E) : \operatorname{End}(E')] = \ell$ (note that this means that $\operatorname{End}(E') \subset$

End(E)), *"up"* if $[\mathrm{End}(E') : \mathrm{End}(E)] = \ell$ (note that this means that $\mathrm{End}(E) \subset \mathrm{End}(E')$), and *"horizontal"* if $\mathrm{End}(E) = \mathrm{End}(E')$. Two curves in an isogeny class are said to "have the same level" if their endomorphism rings are equal. Within each isogeny class, the property of having the same level is an equivalence relation. A horizontal isogeny always goes between two curves of the same level; likewise, an up isogeny enlarges the endomorphism ring and a down isogeny reduces it. Since there are fewer elliptic curves at higher levels than at lower levels, the collection of elliptic curves in an isogeny class visually resembles a "pyramid" or a "volcano" [FM02], with up isogenies ascending the structure and down isogenies descending. If we restrict to the graph of ℓ-isogenies for a single ℓ, then in general the ℓ-isogeny graph is disconnected, having one ℓ-volcano for each intermediate order $\mathbb{Z}[\pi_q] \subset \mathcal{O} \subset \mathcal{O}_K$ such that $\mathcal{O}$ is maximal at ℓ (meaning $\ell \nmid [\mathcal{O}_K : \mathcal{O}]$). The "top level" of the class consists of curves E with $\mathrm{End}(E) = \mathcal{O}_K$, and the "bottom level" consists of curves with $\mathrm{End}(E) = \mathbb{Z}[\pi_q]$.

The structure of an isogeny volcano is illustrated in Figure 2.1.

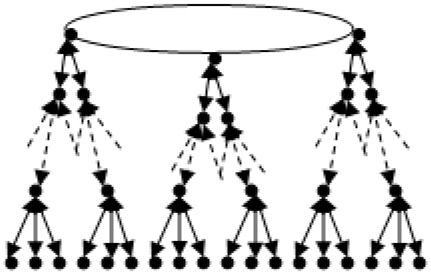

Figure 2.1: Isogeny volcano

We also have the following theorem that states the number of ℓ-isogenies of each type.

Theorem 2.4.7. *Let E be an ordinary elliptic curve over $\mathbb{F}_q$, having endomorphism ring* End(E) *of discriminant Δ. Let ℓ be a prime different from the characteristic of $\mathbb{F}_q$.*

- *Assume $\ell \nmid c_E$. Then there are exactly $1 + \left(\frac{\Delta}{\ell}\right)$ horizontal isogenies $\phi\colon E \to E'$ of degree ℓ.*
    - *If $\ell \nmid c_\pi$, there are no other isogenies $E \to E'$ of degree ℓ over $\mathbb{F}_q$.*
    - *If $\ell \mid c_\pi$, there are $\ell - \left(\frac{\Delta}{\ell}\right)$ down isogenies of degree ℓ.*

- *Assume $\ell \mid c_E$. Then there is one up isogeny $E \to E'$ of degree ℓ.*
  - *If $\ell \nmid \frac{c_\pi}{c_E}$, there are no other isogenies $E \to E'$ of degree ℓ over $\mathbb{F}_q$.*
  - *If $\ell \mid \frac{c_\pi}{c_E}$, there are ℓ down isogenies of degree ℓ.*

Proof. [Koh96, §4.2], [FM02, §2.1] or [Gal99, §11.5]. □

In light of Theorem 2.4.7, we say that ℓ is an *Elkies prime* if $\left(\frac{\Delta}{\ell}\right) = 1$ (implying $\ell \nmid c_E$), or equivalently if and only if E admits exactly two horizontal isogenies of degree ℓ. (Some authors also allow $\left(\frac{\Delta}{\ell}\right) = 0$, but we do not need this case.)

For the rest of this thesis we will only work with horizontal isogenies over finite fields. That is, unless otherwise stated all definitions and theorems are restricted in scope to horizontal isogenies.

Definition 2.4.8. Let E_1, E_2, E_3 be elliptic curves over $\mathbb{F}_q$. Let $\phi\colon E_1 \to E_2$, and $\phi'\colon E_1 \to E_3$ be isogenies over $\mathbb{F}_q$. We say that ϕ and ϕ' are *isomorphic* if there exists an isomorphism $\eta\colon E_2 \to E_3$ such that

$$\eta \circ \phi = \phi'.$$

By the theory of complex multiplication [Cox89], every elliptic curve over $\mathbb{C}$ corresponds to a complex lattice, and every ordinary elliptic curve arises from the reduction of a complex elliptic curve modulo a prime ideal. The correspondence is as follows: for every E there exists a corresponding lattice $L_E \subset \mathbb{C}$ and an isomorphism $E \cong \mathbb{C}/L_E$. We can use this correspondence to represent kernels of isogenies as fractional ideals, as shown in the following theorems.

Theorem 2.4.9. *Let L be a lattice. Then for a number $\alpha \in \mathbb{C}\backslash\mathbb{Z}$, the following statements are equivalent:*

(a) $\alpha L \subset L$.

(b) There is an order $\mathcal{O}$ in an imaginary quadratic field K such that $\alpha \in \mathcal{O}$ and $L = \beta I$ for some $\beta \in \mathbb{C}$ and some proper fractional $\mathcal{O}$-ideal I.

Proof. [Cox89, Theorem 10.14]. □

Theorem 2.4.10. *Let $\phi\colon E \to E'$ be a (horizontal) isogeny. Then the points in* $\ker\phi$ *correspond to a fractional ideal of* $\operatorname{End}(E)$ *under the isomorphism $E \cong \mathbb{C}/L_E$.*

Proof. The proof follows from Theorem 2.4.9. Specifically, let $\mathrm{End}(E) = \mathcal{O}_D$, $\alpha = \frac{D+\sqrt{D}}{2}$, and $\Phi = \ker\phi$. Observe that Φ lifts to $L_{E'}$ under the isomorphism $E \cong \mathbb{C}/L_E$. We will identify $\ker\phi$ with $L_{E'}$ via this transformation. Theorem 2.4.9(b) implies that $\ker\phi = \beta I$, where I is a (proper) fractional ideal of some order $\mathcal{O}$, such that $\mathrm{End}(E) \subset \mathcal{O}$. To show that $\ker\phi$ is itself a fractional $\mathcal{O}$-ideal, it is enough to show that $\ker\phi \subset \frac{1}{n}\mathrm{End}(E)$ for some integer n. But this relationship clearly holds for $n = \deg\phi$.

We now show that $\mathcal{O} \subset \mathrm{End}(E)$. We assume the opposite and proceed by contradiction. Choose $\alpha' \in \mathcal{O} \setminus \mathrm{End}(E)$. In that case, Theorem 2.4.9(b) holds for α', and thus Theorem 2.4.9(a) implies that $\alpha'\Phi = \Phi$, or that $\mathcal{O} \subset \mathrm{End}(E')$, which contradicts the fact that $\mathrm{End}(E') = \mathrm{End}(E)$. □

Theorem 2.4.11. *Let $\phi\colon E \to E'$ be an isogeny. Then, up to isomorphism, the ideal* $\ker\phi$ *uniquely determines ϕ.*

Proof. [Sil92, III.4.12]. □

The above two theorems are very useful in the sense that it is impractical to express isogenies algebraically. Rather than expressing the isogeny ϕ directly, we can represent it using its kernel $\ker\phi$.

We also have the following useful theorem:

Theorem 2.4.12. *Let E be a given elliptic curve. There is a natural 1-1 correspondence between proper ideals $\mathfrak{a}, \mathfrak{b} \subset$* $\mathrm{End}(E)$ *and horizontal isogenies $\phi_{\mathfrak{a}}$ and $\phi_{\mathfrak{b}}$ (up to isomorphism of isogenies) between the corresponding curves. As a result, we also have:*

- $\phi_{\mathfrak{ab}} = \phi_{\mathfrak{a}} \circ \phi_{\mathfrak{b}}$.
- $\deg\phi_{\mathfrak{a}}$ *equals the norm of* $\mathfrak{a}$.

Proof. For the case when $\mathrm{End}(E)$ is a maximal order see [Sil94, II.1.2]. For more general cases see [Lan87]. □

This theorem shows that using ideals to represent isogenies does not affect the main arithmetic properties of isogenies.

2.5 Application: Point Counting

In this section we look at one of the applications of isogenies — counting the number of points on an elliptic curve over a finite field. The results in this section are from Schoof [Sch95].

We let $E : y^2 = x^3 + ax + b$ be an elliptic curve defined over the finite field $\mathbb{F}_p$ where p is a prime (this can be done over some field of size q, where q is a power of a prime, but to make the material easier to read and to follow [Sch95], we work over $\mathbb{F}_p$).

Observe that for a given $x \in \mathbb{F}_p$, there are either 0, 1 or 2 elements of E of the form (x, y), depending on whether $x^3 + ax + b$ is a square modulo p. Thus we can say that the number of points of the form (x, y) for a fixed x is

$$1 + \left(\frac{x^3 + ax + b}{p} \right),$$

where the parentheses denote the Legendre symbol. We also have one point at infinity. Thus,

$$\#E(\mathbb{F}_p) = 1 + p + \sum_{x \in \mathbb{F}_p} \left(\frac{x^3 + ax + b}{p} \right).$$

Therefore computing $\#E(F_p)$ is equivalent to computing

$$\sum_{x \in \mathbb{F}_p} \left(\frac{x^3 + ax + b}{p} \right).$$

When p is relatively small, this can be done directly. However when p is large then this approach is very inefficient. Hence, we must look for a different approach.

Recall that, when we let $t = \text{Trace}(\pi_p)$ (i.e. t is the trace of E), then

$$\#E(\mathbb{F}_p) = p + 1 - t.$$

Thus, we can see that computing $\#E(\mathbb{F}_p)$ is equivalent to computing t. We have a theorem that gives an upper and lower bound on t and $\#E(\mathbb{F}_p)$:

Theorem 2.5.1. *Let E be an elliptic curve defined over the field $\mathbb{F}_p$. Then*

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

Proof. [Sil92, V.1.1]. □

The way the algorithm works is that we want to compute the value of t modulo a set of small primes ℓ and then use the Chinese Remainder Theorem to obtain t. Using Theorem 2.5.1 it suffices to perform this computation using primes of size at most $O(\log p)$.

We can easily do this for $\ell = 2$. Indeed, E contains a point of order 2 if and only if $2 \mid \#E(\mathbb{F}_p)$. Any point of order two is of the form $(x, 0)$. Finding such points is equivalent to checking whether $x^3 + ax + b$ has zeros in $\mathbb{F}_p$ (it does not matter how many zeros there are, only whether there are zeros), which is equivalent to checking

$$\gcd(x^p - x, x^3 + ax + b) \neq 1 \in \mathbb{F}_p[x].$$

This can be done efficiently by evaluating x^p modulo $x^3 + ax + b$ using repeated squaring. Now we want to compute t modulo other small primes $\ell = 3, 5, 7, \ldots$. We have that [Sil92, III.6.4]

$$E[\ell] = \{P \in E(\bar{\mathbb{F}}_p) : \ell P = \infty\} \cong \mathbb{Z}_\ell \times \mathbb{Z}_\ell,$$

and that π_p satisfies the equation

$$\pi_p^2 - t\pi_p + p = 0.$$

We also have the so-called *division polynomials*

$$\Psi_\ell(x) \in \mathbb{F}_p[x],$$

that vanish precisely in the ℓ-torsion points. The degree of such a polynomial is $(\ell^2 - 1)/2$. The division polynomials can be constructed via an explicit recurrence formula [Sil92, p. 105] at a cost of $O(\ell^2)$ multiplications.

The way the algorithm proceeds is that we check for which value of $t' \in \{0, 1, \ldots, \ell - 1\}$ the equation

$$\pi_p^2 - t'\pi_p + p = 0$$

holds on the group $E[\ell]$. Once that t' is found, then we know that $t' \equiv t \mod \ell$. We can do this efficiently by expressing the relationship in terms of polynomials. We have that

$$\pi_p^2(x, y) + p(x, y) = t'\pi(x, y) \text{ for all } (x, y) \in E[\ell]$$

if and only if

$$(x^{p^2}, y^{p^2}) + p'(x, y) \equiv t'(x^p, y^p) \mod (\Psi_\ell(x), y^2 - x^3 - ax - b),$$

where p' is an integer congruent to $p \mod \ell$ satisfying $0 \leq p' < \ell$. Note that by '+', we mean the group addition; similarly for multiplication by an integer. Such an approach gives the algorithm a total running time of

$$O((\log p)^8).$$

Although this running time is polynomial, in practice the algorithm performs poorly due to the large degrees of division polynomials. We now describe a practical improvement to the algorithm using isogenies, due to Elkies.

We view the Frobenius map π_p as a 2×2 matrix. For primes ℓ that split in $\mathbb{Q}(\sqrt{t^2 - 4p})$, the Frobenius map acts on $E[\ell]$ and has eigenvalues in $\mathbb{F}_\ell$. In this case, there exists an eigenspace H such that $\#H = \ell$, and H is Galois invariant. Note that H corresponds to an ideal in $\mathrm{End}(E)$ of norm ℓ. Now, let $f(x) \in \mathbb{F}_p[x]$ be a polynomial of degree $(\ell - 1)/2$ whose zeros are the distinct x-coordinates of points in H. We compute the eigenvalue λ that corresponds to H. Note that the product of eigenvalues is equal to $p \bmod \ell$. Thus we obtain

$$t \equiv \lambda + p/\lambda \mod \ell.$$

To compute λ, we use the fact that it satisfies

$$\pi_p(x, y) = (x^p, y^p) = \lambda \cdot (x, y) \mod f(x) \text{ (in } E).$$

We simply try this equation for all values $\lambda' = 1, 2, \ldots, \ell - 1$, and see which value satisfies the equation.

In order to implement the above approach, we must compute the coefficients of $f(x)$. This computation is done using standard Elkies-Atkin techniques (see Section 3.2). This approach heuristically lowers the cost of calculating $t \bmod \ell$ by a factor of $O((\log p)^2)$, for an overall cost of $O((\log p)^6)$ bit operations.

2.6 Application: Transfer of Discrete Logarithms

In this section we show how isogenies can be used to transfer the discrete logarithm problem from one elliptic curve to another. Although our work does not directly yield improved transfer algorithms, we include this material to indicate some past applications of isogenies and to lay the groundwork for discussing possible future applications. Before we can discuss transfer, we need some background material on the discrete logarithm problem.

Definition 2.6.1. Let G be a group. Let $g \in G$ be an element of order r and $x \in \mathbb{Z}_r$. Define $h = g^x$. Then *Discrete Logarithm (or DLOG) Problem* is defined as follows:

$$\text{Given } g, h \in G, \text{ find } x \in [0, r-1] \text{ such that } h = g^x.$$

Typically, in applications we choose an element g such that $\mathrm{ord}(g) \approx \#G$. We also assume from now on that G is abelian.

Example 2.6.2. *Diffie-Hellman Key Exchange Protocol*

- Let G be a group.
- Select an element $g \in G$ of order r. Note: G, g and r are public knowledge.
- Alice selects *private key* $a \in \mathbb{Z}_r$ and computes her *public key* $A = g^a$. Alice sends A to Bob.
- Bob selects *private key* $b \in \mathbb{Z}_r$ and computes his *public key* $B = g^b$. Bob sends B to Alice.
- Alice computes *key* $g^{ab} = B^a$.
- Bob computes *key* $g^{ab} = A^b$.
- Alice and Bob have established the secure communication key g^{ab}.
- This protocol can be summarized in the following figure:

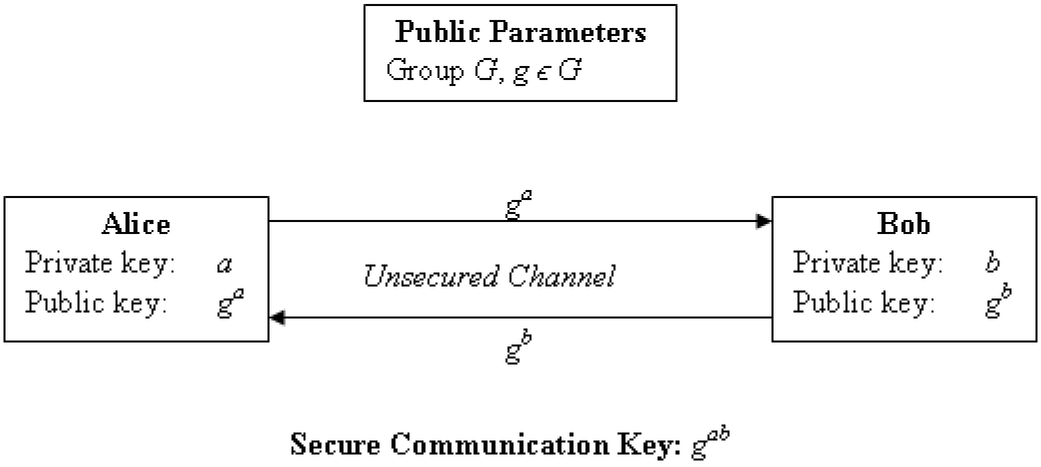

Figure 2.2: Diffie-Hellman Key Exchange Protocol

Using the above example as motivation, we obtain the following definition:

Definition 2.6.3. Let G be a group. Let $g \in G$ be an element of order r and $x, y \in \mathbb{Z}_r$. Define $h_1 = g^x, h_2 = g^y$. Then *Diffie-Hellman Problem (DHP)* is defined as follows:

$$\text{Given } g, h_1 = g^x, h_2 = g^y \in G, \text{ compute } h = g^{xy}.$$

Observe that if one solves the DHP, then the Diffie-Hellman Key Exchange Protocol is broken. Also note that DHP reduces to DLOG problem in polynomial time; that is if we are able to solve DLOG problem then we can easily solve DHP:

- Solve DLOG problem for $g, h_1 \in G$, to obtain x.
- Compute $h_2^x = (g^y)^x = g^{xy}$.

Many applications in cryptography are based on the hardness of solving the DLOG problem in a given group. One question that arises frequently in cryptography is the question of how vulnerable a given group is to DLOG. In some cases, one can relate the DLOG problems on different groups. For example, assume that we have two groups G and G'. Let $\varphi\colon G \to G'$ be an injective homomorphism. If φ is efficiently computable, then we can *transfer* the DLOG problem on G to DLOG problem on G', in the following sense:

- Let $g, h = g^x \in G$ be given. We wish to compute x.
- Evaluate $\varphi(g), \varphi(h) \in G'$. Observe that $\varphi(h) = \varphi(g^x) = \varphi(g)^x$.
- Solve DLOG problem in G' for $\varphi(g), \varphi(h)$. The result is x.

Thus if we want to solve DLOG problem in G, and we can efficiently find an easy to evaluate (injective) homomorphism from G to some other group G', where DLOG problem is easier to solve, then we can transfer the problem to G'. Hence the DLOG problem on G is no harder than on G'. It is not strictly necessary for the homomorphism φ to be injective. If $g \notin \ker\varphi$, then we still gain a lot of information that could aid us in solving DLOG problem in G.

The best known solution to DLOG problem for a general group G with element $g \in G$ of order r is known as Pollard's rho method. This method requires $O(\sqrt{\pi r}/2)$ operations (details can be found in [Pol78]). We consider a group G to be secure if no faster algorithm is known for solving DLOG problem.

We now return to elliptic curves over finite fields. There are some elliptic curves for which the DLOG problem can be solved significantly faster than Pollard's rho method due to the fact that it can be transfered to the Jacobian of a hyperelliptic curve of a certain genus (see [CFA$^+$06, Section 22.3] for details); this is known as a *Weil descent attack*. However, there are elliptic curves which are not vulnerable to that attack. The interesting fact is that there are pairs of isogenous curves such that one is directly vulnerable to the above attack and the other is not. Using this idea, E. Teske has introduced a trapdoor discrete logarithm system [Tes06]. We now present this scheme as an application of isogenies.

Before we continue, we need to describe the idea of *key escrow.* The basic idea is that users use their keys for secure communication, and sometimes some officials (possibly government) want to listen to their communication. However, without any extra information, this is impossible. To support this capability, there exists an escrow agency to which the users submit some extra information. When the escrow agency gets an official request from officials to disclose information about a certain user, they provide that information. Once the officials have that information, they can use it to decrypt encrypted messages sent by the user(s).

The idea of this application is to construct a cryptosystem such that the information submitted to the escrow agencies is sufficient to recover the user's private key; however it would still take a considerable, although feasible, amount of computational effort to do so.

The scheme is based on elliptic curves defined over the field $F = \mathbb{F}_{2^{161}}$. Although the construction works in general over a large class of fields, this field is the only field whose computational requirements are well matched to current technology. Note that since $\operatorname{char} F = 2$, we need to use the following form of Weierstrass equation:

$$E_{a,b} : y^2 + xy = x^3 + ax^2 + b.$$

Here, $a, b \in \mathbb{F}_{2^{161}}$; and for the purposes of the current application we also restrict $b \neq 0$.

We also need the notion of *magic number.* Given $b \in \mathbb{F}_{2^N}$, where N is composite, $N = nf$, we let $q = 2^f$. We also let $b_i = b^{q^i}$. Then we define the magic number to be:

$$m = m_n(b) = \dim_{\mathbb{F}_2}(\operatorname{Span}\{(1, b_0^{1/2}), \ldots, (1, b_{n-1}^{1/2})\}).$$

In our case, $N = 161, n = 7, f = 23$. This yields that for $b \in \mathbb{F}^*_{2^{161}}$, $m_7(b) \in \{1, 4, 7\}$. For approximately 2^{93} values of $b \in \mathbb{F}^*_{2^{161}}$, $m_7(b) = 4$. For approximately 2^{23} values of $b \in \mathbb{F}^*_{2^{161}}$, $m_7(b) = 1$. Hence for the overwhelming majority of $b \in \mathbb{F}^*_{2^{161}}$ we have that $m_7(b) = 7$.

For a given elliptic curve $E_{a,b}$, if $m_7(b) = 4$, then the Weil descent attack applies; if $m_7(b) = 7$, the Weil descent attack gives us an even less efficient solution to the DLOG problem than Pollard's rho method.

We define the following set:

$$I_4 = \{E_{a,b}/\mathbb{F}_{2^{161}} : a \in \{0, 1\}, m_7(b) = 4\}$$

to be the set of representatives of isomorphism classes of elliptic curves with magic number 4. It is a fact that the magic number is invariant under isomorphisms and scalar multiplication isogenies; however, in general the magic number changes under isogenies.

For an elliptic curve to be cryptographically interesting, we want $\#E(\mathbb{F}_{2^{161}}) = 2{\cdot}p$ or $4{\cdot}p$ (where p is a prime). The reason is that we want a selected point $P \in E(\mathbb{F}_{2^{161}})$ to have a large order ($\approx \#E(\mathbb{F}_{2^{161}})$).

Hence, the idea is that we need to choose a pair of isogenous (via horizontal isogeny) elliptic curves $(E_{\text{secret}}, E_{\text{public}})$, such that E_{secret} has magic number equal to 4, while E_{public} has magic number equal to 7.

We need a few more restrictions on the endomorphism ring of (E_{secret} and E_{public}) (note that they have the same endomorphism ring, which we denote by $\mathcal{O}_\Delta$), to make certain computations appropriately feasible (for details on reasons for these restrictions see [Tes06]):

- Δ is squarefree.
- $|\Delta| > 2^{157}$.
- $2^{76} \leq \#\,\mathrm{Cl}(\mathcal{O}_\Delta) < 2^{83}$, where $\mathrm{Cl}(\mathcal{O}_\Delta)$ denotes the ideal class group of $\mathcal{O}_\Delta$.
- The odd, cyclic part of $\mathrm{Cl}(\mathcal{O}_\Delta)$ has cardinality $\geq 2^{68}$.

Once the curve E_{secret} is found, then we need to find the isogenous curve E_{public}. Note that we should involve randomness in this process in order to avoid attacks and at the same time knowing enough information, it should be feasible to transfer the DLOG problem from E_{public} to E_{secret}. In order to do that, a set of possible horizontal isogenies is selected and lengths of chains of isogenies are randomly chosen.

The entire process is implemented using Algorithm 1.

After running this algorithm the user submits E_{secret} and the set $\mathcal{C}$ to the escrow agency and uses E_{public} for ECC. In case the officials need to listen in to the user's communication, they can transfer DLOG from E_{public} to E_{secret} via the sequence of isogenies that they compute iterating through $\mathcal{C}$, and then use the Weil descent attack on E_{secret} to obtain the user's private key. Note that there is a variant of this scheme, which provides greater security for the user. The only thing that changes in the variant is that the user only submits E_{secret} to the escrow agency. In case the officials need to obtain the user's private key, it would take them longer to recover it, but they can still do so in a feasible number of steps.

Algorithm 1 Trapdoor curves

```
Input: Field $F = \mathbb{F}_{2^{161}}$ and the set $I_4$.
Output: A pair of isogenous elliptic curves $(E_{\text{secret}}, E_{\text{public}})$ over $\mathbb{F}_{2^{161}}$, where magic number
    of $E_{\text{secret}}$ is 4 and magic number of $E_{\text{public}}$ is 7.
 1: Randomly choose $E \in I_4$ such that $\#E(\mathbb{F}_{2^{161}}) = 2 \cdot p$ or $4 \cdot p$ for some prime $p$.
 2: Calculate the discriminant of $E$, $\Delta = t^2 - 4 \cdot 2^{161}$, where $t = 2^{161} + 1 - \#E(\mathbb{F}_{2^{161}})$ is
    the trace of $E$.
 3: Check that $\Delta$ is squarefree, if not go to STEP 1.
 4: Check that $|\Delta| > 2^{157}$, if not go to STEP 1.
 5: Check that $2^{76} \leq \# \operatorname{Cl}(\mathcal{O}_\Delta) < 2^{83}$, if not go to STEP 1.
 6: Check that the odd, cyclic part of $\operatorname{Cl}(\mathcal{O}_\Delta)$ has cardinality $\geq 2^{68}$, if not go to STEP 1.
 7: Denote $E_{\text{secret}} = E$.
 8: Obtain $\mathcal{F} = \{l : l \text{ is prime}, 3 \leq l \leq 300, l \text{ splits in } \mathcal{O}_\Delta\}$.
 9: Enumerate $\mathcal{F} = \{l_1, l_2, \ldots, l_{\#\mathcal{F}}\}$.
10: Let $E = E_{\text{secret}}$.
11: Let $\mathcal{C} = \emptyset$.
12: for i=1, ..., $\#\mathcal{F}$ do
13:     Randomly choose $n_1 \in \{0, 1, \ldots 11\}$.
14:     Construct a chain of length $n_i$ of $l_i$-isogenous curves starting from $E$.
15:     Add the resulting curve to $\mathcal{C}$.
16:     Denote the resulting curve by $E$.
17: end for
18: Check that the magic number for $E$ is 7, if not go to STEP 12.
19: Let $E_{\text{public}} = E$
20: Output $(E_{\text{secret}}, E_{\text{public}})$ and $\mathcal{C}$.
```

Chapter 3

Previous Methods for Evaluating Isogenies

In this chapter we examine previous methods used to evaluate isogenies. The problem of computing isogenies between elliptic curves over finite fields reduces to the problem of computing prime degree isogenies. This reduction proceeds as follows: given an isogeny, factor the kernel of the isogeny into a composition series with prime order quotient groups and apply Prop. 2.3.3 to each composition factor. The resulting sequence of prime degree isogenies, when composed together, yields the original isogeny. For this reason, we will only consider prime degree isogenies.

As usual, given an elliptic curve E over the finite field $\mathbb{F}_q$, we wish to find an isogeny $\phi_\ell \colon E \to E'$ of given degree ℓ, and to evaluate $\phi_\ell(P)$ for points $P \in E$.

We first examine Vélu's formulas for computing the isogeny and the isogenous curve, given the kernel of ϕ_ℓ as a subgroup. Then we discuss Elkies-Atkin techniques, which are based on modular polynomials.

3.1 Vélu's Formulas

The material in this section comes from [Vél71].

Let ℓ be a given prime. Let E be an elliptic curve defined over $\mathbb{F}_q$. To maintain consistency with Vélu, we use the general Weierstrass equation for E, valid in all characteristics:

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6.$$

Let H be a subgroup of E, with $\#H = \ell$. We wish to obtain the algebraic form of ϕ_ℓ and the equation for E', where $\phi_\ell \colon E \to E'$ is the unique (up to isomorphism) separable isogeny having kernel H. Vélu obtains these formulas using the theory of elliptic functions. The derivation of these formulas is beyond the scope of this thesis, but we provide the results here for reference.

Let H_2 denote the set of points of order 2 in $H \setminus \{\infty\}$. Now let R be any subset of $(H \setminus \{\infty\}) \setminus H_2$ such that

$$(H \setminus \{\infty\}) \setminus H_2 = R \cup (-R) \text{ and } R \cap (-R) = \emptyset$$

(where by $-R$ we denote $-R = \{-P : P \in R\}$). Finally, we let $S = H_2 \cup R$.

Given a point $Q = (x_Q, y_Q)$, we calculate the following values:

$$\begin{aligned}
g_Q^x &= 3x_Q^2 + 2a_2x_Q + a_4 - a_1y_Q, \\
g_Q^y &= -2y_Q - a_1x_Q - a_3, \\
b_2 &= a_1^2 + 4a_2, \\
b_4 &= a_1a_3 + 2a_4, \\
b_6 &= a_3^2 + 4a_6, \\
t_Q &= \begin{cases} g_Q^x & \text{if } Q \in H_2, \\ 2g_Q^x - a_1g_Q^y = 6x_Q^2 + b_2x_Q + b_4 & \text{if } Q \notin H_2, \end{cases} \\
u_Q &= (g_Q^y)^2 = 4x_Q^3 + b_2x_Q^2 + 2b_4x_Q + b_6.
\end{aligned}$$

So, we get the explicit equation of the isogeny,

$$\phi_\ell(x, y) = (X, Y),$$

where

$$\begin{aligned}
X &= x + \sum_{Q \in S} \left(\frac{t_Q}{x - x_Q} + \frac{u_Q}{(x - x_Q)^2} \right), \\
Y &= y - \sum_{Q \in S} \left(u_Q \frac{2y + a_1x + a_3}{(x - x_Q)^3} + t_Q \frac{a_1(x - x_Q) + y - y_Q}{(x - x_Q)^2} + \frac{a_1u_Q - g_Q^x g_Q^y}{(x - x_Q)^2} \right).
\end{aligned}$$

Now, to obtain the equation of E', we calculate

$$t = \sum_{Q \in S} t_Q,$$

and

$$w = \sum_{Q \in S} (u_Q + x_Q t_Q).$$

We also have

$$\begin{aligned}&A_1 = a_1, A_2 = a_2, A_3 = a_3,\\&A_4 = a_4 - 5t, A_6 = a_6 - b_2t - 7w.\end{aligned}$$

The equation of E' is:

$$Y^2 + A_1XY + A_3Y = X^3 + A_2X^2 + A_4X + A_6.$$

The computational cost of evaluating Vélu's formulas is $O(\ell^2 \log \ell)$ multiplications in $\mathbb{F}_q$ [IJ, Section 5.1].

3.2 Elkies-Atkin Techniques

The Elkies-Atkin technique for computing isogenies is based on modular polynomials. It is the fastest known algorithm for direct computation. In this section we follow [Sch95, Sections 6,7,8] and [BCL08, Section 3.1]. We let

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

be an elliptic curve defined over a finite field $\mathbb{F}_q$. Let ℓ be the degree of the isogeny that we wish to evaluate. Denote that isogeny by $\phi_\ell \colon E \to E'$. Here we assume that ℓ is the Elkies prime, that is, it splits in $\mathrm{End}(E)$. Hence, we have the factorization

$$(\ell) = \mathfrak{L}\bar{\mathfrak{L}}$$

in $\mathrm{End}(E)$. Without loss of generality, we assume $\ker \phi_\ell$ corresponds to $\mathfrak{L}$. For simplicity, we assume that $\mathrm{char}\, \mathbb{F}_q > \ell \geq 3$, and also that $\mathrm{End}(E)$ is not equal to $\mathbb{Z}[i]$ nor $\mathbb{Z}[e^{2\pi i/3}]$.

We now define the *classical modular polynomial of level ℓ*. We give only a brief definition here; the full definition can be found in [Cox89, p. 230]. The modular polynomial $\Phi_\ell(X, Y) \in \mathbb{Z}[X, Y]$ is defined to be the unique polynomial satisfying

$$\Phi_\ell(X, j(\tau)) = \prod_{i=1}^{[\mathrm{SL}_2(\mathbb{Z}):\Gamma_0(\ell)]} (X - j(\ell\gamma_i\tau))$$

for all $\tau \in \mathbb{C}$ with positive imaginary part, where $\{\gamma_i\}$ forms a set of right coset representatives for $[\mathrm{SL}_2(\mathbb{Z}) : \Gamma_0(\ell)]$. Here $j(\tau)$ denotes the j-invariant of the elliptic curve corresponding to the lattice generated by 1 and τ. The polynomial $\Phi_\ell(X, Y) \in \mathbb{Z}[X, Y]$ is symmetric in X and Y, and has terms $X^{\ell+1} - X^\ell Y^\ell + Y^{\ell+1} + \ldots$ plus terms of the form X^iY^j such that $0 \leq i, j \leq \ell$ and $i + j < 2\ell$. It has the property that the zeros of $\Phi(j(E), Y) = 0$ are precisely the set of j-invariants of curves ℓ-isogenous to E. Over $\mathbb{F}_q$, when ℓ is an Elkies prime, it will have 2 roots, where one corresponds to the j-invariant of the codomain of ϕ_ℓ and the other one corresponds to the j-invariant of the codomain of the other isogeny of degree ℓ.

Example 3.2.1. *Classical modular polynomial of level 3*

$$\begin{aligned}\Phi_3(X,Y) = {} & X^4 - X^3Y^3 + 2232X^3Y^2 - 1069956X^3Y + 36864000X^3 + 2232X^2Y^3 \\ & + 2587918086X^2Y^2 + 8900222976000X^2Y + 452984832000000X^2 \\ & - 1069956XY^3 + 8900222976000XY^2 - 770845966336000000XY \\ & + 1855425871872000000000X + Y^4 + 36864000Y^3 \\ & + 452984832000000Y^2 + 1855425871872000000000Y\end{aligned}$$

Now, we solve in $\mathbb{F}_q$ the equation

$$\Phi_\ell(j(E), Y) = 0$$

for Y. Let h be a solution (in $\mathbb{F}_q$). Note that there are two possible solutions. At this point we don't know which one to take, so we pick one randomly and then check at the end whether our choice is correct by testing whether $f_C = f_{\mathfrak{L}}$ (see below) — if it turns out to be the wrong one, we then redo the process from this point with the other solution. We set

$$\begin{aligned} s &= -\frac{18b}{\ell a} \frac{\frac{\partial \Phi}{\partial X}(j(E), h)}{\frac{\partial \Phi}{\partial Y}(j(E), h)} j(E) \in \mathbb{F}_q \\ a' &= -\frac{1}{48} \frac{s^2}{h(h - 1728)} \in \mathbb{F}_q \\ b' &= -\frac{1}{864} \frac{s^3}{h^2(h - 1728)} \in \mathbb{F}_q \end{aligned}$$

Then there exists an ℓ-isogeny $\phi_h \colon E \to E_h$ where the equation of the isogenous curve E_h is

$$E_h : y^2 = x^3 + a'x + b'.$$

Now, we need to figure out which of the two ideals $\mathfrak{L}$ or $\bar{\mathfrak{L}}$ corresponds to the root h. Let $C = \ker \phi_h$. We define $f_C(x)$ to be the polynomial whose zeros are the distinct x-coordinates of points in C. We define $f_{\mathfrak{L}}(x)$ to be the polynomial whose zeros are the distinct x-coordinates of points in $E[\mathfrak{L}]$, where

$$E[\mathfrak{L}] = \{P \in E : \phi(P) = \infty \text{ for all } \phi \in \mathfrak{L}\}.$$

$f_{\bar{\mathfrak{L}}}$ is defined in the similar manner. So, the goal is to figure out whether $f_C = f_{\mathfrak{L}}$ or $f_{\bar{\mathfrak{L}}}$. The Frobenius map π_q acts on the points in $E[\mathfrak{L}]$ as multiplication by $-c/d$, where $c, d \in \mathbb{Z}$ satisfy $\mathfrak{L} = (\ell, c + d\pi_q)$. We test whether $(x^q, y^q) \overset{?}{=} (-c/d)(x, y)$ holds in C. If so, then $f_C = f_{\mathfrak{L}}$, if not, then the j-invariant of the isogenous curve must be the other root of $\Phi_\ell(j(E), Y)$ in $\mathbb{F}_q$. Once we know f_C, we can compute the explicit equation for ϕ_ℓ and evaluate $\phi_\ell(P)$ using Vélu's formulas.

To compute the equation for ϕ_ℓ, we need to compute the coefficients of $f_C(x)$. We will sketch this computation in the case that E is defined over $\mathbb{F}_p$. For more details see [Sch95, Section 7,8] and [BMSS08].

Let p_1 be the sum of the roots of $f_C(x)$. The value of p_1 can be obtained using power series manipulations [Sch95, Section 7]. We define the values of c_i as follows (all computations take place in $\mathbb{F}_p$):

$$\begin{aligned} c_1 &= -\frac{a}{5}, \\ c_2 &= -\frac{b}{7}, \\ c_k &= \frac{3}{(k-2)(2k+3)} \sum_{j=1}^{k-2} c_j c_{k-1-j}, \text{ for } k \geq 3. \end{aligned}$$

We also define:

$$\begin{aligned} c'_1 &= -\frac{1}{5}\frac{a'}{\ell^4}, \\ c'_2 &= -\frac{1}{7}\frac{b'}{\ell^6}, \\ c'_k &= \frac{3}{(k-2)(2k+3)} \sum_{j=1}^{k-2} c'_j c'_{k-1-j}, \text{ for } k \geq 3. \end{aligned}$$

Using these we can compute the coefficients a_i of $f_C(x) = x^{(\ell-1)/2} + a_{\frac{\ell-3}{2}} x^{(\ell-3)/2} + \ldots + a_0$. We have that [Sch95, 8.3]

$$z^{\ell-1} f(\wp(z)) = \exp\left(-\frac{1}{2}p_1 z^2 - \sum_{k=1}^{\infty} \frac{c'_k - \ell c_k}{(2k+1)(2k+2)} z^{2k+2}\right),$$

where z is a variable and

$$\wp(z) = \frac{1}{z^2} + \sum_{k=1}^{\infty} c_k z^{2k}.$$

We match coefficients to obtain the coefficients of $f_C(x)$:

$$\begin{aligned}
a_{\frac{\ell-3}{2}} &= -\frac{p_1}{2}, \\
a_{\frac{\ell-5}{2}} &= \frac{1}{8}p_1^2 - \frac{c_1' - \ell c_1}{12} - \frac{\ell-1}{2}c_1, \\
a_{\frac{\ell-7}{2}} &= -\frac{1}{48}p_1^3 - \frac{c_2' - \ell c_2}{30} + p_1\frac{c_1' - \ell c_1}{24} - \frac{\ell-1}{2}c_2 + \frac{\ell-3}{4}c_1 p_1, \\
&\vdots
\end{aligned}$$

Finally, use Vélu's formulas to obtain the explicit form of ϕ_ℓ, and evaluate it. This approach for evaluating prime degree isogenies has a running time complexity of $O(\ell^3(\log \ell)^{4+\epsilon})$.

3.2.1 Small Characteristic Case

Note that for the previous method we had a restriction

$$\operatorname{char} \mathbb{F}_q > \ell \geq 3.$$

This might cause difficulties when we work with elliptic curves defined over fields of small characteristic. This issue is resolved in [LS08] and [def], where an algorithm is given for evaluating isogenies over small characteristic fields, having the same running time complexity as in the large characteristic case. From [LS08], we obtain the following theorem.

Theorem 3.2.2. *Let E be an elliptic curve defined over the finite field $\mathbb{F}_q$. Assume that $\operatorname{char} \mathbb{F}_q \geq 5$. Let ℓ be an Elkies prime, distinct from the characteristic of $\mathbb{F}_q$. Then there exists an algorithm that computes the polynomial $f_{\mathfrak{L}}(x)$ at cost $O(\ell(\max(\ell, \log q))^2 \log^2(\ell(\max(\ell, \log q))^2))$.*

Proof. [LS08] □

Lercier and Sirvent also mention that this theorem can easily be extended to the cases where the characteristic of the field is 2 or 3. De Feo in [def] presents the algorithms for computing isogenies over fields of characteristic 2 or 3.

3.3 Overview and Remarks on Evaluating Isogenies

In this section, we discuss the evaluation of isogenies of large degree. By 'large' we mean cryptographic size, i.e. $\ell \gtrapprox 2^{160}$. For such degrees, prior algorithms are infeasible since the running time is exponential in $\log \ell$. For example, using the Elkies-Atkin technique one must compute $\Phi_\ell(X,Y)$, which has a theoretical complexity of $O(\ell^{3+\epsilon})$. To illustrate that difficulty, recall that in Example 3.2.1 we gave $\Phi_3(X,Y)$. The next modular polynomial, $\Phi_5(X,Y)$ is as follows:

Example 3.3.1. *Classical modular polynomial of level 5*

$$\begin{aligned}
\Phi_5(X,Y) = {} & X^6 - X^5Y^5 + 3720X^5Y^4 - 4550940X^5Y^3 + 2028551200X^5Y^2 \\
& - 246683410950X^5Y + 1963211489280X^5 + 3720X^4Y^5 \\
& + 1665999364600X^4Y^4 + 107878928185336800X^4Y^3 \\
& + 383083609779811215375X^4Y^2 + 128541798906828816384000X^4Y \\
& + 1284733132841424456253440X^4 - 4550940X^3Y^5 \\
& + 107878928185336800X^3Y^4 - 441206965512914835246100X^3Y^3 \\
& + 26898488858380731577417728000X^3Y^2 \\
& - 192457934618928299655108231168000X^3Y \\
& + 280244777828439527804321565297868800X^3 + 2028551200X^2Y^5 \\
& + 383083609779811215375X^2Y^4 + 26898488858380731577417728000X^2Y^3 \\
& + 5110941777552418083110765199360000X^2Y^2 \\
& + 36554736583949629295706472332656640000X^2Y \\
& + 6692500042627997708487149415015068467200X^2 - 246683410950XY^5 \\
& + 128541798906828816384000XY^4 \\
& - 192457934618928299655108231168000XY^3 \\
& + 36554736583949629295706472332656640000XY^2 \\
& - 264073457076620596259715790247978782949376XY \\
& + 53274330803424425450420160273356509151232000X \\
& + Y^6 + 1963211489280Y^5 + 1284733132841424456253440Y^4 \\
& + 280244777828439527804321565297868800Y^3 \\
& + 6692500042627997708487149415015068467200Y^2 \\
& + 53274330803424425450420160273356509151232000Y \\
& + 141359947154721358697753474691071362751004672000
\end{aligned}$$

It is clear that for large ℓ it is infeasible to compute $\Phi_\ell(X,Y)$. The world record for computing the classical modular polynomial over the integers is $\ell \approx 5000$ ($\ell \approx 20000$ over integers modulo a prime). Note that there do exist modular polynomials which are more efficient for computation; however asymptotically the overall algorithm still has the same time complexity for any choice of modular polynomial.

Certain large degree isogenies are easy to evaluate in any case. For example, consider isogenies of the form $[n]\colon E \to E$ (multiplication by n-map). These isogenies are easy to evaluate using the double-and-add method. In this case, we do not use the algebraic form of the multiplication by n-map, but rather just compute its effects on points.

As another example, if we consider inseparable isogenies, the inseparable part is equal to a power of the Frobenius map [Sil92, II.2.12], and Frobenius maps are also easy to evaluate. We simply need to compute x^q and y^q in the finite field $\mathbb{F}_q$. To do this, we use the well known method of square-and-multiply.

Other types of easy to evaluate isogenies include complex multiplication by a small discriminant and small degree isogenies. In addition, linear combinations and compositions of easy to evaluate isogenies are easy to evaluate. However, all other large degree isogenies are infeasible to evaluate using any of the obvious algorithms. In the following chapters, we will describe a new method for evaluating general large degree isogenies in subexponential time.

Chapter 4

The Bröker-Charles-Lauter Algorithm

In this chapter we describe the algorithm of Bröker, Charles, and Lauter [BCL08]. This algorithm represents a large improvement in the evaluation of large degree (horizontal) isogenies between elliptic curves when the endomorphism ring has small discriminant. The idea is to factor the large prime degree isogeny into a product of small prime degree isogenies and an isogeny corresponding to a principal fractional ideal. To accomplish this factorization, they work in the ideal class group of End(E). Their algorithm provides the basis for our algorithm, which we will present in the next chapter. In this chapter, we summarize the results of [BCL08].

4.1 Preliminaries

Recall that for an ordinary elliptic curve E defined over a finite field $\mathbb{F}_q$, we have $\mathrm{End}(E) \cong \mathcal{O}_\Delta$. We identify End$(E)$ with $\mathcal{O}_\Delta$ via the unique isomorphism ι satisfying $\iota^*(x)\omega = x\omega$ for all invariant differentials ω and all $x \in \mathcal{O}_\Delta$. Also, for split primes ℓ, recall that every horizontal separable isogeny ϕ_ℓ on E of prime degree ℓ corresponds (up to isomorphism) to a unique prime ideal $\mathfrak{L} \subset \mathcal{O}_\Delta$ of norm ℓ for some Elkies prime ℓ. We denote the kernel of such an isogeny by

$$E[\mathfrak{L}] = \{P \in E(\bar{\mathbb{F}}_q) : \alpha(P) = \infty \text{ for all } \alpha \in \mathfrak{L}\}.$$

Note that, for simplicity, we require ℓ to be an Elkies prime, which is equivalent to saying that $\ell \nmid [\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$. Also note that we can write $\mathfrak{L}$ explicitly as

$$\mathfrak{L} = (\ell, c + d\pi_q).$$

Observe that the image curve can be expressed as

$$E' = E/E[\mathfrak{L}].$$

However this gives us E' only up to isomorphism. Any two distinct isomorphic horizontal isogenies induce different maps on the space of differentials of E, and a separable isogeny is uniquely determined by the combination of its kernel and the induced map on the space of differentials. A *normalized* isogeny is an isogeny $\phi\colon E \to E'$ for which $\phi^*(\omega_{E'}) = \omega_E$ where ω_E denotes the invariant differential of E and $\omega_{E'}$ of E'. With this notation, our goal becomes to evaluate normalized horizontal prime degree isogenies.

4.2 The Method of Galbraith, Hess, and Smart

Galbraith, Hess, and Smart in [GHS02] present a method for evaluating (horizontal) isogenies. Their method is based on working with the ideals corresponding to these isogenies. The main idea of their method is to factor the given ideal into a product of prime ideals of small norm modulo a principal fractional ideal. In other words, they obtain a factorization of the form

$$[\mathfrak{L}] = [I_1]^{e_1} \cdot [I_2]^{e_2} \cdots [I_k]^{e_k},$$

where square brackets denote ideal classes, and the ideal classes $[I_j]$ are reduced. Then they recursively compute the isogenies corresponding to the ideals I_j using direct techniques.

This approach still has an exponential running time in the bitlength of the absolute value of the discriminant of the corresponding imaginary quadratic order. As mentioned in the previous section, we only obtain E' up to isomorphism and thus the isogeny obtained in this way is not normalized. This is acceptable for many applications, but for our purposes we wish to obtain the normalized isogeny.

4.3 Description of the Bröker-Charles-Lauter Algorithm

In the Bröker, Charles, and Lauter algorithm, the isogeny ϕ_ℓ is evaluated using a factorization of the kernel ideal of the form

$$\mathfrak{L} = I_1^{e_1} I_2^{e_2} \cdots I_k^{e_k} \cdot (\alpha),$$

where each ideal I_j corresponds to an ideal of prime norm, where the norm is bounded (i.e. small enough to make computations feasible), and (α) is a principal fractional ideal. This factorization is identical to that of Galbraith, Hess, and Smart, except that the principal fractional ideal (α) appears explicitly.

The fact that we obtain a factorization of ideals allows us to evaluate normalized isogenies, since the (α) factor determines the normalization. One restriction on the norms of I_j's is that they all must be Elkies primes. The isogeny that corresponds to (α) is easy to evaluate due to the fact that multiplication by scalars is easy to evaluate.

Obtaining a factorization of this form is difficult when working directly with ideals. Instead, Bröker, Charles, and Lauter (BCL) use the same technique as Galbraith, Hess, and Smart, except that they also obtain (α) and hence evaluate the normalized isogeny. The BCL algorithm begins by finding a factorization in $\mathrm{Cl}(\mathcal{O}_\Delta)$ of the following form:

$$[\mathfrak{L}] = [I_1]^{e_1} \cdot [I_2]^{e_2} \cdots [I_k]^{e_k}.$$

Once this factorization is obtained, the value of α is computed as follows:

- Compute
$$\mathfrak{L}\bar{I_1}^{e_1}\bar{I_2}^{e_2}\cdots\bar{I_k}^{e_k}.$$
- This is equal to
$$I_1{}^{e_1}\bar{I_1}^{e_1}I_2{}^{e_2}\bar{I_2}^{e_2}\cdots I_k{}^{e_k}\bar{I_k}^{e_k}\cdot(\alpha).$$
- Which is equal to the principal ideal
$$m(\alpha),\ \text{ where } m = \mathrm{Norm}(I_1)^{e_1}\cdot\mathrm{Norm}(I_2)^{e_2}\cdots\mathrm{Norm}(I_k)^{e_k}.$$
- Thus using using Cornacchia's Algorithm [HMW90], we obtain the generator of β of
$$\mathfrak{L}\bar{I_1}^{e_1}\bar{I_2}^{e_2}\cdots\bar{I_k}^{e_k}.$$
- Hence $\alpha = \beta/m$.

Once we have the full factorization, we compute the isogenies corresponding to I_j recursively. Let ϕ_{p_1} be the isogeny corresponding to I_1:

$$\phi_{p_1}\colon E \to E_1 = E/E[I_1].$$

Now let ϕ_{p_2} be the isogeny starting from E_1 corresponding to I_2:

$$\phi_{p_2}\colon E_1 \to E_2 = E_1/E_1[I_2] \cong E/E[I_1I_2].$$

Thus we continue recursively in the same manner to obtain the isogeny corresponding to $I_1^{e_1}I_2^{e_2}\cdots I_k^{e_k}$:

$$\phi_c\colon E \to E_c = E/E[I_1^{e_1}I_2^{e_2}\cdots I_k^{e_k}].$$

At each step, since the norm of each ideal I_j is small, we can use direct methods to evaluate the individual isogenies. Next, note that $E_c \cong E/E[\mathfrak{L}] = E'$ since (α) is a principal ideal. Express α in the following way:

$$\alpha = \frac{u + v\pi_q}{mz},$$

where z is an integer such that $z \nmid [\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$.

In order to find E', we first need to obtain the invariant differential $\omega_{E'}$ of E'. We use the relationship

$$\omega_{E'} = (u/(mz))\omega_{E_c}.$$

Using this equation, we can explicitly find the isomorphism

$$\eta \colon E_c \to E' \text{ with } \eta^*(\omega_{E'}) = (u/(mz))\omega_{E_c}.$$

We can then find E' as follows:

- Assume that E_c is given by $y^2 = x^3 + a'x + b'$.
- Use the fact that for $\lambda \in \mathbb{F}_q^*$, the isomorphism given by $(x, y) \to (\lambda^2 x, \lambda^3 y)$, multiplies ω_{E_c} by $1/\lambda$.
- Hence E' is given by $y^2 = x^3 + (u/(mz))^4 a'x + (u/(mz))^6 b'$.

Now we need to find the image $\phi_\ell(P) \in E'$ of the given point $P \in E$. We again recursively obtain $\phi_c(P) \in E_c$. Then we apply η to obtain $\eta(\phi_c(P)) \in E'$. Finally we must compute the action of (α) on $\eta(\phi_c(P)) \in E'$:

$$R = \alpha \cdot \eta(\phi_c(P)) = ((zm)^{-1}(u + v\pi_q))(\eta(\phi_c(P))) \in E'(\mathbb{F}_q).$$

This procedure only determines $\phi_\ell(P)$ up to automorphisms of E'. Hence we only obtain the x-coordinate of R (or the square or cube of the x-coordinate, respectively, if $\Delta = -4$ or -3) as output. The steps of the algorithm are summarized in Algorithm 2.

4.4 The Bröker-Charles-Lauter Algorithm and Main Theorem

The proof of correctness and running time analysis for the Bröker-Charles-Lauter algorithm are given in the following theorem.

Algorithm 2 The Bröker-Charles-Lauter algorithm

```
Input: A discriminant $\Delta$, an elliptic curve $E/\mathbb{F}_q$ with $\mathrm{End}(E) = \mathcal{O}_\Delta$ and a point $P \in E(\mathbb{F}_{q^n})$ such that $[\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$ and $\#E(\mathbb{F}_{q^n})$ are coprime, and an $\mathrm{End}(E)$-ideal $\mathfrak{L} = (\ell, c + d\pi_q)$ of prime norm $\ell \neq \mathrm{char}(\mathbb{F}_q)$ not dividing the index $[\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$.
Output: The unique elliptic curve $E'$ admitting a normalized isogeny $\phi \colon E \to E'$ with kernel $E[\mathfrak{L}]$, and the $x$-coordinate of $\phi(P)$ for $\Delta \neq -3, -4$ and the square (resp. cube) of the $x$-coordinate otherwise.
  Compute the direct sum decomposition $\mathrm{Cl}(\mathcal{O}_\Delta) = \bigotimes \langle [I_i] \rangle$ of $\mathrm{Cl}(\mathcal{O}_\Delta)$ into cyclic groups generated by the degree 1 prime ideals $I_i$ of smallest norm that are coprime to the product $p \cdot \#E(\mathbb{F}_{q^n}) \cdot [\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$.
  Using brute force[1], find $e_1, e_2, \ldots, e_k$ such that $[\mathfrak{L}] = [I_1^{e_1}] \cdot [I_2^{e_2}] \cdots [I_k^{e_k}]$.
  Find $\alpha$ (using Cornacchia's algorithm) and express $\mathfrak{L} = I_1^{e_1} \cdot I_2^{e_2} \cdots I_k^{e_k} \cdot (\alpha)$.
  Compute a sequence of isogenies $(\phi_1, \ldots, \phi_s)$ such that the composition $\phi_c \colon E \to E_c$ has kernel $E[I_1^{e_1} \cdot I_2^{e_2} \cdots I_k^{e_k}]$ using the method of [BCL08, § 3] or equivalently methods described in Chapter 3.
  Evaluate $\phi_c(P) \in E_c(\mathbb{F}_{q^n})$.
  Write $\alpha = (u + v\pi_q)/(zm)$. Compute the isomorphism $\eta \colon E_c \xrightarrow{\sim} E'$ with $\eta^*(\omega_{E'}) = (u/zm)\omega_{E_c}$. Compute $Q = \eta(\phi_c(P))$.
  Compute $(zm)^{-1} \bmod \#E(\mathbb{F}_{q^n})$, and compute $R = ((zm)^{-1}(u + v\pi_q))(Q)$.
  Put $r = x(R)^{|\mathcal{O}_\Delta^*|/2}$ and return $(E', r)$.
```

Theorem 4.4.1. *Let $E/\mathbb{F}_q$ be an ordinary elliptic curve with Frobenius π_q, given by a Weierstrass equation, and let $P \in E(\mathbb{F}_{q^n})$ be a point on E. Let $\Delta = \mathrm{disc}(\mathrm{End}(E))$ be given. Assume that $[\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$ and $\#E(\mathbb{F}_{q^n})$ are coprime, and let $\mathfrak{L} = (\ell, c + d\pi_q)$ be an $\mathrm{End}(E)$-ideal of prime norm $\ell \neq \mathrm{char}(\mathbb{F}_q)$ not dividing the index $[\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$. Algorithm 2 computes the unique elliptic curve E' such that there exists a normalized isogeny $\phi \colon E \to E'$ with kernel $E[\mathfrak{L}]$. Furthermore, it computes the x-coordinate of $\phi(P)$ if $\mathrm{End}(E)$ does not equal $\mathbb{Z}[i]$ or $\mathbb{Z}[\zeta_3]$ and the square, respectively cube, of the x-coordinate of $\phi(P)$ otherwise. The running time of the algorithm is polynomial in $\log(\ell)$, $\log(q)$, n and $|\Delta|$.*

Proof. [BCL08, Section 4] □

[1] Bröker, Charles, and Lauter mention that this computation can be done in "various ways" [BCL08, p. 107], but the only explicit method given in [BCL08] is brute force. The use of brute force limits the algorithm to elliptic curves for which $|\Delta|$ is small, such as pairing-friendly curves.

4.5 Remarks on the Bröker-Charles-Lauter Algorithm

The Bröker-Charles-Lauter algorithm does in fact greatly improve the running time for evaluating isogenies when ℓ is large, since its running time is polynomial in $\log \ell$. However, Bröker, Charles, and Lauter do not provide any cost analysis for the ideal factorization step of their algorithm (step 2 of Algorithm 2), other than to state that it is polynomial in $|\Delta|$. In the examples that they provide, the factorization is obtained by brute force, which does in fact take exponential running time in $\log |\Delta|$. This limits the practicality of their algorithm, since for elliptic curves with endomorphism rings with large $|\Delta|$, the algorithm is impractical to run. Some elliptic curves, such as pairing-friendly curves, have endomorphism rings of small discriminants (in absolute value). However, for a randomly chosen curve, we expect $|\Delta| = 4q - t^2 \approx q$. For cryptographic purposes, q is usually very large, and thus the Bröker-Charles-Lauter algorithm is of limited utility in such cases.

Chapter 5

Our Subexponential Algorithm

5.1 Introduction

In this chapter we present and describe in detail our algorithm for evaluating large prime degree isogenies which has subexponential running time in the size of the magnitude of the discriminant of the endomorphism ring of the elliptic curve and polynomial in the degree of the isogeny. The algorithm given in this chapter was published in ANTS-IX [JS10].

Our objective is to evaluate the unique horizontal normalized isogeny on a given elliptic curve $E/\mathbb{F}_q$ whose kernel ideal is given as $\mathfrak{L} = (\ell, c + d\pi_q)$, at a given point $P \in E(\mathbb{F}_{q^n})$, where ℓ is an Elkies prime. As in [BCL08], we must also impose the additional restriction that $\ell \nmid [\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$; for Elkies primes, an equivalent restriction is that $\ell \nmid [\mathcal{O}_K : \mathbb{Z}[\pi_q]]$, but we retain the original formulation for consistency with [BCL08].

In practice, one is typically given ℓ instead of $\mathfrak{L}$, but since it is easy to calculate the list of (at most two) possible primes $\mathfrak{L}$ lying over ℓ (cf. [BV07]), these two interpretations are for all practical purposes equivalent, and we switch freely between them when convenient. When ℓ is small, one can use modular polynomial based techniques [BCL08, §3.1] (as described in Chapter 3), which have running time $O(\ell^3 \log(\ell)^{4+\varepsilon})$ [Eng09]. However, for isogeny degrees of cryptographic size (e.g. 2^{160}), this approach is impractical. The Bröker-Charles-Lauter algorithm sidesteps this problem, by using an alternative factorization of $\mathfrak{L}$. However, the running time of Bröker-Charles-Lauter is polynomial in $|\Delta|$, and therefore only works for small values of $|\Delta|$. In this chapter we present a modified version of the Bröker-Charles-Lauter algorithm which is suitable for large values of $|\Delta|$.

We begin by giving an overview of our approach. In order to handle large values of $|\Delta|$, there are two main problems to overcome. One problem is that we need a fast way to produce a factorization

$$\mathfrak{L} = I_1^{e_1} I_2^{e_2} \cdots I_k^{e_k} \cdot (\alpha) \tag{5.1}$$

Algorithm 3 Computing a factor base

```
Input: A discriminant $\Delta$, a bound $N$.
Output: The set $\mathcal{I}$ consisting of split prime ideals of norm less than $N$, together with the corresponding set $\mathcal{F}$ of quadratic forms.
Set $\mathcal{F} \leftarrow \emptyset$.
Set $\mathcal{I} \leftarrow \emptyset$.
Find all primes $p < N$ such that $(\frac{\Delta}{p}) = 1$. Call this set $P$. Let $k = |P|$.
For each prime $p_i \in P$, find an ideal $\mathfrak{p}_i$ of norm $p_i$ (using Cornacchia's algorithm).
For each $i$, find a quadratic form $f_i = [(p_i, b_i, c_i)]$ corresponding to $\mathfrak{p}_i$ in $\mathrm{Cl}(\mathcal{O}_\Delta)$, using the technique of [Sey87, §3].
Output $\mathcal{I} = \{\mathfrak{p}_1, \mathfrak{p}_2, \ldots, \mathfrak{p}_k\}$ and $\mathcal{F} = \{f_1, f_2, \ldots, f_k\}$.
```

as in lines 2 and 3 of the BCL algorithm (Algorithm 2). The other problem is that the exponents e_i in Equation (5.1) need to be kept small, since the running times of lines 3 and 4 of Algorithm 2 are proportional to $\sum_i |e_i| \operatorname{Norm}(I_i)^2$. The first problem, that of finding a factorization of $\mathfrak{L}$, can be solved in subexponential time using the index calculus algorithm of Hafner and McCurley [HM89] (see also [BV07, Chapter 11]). To resolve the second problem, we turn to the following idea of Galbraith, Hess, and Smart [GHS02], and recently further refined by Bisson and Sutherland [BS09]. The idea is that, in the process of sieving for smooth norms, one can arbitrarily restrict the input exponent vectors to sparse vectors $(e_1, e_2, ..., e_k)$ such that $\sum_i |e_i| N(I_i)^2$ is kept small.

5.2 Finding the Factor Base

Let $\mathrm{Cl}(\mathcal{O}_\Delta)$ denote the ideal class group of $\mathcal{O}_\Delta$. Algorithm 3 produces a factor base consisting of split primes in $\mathcal{O}_\Delta$ of norm less than some bound N. The optimal value of N will be determined in Section 5.6. The elements of our factor base are represented as reduced quadratic forms. To find the forms we use the technique of [Sey87, §3]: given Δ and a prime p such that $(\frac{\Delta}{p}) = 1$, define $b = \min\{d \in \mathbb{N} : d^2 = \Delta \mod 4p\}$, and $c = (b^2 - \Delta)/(4a)$. Then the quadratic form corresponding to an ideal of norm p is $[(p, b, c)]$.

5.3 "Factoring" Large Prime Degree Ideals

Algorithm 4, based on the algorithm of Hafner and McCurley, takes as input a discriminant Δ, a curve E, a prime ideal $\mathfrak{L}$ of prime norm ℓ in $\mathcal{O}_\Delta$, a smoothness bound N, and an extension degree n. It outputs a factorization

$$\mathfrak{L} = I_1^{e_1} I_2^{e_2} \cdots I_k^{e_k} \cdot (\alpha)$$

as in Equation 5.1, where the I_i's are as in Algorithm 2, the exponents e_i are positive, sparse, and small (i.e., polynomial in N), and the ideal (α) is a principal fractional ideal generated by α.

Algorithm 4 uses Theorem 3.1 of [Sey87]. For convenience we state the full result here.

Theorem 5.3.1. *Let $\mathcal{F}$ be the factor base computed in Algorithm 3. Let $f_i and b_i$ be as in Algorithm 3. Let $[(a, b, c)]$ be a quadratic form in $\mathrm{Cl}(\mathcal{O}_\Delta)$. Let $a = \prod_{i=1}^{k} p_i^{e_i}$, $e_i \in \mathbb{Z}$, p_i prime, be the prime factorization of a. Then:*

- $(\frac{\Delta}{p_i}) = 1$, *$b \equiv \pm b_i \mod 2p_i$ for all p_i, $i = 1, \ldots, k$.*
- $[(a, b, c)] = \prod_{i=1}^{k} f_i^{\pm e_i}$*, where the plus sign in the exponent e_i holds if and only if $b \equiv b_i \mod 2p_i$.*

Proof. [Sey87] □

(Note that the relationship between the ideal classes $[I]$ and $[\bar{I}]$ is $[\bar{I}] = [I]^{-1}$.)

5.4 Algorithm for Evaluating Prime Degree Isogenies

The overall algorithm for evaluating prime degree isogenies is given in Algorithm 5. This algorithm is identical to Algorithm 2, except that the factorization of $\mathfrak{L}$ is performed using Algorithm 4. To maintain consistency with [BCL08], we have included the quantities Δ and $\mathrm{End}(E)$ as part of the input to the algorithm. However, we remark that these quantities can be computed from $E/\mathbb{F}_q$ in $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2})$ operations using the algorithm of Bisson and Sutherland [BS09], even if they are not provided as input, although additional heuristics are required for this running time bound to hold.

5.5 Heuristic Assumptions

Our algorithm shares many elements in common with Bisson and Sutherland's algorithm to compute the endomorphism ring of an elliptic curve [BS09]. Although the initial version of our main result was obtained independently, we subsequently incorporated their ideas into our algorithm in several places, resulting in a simpler presentation as well as a large

Algorithm 4 "Factoring" a prime ideal

```
Input: A discriminant $\Delta$, an elliptic curve $E/\mathbb{F}_q$ with $\mathrm{End}(E) = \mathcal{O}_\Delta$, a smoothness bound
    $N$, a prime ideal $\mathfrak{L}$ of norm $\ell$ in $\mathcal{O}_\Delta$, an extension degree $n$.
Output: Relation of the form $\mathfrak{L} = (\alpha) \cdot \prod_{i=1}^{k} I_i^{e_i}$, where $(\alpha)$ is a fractional ideal, $I_i$ are as
    in Algorithm 2, and $e_i > 0$ are small and sparse.
Run Algorithm 3 on input $\Delta$ and $N$ to obtain $\mathcal{I} = \{\mathfrak{p}_1, \mathfrak{p}_2, \ldots, \mathfrak{p}_k\}$ and $\mathcal{F} =
    \{f_1, f_2, \ldots, f_k\}$. Discard any primes dividing $p \cdot \#E(\mathbb{F}_{q^n}) \cdot [\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$.
Set $p_i \leftarrow \mathrm{Norm}(\mathfrak{p}_i)$. (These values are also calculated in Algorithm 3.)
Obtain the reduced quadratic form $[\mathfrak{L}]$ corresponding to the ideal class of $\mathfrak{L}$.
repeat
    for $i = 1, \ldots, k$ do
        Pick exponents $x_i$ in the range $[0, (N/p_i)^2]$ such that at most $k_0$ are nonzero, where
            $k_0$ is a global absolute constant (in practice, $k_0 = 3$ suffices).
    end for
    Compute the reduced quadratic form $\mathfrak{a} = (a, b, c)$ for which the ideal class $[\mathfrak{a}]$ is
        equivalent to $[\mathfrak{L}] \cdot \prod_{i=1}^{k} f_i^{x_i}$.
until The integer $a$ factors completely into the primes $p_i$, and the relation derived from
    $[\mathfrak{a}] = [\mathfrak{L}] \cdot \prod_{i=1}^{k} f_i^{x_i}$ contains fewer than $\sqrt{\log(|\Delta|/3)/z}$ nonzero exponents.
Write $a = \prod_{i=1}^{k} p_i^{u_i}$.
for i=1, $\ldots, k$ do
    Using the technique of Seysen (Theorem 5.3.1), determine the signs of the exponents
        $y_i = \pm u_i$ for which $\mathfrak{a} = \prod_{i=1}^{k} f_i^{y_i}$.
    Let $e_i = y_i - x_i$. (These exponents satisfy $[\mathfrak{L}] = \prod_{i=1}^{k} f_i^{e_i}$.)
    if $e_i \geq 0$ then
        Set $I_i \leftarrow \bar{\mathfrak{p}}_i$
    else
        Set $I_i \leftarrow \mathfrak{p}_i$
    end if
end for
Compute the principal ideal $I = \mathfrak{L} \cdot \prod_{i=1}^{k} I_i^{|e_i|}$.
Using Cornacchia's algorithm, find a generator $\beta \in \mathcal{O}_\Delta$ of $I$.
Set $m \leftarrow \prod_{i=1}^{k} p_i^{|e_i|}$ and $\alpha \leftarrow \frac{\beta}{m}$.
Output $\mathfrak{L} = (\alpha) \cdot \bar{I}_1^{|e_1|} \cdot \bar{I}_2^{|e_2|} \cdots \bar{I}_k^{|e_k|}$.
```

speedup compared to the original version of our work. Consequently, our algorithm relies on a number of heuristic assumptions, all inherited from [BS09]. Namely we assume the following three heuristics to be true (which are a subset of heuristics Bisson and Sutherland in [BS09] assume to be true):

Algorithm 5 Evaluating prime degree isogenies

```
Input: A discriminant $\Delta$, an elliptic curve $E/\mathbb{F}_q$ with $\mathrm{End}(E) = \mathcal{O}_\Delta$ and a point $P \in E(\mathbb{F}_{q^n})$ such that $[\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$ and $\#E(\mathbb{F}_{q^n})$ are coprime, and an $\mathrm{End}(E)$-ideal $\mathfrak{L} = (\ell, c + d\pi_q)$ of prime norm $\ell \neq \mathrm{char}(\mathbb{F}_q)$ not dividing the index $[\mathrm{End}(E) : \mathbb{Z}[\pi_q]]$.
Output: The unique elliptic curve $E'$ admitting a normalized isogeny $\phi\colon E \to E'$ with kernel $E[\mathfrak{L}]$, and the $x$-coordinate of $\phi(P)$ for $\Delta \neq -3, -4$ and the square (resp. cube) of the $x$-coordinate otherwise.
1: Choose a smoothness bound $N$ (see Section 5.6).
2: Using Algorithm 4 on input $(\Delta, E, N, \mathfrak{L}, n)$, obtain a factorization of the form $\mathfrak{L} = I_1^{e_1} \cdot I_2^{e_2} \cdots I_k^{e_k} \cdot (\alpha)$.
3: Compute a sequence of isogenies $(\phi_1, \ldots, \phi_s)$ such that the composition $\phi_c\colon E \to E_c$ has kernel $E[I_1^{e_1} \cdot I_2^{e_2} \cdots I_k^{e_k}]$ using the method of [BCL08, § 3].
4: Evaluate $\phi_c(P) \in E_c(\mathbb{F}_{q^n})$.
5: Write $\alpha = (u + v\pi_q)/(zm)$. Compute the isomorphism $\eta\colon E_c \overset{\sim}{\to} E'$ with $\eta^*(\omega_{E'}) = (u/zm)\omega_{E_c}$. Compute $Q = \eta(\phi_c(P))$.
6: Compute $(zm)^{-1} \bmod \#E(\mathbb{F}_{q^n})$, and compute $R = ((zm)^{-1}(u + v\pi_q))(Q)$.
7: Put $r = x(R)^{|\mathcal{O}_\Delta^*|/2}$ and return $(E', r)$.
```

1. We assume the Generalized Riemann Hypothesis (GRH).

2. The limitations imposed in line 6 of Algorithm 4 do not affect the probability distribution of the output ideal class $[\mathfrak{a}]$.

3. We assume that the ECM [Len87] technique for factoring integers using elliptic curves finds the prime factor p of an integer n in $L_p(\frac{1}{2}, 2) \cdot (\log n)^2$ time.

5.6 Running Time Analysis

In this section, we determine the theoretical running time of Algorithm 5, as well as the optimal value of the smoothness bound N to use in line 1 of the algorithm. As is typical for subexponential time factorization algorithms involving a factor base, these two quantities depend on each other, and hence both are calculated simultaneously.

As in [CFA$^+$06], we define[1] $L_n(\alpha, c)$ by

$$L_n(\alpha, c) = O(\exp((c + o(1))(\log(n))^\alpha(\log(\log(n)))^{1-\alpha})).$$

[1] The definition of $L_n(\alpha, c)$ in [BV07] differs from that of [CFA$^+$06] in the $o(1)$ term. In places where we cite [BV07], we account for this discrepancy in our text.

The quantity $L_n(\alpha, c)$ interpolates between polynomial and exponential size in $\log n$ as α ranges from 0 to 1. We set $N = L_{|\Delta|}(\frac{1}{2}, z)$ for an unspecified value of z, and in the following paragraphs we determine the optimal value of z which minimizes the running time of Algorithm 5. (The fact that $\alpha = \frac{1}{2}$ is optimal is clear from the analysis below, as well as from prior experience with integer factorization algorithms.) For convenience, we will abbreviate $L_{|\Delta|}(\alpha, c)$ to $L(\alpha, c)$ throughout.

Line 2 of Algorithm 5 involves running Algorithm 4, which in turn calls Algorithm 3. Thus we analyze the running times of these algorithms.

Theorem 5.6.1. *Algorithm 3 has a running time bounded above by*

$$L(\tfrac{1}{2}, z).$$

Proof. Algorithm 3 is almost the same as Algorithm 11.1 from [BV07], which requires $L(\frac{1}{2}, z)$ time, as shown in [BV07]. The only difference is that we add an additional step where we obtain the quadratic form corresponding to each prime ideal in the factor base. This extra step requires $O(\log(\mathrm{Norm}(I))^{1+\varepsilon})$ time for a prime ideal I, using Cornacchia's Algorithm [HMW90]. Thus, the overall running time for Algorithm 3 is bounded above by

$$L(\tfrac{1}{2}, z) \cdot \log(L(\tfrac{1}{2}, z))^{1+\varepsilon} = L(\tfrac{1}{2}, z).$$

□

Theorem 5.6.2. *Algorithm 4 has a (heuristic) running time bounded above by*

$$L(\tfrac{1}{2}, \tfrac{1}{4z}) + (\log \ell + L(\tfrac{1}{2}, 2z))^{1+\varepsilon}.$$

Proof. Line 1 of Algorithm 4 runs Algorithm 3, which has a running time $L(\frac{1}{2}, z)$. Line 2 of Algorithm 4 takes $\log(\ell)$ time using standard algorithms [Cox89]. The loop in lines 4–9 of Algorithm 4 is very similar to the FindRelation algorithm in [BS09], except that we only use one discriminant, and we omit the requirement that $\#R/D_1 > \#R/D_2$ (which in any case is meaningless when there is only one discriminant). Needless to say, this change can only speed up the algorithm. Taking $\mu = \sqrt{2}z$ in [BS09, Prop. 6], we find that the (heuristic) expected running time of the loop in lines 4–9 of Algorithm 4 is $L(\frac{1}{2}, \frac{1}{4z})$.

The next step in Algorithm 4 having nontrivial running time is the computation of the ideal product in line 20. To exponentiate an element of an arbitrary semigroup to a power e requires $O(\log e)$ semigroup multiplication operations [Coh93, §1.2] with square-and-multiply. To multiply two ideals I and J in an imaginary quadratic order (via composition of quadratic forms) requires $O(\max(\log(\mathrm{Norm}(I)), \log(\mathrm{Norm}(J)))^{1+\varepsilon})$ bit operations using fast multiplication [Sch91, §6]. Each of the expressions $|I_i|^{|e_i|}$ therefore requires $O(\log |e_i|)$ ideal multiplication operations to compute, with each individual multiplication requiring

$$O((|e_i| \log(\mathrm{Norm}(I_i)))^{1+\varepsilon}) = O\left(\left(\left(\frac{N}{p_i}\right)^2 \log(p_i)\right)^{1+\varepsilon}\right) = O(N^{2+\varepsilon})$$

bit operations, for a total running time of $(\log e_i)O(N^{2+\varepsilon}) = L(\frac{1}{2}, 2z)$ for each i. This calculation must be performed once for each nonzero exponent e_i. By line 9, the number of nonzero exponents appearing in the relation is at most $\sqrt{\log(|\Delta|/3)}/z$, so the amount of time required to compute all of the $|I_i|^{|e_i|}$ for all i is $(\sqrt{\log(|\Delta|/3)}/z)L(\frac{1}{2}, 2z) = L(\frac{1}{2}, 2z)$. Afterward, the values $|I_i|^{|e_i|}$ must all be multiplied together, a calculation which entails at most $\sqrt{\log(|\Delta|/3)}/z$ ideal multiplications where the log-norms of the input multiplicands are bounded above by

$$\log \mathrm{Norm}(I_i^{|e_i|}) = |e_i| \log \mathrm{Norm}(I_i) \leq \left(\frac{N}{p_i}\right)^2 \log p_i \leq N^2 = L(\tfrac{1}{2}, 2z),$$

and thus each of the (at most) $\sqrt{\log(|\Delta|/3)}/z$ multiplications in the ensuing product can be completed in time at most $(\sqrt{\log(|\Delta|/3)}/z)L(\frac{1}{2}, 2z) = L(\frac{1}{2}, 2z)$. Finally, we must multiply this end result by $\mathfrak{L}$, an operation which requires $O(\max(\log \ell, L(\frac{1}{2}, 2z))^{1+\varepsilon})$ time. All together, the running time of step 20 is $L(\frac{1}{2}, 2z) + O(\max(\log \ell, L(\frac{1}{2}, 2z))^{1+\varepsilon}) \leq \max((\log \ell), L(\frac{1}{2}, 2z))^{1+\varepsilon}$, and the norm of the resulting ideal I is bounded above by $\ell \cdot \exp(L(\frac{1}{2}, 2z))$.

Obtaining the generator β of I in line 21 of Algorithm 4 using Cornacchia's algorithm requires

$$O(\log(\mathrm{Norm}(I))^{1+\varepsilon}) = (\log \ell + L(\tfrac{1}{2}, 2z))^{1+\varepsilon}$$

time. We remark that finding β given I is substantially easier than the usual Cornacchia's algorithm, which entails finding β given only $\mathrm{Norm}(I)$. The usual algorithm requires finding *all* the square roots of Δ modulo $\mathrm{Norm}(I)$, which is very slow when $\mathrm{Norm}(I)$ has a large number of prime divisors. This time-consuming step is unnecessary when the ideal I itself is given, since the embedding of the ideal I in $\mathrm{End}(E)$ already provides (up to sign) the correct square root of Δ mod I. A detailed description of this portion of Cornacchia's algorithm in the context of the full algorithm, together with running time figures specific to each sub-step, is given by Hardy et al. [HMW90]; for our purposes, the running time of a single iteration of Step 6 in [HMW90, §4] is the relevant figure.

The total running time of Algorithm 4 is:

$$\begin{aligned}
&L(\tfrac{1}{2}, z) && \text{(algorithm 3 or line 1 of algorithm 4)}\\
&\quad + L(\tfrac{1}{2}, \tfrac{1}{4z}) && \text{(lines 4–9, algorithm 4)}\\
&\quad + \max((\log \ell), L(\tfrac{1}{2}, 2z))^{1+\varepsilon} && \text{(line 20, algorithm 4)}\\
&\quad + (\log \ell + L(\tfrac{1}{2}, 2z))^{1+\varepsilon} && \text{(line 21, algorithm 4)}
\end{aligned}$$

$$= L(\tfrac{1}{2}, \tfrac{1}{4z}) + (\log \ell + L(\tfrac{1}{2}, 2z))^{1+\varepsilon}.$$

This concludes our analysis of Algorithm 4.

□

Theorem 5.6.3. *The optimal value for z in Algorithm 5 is*

$$z = \frac{1}{2\sqrt{3}}.$$

With this value, Algorithm 5 has a running time bounded above by

$$(\log(\ell) + L(\tfrac{1}{2}, \tfrac{1}{\sqrt{3}}))^{1+\varepsilon} + L(\tfrac{1}{2}, \tfrac{\sqrt{3}}{2}) + L(\tfrac{1}{2}, \tfrac{1}{\sqrt{3}})(\log q^n)^{1+\varepsilon}.$$

Proof. We find that (as in [BCL08]) the computation of the individual isogenies ϕ_i in line 3 of Algorithm 5 is limited by the time required to compute the modular polynomials $\Phi_n(x, y)$. Using the Chinese remainder theorem-based method of Bröker et al. [BLS10], these polynomials can be computed mod q in time $O(n^3 \log^{3+\varepsilon}(n))$, and the resulting polynomials require $O(n^2(\log^2 n + \log q))$ space. For each ideal I_i, the corresponding modular polynomial of level p_i only needs to be computed once, but the polynomial once computed must be evaluated, differentiated, and otherwise manipulated e_i times, at a cost of $O(p_i^{2+\varepsilon})$ field operations in $\mathbb{F}_q$ per manipulation, or $O(p_i^{2+\varepsilon})(\log q)^{1+\varepsilon}$ bit operations using fast multiplication. The total running time of line 3 is therefore

$$\begin{aligned}
O(p_i^{3+\varepsilon}) + \sum_i |e_i| p_i^{2+\varepsilon} (\log q)^{1+\varepsilon} &\leq O(N^{3+\varepsilon}) + \sum_i \left(\left(\frac{N}{p_i} \right)^2 \right) p_i^{2+\varepsilon} (\log q)^{1+\varepsilon}\\
\leq O(N^{3+\varepsilon}) + \frac{\sqrt{\log(|\Delta|/3)}}{z} N^{2+\varepsilon} (\log q)^{1+\varepsilon} &= L(\tfrac{1}{2}, 3z) + L(\tfrac{1}{2}, 2z)(\log q)^{1+\varepsilon}.
\end{aligned}$$

Similarly, the evaluation of ϕ_c in line 4 requires

$$\sum_i |e_i| p_i^{2+\varepsilon} = L(\tfrac{1}{2}, 2z)$$

field operations in $\mathbb{F}_{q^n}$, which corresponds to $L(\frac{1}{2}, 2z)(\log q^n)^{1+\varepsilon}$ bit operations using fast multiplication.

Combining all the above quantities, we obtain a total running time of

$$\begin{aligned}
&L(\tfrac{1}{2}, \tfrac{1}{4z}) + (\log \ell + L(\tfrac{1}{2}, 2z))^{1+\varepsilon} && \text{(line 2, from Theorem 5.6.2)}\\
&\quad + L(\tfrac{1}{2}, 3z) + L(\tfrac{1}{2}, 2z)(\log q)^{1+\varepsilon} && \text{(line 3, algorithm 5)}\\
&\quad + L(\tfrac{1}{2}, 2z)(\log q^n)^{1+\varepsilon} && \text{(line 4, algorithm 5)}
\end{aligned}$$

$$= L(\tfrac{1}{2}, \tfrac{1}{4z}) + (\log \ell + L(\tfrac{1}{2}, 2z))^{1+\varepsilon} + L(\tfrac{1}{2}, 3z) + L(\tfrac{1}{2}, 2z)(\log q^n)^{1+\varepsilon}.$$

When $|\Delta|$ is large, we may impose the reasonable assumption that $\log(\ell) \ll L(\frac{1}{2}, z)$ and $\log(q^n) \ll L(\frac{1}{2}, z)$. In this case, the running time of Algorithm 5 is dominated by the expression $L(\frac{1}{2}, \frac{1}{4z}) + L(\frac{1}{2}, 3z) = L(\frac{1}{2}, \max(\frac{1}{4z}, 3z))$, which attains a minimum at $z = \frac{1}{2\sqrt{3}}$. Taking this value of z, we find that the running time of Algorithm 5 is equal to $L_{|\Delta|}(\frac{1}{2}, \frac{\sqrt{3}}{2})$. Since the maximum value of $|\Delta| \leq |\Delta_\pi| = 4q - t^2$ is $4q$, we can alternatively express this running time as simply $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2})$.

In the general case, $\log(\ell)$ and $\log(q^n)$ might be non-negligible compared to $L(\frac{1}{2}, z)$. This can happen in one of two ways: either $|\Delta|$ is small, or (less likely) ℓ is very large and/or n is large. When this happens, we can still bound the running time of Algorithm 5 by taking $z = \frac{1}{2\sqrt{3}}$ in the foregoing calculation, although such a choice may fail to be optimal. We then find that the running time of Algorithm 5 is bounded above by

$$(\log(\ell) + L(\tfrac{1}{2}, \tfrac{1}{\sqrt{3}}))^{1+\varepsilon} + L(\tfrac{1}{2}, \tfrac{\sqrt{3}}{2}) + L(\tfrac{1}{2}, \tfrac{1}{\sqrt{3}})(\log q^n)^{1+\varepsilon}.$$

□

5.7 The Main Theorem

We summarize our results in the following theorem:

Theorem 5.7.1. *Let $E/\mathbb{F}_q$ be an ordinary elliptic curve with Frobenius π_q, given by a Weierstrass equation, and let $P \in E(\mathbb{F}_{q^n})$ be a point on E. Let $\Delta = \operatorname{disc}(\operatorname{End}(E))$ be given. Assume that $[\operatorname{End}(E) : \mathbb{Z}[\pi_q]]$ and $\#E(\mathbb{F}_{q^n})$ are coprime, and let $\mathfrak{L} = (\ell, c + d\pi_q)$ be an $\operatorname{End}(E)$-ideal of prime norm $\ell \neq \operatorname{char}(\mathbb{F}_q)$ not dividing the index $[\operatorname{End}(E) : \mathbb{Z}[\pi_q]]$. Under the heuristics stated in Section 5.5, Algorithm 5 computes the unique elliptic curve E' such that there exists a normalized isogeny $\phi \colon E \to E'$ with kernel $E[\mathfrak{L}]$. Furthermore, it computes the x-coordinate of $\phi(P)$ if $\operatorname{End}(E)$ does not equal $\mathbb{Z}[i]$ or $\mathbb{Z}[\zeta_3]$ and the square,* *respectively cube, of the x-coordinate of $\phi(P)$ otherwise. The running time of the algorithm is bounded above by*

$$(\log(\ell) + L(\tfrac{1}{2}, \tfrac{1}{\sqrt{3}}))^{1+\varepsilon} + L(\tfrac{1}{2}, \tfrac{\sqrt{3}}{2}) + L(\tfrac{1}{2}, \tfrac{1}{\sqrt{3}})(\log q^n)^{1+\varepsilon}.$$

The running time of the algorithm is subexponential in $\log|\Delta|$*, and polynomial in* $\log(\ell)$, $\log(q)$*, and* n*.*

5.8 Examples

In this section we provide a few examples of how our algorithm works.

5.8.1 Small example

Let $p = 10^{10} + 19$ and let $E/\mathbb{F}_p$ be the curve $y^2 = x^3 + 15x + 129$. Then $E(\mathbb{F}_p)$ has cardinality $10000036491 = 3 \cdot 3333345497$ and trace $t = -36471$. To avoid any bias in the selection of the prime ℓ, we set ℓ to be the smallest Elkies prime of E larger than $p/2$, namely $\ell = 5000000029$. We will evaluate the x-coordinate of $\phi(P)$, where ϕ is an isogeny of degree ℓ, and P is chosen arbitrarily to be the point $(5940782169, 2162385016) \in E(\mathbb{F}_p)$. We remark that, although this example is designed to be artificially small for illustration purposes, the evaluation of this isogeny would already be infeasible if we were using prior techniques based on modular functions of level ℓ.

The discriminant Δ of E is $\Delta = t^2 - 4p = -38669866235$. Set $w = \frac{1+\sqrt{\Delta}}{2}$ and $\mathcal{O} = \mathcal{O}_\Delta$. The quadratic form $(5000000029, -2326859861, 270713841)$ represents a prime ideal $\mathfrak{L}$ of norm ℓ, and we show how to calculate the isogeny ϕ having kernel corresponding to $E[\mathfrak{L}]$. Using an implementation of Algorithm 4 in MAGMA [mag], we find in under one second the relation $\mathfrak{L} = (\frac{\beta}{m}) \cdot \mathfrak{p}_{19} \cdot \mathfrak{p}_{31}^{24}$ where $\beta = 5880483076032100005w - 235788727470005542279904$, $m = 19 \cdot 31^{24}$, $\mathfrak{p}_{19} = (19, 2w + 7)$, and $\mathfrak{p}_{31} = (31, 2w + 5)$. Using this factorization, we can then evaluate $\phi \colon E \to E'$ using the latter portion of Algorithm 5. We find that E' is the curve with Weierstrass equation $y^2 = x^3 + 3565469415x + 7170659769$, and $\phi(P) = (7889337683, \pm 3662693258)$. We omit the details of these steps, since this portion of the algorithm is identical to the algorithm of Bröker, Charles and Lauter, and the necessary steps are already extensively detailed in their article [BCL08].

We can check our computations for consistency by performing a second computation, starting from the curve $E' : y^2 = x^3 + 3565469415x + 7170659769$, the point $P' = (7889337683, 3662693258) \in E'(\mathbb{F}_p)$, and the conjugate ideal $\bar{\mathfrak{L}}$, which is represented by the quadratic form $(5000000029, 2326859861, 270713841)$. Let $\bar{\phi} \colon E' \to E''$ denote the unique normalized isogeny with kernel $E'[\bar{\mathfrak{L}}]$. Up to a normalization isomorphism $\iota \colon E \to E''$, the isogeny $\bar{\phi}$ should equal the dual isogeny $\hat{\phi}$ of ϕ, and the composition $\bar{\phi}(\phi(P))$ should yield $\iota(\ell P)$. Indeed, upon performing the computation, we find that E'' has equation

$$y^2 = x^3 + (15/\ell^4)x + (129/\ell^6),$$

which is isomorphic to E via the isomorphism $\iota\colon E \to E''$ defined by $\iota(x,y) = (x/\ell^2, y/\ell^3)$, and

$$\bar{\phi}(\phi(P)) = (3163843645, 8210361642) = (5551543736/\ell^2, 6305164567/\ell^3),$$

in agreement with the value of ℓP, which is $(5551543736, 6305164567)$.

5.8.2 Medium example

Let E be the ECCp-109 curve [cerb] from the Certicom ECC Challenge [cera], with equation $y^2 = x^3 + ax + b$ over $\mathbb{F}_p$ where

$$\begin{aligned} p &= 564538252084441556247016902735257 \\ a &= 321094768129147601892514872825668 \\ b &= 430782315140218274262276694323197 \end{aligned}$$

As before, to avoid any bias in the choice of ℓ, we set ℓ to be the least Elkies prime greater than $p/2$, and we define $w = \frac{1+\sqrt{\Delta}}{2}$ where $\Delta = \operatorname{disc}(\operatorname{End}(E))$. Let $\mathfrak{L}$ be the prime ideal of norm ℓ in $\operatorname{End}(E)$ corresponding to the reduced quadratic form (ℓ, b, c) of discriminant Δ, where $b = -105137660734123120905310489472471$. For each Elkies prime p, let $\mathfrak{p}_p$ denote the unique prime ideal corresponding to the reduced quadratic form (p, b, c) where $b \geq 0$. Our smoothness bound in this case is $N = L(\frac{1}{2}, \frac{1}{2\sqrt{3}}) \approx 200$. Using Sutherland's `smoothrelation` package [Sut], which implements the FindRelation algorithm of [BS09], one finds in a few seconds (using an initial seed of 0) the relation $\mathfrak{L} = \left(\frac{\beta}{m}\right)\mathfrak{I}$, where

$$\begin{aligned} \mathfrak{I} &= \bar{\mathfrak{p}}_7^{72}\bar{\mathfrak{p}}_{13}^{100}\bar{\mathfrak{p}}_{23}^{14}\bar{\mathfrak{p}}_{47}^{2}\bar{\mathfrak{p}}_{73}^{2}\bar{\mathfrak{p}}_{103}\mathfrak{p}_{179}\mathfrak{p}_{191} \\ m &= 7^{72}13^{100}23^{14}47^{2}73^{2}103^{1}179^{1}191^{1} \end{aligned}$$

and

$$\begin{aligned} \beta = {} & 3383947601020121267815309931891893555677440374614137047492987151\backslash \\ & 22260417314622648471444260197118494483544222058008884837 \\ & - 1713152334033312180094376774440754045496152167352278262491589014\backslash \\ & 0971672388272394276444760757048909796 85 \cdot w \end{aligned}$$

We find that the codomain E' of the normalized isogeny $\phi\colon E \to E'$ of kernel $E[\mathfrak{L}]$ has equation $y^2 = x^3 + a'x + b'$ where

$$\begin{aligned} a' &= 84081262962164770032033494307976 \\ b' &= 506928585427238387307510041944828 \end{aligned}$$

and that the base point

$$P = (97339010987059066523156133908935, 149670372846169285760682371978898)$$

of E given in the Certicom ECC challenge has image

$$(450689656718652268803536868496211, \pm 345608697871189839292674734567941).$$

under ϕ. As with the first example, we checked the computation for consistency by using the conjugate ideal.

5.8.3 Large example

Let E be the ECCp-239 curve [cerb] from the Certicom ECC Challenge [cera]. Then E has equation $y^2 = x^3 + ax + b$ over $\mathbb{F}_p$ where

$$\begin{aligned} p &= 862591559561497151050143615844796924047865589835498401307522524859467869 \\ a &= 820125117492400602839381236756362453725976037283079104527317913759073622 \\ b &= 545482459632327583111433582031095022426858572446976004219654298705912499 \end{aligned}$$

Let $\mathfrak{L}$ be the prime ideal whose norm is the least Elkies prime greater than $p/2$ and whose ideal class is represented by the quadratic form (ℓ, b, c) with $b \geq 0$. We have $N = L(\frac{1}{2}, \frac{1}{2\sqrt{3}}) \approx 5000$, and one finds in a few hours using `smoothrelation` [Sut] that $\mathfrak{L}$ is equivalent to

$$\mathfrak{I} = \bar{\mathfrak{p}}_7^2 \mathfrak{p}_{11} \mathfrak{p}_{19} \mathfrak{p}_{37}^2 \bar{\mathfrak{p}}_{71}^2 \bar{\mathfrak{p}}_{131} \mathfrak{p}_{211} \bar{\mathfrak{p}}_{389} \bar{\mathfrak{p}}_{433} \bar{\mathfrak{p}}_{467} \bar{\mathfrak{p}}_{859}^{18} \mathfrak{p}_{863} \bar{\mathfrak{p}}_{1019} \bar{\mathfrak{p}}_{1151} \bar{\mathfrak{p}}_{1597} \bar{\mathfrak{p}}_{2143}^6 \bar{\mathfrak{p}}_{2207}^5 \bar{\mathfrak{p}}_{3359}$$

where each ideal $\mathfrak{p}_p$ is represented by the reduced quadratic form (p, b, c) having $b \geq 0$ (this computation can be reconstructed with [Sut] using the seed 7). The quotient $\mathfrak{L}/\mathfrak{I}$ is generated by β/m where $m = \text{Norm}(\mathfrak{I})$ and β is

$$\begin{aligned} &-923525986803059652225406070265439117913488592374741428959120914067053307\backslash \\ &4585317 - 917552768623818156695534742084359293432646189962935478129227909w. \end{aligned}$$

Given this relation, evaluating isogenies of degree ℓ is a tedious but routine computation using Elkies-Atkin techniques, described in Chapter 3. Although we do not complete it here, the computation is well within the reach of present technology; indeed, Bröker et al. [BLS10] have computed classical modular polynomials mod p of level up to 20000, well beyond the largest prime of 3389 appearing in our relation.

5.9 Finding Equations in Quasi-Optimal Time

Given two elliptic curves E and E' over $\mathbb{F}_q$ admitting a normalized isogeny $\phi\colon E \to E'$ of degree ℓ, the equation of ϕ as a rational function contains $O(\ell)$ coefficients. Bostan et al. [BMSS08] have published an algorithm which produces this equation, given E, E', and ℓ. Their algorithm has running time $O(\ell^{1+\varepsilon})$, which is quasi-optimal given the size of the output. Using our algorithm, it is possible to compute E' from E and ℓ in time $\log(\ell)L_{|\Delta|}(\frac{1}{2}, \frac{\sqrt{3}}{2})$ for large ℓ. Hence the combination of the two algorithms can produce the equation of ϕ within a quasi-optimal running time of $O(\ell^{1+\varepsilon})$ for $\ell \gg q$, given only E and ℓ (or E and $\mathfrak{L}$), without the need to provide E' in the input.

Chapter 6

A Subexponential Algorithm Assuming Only GRH

In this chapter we present a variant of the algorithm from the previous chapter. Our variant improves on the original algorithm in the sense that we remove all heuristic assumptions except GRH. In practice the new algorithm is slower, although asymptotically it has the same running time as before, with the same constants in the exponents. The main difference is that the exponent bounds in the new algorithm (Step 4 of Algorithm 7) are optimized for provability rather than performance. Portions of this chapter are included in a preprint [CJS10] which represents joint work with Andrew Childs and David Jao.

6.1 Isogeny Graphs Under GRH

Our runtime analysis in Section 6.2 relies on the following result which states, roughly, that random short products of small primes in $\mathrm{Cl}(\mathcal{O}_\Delta)$ yield nearly uniformly random elements of $\mathrm{Cl}(\mathcal{O}_\Delta)$, under GRH.

Theorem 6.1.1. *Let $\mathcal{O}_\Delta$ be an imaginary quadratic order of discriminant $\Delta < 0$ and conductor c. Set $G = \mathrm{Cl}(\mathcal{O}_\Delta)$. Let B and x be real numbers satisfying $B > 2$ and $x \geq (\ln|\Delta|)^B$. Let S_x be the multiset $A \cup A^{-1}$ where*

$$A = \{[\mathfrak{p}] \in G : \gcd(c, \mathfrak{p}) = 1 \textit{ and } \mathrm{Norm}(\mathfrak{p}) \leq x \textit{ is prime}\}.$$

Then, assuming GRH, there exists a positive absolute constant $C > 1$, depending only on B, such that for all Δ, a random walk of length

$$t \geq C \frac{\ln|G|}{\ln\ln|\Delta|}$$

in the Cayley graph $\mathrm{Cay}(G, S_x)$ *from any starting vertex lands in any fixed subset* $S \subset G$ *with probability at least* $\frac{1}{2}\frac{|S|}{|G|}$.

Proof. Apply Corollary 1.3 of [JMV09] with the parameters

- $K =$ the field of fractions of $\mathcal{O}_\Delta$
- $G = \mathrm{Cl}(\mathcal{O}_\Delta)$
- $q = |\Delta|$.

Observe that by Remark 1.2(a) of [JMV09], Corollary 1.3 of [JMV09] applies to the ring class group $G = \mathrm{Cl}(\mathcal{O}_\Delta)$, since ring class groups are quotients of narrow ray class groups [Cox89, p. 160]. By Corollary 1.3 of [JMV09], Theorem 6.1.1 holds for all sufficiently large values of $|\Delta|$, i.e., for all but finitely many $|\Delta|$. To prove the theorem for all $|\Delta|$, simply take a larger (but still finite) value of C. □

Corollary 6.1.2. *Theorem 6.1.1 holds even if the definition of the set* A *is changed to*

$$A = \{[\mathfrak{p}] \in G : \gcd(m\Delta, \mathfrak{p}) = 1 \textit{ and } \mathrm{Norm}(\mathfrak{p}) \leq x \textit{ is prime}\}$$

where m *is any integer having at most* $O(x^{1/2-\varepsilon} \log |\Delta|)$ *prime divisors.*

Proof. The alternative definition of the set A differs from the original definition by no more than $O(x^{1/2-\varepsilon} \log |\Delta|)$ primes. As indicated in [JMV09, p. 1497], the contribution of these primes can be absorbed into the error term $O(x^{1/2} \log(x) \log(xq))$, and hence does not affect the conclusion of the theorem. □

6.2 Evaluating Isogenies Under GRH

In this section, we describe a new algorithm to evaluate the horizontal isogeny corresponding to a given kernel. In contrast with the algorithm of Chapter 5, this algorithm relies on no heuristic assumptions other than GRH. In terms of performance, this algorithm is slightly slower, although its running time is still $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2})$. The algorithm takes as input a discriminant Δ, an elliptic curve E, a point P, and a kernel ideal $\mathfrak{L}$, and outputs $\phi(P)$, where $\phi\colon E \to E'$ is the normalized horizontal isogeny corresponding to $\mathfrak{L}$.

For convenience, we denote $L_{\max\{|\Delta|,q\}}(\frac{1}{2}, c)$ by $L(\frac{1}{2}, c)$.

In this section, we describe the steps in our algorithm. In Section 6.3 we show that, under GRH, our algorithm has a running time of $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2})$, which is subexponential in the

Algorithm 6 Computing a factor base

```
Input: An imaginary quadratic discriminant $\Delta < 0$ and a parameter $z$
Output: A factor base $\mathcal{F}$, or nil
Set $L \leftarrow \lceil L(\frac{1}{2}, z) \rceil$, $k \leftarrow \lceil \ln L \rceil$, $\mathcal{F} \leftarrow \emptyset$
for all primes $p < L$ do
   if kronecker$(\Delta, p) = 1$ then
      $i \leftarrow 0$
      repeat
         $i \leftarrow i + 1$
         $g \leftarrow$ primeForm$(\Delta, p)$
      until $i > 2k$ or $g \neq$ nil
      if $g \neq$ nil then
         $\mathcal{F} \leftarrow \mathcal{F} \cup \{g, g^\sigma\}$
      else
         Return nil
      end if
   end if
end for
Return $\mathcal{F}$
```

input size. We stress that although similar algorithms have appeared in several previous works, our algorithm is the first to achieve provably subexponential running time without appealing to any conditional hypotheses other than GRH.

We present our algorithm in several stages.

Computing a factor base. Algorithm 6 computes a factor base for $\mathrm{Cl}(O_\Delta)$ consisting of all split primes up to $L(\frac{1}{2}, z)$. The optimal value of the parameter z is determined in Section 6.3. The algorithm is based on, and indeed almost identical to, Algorithm 11.1 in [BV07]. The subroutine `primeForm` [BV07, §3.4] calculates a quadratic form corresponding to a prime ideal of norm p, and the subroutine `kronecker` [BV07, §3.4.3] calculates the Kronecker symbol. The map σ denotes complex conjugation.

Computing a relation. Given a factor base $\mathcal{F} = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_f\}$ and an ideal class $[\mathfrak{b}] \in \mathrm{Cl}(O_\Delta)$, Algorithm 7 produces a relation vector $\mathbf{z} = (z_1, \ldots, z_f) \in \mathbb{Z}^f$ for $[\mathfrak{b}]$ satisfying $[\mathfrak{b}] = \mathcal{F}^{\mathbf{z}} := \mathfrak{p}_1^{z_1} \cdots \mathfrak{p}_f^{z_f}$, with the additional property that the L^∞-norm $|\mathbf{z}|_\infty$ of $\mathbf{z}$ is less than $O(\ln |\Delta|)$ for some absolute implied constant (cf. Proposition 6.3.5). It is similar to Algorithm 11.2 in [BV07], except that we impose a constraint on $|\mathbf{v}|_\infty$ in line 1 in order to keep $|\mathbf{z}|_\infty$ small, and (for performance reasons) we use Bernstein's algorithm instead of

Algorithm 7 Computing a relation

```
Input: A discriminant $\Delta < 0$, a parameter $z$, a factor base $\mathcal{F}$ of size $f$, an ideal class $[\mathfrak{b}] \in \mathrm{Cl}(\mathcal{O}_\Delta)$, and an integer $t$ satisfying $C^{\frac{\ln|\mathrm{Cl}(\mathcal{O}_\Delta)|}{\ln\ln|\Delta|}} \leq t \leq C \ln|\Delta|$ where $C$ is the constant of Theorem 6.1.1/Corollary 6.1.2
Output: A relation vector $\mathbf{z} \in \mathbb{Z}^f$ such that $[\mathfrak{b}] = [\mathcal{F}^{\mathbf{z}}]$, or nil
Set $\mathcal{S} \leftarrow \emptyset$, $\mathcal{P} \leftarrow \{\mathrm{Norm}(\mathfrak{p}) : \mathfrak{p} \in \mathcal{F}\}$
Set $\ell \leftarrow L(\frac{1}{2}, \frac{1}{4z})$
for $i = 0$ to $\ell$ do
    Select $\mathbf{v} \in \mathbb{Z}^f_{0..|\Delta|-1}$ uniformly at random subject to the condition that $|\mathbf{v}|_\infty = t$
    Calculate the reduced ideal $\mathfrak{a}_{\mathbf{v}}$ in the ideal class $[\mathfrak{b}] \cdot [\mathcal{F}^{\mathbf{v}}]$
    Set $\mathcal{S} \leftarrow \mathcal{S} \cup \mathrm{Norm}(\mathfrak{a}_{\mathbf{v}})$
end for
Using Bernstein's algorithm [Ber], find a $\mathcal{P}$-smooth element $\mathrm{Norm}(\mathfrak{a}_{\mathbf{v}}) \in \mathcal{S}$ (if there exists one), or else return nil
Find the prime factorization of the integer $\mathrm{Norm}(\mathfrak{a}_{\mathbf{v}})$
Using Seysen's algorithm [Sey87, Thm. 3.1] on the prime factorization of $\mathrm{Norm}(\mathfrak{a}_{\mathbf{v}})$, factor the ideal $\mathfrak{a}_{\mathbf{v}}$ over $\mathcal{F}$ to obtain $\mathfrak{a}_{\mathbf{v}} = \mathcal{F}^{\mathbf{a}}$ for some $\mathbf{a} \in \mathbb{Z}^f$
Return $\mathbf{z} = \mathbf{a} - \mathbf{v}$
```

trial division to find smooth elements.

We remark that Corollary 9.3.12 of [BV07] together with the restriction $C > 1$ in Theorem 6.1.1 implies that there exists a value of t satisfying the inequality in Algorithm 7.

Computing $\phi(P)$. Algorithm 8 evaluates $\phi(P)$, where $\phi\colon E \to E'$ is the normalized isogeny corresponding to the kernel ideal $\mathfrak{L}$.

6.3 Running Time Analysis

Here we determine the theoretical running time of Algorithm 8, as well as the optimal value of the parameter z in Algorithm 6. As before, these two quantities depend on each other, and hence both are calculated simultaneously.

Proposition 6.3.1. *Algorithm 6 takes time $L(\frac{1}{2}, z)$ and succeeds with probability at least* 1/4.

Proof. Since Algorithm 6 is identical to Algorithm 11.1 in [BV07], the proposition follows from Lemmas 11.3.1 and 11.3.2 of [BV07]. □

Algorithm 8 Evaluating prime degree isogenies

```
Input: A discriminant $\Delta < 0$, an elliptic curve $E/\mathbb{F}_q$ with $\mathrm{End}(E) = \mathcal{O}_\Delta$, a point $P \in E(\mathbb{F}_q)$ such that $[\mathrm{End}(E) : \mathbb{Z}[\mathrm{Frob}_q]]$ and $\#E(\mathbb{F}_q)$ are coprime, and an $\mathrm{End}(E)$-ideal $\mathfrak{L} = (\ell, c + d\,\mathrm{Frob}_q)$ of prime norm $\ell \neq \mathrm{char}(\mathbb{F}_q)$ not dividing the index $[\mathrm{End}(E) : \mathbb{Z}[\mathrm{Frob}_q]]$.
Output: The unique elliptic curve $E'$ admitting a normalized isogeny $\phi\colon E \to E'$ with kernel $E[\mathfrak{L}]$, and the $x$-coordinate of $\phi(P)$ for $\Delta \neq -3, -4$ or the square (resp. cube) of the $x$-coordinate otherwise.
1: Using Algorithm 6, compute a factor base; discard any primes dividing $qn$ to obtain a new factor base $\mathcal{F} = \{\mathfrak{p}_1, \mathfrak{p}_2, \dots, \mathfrak{p}_f\}$
2: Using Algorithm 7 with any valid choice of $t$, compute a relation $\mathbf{z} \in \mathbb{Z}^f$ such that $[\mathfrak{L}] = [\mathcal{F}^{\mathbf{z}}] = [\mathfrak{p}_1^{z_1}\mathfrak{p}_2^{z_2}\cdots\mathfrak{p}_f^{z_f}]$
3: Compute a sequence of isogenies $(\phi_1, \dots, \phi_s)$ such that the composition $\phi_c\colon E \to E_c$ of the sequence has kernel $E[\mathfrak{p}_1^{z_1}\mathfrak{p}_2^{z_2}\cdots\mathfrak{p}_f^{z_f}]$, using the method of [BCL08, §3]
4: Using Cornacchia's algorithm, find a generator $\alpha \in \mathcal{O}_\Delta$ of the fractional ideal $\mathfrak{L}/(\mathfrak{p}_1^{z_1}\mathfrak{p}_2^{z_2}\cdots\mathfrak{p}_f^{z_f})$
5: Evaluate $\phi_c(P) \in E_c(\mathbb{F}_q)$
6: Write $\alpha = (u + v\,\mathrm{Frob}_q)/z$, compute the isomorphism $\eta\colon E_c \overset{\sim}{\to} E'$ with $\eta^*(\omega_{E'}) = (u/z)\omega_{E_c}$, and compute $Q = \eta(\phi_c(P))$
7: Compute $z^{-1} \bmod \#E(\mathbb{F}_{q^n})$ and $R = (z^{-1}(u + v\,\mathrm{Frob}_q))(Q)$
8: Put $r = x(R)^{|\mathcal{O}_\Delta^*|/2}$ and return $(E', r)$
```

Proposition 6.3.2. *The running time of Algorithm 7 is at most $L(\frac{1}{2}, z) + L(\frac{1}{2}, \frac{1}{4z})$, assuming GRH.*

Proof. Line 1 of the algorithm requires $L(\frac{1}{2}, z)$ norm computations. Line 2 is negligible. Line 5 requires $C \ln |\Delta|$ multiplications in the class group, each of which requires $O((\ln |\Delta|)^{1+\varepsilon})$ bit operations [Sch91]. Hence the **for** loop in lines 3–7 has running time $L(\frac{1}{2}, \frac{1}{4z})$. Bernstein's algorithm [Ber] in line 8 has a running time of $b(\log_2 b)^{2+\varepsilon}$ where $b = L(\frac{1}{2}, z) + L(\frac{1}{2}, \frac{1}{4z})$ is the combined size of $\mathcal{S}$ and $\mathcal{P}$. Finding the prime factorization in line 9 costs $L(\frac{1}{2}, z)$ using trial division, and Seysen's algorithm [Sey87, Thm. 3.1] in line 10 has negligible cost under ERH (and hence GRH). Accordingly, we find that the running time is

$$L(\tfrac{1}{2}, z) + O((\ln |\Delta|)^{2+\varepsilon}) \cdot L(\tfrac{1}{2}, \tfrac{1}{4z}) + b(\log_2 b)^{2+\varepsilon} + L(\tfrac{1}{2}, z) = L(\tfrac{1}{2}, z) + L(\tfrac{1}{2}, \tfrac{1}{4z}),$$

as desired. □

Proposition 6.3.3. *Under GRH, the probability that a single iteration of the **for** loop of Algorithm 7 produces an $\mathcal{F}$-smooth ideal $\mathfrak{a}_{\mathbf{v}}$ is at least $L(\frac{1}{2}, -\frac{1}{4z})$.*

Proof. We adopt the notation used in Theorem 6.1.1 and Corollary 6.1.2. Apply Corollary 6.1.2 with the values $m = qn$, $B = 3$, and $x = f = L(\frac{1}{2}, z) \gg (\ln|\Delta|)^B$. The ideal class $[\mathfrak{b}] \cdot [\mathcal{F}^{\mathbf{v}}]$ is equal to the ideal class obtained by taking the walk of length t in the Cayley graph $\mathrm{Cay}(G, S_x)$, having initial vertex $[\mathfrak{b}]$, and whose edges correspond to the nonzero coordinates of the vector $\mathbf{v}$. Hence a random choice of vector $\mathbf{v}$ under the constraints of Algorithm 7 yields the same probability distribution as a random walk in $\mathrm{Cay}(G, S_x)$ starting from $[\mathfrak{b}]$.

Let S be the set of reduced ideals in G with $L(\frac{1}{2}, z)$-smooth norm. By [BV07, Lemma 11.4.4], $|S| \geq \sqrt{|\Delta|}L(\frac{1}{2}, -\frac{1}{4z})$. Hence, by Corollary 6.1.2, the probability that $\mathfrak{a}_{\mathbf{v}}$ lies in S is at least

$$\frac{1}{2}\frac{|S|}{|G|} = \frac{1}{2} \cdot \frac{\sqrt{|\Delta|}}{|G|} \cdot L(\tfrac{1}{2}, -\tfrac{1}{4z}).$$

Finally, Theorem 9.3.11 of [BV07] states that $\frac{\sqrt{|\Delta|}}{|G|} \geq \frac{1}{\ln|\Delta|}$. Hence the probability that $\mathfrak{a}_{\mathbf{v}}$ is $\mathcal{F}$-smooth is at least

$$\frac{1}{2} \cdot \frac{1}{\ln|\Delta|} \cdot L(\tfrac{1}{2}, -\tfrac{1}{4z}) = L(\tfrac{1}{2}, -\tfrac{1}{4z}).$$

The result follows. □

Corollary 6.3.4. *Under GRH, the probability that Algorithm 7 succeeds is at least* $1 - \frac{1}{e}$.

Proof. Algorithm 7 loops through $\ell = L(\frac{1}{2}, \frac{1}{4z})$ vectors $\mathbf{v}$, and by Proposition 6.3.3, each such choice of $\mathbf{v}$ has an independent $1/\ell$ chance of producing a smooth ideal $\mathfrak{a}_{\mathbf{v}}$. Therefore the probability of success is at least

$$1 - \left(1 - \frac{1}{\ell}\right)^{\ell} > 1 - \frac{1}{e},$$

as desired. □

The following proposition shows that the relation vector $\mathbf{z}$ produced by Algorithm 7 is guaranteed to have small coefficients.

Proposition 6.3.5. *Any vector* $\mathbf{z}$ *output by Algorithm 7 satisfies* $|\mathbf{z}|_\infty < (C+1)\ln|\Delta|$.

Proof. Since $\mathbf{z} = \mathbf{a} - \mathbf{v}$, we have $|\mathbf{z}|_\infty \leq |\mathbf{a}|_\infty + |\mathbf{v}|_\infty$. But $|\mathbf{v}|_\infty \leq C\ln|\Delta|$ by construction, and the norm of $\mathfrak{a}_v$ is less than $\sqrt{|\Delta|/3}$ [BV07, Prop. 9.1.7], which implies

$$|\mathbf{a}|_\infty < \log_2\sqrt{|\Delta|/3} < \log_2\sqrt{|\Delta|} < \ln|\Delta|.$$

This completes the proof. □

Finally, we analyze the running time of Algorithm 8.

Theorem 6.3.6. *Under GRH, Algorithm 8 succeeds with probability at least* $\frac{1}{4}(1-\frac{1}{e})$ *and runs in time at most*

$$L(\tfrac{1}{2}, \tfrac{1}{4z}) + \max\{L(\tfrac{1}{2}, 3z), L(\tfrac{1}{2}, z)(\ln q)^{3+\varepsilon}\}.$$

Proof. We have shown that Algorithm 6 has running time $L(\frac{1}{2}, z)$ and success probability at least 1/4, and Algorithm 7 has running time $L(\frac{1}{2}, z) + L(\frac{1}{2}, \frac{1}{4z})$ and success probability at least $1-\frac{1}{e}$. Assuming that both these algorithms succeed, the computation of the individual isogenies ϕ_i in line 3 of Algorithm 8 proceeds in one of two ways, depending on whether the characteristic of $\mathbb{F}_q$ is large [BCL08, §3.1] or small [BCL08, §3.2]. The large characteristic algorithm fails when the characteristic is small, whereas the small characteristic algorithm succeeds in all situations, but is slightly slower in large characteristic. For simplicity, we consider only the more general algorithm.

The general algorithm proceeds in two steps. In the first step, we compute the kernel polynomial of the isogeny. The time to perform one such calculation is $O((\ell(\ln q)\max(\ell, \ln q)^2)^{1+\varepsilon})$ in all cases ([LS08, Thm. 1] for characteristic ≥ 5 and [def, Thm. 1] for characteristic 2 or 3). In the second step, we evaluate the isogeny using Vélu's formulae [Vél71]. This second step has a running time of $O(\ell^{2+\varepsilon}(\ln q)^{1+\varepsilon})$ [IJ, p. 214]. Hence the running time of line 3 is at most

$$|\mathbf{z}|_\infty (O((\ell(\ln q)\max(\ell, \ln q)^2)^{1+\varepsilon}) + O(\ell^{2+\varepsilon}(\ln q)^{1+\varepsilon})).$$

By Proposition 6.3.5, this expression is at most

$$\begin{aligned}(C+1)(\ln|\Delta|)(\max\{L(\tfrac{1}{2}, 3z), L(\tfrac{1}{2}, z)(\ln q)^{3+\varepsilon}\} + L(\tfrac{1}{2}, 2z)(\ln q)^{1+\varepsilon})&\\ = \max\{L(\tfrac{1}{2}, 3z), L(\tfrac{1}{2}, z)(\ln q)^{3+\varepsilon}\}.&\end{aligned}$$

Since the running time of all other lines in Algorithm 8 is bounded by that of line 3, the theorem follows. □

Corollary 6.3.7. *Under GRH, Algorithm 8 has a worst-case running time of at most* $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2})$.

Proof. Using the inequality $|\Delta| \leq 4q$, we may rewrite Theorem 6.3.6 in terms of q. We obtain

$$L(\tfrac{1}{2}, \tfrac{1}{4z}) + \max\{L(\tfrac{1}{2}, 3z), L(\tfrac{1}{2}, z)(\ln q)^{3+\varepsilon}\} \leq L_q(\tfrac{1}{2}, \tfrac{1}{4z} + 3z).$$

The optimal choice of $z = \frac{1}{2\sqrt{3}}$ yields the running time bound of $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2})$. □

Chapter 7

Future Work

There are a number of directions in which further research in this area can proceed. The fundamental problem is to be able to efficiently evaluate isogenies between elliptic curves. There are different instances of this problem, which depend on the input information. One type of input is two isogenous elliptic curves, and we want to efficiently evaluate an isogeny between them. The second type is the one we consider in this thesis: given the starting elliptic curve and the degree of the isogeny, we want to find the image curve and evaluate the isogeny.

By the term *efficiently*, we usually mean polynomial time; however in some cases it seems that achieving polynomial running time may not be possible using a classical computer, although we are not aware of any evidence (such as reductions from other hard problems) that this is the case. Currently most of these problems have exponential running times, and improving this by lowering the exponent or achieving subexponential time represents significant progress.

Thus as one of our future goals, we can try to speed up the presented algorithm even further. We believe that a polynomial algorithm is not possible for this problem in the classical setting. However, using quantum algorithms, it could be possible to design a polynomial time quantum algorithm for computing isogenies. This may be hard, since few computational problems admit exponential quantum speedups, but we believe it may be possible. The basic approach is to replace the random walks in this work with quantum walks. Childs et al. [CCD$^+$03] have shown that random quantum walks in certain types of graphs can exponentially speed up classical algorithms based on random walks.

Another problem to consider for further research is to try finding a subexponential algorithm for evaluating isogenies, when the input is two isogenous elliptic curves. Currently the best known algorithm for this is exponential in running time, although subexponential time quantum algorithms have been developed based on this work [CJS10]. This problem has been analyzed already (for example in [JMV05]), but previous analyses were limited to low degree isogenies. This problem can be reanalyzed in view of the new algorithm presented in this thesis. In addition, we can also attempt to find a polynomial time quantum algorithm that solves this problem.

Notice how we considered only horizontal isogenies. For future work vertical isogenies can be explored. At present this would only be of theoretical interest since no examples of pairs of such curves are known. The current fastest algorithm known for solving this problem has a running time complexity of $O(n^{3/2})$ (where n is the degree of the isogeny) [GHS02, Gal99]. To start with, it would be a good idea to explore the possibility of using new isogeny constructions to construct examples of such curves. We do not believe that a subexponential algorithm is possible; however, improving this even to $o(n^{1/2})$ would imply that a non-generic DLOG attack on some curve extends to all curves isogenous to it.

One more direction for future work would be to consider supersingular curves. Although supersingular curves are impractical for many cryptographic applications, there do exist applications of isogenies where supersingular curves can be used. Thus it would be worth trying to develop a similar algorithm to the one presented in this thesis in the context of supersingular curves. The class group of an ordinary curve is abelian and acts on the corresponding elliptic curves. For supersingular curves, the corresponding class group is non-abelian and has a quaternion algebra structure. Similar techniques can be applied as in the presented algorithm, adjusting them to work with non-abelian structures.

Thus, in this area there is still a great amount of possible research that can be performed. Solving the above problems should give a rise to a number of cryptographic applications, and improve our understanding of isogenies.

Bibliography

[BCL08] R. Bröker, D. Charles, and K. Lauter, *Evaluating large degree isogenies and applications to pairing based cryptography*, Pairing '08: Proceedings of the 2nd international conference on Pairing-Based Cryptography (Berlin, Heidelberg), Springer-Verlag, 2008, pp. 100–112. 1, 2, 30, 36, 40, 42, 44, 46, 49, 51, 59, 61

[Ber] Daniel J. Bernstein, *How to find small factors of integers*, URL: http://cr.yp.to/papers.html. Mathematics of Computation (to appear). 58, 59

[BLS10] R. Bröker, K. Lauter, and A. Sutherland, *Modular polynomials via isogeny volcanoes*, 2010. 49, 53

[BMSS08] A. Bostan, F. Morain, B. Salvy, and É. Schost, *Fast algorithms for computing isogenies between elliptic curves*, Math. Comp. **77** (2008), no. 263, 1755–1778. 2, 32, 54

[BS09] G. Bisson and A. Sutherland, *Computing the endomorphism ring of an ordinary elliptic curve over a finite field*, Journal of Number Theory **to appear** (2009). 2, 43, 44, 45, 47, 52

[BV07] J. Buchmann and U. Vollmer, *Binary quadratic forms: An algorithmic approach*, Algorithms and Computation in Mathematics, vol. 20, Springer, Berlin, 2007. 42, 43, 46, 47, 57, 58, 60

[CCD$^+$03] Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman, *Exponential algorithmic speedup by a quantum walk*, Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing (New York), ACM, 2003, pp. 59–68 (electronic). 62

[cera] *Certicom ECC Challenge*, http://www.certicom.com/images/pdfs/cert_ecc_challenge.pdf. 52, 53

[cerb] *Certicom ECC Curves List*, http://www.certicom.com/index.php/curves-list. 52, 53

[CFA$^+$06] H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, and F. Vercauteren (eds.), *Handbook of elliptic and hyperelliptic curve cryptography*, Discrete Mathematics and its Applications (Boca Raton), Chapman & Hall/CRC, Boca Raton, FL, 2006. 1, 24, 46

[CJS10] Andrew Childs, David Jao, and Vladimir Soukharev, *Constructing elliptic curve isogenies in quantum subexponential time*, 2010, preprint. 55, 62

[Coh93] Henri Cohen, *A course in computational algebraic number theory*, Graduate Texts in Mathematics, vol. 138, Springer-Verlag, Berlin, 1993. 47

[Cox89] D. A. Cox, *Primes of the form x^2+ny^2*, A Wiley-Interscience Publication, John Wiley & Sons Inc., New York, 1989, Fermat, class field theory and complex multiplication. 15, 16, 18, 30, 47, 56

[def] *Fast algorithms for computing isogenies between ordinary elliptic curves in small characteristic*, Journal of Number Theory **In Press, Corrected Proof**. 33, 61

[Eng09] A. Enge, *Computing modular polynomials in quasi-linear time*, Math. Comp. **78** (2009), no. 267, 1809–1824. 42

[FM02] M. Fouquet and F. Morain, *Isogeny volcanoes and the SEA algorithm*, Algorithmic number theory (Sydney, 2002), Lecture Notes in Comput. Sci., vol. 2369, Springer, Berlin, 2002, pp. 276–291. 16, 17, 18

[Gal99] S. D. Galbraith, *Constructing isogenies between elliptic curves over finite fields*, LMS J. Comput. Math. **2** (1999), 118–138 (electronic). 1, 16, 18, 63

[GHS02] S. D. Galbraith, F. Hess, and N. P. Smart, *Extending the GHS Weil descent attack*, Advances in cryptology—EUROCRYPT 2002 (Amsterdam), Lecture Notes in Comput. Sci., vol. 2332, Springer, Berlin, 2002, pp. 29–44. 1, 37, 43, 63

[HM89] J. Hafner and K. McCurley, *A rigorous subexponential algorithm for computation of class groups*, J. Amer. Math. Soc. **2** (1989), no. 4, 837–850. 2, 43

[HMW90] Kenneth Hardy, Joseph B. Muskat, and Kenneth S. Williams, *A deterministic algorithm for solving $n = fu^2 + gv^2$ in coprime integers u and v*, Math. Comp. **55** (1990), no. 191, 327–343. 38, 47, 48

[IJ] Sorina Ionica and Antoine Joux, *Pairing the volcano*, Algorithmic number theory: Proceedings of ANTS-IX (Guillaume Hanrot, François Morain, and Emmanuel Thomé, eds.). 30, 61

[JMV05] D. Jao, S. D. Miller, and R. Venkatesan, *Do all elliptic curves of the same order have the same difficulty of discrete log?*, Advances in cryptology—ASIACRYPT 2005, Lecture Notes in Comput. Sci., vol. 3788, Springer, Berlin, 2005, pp. 21–40. 63

[JMV09] David Jao, Stephen D. Miller, and Ramarathnam Venkatesan, *Expander graphs based on GRH with an application to elliptic curve cryptography*, J. Number Theory **129** (2009), no. 6, 1491–1504. 56

[JS10] David Jao and Vladimir Soukharev, *A subexponential algorithm for evaluating large degree isogenies*, Algorithmic number theory: Proceedings of ANTS-IX (Guillaume Hanrot, François Morain, and Emmanuel Thomé, eds.), Lecture Notes in Comput. Sci., vol. 6197, Springer–Verlag, 2010, pp. 219–233. 42

[Koh96] D. Kohel, *Endomorphism rings of elliptic curves over finite fields*, Ph.D. thesis, University of California, Berkeley, 1996. 18

[Lan87] Serge Lang, *Elliptic functions*, second ed., Graduate Texts in Mathematics, vol. 112, Springer-Verlag, New York, 1987, With an appendix by J. Tate. 19

[Len87] H. W. Lenstra, Jr., *Factoring integers with elliptic curves*, Ann. of Math. (2) **126** (1987), no. 3, 649–673. 46

[LS08] Reynald Lercier and Thomas Sirvent, *On Elkies subgroups of l-torsion points in elliptic curves defined over a finite field*, J. Théor. Nombres Bordeaux **20** (2008), no. 3, 783–797. 33, 61

[mag] *MAGMA Computational Algebra System*, http://magma.maths.usyd.edu.au/. 51

[MOV91] Alfred Menezes, Tatsuaki Okamoto, and Scott Vanstone, *Reducing elliptic curve logarithms to logarithms in a finite field*, STOC '91: Proceedings of the twenty-third annual ACM symposium on theory of computing (New York, NY, USA), ACM, 1991, pp. 80–89. 15

[Pol78] J. M. Pollard, *Monte Carlo methods for index computation* (mod p), Math. Comp. **32** (1978), no. 143, 918–924. 24

[Sch91] Arnold Schönhage, *Fast reduction and composition of binary quadratic forms*, ISSAC '91: Proceedings of the 1991 international symposium on Symbolic and algebraic computation (New York, NY, USA), ACM, 1991, pp. 128–133. 48, 59

[Sch95] R. Schoof, *Counting points on elliptic curves over finite fields*, J. Théor. Nombres Bordeaux **7** (1995), no. 1, 219–254, Les Dix-huitièmes Journées Arithmétiques (Bordeaux, 1993). 1, 14, 20, 30, 32

[Sey87] M. Seysen, *A probabilistic factorization algorithm with quadratic forms of negative discriminant*, Math. Comp. **48** (1987), no. 178, 757–780. 43, 44, 58, 59

[Sil92] J. Silverman, *The arithmetic of elliptic curves*, Graduate Texts in Mathematics, vol. 106, Springer-Verlag, New York, 1992, Corrected reprint of the 1986 original. 3, 8, 9, 11, 12, 13, 15, 16, 19, 20, 21, 35

[Sil94] Joseph H. Silverman, *Advanced topics in the arithmetic of elliptic curves*, Graduate Texts in Mathematics, vol. 151, Springer-Verlag, New York, 1994. 19

[Sut] A. Sutherland, *smoothrelation*, http://math.mit.edu/~drew/smoothrelation_v1.tar. 52, 53

[Tat66] J. Tate, *Endomorphisms of abelian varieties over finite fields*, Invent. Math. **2** (1966), 134–144. 1, 14

[Tes06] E. Teske, *An elliptic curve trapdoor system*, J. Cryptology **19** (2006), no. 1, 115–133. 24, 26

[Vél71] Jacques Vélu, *Isogénies entre courbes elliptiques*, C. R. Acad. Sci. Paris Sér. A-B **273** (1971), A238–A241. 28, 61