Post-Quantum Elliptic Curve Cryptography

by

Vladimir Soukharev

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2016

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

We propose and develop new schemes for post-quantum cryptography based on isogenies over elliptic curves. For our first contribution, we show that ordinary elliptic curves have less than exponential security against quantum computers. These results were used as the motivation for De Feo, Jao and Plût's construction of public key cryptosystems using supersingular elliptic curve isogenies. We extend their construction and show that isogenies between supersingular elliptic curves can be used as the underlying hard mathematical problem for other quantum-resistant schemes. For our second contribution, we propose an undeniable signature scheme based on elliptic curve isogenies. We prove its security under certain reasonable number-theoretic computational assumptions for which no efficient quantum algorithms are known. This proposal represents only the second known quantum-resistant undeniable signature scheme, and the first such scheme secure under a number-theoretic complexity assumption. Finally, we also propose a security model for evaluating the security of authenticated encryption schemes in the post-quantum setting. Our model is based on a combination of the classical Bellare-Namprempre security model for authenticated encryption together with modifications from Boneh and Zhandry to handle message authentication against quantum adversaries. We give a generic construction based on Bellare-Namprempre for producing an authenticated encryption protocol from any quantum-resistant symmetric-key encryption scheme together with any digital signature scheme or MAC admitting any classical security reduction to a quantum-computationally hard problem. Using this model, we show how to explicitly construct authenticated encryption schemes based on isogenies.

Acknowledgements

I would like to thank my supervisor David Jao.

Dedication

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

Contents

List of Figures **ix**

1 Introduction **1**

2 Isogenies and Applications to Cryptography **4**

2.1 Algebraic Curves 4

2.2 Elliptic Curves 10

2.3 Isogenies 11

2.4 The Endomorphism Ring of Elliptic Curve 15

2.5 Complex Multiplication and Group Action 21

2.6 Application: Stolbunov's Scheme 22

3 Computation of Isogenies Between Ordinary Elliptic Curves **24**

3.1 Introduction 24

3.2 Isogeny Graphs Under GRH 26

3.3 Computing the Action of $\mathrm{Cl}(\mathcal{O}_\Delta)$ on $\mathrm{Ell}(\mathcal{O}_\Delta)$ 27

3.4 Running Time Analysis 28

3.5 A Quantum Algorithm For Constructing Isogenies 32

4 Isogeny-Based Quantum-Resistant Key Exchange and Encryption **37**

4.1 Introduction 37

4.1.1 Ramanujan Graphs 38

4.1.2 Isogeny Graphs 39

4.2 Public-Key Cryptosystems Based On Supersingular Curves 39
4.2.1 Zero-Knowledge Proof of Identity 40
4.2.2 Key Exchange 42
4.2.3 Public-Key Encryption 43
4.3 Complexity Assumptions 44
4.3.1 Hardness Of The Underlying Assumptions 46
4.4 Security Results 48

5 Isogeny-Based Quantum-Resistant Undeniable Signatures 49
5.1 Introduction 49
5.2 Quantum-Resistant Undeniable Signatures From Isogenies 50
5.2.1 Definition 50
5.2.2 Protocol 51
5.3 Complexity Assumptions 54
5.3.1 Hardness Of The Underlying Assumptions 55
5.4 Security Proofs 56
5.4.1 Confirmation Protocol 56
5.4.2 Disavowal Protocol 57
5.4.3 Unforgeability and Invisibility 59
5.5 Parameter Sizes 59
5.6 Conclusion 60

6 Post-Quantum Security Models For Authenticated Encryption 61
6.1 Introduction 61
6.2 Security Definitions 62
6.3 Main Theorem 65
6.4 Quantum-Resistant Strongly Unforgeable Signature Schemes 71
6.4.1 Strong Designated Verifier Signatures from Isogenies 71
6.4.2 Ring-LWE Signatures 72
6.5 Quantum-Resistant Authenticated Encryption 73

6.6 Isogeny-Based Quantum-Resistant Authenticated Encryption Scheme . . . 74
6.7 Overhead Calculations and Comparisons 77
6.7.1 Communication Overhead 77
6.7.2 Public Key Overhead 77

7 Future Work **79**

Bibliography **81**

List of Figures

2.1 Isogeny volcano 18
2.2 Key agreement protocol by Stolbunov 22
2.3 Public key encryption protocol by Stolbunov 23

4.1 Key-exchange protocol using isogenies on supersingular curves. 43

5.1 Signature generation. 52
5.2 Confirmation protocol. 52
5.3 Disavowal protocol. 54
5.4 Proof of soundness (confirmation) 58
5.5 Proof of soundness (disavowal) 58
5.6 Confirmation ($b = 0$ case) 58
5.7 Confirmation ($b = 1$ case) 58
5.8 Disavowal ($b = 0$ case) 58
5.9 Disavowal ($b = 1$ case) 58

6.1 Games G_0, G_1, and G_2. 68
6.2 Isogenies in Authenticated Encryption Scheme 76

Chapter 1

Introduction

Elliptic curves, over time, have proven themselves to be a reliable mathematical tools for constructing cryptographic primitives. The resulting area of cryptography is known as *Elliptic Curve Cryptography* and remains the object of continued study. The theory of elliptic curves is well-established and plays an important role in many current areas of research in mathematics. However, in cryptography, applications of elliptic curves to practical cryptosystems have so far limited themselves only to the objects, that is, the actual elliptic curves, rather than the maps between the objects. In contrast, in mathematical research, the study of the maps or morphisms between objects typically demands equal if not more attention than that of the objects themselves. We believe that it is time to introduce into cryptography the use of maps, or *isogenies*, between elliptic curves as a direct component in the design and construction of cryptosystems. Such cryptosystems appear to be a good candidate for future *post-quantum* cryptosystems which are intended to be used in the event that quantum computers become a reality.

We currently live in an era where the future development of quantum computers is foreseeable. Many physicists and engineers believe that in about ten to twenty years we will start seeing quantum computers in practical use. The emergence of quantum computers is an exciting prospect for those who can take advantage of the extra computational capabilities that they offer. However, adversaries seeking to attack cryptosystems will also be able to take advantage of quantum computers. It is well-known that quantum computers can efficiently factor large integers and solve the discrete logarithm problem in finite groups using Shor's algorithm [Sho97]. Most modern-day cryptosystems are based on these two mathematical problems, which are safe against classical adversaries, but will not be safe against adversaries with quantum computers. One could in theory use quantum techniques such as quantum key distribution to achieve unbreakable encryption that is immune to attacks unconditionally, but we do not yet know whether these techniques will scale up to satisfy future demand. An alternative approach, called *Post-Quantum Cryptography*, aims

to develop cryptosystems for classical computers which would be secure against quantum adversaries.

In recent years, the topic of post-quantum cryptography has been the subject of a great deal of interest in the cryptographic research community. Existing families of post-quantum cryptosystems can be divided into five broad subcategories: lattice-based schemes, code-based schemes, hash-based schemes, multivariate polynomials-based schemes, and elliptic curve-based schemes. Of these, the first four families are firmly established in the literature; the fifth one, which represents our work, is less mainstream at the moment, but attracting increasing interest. More specifically, our post-quantum elliptic curve cryptosystems are based on isogenies, which are maps between elliptic curves. Compared to other families, our approach has a number of advantages: it is well-suited to key exchange and encryption, and can achieve signatures and authentication as well with slightly less efficiency. In addition, the relationship between the security parameter and the public parameters to be used in the system is more straightforward with isogenies than with other families such as lattice-based schemes. We anticipate another benefit to be that existing cryptographic libraries for elliptic curve arithmetic can be re-used or re-purposed for isogeny-based cryptography, providing a head start in designing high-performance implementations secure against side-channel attacks.

The underlying hard problem for isogeny-based cryptography is: given two isogenous *supersingular* elliptic curves, find an isogeny between them. Currently no quantum algorithm is known for solving this problem in general in less than exponential time. One of the main reasons why this problem seems intractable for quantum computers is that the endomorphism ring for the elliptic curve is non-commutative, which shields the problem against attacks like Shor's algorithm.

In this thesis, we start by providing the necessary mathematical background needed for understanding elliptic curves, isogenies, endmorphism rings and complex multiplication. This material can be found in Chapter 2. We also briefly describe Stolbunov's schemes from [Sto10].

In Chapter 3 we describe the computational theory of isogenies. We show how to evaluate isogenies between *ordinary* elliptic curves in subexponential running time (classically). This result appeared previously in my Master's thesis [Sou10], and portions were also published in [JS10] and [CJS14], but we include it here because it is necessary background for later chapters. We then consider the problem of finding isogenies between ordinary elliptic curves, and show that it can be done in subexponential running time on a quantum computer. This result has not previously appeared in any thesis, although it was also published in [CJS14]. This chapter shows that ordinary elliptic curves, though widely used in traditional elliptic curve cryptography, do not provide a good foundation for post-quantum cryptography. For this reason, in the rest of the thesis we consider only the case of non-ordinary, i.e. supersingular elliptic curves.

In Chapter 4, we review the existing constructions of isogeny-based cryptosystems using supersingular elliptic curve isogenies. We present the key exchange, public-key encryption, and zero-knowledge proof schemes from [JDF11] and [DFJP14], and discuss the mathematical hard problems on which their security is based. This chapter contains no contributions, but it is necessary background for explaining our contributions.

In Chapter 5, we present a quantum-resistant, isogeny-based undeniable signature scheme. Of course, as with most asymmetric cryptography, by *quantum-resistant* we mean that at present, no quantum algorithm is known, and the research indicates that most likely will not be known. We describe our protocol and prove that the scheme satisfies the required security properties against quantum adversaries. Specifically, we prove that the scheme is unforgeable and invisible and that the confirmation and disavowal protocols are complete, sound and zero-knowledge. These results were published in [JS14].

Finally, we present a quantum security model for authenticated encryption based on a combination of existing quantum security models for encryption and signature/MAC schemes and existing classical security models for authenticated encryption. We present a quantum analogue of the security model of Bellare and Namprempre [BN08] for quantum adversaries, and show that a quantum-resistant encryption scheme and a quantum-resistant MAC scheme combined using *encrypt-then-MAC* yields a quantum-resistant authenticated encryption scheme. These results were published in PQCrypto 2016 [SJS16]. As an application of our security model, we construct and present an authenticated encryption scheme based on supersingular elliptic curve isogenies. These results are presented in Chapter 6.

Chapter 2

Isogenies and Applications to Cryptography

In this chapter we give an in-depth treatment of the mathematical and computational theory of isogenies. We start with some mathematical background and definitions. We mention some examples of prior applications of isogenies to cryptography, including counting points on elliptic curves over a finite field and the transfer of discrete logarithms. We also present Stolbunov's scheme for encryption using isogenies over ordinary elliptic curves, which represents the first published isogeny-based cryptosystem.

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). 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 (one whose finite extensions are separable), $\bar{K}$ 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 $\mathbb{A}^n$ corresponding to I:

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

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 of $\{(\lambda x_0, \ldots, \lambda x_n)\}$ by $[x_0, \ldots, x_n]$, 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

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

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 $f \in K[X_0, \ldots, X_n]$ 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 $I \subset \bar{K}[X_0, \ldots, X_n]$ 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 $\bar{K}[X_0, \ldots, X_n]$ generated by

$$\{f \in \bar{K}[X_0, \ldots, X_n] : f \text{ is homogeneous and } f(P) = 0 \text{ for all } P \in V\}.$$

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 $K[X_0, \ldots, X_n]$. 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 $\bar{K}[X_0, \ldots, X_n]$.

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

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

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

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

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

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

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 a non-homogeneous polynomial $f(Y_1, \ldots, Y_n) \in \bar{K}[Y_1, \ldots, Y_n]$, 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

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

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

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

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]. □

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; hence the definition makes sense.

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 variable, 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

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

.

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 $P, Q \neq \infty$

When $\operatorname{char} F$ 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, 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 some 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. 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 $n = 2$,

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

- Note that the degree of $[n]$ is n^2.
- 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
$$\pi(x, y) = (x^q, y^q).$$
- 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 $\#\ker(\pi) = 1$.
- $\deg(\pi) \neq \#\ker(\pi)$ because π is *inseparable*.

Example 2.3.6. *Complex multiplication*

- Let F be a field such that $\sqrt{-1} \in F$.
- Let $E\colon y^2 = x^3 - x$ be defined over F.
- As usual let $i = \sqrt{-1} \in F$.
- 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*

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

that 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 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, including the constant homomorphism, $\mathrm{End}(E)$ 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, $\dim_{\mathbb{Z}} \mathrm{End}(E)$ 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 $\dim_{\mathbb{Z}} \mathrm{End}(E) = 4$, and *ordinary* if $\dim_{\mathbb{Z}} \mathrm{End}(E) = 2$.

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 traditionally more common to use ordinary curves because they are more secure for use in discrete logarithm-based schemes. The reason is that Menezes et al. [MOV91] showed that the discrete logarithm problem on a supersingular elliptic curve can be reduced to a discrete logarithm problem in a finite field, which is easier to solve; this reduction is referred to as the "MOV reduction." (The reduction applies to ordinary curves as well, but it does not speed up the computation of the DLOG in that case.) However, supersingular curves are actually more secure against quantum computers, when we use cryptosystems based on isogenies. These issues will be discussed at length throughout this thesis.

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

Definition 2.4.4. An *order* $\mathcal{O}$ in a quadratic field 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 d_K be the discriminant of K and let

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

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 d_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 D, then the discriminant of the maximal order $\mathcal{O}_K$ is the largest square-free part of D, i.e. $D = c^2 d_K$, where d_K is the discriminant of $\mathcal{O}_K$ and c is a conductor. We say that $\mathcal{O}$ is an *imaginary quadratic order* if $D < 0$, and a *real quadratic order* otherwise. We denote by $\mathcal{O}_\Delta$ an imaginary quadratic order of discriminant Δ.

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*

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

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

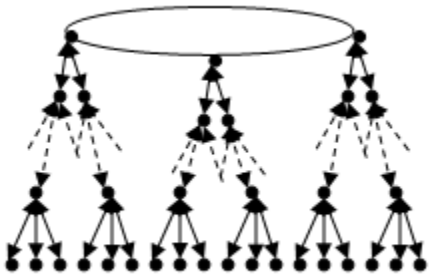

Figure 2.1: Isogeny volcano

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 $[\mathrm{End}(E) : \mathrm{End}(E')] = \ell$ (note that this means that $\mathrm{End}(E') \subset \mathrm{End}(E)$), *"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.

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 $\mathrm{End}(E)$ 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. [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

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

We state a theorem from [Cox89] that we will need.

Theorem 2.4.9. *Let $L \subseteq \mathbb{C}$ be a lattice. Then for a number $\alpha \in \mathbb{C} \setminus \mathbb{Z}$, 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* $\ker\phi$ *is a fractional ideal of* $\mathrm{End}(E)$.

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 the points in Φ lifted to $\mathbb{C}$ form the lattice for E'. Hence, Theorem 2.4.9(a) holds. Therefore 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 because it is usually impractical to express isogenies algebraically. Rather than expressing the isogeny ϕ directly, we can represent it using its kernel $\ker\phi$.

We have mostly discussed the ordinary elliptic curves and the structure of their endomorphism rings. We now need to briefly examine the same for supersingular elliptic curves. In this case, the structure theory is less well-developed, and in particular there is no known analogue of the "volcano" structure that is present for ordinary curves. What is known is that supersingular curves can always be defined over $\mathbb{F}_{p^2}$, and for every prime ℓ that does not divide p, there exist $\ell + 1$ isogenies (counting multiplicities) of degree ℓ originating from each such supersingular curve.

The structure of the endomorphism ring of a supersingular elliptic curve is that of an order in a quaternion algebra, which we define here:

Definition 2.4.12. A quaternion algebra over $\mathbb{Q}$ is an algebra of the form

$$\mathbb{Q} + \mathbb{Q}\alpha + \mathbb{Q}\beta + \mathbb{Q}\alpha\beta,$$

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

Corollary 2.4.13. *The endomorphism ring of an elliptic curve is either $\mathbb{Z}$, an order in a quadratic imaginary field, or an order in a quaternion algebra. In characteristic zero, only the first two are possible, and in a finite field, only the latter two are possible.*

Proof. [Sil92, III.9.4]. □

These results show that, since the endomorphism ring of a supersingular elliptic curve has rank 4, it must be an order in a quaternion algebra. Observe that, unlike the ordinary case with an imaginary quadratic order, the endomorphism ring of a supersingular elliptic curve is non-commutative.

2.5 Complex Multiplication and Group Action

We know that over the complex numbers, every elliptic curve E is isomorphic to $\mathbb{C}/\Lambda$ for some lattice $\Lambda = \langle w_1, w_2 \rangle$. In more detail, let $\Lambda \subset \mathbb{C}$ be a lattice. Then:

- $\wp(z) = \frac{1}{z^2} + \sum_{\substack{w \in \Lambda \\ w \neq 0}} \left(\frac{1}{(z-w)^2} - \frac{1}{w^2} \right)$
- $G_4 = \sum_{\substack{w \in \Lambda \\ w \neq 0}} \frac{1}{w^4}, G_6 = \sum_{\substack{w \in \Lambda \\ w \neq 0}} \frac{1}{w^6}$
- $\phi(z) = (\wp(z), \wp'(z)/2)$
- $E\colon y^2 = x^3 - 15G_4 x - 35G_6$

As already seen in previous sections, the endomorphism ring of an elliptic curve over a finite field is larger than $\mathbb{Z}$ and hence in this case the curve has *complex multiplication*. We state here a few important consequences of complex multiplication, but for more details we refer to [Wat69], [Lan87] and [Sil94, II].

We first state the following useful theorem:

Theorem 2.5.1. *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.

We need to define the following set:

Definition 2.5.2. The set of isomorphism classes of elliptic curves $E/\mathbb{F}_q$ with $\mathrm{End}(E) = \mathcal{O}_K$ is denoted $\mathrm{Ell}_{p,n}(\mathcal{O}_K)$, where $n = \#E$.

Thus, let us denote by $\phi_\mathfrak{b}\colon E \to E_\mathfrak{b}$ the isogeny corresponding to an ideal $\mathfrak{b}$ (keeping in mind that $\phi_\mathfrak{b}$ is only defined up to isomorphism of $E_\mathfrak{b}$). Principal ideals correspond to endomorphisms, so any other ideal equivalent to $\mathfrak{b}$ in the ideal class group $\mathrm{Cl}(\mathcal{O}_\Delta)$ of $\mathcal{O}_\Delta$

yields the same codomain curve $E_{\mathfrak{b}}$, up to isomorphism [Wat69, Thm. 3.11]. Hence one obtains a well-defined group action $*\colon \mathrm{Cl}(\mathcal{O}_\Delta) \times \mathrm{Ell}_{q,n}(\mathcal{O}_\Delta) \to \mathrm{Ell}_{q,n}(\mathcal{O}_\Delta)$ taking $[\mathfrak{b}] * j(E)$ to $j(E_{\mathfrak{b}})$, where $[\mathfrak{b}]$ denotes the ideal class of $\mathfrak{b}$. This group action, which we call the *complex multiplication operator*, is free and transitive [Wat69, Thm. 4.5], and thus $\mathrm{Ell}_{q,n}(\mathcal{O}_\Delta)$ forms a principal homogeneous space over $\mathrm{Cl}(\mathcal{O}_\Delta)$.

2.6 Application: Stolbunov's Scheme

In this section we briefly present two examples by Stolbunov [Sto10] of cryptosystems based on isogenies between ordinary elliptic curves. One scheme is for key exchange and the other is for public key encryption.

For the following two schemes, we let x be an ordinary elliptic curve over some finite field. We let G be a set of isogenies in $\mathrm{End}(x)$, but in practice should be all of $End(x)$. Finally we let $\mathcal{H} = \{H_k : k \in K\}$ be a set of secure hash functions indexed by a finite set K. (The family $\mathcal{H}$ is needed to be able to prove the security of the scheme.)

$\mathcal{A}$		$\mathcal{B}$
Input: –		**Input:** –
$a \xleftarrow{R} G$		$b \xleftarrow{R} G$
$m_A \leftarrow a * x$		$m_B \leftarrow b * x$
	$\xrightarrow{m_A}$	
	$\xleftarrow{m_B}$	
$k_A \leftarrow a * m_B$		$k_B \leftarrow b * m_A$
Output: k_A		**Output:** k_B

Figure 2.2: Key agreement protocol by Stolbunov

We also give the statements of the security proofs from Stolbunov's paper.

Theorem 2.6.1. *The key exchange protocol in figure 2.2 is session-key (SK) secure in the authenticated-links adversarial model (AM).*

Proof. [Sto10, Theorem 1]. □

Theorem 2.6.2. *The public-key encryption protocol in figure 2.3 is secure in the sense of IND-CPA.*

$\mathcal{K}$: Key generation
Input: -
$\text{sk} \xleftarrow{R} G$
$y \leftarrow \text{sk} * x$
$k \xleftarrow{R} K$
$\text{pk} \leftarrow (y, k)$
Output: sk, pk

$\mathcal{E}$: Encryption
Input: pk, m
$a \xleftarrow{R} G$
$u \leftarrow a * y$
$h \leftarrow H_k(u)$
$z \leftarrow a * x$
$c \leftarrow h \oplus m$
$\text{ct} \leftarrow (c, z)$
Output: ct

$\mathcal{D}$: Decryption
Input: sk, pk, ct
$u \leftarrow \text{sk} * z$
$h \leftarrow H_k(u)$
$m \leftarrow h \oplus c$
Output: m

Figure 2.3: Public key encryption protocol by Stolbunov

Proof. [Sto10, Theorem 2]. □

These security results only hold if the underlying mathematical problem of computing isogenies between ordinary elliptic curves is hard. Although this problem seems to be hard for classical computers, we show in the next chapter that it is easier to solve on a quantum computer.

Chapter 3

Computation of Isogenies Between Ordinary Elliptic Curves

3.1 Introduction

This chapter consists of material from our published article in the Journal of Mathematical Cryptology [CJS14], co-authored with Andrew Childs and my supervisor David Jao. Some (but not all) of this material also appeared in [Sou10] (my Master's Thesis). The last section of this chapter has not previously appeared in any thesis, and serves to motivate the use of supersingular elliptic curve isogenies in post-quantum cryptography.

We address two notions in this chapter - evaluating isogenies and computing or constructing isogenies. Evaluating isogenies means that whenever we are given an elliptic curve and an isogeny mapping from it, we wish to evaluate that isogeny. Computing or constructing isogenies means that whenever we are given to isogenous elliptic curves, we wish to find the isogeny between them.

In this chapter we present and describe in details our algorithm for evaluating large prime degree isogenies, having subexponential running time in the size of the magnitude of the discriminant of the endomorphism ring of the elliptic curve and polynomial in the size of the degree of the isogeny. The proofs of correctness and the running-time analysis rely on only one standard assumption, namely the Generalized Riemann Hypothesis (GRH).

Our first objective is to evaluate an isogeny of large degree in subexponential time, given a compact representation. Specifically, we wish to evaluate the unique horizontal normalized [BCL08] isogeny on a given elliptic curve $E/\mathbb{F}_q$ whose kernel ideal in $\mathrm{End}(E)$ 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, π_q denotes the Frobenius map on E, and c and d are rational numbers specifying the ideal $\mathfrak{L}$

and hence the isogeny. 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], 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 $|\Delta|$ (where Δ is the discriminant of $\mathrm{End}(E)$), and therefore even this method 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 give 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

$$\mathfrak{L} = I_1^{e_1} I_2^{e_2} \cdots I_k^{e_k} \cdot (\alpha) \tag{3.1}$$

as in lines 2 and 3 of the BCL algorithm (Algorithm 4.1 in [BCL08]). The other problem is that the exponents e_i in Equation (3.1) need to be kept small, since the running times of lines 3 and 4 of Algorithm 4.1 in [BCL08] are proportional to $\sum_i |e_i| \,\mathrm{Norm}(I_i)^2$. 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 [BS11]. 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. The details of this approach can be found in [JS10].

We present a variant of the algorithm. Our variant improves on the above described 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.

Finally, in this chapter, we give a subexponential-time quantum algorithm for constructing a nonzero isogeny between two given elliptic curves (of the type arising in the aforementioned cryptosystems). We show that the running time of our algorithm is bounded above by $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2})$ under (only) the Generalized Riemann Hypothesis (GRH), where

$$L_N(\alpha, c) := \exp[(c + o(1))(\ln N)^{\alpha}(\ln \ln N)^{1-\alpha}].$$

This result raises serious questions about the viability of isogeny-based cryptosystems over ordinary curves in the context of quantum computers.

3.2 Isogeny Graphs Under GRH

Let $\mathrm{Cl}(\mathcal{O}_\Delta)$ denote the ideal class group of $\mathcal{O}_\Delta$. We want to reduce the number of heuristic assumptions used in Algorithm 3 from [JS10]. In fact, we will remove all of them, except for GRH. Although this change makes the algorithm slower in practice, its asymptotic running time is still unchanged. We start with some results on isogeny graphs, which we need as part of our running-time analysis. The running-time analysis in Section 3.3 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 3.2.1. *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*

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

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 3.2.1 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 3.2.2. *For any fixed integer m, Theorem 3.2.1 holds even if the definition of the set A is changed to*

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

Proof. The alternative definition of A differs from the original definition by at most $O(\ln q)$ primes. As stated in [JMV09, p. 1497], such a change does not affect the conclusion of the theorem. □

3.3 Computing the Action of $\mathrm{Cl}(\mathcal{O}_\Delta)$ on $\mathrm{Ell}(\mathcal{O}_\Delta)$

In this section, we describe a new algorithm to evaluate the horizontal isogeny corresponding to a given kernel. In contrast with the Algorithm 4 in [JS10], 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 $L_{|\Delta|}(\frac{1}{2}, \frac{\sqrt{3}}{2})$. 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 $\mathfrak{L}$.

In this section, we describe the steps in our algorithm. In Section 3.4 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 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 1 computes a factor base for $\mathrm{Cl}(O_\Delta)$ consisting of all split primes up to $L_{|\Delta|}(\frac{1}{2}, z)$. The optimal value of the parameter z is determined in Section 3.4. 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 2 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 3.4.5). It is similar to Algorithm 11.2 in [BV07], except that we impose a constraint on $|\mathbf{v}|_\infty$ in line 1 in order

Algorithm 1 Computing a factor base

Input: An imaginary quadratic discriminant $\Delta < 0$ and a parameter z
Output: A factor base $\mathcal{F}$, or `nil`

Set $L \leftarrow \lceil L_{|\Delta|}(\frac{1}{2}, z) \rceil$, $k \leftarrow \lceil \ln L \rceil$, $\mathcal{F} \leftarrow \emptyset$
for all primes $p < L$ **do**
  if `kronecker`$(\Delta, p) = 1$ **then**
    $i \leftarrow 0$
    repeat
      $i \leftarrow i + 1$
      $g \leftarrow$ `primeForm`(Δ, 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 $\mathcal{F}$

to keep $|\mathbf{z}|_\infty$ small, and (for performance reasons) we use Bernstein's algorithm instead of trial division to find smooth elements.

We remark that Corollary 9.3.12 of [BV07] together with the restriction $C > 1$ in Theorem 3.2.1 implies that there exists a value of t satisfying the inequality in Algorithm 2.

Computing $\phi(P)$. Algorithm 3 evaluates $\phi(P)$, where $\phi\colon E \to E'$ is the normalized isogeny isogeny corresponding to the kernel ideal $\mathfrak{L}$.

3.4 Running Time Analysis

Here we determine the theoretical running time of Algorithm 3, as well as the optimal value of the parameter z in Algorithm 1. As before, these two quantities depend on each other, and hence both are calculated simultaneously.

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

Proposition 3.4.1. *Algorithm 1 takes time $L(\frac{1}{2}, z)$ and succeeds with probability at least $1/4$.*

Algorithm 2 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 3.2.1/Corollary 3.2.2

Output: A relation vector $\mathbf{z} \in \mathbb{Z}^f$ such that $[\mathfrak{b}] = [\mathcal{F}^{\mathbf{z}}]$, or `nil`

1: Set $\mathcal{S} \leftarrow \emptyset$, $\mathcal{P} \leftarrow \{N(\mathfrak{p}) : \mathfrak{p} \in \mathcal{F}\}$
2: Set $\ell \leftarrow L_{|\Delta|}(\frac{1}{2}, \frac{1}{4z})$
3: **for** $i = 0$ to ℓ **do**
4: Select $\mathbf{v} \in \mathbb{Z}^f_{0..|\Delta|-1}$ uniformly at random subject to the condition that $|\mathbf{v}|_\infty = t$
5: Calculate the reduced ideal $\mathfrak{a}_{\mathbf{v}}$ in the ideal class $[\mathfrak{b}] \cdot [\mathcal{F}^{\mathbf{v}}]$
6: Set $\mathcal{S} \leftarrow \mathcal{S} \cup N(\mathfrak{a}_{\mathbf{v}})$
7: **end for**
8: Using Bernstein's algorithm [Ber], find a $\mathcal{P}$-smooth element $N(\mathfrak{a}_{\mathbf{v}}) \in \mathcal{S}$ (if there exists one), or else return `nil`
9: Find the prime factorization of the integer $N(\mathfrak{a}_{\mathbf{v}})$
10: Using Seysen's algorithm [Sey87, Thm. 3.1] on the prime factorization of $N(\mathfrak{a}_{\mathbf{v}})$, factor the ideal $\mathfrak{a}_{\mathbf{v}}$ over $\mathcal{F}$ to obtain $\mathfrak{a}_{\mathbf{v}} = \mathcal{F}^{\mathbf{a}}$ for some $\mathbf{a} \in \mathbb{Z}^f$
11: Return $\mathbf{z} = \mathbf{a} - \mathbf{v}$

Proof. Since Algorithm 1 is identical to Algorithm 11.1 in [BV07], the proposition follows from Lemmas 11.3.1 and 11.3.2 of [BV07]. □

Proposition 3.4.2. *The running time of Algorithm 2 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

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

as desired. □

Proposition 3.4.3. *Under GRH, the probability that a single iteration of the **for** loop of Algorithm 2 produces an $\mathcal{F}$-smooth ideal $\mathfrak{a}_{\mathbf{v}}$ is at least $L(\frac{1}{2}, -\frac{1}{4z})$.*

Algorithm 3 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 1, compute a factor base; discard any primes dividing qn to obtain a new factor base $\mathcal{F} = \{\mathfrak{p}_1, \mathfrak{p}_2, \ldots, \mathfrak{p}_f\}$
2: Using Algorithm 2 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 $(\phi_1, \ldots, \phi_s)$ 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 $\eta\colon E_c \xrightarrow{\sim} E'$ 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)

Proof. We adopt the notation used in Theorem 3.2.1 and Corollary 3.2.2. Apply Corollary 3.2.2 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 2 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 3.2.2, 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 3.4.4. *Under GRH, the probability that Algorithm 2 succeeds is at least* $1 - \frac{1}{e}$.

Proof. Algorithm 2 loops through $\ell = L(\frac{1}{2}, \frac{1}{4z})$ vectors $\mathbf{v}$, and by Proposition 3.4.3, 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 2 is guaranteed to have small coefficients.

Proposition 3.4.5. *Any vector* $\mathbf{z}$ *output by Algorithm 2 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 3.

Theorem 3.4.6. *Under GRH, Algorithm 3 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 1 has running time $L(\frac{1}{2}, z)$ and success probability at least $1/4$, and Algorithm 2 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 3 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 [DF10, 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 $O(\ell^{2+\varepsilon}(\ln q)^{1+\varepsilon})$ [IJ10, 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 3.4.5, 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 3 is bounded by that of line 3, the theorem follows. □

Corollary 3.4.7. *Under GRH, Algorithm 3 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 3.4.6 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 $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2})$. □

3.5 A Quantum Algorithm For Constructing Isogenies

We now move on to the quantum approach to solving the problem of finding and evaluating the isogeny between two given ordinary elliptic curves. Note that this problem is harder than the previous one that we looked at. In fact, it is believed to be exponential for a classical computer. However, quantum computers are able to solve more classes of problems and we take advantage of that. Since evaluating the isogeny can be done subexponentially, we are left to show that finding the isogeny itself can also be done subexponentially, using a quantum computer.

Our quantum algorithm for constructing isogenies uses a simple reduction to the abelian hidden shift problem. This problem is defined as follows. Let A be a known finite abelian group (with the group operation written multiplicatively) and let $f_0, f_1 \colon A \to S$ be black-box functions, where S is a known finite set. We say that f_0, f_1 *hide* a shift $s \in A$ if f_0 is injective and $f_1(x) = f_0(xs)$ (i.e., f_1 is a shifted version of f_0). The goal of the

hidden shift problem is to determine s using queries to such black-box functions. Note that this problem is equivalent to the hidden subgroup problem in the A-dihedral group, the nonabelian group $A \rtimes \mathbb{Z}_2$ where $\mathbb{Z}_2$ acts on A by inversion.

Isogeny construction is easily reduced to the hidden shift problem using the group action defined in Section 2.5. Given two isogenous curves E_0, E_1 with endomorphism ring $\mathcal{O}_\Delta$, we define functions $f_0, f_1 \colon \mathrm{Cl}(\mathcal{O}_\Delta) \to \mathrm{Ell}_{q,n}(\mathcal{O}_\Delta)$ that hide $[\mathfrak{s}] \in \mathrm{Cl}(\mathcal{O}_\Delta)$, where $[\mathfrak{s}]$ is the ideal class such that $[\mathfrak{s}] * j(E_0) = j(E_1)$. Specifically, let $f_i([\mathfrak{b}]) = [\mathfrak{b}] * j(E_i)$. Then, since $*$ is a free and transitive group action [Wat69, Thm. 4.5], f_0 and f_1 hide $[\mathfrak{s}]$:

Lemma 3.5.1. *The function f_0 is injective and $f_1([\mathfrak{b}]) = f_0([\mathfrak{b}][\mathfrak{s}])$.*

Proof. Since $*$ is a group action,

$$\begin{aligned} f_1([\mathfrak{b}]) &= [\mathfrak{b}] * j(E_1) \\ &= [\mathfrak{b}] * ([\mathfrak{s}] * j(E_0)) \\ &= ([\mathfrak{b}][\mathfrak{s}]) * j(E_0) \\ &= f_0([\mathfrak{b}][\mathfrak{s}]). \end{aligned}$$

If there are distinct ideal classes $[\mathfrak{b}], [\mathfrak{b}']$ such that $f_0([\mathfrak{b}]) = f_0([\mathfrak{b}'])$, then $[\mathfrak{b}] * j(E_0) = [\mathfrak{b}'] * j(E_0)$, which contradicts the fact that the action is free and transitive [Wat69, Thm. 4.5]. Thus f_0 is injective. □

Note that a similar connection between isogenies and hidden shift problems was described in [Sto10, Section 7.2]. However, that paper did not mention the injectivity of the hiding functions in the context of the reduction. Without the assumption that f_0 is injective, the hidden shift problem can be as hard as the search problem, and hence requires exponentially many queries [BBBV97] (although for non-injective functions f_0 with appropriate structure, such as the Legendre symbol, the non-injective hidden shift problem can be solved by a quantum computer in polynomial time [DHI02]). On the other hand, injectivity implies that the problem has polynomial quantum query complexity [EH00], allowing for the possibility of faster quantum algorithms.

This reduction allows us to apply quantum algorithms for the hidden shift problem to construct isogenies. The (injective) hidden shift problem can be solved in quantum subexponential time assuming we can evaluate the group action in subexponential time. The latter is possible due to Algorithm 3.

We consider two different approaches to solving the hidden shift problem in subexponential time on a quantum computer. The first, due to Kuperberg [Kup05], has a faster

running time but requires superpolynomial space. The second approach generalizes an algorithm of Regev [Reg]. It uses only polynomial space, but is slower than Kuperberg's original algorithm.

Method 1: Kuperberg's algorithm. Kuperberg's approach to the abelian hidden shift problem is based on the idea of performing a Clebsch-Gordan sieve on coset states.

Theorem 3.5.2 ([Kup05])**.** *The abelian hidden shift problem has a [quantum] algorithm with time and query complexity $2^{O(\sqrt{n})}$, where n is the length of the output, uniformly for all finitely generated abelian groups.*

In our context, we have $2^{O(\sqrt{n})} = 2^{O(\sqrt{\ln|\Delta|})}$ since $|\mathrm{Cl}(\mathcal{O}_\Delta)| = O(\sqrt{\Delta}\ln\Delta)$ [BV07, Theorem 9.3.11]. Furthermore, $2^{O(\sqrt{\ln|\Delta|})} = L(o(1)) = L(0)$ regardless of the value of the implied constant in the exponent, since the exponent on the left has no $\sqrt{\ln\ln|\Delta|}$ term, whereas $L(0)$ does. As mentioned above, Kuperberg's algorithm also requires superpolynomial space (specifically, it uses $2^{O(\sqrt{n})}$ qubits).

Method 2: Regev's algorithm. Regev [Reg] showed that a variant of Kuperberg's sieve leads to a slightly slower algorithm using only polynomial space. In particular, he proved Theorem 3.5.3 below in the case where A is a cyclic group whose order is a power of 2 (without giving an explicit value for the constant in the exponent). Theorem 3.5.3 generalizes Regev's algorithm to arbitrary finite abelian groups.

Theorem 3.5.3. *Let A be a finite abelian group and let functions f_0, f_1 hide some unknown $s \in A$. Then there is a quantum algorithm that finds s with time and query complexity $L_{|A|}(\frac{1}{2}, \sqrt{2})$ using space* $\mathrm{poly}(\log|A|)$.

We now return to the original problem of constructing isogenies. Note that to use the hidden shift approach, the group structure of $\mathrm{Cl}(\mathcal{O}_\Delta)$ must be known. Given Δ, it is straightforward to compute $\mathrm{Cl}(\mathcal{O}_\Delta)$ using existing quantum algorithms (see the proof of Theorem 3.5.5). Thus, we assume for simplicity that the discriminant Δ is given as part of the input. This requirement poses no difficulty, since all existing proposals for isogeny-based public-key cryptosystems [Cou06, RS06, Sto10] stipulate that $\mathcal{O}_\Delta$ is a maximal order, in which case its discriminant can be computed easily: simply calculate the trace $t(E)$ of the curve using Schoof's algorithm [Sch95], and factor $t(E)^2 - 4q$ to obtain the fundamental discriminant Δ (note of course that factoring is easy on a quantum computer [Sho97]).

Remark 3.5.4. One can conceivably imagine a situation where one is asked to construct an isogeny between two given isogenous curves of unknown but identical endomorphism ring.

Algorithm 4 Isogeny construction

Input: A finite field $\mathbb{F}_q$, a discriminant $\Delta < 0$, and Weierstrass equations of isogenous elliptic curves E_0, E_1 with endomorphism ring $\mathcal{O}_\Delta$

Output: $[\mathfrak{s}] \in \mathrm{Cl}(\mathcal{O}_\Delta)$ such that $[\mathfrak{s}] * j(E_0) = j(E_1)$

1: Decompose $\mathrm{Cl}(\mathcal{O}_\Delta) = \langle [\mathfrak{b}_1] \rangle \oplus \cdots \oplus \langle [\mathfrak{b}_k] \rangle$ where $|\langle [\mathfrak{b}_j] \rangle| = n_j$

2: Solve the hidden shift problem defined by functions $f_0, f_1 \colon \mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_k} \to \mathrm{Ell}_{q,n}(\mathcal{O}_\Delta)$ satisfying $f_c(x_1, \ldots, x_k) = ([\mathfrak{b}_1]^{x_1} \cdots [\mathfrak{b}_k]^{x_k}) * j(E_c)$, giving some $(s_1, \ldots, s_k) \in \mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_k}$

3: Output $[\mathfrak{s}] = [\mathfrak{b}_1]^{s_1} \cdots [\mathfrak{b}_k]^{s_k}$

Although we are not aware of any cryptographic applications of this scenario, it presents no essential difficulty. Bisson has shown using Theorem 3.2.1 (see [Bis11, Thm. 6.1]) that the discriminant Δ of any ordinary elliptic curve can be computed in $L_q(\frac{1}{2}, \frac{1}{\sqrt{2}})$ time under only GRH (assuming that factoring is easy, which is the case for quantum computers [Sho97]).

Assuming Δ is known, we decompose $\mathrm{Cl}(\mathcal{O}_\Delta)$ as a direct sum of cyclic groups, with a known generator for each, and then solve the hidden shift problem. The overall procedure is described in Algorithm 4.

Theorem 3.5.5. *Assuming GRH, the running time of Algorithm 4 is $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2})$ (respectively, $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2} + \sqrt{2})$) using Theorem 3.5.2 (respectively, Theorem 3.5.3) to solve the hidden shift problem.*

Proof. We perform Step 1 using [CM01, Algorithm 10], which determines the structure of an abelian group given a generating set and a unique representation for the group elements. We represent the elements uniquely using reduced quadratic forms, and we use the fact that, under ERH (and hence GRH), the set of ideal classes of norm at most $12 \ln^2 |\Delta|$ forms a generating set [Bac90, Thm. 4]. Note that the result in [Bac90, Thm. 4] applies to non-maximal as well as maximal orders—take $\mathfrak{f}$ in the statement of that theorem to be the conductor of the non-maximal order. By Theorem 3.5.2 (resp. Theorem 3.5.3), Step 2 uses $2^{O(\sqrt{\ln |\Delta|})} = L(o(1)) = L(0)$ (resp. $L(\sqrt{2})$) evaluations of the functions f_i. By Corollary 3.4.7, these functions can be evaluated in time $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2})$ using Algorithm 3, assuming GRH. Overall, Step 2 takes time $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2} + o(1)) = L_q(\frac{1}{2}, \frac{\sqrt{3}}{2})$ using Theorem 3.5.2, or $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2} + \sqrt{2})$ using Theorem 3.5.3. The cost of Step 3 is negligible. □

Remark 3.5.6. GRH is a natural assumption. It seems necessary. Without it, the best known bounds on the factor base size are exponential.

Remark 3.5.7. The running time of the algorithm is ultimately limited by two factors: the best known quantum algorithm for the hidden shift problem runs in superpolynomial time,

and the same holds for the best known (classical or quantum) algorithm for computing the complex multiplication operator. Improving only one of these results to take polynomial time would still result in a superpolynomial-time algorithm.

Chapter 4

Isogeny-Based Quantum-Resistant Key Exchange and Encryption

4.1 Introduction

As part of the background material necessary in order to explain our contributions from Chapters 5 and 6 of this thesis, we describe in this chapter the isogeny-based cryptosystems of De Feo and Jao [JDF11] and De Feo et al. [DFJP14]. Portions of these publications were used in this chapter with permission.

The Diffie-Hellman scheme is a fundamental protocol for public-key exchange between two parties. Its original definition over finite fields is based on the hardness of computing the map $g, g^a, g^b \mapsto g^{ab}$ for $g \in \mathbb{F}_p^*$. As already discussed in previous chapters, Stolbunov [Sto10] proposed a Diffie-Hellman type system based on the difficulty of computing isogenies between ordinary elliptic curves, with the stated aim of obtaining quantum-resistant cryptographic protocols. The fastest known (classical) probabilistic algorithm for solving this problem is the algorithm of Galbraith and Stolbunov [GS11], based on the algorithm of Galbraith, Hess, and Smart [GHS02]. This algorithm is exponential, with a worst-case running time of $O(\sqrt[4]{q})$. However, as we have shown in previous chapter (and our paper [CJS14]), the private keys in Stolbunov's system can be recovered in subexponential time. Moreover, even if we only use classical attacks in assessing security levels, Stolbunov's scheme requires 229 seconds (even with precomputation) to perform one key exchange operation at the 128-bit security level on a desktop PC [Sto10, Table 1].

In this chapter, we will look at isogeny-based key-exchange, encryption, and identification schemes proposed by De Feo and Jao in [JDF11]. Their primitive achieves performance on the order of 60 milliseconds at the 128-bit security level (as measured against the fastest known quantum attacks) using desktop PCs, making the schemes far faster

than Stolbunov's. In terms of security, their schemes are not vulnerable to our algorithm presented in Chapter 3, nor to any algorithm of this type, since they are not based on a group action. The fastest known attacks against those schemes, even on quantum computers, require fully exponential time. The schemes involve new computational assumptions upon which their quantum resistance is based, and like all new computational assumptions, further study and the passage of time is needed for validation. Nevertheless, we believe the proposal represents a promising candidate for quantum-resistant isogeny-based public-key cryptography. We also use those computational assumptions for developing further schemes.

The scheme, presented in Section 4.2, uses isogenies between *supersingular* elliptic curves rather than ordinary elliptic curves. The main technical difficulty is that, in the supersingular case, the endomorphism ring is noncommutative, whereas Diffie-Hellman type protocols require commutativity. In Sections 4.3 and 4.4 we will see formal statements of the hardness assumptions and security reductions for the system.

4.1.1 Ramanujan Graphs

Let $G = (\mathcal{V}, \mathcal{E})$ be a finite graph on h vertices $\mathcal{V}$ with undirected edges $\mathcal{E}$. Suppose G is a regular graph of degree k, i.e., exactly k edges meet at each vertex. Given a labeling of the vertices $\mathcal{V} = \{v_1, \ldots, v_h\}$, the adjacency matrix of G is the symmetric $h \times h$ matrix A whose ij-th entry $A_{i,j} = 1$ if an edge exists between v_i and v_j and 0 otherwise.

It is convenient to identify functions on $\mathcal{V}$ with vectors in $\mathbb{R}^h$ via this labeling, and therefore also think of A as a self-adjoint operator on $L^2(\mathcal{V})$. All of the eigenvalues of A satisfy the bound $|\lambda| \leq k$. Constant vectors are eigenfunctions of A with eigenvalue k, which for obvious reasons is called the trivial eigenvalue λ_{triv}. A family of such graphs G with $h \to \infty$ is said to be a sequence of *expander graphs* if all other eigenvalues of their adjacency matrices are bounded away from $\lambda_{\text{triv}} = k$ by a fixed amount.[1] In particular, no other eigenvalue is equal to k; this implies the graph is connected. A Ramanujan graph is a special type of expander which has $|\lambda| \leq 2\sqrt{k-1}$ for any nontrivial eigenvalue which is not equal to $-k$ (this last possibility happens if and only if the graph is bipartite). The Ramanujan property was first defined in [LPS88]. It characterizes the optimal separation between the two largest eigenvalues of the graph adjacency matrix, and implies the expansion property.

A fundamental use of expanders is to prove the rapid mixing of the random walk on $\mathcal{V}$ along the edges $\mathcal{E}$. The following rapid mixing result is standard but we present it below for completeness. For the proof, see [JMV09] or [DSV03, Lub94, Sar90].

[1] Expansion is usually phrased in terms of the number of neighbors of subsets of G, but the spectral condition here is equivalent for k-regular graphs and also more useful for our purposes.

Proposition 4.1.1. *Let G be a regular graph of degree k on h vertices. Suppose that the eigenvalue λ of any nonconstant eigenvector satisfies the bound $|\lambda| \leq c$ for some $c < k$. Let S be any subset of the vertices of G, and x be any vertex in G. Then a random walk of length at least $\frac{\log 2h/|S|^{1/2}}{\log k/c}$ starting from x will land in S with probability at least $\frac{|S|}{2h} = \frac{|S|}{2|G|}$.*

4.1.2 Isogeny Graphs

An isogeny graph is a graph whose nodes consist of all elliptic curves in $\mathbb{F}_q$ belonging to a fixed isogeny class, up to $\bar{\mathbb{F}}_q$-isomorphism (so that two elliptic curves which are isomorphic over $\bar{\mathbb{F}}_q$ represent the same node in the graph). In practice, the nodes are represented using j-invariants, which are invariant up to isomorphism. Isogeny graphs for supersingular elliptic curves were first considered by Mestre [Mes86], and were shown by Pizer [Piz90, Piz98] to have the Ramanujan property.

Every supersingular elliptic curve in characteristic p is defined over either $\mathbb{F}_p$ or $\mathbb{F}_{p^2}$ [Sil92], so it suffices to fix $\mathbb{F}_q = \mathbb{F}_{p^2}$ as the field of definition for this discussion. Thus, in contrast to ordinary curves, there are a finite number of isomorphism classes of supersingular curves in any given isogeny class; this number is in fact $g + 1$, where g is the genus of the modular curve $X_0(p)$, which is roughly $p/12$. It turns out that all supersingular curves defined over $\bar{\mathbb{F}}_p$ belong to the same isogeny class [Mes86]. For a fixed prime value of $\ell \neq p$, we define the vertices of the supersingular isogeny graph $\mathcal{G}$ to consist of these $g + 1$ isomorphism classes of curves, with edges given by isomorphism classes of degree-ℓ isogenies, defined as follows: two isogenies $\phi_1, \phi_2 \colon E_i \to E_j$ are isomorphic if there exists an automorphism $\alpha \in \mathrm{Aut}(E_j)$ (i.e., an invertible endomorphism) such that $\phi_2 = \alpha\phi_1$. Pizer [Piz90, Piz98] has shown that $\mathcal{G}$ is a connected $k = \ell + 1$-regular multigraph satisfying the Ramanujan bound of $|\lambda| \leq 2\sqrt{\ell} = 2\sqrt{k-1}$ for the nontrivial eigenvalues of its adjacency matrix.

4.2 Public-Key Cryptosystems Based On Supersingular Curves

In this section we present a key-exchange protocol and a public-key cryptosystem analogous to those of [RS06, Sto10], and a zero-knowledge identification scheme, all using supersingular elliptic curves.

The protocols require supersingular curves of smooth order. Such curves are normally unsuitable for cryptography since discrete logarithms on them are easy. However, since the discrete logarithm problem is unimportant in our setting, this issue does not affect us. In the supersingular setting, it is easy to construct curves of smooth order, and using a smooth

order curve will give a large number of isogenies that are fast to compute. Specifically, we fix $\mathbb{F}_q = \mathbb{F}_{p^2}$ as the field of definition, where p is a prime of the form $\ell_A^{e_A}\ell_B^{e_B} \cdot f \pm 1$. Here ℓ_A and ℓ_B are small primes, and f is a cofactor such that p is prime. Then we construct a supersingular curve E defined over $\mathbb{F}_q$ of cardinality $(\ell_A^{e_A}\ell_B^{e_B} f)^2$. By construction, $E[\ell_A^{e_A}]$ is $\mathbb{F}_q$-rational and contains $\ell_A^{e_A-1}(\ell_A + 1)$ cyclic subgroups of order $\ell_A^{e_A}$, each defining a different isogeny; the analogous statement holds for $E[\ell_B^{e_B}]$.

The protocols revolve around the following commutative diagram

$$\begin{array}{ccc} E & \xrightarrow{\phi} & E/\langle P\rangle \\ {\scriptstyle\psi}\downarrow & & \downarrow \\ E/\langle Q\rangle & \to & E/\langle P, Q\rangle \end{array} \tag{4.1}$$

where ϕ and ψ are random walks in the graphs of isogenies of degrees ℓ_A and ℓ_B respectively. Their security is based on the difficulty of finding a path connecting two given vertices in a graph of supersingular isogenies.

4.2.1 Zero-Knowledge Proof of Identity

We begin with the protocol which is easiest to understand. Peggy knows a cyclic degree $\ell_A^{e_A}$ isogeny $\phi : E \to E/\langle S\rangle$, with the curves E and $E/\langle S\rangle$ publicly known, and wants to prove to Vic that she knows a generator for $\langle S\rangle$, without revealing it.

The protocol is loosely inspired by the zero-knowledge proof of membership for Graph Isomorphism [GMW91]. In that protocol, Peggy shows that she knows a graph isomorphism $G \simeq G'$ by first publishing a random H such that the following diagram commutes

$$\begin{array}{ccc} G & \longleftrightarrow & G' \\ & {\scriptstyle\phi}\searrow \quad \swarrow{\scriptstyle\psi} & \\ & H & \end{array} \tag{4.2}$$

and then revealing only one among ϕ and ψ. Intuitively, this protocol is *perfectly* zero-knowledge because the information that Peggy reveals (i.e., a random permutation of G or G') could be easily computed by anyone without her help.

In an analogous way, the protocol consists in publishing the vertices of diagram (4.1), and then revealing some, but not all, of its arrows. Unlike the case of Graph Isomorphism, in the protocol Peggy needs to use her secret knowledge to create the diagram, thus we cannot achieve a *perfect* zero-knowledge. Nevertheless, we will show in Section 4.4 that, under suitable assumptions, the protocol is *computationally* zero-knowledge.

We show below the diagram used in the protocol. $\langle S \rangle$ is the kernel of the secret isogeny ϕ of degree $\ell_A^{e_A}$, while $\langle R \rangle$ is a cyclic group of order $\ell_B^{e_B}$.

$$\begin{array}{ccc} E & \xrightarrow{\phi} & E/\langle S \rangle \\ {\scriptstyle\psi}\downarrow & & \downarrow{\scriptstyle\psi'} \\ E/\langle R \rangle & \xrightarrow{\phi'} & E/\langle S, R \rangle \end{array} \tag{4.3}$$

Peggy can compute the diagram as follows:

- She uses Vélu's formulas to compute the isogeny $\psi : E \to E/\langle R \rangle$;
- She computes $R' = \phi(R)$ and the isogeny $\psi' : E/\langle S \rangle \to E/\langle S, R \rangle$;
- She computes $S' = \psi(S)$ and the isogeny $\phi' : E/\langle R \rangle \to E/\langle S, R \rangle$.

Now, the natural question is: which arrows of the diagram can Peggy reveal without compromising her secret ϕ? It is not hard to see, and we will show it in Theorem 4.4.3, that the knowledge of (ψ, ϕ') or (ψ', ϕ') allows anyone to compute the kernel of ϕ. However, we will argue that there is no obvious way to compute ϕ from the sole knowledge of ϕ'. Revealing one of ψ or ψ' is no problem either, however revealing (ψ, ψ') altogether is more subtle. Indeed, revealing the points R and $\phi(R)$ uncovers some information on the action of ϕ on $E[\ell_B^{e_B}]$: it is to be expected that after a few iterations Peggy will reveal a basis (P, Q) of $E[\ell_B^{e_B}]$ and the respective images $\phi(P)$, $\phi(Q)$, thus allowing anyone to evaluate ϕ on the whole $E[\ell_B^{e_B}]$. Nevertheless, we conjecture that this leakage does not compromise Peggy's secret either, and we make these data part of the public parameters.[2]

Finally, we present the protocol.

Secret parameters A supersingular curve E defined over $\mathbb{F}_q$ and a primitive $\ell_A^{e_A}$-torsion point S defining an isogeny $\phi : E \to E/\langle S \rangle$.

Public parameters The curves E and $E/\langle S \rangle$. Generators P, Q of $E[\ell_B^{e_B}]$ and their images $\phi(P), \phi(Q)$.

Identification Repeat m times:

1. Peggy chooses a random point R of order $\ell_B^{e_B}$ and computes diagram (4.3).

[2] An alternative solution, that intuitively leaks less information on ϕ, would be to publish random generators of $\langle R \rangle$ and $\langle \phi(R) \rangle$. However, it is not clear that this idea would considerably improve the security of the protocol, and we will not pursue it further for coherence with the protocols that will follow.

2. Peggy sends the curves $E_1 = E/\langle R \rangle$ and $E_2 = E/\langle S, R \rangle$ to Vic.
3. Vic selects a random bit b and sends it to Peggy.
4. If $b = 0$, Peggy reveals the points R and $R' = \phi(R)$. Vic accepts if they have order $\ell_B^{e_B}$ and generate the kernels of isogenies $E \to E_1$ and $E/\langle S \rangle \to E_2$, respectively.
5. If $b = 1$, Peggy reveals the point $\psi(S)$. Vic accepts if it has order $\ell_A^{e_A}$ and generates the kernel of an isogeny $E_1 \to E_2$.

4.2.2 Key Exchange

The key exchange protocol is a variation *à la* Diffie-Hellman over diagram (4.1). The idea is to let Alice choose ϕ, while Bob chooses ψ. Although similar in spirit to the protocol based on the action of the class group on ordinary elliptic curves of [Sto10], a main technical difference is that, since ideal classes no longer commute (or indeed even multiply together) in the supersingular case, extra information must be communicated as part of the protocol in order to ensure that both parties arrive at the same common value.

We fix as public parameters a supersingular curve E_0 defined over $\mathbb{F}_{p^2}$, and bases $\{P_A, Q_A\}$ and $\{P_B, Q_B\}$ which generate $E_0[\ell_A^{e_A}]$ and $E_0[\ell_B^{e_B}]$ respectively, so that $\langle P_A, Q_A \rangle = E_0[\ell_A^{e_A}]$ and $\langle P_B, Q_B \rangle = E_0[\ell_B^{e_B}]$. Note that all torsion groups are always defined over $\mathbb{F}_{p^2}$, hence this requires picking a curve having the correct order over $\mathbb{F}_{p^2}$. Alice chooses two random elements $m_A, n_A \in_R \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$, not both divisible by ℓ_A, and computes an isogeny $\phi_A \colon E_0 \to E_A$ with kernel $K_A := \langle [m_A]P_A + [n_A]Q_A \rangle$. Alice also computes the image $\{\phi_A(P_B), \phi_A(Q_B)\} \subset E_A$ of the basis $\{P_B, Q_B\}$ for $E_0[\ell_B^{e_B}]$ under her secret isogeny ϕ_A, and sends these points to Bob together with E_A. Similarly, Bob selects random elements $m_B, n_B \in_R \mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$ and computes an isogeny $\phi_B \colon E_0 \to E_B$ having kernel $K_B := \langle [m_B]P_B + [n_B]Q_B \rangle$, along with the points $\{\phi_B(P_A), \phi_B(Q_A)\}$. Upon receipt of E_B and $\phi_B(P_A), \phi_B(Q_A) \in E_B$ from Bob, Alice computes an isogeny $\phi'_A \colon E_B \to E_{AB}$ having kernel equal to $\langle [m_A]\phi_B(P_A) + [n_A]\phi_B(Q_A) \rangle$; Bob proceeds *mutatis mutandis*. Alice and Bob can then use the common j-invariant of

$$E_{AB} = \phi'_B(\phi_A(E_0)) = \phi'_A(\phi_B(E_0)) = E_0/_{\langle [m_A]P_A + [n_A]Q_A, [m_B]P_B + [n_B]Q_B \rangle}$$

to form a secret shared key.

The full protocol is given in Figure 4.1. We denote by A and B the identifiers of Alice and Bob, and use sID to denote the unique session identifier.

$\mathcal{A}$		$\mathcal{B}$
Input: A, B, sID		**Input:** B
$m_A, n_A \in_R \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$		$m_B, n_B \in_R \mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$
$\phi_A := E_0/\langle [m_A]P_A + [n_A]Q_A \rangle$		$\phi_B := E_0/\langle [m_B]P_B + [n_B]Q_B \rangle$
	$A, \text{sID}, \phi_A(P_B), \phi_A(Q_B), E_A \longrightarrow$	
	$\leftarrow B, \text{sID}, \phi_B(P_A), \phi_B(Q_A), E_B$	
$E_{AB} :=$		$E_{BA} :=$
$E_B/\langle [m_A]\phi_B(P_A) + [n_A]\phi_B(Q_A) \rangle$		$E_A/\langle [m_B]\phi_A(P_B) + [n_B]\phi_A(Q_B) \rangle$
Output: $j(E_{AB}), \text{sID}$		**Output:** $j(E_{BA}), \text{sID}$

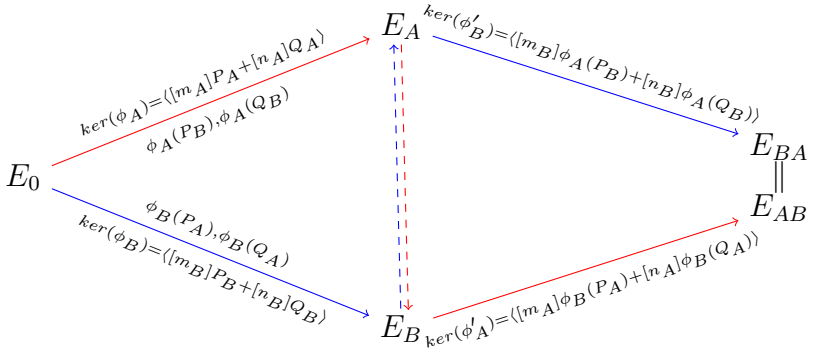

Figure 4.1: Key-exchange protocol using isogenies on supersingular curves.

4.2.3 Public-Key Encryption

The key-exchange protocol of Section 4.2.2 can easily be adapted to yield a public-key cryptosystem, in much the same way that Elgamal encryption follows from Diffie-Hellman. We briefly give the details here. All notation is the same as above. Stolbunov [Sto10] uses a similar construction, upon which this one is based.

Setup Choose $p = \ell_A^{e_A}\ell_B^{e_B} \cdot f \pm 1$, E_0, $\{P_A, Q_A\}$, $\{P_B, Q_B\}$ as above. Let $\mathcal{H} = \{H_k : k \in K\}$ be a hash function family indexed by a finite set K, where each H_k is a function from $\mathbb{F}_{p^2}$ to the message space $\{0,1\}^w$.

Key generation Choose two random elements $m_A, n_A \in_R \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$, not both divisible by ℓ_A. Compute $E_A, \phi_A(P_B), \phi_A(Q_B)$ as above, and choose a random element $k \in_R$

K. The public key is the tuple $(E_A, \phi_A(P_B), \phi_A(Q_B), k)$ and the private key is (m_A, n_A, k).

Encryption Given a public key $(E_A, \phi_A(P_B), \phi_A(Q_B), k)$ and a message $m \in \{0,1\}^w$, choose two random elements $m_B, n_B \in_R \mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$, not both divisible by ℓ_B, and compute

$$h = H_k(j(E_{AB})),$$
$$c = h \oplus m.$$

The ciphertext is $(E_B, \phi_B(P_A), \phi_B(Q_A), c)$.

Decryption Given a ciphertext $(E_B, \phi_B(P_A), \phi_B(Q_A), c)$ and a private key (m_A, n_A, k), compute the j-invariant $j(E_{AB})$ and set

$$h = H_k(j(E_{AB})),$$
$$m = h \oplus c.$$

The plaintext is m.

For detail on the algorithmic aspects of the schemes, please refer to the original paper.

4.3 Complexity Assumptions

As before, let p be a prime of the form $\ell_A^{e_A}\ell_B^{e_B} \cdot f \pm 1$, and fix a supersingular curve E_0 over $\mathbb{F}_{p^2}$ together with bases $\{P_A, Q_A\}$ and $\{P_B, Q_B\}$ of $E_0[\ell_A^{e_A}]$ and $E_0[\ell_B^{e_B}]$ respectively. In analogy with the case of isogenies over ordinary elliptic curves, we define the following computational problems, adapted for the supersingular case:

Problem 4.3.1 (Decisional Supersingular Isogeny (DSSI) problem). Let E_A be another supersingular curve defined over $\mathbb{F}_{p^2}$. Decide whether E_A is $\ell_A^{e_A}$-isogenous to E_0.

Problem 4.3.2 (Computational Supersingular Isogeny (CSSI) problem). Fix an isogeny $\phi_A \colon E_0 \to E_A$ whose kernel is $\langle [m_A]P_A + [n_A]Q_A \rangle$, where m_A and n_A are chosen at random from $\mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ and not both divisible by ℓ_A. Given E_A and the values $\phi_A(P_B)$, $\phi_A(Q_B)$, find a generator R_A of $\langle [m_A]P_A + [n_A]Q_A \rangle$.

We remark that given a generator $R_A = [m_A]P_A + [n_A]Q_A$, it is easy to solve for (m_A, n_A), since E_0 has smooth order and thus extended discrete logarithms are easy in E_0 [Tes99].

Problem 4.3.3 (Supersingular Computational Diffie-Hellman (SSCDH) problem). Fix an isogeny $\phi_A \colon E_0 \to E_A$ whose kernel is equal to $\langle [m_A]P_A + [n_A]Q_A \rangle$, and let $\phi_B \colon E_0 \to E_B$ be an isogeny whose kernel is $\langle [m_B]P_B + [n_B]Q_B \rangle$, where m_A, n_A (respectively m_B, n_B) are chosen at random from $\mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ (respectively $\mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$) and not both divisible by ℓ_A (respectively ℓ_B). Given the curves E_A, E_B and the points $\phi_A(P_B)$, $\phi_A(Q_B)$, $\phi_B(P_A)$, $\phi_B(Q_A)$, find the j-invariant of $E_0/\langle [m_A]P_A + [n_A]Q_A, [m_B]P_B + [n_B]Q_B \rangle$.

Problem 4.3.4 (Supersingular Decisional Diffie-Hellman (SSDDH) problem). Given a tuple sampled with probability 1/2 from one of the following two distributions:

- $(E_A, \phi_A(P_B), \phi_A(Q_B), E_B, \phi_B(P_A), \phi_B(Q_A), E_{AB})$, where the quantities E_A, $\phi_A(P_B)$, $\phi_A(Q_B)$, E_B, $\phi_B(P_A)$, and $\phi_B(Q_A)$ are as in the SSCDH problem and
$$E_{AB} \cong E_0/\langle [m_A]P_A + [n_A]Q_A, [m_B]P_B + [n_B]Q_B \rangle,$$
- $(E_A, \phi_A(P_B), \phi_A(Q_B), E_B, \phi_B(P_A), \phi_B(Q_A), E_C)$, wherein the quantities E_A, $\phi_A(P_B)$, $\phi_A(Q_B)$, E_B, $\phi_B(P_A)$, and $\phi_B(Q_A)$ are as in the SSCDH problem and
$$E_C \cong E_0/\langle [m'_A]P_A + [n'_A]Q_A, [m'_B]P_B + [n'_B]Q_B \rangle,$$
where m'_A, n'_A (respectively m'_B, n'_B) are chosen at random from $\mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ (respectively $\mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$) and not both divisible by ℓ_A (respectively ℓ_B),

determine from which distribution the tuple is sampled.

The ordinary case analogue of the following problem is trivially solvable in polynomial time. Its supposed difficulty in the supersingular case is at the heart of the security of this identification scheme.

Problem 4.3.5 (Decisional Supersingular Product (DSSP) problem). Given a degree $\ell_A^{e_A}$ isogeny $\phi : E_0 \to E_3$ and a tuple sampled with probability 1/2 from one of the following two distributions:

- (E_1, E_2, ϕ'), where the product $E_1 \times E_2$ is chosen at random among those $\ell_B^{e_B}$-isogenous to $E_0 \times E_3$, and where $\phi' : E_1 \to E_2$ is an isogeny of degree $\ell_A^{e_A}$, and
- (E_1, E_2, ϕ'), where E_1 is chosen at random among the curves having the same cardinality as E_0, and $\phi' : E_1 \to E_2$ is a random isogeny of degree $\ell_A^{e_A}$,

determine from which distribution the tuple is sampled.

We conjecture that these problems are computationally infeasible, in the sense that for any polynomial-time solver algorithm, the advantage of the algorithm is a negligible function of the security parameter $\log p$. The resulting security assumptions are referred to as the DSSI assumption, CSSI assumption, etc.

4.3.1 Hardness Of The Underlying Assumptions

In this section we discuss the feasibility of solving isogeny-based computational problems on a quantum computer. At a high level, the fastest known attacks against CSSI, DSSP involve claw-finding in the isogeny graph. Known existing lower bounds for quantum claw-finding provide some assurance that a faster attack will never be found since such an attack would require a non-generic solutions to the claw-finding problem. The lack of any related commutative group structure or periodic function in the isogeny graph indicates that analogues to Shor's algorithm are unlikely to apply to isogeny computation.

Given a CSSI (respectively, SSCDH) solver, it is trivial to solve SSCDH (respectively, SSDDH). It is also trivial to solve SSDDH given a DSSI solver. There are no known reductions in the other direction, and given that the corresponding question of equivalence for discrete logarithms and Diffie-Hellman has not yet been completely resolved in all cases, it is reasonable to assume that the question of equivalence of CSSI, SSCDH, and SSDDH is at least hard to resolve. For the purposes of this discussion, we will presume that DSSI and CSSI are equivalent to SSDDH. Concerning DSSP, there is an evident reduction to DSSI. However, it seems reasonable to assume that DSSP is easier than the latter.

In the context of cryptography, the problem of computing an isogeny between isogenous supersingular curves was first considered by Galbraith [Gal99] in 1999. The first published cryptographic primitive based on supersingular isogeny graphs is the hash function proposal of Charles et al. [CLG09], which remains unbroken to date (the cryptanalysis of [PLQ08] applies only to the LPS graph-based hash function from [CLG09], and not to the supersingular isogeny graph-based hash functions). The fastest known algorithm for finding isogenies between supersingular curves in general takes $O(\sqrt{p}\log^2 p)$ time [CLG09, §5.3.1]; however the presented problem is less general because the degree of the isogeny is known in advance and is smooth. In addition, the distribution of isogenous curves obtained from taking kernels of the form $\langle [m_A]P_A + [n_A]Q_A \rangle$ is not quite uniform: a simple calculation against Proposition 4.1.1 indicates that a sequence of e_A isogenies of degree ℓ_A falls short of the length needed to ensure uniform mixing, regardless of the value of p. Since our research group is the first to propose using isogenies of this type, there is no existing literature addressing the security of the isogenies of the special form proposed.

There are easy exponential attacks against DSSI and CSSI that improve upon exhaustive search. To find an isogeny of degree $\ell_A^{e_A}$ between E and E_A, an attacker builds two trees of all curves isogenous to E (respectively, E_A) via isogenies of degree $\ell_A^{e_A/2}$. Once the trees are built, the attacker tries to find a curve lying in both trees. Since the degree of the isogeny ϕ_A is $\sim \sqrt{p}$ (much shorter than the size of the isogeny graph), it is unlikely that there will be more than one isogeny path—and thus more than one match—from E to E_A. Given two functions $f : A \to C$ and $g : B \to C$ with domain of equal size, finding a pair (a, b) such that $f(a) = g(b)$ is known as the *claw problem* in complexity

theory. The claw problem can obviously be solved in $O(|A| + |B|)$ time and $O(|A|)$ space on a classical computer by building a hash table holding $f(a)$ for any $a \in A$ and looking for hits for $g(b)$ where $b \in B$. This gives a $O(\ell_A^{e_A/2}) = O(\sqrt[4]{p})$ classical attack against those cryptosystems. With a quantum computer, one can do better using the algorithm in [Tan08], which has complexity $O(\sqrt[3]{|A||B|})$, thus giving an $O(\ell_A^{e_A/3}) = O(\sqrt[6]{p})$ quantum attack against the presented cryptosystems. These complexities are optimal for a black-box claw attack [Zha05].

We consider the question of whether the auxiliary data points $\phi_A(P_B)$ and $\phi_A(Q_B)$ might assist an adversary in determining ϕ_A. Since (P_B, Q_B) forms a basis for $E_0[\ell_B^{e_B}]$, the values $\phi_A(P_B)$ and $\phi_A(Q_B)$ allow the adversary to compute ϕ_A on all of $E_0[\ell_B^{e_B}]$. This is because any element of $E_0[\ell_B^{e_B}]$ is a (known) linear combination of P_B and Q_B (known since extended discrete logarithms are easy [Tes99]). However, there does not appear to be any way to use this capability to determine ϕ_A. Even on a quantum computer, where finding abelian hidden subgroups is easy, there is no hidden subgroup to find, since ϕ_A has degree $\ell_A^{e_A}$, and thus does not annihilate any point in $E_0[\ell_B^{e_B}]$ other than the identity. Of course, if one could evaluate ϕ_A on arbitrary points of $E_0[\ell_A^{e_A}]$, then a quantum computer could easily break the scheme, and indeed in this case the scheme is also easily broken classically by using a few calls to the oracle to compute a generator of the kernel of the dual isogeny $\hat{\phi}_A$. However, it does not seem possible to translate the values of ϕ_A on $E_0[\ell_B^{e_B}]$ into values on $E_0[\ell_A^{e_A}]$.

Recall that, for both ordinary and supersingular curves, there is a natural bijection between isogenies (up to isomorphism) and (left) ideals in the endomorphism ring. In the ordinary case the endomorphism ring is commutative, and ideal classes form a finite abelian group. This property has been used by Childs et al. [CJS14] to solve the ordinary analogue of CSSI in quantum subexponential time. It is natural to ask whether their algorithm can be adapted to the supersingular setting. Here the endomorphism ring is a maximal order in a noncommutative quaternion algebra, and the left ideal classes do not form a group at all (though they do form a groupoid). Since the algorithm of Childs et al. depends crucially on the properties of abelian groups, we believe that no reasonable variant of this strategy would apply.

The same correspondence between isogenies and ideals can be applied to DSSP. Indeed, deciding DSSP amounts to deciding whether the ideals S, S' associated to ϕ, ϕ' are conjugated, i.e., whether there exists a left ideal $R \in \mathrm{End}(E_0)$ such that $S = RS'R^{-1}$. Although it can be hoped that deciding conjugacy of ideal classes in the quaternion algebra $\mathbb{Q}_{p,\infty}$ is feasible, we are still faced with the problem that the best known algorithms to compute the endomorphism rings of supersingular curves are exponential in $\log p$ [Koh96, Cn04, Bel08]. Hence, we deem DSSP secure given the current knowledge.

The fact that it is possible to obtain a zero-knowledge identification scheme from CSSI comes as no surprise, since it is well known that a zero-knowledge protocol can be ob-

tained from any problem in NP [GMW91]. Nevertheless, the generic construction is not very efficient, and many efforts have been made to obtain efficient *ad-hoc* schemes from NP-complete problems [Sha89, Ste94a, Ste94b, Poi95]. While the security of most of these schemes is based on two solid assumptions, namely that P $\neq$ NP and that *secure commitment schemes* exist, the presented identification scheme stands on a much weaker ground: the CSSI and DSSP problems. As performances go, it is reasonable to assume that the presented scheme will be some orders of magnitude slower than the best zero-knowledge protocols. We can thus conclude that presented scheme is of a purely theoretical and pedagogical interest. Yet it is remarkable that an efficient identification scheme based on graphs of supersingular isogenies simply exists, while the analogous construction for ordinary curves is trivially broken and no other identification scheme is currently known to work in that case [Sto10].

4.4 Security Results

In this section we state the security results for the schemes presented in this chapter. The statements of the theorems are as follows:

Theorem 4.4.1. *If the SSDDH assumption holds, then the key-agreement protocol of Section 4.2.2 is session-key secure in the authenticated-links adversarial model of Canetti and Krawczyk [CK01].*

Theorem 4.4.2. *If the SSDDH assumption holds, and the hash function family $\mathcal{H}$ is entropy-smoothing, then the public-key cryptosystem of Section 4.2.3 is IND-CPA.*

Theorem 4.4.3. *Under the CSSI and DSSP assumptions, the identification scheme of Section 4.2.1 is zero-knowledge.*

For proofs of the theorems, we refer to the original paper [JDF11].

Chapter 5

Isogeny-Based Quantum-Resistant Undeniable Signatures

5.1 Introduction

This chapter is based on [JS14], authored jointly with my supervisor David Jao.

Many current cryptographic schemes are based on mathematical problems that are considered difficult with classical computers, but can easily be solved using quantum algorithms. To prepare for the emergence of quantum computers, we aim to design cryptographic primitives for common operations such as encryption and authentication which resist quantum attacks. One family of such primitives, proposed by De Feo, Jao, and Plût [DFJP14, JDF11], which we described in the previous chapter, uses isogenies between supersingular elliptic curves to construct cryptographic protocols for public-key encryption, key exchange, and entity authentication which are believed to be quantum-resistant. To date, however, this protocol family lacks comprehensive techniques for achieving data authentication.

In this chapter, we present a new construction of quantum-resistant undeniable signatures based on the difficulty of computing isogenies between supersingular elliptic curves. Few such constructions are known, and indeed the only other proposed quantum-resistant undeniable signature scheme in the literature is the code-based scheme of Aguilar-Melchor et al. [AMBGS13]. Our scheme uses a completely different approach and is based on completely different assumptions, making it a useful alternative in the event that some breakthrough arises in the cryptanalysis of code-based systems.

Undeniable signatures provide tools for signer to prove that the signature is valid for a given message, if it was truly signed by him. If the signature is fake, the signer has the tools to prove that it is fake.

5.2 Quantum-Resistant Undeniable Signatures From Isogenies

In this section, we present a new construction of an undeniable signature scheme from isogenies. An undeniable signature can be verified by any party, but verification requires interaction with the signer. To distinguish between invalid (forged) signatures and valid signatures that the verifier refuses to verify, an undeniable signature scheme also includes a mechanism for the signer to prove (interactively) that an invalid signature is forged. Our construction uses a three-prime variant of the original two-prime protocol given in Section 4.2.2. As a consequence, the resulting commutative diagrams for zero-knowledge proofs become 3-dimensional rather than 2-dimensional.

5.2.1 Definition

An undeniable signature scheme [KF08] consists of a key generation algorithm, a signing algorithm, a validity check, a signature simulator, a confirmation protocol π_{con} and a disavowal protocol π_{dis}. The role of the confirmation protocol π_{con} is for the signer to prove to the verifier that the signature is valid. The role of the disavowal protocol π_{dis} is for a valid signer to be able to prove to the verifier that the signature that the verifier has received is not valid.

Unforgeability is defined using the following game between a challenger and an adversary A.

1. The challenger generates a key pair (vk, sk) randomly, and gives the verification key vk to A.
2. For $i = 1, 2, \ldots, q_s$ for some q_s, A queries the signing oracle adaptively with a message m_i and receives a signature σ_i.
3. Eventually, A outputs a forgery (m^*, σ^*).

We also allow the adversary A to submit pairs (m_j, σ_j) to the confirmation/disavowal oracle adaptively in step 2, where the confirmation/disavowal oracle responds as follows:

- If (m_j, σ_j) is a valid pair, then the oracle returns a bit $\mu = 1$ and proceeds with the execution of the confirmation protocol π_{con} with A.
- Otherwise, the oracle returns a bit $\mu = 0$ and proceeds with the execution of the disavowal protocol π_{dis} with A.

We say that A succeeds in producing a strong forgery if (m^*, σ^*) is valid and (m^*, σ^*) is not among the pairs (m_i, σ_i) generated during the signing queries. The signature scheme is *strongly unforgeable* if the probability that A succeeds in producing a strong forgery is negligible for any PPT adversary A in the above game.

Invisibility is defined using the following game between a challenger and an adversary A.

1. The challenger generates a key pair $(\mathrm{vk}, \mathrm{sk})$ randomly, and gives the verification key vk to A.
2. A is permitted to issue a series of signing queries m_i to the signing oracle adaptively and receive a signature σ_i.
3. At some point, A chooses a message m^* and sends it to the challenger.
4. The challenger chooses a random bit b. If $b = 1$, then he computes the real signature for m^* using sk and sets it to be σ^*. Otherwise he computes a fake signature m^* using vk and sets it to be σ^*. He sends σ^* to A.
5. A performs some signing queries again.
6. At the end of this game, A outputs a guess b'.

We allow the adversary A to submit pairs (m_j, σ_j) to the confirmation/disavowal oracle adaptively in step 2 and in step 5. However, A is not allowed to submit the challenge (m^*, σ^*) to the confirmation/disavowal oracle in step 5. Also, A is not allowed to submit m^* to the signing oracle. We say that the signature scheme is *invisible* if no PPT adversary A has non-negligible advantage in this game.

For an undeniable signature scheme to be secure, it must satisfy unforgeability and invisibility. In addition, the confirmation π_{con} and disavowal π_{dis} protocols must be complete, sound, and zero-knowledge.

5.2.2 Protocol

Let p be a prime of the form $\ell_A^{e_A}\ell_M^{e_M}\ell_C^{e_C} \cdot f \pm 1$, and fix a supersingular curve E over $\mathbb{F}_{p^2}$ together with bases $\{P_A, Q_A\}$, $\{P_M, Q_M\}$ and $\{P_C, Q_C\}$ of $E[\ell_A^{e_A}]$, $E[\ell_M^{e_M}]$ and $E[\ell_C^{e_C}]$ respectively. The design of the protocol is such that, generally speaking, points in $\langle P_A, Q_A\rangle$ are used for key material, points in $\langle P_M, Q_M\rangle$ are used for message data, and points in $\langle P_C, Q_C\rangle$ correspond to commitment data.

The signer generates two secret random integers $m_A, n_A \in \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$, obtains $K_A = [m_A]P_A + [n_A]Q_A$ and computes $E_A = E/\langle K_A\rangle$. Let ϕ_A be an isogeny from E to E_A.

Public parameters: $p, E, \{P_A, Q_A\}, \{P_M, Q_M\}, \{P_C, Q_C\}$, and a public hash function $H\colon \{0,1\}^* \to \mathbb{Z}$.

Public key: E_A, $\phi_A(P_C)$, $\phi_A(Q_C)$.

Private key: m_A, n_A.

To sign a message M, we compute the hash $h = H(M)$. Let $K_M = P_M + [h]Q_M$. Then the signer computes the isogenies

- $\phi_M\colon E \to E_M = E/\langle K_M \rangle$
- $\phi_{M,AM}\colon E_M \to E_{AM} = E_M/\langle \phi_M(K_A) \rangle$
- $\phi_{A,AM}\colon E_A \to E_{AM} = E_A/\langle \phi_A(K_M) \rangle$

along with the auxiliary points $\phi_{M,AM}(\phi_M(P_C))$ and $\phi_{M,AM}(\phi_M(Q_C))$. The signer then presents these two auxiliary points along with E_{AM} as the signature. (See Figure 5.1.)

The *confirmation protocol* proceeds as follows. We must confirm E_{AM} without revealing the isogenies used to produce it. We do so by "blinding" E_{AM} using ϕ_C and disclosing the blinded isogenies (see Figure 5.2).

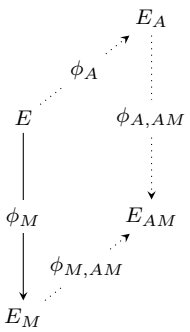

Figure 5.1: Signature generation.

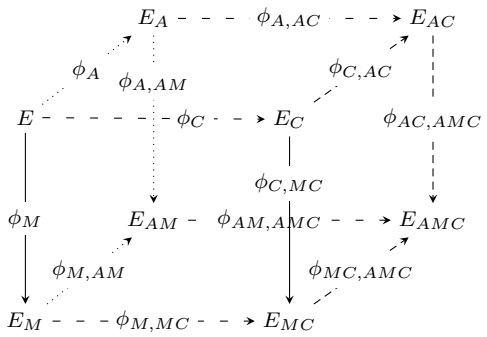

Figure 5.2: Confirmation protocol.

1. The signer secretly selects random integers $m_C, n_C \in \mathbb{Z}/\ell_C^{e_C}\mathbb{Z}$, and computes the point $K_C = [m_C]P_C + [n_C]Q_C$ together with the curves and isogenies in Figure 5.2. Here $E_C = E/\langle K_C \rangle$, $E_{MC} = E_M/\langle \phi_M(K_C) \rangle = E_C/\langle \phi_C(K_M) \rangle$, $E_{AC} = E_A/\langle \phi_A(K_C) \rangle = E_C/\langle \phi_C(K_A) \rangle$, and $E_{AMC} = E_{MC}/\langle \phi_{C,MC}(K_A) \rangle$.
2. The signer outputs E_C, E_{AC}, E_{MC}, E_{AMC}, and $\ker(\phi_{C,MC})$ as the commitment.
3. The verifier randomly selects $b \in \{0,1\}$.

4. If $b = 0$, the signer outputs $\ker(\phi_C)$. Using the signer's public key, the verifier computes $\ker(\phi_{A,AC})$. Using knowledge of $\ker(\phi_M)$, the verifier computes $\phi_{M,MC}$. Using the auxiliary points given as part of the signature, the verifier can compute $\phi_{AM,AMC}$. The verifier checks that each isogeny maps between the corresponding two curves specified in the commitment. Using knowledge of $\ker(\phi_C)$, the verifier also independently re-computes $\phi_{C,MC}$ and checks that it matches the commitment.

5. If $b = 1$, the signer outputs $\ker(\phi_{C,AC})$. The verifier computes $\phi_{MC,AMC}$ and $\phi_{AC,AMC}$, and checks that each of $\phi_{C,AC}$, $\phi_{MC,AMC}$, and $\phi_{AC,AMC}$ maps between the corresponding two curves specified in the commitment.

We now describe the *disavowal protocol*. Suppose the signer is presented with a falsified signature (E_F, F_P, F_Q) for a message M, where E_F is the falsified E_{AM}, and $\{F_P, F_Q\}$ are the falsified auxiliary points corresponding to $\phi_{M,AM}(\phi_M(P_C))$ and $\phi_{M,AM}(\phi_M(Q_C))$ respectively. We must disavow E_F without disclosing E_{AM}. To do this, we blind E_{AM} as before to obtain E_{AMC}, and disclose enough information to allow the verifier to compute E_{FC} and check that $E_{FC} \neq E_{AMC}$.

1. The signer secretly selects random integers $m_C, n_C \in \mathbb{Z}/\ell_C^{e_C}\mathbb{Z}$, and computes $K_C = [m_C]P_C + [n_C]Q_C$ along with all the curves and isogenies in Figure 5.3.

2. The signer outputs E_C, E_{AC}, E_{MC}, E_{AMC}, and $\ker(\phi_{C,MC})$ as the commitment.

3. The verifier randomly selects $b \in \{0, 1\}$.

4. If $b = 0$, the signer outputs $\ker(\phi_C)$. The verifier computes ϕ_C, $\phi_{M,MC}$, $\phi_{A,AC}$, and $\phi_F \colon E_F \to E_{FC} = E_F/\langle [m_C]F_P + [n_C]F_Q \rangle$, and checks that each isogeny maps between the corresponding two curves specified in the commitment. The verifier independently re-computes $\phi_{C,MC}$ and checks that it matches the commitment. The verifier also checks that $E_{FC} \neq E_{AMC}$.

5. If $b = 1$, the signer outputs $\ker(\phi_{C,AC})$. The verifier computes $\phi_{AC,AMC}$ and $\phi_{MC,AMC}$, and checks that these isogenies map to E_{AMC}.

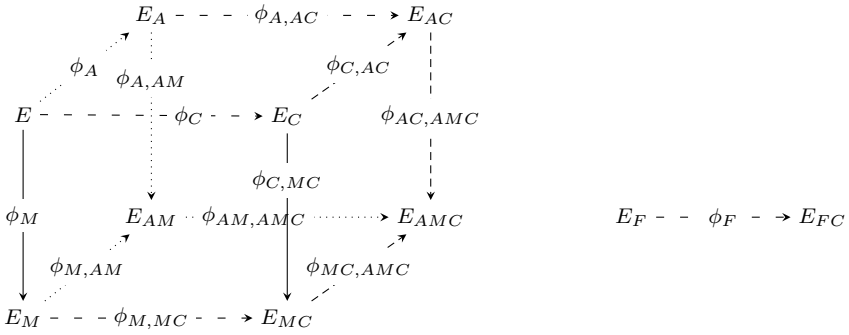

Figure 5.3: Disavowal protocol.

5.3 Complexity Assumptions

As before, let p be a prime of the form $\ell_A^{e_A}\ell_B^{e_B}\ell_C^{e_C} \cdot f \pm 1$, and fix a supersingular curve E over $\mathbb{F}_{p^2}$ together with bases $\{P_A, Q_A\}$, $\{P_B, Q_B\}$, and $\{P_C, Q_C\}$ of $E[\ell_A^{e_A}]$, $E[\ell_B^{e_B}]$, and $E[\ell_B^{e_B}]$ respectively.

We recall that we have assumptions stated in Section 5.3, which are DSSI, CSSI, SSCDH, SSDDH, and DSSP.

In analogy, we define the following computational problems, which we assume are quantum-infeasible:

Problem 5.3.1 (Modified Supersingular Computational Diffie-Hellman (MSSCDH) problem). With notation as in the SSDDH problem, given E_A, E_B, and $\ker(\phi_B)$, determine E_{AB}. Note that no auxiliary points for ϕ_A are given.

Problem 5.3.2 (Modified Supersingular Decisional Diffie-Hellman (MSSDDH) problem). With notation as in the SSDDH problem, given E_A, E_B, E_C, and $\ker(\phi_B)$, determine whether $E_C = E_{AB}$. Note that no auxiliary points for ϕ_A are given.

Problem 5.3.3 (One-sided Modified Supersingular Computational Diffie-Hellman (OMSSCDH) problem). For fixed E_A and E_B, given an oracle to solve MSSCDH for any E_A, $E_{B'}$, $\ker(\phi_{B'})$ where $E_{B'} \ncong E_B$, solve MSSCDH for E_A, E_B, and $\ker(\phi_B)$.

Problem 5.3.4 (One-sided Modified Supersingular Computational Diffie-Hellman (OMSSCDH) problem). For fixed E_A, E_B, and E_C, given an oracle to solve MSSCDH for any E_A, $E_{B'}$, $\ker(\phi_{B'})$ where $E_{B'} \ncong E_B$, solve MSSDDH for E_A, E_B, E_C, and $\ker(\phi_B)$.

We conjecture that these problems are computationally infeasible, in the sense that for any polynomial-time solver algorithm, the advantage of the algorithm is a negligible function of the security parameter $\log p$. The resulting security assumptions are referred to as the DSSI assumption, CSSI assumption, etc.

We also need a heuristic assumption concerning the distribution of blinded false signatures:

Assumption 5.3.5. Fix a supersingular elliptic curve E, an $\ell_A^{e_A}$-isogeny ϕ_A, an $\ell_B^{e_B}$-isogeny ϕ_B, and a curve E_F, not isomorphic to E_{AB}. For any pair of points $\{P, Q\}$ in E_F, only a negligibly small fraction of integer pairs m_C, n_C satisfy
$E_F/\langle m_C P + n_C Q\rangle = E_{AB}/\langle \phi_{B,AB}(\phi_B(m_C P_C + n_C Q_C))\rangle$.

This assumption is desirable, as in the disavowal protocol, it should be intractable to find m_C, n_C such that we obtain a map $E_F \to E_{ABC}$. This assumption makes sense, as the adversary does not have sufficient information to obtain that map, as the curves E_{FC} and E_{ABC} would be independent of each other. This assumption has also been computationally tested.

5.3.1 Hardness Of The Underlying Assumptions

All of our unmodified complexity assumptions (those not containing "Modified" in the name) are identical to the corresponding assumptions from [DFJP14, JDF11], except that our assumptions are formulated using primes of the form $p = \ell_A^{e_A}\ell_B^{e_B}\ell_C^{e_C} \cdot f \pm 1$, rather than primes of the form $p = \ell_A^{e_A}\ell_B^{e_B} \cdot f \pm 1$. We have no reason to believe that this alteration would affect the validity of these assumptions. A close analogy to this situation is the comparison between three-prime RSA and two-prime RSA, where the use of three-prime RSA incurs some efficiency loss but no known security concerns.

Our modified assumptions are needed in order to prove the security of our undeniable signature scheme. The MSSCDH and MSSDDH assumptions are complementary to the SSCDH and SSDDH assumptions, with the main difference being that the input consists of a kernel but not a pair of auxiliary points (rather than the other way around). We believe these assumptions are credible and comparable to SSCDH/SSDDH. The OMSSCDH and OMSSDDH assumptions are somewhat more artificial, and more study will be needed to justify confidence in them. They arise naturally in the analysis of our undeniable signature scheme.

Our heuristic assumption (Assumption 5.3.5) seems quite natural, and we have conducted numerous empirical experiments confirming it in practice. It would in fact be quite surprising if the assumption failed to hold. However, we have not yet succeeded in finding a proof of the assumption.

5.4 Security Proofs

To prove the security of our scheme, we must show that the confirmation and disavowal protocols are complete, sound and zero-knowledge, and that the overall scheme satisfies the unforgeability and invisibility properties.

The basic principle behind the proofs is that, as was the case in the basic key-exchange protocol (Section 4.2.2), knowledge of (the kernels of) any two opposite-side isogenies lying in a given cube face reveals no information about the other edges in the cube, by the DSSI and DSSP assumptions. On the other hand, knowledge of any two adjacent isogenies in a given commutative square yields full information about all the isogenies in the square. It does not matter which direction the arrows point, since one can reverse the direction of any arrow using dual isogenies.

Remark 5.4.1. To compute the dual isogeny of an isogeny $\phi\colon E \to E_A = E/\langle A\rangle$ whose kernel is generated by a point A, pick any point $B \in E \setminus \langle A\rangle$, and compute $\phi(B)$. Then $\phi(B)$ generates a kernel subgroup whose corresponding isogeny $\phi'\colon E_A \to E = E_A/\langle\phi(B)\rangle$ is isomorphic to the dual isogeny $\hat{\phi}$. In general, $E_A/\langle\phi(B)\rangle$ is isomorphic but not equal to E, so we also need to compute the appropriate isomorphism, but computing isomorphisms in general is known to be easy [Gal99].

5.4.1 Confirmation Protocol

We need to prove three things: *completeness*, *soundness* and *zero-knowledge*. We apply classical techniques from [FFS88, GMW91].

Proof of completeness. Completeness for this protocol is obvious. Using the algorithm presented in Section 5.2.2, the signer can always compute the diagram in Figure 5.2 and make the verifier accept. □

Proof of soundness. Let Charles be a cheating prover that is able to convince the verifier with non-negligible probability. We also assume that Charles is polynomially bounded. We treat Charles as a black-box that we can control in the sense that we can restart it a polynomial number of times on the same input and each time ask a different set of questions. We can then learn with high probability the diagram in Figure 5.4. Knowing this diagram, we can compute $\ker(\phi_A)$, since we know 3 out of 4 edges in the top face. We then have full knowledge of ϕ_A, and can then trivially solve the MCSSI problem for the left face. □

Proof of zero-knowledge. We show how a cheating verifier (CV) can construct a simulator S. The simulator S makes uniformly random guesses about what the verifier's challenge

will be. Regardless of the guess, S chooses random integers $m_C, n_C \in \mathbb{Z}/\ell_C^{e_C}\mathbb{Z}$ and computes $\phi_C \colon E \to E_C = E/\langle m_C P_C + n_C Q_C \rangle$.

If S guesses $b = 0$, it computes the diagram given in Figure 5.6. The simulator can now answer the CV's challenge in the case $b = 0$. The simulator's response is indistinguishable from, and indeed identical to, that of the real prover.

If S guesses $b = 1$, it chooses some random isogeny $\phi_{C,AC} \colon E_C \to E_{AC}$, and computes the diagram given in Figure 5.7. The simulator uses this diagram to answer the CV's challenge in the case $b = 1$. In this diagram, the curves E_C and E_{MC} are genuine, and the curves E_{AC} and E_{AMC} are fake. However, the CV cannot tell that these curves are fake, or else it would be able to solve DSSP for the top face of the cube. Hence the simulator's response is indistinguishable from that of the real prover. □

5.4.2 Disavowal Protocol

As before, we prove *completeness*, *soundness* and *zero-knowledge*.

Proof of completeness. Completeness for this protocol is obvious. Using the algorithm presented in Section 5.2.2, the signer can always compute the diagram in Figure 5.2 and make the verifier accept. Assumption 5.3.5 is needed to guarantee acceptance. □

Proof of soundness. Let Charles be a cheating prover that is able to convince the verifier with non-negligible probability. We also assume that Charles is polynomially bounded. We treat Charles as a black-box that we can control in the sense that we can restart it a polynomial number of times on the same input and each time ask a different set of questions. We can then learn with high probability the diagram in Figure 5.5. Knowing this diagram, we can compute $\ker(\phi_A)$, since we know 3 out of 4 edges in the top face. We then have full knowledge of ϕ_A, and can then trivially solve the MCSSI problem for the left face. □

Proof of zero-knowledge. We show how a cheating verifier (CV) can construct a simulator S. The simulator S makes uniformly random guesses about what the verifier's challenge will be. The simulator S first chooses random integers $m_C, n_C \in \mathbb{Z}/\ell_C^{e_C}\mathbb{Z}$ and computes $\phi_{M,MC} \colon E_M \to E_{MC} = E_M/\langle m_C \phi_M(P_C) + n_C \phi_M(Q_C) \rangle$.

If S guesses $b = 0$, it computes the diagram given in Figure 5.8. Here the curves E_C, E_{MC}, and E_{AC} are genuine, and the curves E_{AM} and E_{AMC} are fake. The simulator uses the diagram to answer the CV's challenge in the case $b = 0$. The simulator's response is indistinguishable from the real prover, since otherwise the CV could solve DSSP for the bottom face of the cube.

If S guesses $b = 1$, it chooses some random isogeny $\phi_{C,AC} \colon E_C \to E_{AC}$, and computes the diagram given in Figure 5.9. The simulator uses this diagram to answer the CV's challenge in the case $b = 1$. In this diagram, the curves E_C and E_{MC} are genuine, and the curves E_{AC} and E_{AMC} are fake. However, the CV cannot tell that these curves are fake, or else it would be able to solve DSSP for the top face of the cube. Hence the simulator's response is indistinguishable from that of the real prover.

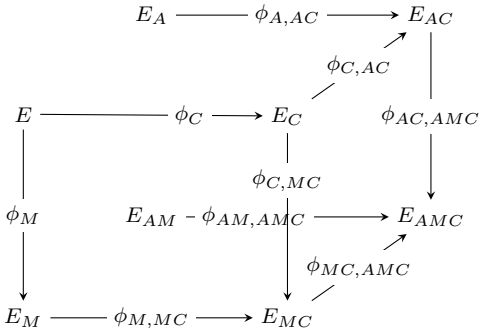

Figure 5.4: Proof of soundness (confirmation)

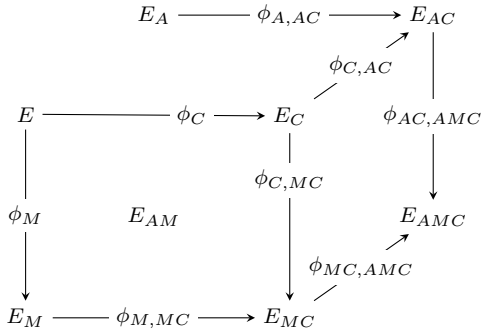

Figure 5.5: Proof of soundness (disavowal)

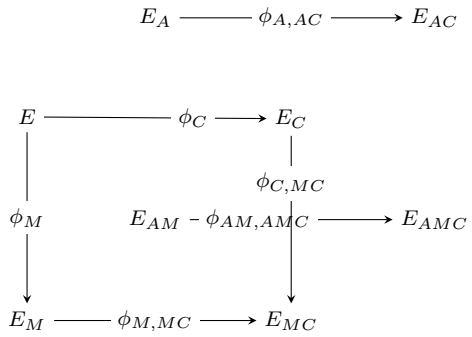

Figure 5.6: Confirmation ($b = 0$ case)

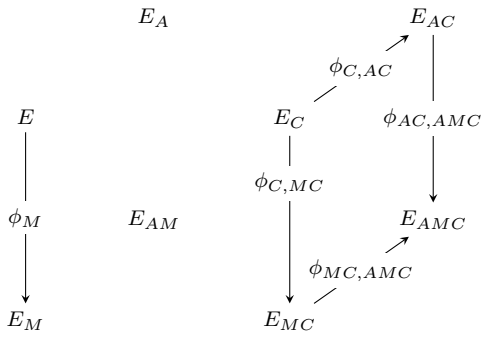

Figure 5.7: Confirmation ($b = 1$ case)

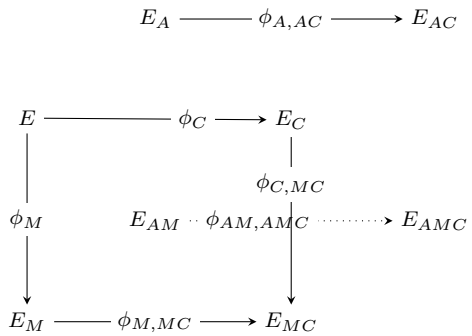

Figure 5.8: Disavowal ($b = 0$ case)

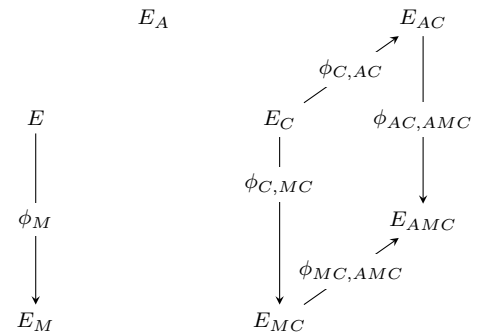

Figure 5.9: Disavowal ($b = 1$ case)

□

5.4.3 Unforgeability and Invisibility

Finally, we prove that the protocol satisfies the unforgeability and invisibility properties from Section 5.2.

Proof of unforgeability. To prove unforgeability, we must show that after making a polynomial number of queries to a signing oracle, an adversary is still unable to generate a valid signature. Note that we have shown that the confirmation and disavowal protocols are zero-knowledge. Forging signatures is then equivalent to solving OMSSCDH. □

Proof of invisibility. To prove invisibility, we must show that after making a polynomial number of queries to a signing oracle, an adversary will still be unable to decide whether a given signature is valid. This problem is equivalent to OMSSDDH. □

5.5 Parameter Sizes

As stated in [DFJP14, JDF11], the fastest known quantum isogeny finding algorithms in our setting require $O(n^{1/3})$ running time, where n is the size of the kernel. Based on this figure, we obtain the following parameter sizes and signature sizes for various levels of security:

Security level	$\log_2 p$	Signature size
80 bits	720	5760 bits
112 bits	1008	8064 bits
128 bits	1152	9216 bits
192 bits	1728	13824 bits
256 bits	2304	18432 bits

These numbers compare favorably with those of the only other available quantum-resistant undeniable signature scheme, that of Aguilar-Melchor et al. [AMBGS13]. For example, at the 128-bit security level, the scheme of [AMBGS13] requires a signature size of 5000 bits for the code-based portion plus an additional "roughly 40k Bytes" [AMBGS13, p. 116] for the conventional digital signature portion.

Regarding performance, a comparison is difficult because [AMBGS13] does not provide any performance numbers. For isogeny computations, recent implementation work of De

Feo et al. [DFJP14, Table 3] and Fishbein [Fis14, Figure 4.1] demonstrates that a single 1024-bit isogeny computation can be performed in 120 ms on a desktop PC, and in under 1 second on an Android device. Our protocol requires three such computations for signing, up to eight for confirmation, and up to nine for disavowal.

5.6 Conclusion

In this chapter we presented a quantum-resistant undeniable signature scheme based on the hardness of computing isogenies between supersingular elliptic curves. Our scheme represents the first quantum-resistant undeniable signature scheme based on a number-theoretic computational assumption, and compares well with the only prior undeniable quantum-resistant signature scheme (a code-based scheme) in terms of performance and bandwidth. Future work may entail developing new protocols such as digital signature schemes or more efficient schemes based on weaker assumptions.

Chapter 6

Post-Quantum Security Models For Authenticated Encryption

6.1 Introduction

This chapter is based on [SJS16], authored jointly with my supervisor David Jao, and Srinath Seshadri.

Authenticated encryption (AE) forms a critical component of our existing internet infrastructure, with many widely used protocols such as TLS, SSH, and IPsec depending on AE for their basic functionality. Despite this importance, there is relatively little existing literature on the subject of combining post-quantum authentication and encryption schemes in a provably secure way. A few works [BCNS14, FSXY13, SWZ15] have dealt with the problem of post-quantum authenticated key exchange, but do not provide any self-contained discussion of AE outside of the (much) more complicated context of key exchange; moreover, [BCNS14] and [SWZ15] simply use RSA and DH respectively for long-term authentication keys, on the grounds that there is no immediate need for quantum-safe authenticity. In this work, we adopt a different goal: we propose security definitions for post-quantum AE with the goal of achieving authentication and confidentiality against fully quantum adversaries, and give examples of such AE schemes constructed from existing underlying symmetric-key and digital signature primitives, using the quantum random oracle for the latter. Although our definitions are technically new, they are largely based on combinations of existing ideas, allowing us to reuse security proofs from other settings in the present context.

Note that our emphasis in this work is on constructing generic compositions of confidentiality and authentication primitives, rather than specialized authenticated encryption modes of operation as in the CAESAR competition [MR14]. While specialized first-class

primitives are certainly valuable, we feel that understanding composed primitives represents a natural first step.

6.2 Security Definitions

Bellare and Namprempre [BN08] showed that an IND-CPA encryption scheme combined with a SUF-CMA message authentication code under the Encrypt-then-MAC paradigm yields an IND-CCA authenticated encryption scheme. We wish to obtain a generalization of this construction which works against quantum adversaries. As a starting point, we review the security definitions of Boneh and Zhandry [BZ13b] for symmetric-key encryption schemes and digital signatures.

The main idea in these definitions is to allow quantum queries. One might question why quantum queries would be needed. One answer is that, if we want our schemes to be implementable on quantum computers, then in this scenario a quantum query from an adversary could receive a quantum response. It seems prudent to consider the security of AE schemes in this situation. Of course, our specific proposals are post-quantum schemes, and they can also be implemented on a classical computer.

The most natural extension of IND-CPA security to the quantum setting consists of allowing full unrestricted quantum queries to the encryption oracle. However, Boneh and Zhandry showed [BZ13b, Theorems 4.2 and 4.4] that this definition is too powerful, in the sense that no encryption scheme satisfies this security definition. In place of full quantum queries, Boneh and Zhandry propose a definition in which challenge messages can only be encrypted classically [BZ13b, Definition 4.5]:

Definition 6.2.1 (IND-qCPA)**.** We say that a symmetric-key encryption scheme $\mathcal{E} = (\mathrm{Enc}, \mathrm{Dec})$ is indistinguishable under a quantum chosen message attack (IND-qCPA secure) if no efficient adversary $\mathcal{A}$ can win in the following game, except with probability at most $1/2 + \epsilon$:

Key generation: The challenger picks a random key k and a random bit b.

Queries: $\mathcal{A}$ is allowed to make two types of queries:

Challenge queries: $\mathcal{A}$ sends two messages m_0, m_1 (of equal length), to which the challenger responds with $c* = \mathrm{Enc}(k, m_b)$. (Note that only one such query can be made within one game.)

Encryption queries: For each such query, the challenger chooses randomness r, and encrypts each message in the superposition using r as randomness:

$$\sum_{m,c} \psi_{m,c} |m, c\rangle \mapsto \sum_{m,c} \psi_{m,c} |m, c \oplus \mathrm{Enc}(k, m; r)\rangle$$

Guess: $\mathcal{A}$ produces a bit b', and wins if $b = b'$.

Similarly, Boneh and Zhandry define the notion of quantum chosen ciphertext security [BZ13b, Definition 4.6]:

Definition 6.2.2 (IND-qCCA)**.** We say that a symmetric-key encryption scheme $\mathcal{E} = (\mathrm{Enc}, \mathrm{Dec})$ is indistinguishable under a quantum chosen ciphertext attack (IND-qCCA secure) if no efficient adversary $\mathcal{A}$ can win in the following game, except with probability at most $1/2 + \epsilon$:

Key generation: The challenger picks a random key k and a random bit b. It also creates creates a list $\mathcal{C}$ which will store challenger ciphertexts.

Queries: $\mathcal{A}$ is allowed to make three types of queries:

Challenge queries: $\mathcal{A}$ sends two messages m_0, m_1 (of equal length), to which the challenger responds with $c* = \mathrm{Enc}(k, m_b)$. (Note that only one such query can be made within one game.)

Encryption queries: For each such query, the challenger chooses randomness r, and encrypts each message in the superposition using r as randomness:

$$\sum_{m,c} \psi_{m,c} |m, c\rangle \mapsto \sum_{m,c} \psi_{m,c} |m, c \oplus \mathrm{Enc}(k, m; r)\rangle$$

Decryption queries: For each such query, the challenger decrypts all ciphertexts in the superposition, except those that were the result of a challenge query:

$$\sum_{c,m} \psi_{c,m} |c, m\rangle \mapsto \sum_{c,m} \psi_{c,m} |c, m \oplus f(c)\rangle$$

where

$$f(c) = \begin{cases} \perp & \text{if } c \in \mathcal{C} \\ \mathrm{Dec}(k, c) & \text{otherwise.} \end{cases}$$

Guess: $\mathcal{A}$ produces a bit b', and wins if $b = b'$.

For the above definitions, $\mathcal{A}$ is allowed to make polynomial (in the size of the security parameter) number of queries. The value r is the same for all messages within the same query. There is no limit on the number of messages within the same query (i.e. the superposition can involve as many messages, as wanted).

We now discuss Boneh and Zhandry's quantum security definition for signatures. It is assumed that the adversary can query for signatures of superpositions of messages. In this

situation, the definition of existential unforgeability needs to be modified, since a naive reading of the definition would allow the adversary simply to measure a superposition and claim the resulting signature as an existential forgery. Let q be a polynomial function in the security parameter. For our purposes, the security parameter is the finite field size. To solve this problem we simply require the adversary to produce $q+1$ signatures from q queries [BZ13b, Definition 3.2]:

Definition 6.2.3 (SUF-qCMA)**.** A signature scheme $\mathcal{S} = (\mathrm{Gen}, \mathrm{Sign}, \mathrm{Ver})$ is strongly unforgeable under a quantum chosen message attack (SUF-qCMA secure) if, for any efficient quantum algorithm $\mathcal{A}$ and any polynomial q, the algorithm $\mathcal{A}$'s probability of success in the following game is negligible in λ:

Key generation: The challenger runs $(sk, pk) \leftarrow \mathrm{Gen}(\lambda)$, and gives pk to $\mathcal{A}$.

Signing Queries: $\mathcal{A}$ makes a polynomial number q of chosen message queries. For each query, the challenger chooses randomness r, and responds by signing each message in the query using r as randomness:

$$\sum_{m,t} \psi_{m,t} |m, t\rangle \mapsto \sum_{m,t} \psi_{m,t} |m, t \oplus \mathrm{Sign}(sk, m; r)\rangle$$

Forgeries: $\mathcal{A}$ is required to produce $q+1$ message-signature pairs. The challenger then checks that all the signatures are valid, and that all message-signature pairs are distinct. If so, the adversary wins.

Definition 6.2.4 (WUF-qCMA)**.** A signature scheme $\mathcal{S}$ is weakly unforgeable under a quantum chosen message attack (WUF-qCMA secure) if it satisfies the same definition as SUF-qCMA, except that we require the $q+1$ message-signature pairs to have distinct messages.

Note that our terminology differs slightly from Boneh and Zhandry [BZ13b], although the content of the definitions is identical: Boneh and Zhandry use the terms "strongly EUF-qCMA" and "weakly EUF-qCMA" instead of SUF-qCMA and WUF-qCMA. In addition, Boneh and Zhandry have similar definitions for SUF-qCMA and WUF-qCMA secure message authentication codes [BZ13a].

Finally, we give our definitions of INT-qCTXT and INT-qPTXT. We constructed these definitions by starting with the classical security definitions of INT-CTXT and INT-PTXT from Bellare and Namprempre [BN08, §2], and modifying them in a manner similar to Boneh and Zhandry's definition for digital signatures (Definition 6.2.3).

Definition 6.2.5 (INT-qCTXT)**.** An encryption scheme $\mathcal{E} = (\text{Enc}, \text{Dec})$ satisfies integrity of ciphertext under a quantum attack (INT-qCTXT security) if, for any efficient quantum algorithm $\mathcal{A}$ and any polynomial q, the probability of success of $\mathcal{A}$ in the following game is negligible in λ:

Key generation: The challenger picks a random key k.

Encryption queries: $\mathcal{A}$ makes a polynomial q such queries. For each such query, the challenger chooses randomness r, and encrypts each message in the superposition using r as randomness:

$$\sum_{m,c} \psi_{m,c}|m, c\rangle \mapsto \sum_{m,c} \psi_{m,c}|m, c \oplus \text{Enc}(k, m; r)\rangle$$

Decryption queries: For each such query, the challenger decrypts all ciphertexts in the superposition, except those that were the result of a challenge query:

$$\sum_{c,m} \psi_{c,m}|c, m\rangle \mapsto \sum_{c,m} \psi_{c,m}|c, m \oplus Dec(k, c)\rangle$$

Forgeries: $\mathcal{A}$ is required to produce $q + 1$ message-ciphertext pairs. The challenger then checks that all the ciphertexts are valid, and that all message-ciphertexts pairs are distinct. If so, the adversary wins.

Definition 6.2.6 (INT-qPTXT)**.** An encryption scheme $\mathcal{E} = (\text{Enc}, \text{Dec})$ satisfies the integrity of plaintext under a quantum attack (INT-qPTXT secure) if it satifies the same definition as INT-qCTXT, except that we require the $q + 1$ message-ciphertext pairs to have distinct messages.

6.3 Main Theorem

In this section, we prove that an IND-qCPA encryption scheme together with a SUF-qCMA signature or MAC scheme yields an authenticated encryption scheme via the Encrypt-then-MAC method (where the sender first encrypts the message and then signs or produces the MAC for ciphertext), satisfying the respective privacy and integrity guarantees of IND-qCCA (Definition 6.2.2) and INT-qCTXT (Definition 6.2.5), the quantum analogues of the classical notions of IND-CCA and INT-CTXT security used in Bellare and Namprempre [BN08]. We adopt the proofs to work with the definitions for quantum adversary model. We will interchangeably be using Signature and MAC notion as they can replace each other. For verification in case of MAC, we simply mean that the party checks and verifies whether or not MAC of the given value is correct. We begin by showing a WUF-qCMA MAC implies INT-qPTXT security:

Theorem 6.3.1. *Let $\mathcal{SE} = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$ be a symmetric-key encryption scheme, let $\mathcal{MA} = (\mathcal{K}_m, \mathcal{T}, \mathcal{V})$ be a message authentication scheme, and let $\overline{\mathcal{SE}} = (\bar{\mathcal{K}}, \bar{\mathcal{E}}, \bar{\mathcal{D}})$ be the authenticated encryption scheme obtained from $\mathcal{SE}$ and $\mathcal{MA}$ via the Encrypt-then-MAC method. Given any adversary I against $\overline{\mathcal{SE}}$, we can construct an adversary F such that*

$$\mathrm{Adv}^{\text{INT-qPTXT}}_{\overline{\mathcal{SE}}}(I) \leq \mathrm{Adv}^{\text{WUF-qCMA}}_{\mathcal{MA}}(F).$$

Proof. (Based on [BN08, Theorem 4.1]) We construct the adversary F as follows:

1. Use the key $\mathcal{K}_e$.
2. Run I.
3. On query $\mathrm{Enc}(M)$ (where M can be in superposition):

$$C' \leftarrow \mathcal{E}(K_e, M); \tau \leftarrow \mathrm{Tag}(C');\ \text{Return } C' \parallel \tau \text{ to } I$$

4. On query $\mathrm{Ver}(C)$:

$$\text{Parse } C \text{ as } C' \parallel \tau'; v \leftarrow \mathrm{Ver}(C', \tau');\ \text{Return } v \text{ to } I$$

until I halts.

Let $C_i = C'_i \parallel \tau_i$ for $i \in \{1, \ldots, q+1\}$ be the Ver queries of I that lead to winning game INT-qPTXT$_{\overline{\mathcal{SE}}}$, after q queries to Enc. Let $M_i = \mathcal{D}(K_e, C'_i)$. We know that due to the property of INT-qPTXT of $\overline{\mathcal{SE}}$, at most q of them were obtained from the q queries to Enc of I; hence $C'_i s$ were the result of at most q queries of F to Tag, but we obtained $q+1$ valid tags. Hence, F wins whenever WUF-qCMA$_{\mathcal{MA}}$ I wins INT-qPTXT$_{\overline{\mathcal{SE}}}$. □

Although our proof of Theorem 6.3.1 is for MACs, the same proof works for digital signatures (replacing the Tag oracle with the Sign oracle). Of course, in the content of using signatures, we assume that the public keys are authenticated.

Next we show that a SUF-qCMA signature or MAC implies an INT-qCTXT authenticated encryption scheme.

Theorem 6.3.2. *Let $\mathcal{SE} = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$ be a symmetric-key encryption scheme, let $\mathcal{MA} = (\mathcal{K}_m, \mathcal{T}, \mathcal{V})$ be a message authentication scheme, and let $\overline{\mathcal{SE}} = (\bar{\mathcal{K}}, \bar{\mathcal{E}}, \bar{\mathcal{D}})$ be the authenticated encryption scheme obtained from $\mathcal{SE}$ and $\mathcal{MA}$ via encrypt-then-MAC composition method. Given any adversary I against $\overline{\mathcal{SE}}$, we can construct an adversary F such that*

$$\mathrm{Adv}^{\text{INT-qCTXT}}_{\overline{\mathcal{SE}}}(I) \leq \mathrm{Adv}^{\text{SUF-qCMA}}_{\mathcal{MA}}(F).$$

Proof. (Based on [BN08, Theorem 4.4]) Here we use the same adversary as in Theorem 6.3.1. Let $C_i = C_i' \parallel \tau_i$ for $i \in \{1, \ldots, q+1\}$ be the Ver queries of I that lead to winning game INT-qCTXT$_{\overline{\mathcal{SE}}}$, after q queries to Enc. If only at most q of the C_i's were returned to I by Enc, then at most q were queried by F with Tag (i.e., the corresponding $C_i's$). Hence, F wins whenever SUF-qCMA$_{\mathcal{MA}}$ I wins INT-qCTXT$_{\overline{\mathcal{SE}}}$. □

Again, the proof of Theorem 6.3.2 carries over to digital signatures as well, replacing the Tag oracle with a Sign oracle.

We now show that the authenticated encryption scheme in Encrypt-then-MAC inherits the IND-qCPA property from the underlying encryption scheme:

Theorem 6.3.3. *Let $\mathcal{SE} = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$ be a symmetric-key encryption scheme, let $\mathcal{MA} = (\mathcal{K}_m, \mathcal{T}, \mathcal{V})$ be a message authentication scheme, and let $\overline{\mathcal{SE}} = (\bar{\mathcal{K}}, \bar{\mathcal{E}}, \bar{\mathcal{D}})$ be the authenticated encryption scheme obtained from $\mathcal{SE}$ and $\mathcal{MA}$ via the Encrypt-then-MAC composition method. Given any adversary $\mathcal{A}$ against $\overline{\mathcal{SE}}$, we can construct an adversary $\mathcal{A}_p$ such that*

$$\mathrm{Adv}^{\text{IND-qCPA}}_{\overline{\mathcal{SE}}}(\mathcal{A}) \leq \mathrm{Adv}^{\text{IND-qCPA}}_{\mathcal{SE}}(\mathcal{A}_p).$$

Furthermore, $\mathcal{A}_p$ uses the same resources as $\mathcal{A}$.

Proof. (Based on [BN08, Theorem 4.3]) We construct $\mathcal{A}_p$ as follows:

$\mathcal{K}_m \leftarrow \mathcal{K}_m$
Run $\mathcal{A}$
On query to Enc
$C \leftarrow Enc(M)$
$\tau \leftarrow \mathrm{Tag}(\mathcal{K}_m, C)$
Return $C \parallel \tau$ to $\mathcal{A}$
Until $\mathcal{A}$ halts and returns b
Return b.

We can see that if $\mathcal{A}$ wins, then so does $\mathcal{A}_p$, since a winning output for $\mathcal{A}$ is a winning output for $\mathcal{A}_p$; the tag can be ignored. □

Finally, we prove that INT-qCTXT and IND-qCPA security imply IND-qCCA security (Theorem 6.3.7). The proof relies on three games G_0, G_1, and G_2 as defined in Figure 6.1. These games are based on the corresponding three games from Figure 7 of [BN08], except that we modify the games mutadis mutandis to conform to our quantum definitions (Definitions 6.2.1 and 6.2.2).

Games G_0 and $\boxed{G_1}$:	Game G_2 :
proc Initialize	**proc Initialize**
$K \xleftarrow{\$} \mathcal{K}; b \xleftarrow{\$} \{0,1\}; S \leftarrow \emptyset$	$K \xleftarrow{\$} \mathcal{K}; b \xleftarrow{\$} \{0,1\}$
proc Challenge(M_0, M_1)	**proc Challenge**(M_0, M_1)
$C \xleftarrow{\$} \mathrm{Enc}(K, M_b); S \leftarrow S \cup \{C\};$ Return C	$C \xleftarrow{\$} \mathrm{Enc}(K, M_b);$ Return C
proc Enc(M)	**proc Enc**(M)
$C \xleftarrow{\$} \mathrm{Enc}^*(K, M); S \leftarrow S \cup \{C\};$ Return C	$C \xleftarrow{\$} \mathrm{Enc}^*(K, M);$ Return C
proc Dec(C)	**proc Dec**(C)
$M \leftarrow \mathrm{Dec}^*(C)$	Return $\perp$
If $M \neq \perp$ then bad $\leftarrow$ true; $\boxed{M \leftarrow \perp}$	
Return M	
proc Finalize(d)	**proc Finalize**(d)
Return $(d = b)$	Return $(d = b)$

Figure 6.1: Games G_0, G_1, and G_2.

In figure 6.1 Game G_1 contains the code in the box while G_0 does not. The functions Enc^* and Dec^* refer to the encryption and decryption oracle functions from Definition 6.2.2.

The proof of Theorem 6.3.7 uses the identical until bad lemma [BN08, Lemma 2.1]:

We first define the term *identical until bad* using the definition in [BR06], and then move on to the lemma.

Definition 6.3.4 (Identical until bad)**.** Games G and H are said to be *identical until bad* if one has the statement if bad then S where the other has the empty statement.

Lemma 6.3.5. *(Identical until bad lemma) Let G_i and G_j be identical until bad games, and $\mathcal{A}$ an adversary. Then for any y: $Pr[G_i^{\mathcal{A}} \implies y] - Pr[G_j^{\mathcal{A}} \implies y] \leq Pr[G_j$ sets bad$]$.*

It is not immediately clear (to us, anyway) that the identical until bad lemma holds for quantum adversaries. Fortunately, in Theorem 6.3.7, we only need the special case $i = 0$, $j = 1$, and $y =$ true, and in this case we can prove the result for quantum adversaries. We use the following lemma of Shoup [Sho01, Lemma 1].

Lemma 6.3.6. *Let E, E', and F be events defined on a probability space such that $Pr[E \wedge \neg F] = Pr[E' \wedge \neg F]$. Then we have $|Pr[E] - Pr[E']| \leq Pr[F]$.*

This lemma holds regardless of whether or not the adversary is classical or quantum, as it is a mathematical statement. Define the event E to be $[G_0^{\mathcal{A}} \implies \text{true}]$ and E' to be $[G_1^{\mathcal{A}} \implies \text{true}]$. Define F to be $[G_1^{\mathcal{A}} \text{ sets bad}]$. Observe that in this case $E \wedge \neg F$ corresponds to the outcome $M = \perp$ in the game G_0, meaning that A wins the game. Similarly, $E' \wedge \neg F$ corresponds to the outcome $M = \perp$ in G_1, meaning that $\mathcal{A}$ wins the game. Note that for $M = \perp$, both G_0 and G_1 return the same responses, and hence have the same probability of winning. Hence, $Pr[E \wedge \neg F] = Pr[E' \wedge \neg F]$, which means Lemma 1 of [Sho01] can be applied to obtain $|Pr[E] - Pr[E']| \leq Pr[F]$. Finally, we need to remove the absolute values, to obtain $Pr[E'] \leq Pr[E]$. It is easy to see that we can do so, because for G_0 we sometimes return the message, while for G_1, we always return $M = \perp$, so that the success probability of G_0 is at least that of G_1. Hence the identical until bad lemma holds for quantum adversaries in the special case where $i = 0$, $j = 1$, and $y = \text{true}$.

We recall Definition (1) in [BN08]:

$$\mathrm{Adv}_{\mathcal{SE}}^{\text{IND-CCA}}(\mathcal{A}) = 2 \cdot \Pr[\text{IND-CCA}_{\mathcal{SE}}^{\mathcal{A}} \implies 1] - 1.$$

The quantum version of this definition is:

$$\mathrm{Adv}_{\mathcal{SE}}^{\text{IND-qCCA}}(\mathcal{A}) = 2 \cdot \Pr[\text{IND-qCCA}_{\mathcal{SE}}^{\mathcal{A}} \implies 1] - 1.$$

Theorem 6.3.7. *Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme. Let $\mathcal{A}$ be an IND-qCCA adversary against $\mathcal{SE}$ running in time t and making q_e Enc queries and q_d Dec queries. Then, we can construct an INT-qCTXT adversary $\mathcal{A}_c$ and IND-qCPA adversary $\mathcal{A}_p$ such that*

$$\mathrm{Adv}_{\mathcal{SE}}^{\text{IND-qCCA}}(\mathcal{A}) \leq 2 \cdot \mathrm{Adv}_{\mathcal{SE}}^{\text{INT-qCTXT}}(\mathcal{A}_c) + \mathrm{Adv}_{\mathcal{SE}}^{\text{IND-qCPA}}(\mathcal{A}_p).$$

Furthermore, $\mathcal{A}_c$ runs in time $O(t)$ and makes q_e Enc queries and q_d Ver queries, while $\mathcal{A}_p$ runs in time $O(t)$ and makes q_e queries of target messages M_i.

Proof. We have:

$$\begin{aligned}
\Pr[\text{IND-qCCA}_{\mathcal{SE}}^{\mathcal{A}} \implies \text{true}] &= \Pr[G_0^{\mathcal{A}} \implies \text{true}] \\
&= \Pr[G_1^{\mathcal{A}} \implies \text{true}] + \\
&\quad (\Pr[G_0^{\mathcal{A}} \implies \text{true}] - Pr[G_1^{\mathcal{A}} \implies \text{true}]) \\
&\leq \Pr[G_1^{\mathcal{A}} \implies \text{true}] + \Pr[G_1^{\mathcal{A}} \text{ sets bad}] \qquad (6.1)
\end{aligned}$$

The last inequality follows from the identical until bad lemma in the special case $i = 0$, $j = 1$, and $y = \text{true}$ (which we proved above). Now, observe that for Dec, G_1 always returns $\perp$, and hence

$$\Pr[G_1^{\mathcal{A}} \implies \text{true}] = \Pr[G_2^{\mathcal{A}} \implies \text{true}]. \tag{6.2}$$

Let us now define the adversary $\mathcal{A}_p$. It simply runs $\mathcal{A}$, answering $\mathcal{A}$'s challenge and encryption queries with its own queries, and answering $\mathcal{A}$'s queries for decryption with $\perp$. It outputs whatever $\mathcal{A}$ outputs. Hence, we get:

$$\Pr[G_2^{\mathcal{A}} \implies \text{true}] \leq \Pr[\text{IND-qCPA}_{\mathcal{SE}}^{\mathcal{A}_p} \implies \text{true}]. \tag{6.3}$$

Next, we define the adversary $\mathcal{A}_c$. The adversary $\mathcal{A}_c$ picks a random bit b, then runs $\mathcal{A}$ and answers its queries as follows. For challenge and encryption queries, $\mathcal{A}_c$ submits challenge and encryption queries and returns the results to $\mathcal{A}$. For the Dec query, $\mathcal{A}_c$ submits it to the Ver oracle, and, regardless of the response, returns $\perp$ to $\mathcal{A}$. Hence, we get:

$$\Pr[G_1^{\mathcal{A}} \text{ sets bad}] \leq \Pr[\text{INT-qCTXT}_{\mathcal{SE}}^{\mathcal{A}_c} \implies \text{true}]. \tag{6.4}$$

Combining the definition

$$\text{Adv}_{\mathcal{SE}}^{\text{IND-qCCA}}(\mathcal{A}) = 2 \cdot \Pr[\text{IND-qCCA}_{\mathcal{SE}}^{\mathcal{A}} \implies 1] - 1$$

with Equations (6.1), (6.2), (6.3), and (6.4), we obtain

$$\text{Adv}_{\mathcal{SE}}^{\text{IND-qCCA}}(\mathcal{A}) \leq 2 \cdot \text{Adv}_{\mathcal{SE}}^{\text{INT-qCTXT}}(\mathcal{A}_c) + \text{Adv}_{\mathcal{SE}}^{\text{IND-qCPA}}(\mathcal{A}_p).$$

□

Combining Theorems 6.3.2, 6.3.3, and 6.3.7, we obtain our main theorem:

Theorem 6.3.8. *Let $\mathcal{SE} = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$ be a symmetric-key encryption scheme, let $\mathcal{MA} = (\mathcal{K}_m, \mathcal{T}, \mathcal{V})$ be a message authentication scheme, and let $\overline{\mathcal{SE}} = (\bar{\mathcal{K}}, \bar{\mathcal{E}}, \bar{\mathcal{D}})$ be the authenticated encryption scheme obtained from $\mathcal{SE}$ and $\mathcal{MA}$ via the Encrypt-then-MAC composition method. Given that $\mathcal{SE}$ is IND-qCPA and $\mathcal{MA}$ is SUF-qCMA, then the resulting $\overline{\mathcal{SE}}$ is IND-qCCA.*

Proof. By Theorem 6.3.2, since $\mathcal{MA}$ is SUF-qCMA, we get that $\overline{\mathcal{SE}}$ is INT-qCTXT. Also, by Theorem 6.3.3, since $\mathcal{SE}$ is IND-qCPA, we get that $\overline{\mathcal{SE}}$ is also IND-qCPA. Finally, because $\overline{\mathcal{SE}}$ is INT-qCTXT and IND-qCPA, by Theorem 6.3.7, we get that it is IND-qCCA. □

As with Theorems 6.3.1 and 6.3.2, Theorem 6.3.8 also holds with digital signature schemes used in place of MACs.

6.4 Quantum-Resistant Strongly Unforgeable Signature Schemes

In this section we examine some concrete choices of strongly unforgeable signature/MAC schemes which could be suitable for our AE construction. We limit ourselves to only a few representative examples to illustrate the general idea. We focus on signature schemes as in our view they are somewhat more interesting, but similar ideas apply to MACs [BZ13a]. We begin with a review of the Boneh-Zhandry transformation [BZ13b, Construction 3.12] for transforming any classically strongly secure digital signature scheme into a SUF-qCMA scheme:

Construction 6.4.1. Let $S_c = (\mathrm{Gen}_c, \mathrm{Sign}_c, \mathrm{Ver}_c)$ be a be a signature scheme, H be a hash function, and $\mathcal{Q}$ be a family of pairwise independent functions mapping messages to the randomness used by Sign_c, and k some polynomial in λ. Define $S = (\mathrm{Gen}, \mathrm{Sign}, \mathrm{Ver})$ where:

- $\mathrm{Gen}(\lambda) = \mathrm{Gen}_c(\lambda)$
- $\mathrm{Sign}(sk, m)$:
  - Select $Q \in \mathcal{Q}$, $r \in \{0,1\}^k$ at random.
  - Set $s = Q(m), h = H(m, r), \sigma = \mathrm{Sign}_c(sk, h; s)$. Output (r, σ).
- $\mathrm{Ver}(pk, m, (r, \sigma))$:
  - Set $h = H(m, r)$. Output $\mathrm{Ver}_c(pk, h, \sigma)$.

If the original signature scheme S_c is SUF-CMA against a classical chosen message attack performed by a quantum adversary, then by [BZ13b, Corollary 3.17] the transformed scheme S is SUF-qCMA in the quantum random oracle model.

Furthermore, if the verification function in the signature scheme S_c involves independently deriving the value of σ and checking whether or not the derived value matches the value which was originally sent, a further optimization is possible: one can hash σ to reduce its length to a minimum. We employ this optimization in our examples.

Note that we have included $\mathcal{Q}$. An example of $\mathcal{Q}$, can be a family of hash functions. We can further assume that whenever needed, we are using this example.

6.4.1 Strong Designated Verifier Signatures from Isogenies

A strong designated verifier signature (SDVS) scheme [JSI96] is a digital signature scheme in which only a designated party (specified at the time of signing) can verify signatures, and

verification requires that party's private key. Note that an SDVS is enough for AE, since only the two parties participating in the AE protocol need to be able to verify signatures.

Sun, Tian, and Wang in [STW12] present an isogeny-based SDVS scheme, and give a classical security reduction to the SSDDH problem [JDF11], which is believed to be infeasible on quantum computers. This reduction qualifies as a straight-line reduction in the sense of the security framework of Song [Son14], and hence remains valid for quantum adversaries. However, the reduction only establishes SUF-CMA security, not SUF-qCMA security. Applying the Boneh-Zhandry transformation (Construction 6.4.1), we obtain the following SDVS scheme, which is SUF-qCMA:

Setup: Fix a prime $p = \ell_A^{e_A}\ell_B^{e_B} \cdot f \pm 1$, a supersingular base curve E over $\mathbb{F}_{p^2}$, generators $\{P_A, Q_A\}$ of $E[\ell_A^{e_A}]$, and generators $\{P_B, Q_B\}$ of $E[\ell_B^{e_B}]$. Let $H_1, H_2 \colon \{0,1\}^* \to \{0,1\}^k$ be independent secure hash functions (with parameter k), and $\mathcal{Q}$ a family of pairwise independent functions mapping messages to the randomness used in signing.

Key generation: A signer selects at random $m_S, n_S \in \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$, not both divisible by ℓ_A, and then computes an isogeny $\phi_S \colon E \to E_S = E/\langle [m_S]P_A + [n_S]Q_A \rangle$ and the values $\phi_S(P_B)$ and $\phi_S(Q_B)$. The private key is (m_S, n_S) and the public key is the curve E_S and the points $\phi_S(P_B)$ and $\phi_S(Q_B)$.
A designated verifier selects at random $m_V, n_V \in \mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$, not both divisible by ℓ_B, and then computes an isogeny $\phi_V \colon E \to E_V = E/\langle [m_V]P_B + [n_V]Q_B \rangle$ and the values $\phi_V(P_A)$ and $\phi_V(Q_A)$. The private key is (m_V, n_V) and the public key is the curve E_V and the points $\phi_V(P_A)$ and $\phi_V(Q_A)$.

Signing: Select at random $Q \in \mathcal{Q}, r \in \{0,1\}^k$ for use in the Boneh-Zhandry transformation. Compute $s = Q(m)$, $h = H_1(m, r)$, and $\phi'_S \colon E_V \to E_{SV} = E_V/\langle [m_S]\phi_V(P_A) + [n_S]\phi_V(Q_A) \rangle$. Set $\sigma = H_2(h||j(E_{SV})||s)$. The signature is (r, σ).

Verification: Compute $\phi'_V \colon E_S \to E_{SV} = E_S/\langle [m_V]\phi_S(P_B) + [n_V]\phi_S(Q_B) \rangle$ and $h = H_1(m, r)$. Set $\sigma' = H_2(h||j(E_{SV})||Q(m))$. Verify that $\sigma' \stackrel{?}{=} \sigma$.

6.4.2 Ring-LWE Signatures

To give another example, we combine the Ring-LWE signature scheme of Güneysu et al. [GLP15] with Construction 6.4.1 from [BZ13b] to obtain a SUF-qCMA signature scheme based on Ring-LWE:

Setup: Set $R = \mathbb{F}_q/\langle x^n + 1 \rangle$ where n is a power of 2. Let $H_1 \colon \{0,1\}^* \to \{0,1\}^k$ and $H_3 \colon \{0,1\}^* \to R$ be independent secure hash functions (with parameter k) and $\mathcal{Q}$ a family of pairwise independent functions mapping messages to the randomness used in the signing function. Choose a bound B on the maximum coefficient size.

Key generation: A signer generates two small polynomials $s_1(x), s_2(x) \in R$, selects $a(x) \in R$ at random, and computes the public key $t(x) = as_1(x) + s_2(x)$.

Signing: Select $Q \in \mathcal{Q}$, $r \in \{0,1\}^k$ at random for the Boneh-Zhandry transformation, and $y_1(x), y_2(x) \in R$ at random for the signature scheme. Compute $s = Q(m)$, $h = H_1(m, r)$, and $c(x) = H_3(\text{BitString}(a(x)y_1(x) + y_2(x))||h||s)$. Finally, compute $z_1(x) = s_1(x)c(x) + y_1(x)$ and $z_2(x) = s_2(x)c(x) + y_2(x)$. Check that the coefficients of the polynomials $z_1(x), z_2(x)$ are within the bound B; if not, restart. The signature is $(r, z_1(x), z_2(x), c(x))$

Verification: Check that the coefficients of the polynomials $z_1(x), z_2(x)$ are within the bound B; if not, reject. Compute x $h = H_1(m, r)$, and check whether $c(x) \stackrel{?}{=} H_3(a(x)z_1(x) + z_2(x) - t(x)c(x)||h||Q(m))$. If so, accept; otherwise reject.

6.5 Quantum-Resistant Authenticated Encryption

We give a generic construction of authenticated encryption schemes which are provably quantum-resistant in the sense of INT-qCTXT and IND-qCCA. For the underlying encryption scheme, we assume that a classical symmetric-key block cipher $\mathcal{E}$ in a suitable block cipher mode of operation with random IVs will suffice to provide quantum security, taking care to use 2ℓ key sizes to obtain ℓ bits of security. We refer to [ATTU16] for a discussion of the choice of the mode of operation. For the MAC/signature scheme we can employ the Boneh-Zhandry transformation on any SUF-CMA scheme secure against quantum adversaries as described in Section 6.4. Combining those two components, we obtain an IND-qCCA and INT-qCTXT authenticated encryption scheme as follows:

Setup:

1. Choose parameters for the underlying encryption and signature schemes.
2. Let $H\colon \{0,1\}^* \to \{0,1\}^k$ be a secure hash function (with security parameter k).
3. Let $\mathcal{Q}$ be a family of pairwise independent functions mapping messages to the randomness used in the signature scheme.

Key generation:

1. Alice chooses her private parameters for the encryption and signature schemes. If required, she produces and publishes the corresponding public keys.
2. Bob chooses his private parameters for the encryption and signature schemes. If required, he produces and published the corresponding public keys.

Encryption:

Suppose Bob wants to send a message $m \in \{0,1\}^*$ to Alice.

1. Using the common encryption key e that he shares with Alice, encrypt the message using the underlying symmetric-key encryption scheme to obtain $c = \mathcal{E}(e, m)$.
2. Select $Q \in \mathcal{Q}$, $r \in \{0,1\}^k$ at random.
3. Compute $t = Q(m)$.
4. Computes the value $h = H(c, r)$.
5. Using h and his private signing key s, Bob computes the authentication tag $\sigma = \text{Sign}(s, h; t)$.
6. The ciphertext is $\{c, r, \sigma\}$.

Decryption:

Suppose Alice receives ciphertext $\{c, r, \sigma\}$ from Bob.

1. Compute the value $h = H(c, r)$.
2. Using h and Bob's public signing key p, compute the verification function $\text{Ver}(s, h, r, \sigma)$, if it returns true, continue; if not, stop.
3. Using the common encryption key e that she shares with Bob, decrypt the message and obtain $m = \mathcal{D}(e, c)$.

Again, in the case where the verification function in the signature scheme involves independently deriving the value of σ and checking that the derived value matches the value which was originally sent, we can hash σ prior to transmission to reduce its length to a minimum.

6.6 Isogeny-Based Quantum-Resistant Authenticated Encryption Scheme

In this section, we propose a quantum-resistant authenticated encryption scheme based on isogenies between supersingular elliptic curves. For the key exchange step, we use the previous key exchange scheme of Jao and De Feo from [JDF11] (presented in Chapter 4), and for the signature scheme, we use the strong designated-verifier scheme from Section 6.4.1.

Setup:

1. Choose primes $\ell_A, \ell_B, \ell_{A'}, \ell_{B'}, p, p'$ and exponents $e_A, e_B, e_{A'}, e_{B'}$ such that $p = \ell_A^{e_A}\ell_B^{e_B} \cdot f \pm 1$ and $p' = \ell_{A'}^{e_{A'}}\ell_{B'}^{e_{B'}} \cdot f' \pm 1$ give us supersingular elliptic curves $E/\mathbb{F}_{p^2}$ (which denote simply by E) and $E/\mathbb{F}_{p'^2}$ (which denote simply by E').

2. Choose bases $\{P_A, Q_A\}$ and $\{P_B, Q_B\}$, which generate $E[\ell_A^{e_A}]$ and $E[\ell_B^{e_B}]$, respectively.

3. Choose bases $\{P_{A'}, Q_{A'}\}$ and $\{P_{B'}, Q_{B'}\}$, which generate $E'[\ell_{A'}^{e_{A'}}]$ and $E'[\ell_{B'}^{e_{B'}}]$, respectively.

4. Let $H_1, H_2 : \{0,1\}^* \to \{0,1\}^k$ be independent secure hash functions (with parameter k).

Key Generation:

1. Alice chooses random integers $m_A, n_A \in \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ not divisible by ℓ_A and $m'_A, n'_A \in \mathbb{Z}/\ell_{A'}^{e_{A'}}\mathbb{Z}$ not divisible by $\ell_{A'}$. Then, using these values, computes $\phi_A \colon E \to E_A = E/\langle [m_A]P_A + [n_A]Q_A \rangle$ and $\phi'_A \colon E' \to E'_A = E'/\langle [m'_A]P_{A'} + [n'_A]Q_{A'} \rangle$. Then, she computes $\phi_A(P_B), \phi_A(Q_B), \phi'_A(P_{B'}), \phi'_A(Q_{B'})$ and publishes her public key as:
$\{E_A, E'_A, \phi_A(P_B), \phi_A(Q_B), \phi'_A(P_{B'}), \phi'_A(Q_{B'})\}$.
Her private key is $\{m_A, n_A, m'_A, n'_A\}$.

2. Bob chooses random integers $m_B, n_B \in \mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$ not divisible by ℓ_B and $m'_B, n'_B \in \mathbb{Z}/\ell_{B'}^{e_{B'}}\mathbb{Z}$ not divisible by $\ell_{B'}$. Then, using these values, computes $\phi_B \colon E \to E_B = E/\langle [m_B]P_B + [n_B]Q_B \rangle$ and $\phi'_B \colon E' \to E'_B = E'/\langle [m'_B]P_{B'} + [n'_B]Q_{B'} \rangle$. Then, he computes $\phi_B(P_A), \phi_B(Q_A), \phi'_B(P_{A'}), \phi'_B(Q_{A'})$ and publishes his public key as:
$\{E_B, E'_B, \phi_B(P_A), \phi_B(Q_A), \phi'_B(P_{A'}), \phi'_B(Q_{A'})\}$.
His private key is $\{m_B, n_B, m'_B, n'_B\}$.

Encryption: Suppose Bob wants to send a message $m \in \{0,1\}^*$ to Alice.

1. Using Alice's public parameters $\{E_A, \phi_A(P_B), \phi_A(Q_B)\}$ and his private key, Bob computes $E_{AB} = E_A/\langle [m_B]\phi_A(P_B) + [n_B]\phi_A(Q_B) \rangle$. Then he computes the j-invariant of E_{AB}, $j(E_{AB})$.

2. Using the j-invariant and the key to the symmetric encryption scheme, Bob encrypts the message and obtains $c = \mathcal{E}(j(E_{AB}), m)$.

3. Using Alice's public parameters $\{E'_A, \phi'_A(P_{B'}), \phi'_A(Q_{B'})\}$ and his private key, Bob computes $E'_{AB} = E'_A/\langle [m'_B]\phi'_A(P_{B'}) + [n'_B]\phi'_A(Q_{B'}) \rangle$. Then he computes the j-invariant of E'_{AB}, $j(E'_{AB})$.

4. Bob select $r \in \{0,1\}^k$ at random.

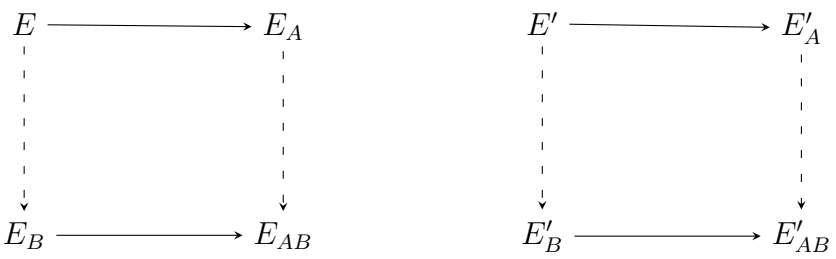

Figure 6.2: Isogenies in Authenticated Encryption Scheme

5. Bob computes the value $h = H_1(c, r)$.
6. Using h and $j(E'_{AB})$, Bob computes the authentication tag $\sigma = H_2(h||j(E'_{AB}))$.
7. The ciphertext is $\{c, r, \sigma\}$.

Decryption: Suppose Alice receives ciphertext $\{c, r, \sigma\}$ from Bob.

1. Using Bob's public parameters $\{E'_B, \phi'_B(P_{A'}), \phi'_B(Q_{A'})\}$ and her private key, Alice computes $E'_{AB} = E'_B/\langle [m'_A]\phi'_B(P_{A'}) + [n'_A]\phi'_B(Q_{A'})\rangle$. Then she computes the j-invariant of E'_{AB}, $j(E'_{AB})$.
2. Alice computes the value $h = H_1(c, r)$.
3. Using h and $j(E'_{AB})$, Alice computes $H_2(h||j(E'_{AB}))$ and compares it to the authentication tag σ. If it matches, she continues, if not, stops.
4. Using Bob's public parameters $\{E_B, \phi_B(P_A), \phi_B(Q_A)\}$ and her private key, Alice computes $E_{AB} = E_B/\langle [m_A]\phi_B(P_A)+[n_A]\phi_B(Q_A)\rangle$. Then she computes the j-invariant of E_{AB}, $j(E_{AB})$.
5. Using the j-invariant and the key to the symmetric encryption scheme, decrypts the message and obtains $m = \mathcal{D}(j(E_{AB}), c)$.

Figure 6.2 depicts the scheme, where we edges represent isogenies, solid ones are known to Alice, dashed are known to Bob.

Remark 6.6.1. For the ease of presentation, we did not include the steps related to mapping messages to randomness, using $\mathcal{Q}$.

We discuss the security of our scheme. Our scheme uses elliptic curve isogenies in the same manner as [JDF11]. Thus, under the same security assumptions, namely SSCDH, SSDDH, CSSI, and DSSI, we see that this approach is quantum-secure. For the encryption part, we are using a classical symmetric-key encryption scheme, which is believed to be IND-CPA secure against quantum attacks. Assuming that the symmetric-key encryption scheme is actually secure, we can then achieve IND-qCPA security against quantum adversaries (doubling the key size if necessary to fend off quantum brute-force attacks). As previously mentioned, AES is believed to be a suitable such scheme. For the authentication part, we used an SDVS scheme, which we transformed to be SUF-qCMA secure. Finally, by Theorem 6.3.8, we conclude that the resulting scheme is an IND-qCCA and INT-qCTXT secure authenticated encryption scheme.

6.7 Overhead Calculations and Comparisons

In this section we study the communication costs of our AE scheme, from the point of view of both per-message communication overhead and key transmission overhead.

6.7.1 Communication Overhead

Recall that the ciphertext which Bob sends to Alice consists of the triplet (c, r, σ), where c is the underlying ciphertext content, r is a k-bit nonce, and σ is the signature tag. In the case where the verification function in the signature scheme involves independently deriving the value of σ, we can hash σ down to k bits as well. For a security level of ℓ bits, the minimum value of k required for collision resistance is 2ℓ bits in the quantum setting [Ber09]. The per-message communication overhead of the scheme is thus 4ℓ bits in the case where the signature tag can be hashed, and $2\ell + |\sigma|$ bits otherwise. Note that in the former case the per-message communications overhead is always the same, independent of which component schemes are chosen.

6.7.2 Public Key Overhead

For the overhead involved in transmitting the public keys to be used for the signature scheme, we use the table of Fujioka et al. [FSXY13], augmented with some more recent results as described below. Although [FSXY13] deals with the case of post-quantum authenticated key exchange, the same key sizes apply to the AE setting.

With the exception of Ring-LWE as explained below, we aim for 128-bit quantum security. For Ring-LWE, we use the numbers from [GLP15]. Since the scheme in [GLP15]

Signature scheme	**Bits**
Ring-LWE (80-bit security) [GLP15]	11600
Ring-LWE (256-bit security) [GLP15]	25000
NTRU [SWZ15]	5544
Code-based [FSXY13]	52320
Multivariate polynomials [HLY12] (via [FSXY13])	7672000
Isogeny-based [AJK$^+$16]	3073

Table 6.1: Key transmission overhead

is based on power-of-2 cyclotomic rings, there is a large jump in parameter size between $n = 2^9$ and $n = 2^{10}$, with the former providing 80 bits of security and the latter 256 bits of security. There is no intermediate power of 2 that would provide 128 bits of security. For this reason, we list both 80-bit and 256-bit security levels in our table. The numbers for NTRU are from Schanck et al. [SWZ15]. For isogeny-based SDVS schemes we use the recent results of [AJK$^+$16]. Note that SDVS schemes require two-way transmission of public keys even if the encrypted communication is one-way, whereas standard signature schemes require two-way transmission of public keys only for two-way communication.

Chapter 7

Future Work

There are a number of possible directions for further research. Our schemes admit efficient implementations in the sense that the running time is polynomial. However, they are still much slower than traditional schemes such as ECC (which are safe only against classical adversaries), as well as certain high-performance quantum-resistant schemes such as NTRU. There is always a security vs. efficiency trade-off, but we are nevertheless interested in speeding up implementations to the extent that we can. Some low-hanging fruit may be available in this regard thanks to the existing literature of known optimizations for elliptic curve cryptography and elliptic curve arithmetic. For example, existing results on addition chains could be used to speed up isogeny evaluation.

It is standard in cryptography to cryptanalyze both the details of the proposed protocols as well as the hardness of the underlying mathematical problem upon which the protocols are based. Further work in both of these areas is critical in order to build up confidence in the schemes among the wider cryptologic research community.

We seek to construct standard (i.e. non-interactive) digital signature schemes using isogenies. Such schemes are a fundamental requirement for internet security today. Currently the best we can do is to apply generic transformations which convert interactive authentication protocols into non-interactive digital signature schemes. These transformations work, but they are slightly inefficient, introducing a polynomial factor of overhead in both computations and communications. A direct construction of an efficient digital signature scheme would be very helpful in order to help make the case for isogeny-based cryptography.

Another direction is to work on other definitions of security for post-quantum cryptography. As observed by Boneh and Zhandry in [BZ13b] and by us in Chapter 6, a number of definitions cannot be directly taken from the classical works and simply used without modification in quantum settings. It is important to review all existing cryptographic secu-

rity definitions and analyze whether they are suitable for quantum adversaries, modifying them as necessary.

Aside from encryption and digital signatures, the third main pillar of internet security is authenticated key exchange (AKE). Developing an AKE scheme entails two tasks. First, we need a quantum-aware security model for AKE. Second, we need to develop an actual candidate for a post-quantum AKE scheme. Our work on authenticated encryption serves as a good foundation for developing security models for AKE.

We have in this thesis laid the initial foundation for post-quantum cryptography based on supersingular elliptic curve isogenies. This area has great potential and we hope that the wider cryptographic community will express an interest and perform more research in this direction.

Bibliography

[AJK+16] Reza Azarderakhsh, David Jao, Kassem Kalach, Brian Koziel, and Christopher Leonardi, *Key compression for isogeny-based cryptosystems*, Cryptology ePrint Archive, Report 2016/229, 2016, to appear in AsiaPKC 2016. 78

[AMBGS13] Carlos Aguilar-Melchor, Slim Bettaieb, Philippe Gaborit, and Julien Schrek, *A code-based undeniable signature scheme*, Cryptography and Coding (Martijn Stam, ed.), Lecture Notes in Computer Science, vol. 8308, Springer Berlin Heidelberg, 2013, pp. 99–119. 49, 59

[ATTU16] Mayuresh Vivekanand Anand, Ehsan Ebrahimi Targhi, Gelo Noel Tabia, and Dominique Unruh, *Post-quantum security of the CBC, CFB, OFB, CTR, and XTS modes of operation*, in Takagi [Tak16], pp. 44–63. 73

[Bac90] Eric Bach, *Explicit bounds for primality testing and related problems*, Math. Comp. **55** (1990), no. 191, 355–380. 35

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani, *Strengths and weaknesses of quantum computing*, SIAM J. Comput. **26** (1997), 1510–1523. 33

[BCL08] Reinier Bröker, Denis Xavier Charles, and Kristin 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. 24, 25, 30, 31

[BCNS14] Joppe W. Bos, Craig Costello, Michael Naehrig, and Douglas Stebila, *Post-quantum key exchange for the TLS protocol from the ring learning with errors problem*, Cryptology ePrint Archive, Report 2014/599, 2014, `http://eprint.iacr.org/`. 61

[Bel08] Juliana V. Belding, *Number theoretic algorithms for elliptic curves*, Ph.D. thesis, University of Maryland, 2008. 47

[Ber] Daniel J. Bernstein, *How to find smooth parts of integers*, URL: `http://cr.yp.to/papers.html#smoothparts`. Note: draft. 29

[Ber09] Daniel J. Bernstein, *Cost analysis of hash collisions: will quantum computers make SHARCS obsolete?*, Workshop Record of SHARCS'09: Special-purpose Hardware for Attacking Cryptographic Systems, 2009, pp. 51–82. 77

[Bis11] Gaetan Bisson, *Computing endomorphism rings of elliptic curves under the GRH*, Journal of Mathematical Cryptology **5** (2011), no. 2, 101–113, `arXiv:1101.4323v2 [math.NT]`. 35

[BN08] Mihir Bellare and Chanathip Namprempre, *Authenticated encryption: Relations among notions and analysis of the generic composition paradigm*, J. Cryptol. **21** (2008), no. 4, 469–491. 3, 62, 64, 65, 66, 67, 68, 69

[BR06] Mihir Bellare and Phillip Rogaway, *The security of triple encryption and a framework for code-based game-playing proofs*, Proceedings of the 24th Annual International Conference on The Theory and Applications of Cryptographic Techniques (Berlin, Heidelberg), EUROCRYPT'06, Springer-Verlag, 2006, pp. 409–426. 68

[BS11] Gaetan Bisson and Andrew V. Sutherland, *Computing the endomorphism ring of an ordinary elliptic curve over a finite field*, Journal of Number Theory **131** (2011), no. 5, 815 – 831, Elliptic Curve Cryptography. 25

[BV07] Johannes Buchmann and Ulrich Vollmer, *Binary quadratic forms*, Algorithms and Computation in Mathematics, vol. 20, Springer, Berlin, 2007, An algorithmic approach. 25, 27, 28, 29, 30, 31, 34

[BZ13a] Dan Boneh and Mark Zhandry, *Quantum-secure message authentication codes*, Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings (Thomas Johansson and Phong Q. Nguyen, eds.), Lecture Notes in Computer Science, vol. 7881, Springer, 2013, pp. 592–608. 64, 71

[BZ13b] Dan Boneh and Mark Zhandry, *Secure signatures and chosen ciphertext security in a quantum computing world*, Proc. of Crypto, LNCS, vol. 8043, 2013, pp. 361–379. 62, 63, 64, 71, 72, 79

[CJS14] Andrew M. Childs, David Jao, and Vladimir Soukharev, *Constructing elliptic curve isogenies in quantum subexponential time*, J. Mathematical Cryptology **8** (2014), no. 1, 1–29. 2, 24, 37, 47

[CK01] Ran Canetti and Hugo Krawczyk, *Analysis of key-exchange protocols and their use for building secure channels*, EUROCRYPT (Birgit Pfitzmann, ed.), Lecture Notes in Computer Science, vol. 2045, Springer, 2001, pp. 453–474. 48

[CLG09] Denis Xavier Charles, Kristin E. Lauter, and Eyal Z. Goren, *Cryptographic hash functions from expander graphs*, Journal of Cryptology **22** (2009), 93–113. 46

[CM01] Kevin K. H. Cheung and Michele Mosca, *Decomposing finite abelian groups*, Quantum Inform. Comput. **1** (2001), no. 3, 26–32, `arXiv:cs/0101004v1 [cs.DS]`. 35

[Cn04] Juan M. Cerviño, *On the correspondence between supersingular elliptic curves and maximal quaternionic orders*, April 2004, `http://arxiv.org/abs/math/0404538`. 47

[Cou06] Jean-Marc Couveignes, *Hard homogeneous spaces*, 2006, `http://eprint.iacr.org/2006/291`. 34

[Cox89] David 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. 16, 17, 19, 26

[DF10] Luca De Feo, *Fast algorithms for computing isogenies between ordinary elliptic curves in small characteristic*, Journal of Number Theory **131** (2010), no. 5, 873–893. 32

[DFJP14] Luca De Feo, David Jao, and Jérôme Plût, *Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies*, J. Mathematical Cryptology **8** (2014), no. 3, 209–247. 3, 37, 49, 55, 59, 60

[DHI02] Wim van Dam, Sean Hallgren, and Lawrence Ip, *Quantum algorithms for some hidden shift problems*, SODA '02: Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms, 2002, `arXiv:quant-ph/0211140v1`, pp. 489–498. 33

[DSV03] Giuliana Davidoff, Peter Sarnak, and Alain Valette, *Elementary number theory, group theory, and Ramanujan graphs*, London Mathematical Society Student Texts, vol. 55, Cambridge University Press, Cambridge, 2003. 38

[EH00] Mark Ettinger and Peter Høyer, *On quantum algorithms for noncommutative hidden subgroups*, Advances in Applied Mathematics **25** (2000), 239–251. 33

[Eng09] Andreas Enge, *Computing modular polynomials in quasi-linear time*, Math. Comp. **78** (2009), no. 267, 1809–1824. 25

[FFS88] Uriel Feige, Amos Fiat, and Adi Shamir, *Zero-knowledge proofs of identity*, Journal of Cryptology **1** (1988), no. 2, 77–94. 56

[Fis14] Dieter Fishbein, *Machine-level software optimization of cryptographic protocols*, Master's thesis, University of Waterloo, 2014, `http://hdl.handle.net/10012/8400`. 60

[FM02] Mirelle Fouquet and François 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. 18, 19

[FSXY13] Atsushi Fujioka, Koutarou Suzuki, Keita Xagawa, and Kazuki Yoneyama, *Practical and post-quantum authenticated key exchange from one-way secure key encapsulation mechanism*, Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security (New York, NY, USA), ASIA CCS '13, ACM, 2013, pp. 83–94. 61, 77, 78

[Gal99] Steven D. Galbraith, *Constructing isogenies between elliptic curves over finite fields*, LMS J. Comput. Math. **2** (1999), 118–138 (electronic). 18, 19, 46, 56

[GHS02] Steven D. Galbraith, Florian Hess, and Nigel 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. 25, 37

[GLP15] Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann, *Lattice-based signatures: Optimization and implementation on reconfigurable hardware*, IEEE Trans. Computers **64** (2015), no. 7, 1954–1967. 72, 77, 78

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson, *Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems*, Journal of the Association for Computing Machinery **38** (1991), no. 3, 690–728. 40, 48, 56

[GS11] Steven D. Galbraith and Anton Stolbunov, *Improved algorithm for the isogeny problem for ordinary elliptic curves*, 2011, `http://arxiv.org/abs/1105.6331/`. 37

[HLY12] Yun-Ju Huang, Feng-Hao Liu, and Bo-Yin Yang, *Public-key cryptography from new multivariate quadratic assumptions*, Public Key Cryptography —

PKC 2012 (Marc Fischlin, Johannes Buchmann, and Mark Manulis, eds.), Lecture Notes in Computer Science, vol. 7293, Springer Berlin Heidelberg, 2012, pp. 190–205 (English). 78

[HM89] James L. Hafner and Kevin S. McCurley, *A rigorous subexponential algorithm for computation of class groups*, J. Amer. Math. Soc. **2** (1989), no. 4, 837–850. 25

[IJ10] Sorina Ionica and Antoine Joux, *Pairing the volcano*, Algorithmic Number Theory (Guillaume Hanrot, François Morain, and Emmanuel Thomé, eds.), Lecture Notes in Comput. Sci., vol. 6197, Springer–Verlag, 2010, 9th International Symposium, Nancy, France, ANTS-IX, July 19-23, 2010, Proceedings. 32

[JDF11] David Jao and Luca De Feo, *Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies*, PQCrypto (Bo-Yin Yang, ed.), Lecture Notes in Computer Science, vol. 7071, Springer, 2011, pp. 19–34. 3, 37, 48, 49, 55, 59, 72, 74, 77

[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. 26, 27, 38

[JS10] David Jao and Vladimir Soukharev, *A subexponential algorithm for evaluating large degree isogenies*, Algorithmic number theory, Lecture Notes in Comput. Sci., vol. 6197, Springer, Berlin, 2010, pp. 219–233. 2, 25, 26, 27

[JS14] David Jao and Vladimir Soukharev, *Isogeny-based quantum-resistant undeniable signatures*, Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings (Michele Mosca, ed.), Lecture Notes in Computer Science, vol. 8772, Springer, 2014, pp. 160–179. 3, 49

[JSI96] Markus Jakobsson, Kazue Sako, and Russell Impagliazzo, *Designated verifier proofs and their applications*, Advances in Cryptology — EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding (Ueli M. Maurer, ed.), Lecture Notes in Computer Science, vol. 1070, Springer, 1996, pp. 143–154. 71

[KF08] Kaoru Kurosawa and Jun Furukawa, *Universally composable undeniable signature*, Proceedings of the 35th International Colloquium on Automata, Lan-

guages and Programming, Part II (Berlin, Heidelberg), ICALP '08, Springer-Verlag, 2008, pp. 524–535. 50

[Koh96] David Kohel, *Endomorphism rings of elliptic curves over finite fields*, Ph.D. thesis, University of California, Berkeley, 1996. 47

[Kup05] Greg Kuperberg, *A subexponential-time quantum algorithm for the dihedral hidden subgroup problem*, SIAM J. Comput. **35** (2005), no. 1, 170–188. 33, 34

[Lan87] Serge Lang, *Elliptic functions*, second ed., Graduate Texts in Mathematics, vol. 112, Springer-Verlag, New York, 1987, With an appendix by J. Tate. 21

[LPS88] Alexander Lubotzky, Ralph Phillips, and Peter Sarnak, *Ramanujan graphs*, Combinatorica **8** (1988), no. 3, 261–277. 38

[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. 32

[Lub94] Alexander Lubotzky, *Discrete groups, expanding graphs and invariant measures*, Progress in Mathematics, vol. 125, Birkhäuser Verlag, Basel, 1994, With an appendix by Jonathan D. Rogawski. 38

[Mes86] Jean-François Mestre, *La méthode des graphes. Exemples et applications*, Proceedings of the international conference on class numbers and fundamental units of algebraic number fields (Katata, 1986) (Nagoya), Nagoya Univ., 1986, pp. 217–242. 39

[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. 16

[MR14] Diana Maimut and Reza Reyhanitabar, *Authenticated encryption: Toward next-generation algorithms*, IEEE Security & Privacy **12** (2014), no. 2, 70–72. 61

[Piz90] Arnold K. Pizer, *Ramanujan graphs and Hecke operators*, Bull. Amer. Math. Soc. (N.S.) **23** (1990), no. 1, 127–137. 39

[Piz98] Arnold K. Pizer, *Ramanujan graphs*, Computational perspectives on number theory (Chicago, IL, 1995), AMS/IP Stud. Adv. Math., vol. 7, Amer. Math. Soc., Providence, RI, 1998, pp. 159–178. 39

[PLQ08] Christophe Petit, Kristin Lauter, and Jean-Jacques Quisquater, *Full cryptanalysis of LPS and Morgenstern hash functions*, Proceedings of the 6th international conference on Security and Cryptography for Networks (Berlin, Heidelberg), SCN '08, Springer-Verlag, 2008, pp. 263–277. 46

[Poi95] David Pointcheval, *A new identification scheme based on the perceptrons problem*, Advances in Cryptology EUROCRYPT '95 (Berlin, Heidelberg), Lecture Notes in Computer Science, vol. 921, Springer Berlin / Heidelberg, 1995, pp. 319–328. 48

[Reg] Oded Regev, *A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space*, `arXiv:quant-ph/0406151v1`. 34

[RS06] Alexander Rostovtsev and Anton Stolbunov, *Public-key cryptosystem based on isogenies*, 2006, `http://eprint.iacr.org/2006/145`. 34, 39

[Sar90] Peter Sarnak, *Some applications of modular forms*, Cambridge Tracts in Mathematics, vol. 99, Cambridge University Press, Cambridge, 1990. 38

[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. 29

[Sch95] Réne 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). 15, 34

[Sey87] Martin Seysen, *A probabilistic factorization algorithm with quadratic forms of negative discriminant*, Math. Comp. **48** (1987), no. 178, 757–780. 29

[Sha89] Adi Shamir, *An efficient identification scheme based on permuted kernels (extended abstract)*, Proceedings on Advances in cryptology (New York, NY, USA), CRYPTO '89, Springer-Verlag New York, Inc., 1989, pp. 606–609. 48

[Sho97] Peter W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. Comput. **26** (1997), no. 5, 1484–1509, Preliminary version in FOCS '94. `arXiv:quant-ph/9508027v2`. 1, 34, 35

[Sho01] Victor Shoup, *OAEP reconsidered*, Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings (Joe Kilian, ed.), Lecture Notes in Computer Science, vol. 2139, Springer, 2001, pp. 239–259. 68, 69

[Sil92] Joseph H. Silverman, *The arithmetic of elliptic curves*, Graduate Texts in Mathematics, vol. 106, Springer-Verlag, New York, 1992, Corrected reprint of the 1986 original. 4, 9, 10, 11, 12, 14, 15, 17, 20, 39

[Sil94] Joseph H. Silverman, *Advanced topics in the arithmetic of elliptic curves*, Graduate Texts in Mathematics, vol. 151, Springer-Verlag, New York, 1994. 21

[SJS16] Vladimir Soukharev, David Jao, and Srinath Seshadri, *Post-quantum security models for authenticated encryption*, Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016, Proceedings, 2016, pp. 64–78. 3, 61

[Son14] Fang Song, *A note on quantum security for post-quantum cryptography*, Cryptology ePrint Archive, Report 2014/709, 2014, `http://eprint.iacr.org/`. 72

[Sou10] Vladimir Soukharev, *Evaluating large degree isogenies between elliptic curves*, Master's thesis, University of Waterloo, 2010, `http://hdl.handle.net/10012/5674`. 2, 24

[Ste94a] Jacques Stern, *Designing identification schemes with keys of short size*, Advances in Cryptology CRYPTO '94 (Berlin, Heidelberg), Lecture Notes in Computer Science, vol. 839, Springer Berlin / Heidelberg, 1994, pp. 164–173. 48

[Ste94b] Jacques Stern, *A new identification scheme based on syndrome decoding*, Advances in Cryptology CRYPTO' 93 (Berlin, Heidelberg), Lecture Notes in Computer Science, vol. 773, Springer Berlin / Heidelberg, 1994, pp. 13–21. 48

[Sto10] Anton Stolbunov, *Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves*, Adv. Math. Commun. **4** (2010), no. 2, 215–235. 2, 22, 23, 33, 34, 37, 39, 42, 43, 48

[STW12] Xi Sun, Haibo Tian, and Yumin Wang, *Toward quantum-resistant strong designated verifier signature from isogenies.*, INCoS (Fatos Xhafa, Leonard Barolli, Florin Pop, Xiaofeng Chen 0001, and Valentin Cristea, eds.), IEEE, 2012, pp. 292–296. 72

[SWZ15] John Schanck, William Whyte, and Zhenfei Zhang, *A quantum-safe circuit-extension handshake for tor*, Cryptology ePrint Archive, Report 2015/287, 2015, `http://eprint.iacr.org/`. 61, 78

[Tak16] Tsuyoshi Takagi (ed.), *Post-quantum cryptography - 7th international workshop, pqcrypto 2016, fukuoka, japan, february 24-26, 2016, proceedings*, Lecture Notes in Computer Science, vol. 9606, Springer, 2016. 81

[Tan08] Seiichiro Tani, *Claw Finding Algorithms Using Quantum Walk*, `http://arxiv.org/abs/0708.2584`, March 2008. 47

[Tat66] John Tate, *Endomorphisms of abelian varieties over finite fields*, Invent. Math. **2** (1966), 134–144. 15

[Tes99] Edlyn Teske, *The pohlig-hellman method generalized for group structure computation*, Journal of Symbolic Computation **27** (1999), no. 6, 521–534. 44, 47

[Vél71] Jacques Vélu, *Isogénies entre courbes elliptiques*, C. R. Acad. Sci. Paris Sér. A-B **273** (1971), A238–A241. 32

[Wat69] William C. Waterhouse, *Abelian varieties over finite fields*, Ann. Sci. École Norm. Sup. (4) **2** (1969), 521–560. 21, 22, 33

[Zha05] Shengyu Zhang, *Promised and Distributed Quantum Search Computing and Combinatorics*, Proceedings of the Eleventh Annual International Conference on Computing and Combinatorics (Berlin, Heidelberg), Lecture Notes in Computer Science, vol. 3595, Springer Berlin / Heidelberg, 2005, pp. 430–439. 47