Key Compression for Isogeny-Based Cryptosystems

by

Christopher Leonardi

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

Waterloo, Ontario, Canada, 2016

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the 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.

Statement of Contributions

I am the sole author of chapters $1, 2$ and 7. Chapter 3 contains some material from "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies" by L. De Feo, D. Jao, and J. Plût. Chapters $4, 5$, and 6 contains material from "Key compression for isogeny-based cryptography" by R. Azarderakhsh, D. Jao, K. Kalach, B. Koziel, and C. Leonardi, to appear in AsiaPKC 2016. I am the sole author of all parts of that paper that appear in this thesis.

Abstract

We present a method for key compression in quantum-resistant isogeny-based cryptosystems, which reduces storage and transmission costs of per-party public information by a factor of two, with no effect on the security level of the scheme. We achieve this reduction by compressing both the representation of an elliptic curve, and torsion points on said curve. Compression of the elliptic curve is achieved by associating each j-invariant to a canonical choice of elliptic curve, and the torsion points will be represented as linear combinations with respect to a canonical choice of basis for this subgroup. This method of compressing public information can be applied to numerous isogeny-based protocols, such as key exchange, zero-knowledge identification, and public-key encryption. The details of utilizing compression for each of these cryptosystems is explained. We provide implementation results showing the computational cost of key compression and decompression at various security levels. Our results show that isogeny-based cryptosystems achieve the smallest possible key sizes among all existing families of post-quantum cryptosystems at practical security levels.

Acknowledgements

I would like to thank my supervisor David Jao for being an excellent mentor throughout my Master's program, and Kassem Kalach for his valuable advice and expertise. I am very grateful to the members of the Combinatorics and Optimization department administration for all their help over the past two years, and my readers Alfred Menezes and Michele Mosca. Finally, thank you to my family for their constant support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Traditional elliptic curve cryptography is based on the intractability of the elliptic curve discrete logarithm problem. These systems are not safe to use in a post-quantum setting, since on a quantum computer Shor's algorithm [32] can compute solutions to the discrete logarithm problem in any group (including elliptic curve groups) in polynomial time.

Couveignes [11] and Stolbunov [37] independently discovered an encryption scheme from the theory of elliptic curves that does not rely on the discrete logarithm problem. Instead, their cryptosystems rely on the computational difficulty of finding isogenies between ordinary elliptic curves. Soon after it was shown that one could determine the private keys of this system in subexponential time with a quantum attack [7]. Recent work of De Feo, Jao, and Plût [14] proposes to use isogenies between supersingular elliptic curves as the basis for quantum-safe elliptic curve cryptosystems. Unlike with discrete logarithms, there is no known polynomial time algorithm to compute isogenies between elliptic curves in the general case, even on a quantum computer [2, 7]. Implementation results [15] have shown that isogeny-based cryptosystems exhibit practical performance characteristics at standard security levels. Isogeny-based cryptography, though not yet considered mainstream, thus represents a promising candidate for post-quantum security.

We provide an explicit method for reducing the transmission cost and per-party public information in isogeny-based cryptosystems. The proposed compression of public information in this work is modeled for the existing key exchange and public-key cryptosystems of [14]. The reduction takes advantage of algebraic properties of elliptic curves, while only incurring the computational costs of compression and decompression once per key. The algorithm compresses keys to half their original size, with no effect on security.

Chapter 2 is an introduction to elliptic curves, including the different coordinate systems one can use, the morphisms of elliptic curves, torsion subgroups, the difference between ordinary and supersingular elliptic curves, and the required material from the theory of Hilbert class fields. This chapter also includes algorithms for computing and evaluating isogenies, and constructing supersingular elliptic curves. The history and current standing of isogeny-based cryptography (both ordinary and supersingular) is given in Chapter 3. Chapter 4 contains the steps for compression of public keys, as well as an explanation of all computational requirements for this procedure. Chapter 5 modifies the standard isogeny-based key-exchange, zero-knowledge identification, and public-key encryption protocols to include compression and decompression. A presentation of the time complexity, empirical cost measurements for our implementation, and a comparison of our compressed key sizes with those of other major families of post-quantum cryptographic primitives is given in Chapter 6. Lastly, Chapter 7 covers the applications of the results and possible future improvements or changes.

Chapter 2

Introduction to Elliptic Curves

This opening chapter serves as an introduction to the theory of elliptic curves and the multitude of maps between them. A goal of this thesis is to be self-contained, and so all required background material will be presented. The materials in sections 2.1 and 2.2 of this chapter are collected primarily from the texts [9], [10], [17], and [33].

2.1 Coordinates and the Group Law

The central objects of isogeny-based cryptography, and the work of this thesis, are elliptic curves. The definition we will begin with is of an elliptic curve in **projective coordinates**.

Definition 2.1.1. *An* ***elliptic curve*** *E over a field K, denoted E/K, is given by the non-singular projective curve of the form*

$$E : Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3, \tag{2.1}$$

for $a_1, a_2, a_3, a_4, a_6 \in K$, along with a base point $O = O(E) = [0, 1, 0]$ which is referred to as the ***point at infinity****.*

From the coefficients of an elliptic curve in (2.1), we can define the b-invariants and the

c-invariants of a curve to simplify later algebra:

$$\begin{aligned}
b_2 &= a_1^2 + 4a_4,\\
b_4 &= 2a_4 + a_1a_3,\\
b_6 &= a_3^2 + 4a_6,\\
c_4 &= b_2^2 - 24b_4,\\
c_6 &= -b_2^3 + 36b_2b_4 - 216b_6.
\end{aligned}$$

Throughout this work we will be using the Weierstraß equations for an elliptic curve over K instead of projective coordinates. The **long Weierstraß form** can be obtained from (2.1) by the change of coordinates $x = X/Z$ and $y = Y/Z$:

Definition 2.1.2. *An elliptic curve E over a field K is given by the equation*

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \tag{2.2}$$

where the coefficients $a_1, a_2, a_3, a_4, a_6 \in K$ are such that for each point $P = (x_1, y_1)$ with coordinates in $\bar{K}$ satisfying (2.2), the partial derivatives at P ($2y_1 + a_1x_1 + a_3$ and $3x_1^2 + 2a_2x_1 + a_4 - a_1y_1$) do not vanish simultaneously.

The last condition in (2.1.2) is equivalent to the non-singular condition of (2.1.1). This change of coordinates requires that $Z \neq 0$. Indeed, a point (x, y) in Weierstraß coordinates is equivalent to $(x, y, 1)$ in projective coordinates for all points on the curve other than O, which is exactly when $Z = 0$. When the characteristic of the field K is not 2 we can apply a change of coordinates to eliminate the xy and y terms:

$$E : y^2 = 4x^3 + b_2x^2 + 2b_4 + b_6. \tag{2.3}$$

Further, if $char(K) \neq 3$, then we can apply one last substitution to obtain the **short Weierstraß form**:

$$E : y^2 = x^3 - 27c_4x - 54c_6. \tag{2.4}$$

A useful invariant of elliptic curves is their discriminant. The cubic polynomial has only simple roots over $\bar{K}$ if and only if the discriminant is non-zero. Therefore, checking the non-singularity condition (or checking that an equation does in fact define an elliptic curve) can be done by computing the discriminant from the coefficients.

Definition 2.1.3. *Let E be a curve as defined in (2.1), and the b-invariants be as above. The* ***discriminant*** *of the curve E is*

$$\Delta = -b_2^2b_8 - 8b_4^3 - 27b_6^2 + 9b_2b_4b_6,$$

where $b_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2$.

Additionally, if E is an elliptic curve in short Weierstraß form (2.4), $E : y^2 = x^3 + ax + b$, then we can compute the discriminant as follows:

$$\Delta(E) = -16(4a^3 + 27b^2).$$

Hence, an equation in short Weierstraß form defines an elliptic curve if and only if

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

There is a well known geometrically constructed law that provides elliptic curves with a natural group structure. A solution $(x, y) \in K \times K$ to the defining equation of E is called a K-rational point of E. The set of all K-rational points, along with the point at infinity, form an Abelian group denoted $E(K)$. The group law can be expressed by rational polynomials with coefficients in K, and it enables us to add and subtract points, as well as multiply a K-rational point by an integer efficiently using the standard double-and-add technique.

2.1.1 Torsion Subgroups

Definition 2.1.4. *Let E be an elliptic curve over the field K, and let $P \in E(\bar{K})$. For any $m \in \mathbb{Z}$ define the map $[m] : E(\bar{K}) \to E(\bar{K}), [m]P = P + \ldots + P$, to be the* ***multiplication-by-m map****.*

Example 1. *The map $[1]$ is the identity map. The map $[0]$ is defined so that $[0]P = O$ for all $P \in E(\bar{K})$.*

The multiplication map is defined for negative integers in the following way: $[-m]P = [m](-P)$, where $-P$ is the group inverse of P.

Definition 2.1.5. *For any $m \in \mathbb{Z}$, and elliptic curve $E(K)$, the subgroup*

$$E[m] := \{P \in E(\bar{K}) : [m]P = \infty\}$$

is called the ***m-torsion subgroup*** *of $E(\bar{K})$. An element $P \in E[m]$ is called an* ***m-torsion point****. We will denote $E[m] \cap E(\bar{K})$ by $E(\bar{K})[m]$.*

This subgroup can be viewed as the kernel of $[m]$. That is, points in $E[m]$ all have order dividing m.

Theorem 2.1.1. [41, *Theorem* 3.2] *Let E be an elliptic curve defined over K. If char(K) is 0 or coprime to m, then*

$$E[m] \cong \mathbb{Z}/m\mathbb{Z} \oplus \mathbb{Z}/m\mathbb{Z}.$$

The case where the characteristic of K is some prime p and $m = p^r$ is covered in Section 2.2.3.

2.2 Maps Between Curves

In this section we will focus on the algebraic relationships between elliptic curves.

2.2.1 Morphisms

The material in this section is well known, but two quality sources are [17], and [36]. We start with the most elementary maps.

Definition 2.2.1. *Let E and E' be two elliptic curves defined over K. A* ***morphism*** *$\phi : E(\bar{K}) \to E'(\bar{K})$ over K is a polynomial mapping with coefficients from K. If the curves are in projective coordinates we can write*

$$\phi(X : Y : Z) = (\phi_0(X, Y, Z) : \phi_1(X, Y, Z) : \phi_2(X, Y, Z)),$$

where ϕ_0, ϕ_1, ϕ_2 are homogeneous polynomials of equal degree satisfying the defining equation of E'. Alternatively, in Weierstraß coordinates, a morphism ϕ is a rational map

$$\phi(x, y) = \left(\frac{\phi_0(x, y, 1)}{\phi_2(x, y, 1)}, \frac{\phi_1(x, y, 1)}{\phi_2(x, y, 1)} \right).$$

A morphism defined over field K is commonly referred to as a K-rational morphism. Each morphism has an integer degree which will be defined later (2.2.11), but for now note that a degree m morphism from E to E' typically implies the kernel of the morphism has cardinality m. That is, the morphism is m-to-1 from $E(\bar{K})$ to $E'(\bar{K})$.

One family of morphisms are the **translation morphisms**. For each point $P \in E$, we can define

$$\tau_P : E \to E, \; \tau_P(Q) = Q + P.$$

These are morphisms because the elliptic curve group law is defined by polynomials.

Definition 2.2.2. *A* ***homomorphism*** ϕ *is a morphism of elliptic curves such that*

$$\phi(P+Q) = \phi(P) + \phi(Q),$$

for all $P, Q \in E(K)$*. That is,* ϕ *respects the group structure of the curve.*

Proposition 2.2.1. *Every morphism from* $E \to E'$ *that maps* $O(E)$ *to* $O(E')$ *is a homomorphism.*

A translation morphism τ_P is not a homomorphism unless $P = O$, in which case it is the trivial homomorphism between E and itself.

Definition 2.2.3. *An* ***endomorphism*** *of an elliptic curve* E *is a homomorphism from* E *to itself. The set of all endomorphisms is denoted* End(E)*.*

The set End(E) is a ring due to the group structure of E. Here we note that for every $m \in \mathbb{Z}$, the multiplication-by-m map is an endomorphism, and the degree of $[m]$ is m^2. Therefore, for an elliptic curve E defined over a finite field End(E) will always contain a subring isomorphic to $\mathbb{Z}$. One of the most important morphisms in the study of elliptic curves over finite fields is the Frobenius endomorphism.

Example 2. *Let* $E/\mathbb{F}_{p^n}$ *be an elliptic curve. Define the* p^n*-th* ***Frobenius endomorphism*** $\pi_E : E(\mathbb{F}_{p^n}) \to E(\mathbb{F}_{p^n})$ *by* $(x, y) \mapsto (x^{p^n}, y^{p^n})$*. The degree of* π_E *is* p^n*.*

It can be shown that if K is finite, then π_E is not equal to $[m]$ for any integer m. This implies that End(E/K) will always contain a *strict* subring isomorphic to $\mathbb{Z}$ when K is finite.

Definition 2.2.4. *A* K*-isomorphism between elliptic curves is a group isomorphism defined over* $\bar{K}$*. In other words, an elliptic curve isomorphism is a* K*-morphism of degree* 1*.*

A useful property of elliptic curves in Weierstraß form is that all isomorphisms between them have been classified.

Proposition 2.2.2. [33, *III*.1] *Elliptic curves* $E/K : y^2 = x^3 + ax + b$ *and* $E'/K : y^2 = x^3 + a'x + b'$ *are isomorphic over* $\bar{K}$ *if and only if there exists* $\mu \in \bar{K}^*$ *such that*

$$\begin{aligned} a' &= \mu^2 a, \\ b' &= \mu^3 b. \end{aligned}$$

If so, the isomorphism $E \to E'$ *is given by* $(x, y) \mapsto (\mu x, \mu^{\frac{3}{2}} y)$*.*

A special case of Proposition 2.2.2 is when μ is a quadratic non-residue in K. In this case the essential element $\mu^{\frac{3}{2}}$ is undefined in K, and so the isomorphism is defined over $K(\sqrt{\mu})$ instead.

Definition 2.2.5. *Let $E/K : y^2 = x^3 + ax + b$ be an elliptic curve and char$(K) \neq 2$. Then for any quadratic non-residue $\mu \in K \backslash \{O\}$ we define the elliptic curve $E^{(\mu)} : y^2 = x^3 + \mu^2 ax + \mu^3 b$ to be the* ***quadratic twist*** *of E by μ.*

One can verify by this definition that the twist of an elliptic curve will give a non-singular equation.

We can now define the isomorphism class of elliptic curves defined over a given field. Further, there exists a unique quantity for each such class that we can use as a label.

Definition 2.2.6. *For an elliptic curve $E : y^2 = x^3 + ax + b$ defined over a field K, define*

$$j(E) = \frac{c_4^3}{\Delta} = 1728\frac{c_4^3}{c_4^3 - c_6^2} = 1728\frac{4a^3}{4a^3 + 27b^2} \in K$$

to be the ***j-invariant*** *of E.*

It is simple to check, using Proposition 2.2.2 that this quantity is invariant for a K-isomorphism class of elliptic curves. The converse is true as well, in the case of algebraically closed fields.

Theorem 2.2.1. [33, $III.1.4$] *Two elliptic curves are isomorphic over $\bar{K}$ if and only if they have the same j-invariant.*

During compression, we will be sending a j-invariant which represents an isomorphism class of elliptic curves. However, in order to send a point from the elliptic curve, or to implement Vélu's formula for constructing an isogeny, we will need an explicit elliptic curve equation. Thankfully, the construction of an elliptic curve for a given j-invariant is known:

Theorem 2.2.2. [33, $III.1.4$] *Given some $j \in K$, the following formulas determine an elliptic curve defined over K whose j-invariant is equal to j:*

(1) If $j = 0$, then $E : y^2 + y = x^3$,

(2) If $j = 1728$, then $E : y^2 = x^3 + x$,

(3) Otherwise, $E : y^2 + xy = x^3 - \frac{36}{j-1728}x - \frac{1}{j-1728}$.

Observe that neither (1) nor (3) yield equations in short Weierstraß form, but since for our applications $p \neq 2, 3$ we can rewrite these curves in the form $y^2 = x^3 - 27c_4x - 54c_6$ using the c-invariants c_4 and c_6.

It is worth noting that for finite fields K, there exist twisted elliptic curves other than quadratic twists (2.2.5). When $j(E) = 0$, one can twist by a cubic character μ and the twisted curve is of the form $y^2 = x^3 + \mu b$. When $j(E) = 1728$ quartic twists are possible and of the form $y^2 = x^3 + \mu a x$ [33, Prop 5.4]. In these cases, the isomorphism will be defined over the appropriate cubic or quartic extension of K.

2.2.2 Isogenies

Definition 2.2.7. *Let K be a field and E/K, E'/K be two elliptic curves. If F is an extension of K (possibly $\bar{K}$ or K itself), define an* ***isogeny over F*** *between E and E' to be a morphism*

$$\phi : E(\bar{K}) \to E'(\bar{K})$$

mapping $O(E)$ to $O(E')$, with coefficients from F. Two elliptic curves are defined to be ***isogenous*** *if and only if there is a non-trivial isogeny between them, that is, $\phi(E) \neq \{O(E')\}$.*

Theorem 2.2.1 immediately shows that every isogeny is necessarily a homomorphism between E and E'.

Example 3. *An isomorphism that preserves the point at infinity is an isogeny. Such isogenies are called* ***pointed isomorphisms****.*

Since pointed isomorphisms are isogenies between isomorphic curves, we can use them to define the notion of an isomorphism of isogenies.

Definition 2.2.8. *Isogenies $\phi_1 : E \to E'$ and $\phi_2 : E \to E''$ are said to be* ***isomorphic isogenies*** *if there exists a pointed isomorphism $\psi : E' \to E''$ such that $\phi_2 = \psi \circ \phi_1$.*

Definition 2.2.9. *The kernel of an isogeny ϕ is*

$$ker(\phi) := \{P | P \in E(\bar{K}) \text{ and } \phi(P) = O\}.$$

Recall, the form of an isogeny (2.2.1) between elliptic curves in Weierstraß form:

$$\phi(x, y) = \left(\frac{\phi_0(x, y, 1)}{\phi_2(x, y, 1)}, \frac{\phi_1(x, y, 1)}{\phi_2(x, y, 1)} \right)$$

such that ϕ fixes the identity. From this it is clear that the kernel of ϕ will be exactly the set of zeros of ϕ_2.

Definition 2.2.10. *The* ***coordinate ring*** *of an elliptic curve E over field K is*

$$K[E] := K[x,y]/\langle y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6\rangle.$$

The ***function field*** *of E over K is the field of fractions of $K[E]$, denoted $K(E)$.*

An early observation is that all morphisms between elliptic curves are either constant or surjective. We use this result to construct an injective homomorphism of function fields from each isogeny. Let $\phi : E \to E'$ be an isogeny, and define $\phi^* : \bar{K}(E') \to \bar{K}(E)$ such that $\phi^*(f) = f \circ \phi$.

Definition 2.2.11. *The* ***degree*** *of a morphism $\phi : E \to E'$ between elliptic curves is*

$$deg(\phi) := [\bar{K}(E) : \phi^*(\bar{K}(E'))],$$

the degree of the function field extension induced by ϕ. A degree ℓ isogeny is often referred to as an ℓ-isogeny.

Proposition 2.2.3. [33, III.6] *Let $\phi : E_1 \to E_2$ be an isogeny. There exists a unique isogeny $\hat{\phi} : E_2 \to E_1$ such that $deg(\phi) = deg(\hat{\phi})$, and the composition of these two isogenies is the multiplication-by-$deg(\phi)$ map. That is, $\phi \circ \hat{\phi} = \hat{\phi} \circ \phi = [deg(\phi)]$. This map is defined as the* ***dual isogeny*** *of ϕ.*

In isogeny-based cryptography, we are interested in separable isogenies over finite fields.

Definition 2.2.12. *Let $\phi : E/K \to E'/K$ be an isogeny between elliptic curves. If $\bar{K}(E)$ is a separable (inseparable, purely inseparable, resp.) field extension of $\phi^*(\bar{K}(E'))$, then we say ϕ is* ***separable*** *(inseparable, purely inseparable, resp.).*

Theorem 2.2.3. *A K-isogeny ϕ is separable if and only if $char(K) \nmid deg(\phi)$.*

The following result shows where our early "definition" of degree came from:

Proposition 2.2.4. [33, II.2.6] *Let ϕ be a separable isogeny. Then, $deg(\phi) = |ker(\phi)|$.*

For elliptic curves defined over a finite field, Tate has shown that being isogenous is equivalent to having the same cardinality.

Theorem 2.2.4. [39] *(Tate's Isogeny Theorem) Let E and E' be elliptic curves defined over some finite field $\mathbb{F}_{p^n}$. Then E and E' are isogenous over $\mathbb{F}_{p^n}$ if and only if*

$$|E(\mathbb{F}_{p^n})| = |E'(\mathbb{F}_{p^n})|.$$

This leads to the main isogeny theorem that we will need.

Theorem 2.2.5. [17, 9.6.19] *Let E be an elliptic curve over K. Let $G \subset E(\bar{K})$ be a finite subgroup that is defined over K (i.e., $\sigma(P) \in G$ for all $P \in G$ and $\sigma \in Gal(\bar{K}/K)$). Then there is a unique elliptic curve (up to isomorphism over $\bar{K}$) E' over K, and a unique isogeny (up to isomorphism over $\bar{K}$) $\phi : E \to E'$ over K such that $ker(\phi) = G$.*

The standard way of computing E', ϕ, or $\phi(P)$ for some $P \in E(\bar{K})$ is to use Vélu's formulas [21], which involves calculating a summation over all the elements of that subgroup $G = \ker(\phi)$ (see Section 2.3.1). The security of isogeny-based cryptography depends on the cardinality of these kernels, so the subgroup G must be large which makes Vélu's formulas impractical. However, as we will see in Section 3.2 the cardinality of G will be chosen to have small characteristic (2 or 3) and so we can apply Vélu's formulas to compute isogenies efficiently in such cases, as explained and optimized in [14].

2.2.3 Supersingularity

Let E/K be an elliptic curve defined over a field of characteristic p. A group of interest is that of the p-torsion points, $E[p]$. In fact, the structure of p-torsion points directly determines the endomorphism ring of E/K.

Definition 2.2.13. *Let K be a $\mathbb{Q}$-algebra that is finitely generated over $\mathbb{Q}$ (K can be non-Abelian). An* ***order*** *R of K is a subring of K that is finitely generated as a $\mathbb{Z}$-module and satisfies $R \otimes_{\mathbb{Z}} \mathbb{Q} = K$.*

Theorem 2.2.6. [33, V.3.1] *Let K be a field of prime characteristic p, and let E/K be an elliptic curve. For each integer $r \geq 1$ let*

$$\phi_r : E \to E^{(p^r)}$$

be the p^r-th Frobenius map.

The following are equivalent:

(i) $E[p^r] = \{O\}$ for all $r \geq 1$.

(ii) The map $[p] : E \to E$ *is purely inseparable and* $j(E) \in \mathbb{F}_{p^2}$*.*

(iii) End(E) *is an order in a quaternion algebra.*

If the equivalent conditions do not hold, then

$$E[p^r] = \mathbb{Z}/p^r\mathbb{Z}, \text{ for all } r \geq 1,$$

and End(E) *is an order in an imaginary quadratic field extension of* $\mathbb{Q}$*.*

If the above conditions hold, then we say the curve E is **supersingular**, otherwise we say E is **ordinary**. The term supersingular is unrelated to the notion of singular curves, and instead refers to how elliptic curves with these endomorphism rings are uncommon. Part (ii) of Theorem 2.2.6 states that all supersingular elliptic curves are always defined over $\mathbb{F}_{p^2}$, and this will be useful in the cryptanalysis of supersingular isogeny cryptography (Section 3.2.1).

A direct consequence of Theorem 2.2.6.(i) is that isogenies preserve the type of elliptic curve and so we can discuss supersingular elliptic curve isogenies and ordinary elliptic curve isogenies.

Theorem 2.2.7. *Let* $\phi : E_1 \to E_2$ *be an isogeny.* E_1 *is supersingular if and only if* E_2 *is supersingular.* E_1 *is ordinary if and only if* E_2 *is ordinary.*

Our compression algorithm applies to the cryptosystems described in [14], which use supersingular elliptic curve and their isogenies.

2.2.4 Hilbert Class Field Theory

The following section requires knowledge of Hilbert class fields and their general theory, and so a brief introduction is provided here (the sources of this subsection are [8] and [33]).

Let L be a number field. The set of algebraic integers in L form a ring, denoted $\mathcal{O}_L$, called the *ring of integers* of L.

Theorem 2.2.8. [*8*, *Theorem* 4.4.2] *The ring* $\mathcal{O}_L$ *is a free* $\mathbb{Z}$*-module of rank* $[L : \mathbb{Q}]$*.*

Definition 2.2.14. *An* ***integral ideal*** *is a* $\mathbb{Z}$*-submodule* $\mathfrak{a} \subset \mathcal{O}_L$ *such that for every* $\alpha \in \mathcal{O}_L$ *and* $a \in \mathfrak{a}$ *we have* $\alpha a \in \mathfrak{a}$*.*

Definition 2.2.15. *A* ***fractional ideal*** $\mathfrak{a} \subset L$ *is a non-zero submodule of* L *such that there exists a non-zero integer* α *with* $\alpha\mathfrak{a}$ *an integral ideal of* $\mathcal{O}_L$*.*

We would like to have a group structure for fractional ideals. The full ring of integers $\mathcal{O}_L$ will serve as the identity. We define the product of fractional ideals $\mathfrak{a}, \mathfrak{b}$ as:

$$\mathfrak{a}\mathfrak{b} := \{\sum ij | i \in \mathfrak{a}, j \in \mathfrak{b}\}.$$

Note this operation is Abelian. We make use of the following theorem to define inverses:

Theorem 2.2.9. [8, *Theorem* 4.6.14] *If $\mathfrak{a}$ is a fractional ideal of $\mathcal{O}_L$ and if we set*

$$\mathfrak{a}^{-1} := \{\alpha \in L | \alpha\mathfrak{a} \subset \mathcal{O}_L\},$$

then $\mathfrak{a}\mathfrak{a}^{-1} = \mathfrak{a}^{-1}\mathfrak{a} = \mathcal{O}_L$.

Lastly, we say that two fractional ideals are equivalent if they differ by a non-zero element of L. Now we can define the **class group** of $\mathcal{O}_L$ to be the finite group of equivalence classes of fractional ideals with the above operation. We denote this group by $\mathcal{CL}(\mathcal{O}_L)$ and define the **class number** as $h(\mathcal{O}_L) := |\mathcal{CL}(\mathcal{O}_L)|$.

Since integral ideals of $\mathcal{O}_L$ are modules of maximal rank [8, Theorem 4.6.3], the quotient $\mathcal{O}_L/\mathfrak{a}$ is a finite ring. In the case where this quotient ring is a field we say that $\mathfrak{a}$ is a **prime ideal** of $\mathcal{O}_L$.

Definition 2.2.16. *Let p be a prime number and $\mathfrak{p}$ a prime ideal of $\mathcal{O}_L$. Then $\mathfrak{p}$ is said to be a prime ideal* ***above*** *p if $\mathfrak{p} \cap \mathbb{Z} = p\mathbb{Z}$.*

Theorem 2.2.10. [8, *Theorem* 4.8.3] *Let p be a prime number. There exist positive integers e_i such that*

$$p\mathcal{O}_L = \prod_i \mathfrak{p}_i^{e_i},$$

where the $\mathfrak{p}_i$ are all the prime ideals above p.

Definition 2.2.17. *Depending on the structure of the product in Theorem* 2.2.8 *we give the prime number p different names. If $p\mathcal{O}_L = \mathfrak{p}$, then p is said to be* ***inert****. If $p\mathcal{O}_L = \prod_{i=1}^{n} \mathfrak{p}_i$, where $n = [L : \mathbb{Q}]$ and all the $\mathfrak{p}_i$'s are different, then p said to* ***split completely****. If $e_i \geq 2$ for some i, then p is* ***ramified****.*

We end this section with another look at the Frobenius endomorphism. For elliptic curves defined over $\mathbb{Q}$, the m-torsion points have coordinates in $\bar{\mathbb{Q}}$. Each element of the Galois group $\mathrm{Gal}(\bar{\mathbb{Q}}/\mathbb{Q})$ fixes the m-torsion points of E, $\forall m \in \mathbb{Z}$. By Theorem 2.1.1, since

$char(\mathbb{Q}) = 0$, we know that $E(\mathbb{Q})[m] \cong \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$ and so the group of homomorphisms from $E[m]$ to itself is $\mathrm{GL}_2(\mathbb{Z}/m\mathbb{Z})$. We define the map

$$R_{E,m} : \mathrm{Gal}(\bar{\mathbb{Q}}, \mathbb{Q}) \to \mathrm{GL}_2(\mathbb{Z}/m\mathbb{Z})$$

as the **mod m representation attached to** E. While the particular matrix associated with $R_{E,m}(\sigma)$ depends on the choice of basis for $E[m]$, the determinant and trace are invariants.

Let p be prime and $\mathfrak{p}$ be a prime ideal above p. There are **Frobenius elements** $\sigma_p \in \mathrm{Gal}(\bar{\mathbb{Q}}, \mathbb{Q})$ defined by the property that

$$\sigma_p(\alpha) \equiv \alpha^p \bmod \mathfrak{p}$$

for all $\alpha \in \bar{\mathbb{Q}}$. Evaluating $R_{E,m}$ at σ_p gives a matrix in $\mathrm{GL}_2(\mathbb{Z}/m\mathbb{Z})$, and its trace is defined to be the trace of the Frobenius endomorphism t. Any ambiguity from the choice of m is extinguished by the following:

Theorem 2.2.11. *For all $m \geq 1$, $Trace(R_{E,m}(\sigma_p)) \equiv t \bmod m$.*

2.3 Isogeny Computations

This section examines a few algorithms that are commonly used within isogeny-based cryptography.

2.3.1 Vélu's Formula

In 1971, Jacques Vélu provided explicit formulas to compute isogenies in time proportional to half the cardinality of its kernel [21]. Later work [14] optimized this result for the case when the cardinality of the kernel is a power of a small prime number. Below are the formulas and algorithms for these computations.

Using the notation in Vélu's paper: let $E : y^2 = x^3 + Ax + B$ be an elliptic curve defined over an algebraically closed field, let F be a finite subgroup of E, and let $f : E \to E'$ be the isogeny with kernel F.

For a point $P = (x_P, y_P)$ on E, define the following quantities:

$$\begin{aligned} g_P^x &= 3x_P^2 + A, \\ g_P^y &= -2y_P, \\ v_P &= 2g_P^x, \\ u_P &= (g_P^y)^2. \end{aligned}$$

In order to distinguish points in F from their inverse we define F_2 to be the non-trivial points of order 2 in F, and we define R to be a subset of $(F \backslash \{O\}) \backslash F_2$ such that

$$R \cap (-R) = \emptyset \text{ and } (F \backslash \{O\}) \backslash F_2 = R \cup (-R).$$

Setting $S = F_2 \cup R$, we can define

$$f(x,y) = \left(x + \sum_{P \in S} \frac{v_P}{x - x_P} - \frac{u_P}{(x - x_P)^2},\ y - \sum_{P \in S} u_P \frac{2y}{(x - x_P)^3} + v_P \frac{y - y_P - g_P^x g_P^y}{(x - x_P)^2} \right),$$

for points $(x, y) \notin F$, and $f(x, y) = O$ when $(x, y) \in F$. Further, the equation for the image curve is given by

$$E' : y^2 = x^3 + \left(A - 5 \sum_{P \in S} v_P \right) x + \left(B - 7 \sum_{P \in S} u_P + x_P v_P \right).$$

From these formulas it is plain to see that this computation is inefficient for cryptographically secure kernel sizes. Suppose that $|ker(f)| = p^n$ for prime p and $n \geq 1$. De Feo and Jao [14] show how we can compute f in $O(np)$ instead of the runtime $O(p^n)$ from Vélu's formula.

Let Q_1 be a point in F with order p (see 4.1.2 for one possible method). Applying Vélu's formula to the subgroup $\langle Q_1 \rangle$ of E will give an isogeny

$$f_1 : E \to E/\langle Q_1 \rangle.$$

Then the image of F under f_1 will have size p^{n-1}. If we then find a point Q_2 in $f_1(F) \subset E/\langle Q_1 \rangle$ with order p, we can perform this procedure for a second time to determine

$$f_2 : E/\langle Q1 \rangle \to (E/\langle Q_1 \rangle)/\langle Q_2 \rangle,$$

and codomain. Iterating this n times gives n isogenies, $f_1, f_2, \ldots, f_n$ whose composition is the desired isogeny f with kernel F. The codomain of f is the elliptic curve $E' = (\ldots((E/\langle Q_1 \rangle)/\langle Q_2 \rangle) \ldots)/\langle Q_n \rangle$.

2.3.2 Bröker's Algorithm

For the purposes of isogeny-based cryptography, we need to be able to efficiently find a supersingular elliptic curve over $\mathbb{F}_{p^n}$ with trace t of the p^n-Frobenius endomorphism, for a given p^n and t. In 2000, Reinier Bröker solved this computational problem [4] and this section is based on that work. The existence of such a curve is guaranteed by the following result of Waterhouse:

Theorem 2.3.1. [42, 4.1] *There exists a supersingular elliptic curve E over $\mathbb{F}_{p^n}$ with trace t of the p^n-Frobenius endomorphism π_E if and only if one of the following holds:*

(a) if n is even and one of the following is true:

(i) $t = \pm 2\sqrt{p^n}$,

(ii) $t = \pm\sqrt{p^n}$ and $p \not\equiv 1$ mod 3,

(iii) $t = 0$ and $p \not\equiv 1$ mod 4;

(b) if n is odd and one of the following is true:

(i) $t = 0$,

(ii) $t = \pm\sqrt{2p^n}$ and $p = 2$,

(iii) $t = \pm\sqrt{3p^n}$ and $p = 3$.

The cases which are relevant in the context of supersingular isogeny-based cryptography are when $t = 0$. The first step to construct a supersingular curve over $\mathbb{F}_p$ as a reduction of a curve in characteristic 0 using the following result of Deuring.

Theorem 2.3.2. [24, 13.12] *Let E be an elliptic curve defined over a number field L whose endomorphism ring is the maximal order $\mathcal{O}_K$ in an imaginary quadratic field K. Let $\mathfrak{p}$ be a prime ideal of L, let p be a prime number such that $p \nmid \Delta(E)$ and $\mathfrak{p}$ is above p. Then $E/(L/\langle\mathfrak{p}\rangle)$ is supersingular if and only if p does not split in K.*

From the theory of complex multiplication we have that the j-invariant of E generates the Hilbert class field of K, H, when adjoined to K. That is,

$$H = K[x]/\langle P_K\rangle,$$

where P_K is the minimal polynomial of $j(E)$ over $\mathbb{Q}$. The polynomial P_K can be explicitly computed [5] and its degree is equal to the Hilbert class number h_K. As $j(E) \in \mathbb{F}_{p^2}$, [33, V.3.1] the polynomial P_K splits over $\mathbb{F}_{p^2}$. The lemma due to Bröker gives a sufficient condition for $P_K \in \mathbb{F}_p[x]$ to have a root in $\mathbb{F}_p$.

Lemma 2.3.1. [4, 2.3] *Let K be an imaginary quadratic field with odd class number h_K. Then, $K = \mathbb{Q}(i)$, or $K = \mathbb{Q}(\sqrt{-2})$, or $K = \mathbb{Q}(\sqrt{-q})$ with q prime and congruent to 3 modulo 4.*

Combining these results, Bröker gives an algorithm for constructing a supersingular elliptic curve over $\mathbb{F}_p$.

Algorithm 1. *Input: a prime number p. Output: a supersingular elliptic curve over $\mathbb{F}_p$.*

```
1. If $p = 2$, return $y^2 + y = x^3$.
2. If $p \equiv 3 \mod 4$, return $y^2 = x^3 - x$.
3. Let $q \equiv 3 \mod 4$ be the smallest prime with $-q$ a non-quadratic residue mod $p$.
4. Compute $P_K \in \mathbb{Z}[x]$ for $K = \mathbb{Q}(\sqrt{-q})$.
5. Compute a root $j \in \mathbb{F}_p$ of $P_K \in \mathbb{F}_p[x]$.
6. If $q = 3$, return $y^2 = x^3 - 1$. Otherwise, set $a \leftarrow \frac{27j}{4(1728-j)} \in \mathbb{F}_p$ and return
   $y^2 = x^3 + ax - a$.
```

Let $q = p^n$ be a prime power and let t be a trace of the Frobenius endomorphism of the form described in 2.3.1. From Algorithm 1, we compute a supersingular elliptic curve E over $\mathbb{F}_p$. Let $E'/\mathbb{F}_q$ be E defined over $\mathbb{F}_q$, and let t' be the trace of the Frobenius endomorphism of $E'(\mathbb{F}_q)$.

Lemma 2.3.2. [4, 3.1] *If n is odd, then $t' = 0$. If $n \equiv 0 \mod 4$, then $t' = 2\sqrt{q}$. Otherwise, $n \equiv 2 \mod 4$, and $t' = -2\sqrt{q}$.*

If $p \not\equiv 1 \bmod 4$, then a twist by a primitive fourth root of unity $i \in \mathbb{F}_q$ will give curves with Frobenius trace $\pm 2\sqrt{q}$ and 0. If $p \not\equiv 1 \bmod 3$, then a twist by a primitive sixth root of unity $\zeta_6 \in \mathbb{F}_q$ will give curves with Frobenius trace $\pm 2\sqrt{q}$, and $\pm\sqrt{q}$. If $p \equiv 1 \bmod 12$, then a twist by -1 suffices. By Theorem 2.3.1 we know that these are the only three cases.

Chapter 3

Isogeny-Based Cryptography

This chapter will cover the two types of isogeny-based cryptography, the computational problems their security depend on, and their susceptibility to known attacks. Recall the definitions for ordinary and supersingular curves in Section 2.2.3 are based on the p-torsion points of the curve when defined over a finite field of characteristic p.

3.1 Ordinary Elliptic Curve Cryptography

The first public-key cryptosystems based on the intractability of constructing an isogeny between two known elliptic curves is due to Couveignes [11] and Stolbunov [37] independently, using ordinary curves. Couveignes introduced the notion of a Hard Homogeneous Space (HHS) to generalize the discrete logarithm problem and showed how it can be used for key exchange and authentication schemes.

Definition 3.1.1. *Let G be a finite, Abelian group. Then a* ***homogeneous space*** *H for G is a set that is acted on by G such that $|H| = |G|$ and the action is simply transitive (for all $h_0, h_1 \in H$ there exists a unique $g \in G$ such that $g \cdot h_0 = h_1$). For $h_1, h_2 \in H$ denote the unique element $g \in G$ with $g \cdot h_1 = h_2$ by $\delta(h_1, h_2)$.*

This definition alone is not enough to produce a cryptographic scheme. As in the setting of the discrete logarithm problem over a finite field, we require that the basic group operations are efficiently computable while the inverse of the action is not.

Definition 3.1.2. [11] *Let H be a homogeneous space for G, and suppose the elements of G and H are represented by strings (not necessarily uniquely).*

Suppose the following computations are efficient:

(i) the group operation of G,

(ii) inverting an element of G,

(iii) testing membership in G and H,

(iv) testing equality in G and in H,

(iv) finding a random element in G with uniform probability,

(v) computing $g \cdot h$ for all $g \in G$ and $h \in H$.

Further suppose these two problems are computationally difficult:

(vi) given $h_1, h_2 \in H$ compute $\delta(h_1, h_2)$,

(vii) given $h_1, h_2, h_3 \in H$ compute the unique $h_4 \in H$ with $\delta(h_1, h_2) = \delta(h_3, h_4)$.

*Then we say H is a **hard homogeneous space**.*

This basic setup allows us to create cryptosystems based on the action of isogenies on ordinary elliptic curves defined over finite fields. Let E be an ordinary elliptic curve over $\mathbb{F}_{p^n}$. Recall from 2.2.3 that $\mathcal{O} := End(E)$ is an order in an imaginary quadratic field extension over $\mathbb{Q}$, say $K = \mathbb{Q}(\sqrt{\Delta(E)}) = \mathcal{O} \otimes_{\mathbb{Z}} \mathbb{Q}$. If we assume that the discriminant $\Delta(E)$ is square-free, then $\mathcal{O}$ is the maximal order, $\mathcal{O}_K$, in K. From the theory of complex multiplication [34, II.1.5], the ideal class group of $\mathcal{O}_K$ induces a simply transitive action on the set of elliptic curves isogenous to E.

Definition 3.1.3. [34, *Chapter 2 - Section 1*] *Define $\mathcal{ELL}(\mathcal{O})$ to be the quotient space $\{E/\mathbb{C}$ with $\mathcal{O} \cong End(E)\}/\{$isomorphism over $\mathbb{C}\}$.*

By Theorem 2.2.1, we can associate each element of $\mathcal{ELL}(\mathcal{O})$ with a j value. Couveignes and Stolbunov independently determined that if we set

$$G = \mathcal{CL}(\mathcal{O}_K) = \text{ ideal class group of } \mathcal{O}_K, \text{ and } H = \mathcal{ELL}(\mathcal{O}_K),$$

then there is an action

$$* : \mathcal{CL}(\mathcal{O}_K) \times \mathcal{ELL}(\mathcal{O}_K) \to \mathcal{ELL}(\mathcal{O}_K)$$

that satisfies the conditions of the hard homogeneous space definition. Some of the necessary material to confirm this is true has been given in 2.2.1, 2.2.2, and 2.2.4, however [34, Chapter 2 - Section 1] is the recommended source for details (specifically Proposition 1.2). To test equality of elements in H, and [25] is an early source for computing the action required in condition (v) of 3.1.2 while [3] details a more efficient computation.

Figure 3.1 details the ordinary isogeny key exchange protocol. Here $\mathcal{CL}(\mathcal{O}_K)$, $\mathcal{ELL}(\mathcal{O}_K)$, and $x \in \mathcal{ELL}(\mathcal{O}_K)$ are all public systems parameters.

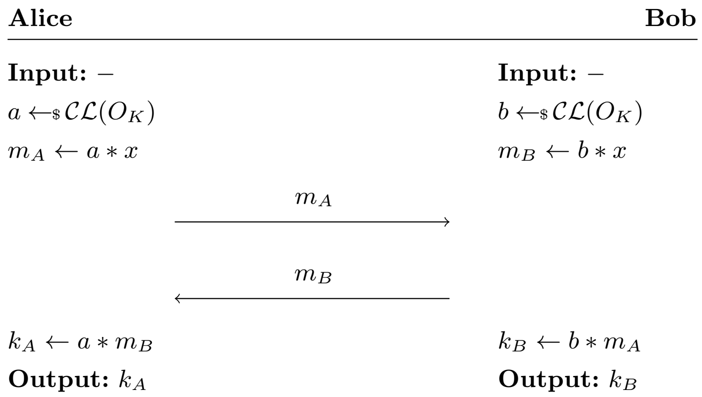

Figure 3.1: Ordinary isogeny-based key exchange

3.1.1 Security and the Ordinary Isogeny Graph

As one would expect, the underlying computational problems of this ordinary elliptic curve isogeny scheme are conditions (vi) and (vii) of 3.1.2.

Problem 1. *Let $E_1/\mathbb{F}_{p^n}$ and $E_2/\mathbb{F}_{p^n}$ be ordinary elliptic curves with $|E_1(\mathbb{F}_{p^n})| = |E_2(\mathbb{F}_{p^n})|$. Compute an $\mathbb{F}_{p^n}$-isogeny $\phi : E_1 \to E_2$.*

Problem 2. *Let $E_1/\mathbb{F}_{p^n}, E_2/\mathbb{F}_{p^n}$ and $E_3/\mathbb{F}_{p^n}$ be ordinary elliptic curves with $|E_1(\mathbb{F}_{p^n})| = |E_2(\mathbb{F}_{p^n})| = |E_3(\mathbb{F}_{p^n})|$. Let $[\alpha] \in \mathcal{CL}(O_K)$ be such that $[\alpha] * E_1 = E_2$. Compute the unique (up to $\mathbb{F}_{p^n}$-isomorphism) elliptic curve $E_4 = [\alpha] * E_3$.*

The fastest known classical algorithm solving the ordinary isogeny problem 1 is probabilistic with a worst-case and average-case of $O((p^n)^{1/4+o(1)} \log^2(p^n) \log(\log(p^n)))$ [19]. This result is an improvement of [18] and is achieved by taking a pseudorandom walk of the isogeny-graph and using the easily computable small degree isogenies more often than larger degree isogenies.

With a quantum computer, the most efficient algorithm [7] for the same problem has a subexponential running time of $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2})$ under the Generalized Riemann Hypothesis, where

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

The authors propose reducing the problem to an instance of the Abelian hidden shift problem, and then using Kuperberg's quantum algorithm [23] which applies here because the reduction will be an *injective* hidden shift problem [7, 4.1].

3.2 Supersingular Elliptic Curve Cryptography

In 2011, De Feo, Jao and Plût [14] proposed a cryptosystem from supersingular elliptic curve isogenies. The central difference in the supersingular setting is that the endomorphism ring of the curve is non-Abelian (see Theorem 2.2.6, (iii)). The authors overcome this difficulty by having the participating members of the key-exchange send additional information about the isogeny; the image of four points on the curve. The compression techniques presented in 4.2 will reduce this additional information by a factor of two.

Setup: Let p be a fixed prime number of the form $p = \ell_A^{e_A}\ell_B^{e_B}f \pm 1$, where ℓ_A and ℓ_B are distinct primes, and f is some small prime. The typical choices are $\ell_A = 2$ and $\ell_B = 3$, and so p will be assumed to be of this form. Let $E/\mathbb{F}_{p^2}$ be a supersingular elliptic curve with $E[2^{e_A}]$ and $E[3^{e_B}]$ defined over $\mathbb{F}_{p^2}$. Let $P_A, Q_A, P_B, Q_B \in E(\mathbb{F}_{p^2})$ be four points such that $\langle P_A, Q_A\rangle = E[2^{e_A}]$ and $\langle P_B, Q_B\rangle = E[3^{e_B}]$ (by 2.1.1 each of these torsion subgroups require two points to generate).

Alice chooses two random elements $m_A, n_A \in \mathbb{Z}/2^{e_A}\mathbb{Z}$, not both even, and computes the point $R_2 := [m_A]P_A + [n_A]Q_A$ and the isogeny

$$\phi_A : E \to E_A$$

such that $ker(\phi_A) = \langle R_2\rangle$. Additionally, Alice computes the images of the $E[3^{e_B}]$ generators; $\phi_A(P_B), \phi_A(Q_B) \in E_A(\mathbb{F}_{p^2})$. Similarly, Bob computes a random linear combination R_3 (chosen so that not both m_B and n_B are divisible by 3) of P_B and Q_B, the isogeny

$$\phi_B : E \to E_B$$

with kernel $\langle R_3\rangle$, and the points $\phi_B(P_A), \phi_B(Q_A) \in E_B(\mathbb{F}_{p^2})$. Alice and Bob's secret keys are the numbers m_A, n_A and m_B, n_B, respectively.

Using an unsecured channel, Alice sends $(E_A, \phi_A(P_B), \phi_A(Q_B))$ to Bob, and Bob sends $(E_B, \phi_B(P_A), \phi_B(Q_A))$ to Alice. The shared secret elliptic curve can now be computed by both parties. Alice computes the point $S_2 := [m_A]\phi_B(P_A) + [n_A]\phi_B(Q_A) \in E_B(\mathbb{F}_{p^2})$ and the isogeny from E_B with kernel generated by S_2,

$$\phi'_A : E_B \to E_{BA}.$$

Alice		**Bob**
$m_A, n_A \xleftarrow{\$} \mathbb{Z}_{2^{e_A}}$		$m_B, n_B \xleftarrow{\$} \mathbb{Z}_{3^{e_B}}$
$R_2 \leftarrow [m_A]P_A + [n_A]Q_A$		$R_3 \leftarrow [m_B]P_B + [n_B]Q_B$
$\phi_A : E \rightarrow E/\langle R_2 \rangle$		$\phi_B : E \rightarrow E/\langle R_3 \rangle$
$\phi_A(P_B), \phi_A(Q_B)$		$\phi_B(P_A), \phi_B(Q_A)$
	$\xrightarrow{E_A, \phi_A(P_B), \phi_A(Q_B)}$	
	$\xleftarrow{E_B, \phi_B(P_A), \phi_B(Q_A)}$	
$S_2 \leftarrow [m_A]\phi_B(P_A) + [n_A\phi_B(Q_A)$		$S_3 \leftarrow [m_B]\phi_A(P_B) + [n_B]\phi_A(Q_B)$
$\phi'_A : E/\langle R_3 \rangle \rightarrow (E/\langle R_3 \rangle)/\langle S_2 \rangle$		$\phi'_B : E/\langle R_2 \rangle \rightarrow (E/\langle R_2 \rangle)/\langle S'_3 \rangle$
$j_A \leftarrow j((E/\langle R_3 \rangle)/\langle S_2 \rangle)$		$j_B \leftarrow j((E/\langle R_2 \rangle)/\langle S_3 \rangle)$

Figure 3.2: Supersingular isogeny-based key exchange protocol

Similarly, Bob computes $S_3 := [m_B]\phi_A(P_B) + [n_B]\phi_A(Q_B) \in E_A(\mathbb{F}_{p^2})$ and

$$\phi'_B : E_A \rightarrow E_{AB}$$

such that $ker(\phi'_B) = \langle S_3 \rangle$. The two curves E_{AB} and E_{BA} are isomorphic over $\mathbb{F}_{p^2}$, in particular they have equal j-invariants, and so the shared secret key is $j(E_{AB}) = j(E_{BA})$.

Figure 3.2 details the supersingular isogeny key exchange protocol. Here E, P_A, Q_A, P_B, and Q_B are all public systems parameters.

3.2.1 Security and the Supersingular Isogeny Graph

Listed below are the security assumptions under which the security of supersingular isogeny-based cryptosystems can be proven. The corresponding security proofs can be found in [14]. Figure 3.3 helps understand why these are the underlying security assumptions.

Problem 3 (SSI)**.** *Let $\phi_A : E \rightarrow E_A$ be an isogeny whose kernel is $\langle [m_A]P_A + [n_A]Q_A \rangle$, where m_A and n_A are randomly chosen from $\mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ and are not both divisible by ℓ_A. Given E_A and the values $\phi_A(P_B), \phi_A(Q_B)$, the Supersingular Isogeny problem is to find a generator R_A of $\langle [m_A]P_A + [n_A]Q_A \rangle$.*

Problem 4 (SSCDH)**.** *Let $\phi_A : E \rightarrow E_A$ be an isogeny whose kernel is $\langle [m_A]P_A + [n_A]Q_A \rangle$, and let $\phi_B : E \rightarrow E_B$ be an isogeny whose kernel is $\langle [m_B]P_B + [n_B]Q_B \rangle$, where*

$$\begin{tikzcd}
 & E/\langle S\rangle \arrow[dr, "ker(\psi_3)=\langle\phi_2(R)\rangle"] \arrow[dd, dashed, shift left] & \\
E \arrow[ur, "ker(\phi_2)=\langle S\rangle"] \arrow[dr, "ker(\phi_3)=\langle R\rangle"'] & & E/\langle S,R\rangle \\
 & E/\langle R\rangle \arrow[ur, "ker(\psi_2)=\langle\phi_3(S)\rangle"'] \arrow[uu, dashed, shift left] &
\end{tikzcd}$$

Figure 3.3: Supersingular key-exchange diagram

m_A, n_A (respectively m_B, n_B) are randomly chosen from $\mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ (respectively $\mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$) and are not both divisible by ℓ_A (respectively ℓ_B). Given the curves E_A, E_B and the points $\phi_A(P_B), \phi_A(Q_B), \phi_B(P_A), \phi_B(Q_A)$, the Supersingular Computational Diffie-Hellman problem is to find the j-invariant of $E/\langle [m_A]P_A + [n_A]Q_A, [m_B]P_B + [n_B]Q_B\rangle$.

Problem 5 (SSDDH)**.** *Given a tuple sampled with probability 1/2 from one of the following two distributions:*

— $(E_A, E_B, \phi_A(P_B), \phi_A(Q_B), \phi_B(P_A), \phi_B(Q_A), E_{AB})$, where $E_A, E_B, \phi_A(P_B), \phi_A(Q_B), \phi_B(P_A), \phi_B(Q_A), E_{AB}$ are as in the SSCDH problem and $E_{AB} \cong E/\langle [m_A]P_A + [n_A]Q_A, [m_B]P_B + [n_B]Q_B\rangle$,

— $(E_A, E_B, \phi_A(P_B), \phi_A(Q_B), \phi_B(P_A), \phi_B(Q_A), E_C)$, where $E_A, E_B, \phi_A(P_B), \phi_A(Q_B), \phi_B(P_A), \phi_B(Q_A)$ are as in the SSCDH problem, m'_A, n'_A (respectively m'_B, n'_B) are randomly chosen from $\mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ (respectively $\mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$) and are not both divisible by ℓ_A (respectively ℓ_B), and $E_C \cong E/\langle [m'_A]P_A + [n'_A]Q_A, [m'_B]P_B + [n'_B]Q_B\rangle$.

the Supersingular Decision Diffie-Hellman problem is to determine from which distribution the tuple is sampled.

For any field K with $char(K) = p > 0$ and non-empty set of prime L with $p \notin L$, define the *supersingular isogeny graph* $X(K, L)$ where each vertex is a K-isomorphism class of elliptic curves defined over K (vertices have an associated, unique j-invariant), and the edges are equivalence classes of degree ℓ isogenies defined over K for $\ell \in L$, connecting isogenous curves. Given a path on $X(K, L)$ connecting say j_1 and j_2, an explicit starting curve with j-invariant j_1 must be chosen before the isogeny can be computed (see the canonical elliptic curve associated to each j in Section 2.2.2). Once this is done, the isogeny can be computed by composing each isogeny (edge) in the path [6].

If E_1 and E_2 are such that $j(E_1) = j_1$ and $j(E_2) = j_2$, then the isogeny computed in this way may have codomain elliptic curve isomorphic to E_2 in which case the composition of isogeny and isomorphism will give the correct isogeny. This differs from the ordinary isogeny graph where two elliptic curves in the same equivalence class (vertex) may be quadratic twists of each other.

By Theorem 2.2.6.(ii) every supersingular curve has j-invariant in $\mathbb{F}_{p^2}$, so setting $K = \mathbb{F}_{p^2}$ gives the full supersingular isogeny graph of L-isogenies. The authors of [12] instead set $K = \mathbb{F}_p$ and look at this restricted graph; below is their main result. First, let $h(\theta)$ denote the Hilbert class number of $\mathbb{Q}(\sqrt{\theta})$, and $(\frac{a}{b})$ denote the Legendre symbol for quadratic residues.

Theorem 3.2.1. *Let $p > 3$ be prime.*

(a) If $p \equiv 1 \pmod 4$, then there are $h(-4p)$ $\mathbb{F}_p$-isomorphism classes of supersingular elliptic curves over $\mathbb{F}_p$, all with the endomorphism ring $\mathbb{Z}[\sqrt{-p}]$. From each vertex there is exactly one outgoing $\mathbb{F}_p$-rational 2-isogeny, and two outgoing ℓ-isogenies for every prime $\ell > 2, \ell \in L$ such that $(\frac{-p}{\ell}) = 1$.

(b) If $p \equiv 3 \pmod 4$, then there are two cases. In both cases each vertex has two ℓ-isogenies for every prime $\ell > 2, \ell \in L$ such that $(\frac{-p}{\ell}) = 1$. Additionally, in both cases there are two "levels" to the graph referred to as the "surface" and the "floor" of the graph. The endomorphism ring of every vertex on the surface is isomorphic to the order $\mathbb{Z}[\frac{1+\sqrt{-p}}{2}]$, while the endomorphism ring of every vertex on the floor is isomorphic to the order $\mathbb{Z}[\sqrt{-p}]$.

(i) If $p \equiv 7 \pmod 8$, then each level has $h(-p)$ vertices. The surface and the floor are connected $1:1$ with 2-isogenies, and on the surface there are also two 2-isogenies from each vertex to other vertices on the surface.

(ii) If $p \equiv 3 \pmod 8$, then there are $h(-p)$ vertices on the surface and $3h(-p)$ vertices on the floor. The surface and the floor are connected $1:3$ with 2-isogenies.

A consequence of this theorem is an algorithm to solve the supersingular isogeny problem over $\mathbb{F}_p$ with a classical running time of $O(p^{1/4})$ (assuming the Generalized Riemann Hypothesis), and the full supersingular isogeny problem over $\bar{\mathbb{F}}_p$ in $O(p^{1/2})$. Given two vertices of the graph (j-invariants) representing E and say E_A, take isogenies to the surface if necessary and then perform a random walk using isogenies of degree $2^{e_A/2}$ from each vertex until a collision, j', is found. Choose an elliptic curve E' with $j(E') = j'$. The isogeny from E to E_A will be the composition of each isogeny in the walk from E to E' and the dual of each isogeny in the walk from E' to E_A.

Another consequence of this result is a quantum algorithm for the full supersingular isogeny problem (that is, over the base field $\mathbb{F}_{p^2}$ instead of $\mathbb{F}_p$) [2]. The first step is: given two supersingular elliptic curves over $\mathbb{F}_{p^2}$, use Grover's algorithm to find isogenous elliptic curves defined over $\mathbb{F}_p$. This computation has a quantum runtime of $O(p^{1/4})$. Then, using 3.2.1 and the algorithm of [12], the isogeny between these to elliptic curves defined over $\mathbb{F}_p$ can be computed classically in $O(p^{1/4})$. This gives a path in the supersingular isogeny graph and so the composition of each directed isogeny (dual isogenies give opposite directions) in the path will give the correct output (up to isomorphism) with a total quantum runtime of $O(p^{1/4})$.

Formally, the problem of finding a collision in a graph can be formulated in the following way:

Problem 6 (Claw Problem)**.** *Given function $f : A \to C, g : B \to C$ with $|A| = |B|$, find a pair $(a, b) \in A \times B$ such that $f(a) = g(b)$.*

A solution to this complexity problem using quantum computers was shown to be optimal in the black-box model [38], with runtime $O(\sqrt[3]{|A||B|})$. We can apply this to the supersingular isogeny graph. The degree of the isogeny between E and Alice's elliptic curve E_A is 2^{e_A}, so let A be the set of all isogenies of degree $2^{e_A/2}$ from E and let B be the set of all isogenies of degree $2^{e_A/2}$ from E_A. Here C will be the set of all elliptic curves E' defined over $\mathbb{F}_{p^2}$ with $|E(\mathbb{F}_{p^2})| = |E'(\mathbb{F}_{p^2})|$ (see 2.2.4). The sets A and B have equal cardinality and so the algorithm [38] applies. Hence, there is a quantum attack in $O(\sqrt[3]{2^{e_A/2}2^{e_A/2}}) = O(2^{e_A/3}) = O(p^{1/6})$ against supersingular isogeny schemes.

Chapter 4

Public-Key Compression

This chapter contains results from "Key compression for isogeny-based cryptosystems" by R. Azarderakhsh, D. Jao, K. Kalach, B. Koziel, and C. Leonardi, to appear in AsiaPKC 2016. In this chapter we cover the necessary material for compression of public information and the actual compression techniques.

4.1 Computational Aspects

There are two pieces of information to be compressed; the elliptic curve that each participant computes, and the auxiliary points on the curve that will be used to determine the shared elliptic curve.

Once Alice computes her public elliptic curve $E_A : y^2 = x^3 + ax + b$ in short Weierstraß form, the naive way to inform Bob of her curve is to send the two defining coefficients, a and b. But since Bob only needs to know the isomorphism class of E_A, there are other ways to convey this information.

The standard way of sending Alice's auxiliary points, $\phi_A(P_B), \phi_A(Q_B)$, is to send the x-coordinate of each point as this will determine the point up to a sign of the y-coordinate, and this does not interfere with later steps because $\langle P \rangle = \langle -P \rangle$ If instead Alice can represent each point in terms of a basis for the $\mathbb{Z}$-module $E_A[3^{e_B}]$, then she is able to reduce the amount of bits needed to store each point.

This section covers the computational techniques and algorithms that are required to compress the elliptic curve and the auxiliary points.

4.1.1 Weil Pairing

For any Abelian ring R, a pairing is an R-bilinear map. Pairings have many uses in elliptic curve cryptography such as the Boneh-Franklin identity-based encryption scheme and the Menezes-Okamoto-Vanstone reduction. We will be utilizing the Weil pairing in the following subsection to speed up the elliptic curve discrete logarithm computation.

Definition 4.1.1. *For $m \in \mathbb{N}$ and $E[m] = \{P \in E(\mathbb{F}_{p^n}) : mP = O(E)\}$, the Weil pairing is a map $e\colon E[m] \times E[m] \to \mathbb{F}_{p^n}$, satisfying bilinearity and non-degeneracy:*

$$\begin{gathered} e(P_1 + P_2, Q) = e(P_1, Q)e(P_2, Q) \\ e(P, Q_1 + Q_2) = e(P, Q_1)e(P, Q_2) \\ \forall P \in E[m] \backslash \{O\}, \exists Q \in E[m] \text{ such that } e(P, Q) \neq 1. \end{gathered}$$

Miller's algorithm [26] provides an efficient way to compute the Weil pairing. From the definition we see that

$$\forall P \in E[m], e(P, P) = O \text{ and } \forall a \in \mathbb{N}, e(aP, Q) = e(P, Q)^a = e(P, aQ). \qquad (4.1)$$

4.1.2 Torsion Basis

For the upcoming compression, we will need a deterministic way of computing a basis for the torsion subgroups $E[2^{e_A}]$ and $E[3^{e_B}]$ for a supersingular elliptic curve E over $\mathbb{F}_{p^2}$ with $p = 2^{e_A}3^{e_B}f \pm 1$. This can be accomplished with the following procedure.

Fix a hash function $H : \mathbb{Z} \to E_A$ (not required to be a cryptographic hash function). Set the counter $i = 0$ and compute the point $P = [2^{e_A}f]H(i)$ and determine if its order is 3^{e_B} by checking if $[3^{e_B-1}]P \neq O$. If P does not have the correct order (this occurs when $3 \mid \text{order}(H(i))$), then increase the counter by 1 and compute a new P. If p does have order 3^{e_B}, then set this point to B_1. Increase the counter by 1 and continue the process by computing $Q = [2^{e_A}f]H(i)$. As before, check if $[3^{e_B-1}]Q \neq O$ to ensure Q has the correct order, increasing the counter and repeating if necessary. Finally, check that Q is independent from B_1. One way to do this is to check that their Weil pairing is not 1. If Q is independent from B_1, then set $B_2 = Q$ and return the basis $\{B_1, B_2\}$ for $E_A[3^{e_B}]$. If both Alice and Bob follow this procedure they will have computed the same basis for $E[3^{e_B}]$.

4.1.3 Two-Dimensional Elliptic Curve Discrete Logarithms

We consider the following generalization of the discrete logarithm problem on elliptic curves: Let E be a supersingular elliptic curve defined over the finite field $\mathbb{F}_{p^2}$, for some prime number p, and $N \in \mathbb{Z}$ with $N|p+1$. Given two points $\{A, B\}$ generating the N-torsion subgroup $E[N]$, and an element $C \in E[N]$, the two-dimensional elliptic curve discrete logarithm problem is to determine $m, n \in \mathbb{Z}/N\mathbb{Z}$ such that $C = [m]A + [n]B$. Below are two practical algorithms for solving in our context.

If we denote the largest prime dividing $|S|$ by p_{max}, then there is a variation of the Pohlig-Hellman algorithm [40] that solves this problem with time complexity $O(\sqrt{p_{max}} \log p)$. In the case of the isogeny-based cryptosystem [14] $|S|$ is 3-smooth, and so this algorithm is practical.

Algorithm 2. *Pohlig-Hellman for the two-dimensional ECDLP*

```
Input: Points $P, Q$ which generate $E[\ell^e]$, and $R \in E[\ell^e]$.
Output: $m, n \in \mathbb{Z}/\ell^e\mathbb{Z}$ such that $[m]P + [n]Q = R$.
$A, B \leftarrow []$;
for $i$ from 1 to $e$ do
    for $(x, y)$ in $\mathbb{Z}/\ell\mathbb{Z}$ do
        if $[\ell^{e-i}]R - \sum_{j=1}^{i-1}[\ell^{e-1+j-i}]R == [\ell^{i-1}]([x]P + [y]Q)$ then
            $A[i] \leftarrow x$;
            $B[i] \leftarrow y$;
        end if
    end for
end for
return $(m, n) = (\sum_{i=1}^{e} A[i]\ell^{i-1}, \sum_{i=1}^{e} B[i]\ell^{i-1})$.
```

Another approach to this problem is to use the Weil pairing. Observe,

$$e(P, R) = e(P, [m]P + [n]Q) = e(P, P)^m e(P, Q)^n = e(P, Q)^n$$

by applying 4.1, and

$$e(Q,R) = e(Q,[m]P+[n]Q) = e(Q,P)^m e(Q,Q)^n = e(Q,P)^m.$$

By computing the pairings $e(P,R)$, $e(P,Q)$, $e(Q,R)$ and $e(Q,P)$ (which is simply $e(P,Q)^{-1}$), the two-dimensional elliptic curve discrete logarithm problem can be reduced to two separate instances of the discrete logarithm problem in the group of N-roots of unity.

In languages with a fast implementation of the Weil pairing (e.g. Magma) the pairing solution is the more efficient of the two mentioned options.

4.1.4 Montgomery Form and Change of Coordinates

Definition 4.1.2. *An elliptic curve over a field K in Montgomery form is a non-singular curve given by*

$$E : By^2 = x^3 + Ax^2 + x,$$

for $A, B \in K$.

The Montgomery form of an elliptic curve was first discovered in 1987 [28] as a way to speed up elliptic curve arithmetic. Elliptic curves in Montgomery form also protect against timing attacks [22], which are a threat when attackers are able to accurately measure the time taken to compute private-key operations. While the timing of these operations may still be determined, they cannot be used to reveal any information about the private-key when the elliptic curve is in Montgomery form [29].

A side-channel attack performed during the computation of $[m_A]P_A + [n_A]Q_A$ in the supersingular isogeny scheme could reveal Alice's private key. For this reason, the isogeny-based cryptosystems proposed [14] use the Montgomery form of the curve. As the compression techniques are presented in short Weierstraß form, below are the conversions between the two forms.

Montgomery to Weierstraß [14]: Let $E : By^2 = x^3 + Ax^2 + x$. Setting $\bar{x} = x/B$ and $\bar{y} = y/B$ gives the long Weierstraß equation

$$\bar{y}^2 = \bar{x}^3 + \frac{A}{B}\bar{x}^2 + \frac{1}{B^2}\bar{x}.$$

To rewrite the long Weierstraß equation in short Weierstraß form, one sets $\widetilde{x} = \bar{x} + \frac{A}{3B}$ and $\widetilde{y} = \bar{y}$. This gives

$$\widetilde{y}^2 = \widetilde{x}^3 + \frac{3-A^2}{3B^2}\widetilde{x} + \frac{2A^3-9A}{27B^3}.$$

Substitution of the two steps gives the transformation $\widetilde{x} = \frac{3x+A}{3B}$ and $\widetilde{y} = \frac{y}{B}$.

Weierstraß to Montgomery [14]: Let $E : y^2 = x^3 + ax + b$, and suppose E has a K-rational point P_4 of order 4. Let $P_2 = (x_p, y_p) := [2]P_4$ and note that $[2]P_2 = O$, or $P_2 = -P_2$. As $-P_2 = (x_p, -y_p)$ it follows that $y_p = -y_p$ and so $y_p = 0$. From this, we observe that $y_p^2 = 0$ and so $x_p^3 + ax_p + b$ must also equal 0. Now if we set $\bar{x} = x - x_p$ and $\bar{y} = y$, then the defining equation of E becomes:

$$\bar{y}^2 = \bar{x}^3 + (3x_p)\bar{x}^2 + (3x_p^2 + a)\bar{x} + (x_p^3 + ax_p + b) = \bar{x}^3 + (3x_p)\bar{x}^2 + (3x_p^2 + a)\bar{x}.$$

We have mapped from short Weierstraß form to long Weierstraß form. Finally, we notice that the new x-coordinate of P_4 has become $x(P_4) - x_p$; call this β^{-1}. Setting $\widetilde{x} = \bar{x}\beta$, $\widetilde{y} = \bar{y}\beta$ and multiplying by β^3 changes the equation of E to the Montgomery form:

$$\beta\widetilde{y}^2 = \widetilde{x}^3 + a\beta\widetilde{x}^2 + \widetilde{x}.$$

Therefore, the changes of coordinates required is $\widetilde{x} = \beta(x - x_p) = \frac{x-x_p}{x(P_4)-x_p}$ and $\widetilde{y} = \beta y = \frac{y}{x(P_4)-x_p}$.

4.2 Compression and Decompression

This section will outline the general techniques that will be used for compression/decompression in each of the isogeny-based cryptosystems. Recall Alice's public-key in the supersingular cryptosystem of [14] is $(\alpha, \beta, \phi_A(P_B), \phi_A(Q_B))$ with corresponding secret-key (m_A, n_A), when her elliptic curve is defined by $E_A : y^2 = x^3 + \alpha x + \beta$. We will compress each of the two components of the public-key individually: the elliptic curve and the auxiliary torsion points.

4.2.1 Compression

Suppose Alice sends $j(E_A)$ instead of α and β. From 2.2.5 we know that the j-invariant of E_A determines the curve up to isomorphism. Replacing α and β by $j(E_A)$ we will reduce the bit-size by half, but only convey an isomorphism class of elliptic curves. By 2.2.2, there is a canonical elliptic curve associated with each j-invariant; define E_j to be the elliptic curve associated with $j(E_A)$. When Bob receives $j(E_A)$ he can compute E_j.

Before compressing the auxiliary points Alice must first map them to E_j, as this will be the elliptic curve Bob computes. To do this, Alice must compute $j(E_A)$, E_j, the isomorphism $\psi_A : E_A \to E_j$, and the image of her points in ψ_A.

Now that compression of the curve is accomplished we note that both $\psi_A(\phi_A(P_B))$ and $\psi_A(\phi_A(Q_B))$ have order 3^{e_B} (both morphisms preserve order), meaning they are elements of the subgroup $E_j[3^{e_B}]$. If Alice and Bob could compute the same basis $\{B_1, B_2\}$ for $E_j[3^{e_B}]$ then they could use integers in $\mathbb{Z}/3^{e_B}\mathbb{Z}$ to represent $\phi_A(P_B)$ and $\phi_A(Q_B)$ as linear combinations of B_1 and B_2:

$$\phi_A(P_B) = \alpha_1 B_1 + \beta_1 B_2, \text{ and } \phi_A(Q_B) = \alpha_2 B_1 + \beta_2 B_2.$$

These steps are summarized in the following procedure to compress Alice's public key.

Input: $(\alpha, \beta, \phi_A(P_3), \phi_A(Q_3))$

1. Let $E_A : y^2 = x^3 + \alpha x + \beta$
2. Compute $j = j(E_A)$ with 2.2.6.
3. Compute the canonical curve E_j associated to j with 2.2.2.
4. Compute the isomorphism $\psi_A : E_A \to E_j$.
5. Compute $R_1 = \psi_A(\phi_A(P_B))$ and $R_2 = \psi_A(\phi_A(Q_B))$.
6. Compute the basis $\{B_1, B_2\}$ for $E_j[3^{e_B}]$ with 4.1.2.
7. Compute $\alpha_1, \beta_1, \alpha_2, \beta_2 \in \mathbb{Z}/3^{e_B}\mathbb{Z}$ with 2 such that
$$R_1 = \alpha_1 B_1 + \beta_1 B_2, R_2 = \alpha_2 B_1 + \beta_2 B_2.$$

Output: $(j, \alpha_1, \beta_1, \alpha_2, \beta_2)$.

4.2.2 Decompression

The procedure for decompression uses the same techniques as compression, but does not need to solve the discrete log problem.

Input: $(j, \alpha_1, \beta_1, \alpha_2, \beta_2)$

1. Compute the canonical curve E_j associated to j with 2.2.2.
2. Compute the basis $\{B_1, B_2\}$ for $E_j[3^{e_B}]$ with 4.1.2.
3. Compute $R_1 = \alpha_1 B_1 + \beta_1 B_2$, and $R_2 = \alpha_2 B_1 + \beta_2 B_2$.

Output: (E_j, R_1, R_2)

Chapter 5

Isogeny-Based Cryptosystems Using Compression

This chapter contains results from “Key compression for isogeny-based cryptosystems” by R. Azarderakhsh, D. Jao, K. Kalach, B. Koziel, and C. Leonardi, to appear in AsiaPKC 2016. In this section, we explain how to apply the compression techniques (Section 4.2) within the framework of three public-key isogeny-based cryptosystems: key exchange, public-key encryption, and zero-knowledge proof of identity.

5.1 Key-Exchange

Setup: Fix $\mathbb{F}_{p^2}$ as the field of definition, where p is a prime number of the form $2^{e_A}3^{e_B} \cdot f \pm 1$, where f is chosen so that p is prime. Use Bröker's algorithm to efficiently compute a supersingular curve, $E : y^2 = x^3 + a_4x + a_6$, defined over $\mathbb{F}_{p^2}$, having cardinality $(p \mp 1)^2 = (2^{e_A}3^{e_B} \cdot f)^2$. Fix a non-square element $d \in \mathbb{F}_{p^2}$. Lastly, fix a basis P_A, Q_A which generates $E[2^{e_A}]$, and a basis P_B, Q_B which generates $E[3^{e_B}]$.

Compression of public information: Alice chooses two random elements $m_A, n_A \in_R \mathbb{Z}/2^{e_A}\mathbb{Z}$ not both divisible by 2. Using Vélu's formulas, she computes $E_A, \phi_A(P_B)$, and $\phi_A(Q_B)$ where $\ker(\phi_A) = \langle [m_A]P_A + [n_A]Q_A \rangle$. Normally, Alice would just send $E_A, \phi_A(P_B)$, and $\phi_A(Q_B)$ to Bob, but we now add key compression. Alice computes the canonical curve E_{j_A} from $j(E_A)$, along with

$$E^*_{j_A} : y^2 = x^3 - 27c_1(E_{j_A})x - 54c_2(E_{j_A})$$

to put E_{j_A} in short Weierstraß form. If E_A is not isomorphic to $E^*_{j_A}$ over $\mathbb{F}_{p^2}$, then she sets $T_A = 1$ and computes the twist

$$E^*_A : y^2 = x^3 + d^2 a_4(E^*_{j_A})x + d^3 a_6(E^*_{j_A})$$

of $E^*_{j_A}$. Otherwise she sets E^*_A to $E^*_{j_A}$, and T_A to 0.

Next, Alice computes the isomorphism $\psi_A : E_A \to E^*_A$ and canonical basis $\{R_1, R_2\}$ for $E^*_A[3^{e_B}]$. Finally, she solves the 2-dimensional discrete log problem to determine $\alpha_1, \beta_1, \alpha_2, \beta_2 \in \mathbb{Z}/3^{e_B}\mathbb{Z}$ such that

$$\alpha_1 R_1 + \beta_1 R_2 = \psi_A(\phi_A(P_B)), \text{ and } \alpha_2 R_1 + \beta_2 R_2 = \psi_A(\phi_A(Q_B)).$$

Alice's compressed public information is the tuple $(j(E_A), \alpha_1, \beta_1, \alpha_2, \beta_2, T_A)$, and her private key is still (m_A, n_A). She exchanges this information with Bob, who in turn sends Alice his public information $(j(E_B), \gamma_1, \kappa_1, \gamma_2, \kappa_2, T_B)$, where

- $\{S_1, S_2\}$ is the canonical basis for $E^*_B[2^{e_A}]$,
- $\gamma_1 S_1 + \kappa_1 S_2 = \psi_B(\phi_B(P_A))$,
- $\gamma_2 S_1 + \kappa_2 S_2 = \psi_B(\phi_B(Q_A))$,
- $\ker(\phi_B) = \langle [m_B]P_B + [n_B]Q_B \rangle$,
- $\psi_B \colon E_B \to E^*_B$,

and T_B is 1 if a twist is required, and 0 otherwise.

Decompression and computing a shared secret key: Alice determines E^*_B by computing the canonical curve associated with $j(E_B)$, putting it in short Weierstraß form, and computing a quadratic twist depending on the bit from Bob. After computing the canonical basis $\{S_1, S_2\}$ for $E^*_B[2^{e_A}]$, Alice uses $\gamma_1, \kappa_1, \gamma_2, \kappa_2$ to compute $\psi_B(\phi_B(P_A))$ and $\psi_B(\phi_B(Q_A))$. Using Vélu's formulas once more, Alice computes the isogeny

$$\phi'_A : E^*_B \to E_{AB},$$

with $\ker(\phi'_A) = \langle [m_A]\psi_B(\phi_B(P_A)) + [n_A]\psi_B(\phi_B(Q_A)) \rangle$. After Bob determines the curve

$$E_{BA} = E^*_A / \langle [m_B]\psi_A(\phi_A(P_B)) + [n_B]\psi_A(\phi_A(Q_B)) \rangle$$

by performing his analogous decompression, both Alice and he possess the shared secret key $j(E_{AB}) = j(E_{BA}) \in \mathbb{F}_{p^2}$.

A comparison of Figure 3.3 with Figure 5.1 illustrates how the additional steps of compression change the underlying scheme.

$$\begin{tikzcd}
 & E_A \arrow[r, "\psi_A"] & E_A^* \arrow[dr, "ker(\phi'_B) = \langle [m_B]\psi_A(\phi_A(P_B)) + [n_B]\psi_A(\phi_A(Q_B))\rangle"] \arrow[dd, dashed, shift left] & \\
E \arrow[ur, "ker(\phi_A) = \langle [m_A]P_A + [n_A]Q_A\rangle"] \arrow[dr, "ker(\phi_B) = \langle [m_B]P_B + [n_B]Q_B\rangle"'] & & & E_{AB} \cong E_{BA} \\
 & E_B \arrow[r, "\psi_B"'] & E_B^* \arrow[ur, "ker(\phi'_A) = \langle [m_A]\psi_B(\phi_B(P_A)) + [n_A]\psi_B(\phi_B(Q_A))\rangle"'] \arrow[uu, dashed, shift left] &
\end{tikzcd}$$

Figure 5.1: Supersingular key exchange diagram with compression

5.2 Encryption

Setup: In addition to the setup from the key-exchange system above, this cryptosystem also requires a family of hash functions, $\mathcal{H} = \{H_k : k \in K\}$, from $\mathbb{F}_{p^2}$ to the message space $\{0,1\}^w$, indexed by a finite set K.

Compressed key generation: Choose two random elements $m_A, n_A \in_R \mathbb{Z}/2^{e_A}\mathbb{Z}$ not both divisible by 2, and a random $k \in_R K$. As in Section 5.1, compute and publish the tuple $(j(E_A), \alpha_1, \beta_1, \alpha_2, \beta_2, T_A)$ as the public key, and store the private key tuple (m_A, n_A, k).

Decompression: Given a public key $(j(E_A), \alpha_1, \beta_1, \alpha_2, \beta_2, T_A, k)$ it is described above (5.1) how to decompress to the tuple $(E_A^*, \psi_A(\phi_A(P_B)), \psi_A(\phi_A(Q_B)), k)$.

Encryption: Given the decompressed public key $(E_A^*, \psi_A(\phi_A(P_B)), \psi_A(\phi_A(Q_B)), k)$, and message $m \in \{0,1\}^w$, the sender chooses $m_B, n_B \in_R \mathbb{Z}/3^{e_B}\mathbb{Z}$ not both divisible by 3. Next, as in Section 5.1, the sender computes $E_B, \phi_B(P_A), \phi_B(Q_A), E_B^*, T_B, \psi_B, \{S_1, S_2\}$, the coefficients $\gamma_1, \kappa_1, \gamma_2, \kappa_2 \in \mathbb{Z}/2^{e_A}\mathbb{Z}$ such that $\gamma_1 S_1 + \kappa_1 S_2 = \psi_B(\phi_B(P_A))$, and $\gamma_2 S_1 + \kappa_2 S_2 = \psi_B(\phi_B(Q_A))$, and $\phi'_B : E_A^* \to E_{BA}$, with $\ker(\phi'_B) = \langle [m_B]\psi_A(\phi_A(P_B)) + [n_B]\psi_A(\phi_A(Q_B))\rangle$.

Finally, the sender sets

$$c = H_k(j(E_{BA})) \oplus m.$$

The ciphertext is the tuple $(c, j(E_B), \gamma_1, \kappa_1, \gamma_2, \kappa_2, T_B)$.

Decryption: Given a ciphertext $(c, j(E_B), \gamma_1, \kappa_1, \gamma_2, \kappa_2, T_B)$ and private key (m_A, n_A, k), compute the curve E_B^* from $j(E_B)$ and T_B, the points $\psi_B(\phi_B(P_A))$ and $\psi_B(\phi_B(Q_A))$ from $\gamma_1, \kappa_1, \gamma_2, \kappa_2$, and the isogeny $\phi'_A : E_B^* \to E_{AB}$, with $\ker(\phi'_A) = \langle [m_A]\psi_B(\phi_B(P_A)) + [n_A]\psi_B(\phi_B(Q_A))\rangle$. The plaintext is

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

5.3 Zero-Knowledge Proof of Identity

Here we outline how to use the compression technique to reduce the amount of information sent in each round of an isogeny-based zero-knowledge proof of identity. Let p be a prime number of the form $2^{e_A}3^{e_B} \cdot f \pm 1$. Throughout this subsection let E^* denote the canonical curve associated to the j-invariant of the isomorphism class of E.

Secret parameters: A supersingular curve E defined over $\mathbb{F}_{p^2}$, a primitive 2^{e_A}-torsion point S defining an isogeny $\phi : E \to E/\langle S\rangle$, and an isomorphism $\phi_0 : E/\langle S\rangle \to (E/\langle S\rangle)^*$.

Public parameters: The elliptic curves E and $E/\langle S\rangle$, generators P, Q for $E[2^{e_A}]$, and the points $\phi_0(\phi(P)), \phi_0(\phi(Q))$.

Identification: Repeat m times:

1. Peggy picks $R \in E[3^{e_B}]$ and, using Vélu's formulas, computes the elliptic curves

   $$E/\langle R\rangle, (E/\langle R\rangle)^*, E/\langle S, R\rangle,$$

   and the isogenies

   $$\psi : E \to E/\langle R\rangle, \phi' : (E/\langle S\rangle)^* \to E/\langle S, R\rangle, \psi' : (E/\langle R\rangle)^* \to E/\langle S, R\rangle.$$

   She also computes the isomorphism

   $$\psi_0 : E/\langle R\rangle \to (E/\langle R\rangle)^*.$$

2. Peggy sends $j(E/\langle R\rangle)$ and $j(E/\langle S,R\rangle)$ to Victor.

3. Victor randomly selects a bit b and sends it to Peggy.

4. If $b=0$, then Peggy computes the canonical bases $\{B_1,B_2\}$ for $E[3^{e_B}]$ and $\{B_1',B_2'\}$ for $(E/\langle S\rangle)^*[3^{e_B}]$. Peggy then sends $\alpha_1,\beta_1,\alpha_2,\beta_2\in Z/3^{e_B}Z$ to Victor, where $\alpha_1 B_1+\beta_1 B_2=R$, and $\alpha_2 B_1'+\beta_2 B_2'=\phi_0(\phi(R))$.

5. If $b=1$, then Peggy computes the canonical basis $\{B_1'',B_2''\}$ for $(E/\langle R\rangle)^*[2^{e_A}]$. Peggy then sends $\alpha_3,\beta_3\in Z/2^{e_A}Z$ to Victor, such that

$$\alpha_3 B_1''+\beta_3 B_2''=\psi_0(\psi(S)).$$

Chapter 6

Complexity

This chapter contains results from “Key compression for isogeny-based cryptosystems” by R. Azarderakhsh, D. Jao, K. Kalach, B. Koziel, and C. Leonardi, to appear in AsiaPKC 2016. The compression and decompression operations require no private key material, and indeed the operations can be unilaterally performed by any party or any observer. Accordingly, the operations cannot have any effect whatsoever on the security of the underlying scheme.

6.1 Space Requirements

For supersingular isogeny-based encryption to achieve a k-bit security level against a quantum adversary, p needs to be approximately $6k$-bits (3.2.1). For example, for 128-bit security, take $p \approx 2^{6 \cdot 128} = 2^{768}$. Hence, each of 2^{e_A} and 3^{e_B} need to have $3k$ bits. Alice’s public key is $j(E_A) \in \mathbb{F}_{p^2}$ ($12k$ bits) and the four elements of $\mathbb{Z}/3^{e_B}\mathbb{Z}$ ($3k$ bits for each), giving a total of $24k$ bits for the k-bit level of security. Our compressed keys require $4 \log p + 1$ bits to store (the extra bit is to convey if a twist was used or not).

In Table 6.1 we present a comparison between the size of our compressed keys and the minimum key size required for the 128-bit and 256-bit security levels from other major families of post-quantum encryption primitives: Lattice-based and Code-based.

The key sizes in Table 6.1 for lattice and code-based schemes are based on classical attackers only, because these schemes are often useful even in a classical setting, so the published security analysis are classical. Quantum attacks may be slightly faster: for

Table 6.1: Public key sizes in bits

	Security level	
Scheme	128-bit	256-bit
NTRU [20]	4 939	11 957
Ring-LWE [35]	7 498	15 690
McEliece (Goppa)[1]	1 991 880	9 276 241
McEliece (QC-MDPC) [27]	9 857	32 771
Isogenies	3 073	6 145

instance [16] demonstrates a 2^{112} quantum attack against the "128-bit" NTRU parameters. By contrast, isogeny-based cryptosystems are motivated mainly by post-quantum applications, so the security analysis and key sizes for isogeny-based cryptosystems assume quantum attackers by default. Against classical attacks, key sizes for isogeny-based cryptosystems can be further reduced by 33%.

6.2 Computational Requirements

It is computationally easy to compute j-invariants, canonical curves from j-invariants, quadratic twists, scalar multiples of points on elliptic curves, isomorphisms between elliptic curves, sums of points on elliptic curves, and to put elliptic curves in Weierstraß form. These costs are relatively negligible compared to the costs of computing a basis and performing discrete logarithms, so we ignore them.

The basis computation is done probabilistically. For a random point P, the point $[2^{e_A}f]P$ has order 2^{e_A} with probability $1/e_A$. Hence, the expected number of point multiplications required is $O(e_A)$. Since the primes 2 and 3 will always be much smaller than their exponents e_A, e_B, we have that $\sqrt{2} \lll e_A$. Then to solve two discrete logarithms in $\mathbb{F}_{p^2}$ one can use Pohlig-Hellman twice, each with $O(e_A^2)$ elliptic curve point multiplications [30]. To compress her information, Alice must perform both of the above steps. After she has exchanged information with Bob, she performs another basis computation, with cost $O(e_A)$ (using $\sqrt{2} \lll e_A$ again), to decompress. This gives a total theoretical runtime of $O(e_A^2)$.

We implemented the compression and decompression routines in Magma Computational Algebra System and C (by Brian Koziel) in order to benchmark these operations

$p = 2^{e_A}3^{e_B}f \pm 1$	$2^{258}3^{161}186 - 1$	$2^{341}3^{218}3 - 1$	$2^{386}3^{242}2 - 1$	$2^{514}3^{323}353 - 1$
Classical security (bits)	128	168	192	256
Quantum security (bits)	85	112	128	168
Alice Compression (sec)	0.238	0.873	1.988	3.318
Bob Compression (sec)	0.248	0.641	1.403	2.404
Alice Decompression (sec)	0.130	0.270	0.530	1.020
Bob Decompression (sec)	0.100	0.300	0.650	1.100

Table 6.2: Runtimes of our implementation for compression in Magma

$p = 2^{e_A}3^{e_B}f \pm 1$	$2^{258}3^{161}186 - 1$	$2^{386}3^{242}2 - 1$	$2^{514}3^{323}353 - 1$
Classical security (bits)	128	192	256
Quantum security (bits)	85	128	168
Alice Compression (sec)	0.453	1.518	3.697
Bob Compression (sec)	0.576	1.930	4.639
Alice Decompression (sec)	0.062	0.149	0.264
Bob Decompression (sec)	0.056	0.128	0.247

Table 6.3: Runtimes of our implementation for compression in C

and obtain an upper bound on their computational cost. Tables 6.2 and 6.3 gives the empirically measured runtimes of our compression and decompression implementations for four different choices of parameters. Magma computations were performed on the University of Waterloo's Biglinux server pool which consists of a 2.3 GHz SGI Altix XE H2106-G7 and a 2.3 GHz Dell PowerEdge R815. C computations were performed on an i7-4790k processor at 4.0 GHz.

Weil pairings have been implemented in Magma, and so the two-dimensional ECDLP computation is much faster than in C. For this reason, compression is more efficient in Magma. However, C was more efficient in all other aspects, and since decompression does not require a discrete log computation, the C implementation of decompression runs in less time.

We remark that, assuming cheap storage and expensive bandwidth, the logical strategy is to store both compressed and uncompressed copies of the key, and transmit only compressed copies. In this scenario, the computational costs of compression and decompression are incurred only once per key; for compression, once per key for the lifetime of the key, and for decompression, once per key per recipient. Unlike space-saving strategies with NTRU and LWE, public key compression imposes no per-message encryption or decryption overhead. NTRU and LWE also have a nonzero probability of decryption failure (they are based on adding error vectors into the ciphertext, and occasionally the error overcomes the intended signal), which causes a tradeoff between security, efficiency, and error-rate, and represents a limiting factor in reducing key size for these two schemes, whereas isogeny-based encryption schemes have no possibility of mathematical error.

Chapter 7

Applications and Future Work

The most relevant application today for reduced key sizes is when there is an upper bound in the protocol on the amount of bits allowed to be sent at a time. Popular systems Tor and Bitcoin have this restriction in place, and in systems like SSH and TLS it is also usually preferable to send as little data as possible.

Tor is currently the most widely deployed software for anonymous communication, which it achieves by directing internet traffic through thousands of relays. Each cell of data in Tor's onion rooting network must be less than 514 bytes [13], and public keys are transmitted within blocks of this size. Compared to isogeny-based cryptography, no other known quantum-resistant cryptosystem can function well under this restriction (recently, [31] showed how to incorporate NTRUEncrypt into the NTor protocol, but only after increasing the cell size).

Both SSH and SSL currently provide confidentiality or privacy using public-key encryption schemes which are not secure against quantum attacks. Isogeny-based public-key encryption using our compression method is extremely space efficient, providing a strong candidate for quantum-resistant deployment of these protocols.

Currently, the most time consuming step is solving the two-dimensional discrete logarithm problem, and so any developments for this computation will directly affect the runtime of compression. There are also opportunities in the compression steps for assembly optimizations and other performance enhancements.

The compression itself may also be subject to further optimization in the form of larger reductions. We have recommended the use of the j-invariant for compression of the elliptic curve, however there are other options. One such way is to fix a coefficient of

$E_A : y^2 = x^3 + ax + b$, say $a = -3$, determine which elliptic curve in the isomorphism class $j(E_A)$ satisfies this property, and send the corresponding b coefficient. However, for a fixed j, there exists an elliptic curve in this class with $a = -3$ if and only if $1 - \frac{1728}{j}$ is a quadratic residue in $\mathbb{F}_{p^2}$ $\left(\text{in which case the other coefficient is } b = 2\sqrt{1 - \frac{1728}{j}}\right)$. To overcome this, define a global parameter $\beta \in \mathbb{F}_{p^2}$ to be a quadratic non-residue, but a cubic residue. The product $\beta(1 - \frac{1728}{j})$ will be a quadratic-residue, and so Alice only needs to send the b coefficient $2\sqrt{\beta(1 - \frac{1728}{j})}$ and an additional bit to alert Bob that she used β. Bob can now compute the elliptic curve $E'_A : y^2 = x^3 - 3\beta^{1/3}x + 2\sqrt{\beta(1 - \frac{1728}{j(E_A)})}$, which is isomorphic to E_A. For this to work, a specific cube root of β must also be publicly determined. This alternate method required the same amount of space as sending the j-invariant, slightly increases the amount of computation Alice must perform, and decreases the amount of computation Bob must do to determine the short Weierstraß equation of Alice's public elliptic curve.

References

[1] P. Barreto, F. Biasi, R. Dahab, J. López-Hernández, E. de Morais, A. Salina de Oliveira, G. Pereira, and J. Ricardini. *Open Problems in Mathematics and Computational Science*, chapter A Panorama of Post-quantum Cryptography, pages 387–439. Springer, 2014.

[2] J.F. Biasse, D. Jao, and A. Sankar. A quantum algorithm for computing isogenies between supersingular elliptic curves. In *Progress in Cryptology – INDOCRYPT 2014*, volume 8885 of *Lecture Notes in Computer Science*, pages 428–442. Springer International Publishing, 2014.

[3] A. Bostan, F. Morain, B. Salvy, and E. Schost. Fast algorithms for computing isogenies between elliptic curves. *Mathematics of Computation*, 77(263):1755–1778, 2008.

[4] R. Bröker. Constructing supersingular elliptic curves. *J. Comb. Number Theory*, 1:269–273, 2009.

[5] R. Bröker, J. Belding, A. Enge, and K. Lauter. Computing Hilbert class polynomials. *Algorithmic Number Theory Symposium*, VIII:282–295, 2008.

[6] R. Bröker, D. Charles, and K. Lauter. Evaluating large degree isogenies and applications to pairing based cryptography. In *Pairing-Based Cryptography - Pairing 2008*, volume 5209 of *LNCS*, pages 100–112, 2008.

[7] A. Childs, D. Jao, and V. Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *J. Math. Cryptol*, 8:1–29, 2014.

[8] H. Cohen. *A Course in Computational Algebraic Number Theory*, volume 138 of *Graduate Texts in Mathematics*. Springer-Verlag, 1993.

[9] H. Cohen and G. Frey, editors. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. Chapman and Hall/CRC, 2005.

[10] I. Connell. Elliptic curve handbook. http://www.math.mcgill.ca/connell/public/, 1999.

[11] J.M. Couveignes. Hard homogeneous spaces. https://eprint.iacr.org/2006/291.pdf, 2006.

[12] C. Delfs and S. Galbraith. Computing isogenies between supersingular elliptic curves over Fp. In *Designs, Codes and Cryptography*, volume 78, pages 425–440, 2016.

[13] R. Dingledine and N. Mathewson. Tor protocol specification.

[14] L. De Feo, D. Jao, and J. Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology*, 8(3):209–247, 2014.

[15] D. Fishbein. Machine-level software optimization of cryptographic protocols. Master's thesis, University of Waterloo, 2014.

[16] S. Fluhrer. Quantum cryptanalysis of NTRU. Cryptology ePrint Archive, Report 2015/676, 2015.

[17] S. Galbraith. *Mathematics of Public Key Cryptography.* Cambridge University Press, 2012.

[18] S. Galbraith, F. Hess, and N.P. Smart. Extending the GHS Weil descent attack. In *Advances in Cryptology - EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 29–44, 2002.

[19] S. Galbraith and A. Stolbunov. Improved algorithm for the isogeny problem for ordinary elliptic curves. *Applicable Algebra in Engineering, Communication and Computing*, 24(2):107–131, 2013.

[20] P. Hirschhorn, J. Hoffstein, N. Howgrave-Graham, and W. Whyte. Choosing NTRUEncrypt parameters in light of combined lattice reduction and MITM approaches. In *In Proc. ACNS 2009, LNCS 5536*, pages 437–455. Springer-Verlag, 2009.

[21] V. Jacques. Isogénies entre courbes elliptiques. *C. R. Acad. Sci. Paris Sér. A-B*, 273:A238–A241, 1971.

[22] P. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Advances in Cryptology - CRYPTO '96*, volume 1109 of *LNCS*, pages 104–113. Springer, 1996.

[23] G. Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM Journal on Computing*, 35:170–188, 2005.

[24] S. Lang. *Elliptic Functions*, volume 112 of *Graduate Texts in Mathematics*. Springer, 1987.

[25] R. Lercier and F. Morain. Counting the number of points on elliptic curves over finite fields: strategies and performances. In *Advances in Cryptology - EUROCRYPT '95*, volume 921 of *LNCS*, pages 79–94. Springer, 1995.

[26] V. Miller. Short programs for functions on curves. *Unpublished*, 1986.

[27] R. Misoczki, J.P. Tillich, N. Sendrier, and P. Barreto. MDPC-McEliece: New McEliece Variants from Moderate Density Parity-Check Codes. Cryptology ePrint Archive, Report 2012/409, 2012.

[28] P. Montgomery. Speeding the Pollard and elliptic curve methods of factorization. *Mathematics of Computation*, 48(177):243–264, 1987.

[29] K. Okeya, H. Kurumatani, and K. Sakurai. Elliptic Curves with the Montgomery-Form and Their Cryptographic Applications. In *Public Key Cryptography*, volume 1751 of *LNCS*, pages 238–257, 2000.

[30] S. Pohlig and M. Hellman. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance. In *IEEE Transactions on Information Theory*, volume 24, pages 106–110, 1978.

[31] J. Schanck, W. Whyte, and Z. Zhang. A quantum-safe circuit-extension handshake for Tor. Cryptology ePrint Archive, Report 2015/287, 2015.

[32] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26:1484–1509, 1997.

[33] J. Silverman. *The Arithmetic of Elliptic Curves*. Springer, 1986.

[34] J. Silverman. *Advanced topics in the arithmetic of elliptic curves*. Number 151 in Lecture Notes in Mathematics. Springer, 1994.

[35] V. Singh. A practical key exchange for the internet using lattice cryptography. Cryptology ePrint Archive, Report 2015/138, 2015.

[36] B. Smith. Mappings of elliptic curves. Webpage, 2008.

[37] A. Stolbunov. Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves. *NISK*, pages 97–109, 2009.

[38] S. Tani. Claw finding algorithms using quantum walk. *Theoretical Computer Science*, 410:5285–5297, 2009.

[39] J. Tate. Endomorphisms of abelian varieties over finite fields. *Inventiones mathemticae*, 2(2):134–144, 1966.

[40] E. Teske. The Pohlig-Hellman method generalized for group structure computation. *J. Symbolic Computation*, pages 521–534, 1999.

[41] L. Washington. *Elliptic Curves: Number Theory and Cryptography.* Chapman and Hall/CRC, 2008.

[42] W. Waterhouse. Abelian varieties over finite fields. *Annales scientifiques de l'École Normale Supérieure*, 2(4):521–560, 1969.