

Isogeny-Based Zero-Knowledge Proofs and Their Applications

by

Youcef Mokrani

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

Waterloo, Ontario, Canada, 2025

© Youcef Mokrani 2025

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Benjamin Smith
Research Scientist
Inria

Supervisor: David Jao
Professor
Department of Combinatorics and Optimization
University of Waterloo

Internal Members: Douglas Stebila
Associate Professor
Department of Combinatorics and Optimization
University of Waterloo
Samuel Jaques
Assistant Professor
Department of Combinatorics and Optimization
University of Waterloo

Internal-External Member: Yu-Ru Liu
Professor
Department of Pure Mathematics
University of Waterloo

Author's Declaration

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

Isogeny-based cryptography is one of the main avenues of research in post-quantum cryptography. The fundamental idea of this branch is that there is currently no known efficient algorithm to compute an isogeny between two supersingular elliptic curves, even when one has access to a quantum computer. However, this pure primitive keeps too much information about the secret isogeny hidden to be directly applied to most applications. As such, almost every protocol based on isogeny reveals some extra information about the secret isogeny. This is famously the case for Supersingular Isogeny Diffie–Hellman (SIDH), which transmits the mapping of the isogeny on a torsion subgroup, the degree of the isogeny, and the endomorphism ring of the domain curve.

The recent polynomial-time attacks on SIDH have shown that leaking the torsion subgroup mapping gives away too much information to an attacker. Because of this, the SIDH variants proposed to resist these attacks all mask the mapping in some way. However, less attention has been paid to the other types of information that SIDH and most of its new variants transmit. This is especially worrying when it comes to the endomorphism ring of the starting curve, as it was shown multiple times that it can lead to easier attacks. In fact, the first of the recent polynomial-time attacks on SIDH made direct use of the endomorphism ring. Also, before these attacks fully broke SIDH, Petit showed that knowledge of the endomorphism ring could lead to a polynomial-time attack on SIDH when the parameter sets were unbalanced. Castryck and Vercauteren recently showed that the same attacks on unbalanced parameters with known endomorphism rings can be extended to some of the new SIDH variants, such as M-SIDH.

An interesting fact about SIDH variants is that most of them do not explicitly need to transmit the endomorphism ring. Most implementations do so because it is simpler. The goal of this thesis is therefore to further study the case for masking the endomorphism ring of the domain curve for SIDH variants. We start by showing that, for well-chosen parameter sets, working with a random starting curve can never lead to a loss of security. This thesis also explores the use of multiparty computations to generate curves of unknown endomorphism rings. Finally, we present a new set of zero-knowledge proofs for SIDH variants that do not require knowledge of any endomorphism rings and can be made to mask the degree of the secret isogeny.

Acknowledgements

I would like to thank my advisor, David Jao, whose advice has been crucial, not only for this thesis, but for every research project during my stay in Waterloo. His deep knowledge of isogenies and the subjects relating to them has been vital throughout my PhD, and his willingness to always entertain my numerous questions is something I appreciate very much.

I would also like to thank the members of my Examining Committee, Benjamin Smith, Douglas Stebila, Samuel Jaques and Yu-Ru Liu, for their detailed and helpful advice, which led to the vast improvement of this thesis.

Finally, I need to thank my family for their immense support both through the last five years in Waterloo and the times before. I would not even have been able to start a PhD if it was not been able to rely on them repeatedly.

Table of Contents

Examining Committee	ii
Author's Declaration	iii
Abstract	iv
Acknowledgements	v
List of Abbreviations	ix
List of Symbols	x
1 Introduction	1
2 Background Knowledge	4
2.1 Elliptic Curves over Finite Fields	4
2.2 Isogenies	5
2.3 Level Structure	10
2.4 Supersingular Isogeny Graph	10
2.5 Isogeny Problems	14
2.6 Small Functions for Protocols	17
2.7 SIDH	18

2.8	CSIDH	20
2.9	Other Cryptography Concepts	22
2.10	Other Mathematical Concepts	25
3	Random Self-Reducibility	27
3.1	Main Result	29
3.2	Applications to Isogeny Problems	41
3.2.1	$H = GL_2(\mathbb{Z}/N\mathbb{Z})$	41
3.2.2	$H = \left\{ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right\}$	42
3.2.3	$H = \left\{ \begin{pmatrix} 1 & * \\ 0 & 1 \end{pmatrix} \right\}$	42
3.2.4	$H = \left\{ \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \right\}$	43
3.2.5	$H = \left\{ \begin{pmatrix} * & * \\ 0 & * \end{pmatrix} \right\}$	44
3.2.6	$H = \left\{ \begin{pmatrix} * & 0 \\ 0 & * \end{pmatrix} \right\}$	45
3.3	Conclusion	45
4	Starting Curves	47
4.1	Existing Solutions	48
4.1.1	Signature Schemes	48
4.1.2	Multiparty Key Generation	50
4.2	Assumptions	50
4.3	A New Zero-Knowledge Proof	52
4.3.1	Avoiding the Random Oracle Model	54
4.4	Secure Curve Generation	56
4.4.1	Generating Secure Curves Without a Random Oracle	58
4.5	Curve Randomizer	60
4.6	Conclusion	61

5	Signatures	63
5.1	Masking the Degree	64
5.2	Masked Torsion	68
5.3	Double Masked Subgroup	73
5.4	Masked Degree and Double Subgroup	77
5.5	Conclusion	82
6	Conclusion	83
	References	85

List of Abbreviations

Γ -SIDH Known Level Structure Supersingular Isogeny Problem [15](#)

CCSIDH Computational Commutative Supersingular Isogeny Diffie–Hellman Problem [50](#)

CSIDH Commutative Supersingular Isogeny Diffie–Hellman [20](#)

CSIDHKoE CSIDH Knowledge of Exponent Assumption [51](#)

DMSISOZKP Double Masked Scalar ISOgeny Zero-Knowledge Proof [74](#)

DSSP Decisional SuperSingular Product problem [16](#)

MDISOZKP Masked Degree ISOgeny Zero-Knowledge Proof [65](#)

MDTISOZKP Masked Degree and Torsion ISOgeny Zero-Knowledge Proof [78](#)

MTISOZKP Masked Torsion ISOgeny Zero-Knowledge Proof [69](#)

SIDH Supersingular Isogeny Diffie–Hellman [15](#), [18](#)

SIP Supersingular Isogeny Problem [14](#)

List of Symbols

$\mathbf{End}(E)$ The endomorphism ring of E . 9

$\mathcal{O}$ The point at infinity of an elliptic curve. 4

$E[[N]]$ The set of points of E whose order is exactly N . 5

$G[N]$ N -torsion subgroup of a group G . 5

$\mathbf{GL}_2(\mathbb{Z}/N\mathbb{Z})$ The group of invertible 2×2 matrices with components in $\mathbb{Z}/N\mathbb{Z}$. 10

$\mathcal{G}_{H,\ell}$ The supersingular ℓ -isogeny graph with H -level structure. 12

$\mathcal{G}_\ell$ The supersingular ℓ -isogeny graph. 10

Chapter 1

Introduction

In 1995, Shor [58] proved that it is possible to solve the prime factorization and discrete logarithm problems in polynomial time using a quantum computer. This implied that once someone successfully builds a large enough quantum computer, most of the Internet communications at the time, encrypted using RSA [54] and Diffie–Hellman [29], would stop being secure. This risk is increased by the fact that anybody can record current encrypted messages and decrypt them in the future once they gain access to a quantum computer.

Hence, since this discovery, there has been an effort for new cryptographic schemes that can be used by classical computers while being resistant to attacks by quantum computers. Such protocols are called post-quantum, and there are many primitives on which one can base their security assumptions. The main branches of post-quantum cryptography are lattices, codes, hashes, multivariate polynomials and isogenies. This thesis focuses on this last branch.

Isogenies are non-zero morphisms between elliptic curves that map the identity of the domain curve to the identity of the codomain curve. Isogeny-based schemes utilize the Supersingular Isogeny Problem, which asks an adversary to find an isogeny between two supersingular curves defined over $\mathbb{F}_{p^2}$. As far as current research goes, the best-known algorithms for solving this problem all take exponential time, even when using a quantum computer [12]. However, the number of schemes based solely on the Supersingular Isogeny Problem is limited, and most isogeny-based schemes have to be based on stronger assumptions.

A notable historical example is Supersingular Isogeny Diffie–Hellman (SIDH) [40], a key exchange scheme that works somewhat like the original Diffie–Hellman in order to generate a shared secret. This protocol is based on a variant of the Supersingular Isogeny

Problem where the challenger computes a secret isogeny and, in addition to its domain and codomain, the adversary knows its degree, its mapping on a torsion subgroup, and the endomorphism ring of the domain curve.

Using this primitive, the key encapsulation mechanism SIKE was proposed and was a fourth-round candidate in the NIST Post-Quantum Cryptography Standardization project. However, in 2022, Castryck and Decru [14] showed that SIDH can be broken when the endomorphism ring of the starting curve is known. A similar polynomial time attack was independently developed by Maino and Martindale [42], with the added benefit of obtaining a subexponential time attack when the endomorphism ring is unknown. These attacks were improved again by Robert [55] to obtain a generic attack on SIDH that only requires knowledge of the torsion subgroup and the degree.

These attacks were a major setback for the public’s trust in isogeny-based cryptography and indicated that we still do not fully understand the complexity of isogeny primitives.

Since then, many new SIDH variants have been proposed, with the goal of having similar applications to SIDH while not being vulnerable to the same attacks. This is mainly done by somehow masking the torsion point mapping information. For example, Masked SIDH (M-SIDH) [33] masks the mapping with a secret random scalar. Masked-Degree SIDH (MD-SIDH) [33] does the same, but also masks the degree of the secret isogeny. BinSIDH [8] and FESTA [9] mask each torsion point with its own random scalar, and TerSIDH [8] also does so in addition to masking the degree.

Most of the study of these SIDH variants focuses on the information gained from the knowledge of the torsion point mapping. However, recall that this is not the only information revealed by SIDH. Indeed, SIDH uses a known curve as the domain of its secret isogenies, which makes the endomorphism ring of the starting curve public. This information is non-trivial and seems to help attackers, as shown by the fact that the first break of SIDH uses this information.

This phenomenon is interesting since no step in the original SIDH protocol or its new variants explicitly makes use of the knowledge of any endomorphism ring. Some reasons for why it is revealed are that working with a known fixed curve makes SIDH about 35% more efficient [19], and that generating curves of unknown endomorphism rings is not easy in practice. However, in a time when we need a precise understanding of the security of our protocols, transmitting a piece of information should be avoided if not necessary. Avoiding unnecessary transmissions is even more crucial given that computing the endomorphism ring of a random curve has been shown by Wesolowski [64] to be equivalent to breaking the Supersingular Isogeny Problem.

From here, many questions arise. How hard is it to hide the endomorphism ring of

the starting curve? Is doing so actually more secure, or do we possibly risk creating new vulnerabilities? What other schemes can we create using SIDH variants as primitives while keeping the endomorphism ring unknown? The goal of this thesis is to answer these questions.

This thesis starts with Chapter 2, which consists of an overview of the background knowledge required for the following parts.

In Chapter 3, we show that for some SIDH variants and choices of parameters, the associated problem is randomly self-reducible. This means that, in such cases, we can be assured that using random starting elliptic curves cannot make the protocol less secure. This finding gives us one more reason not to work with fixed curves.

In Chapter 4, we give an overview of the currently known methods to generate supersingular elliptic curves of unknown endomorphism rings. To that, we add a new method to generate curves defined over $\mathbb{F}_p$ using multiparty protocols. Such curves are required for use in schemes like CSIDH [15].

In Chapter 5, we present a number of new zero-knowledge proofs based on SIDH variants masking the torsion mapping or the degree of the secret isogeny. These proofs do not rely on any knowledge of endomorphism rings, and we provide formal proofs of their security.

Chapter 2

Background Knowledge

2.1 Elliptic Curves over Finite Fields

As the name suggests, the fundamental mathematical objects in isogeny-based cryptography are isogenies, which themselves are based on elliptic curves.

The formal definition of elliptic curves is as follows:

Definition 2.1.1. *An elliptic curve is a smooth algebraic curve of genus one with a distinguished rational point.*

Fully introducing the terms required for the formal definition would require an in-depth study of algebraic varieties. As this thesis does not really focus on the algebraic variety structure, we avoid engaging in such a study and refer instead to Silverman's book on the subject [59]. We assume, in this thesis, that the characteristic of any field studied is not 2 or 3. This assumption allows us to represent every elliptic curve (up to isomorphism, which we define later) by an equation in short Weierstrass form, as seen in the following lemma.

Lemma 2.1.2 ([59, p. 42]). *An elliptic curve over a field $\mathbb{F}$ of characteristic not equal to 2 or 3 can be represented using an equation of the form $y^2 = x^3 + ax + b$, where $a, b \in \mathbb{F}$, with the addition of a point at infinity $\mathcal{O}$.*

When describing an elliptic curve, we will either use an equation in short Weierstrass form or give its j -invariant, defined as follows.

Definition 2.1.3. *The j -invariant of an elliptic curve $E : y^2 = x^3 + ax + b$ in short Weierstrass form is*

$$1728 \frac{4a^3}{4a^3 + 27b^2}.$$

The most important thing to know about elliptic curves is that we can define an abelian group using its points.

Theorem 2.1.4 ([59, Prop. III.2.2]). *Let E be an elliptic curve defined over $\mathbb{F}$. The points of E form a commutative group.*

In both this thesis and the study of elliptic curves in general, we often work with the torsion subgroups of elliptic curves.

Definition 2.1.5 (Torsion Subgroup). *Let N be a positive integer and let G be an additive group. The N -torsion of G , noted $G[N]$, is the subgroup of G containing every element P such that $NP = \mathcal{O}$, where $\mathcal{O}$ is the identity.*

SIDH works by transmitting the mapping of a secret isogeny on a chosen torsion subgroup. It turns out that, as long as the value N of the picked torsion is well-chosen, we have a good grasp on the structure of the N -torsion subgroup of an elliptic curve, as seen in the following theorem.

Theorem 2.1.6. [59, p. 86] *Let N be a positive integer and p be a prime number not dividing N . Let E be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. There exists an extension of $\mathbb{F}_{p^2}$ for which the N -torsion of $E[N]$ is isomorphic to $(\mathbb{Z}/N\mathbb{Z})^2$.*

When dealing with the torsion subgroup, we will sometimes want an element of maximal order. We will use the following notation to represent it.

Definition 2.1.7. $E[[N]]$ *is the subset of $E[N]$ consisting of the points of order exactly N .*

2.2 Isogenies

Elliptic curves, as the name suggests, are curves. It therefore makes sense to define rational maps between two such curves. At the same time, as elliptic curves are groups, it also makes sense to define group homomorphisms between them. Isogenies are a combination of the two concepts.

Definition 2.2.1 (Isogeny). *A rational map $\phi : E \rightarrow E'$ between two elliptic curves defined over $\mathbb{F}$ is called a morphism if it is regular at every point. An isogeny is a surjective (when defined over the algebraic closure of $\mathbb{F}$) morphism mapping $\mathcal{O}$ to $\mathcal{O}$.*

Theorem 2.2.2 ([59, Thm. III.4.8]). *Isogenies between elliptic curves are group homomorphisms.*

As we now have the concept of isogenies, we can explain the notion of isomorphism mentioned earlier in this thesis.

Definition 2.2.3 (Isomorphic elliptic curves [59, p. 13]). *Let E_1 and E_2 be elliptic curves. We say that E_1 and E_2 are isomorphic, and write $E_1 \cong E_2$, if there are isogenies $\phi : E_1 \rightarrow E_2$ and $\psi : E_2 \rightarrow E_1$ such that $\psi \circ \phi$ and $\phi \circ \psi$ are the identity maps on E_1 and E_2 , respectively. Such isogenies are called isomorphisms.*

For most isogeny-based cryptography schemes, as well as this thesis, two isomorphic elliptic curves are considered to be the same. The following lemma explains why we like to use j -invariants to describe an elliptic curve.

Lemma 2.2.4 ([59, Prop. III.1.4]). *Two elliptic curves defined over the same field are isomorphic if and only if they have the same j -invariant. The isomorphism might only be defined over a field extension.*

Similarly to elliptic curves, isogenies can also be isomorphic.

Definition 2.2.5 (Isomorphic isogenies). *Let $\phi : E_1 \rightarrow E_2$ and $\psi : F_1 \rightarrow F_2$ be isogenies. We say that ϕ and ψ are isomorphic if there exist elliptic curve isomorphisms $\mu : E_1 \rightarrow F_1$ and $\rho : E_2 \rightarrow F_2$ such that $\rho \circ \phi = \psi \circ \mu$.*

We will mainly discuss separable isogenies in this thesis, as the only non-separable isogeny appearing in this thesis is the Frobenius isogeny.

Definition 2.2.6 (Separable isogenies [59, p. 21]). *Let E_1 and E_2 be elliptic curves defined over the field $\mathbb{F}$ and let $\phi : E_1 \rightarrow E_2$ be an isogeny. Let $\mathbb{F}(E_1)$ and $\mathbb{F}(E_2)$ be the function field of the respective elliptic curve over $\mathbb{F}$. Let $\phi^* : \mathbb{F}(E_2) \rightarrow \mathbb{F}(E_1)$ be the injection of function fields such that $\phi^* f = f \circ \phi$. We say that ϕ is separable if the field extension $\mathbb{F}(E_1)/\phi^* \mathbb{F}(E_2)$ is separable.*

Separable isogenies have many useful properties.

Theorem 2.2.7 ([59, Cor. III.4.9.c]). *The degree of a separable isogeny is equal to the size of its kernel.*

Theorem 2.2.8 ([59, Prop. III.4.12]). *Let E be an elliptic curve and let K be a finite subgroup of E . There is a unique (up to isomorphism) elliptic curve E' and a unique separable isogeny (also up to isomorphism) $\phi : E \rightarrow E'$ such that $\ker(\phi) = K$.*

Because of the above results, we sometimes use the following notation.

Definition 2.2.9. *Let E be an elliptic curve and K be a subgroup of E . We define E/K to be the codomain of the isogeny whose kernel is K , up to isomorphism.*

To compute these isogenies, we use Vélu's formulas.

Theorem 2.2.10 ([62]). *Let $E : y^2 = x^3 + ax + b$ be an elliptic curve defined over a finite field. Let K be a subgroup of E . Let K_2 be the set of points of K of order exactly 2. Let R be a subset of $K \setminus (\{\mathcal{O}\} \cup K_2)$ such that*

$$\begin{cases} K \setminus (\{\mathcal{O}\} \cup K_2) = R \cup (-R), \\ R \cap (-R) = \emptyset. \end{cases}$$

Let $S = K_2 \cup R$. Let f be the morphism such that $f : (x, y) \rightarrow (X, Y)$ with

$$\begin{cases} X = x + \sum_{Q \in S} \left(\frac{t_Q}{x - x_Q} + \frac{4(x_Q^3 + a + b)}{(x - x_Q)^2} \right), \\ Y = y - \sum_{Q \in S} \left(\frac{8y(x_Q^3 + a + b)}{(x - x_Q)^3} + t_Q \frac{y - y_Q}{(x - x_Q)^2} + \frac{2y_Q(3x_Q^2 + a)}{(x - x_Q)^2} \right), \end{cases}$$

where

$$t_Q = \begin{cases} 3x_Q^2 + a & \text{if } Q \in K_2, \\ 6x_Q^2 + 2a & \text{otherwise.} \end{cases}$$

f is an isogeny with domain E and kernel K .

Vélu's formulas give an implicit algorithm to compute an isogeny of a given kernel. We therefore have the following result.

Theorem 2.2.11. *Given a subgroup K of size d of an elliptic curve defined over a finite field, it is possible to compute an isogeny with K as its kernel in time polynomial in terms of d .*

On face value, computing an isogeny of large degree d seems to be inefficient. However, when we can choose the isogeny to compute, we can use the following result by Couveignes.

Theorem 2.2.12 ([20]). *An isogeny defined over a finite field can be written as the composition of isogenies of prime degree, possibly defined over a field extension.*

In practice, we compute isogenies by choosing a B -smooth number as the degree (for some small value B) and writing the large isogeny as the composition of many small ones. For example, an isogeny of degree 2^ℓ can be seen as the composition of ℓ isogenies, each of degree 2, making the computation time polynomial in terms of $\log(d)$.

Definition 2.2.13 (Smooth numbers). *Let B be a positive integer. An integer n is said to be B -smooth if it has no prime factor greater than B .*

It is common to fix a value of B at the start of a paper and then refer to all B -smooth integers as smooth numbers. We do so in this thesis, but the exact value of B is not important for our results, as long as it is small enough for isogenies of degree B to be efficiently computed. In practice, some schemes like SIDH only require two isogeny degrees and can therefore set $B = 3$, while MD-SIDH [33] can have B be as large as 14627 depending on the parameter set. This idea of chaining isogenies is also useful when discussing pushforward isogenies.

Definition 2.2.14 (Pushforward Isogeny [7, 56]). *Let E be a supersingular elliptic curve and let ϕ_1, ϕ_2 be isogenies on E of relatively prime degrees. We define the pushforward isogeny $[\phi_1]_*\phi_2$ to be the isogeny with kernel $\phi_1(\ker(\phi_2))$.*

In this thesis, we rarely use a pushforward isogeny by itself, but instead, compose it with the isogeny that was used to push it. As such, we define the following operation.

Definition 2.2.15. *We define $\phi_2 \boxdot \phi_1 := ([\phi_1]_*\phi_2) \circ \phi_1$.*

This operation, with the help of the following theorem, forms a key building block of most isogeny-based schemes.

Theorem 2.2.16. *Let ϕ_1, ϕ_2 be isogenies on E such that $\ker(\phi_1) \cap \ker(\phi_2) = \{\mathcal{O}\}$. Up to isomorphism, $\phi_2 \boxdot \phi_1$ and $\phi_1 \boxdot \phi_2$ are equal.*

Proof. Since the kernel of both isogenies is $\ker(\phi_1) + \ker(\phi_2)$, Theorem 2.2.8 implies that the isogenies are equal. $\square$

This result will mainly be used in this thesis when dealing with two isogenies of relatively prime degrees. As the concept of endomorphisms is core to this thesis, we define it here.

Definition 2.2.17 (Endomorphism). *An endomorphism is an isogeny whose domain and codomain are the same.*

Of course, we can define the ring of endomorphisms defined on a chosen curve.

Theorem 2.2.18 (Endomorphism Ring). *Let E be an elliptic curve. The set $\text{End}(E)$ of endomorphisms of E (possibly defined over a field extension), with the addition of the morphism 0 that maps every point to $\mathcal{O}$, forms a ring using pointwise addition and functional composition as operations.*

This leads us to the concept of supersingular elliptic curves. As seen in the following theorem, there are multiple ways to define such curves.

Theorem 2.2.19 (Supersingular Elliptic Curve [59, Thm. V.3.1]). *Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$ of characteristic p . The following properties are equivalent:*

1. $\text{End}(E)$ has rank 4 when seen as a $\mathbb{Z}$ -module.
2. $|E(\mathbb{F}_q)| \equiv 1 \pmod{p}$.

If the above conditions are respected, we say that E is supersingular.

Counterintuitively, supersingular elliptic curves are not singular. The name implies that such curves are rarer than non-supersingular curves. Such curves are said to be ordinary and can be defined by the fact that their endomorphism ring has rank 1 or 2.

Another important property of supersingular elliptic curves is the following.

Theorem 2.2.20 ([59, Thm. V.3.1.a.iii]). *Every supersingular elliptic curve has a j -invariant in $\mathbb{F}_{p^2}$ and is therefore isomorphic (with the isomorphism possibly being defined over a field extension) to a curve that can be defined over $\mathbb{F}_{p^2}$.*

Because of this, unless stated otherwise, every elliptic curve in this thesis is defined over $\mathbb{F}_{p^2}$.

2.3 Level Structure

Since SIDH was broken, there have been many attempts to salvage it by masking the mapping. As shown by De Feo, Fouotsa and Panny [26], this technique can be regarded as revealing the mapping of the secret isogeny for a level-structure.

Definition 2.3.1 (Level Structure). *Let E be a supersingular elliptic curve, let N be a positive integer and let H be a subgroup of $GL_2(\mathbb{Z}/N\mathbb{Z})$. For an ordered basis (P, Q) of $E[N]$ and for $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in H$, we define the action of H on the set of ordered bases of $E[N]$ to be*

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} (P, Q) := (aP + bQ, cP + dQ).$$

An H -level structure of E is an orbit of the above action.

We can extend the concept of isogenies directly to level structure.

Definition 2.3.2 (Isogeny with level structure). *Let (E_1, C_1) and (E_2, C_2) be two elliptic curve / H -level structure pairs, and let $\phi: E_1 \rightarrow E_2$ be an isogeny. We say that ϕ extends to an isogeny $\phi: (E_1, C_1) \rightarrow (E_2, C_2)$ between level structures if $\phi(C_1) = C_2$.*

Similarly, we can talk of isomorphic level structures.

Definition 2.3.3 (Isomorphic Level Structures). *Let (E_1, C_1) and (E_2, C_2) be two elliptic curve / H -level structure pairs. We say that these level structures are isomorphic, and write $(E_1, C_1) \cong (E_2, C_2)$, if there is an elliptic curve isomorphism $\phi: E_1 \rightarrow E_2$ such that $\phi(C_1) = C_2$. In the same fashion as without level structure, such isogenies are called isomorphisms of level structures.*

2.4 Supersingular Isogeny Graph

With elliptic curves and isogenies introduced, we can discuss the supersingular isogeny graph.

Definition 2.4.1 (Supersingular isogeny Graph). *For a prime characteristic p and prime degree ℓ , the supersingular ℓ -isogeny graph $\mathcal{G}_\ell$ is the (directed) graph whose vertices are the supersingular elliptic curves over $\mathbb{F}_{p^2}$, up to isomorphism and fixing a representative elliptic curve for each vertex, and whose edges are isogenies (up to isomorphism) of degree ℓ between such curves.*

Remark 2.4.2. *For this thesis, we allow graphs to have multiple edges between vertices, as well as loops. When we mention undirected graphs, we mean a multigraph whose adjacency matrix is symmetric. In other words, undirected multigraphs.*

The size of the supersingular isogeny graph is well understood.

Lemma 2.4.3 ([59, Thm. V.4.1.c]). *The number of vertices of $\mathcal{G}_\ell$ is*

$$|V(\mathcal{G}_\ell)| = \left\lfloor \frac{p}{12} \right\rfloor + \epsilon$$

where

$$\epsilon = \begin{cases} 0 & \text{if } p \equiv 1 \pmod{12}, \\ 1 & \text{if } p \equiv 5 \pmod{12}, \\ 1 & \text{if } p \equiv 7 \pmod{12}, \\ 2 & \text{if } p \equiv 11 \pmod{12}. \end{cases}$$

These graphs were introduced by Pizer [52] and represent one of the first known explicit families of Ramanujan graphs.

Definition 2.4.4 (Ramanujan Graph). *A Ramanujan graph is a k -regular connected graph, for some positive integer k , whose largest eigenvalue is k and whose other eigenvalues have absolute value at most $2\sqrt{k-1}$.*

Theorem 2.4.5 ([52]). *$\mathcal{G}_\ell$ is an $(\ell+1)$ -regular Ramanujan Graph. In addition, if $p \equiv 1 \pmod{12}$, then the adjacency matrix of the graph is symmetrical and thus $\mathcal{G}_\ell$ can be treated as an undirected multigraph.*

We use many tools from algebraic graph theory in this thesis. This is because the upper bound on eigenvalues given by Ramanujan graphs will permit us to use random isogenies of large degree to generate supersingular elliptic curves that are indistinguishable from random, as shown by the following theorem.

Theorem 2.4.6 ([41]). *Let G be a finite k -regular graph for which the non-trivial eigenvalues λ of the adjacency matrix are bounded by $|\lambda| \leq c$, for some $c < k$. Let S be any subset of the vertices of G , and v be any vertex of G . Then a random walk of length at least $\frac{\log(2|G|/\sqrt{|S|})}{\log(k/c)}$ will end in S with probability between $\frac{|S|}{2|G|}$ and $\frac{3|S|}{2|G|}$.*

Supersingular isogeny graphs can be generalized to include level structures intuitively.

Definition 2.4.7 (Supersingular Isogeny Graph with Level Structure). *For a field $\mathbb{F}_{p^k}$, a subgroup H of $GL_2(\mathbb{Z}/N\mathbb{Z})$, and an integer $\ell \nmid N$, the supersingular ℓ -isogeny graph with H -structure $\mathcal{G}_{H,\ell}$ is the graph whose vertices are the pairs (E, C) of supersingular elliptic curves E over $\mathbb{F}_{p^2}$, up to isomorphism, fixing a representative elliptic curve for each vertex, and an H -level structure C of E , and whose edges are the isogenies (up to isomorphism) of degree ℓ between said pairs.*

We need an upper bound on the size of this graph. A simple one to obtain is the following.

Lemma 2.4.8.

$$|\mathcal{E}_H| < \left(\frac{p}{12} + 2\right) \frac{|GL_2(\mathbb{Z}/N\mathbb{Z})|}{|H|}$$

Proof. From Lemma 2.4.3, we have that the number of supersingular curves, up to isomorphism, is less than $\frac{p}{12} + 2$. Since we are looking for an upper bound, we can ignore the possible isomorphism between two level structures of the same curve.

Let E be a supersingular elliptic curve, and let B be the set of ordered bases of $E[N]$. The number of level structures on E is the number of orbits of the group action of H on B . Since $E[N] \cong (\mathbb{Z}/N\mathbb{Z})^2$, $|B| = |GL_2(\mathbb{Z}/N\mathbb{Z})|$.

Since, for any ordered basis, the only matrix stabilizing it is the identity, we have that, by the orbit-stabilizer theorem, each orbit has size $|H|$. We can then combine Burnside's lemma with Lagrange's theorem to get that the number of orbits is $\frac{|B|}{|H|}$, leading to the desired inequality. $\square$

The exact size of $GL_2(\mathbb{Z}/N\mathbb{Z})$ can be computed using the following theorem.

Theorem 2.4.9 ([36]). *Let $N = \prod_{k=1}^t \rho_k^{\alpha_k}$ be the prime factorization of N . Then*

$$|GL_2(\mathbb{Z}/N\mathbb{Z})| = \prod_{k=1}^t \rho_k^{4(\alpha_k-1)} (\rho_k^2 - 1)(\rho_k^2 - \rho_k).$$

As shown in a recent paper by Codogni and Lido, depending on the choice of subgroup for our level structure, we are once again able to obtain Ramanujan graphs.

Theorem 2.4.10 ([18]). *If every invertible multiple of the identity¹, as well as a matrix of every invertible determinant, are in H , then $\mathcal{G}_{H,\ell}$ is an $(\ell+1)$ -Ramanujan graph and its adjacency matrix is diagonalizable.*

¹That is to say, the center of $GL_2(\mathbb{Z}/N\mathbb{Z})$.

The conditions for Codogni and Lido's theorem appear multiple times in this thesis. As such, we use the following notation.

Definition 2.4.11 (CL-admissible). *Let H be a subgroup of $GL_2(\mathbb{Z}/N\mathbb{Z})$. We say that H is CL-admissible if the following properties hold:*

- H contains every multiple of the identity matrix,
- $\forall k \in (\mathbb{Z}/N\mathbb{Z})^\times$, H contains a matrix whose determinant is equal to k .

Remark 2.4.12. *If $p \equiv 1 \pmod{12}$, and $\begin{pmatrix} \ell & 0 \\ 0 & \ell \end{pmatrix}$ belongs to H , the graph $\mathcal{G}_{H,\ell}$ is always an undirected multigraph [18, Prop. 2.2.2]. In other cases, the graph may still be undirected, depending on the values of p, N, ℓ , and H .*

For example, when N is prime, the curve with $j = 1728$ is supersingular if and only if $p \equiv 3 \pmod{4}$, in which case [3, Prop. 3.6] states that complex multiplication by i has no fixed points in the set of order N subgroups precisely when $N \equiv 3 \pmod{4}$. In such cases, all vertices of the isogeny graph with level structure have the same number of automorphisms.

Similarly, the curve $j = 0$ is supersingular if and only if $p \equiv 2 \pmod{3}$, and complex multiplication by ζ_3 has no fixed points among the set of order N subgroups precisely when $N \equiv 2 \pmod{3}$.

We emphasize that uniformity of the number of automorphisms, by itself, is not sufficient to conclude that the adjacency matrix of the isogeny graph with level structure is symmetric. As shown in [18, Prop. 2.2.2], we also need $\begin{pmatrix} \ell & 0 \\ 0 & \ell \end{pmatrix}$ to belong to H in order to ensure symmetry.

For a fixed value of p and H , the number of supersingular elliptic curves with H -level structure up to isomorphism is also fixed. As such, the dimensions of the adjacency matrix of $\mathcal{G}_\ell$ are the same for every value of ℓ . In fact, we also have that these matrices commute.

Lemma 2.4.13. *Let ℓ_1 and ℓ_2 be distinct primes and let A_1 and A_2 be the adjacency matrices of $\mathcal{G}_{H,\ell_1}$ and $\mathcal{G}_{H,\ell_2}$ respectively. We have that $A_1 A_2 = A_2 A_1$.*

Proof. For two elliptic curves E and F , $A_1(E, F)$ is the number of ℓ_1 isogenies from E to F . Similarly, $A_2 A_1(E, F)$ is the number of isogenies from E to F that can be constructed as $\phi_2 \square \phi_1$ where ϕ_2 is an isogeny of degree ℓ_2 and ϕ_1 is an isogeny of degree ℓ_1 . By Theorem 2.2.16, each such isogeny can be written as $\phi_1 \square \phi_2$. Therefore, $A_1 A_2(E, F) \geq A_2 A_1(E, F)$. Since this argument is symmetric, we can conclude that $A_1 A_2(E, F) = A_2 A_1(E, F)$. As this holds for any pair of elliptic curves, we have that $A_1 A_2 = A_2 A_1$. $\square$

An important question in Chapter 3 is whether there exists an isogeny of fixed degree between two supersingular curves with level structure. As such, we use the following notation.

Definition 2.4.14. *Let $(E, C) \in \mathcal{E}_H$. We define $\Xi_{E,H,C,d}$ as*

$$\Xi_{E,H,C,d} = \{(F, D) \in \mathcal{E}_H \mid \exists \phi: E \rightarrow F, \phi(C) = D \text{ and } \deg(\phi) = d\}.$$

2.5 Isogeny Problems

Every isogeny-based scheme discussed in this thesis is defined over $\mathbb{F}_p$ or $\mathbb{F}_{p^2}$. As such, the key sizes are polynomial in terms of $\log(p)$. Because of this, we study the complexity of the problems in this thesis in terms of $\log(p)$.

Definition 2.5.1. *A function in terms of $\log(p)$ is said to be non-negligible if it can be lower bounded in absolute value by a non-zero rational function in terms of $\log(p)$. It is said to be negligible otherwise.*

Definition 2.5.2. *An algorithm is said to be efficient if the number of basic operations it requires to complete can be upper bounded by a polynomial in terms of $\log(p)$.*

Definition 2.5.3. *A problem is said to be hard if there is an algorithm solving it that is both efficient and has a non-negligible probability of success.*

We now have enough mathematical background to start discussing isogeny-based cryptography. The fundamental problem in this branch of post-quantum cryptography is the Supersingular Isogeny Problem.

Definition 2.5.4 (**Supersingular Isogeny Problem (SIP)**). *Given two supersingular elliptic curves E and F over $\mathbb{F}_{p^2}$, compute an isogeny $\phi: E \rightarrow F$.*

As a post-quantum problem, there is no known efficient way to solve SIP, even when using a quantum computer. The fastest known quantum attack uses Tani's algorithm [60], which has a time and space cost of $O(\sqrt[6]{p})$.

Another hard problem in the domain is the Endomorphism Ring Computation Problem.

Definition 2.5.5 (**Endomorphism Ring Computation Problem**). *Given a random supersingular elliptic curve E , compute $\text{End}(E)$.*

These two problems have been proven to be equivalent by Wesolowski [63].

Theorem 2.5.6 ([63, §7], [64, Cor. 5]). *The Supersingular Isogeny Problem and the Endomorphism Ring Computation Problem can be polynomially reduced to each other.*

The main issue with SIP is that there is no known way to generate a public key encryption scheme based solely on it. As such, we need to transmit some extra information. For a long time, the main candidate for such a scheme was Supersingular Isogeny Diffie–Hellman (SIDH). The idea of this protocol is to transmit the mapping of the secret isogeny on a large torsion subgroup of the domain curve in order to allow for a Diffie–Hellman-like secret sharing scheme.

Definition 2.5.7 (SIDH Problem). *Let d and N be relatively prime positive integers. Let E and F be supersingular elliptic curves over $\mathbb{F}_{p^2}$ and let $\{P, Q\}$ be a basis of $E[N]$. Given a guarantee that there is a d -isogeny $\psi: E \rightarrow F$ such that $\psi(P) = R$ and $\psi(Q) = S$, compute an isogeny $\phi: E \rightarrow F$ such that $(\phi(P), \phi(Q)) = (R, S)$.*

However, this scheme was proven insecure by [16] when the endomorphism ring of the starting curve is known, with [42] and [55] later removing the conditions. This means that transmitting the entire torsion mapping is too much, but transmitting nothing is hard to work with. As such, there have been many attempts since the attacks to develop a new scheme that transmits just enough information to be usable while still being secure. For example, M-SIDH [33] masks the torsion-point mapping with a single random scalar, while BinSIDH [8] and FESTA [9] mask each of the two points with its own secret random scalar. Recently, a paper by De Feo, Fouotsa and Panny [26] generalized the key recovery problem against most of these schemes into a single problem named Γ -SIDH.

Definition 2.5.8 (Known Level Structure Supersingular Isogeny Problem (Γ -SIDH) [26]). *Let d and N be relatively prime positive integers, and let $H < GL_2(\mathbb{Z}/N\mathbb{Z})$. Let (E, C) and (F, D) be supersingular elliptic curves- H -level structure pairs over $\mathbb{F}_{p^2}$. Given a guarantee that there is a d -isogeny $\psi: (E, C) \rightarrow (F, D)$, compute an isogeny $\phi: E \rightarrow F$ such that $\phi(C) = D$.*

We can describe most SIDH variants using Γ -SIDH, as well as some zero-knowledge proofs and CSIDH, which will be discussed later.

H	Level Structure	Application
$GL_2(\mathbb{Z}/N\mathbb{Z})$	Trivial	Generic
$\left\{ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right\}$	Full	SIDH
$\left\{ \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \right\}$		M-SIDH
$\left\{ \begin{pmatrix} * & * \\ 0 & * \end{pmatrix} \right\}$	Borel	ZKPs
$\left\{ \begin{pmatrix} * & 0 \\ 0 & * \end{pmatrix} \right\}$	Cartan	FESTA, CSIDH*

A condition that remains for the SIDH attacks to work is to know the degree of the secret isogeny. As such, there have been schemes proposed that mask such a degree. For example, MD-SIDH [33] is a variant of M-SIDH where a large multiple of the degree is transmitted instead of the degree. TerSIDH [8] works in a similar fashion, using BinSIDH as a basis. We can once again combine all these primitives in a single problem.

Definition 2.5.9 (Masked Degree Γ -SIDH). *Let d' and N be relatively prime positive integers, and let $H < GL_2(\mathbb{Z}/N\mathbb{Z})$. Let (E, C) and (F, D) be supersingular elliptic curves- H -level structure pairs over $\mathbb{F}_{p^2}$. Let d be a secretly sampled uniformly random divisor of d' . Given a guarantee that there is a d -isogeny $\psi: (E, C) \rightarrow (F, D)$, compute an isogeny $\phi: E \rightarrow F$ such that $\phi(C) = D$.*

While Γ -SIDH is a computational problem, it is also possible to study its decisional variant. When the degree of the secret isogeny is smooth, it is easy to show that the decisional problem is equivalent to the computational one. If we had an algorithm solving the decisional variant, then, because the degree of the secret isogeny is a smooth number, we can guess the first small isogeny in the chain and use the decisional oracle to check if the guess was correct. It is then possible to repeat the process until the entire chain is recovered. Castryck and Decru's attack on SIDH [14] first breaks the decisional variant and recovers the key by doing exactly that.

An example of a decisional Γ -SIDH problem is the Decisional SuperSingular Product problem (DSSP).

Definition 2.5.10 (DSSP). *Let A and B be two large, smooth and relatively prime integers. Given a cyclic isogeny $\phi: E_0 \rightarrow E_1$ of degree A , the decisional supersingular product problem is to distinguish between the following two distributions:*

1. $\mathcal{D}_0 = \{(E_2, E_3, \phi')\}$ such that there exists a cyclic subgroup $G \subseteq E_0[B]$ of order B and $E_2 \cong E_0/G$ and $E_3 \cong E_1/\phi(G)$, and $\phi' : E_2 \rightarrow E_3$ is a degree A cyclic isogeny.
2. $\mathcal{D}_1 = \{(E_2, E_3, \phi')\}$ such that E_2 is a random supersingular elliptic curve with the same cardinality as E_0 and E_3 is the codomain of a random cyclic isogeny $\phi' : E_2 \rightarrow E_3$ of degree A .

In both cases, ϕ and ϕ' are presented as a list of prime degree isogenies, for the sake of a compact representation.

This problem is equivalent to deciding if there exists a cyclic isogeny of degree B mapping $\ker(\phi)$ to $\ker(\phi')$. Since both published isogenies are cyclic, computing such an isogeny is equivalent to solving Γ -SIDH with Borel level structure. DSSP can therefore be interpreted as its decisional variant. The DSSP assumption states that the titular problem is hard.

Assumption 2.5.11 (DSSP assumption). *The DSSP problem is hard, as defined in Definition 2.5.3.*

2.6 Small Functions for Protocols

The protocols presented in this thesis use multiple functions, many of these being used for more than one protocol. In order to avoid repeating these definitions every time, we present them once in this section.

Definition 2.6.1. *Given an elliptic curve E , $\text{JINV}(E)$ returns the j -invariant of E .*

Definition 2.6.2. *Given a cyclic isogeny $\phi : E \rightarrow F$, $\text{GENERATINGPOINT}(\phi)$ returns a point K generating the kernel of ϕ . Given an elliptic curve point $K \in E$, $\text{ISOGENYFROMKERNEL}(K)$ returns an isogeny whose kernel is generated by K .*

Definition 2.6.3. *Given a supersingular elliptic curve E and a positive integer n , $\text{CYCLICISOGENY}(E, n)$ returns a random cyclic isogeny whose domain is E and degree is n .*

Definition 2.6.4. *Given two isogenies $\phi : E \rightarrow E_1$ and $\psi : E \rightarrow E_2$ of relatively prime degrees, $\text{PARALLELISOGENY}(\phi, \psi)$ returns $[\psi]_*\phi$. This algorithm can and must be implemented in a way that guarantees that $\text{PARALLELISOGENY}(\phi, \psi) \circ \psi = \text{PARALLELISOGENY}(\psi, \phi) \circ \phi$.*

Definition 2.6.5. Given an elliptic curve E and a positive integer n , $\text{RANDOMBASIS}(E, n)$ returns a uniformly random basis (P, Q) of $E[n]$.

Definition 2.6.6. Given three elliptic curve points P, Q, R , $\text{DDLOG}(P, Q, R)$ returns a pair of integers (e, f) such that $[e]P + [f]Q = R$.

Remark 2.6.7. We only use DDLOG in groups whose order is a smooth number, making the function efficient.

Definition 2.6.8. Given an isogeny $\phi : E \rightarrow F$, $\text{CODOMAIN}(\phi)$ returns F .

Definition 2.6.9. Given a positive integer A and two integers (a, b) , $\text{INVERPAIR}(a, b, A)$ returns a pair of integers (a', b') such that $a'b - b'a$ is invertible modulo A . Note that we only use this function in cases where a valid solution exists.

Definition 2.6.10. Given a positive integer A , $\text{FACSET}(A)$ is the set of positive factors of A . Given two positive integers A and B , $\text{FACSETTWO}(A, B)$ is the set of integers d such that $A \mid d \mid B$.

Definition 2.6.11. Given a possibly non-cyclic isogeny ϕ , $\text{CYCLIPHY}(\phi)$ returns a cyclic isogeny with the same domain and codomain. This can be easily obtained by seeing ϕ as a walk on the isogeny graph and removing every pair of steps using the same edge in opposite directions.

Definition 2.6.12. The function $\text{ISOVALID}(E, F, \phi)$ returns **true** if ϕ is a well-defined isogeny from E to F and **false** otherwise.

Definition 2.6.13. During the Verification step of the zero-knowledge proofs in this thesis, we use **accept** to note that the Verifier accepts and $\perp$ to note refusal.

2.7 SIDH

While this thesis mostly deals with the primitives for key recovery attacks of SIDH-like schemes rather than the full implementations, knowing how the entire scheme works can give intuition on why some parameter sets are chosen. Hence, here is a simplified version of SIDH.

Definition 2.7.1 ([Supersingular Isogeny Diffie–Hellman](#)). Let Alice and Bob be two parties wanting to generate a shared secret. Let A, B be two large, smooth, relatively prime positive integers, and let $p = AB - 1$ be a prime number. Let E_0 be a supersingular elliptic curve over $\mathbb{F}_{p^2}$ and let $P, Q \in E[AB]$ be a basis of $E[AB]$. Supersingular Isogeny Diffie–Hellman consists of the following steps.

1. *Key Generation:* Alice and Bob each generate their public and secret keypairs using **KEYGEN**. Alice inputs (A, B) as her values for (d, N) in order to obtain $(s_A, (E_A, P_A, Q_A))$ and Bob inputs (B, A) in order to obtain $(s_B, (E_B, P_B, Q_B))$. The public keys are then published.
2. *Secret Sharing:* Alice can generate a shared secret by inputting $(s_A, (P_B, Q_B))$ into **SECRETSHARE**. Bob can obtain the same curve (up to isomorphism) by inputting $(s_B, (P_A, Q_A))$ into the same algorithm.

KEYGEN (d, N)	SECRETSHARE $(s, (P_2, Q_2))$
$s \leftarrow \mathbb{F}_{p^2}$	$K' \leftarrow P_2 + sQ_2$
$K \leftarrow NP + sNQ$	$\phi' \leftarrow \text{ISOGENYFROMKERNEL}(K')$
$\phi \leftarrow \text{ISOGENYFROMKERNEL}(K)$	$E'_1 \leftarrow \text{CODOMAIN}(\phi')$
$E_1 \leftarrow \text{CODOMAIN}(\phi)$	return $\text{JINV}(E'_1)$
return $(sk := \phi, pk := (E_1, \phi(dP), \phi(dQ)))$	

Because of Theorem 2.2.16, Alice and Bob will arrive at the same curve, up to isomorphism. The j-invariant of the resulting curve is used as the shared secret to guarantee that isomorphic curves lead to the same shared secret.

While SIDH is no longer secure, its variants keep similar ideas of using the commutative property of isogenies of relatively prime degrees. M-SIDH in particular changes almost nothing other than masking the mappings of the secret isogeny with a random scalar.

Even before the discovery of the polynomial attacks against it, SIDH was known to be vulnerable to adaptive attacks [35] if no countermeasure was added. We discuss the use of zero-knowledge proofs to create these countermeasures in Chapter 5. However, the more common modification on SIDH and its variants is to use the Hofheinz–Hövelmanns–Kiltz variant [37] of the Fujisaki–Okamoto transform.

Definition 2.7.2 (Fujisaki–Okamoto transform [34]). *Let $(\text{KEYGEN}^{pub}, E^{pub}, D^{pub})$ be the key generation, encryption, and decryption protocols of a public key encryption scheme. (E^{sec}, D^{sec}) be the encryption and decryption protocols of a secret key encryption scheme. Let K be the key space of the secret key encryption scheme. Let the message space $M := M_1 \| M_2$ of the public key scheme be written as the concatenation of two sets. Let $H : \{0, 1\}^* \rightarrow M_2$ and $G : \{0, 1\}^* \rightarrow K$ be random oracle functions. The Fujisaki–Okamoto transform is a hybrid scheme consisting of the following protocols.*

KEYGEN(1^λ)	ENCRYPT(pk, m)	DECRYPT(sk, c_1, c_2)
$(sk, pk) \leftarrow \text{KEYGEN}^{pub}(1^\lambda)$	$\sigma \leftarrow \$ M_1$	$\hat{\sigma} \ r'_1 \leftarrow D^{pub}(sk, c_1)$
return (sk, pk)	$r_1 \leftarrow H(\sigma, m)$	$\hat{r}_2 \leftarrow G(\hat{\sigma})$
	$r_2 \leftarrow G(\sigma)$	$\hat{m} \leftarrow D^{sec}(\hat{r}_2, c_2)$
	$c_1 \leftarrow E^{pub}(pk, \sigma \ r_1)$	$\hat{r}_1 \leftarrow H(\hat{\sigma}; \hat{m})$
	$c_2 \leftarrow E^{sec}(r_2, m)$	if $c_1 = E^{pub}(pk, \hat{\sigma} \ \hat{r}_1)$:
	return (c_1, c_2)	return $\hat{m}$
		else :
		return $\perp$

The core idea of the Fujisaki–Okamoto transform is that, because of the random oracles, one cannot feasibly dishonestly generate ciphertexts. It was proven that this transform can convert a public scheme secure against chosen plaintext attacks into a scheme secure against chosen ciphertext attacks. The Hofheinz–Hövelmanns–Kiltz variant adds some checks required for the security proof to hold in the quantum random oracle model. This variant was used on SIDH to create Supersingular Isogeny Key Encapsulation (SIKE) [39], which resists adaptive attacks. The same transform is also used for most SIDH variants, as they are vulnerable to adaptation attacks without it. However, this transform comes at the cost of needing to use random oracles and ephemeral keys (σ can be considered as one such key), which limits some possible applications.

2.8 CSIDH

Commutative Supersingular Isogeny Diffie–Hellman (CSIDH) is another secret sharing scheme based on isogenies. It was developed by Castryck, Lange, Martindale, Panny and Renes in [15]. While the protocol is vulnerable to subexponential time quantum attacks [17, 49], it has received increased interest as it is not broken by the polynomial time SIDH attacks.

CSIDH works on supersingular elliptic curves defined over $\mathbb{F}_p$ as it makes it easier to use the ideal class group.

Definition 2.8.1 (Ideal Class Group). *Let $\mathcal{O}$ be a Dedekind domain.*

A fractional ideal of $\mathcal{O}$ is in the form $\alpha \mathfrak{a}$ where $\alpha \in \mathcal{O} \setminus \{0\}$ and $\mathfrak{a}$ is an $\mathcal{O}$ -ideal. A fractional ideal A is invertible if there is a fractional ideal B such that $AB = \mathcal{O}$. Let the set of invertible fractional ideals be $I(\mathcal{O})$. $I(\mathcal{O})$ forms a group.

A principal fractional ideal is a fractional ideal generated by a single element and can therefore be written as $\alpha\mathcal{O}$ (every fractional ideal is invertible). Let the set of principal fractional ideals be $P(\mathcal{O})$. $P(\mathcal{O})$ forms a subgroup of $I(\mathcal{O})$.

The ideal class group $\mathcal{C}$ is defined as $I(\mathcal{O})/P(\mathcal{O})$.

When we refer to an ideal class group in this thesis, it will always be one associated with the endomorphism ring of an elliptic curve defined over $\mathbb{F}_p$. This ideal class group can be used to define a group action.

Theorem 2.8.2 (Ideal Class Group Action [15]). *Let $p \equiv 1 \pmod{4}$ and let E be a supersingular elliptic curve defined over $\mathbb{F}_p$. Let $\mathfrak{a}$ be ideal of $\text{End}(E)$. We define the isogeny $\phi_{\mathfrak{a}} : E \rightarrow E_{\mathfrak{a}}$ to be the separable isogeny whose kernel is $\cap_{a \in \mathfrak{a}} \ker([a])$. We defined the action of $\mathcal{C}$ on the set of supersingular elliptic curves over $\mathbb{F}_p$ to be $\mathfrak{a} \star E \mapsto E_{\mathfrak{a}}$.*

This action is well defined, commutative and transitive.

This group action can then be used to create a Diffie–Hellman-like scheme whose key recovery problem is equivalent to the following.

Definition 2.8.3 (CSIDH Problem With Known Ideal Class Group). *Let E be a supersingular elliptic curve defined over $\mathbb{F}_p$ and let $\mathcal{C}$ be its ideal class group. Let $\mathfrak{a}$ be uniformly randomly sampled from $\mathcal{C}$. Given $\mathfrak{a} \star E$, compute an element of $\mathfrak{a}$.*

CSIDH is therefore able to share secrets without having to transmit the torsion mapping of its secret isogeny. It does come with its own limitations. First, we are restricted to curves defined over $\mathbb{F}_p$, leading to subexponential time quantum attacks [17]. Also, the way the isogeny is crafted means it always commutes with the Frobenius endomorphism. The information this reveals can be converted to Γ -SIDH with Cartan level structure [26]. Also, if the degree of the secret isogeny were revealed, it would leak the norm of the chosen ideal and allow an attack. This means that the degree must be kept secret and this limits some possible applications, as we will discuss in Chapter 4.

Finally, a core issue of CSIDH is that, for a classical computer, computing the ideal class group is a hard problem, as defined in Definition 2.5.3. This will not be an issue once someone has a quantum computer, but it makes it hard to generate parameters right now. One way to do it is to try computing the ideal class group for a specific large prime, even if it takes a very long time. This was done for some CSIDH-based schemes like CSI-FiSh [11]. Another way to solve this issue is to realize that, while computing the entire group is hard, it is feasible to find a set of ideals with small norm conjecturally generating the group, without computing the whole group. This technique is used for most implementations of CSIDH and leads to the following definitions.

Definition 2.8.4. Let B be a positive integer and let $\mathcal{S}$ be a set of t generators of $\mathcal{C}$. Define $\text{CSIDHSAMPLE}(\mathcal{S}, B)$ samples a random element in $[-B, B]^t$.

Remark 2.8.5. Because $\mathcal{S} := (\mathfrak{l}_1, \dots, \mathfrak{l}_t)$ is part of the public parameters, the output $(e_1, \dots, e_t)$ implicitly represent the ideal class generated by $\mathfrak{a} := \prod_{i=1}^t \mathfrak{l}_i^{e_i}$. For the sake of simpler notation, we write that CSIDHSAMPLE outputs an ideal class. However, in practice, CSIDH works directly with the exponent vector and computes the action $\mathfrak{a} \star E$ by applying the action of each $\mathfrak{l}_i$ one by one.

Definition 2.8.6 (CSIDH with Unknown Ideal Class Group). Let E be a supersingular elliptic curve defined over $\mathbb{F}_p$ and let $\mathcal{C}$ be the ideal class group of $\text{End}(E)$. Let $\mathfrak{a}$ be randomly sampled using $\text{CSIDHSAMPLE}(\mathcal{C}, B)$. Given $\mathfrak{a} \star E$, compute $\mathfrak{a}$.

2.9 Other Cryptography Concepts

We use this section to introduce the required cryptographic concepts for this thesis. We avoid introducing encryption schemes as we mostly work on the primitives associated with them rather than any specific implementation. On the other hand, zero-knowledge proofs are the main focus in Chapter 5 and are instrumental in Chapter 4. We therefore introduce them here.

Definition 2.9.1 (Zero-Knowledge Proof). Let f be a mapping. A zero-knowledge proof is an interactive protocol between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$, where $\mathcal{P}$ wants to prove to $\mathcal{V}$ that, for a public output y they know an input x such that $f(x) = y$. A zero-knowledge proof is said to be secure if the following properties hold:

1. **Correctness:** If both parties act honestly, the protocol should always succeed.
2. **Soundness:** It should be hard for $\mathcal{P}$ to convince $\mathcal{V}$ without actually knowing a valid x .
3. **Zero-Knowledge:** The protocol should give no information x .

By far the most common form of zero-knowledge proofs is the sigma protocol, first introduced by Cramer [21].

Definition 2.9.2 (Sigma Protocol). A sigma protocol is a zero-knowledge proof with the following steps:

1. **Commitment:** $\mathcal{P}$ generates some randomness and uses it to compute and publish a commitment c .
2. **Challenge:** After seeing the commitment, $\mathcal{V}$ publishes a random challenge t .
3. **Answer:** With the use of x , $\mathcal{P}$ generates an answer z depending on both c and t .
4. $\mathcal{V}$ checks that (y, c, t, z) respect some desired relation.

Sigma protocols are called so because the back-and-forth between the parties makes a diagram that looks like the Greek letter Σ . The main advantage of these protocols is that it is easy to formally define their security properties. Correctness is simple enough that we do not need to discuss further.

Zero-knowledge is formally proven by showing that anybody can simulate, with non-negligible success probability, transcripts (y, c, t, z) that are indistinguishable from those of two honest parties doing the protocol. Sometimes it is possible to prove that a proof is Honest Verifier Zero-Knowledge (HVZK), which means that the protocol leaks no information on the assumption that the verifier acts honestly. Such schemes have more limited applications compared to full secure zero-knowledge proofs, but they pose no issue when used for signature using a Fiat–Shamir transform. When the protocol is zero-knowledge, it can also be sound since, contrary to a simulator, $\mathcal{P}$ needs to do the steps in order and cannot control the output of $\mathcal{V}$.

Regarding soundness, a common way to define it is using Special Soundness [22], which consists in proving that if one was given two distinct valid answers z_1 and z_2 to two distinct challenges t_1 and t_2 of the same commitment c , then an attacker can recover a valid value x such that $f(x) = y$. We use a generalization of this property in this thesis.

Definition 2.9.3 (k -Soundness). *A sigma protocol is said to be k -sound if, for any valid fixed commitment, as well as k distinct pairs $(t_1, z_1), \dots, (t_k, z_k)$ such that (y, c, t_i, z_i) is valid transcript, there exists a polynomial algorithm outputting a value x such that $f(x) = y$.*

In practice, if a zero-knowledge proof is k -sound, it means that there is a probability of $\frac{1}{k}$ of a dishonest prover being caught. A scheme using such a proof will need to repeat it until $\left(\frac{k-1}{k}\right)^n$ is negligible, where n is the number of repetitions.

Zero-knowledge proofs are often used as parts of a signature scheme by using the Fiat–Shamir transform. In order to do so, we require a random oracle.

Assumption 2.9.4 (Random Oracle Assumption). *We assume that W is a random oracle function. Let $\mathcal{W}$ be the codomain of W . We also assume that $\mathcal{W}$ is large enough for collisions to be infeasible to find and that the mapping of any input by W can be efficiently computed.*

To be more precise, in this thesis, we require that, for any function f with domain X , the problem of distinguishing between $(f(x), W(x))$ for a random $x \in X$ and $(f(x), r)$, where r is a uniformly random element of $\mathcal{H}$ is equivalent to the problem of computing $x \in X$ when given $f(x)$.

In cases where we use W on arbitrary data, we implicitly assume that this data is encoded in the form of a binary string using a suitable encoding scheme.

Using this assumption, we can apply the Fiat–Shamir transform to convert an HVZK Sigma protocol into a signature scheme. As the name implies, this method was first applied to RSA by Fiat and Shamir [32].

Definition 2.9.5 (Fiat–Shamir Signature). *Let Π be a secure honest verifier zero-knowledge Sigma protocol. Let C be the set of possible commitments and T be the set of possible challenges. Let s be a positive integer and let $W : C^s \rightarrow T^s$ be a random oracle.*

The Fiat–Shamir signature based on Π consists of the prover first sampling s random commitments $c_1, \dots, c_s$ and inputting them into W to obtain s challenges $t_1, \dots, t_s$. The prover can then compute a valid answer z_i for each pair (c_i, t_i) . $\mathcal{P}$ can then publish $(c_1, t_1, z_1), \dots, (c_s, t_s, z_s)$.

The verifier can then check that each (c_i, t_i, z_i) triple is valid and that $W((c_1, \dots, c_s)) = (t_1, \dots, t_s)$.

The above definition of Fiat–Shamir is more generic than the original one since it can be used for any Sigma protocol and it allows for multiple copies of said protocol to be used. This is vital if we want to use the transform of k -sound zero-knowledge proofs, since a single copy can be brute forced in a small number of attempts.

Another concept important for this thesis is random-self reducibility. We give the definition here and will discuss the concept further in Chapter 3, where random self-reducibility is the main focus.

Definition 2.9.6 (Random Self-Reducibility). *We say that a problem is randomly self-reducible if every algorithm $\mathcal{A}$ solving it for a non-negligible fraction of cases implies the existence of an algorithm $\mathcal{B}$ solving every case, whose time complexity is at most a polynomial factor larger than $\mathcal{A}$'s in terms of the input size.*

2.10 Other Mathematical Concepts

In this final section of this chapter, we insert some concepts that are needed for some of our results, but do not fit in the previous sections.

One such concept is Chebyshev polynomials of the second kind.

Definition 2.10.1 (Chebyshev polynomials of the second kind). *The family $\{U_n(x)\}_{n \in \mathbb{N}}$ of Chebyshev polynomials of the second kind is defined by the following recurrence relation:*

$$\begin{cases} U_0(x) = 1, \\ U_1(x) = 2x, \\ U_{n+1}(x) = 2xU_n(x) - U_{n-1}(x), \text{ for } n \geq 1. \end{cases}$$

These polynomials can be used as building blocks to count the number of non-backtracking walks between two vertices in a graph. This fact was previously used both by Alon, Benjamini, Lubetzky and Sodinand [2], as well as Sardari [57]. This property will be vital in our proof of the random self-reducibility of SIDH variants.

An important property of Chebyshev polynomials of the second kind is the following bound.

Theorem 2.10.2 ([1, Prop. 22.14.6]). *For $-1 \leq x \leq 1$, we have that $-(n+1) \leq U_n(x) \leq (n+1)$.*

We also require the following results from linear algebra.

Theorem 2.10.3 ([38, Thm. 1.3.21]). *If a family of diagonalizable matrices commutes, then there is an eigenvector basis that diagonalizes all of them at the same time (with possibly different eigenvalues).*

Theorem 2.10.4 ([50, Thm. 4.3.4]). *If A is a real symmetric matrix, then it is diagonalizable and its eigenvectors are orthonormal.*

In addition, let $\lambda_1, \dots, \lambda_n$ be the eigenvalues of A with associated orthonormal eigenvectors $\phi_1, \dots, \phi_n$. We have that

$$A = \sum_{i=1}^n \lambda_i \phi_i^t \phi_i.$$

Theorem 2.10.5. *If A is diagonalizable with eigenvalues $\lambda_1, \dots, \lambda_n$ with associated orthonormal eigenvectors $\phi_1, \dots, \phi_n$ and let p be a polynomial, then*

$$p(A) = \sum_{i=1}^n p(\lambda_i) \phi_i^t \phi_i.$$

Proof. Let $p(x) = \sum_{j=0}^k c_j x^j$. We have that

$$\begin{aligned} p(A) &= \sum_{j=0}^k c_j A^j \\ &= \sum_{j=0}^k c_j \left(\sum_{i=1}^n \lambda_i \phi_i^t \phi_i \right)^j \end{aligned}$$

Because the eigenvectors are orthonormal, the product of two distinct eigenvectors is equal to zero. Hence, any term of $(\sum_{i=1}^n \lambda_i \phi_i^t \phi_i)^j$ containing more than a single eigenvector is equal to zero. Therefore

$$\begin{aligned} p(A) &= \sum_{j=0}^k c_j \sum_{i=1}^n (\lambda_i \phi_i^t \phi_i)^j \\ &= \sum_{j=0}^k c_j \sum_{i=1}^n \lambda_i^j (\phi_i^t \phi_i)^j \\ &= \sum_{j=0}^k c_j \sum_{i=1}^n \lambda_i^j \phi_i^t \phi_i \\ &= \sum_{i=1}^n \sum_{j=0}^k c_j \lambda_i^j \phi_i^t \phi_i \\ &= \sum_{i=1}^n p(\lambda_i) \phi_i^t \phi_i. \end{aligned}$$

□

Chapter 3

Random Self-Reducibility

The results presented in this chapter mostly come from [44], the latest of the three papers that fuel this thesis. The main reason that we start with this paper, instead of going in chronological order, is that the results in the chapter give motivation for those in the following chapters. Indeed, the other chapters are about zero-knowledge proofs using elliptic curves of unknown endomorphism rings and the generation of such curves. In practice, in order to have a supersingular elliptic curve of unknown endomorphism ring, it must be random.

The principal reason we are interested in curves of unknown endomorphism rings is security. In the history of isogeny-based cryptographic protocols, some of the earliest ideas for attacks against a scheme utilize endomorphism rings. If we take [SIDH](#) as an example, before it was broken, there was a known polynomial attack against it when the starting endomorphism ring was known and the degree of the isogeny was much smaller than the order of the revealed torsion [51]. At the time, the chosen countermeasure was to simply have both parties use isogenies of similarly sized degrees in order to never have unbalanced parameters. However, some applications, such as some Key-Updatable Public-Key Encryption schemes based on SIDH [30], could not do so and therefore needed the endomorphism ring to be secret. When SIDH was broken, the first attack against it [14] required knowledge of the starting endomorphism ring. Later attacks [55] that avoided it required working in higher dimensions and were therefore less efficient. Even for new SIDH variants, such as M-SIDH, knowledge of the endomorphism ring, among other things, can lead to attacks [16].

Because of this, there is an interest in working with random starting elliptic curves. However, can we be sure that doing so does not somehow create new vulnerabilities? As

an extreme example, it is commonly known that the discrete logarithm problem can be easier to solve if we let any random integer be the order of the group. If the selected order is one that can be factorized easily, the problem can be reduced to solving DLOG on a group whose order is equal to the largest prime dividing the original order [53]. In order to prove that working with random supersingular elliptic curves is secure, it is sufficient to show that the worst case of an isogeny problem can be reduced to its average case, and therefore is randomly self-reducible.

There are other reasons why we care that a problem is randomly self-reducible. For example, Tompa and Woll [61] show that a randomly self-reducible problem can be used to create zero-knowledge proofs.

For an example of a non-randomly self-reducible problem. Consider the Decisional Diffie–Hellman problem on $(\mathbb{Z}/p\mathbb{Z})^\times$ for some prime p : Let g be a generator of $(\mathbb{Z}/p\mathbb{Z})^\times$, let a, b be secret integers and let g^a, g^b be publicly known. A challenger randomly samples a coin flip, then publishes g^{ab} if the coin fell on heads and instead publishes a random $g^r \leftarrow \mathbb{Z}/p\mathbb{Z}^\times$ if the coin fell on tails. The adversary must then guess the result of the coin flip. If we let r be sampled uniformly, then the adversary can use the property of Legendre symbols to gain a non-negligible advantage. By checking the values $\left(\frac{g^a}{p}\right)$ and $\left(\frac{g^b}{p}\right)$, the adversary can obtain the parity of both a and b . If r is sampled naively, there is a 50% probability that it does not have the same parity as ab . The adversary can therefore compute the Legendre symbol of the published challenge to detect that fact, giving them a non-negligible advantage. However, if a, b and r are sampled to always be odd, the above strategy does not work and the problem is considered hard for classical computers. Since the fraction of trivial cases is non-negligible, this problem is not randomly self-reducible for classical attackers.

There are many ways to try to prove results for isogeny problems. For example, Atapoor, Bagheri, Cozzo and Pedersen [4] prove the security of their CSIDH-based Distributed Key Generation protocol by relying on algebra and combinatorics, without using any graph theory. However, the method we chose to prove random self-reducibility uses supersingular isogeny graphs. The core idea is that the supersingular isogeny graph is Ramanujan, as shown by Pizer [52]. The reason we care about Ramanujan graphs in this thesis is that random non-backtracking walks on a Ramanujan graph from a fixed starting vertex converge towards a uniform distribution of all vertices on the graph, with Jao, Miller and Venkatesan [41] giving a characterization of the rate of convergence. Since every cyclic isogeny can be represented by a non-backtracking walk on the isogeny graph, we can easily prove that the codomain of a random cyclic isogeny of large enough prime power degree is indistinguishable from a random supersingular elliptic curve. As we show in Theorem 3.1.1,

this fact is enough to show that the general Supersingular Isogeny Problem is randomly self-reducible.

The complication comes from the fact that SIDH variants are not based on the general problem, as they reveal the degree and some information on the mapping of the torsion subgroup.

To get around the degree transmission, we must show that a random supersingular isogeny of a given degree can potentially reach a non-negligible fraction of all supersingular elliptic curves.

For the second point, since we have torsion point information, we have to work with the supersingular isogeny graph with level structure. Luckily for us, there is a recent paper by Codogni and Lido [18] describing which level structures lead to Ramanujan graphs.

3.1 Main Result

We start by showing that the generic Supersingular Isogeny Problem is randomly reducible. Since the degree of the secret isogeny is secret and the generic problem leaks nothing, the only inputs are the domain and codomain curves. There is nothing stopping us from using the expander properties of the supersingular isogeny graph to convert fixed curves to random curves, making the following proof of random self-reducibility pretty simple.

Theorem 3.1.1. *Problem 2.5.4, that is, SIP, is randomly self-reducible.*

Proof. Let ℓ be any small prime number (e.g., 2) and consider E and F as vertices on the graph $\mathcal{G}_\ell$. Let $\mathcal{A}$ be an algorithm solving Problem 2.5.4 for a set $\mathcal{T}$ of pairs $(E, F) \in \mathcal{E}^2$. By hypothesis, there exists an algorithm $\mathcal{A}$ such that $\frac{|\mathcal{T}|}{|\mathcal{E}^2|}$ is non-negligible in terms of $\log(p)$. Let T_E be the set of values of F such that $\mathcal{A}$ solves Problem 2.5.4 for (E, F) . Let ε be a non-negligible function in terms of $\log(p)$ and let T' be the set of values of E for which the size of T_E is at least $\varepsilon|\mathcal{E}|$. Since, by hypothesis, $\mathcal{A}$ has a non-negligible probability of success, there must be a choice of ε such that $\delta := \frac{|T'|}{|\mathcal{E}|}$ is non-negligible. Consider the following algorithm, denoted REDU1.

```

1: function REDU1( $\ell, p, E, F$ )
2:   Evaluate  $\lambda = \left\lceil \frac{\log(2|\mathcal{E}|)}{\log((\ell+1)/(2\sqrt{\ell}))} \right\rceil$ 
3:   Let  $\phi_1 : E \rightarrow E'$  be a  $\ell^\lambda$ -isogeny generated by a random non-backtracking walk on
   the  $\ell$ -isogeny graph from  $E$ 
4:   Let  $\phi_2 : F \rightarrow F'$  be a  $\ell^\lambda$ -isogeny generated by a random non-backtracking walk on
   the  $\ell$ -isogeny graph from  $F$ 
5:   Use  $\mathcal{A}$  to evaluate an isogeny  $\psi : E' \rightarrow F'$ 
6:   if  $\mathcal{A}$  succeeds: return  $\hat{\phi}_2 \psi \phi_1$ 
7:   else : return false
8: end function

```

Since ℓ is small and λ is of polynomial size in terms of $\log(p)$, the evaluation of isogenies and duals can be done in polynomial time. Let α be the probability of success of $\mathcal{A}$ on the pairs that it can solve. By Theorem 2.4.6, the probability that we have $E' \in T'$ is at least $\frac{\delta}{2}$, and the probability that $F' \in T_E$ is at least $\frac{\varepsilon}{2}$. Hence, the probability of success of REDU1 is at least $\frac{\alpha\delta\varepsilon}{4}$, which is non-negligible. $\square$

When working with Γ -SIDH instead of SIP, we have to deal with the fact that the degree and level structure mapping of the secret isogeny are known. Luckily for us, we can obtain a similar result to the one for SIP if the associated supersingular isogeny graph with level structure is Ramanujan.

Theorem 3.1.2. *If $\mathcal{G}_{H,\ell}$ is Ramanujan and $\frac{|\Xi_{E,H,C,d}|}{|\mathcal{E}_H|}$ is non-negligible, as defined in Definition 2.5.1, for all $(E, C) \in \mathcal{E}_H$, then Problem 2.5.8 (Γ -SIDH) is randomly self-reducible.*

Proof. Let ℓ be a small prime number not dividing N .

Let $\mathcal{B}$ be an algorithm that can solve Problem 2.5.8 for a non-negligible fraction of pairs $((E, C), (F, D))$ with $(F, D) \in \Xi_{E,H,C,d}$.

For any given pair $((E, C), (F, D))$ and any secret d -isogeny $\psi : (E, C) \rightarrow (F, D)$, we can use the following reduction algorithm:

```

1: function REDU2( $\ell, p, (E, C), (F, D)$ )
2:   Evaluate  $\lambda = \left\lceil \frac{\log(2|\mathcal{E}_H|)}{\log((\ell+1)/(2\sqrt{\ell}))} \right\rceil$ 
3:   Let  $\phi_1: (E, C) \rightarrow (E', C')$  be a  $\ell^\lambda$ -isogeny generated by a random non-backtracking
   walk on the  $\ell$ -isogeny graph from  $(E, C)$ 
4:   Let  $\phi_2: (F, D) \rightarrow (F', D')$  be a  $\ell^\lambda$ -isogeny generated by a random non-backtracking
   walk on the  $\ell$ -isogeny graph from  $(F, D)$ 
5:   Use  $\mathcal{B}$  to evaluate an isogeny  $\psi': (E', C') \rightarrow (F', D')$ 
6:   if  $\mathcal{B}$  succeeds: return  $\phi_2\psi'\phi_1$ 
7:   else : return false
8: end function

```

We only need to check that the probability of success is non-negligible in terms of $\log(p)$. Let $T_{(E,C)}$ be the set of values of (F, D) such that $\mathcal{B}$ solves Problem 2.5.8 for $((E, C), (F, D))$. Let ε be a non-negligible function in terms of $\log(p)$ and let T' be the set of values of (E, C) for which the size of T_E is at least $\varepsilon|\Xi_{E,C,H,d}|$. Since the number of pairs $((E, C), (F, D))$ for which $\mathcal{B}$ solves Problem 2.5.8 is equal to $\sum_{(E,C) \in \mathcal{E}_H} |T_{(E,C)}|$, we have that, for a well chosen ε , $\frac{|T'|}{|\Xi_{E,C,H,d}|}$ is non-negligible. Because $\frac{|\Xi_{E,C,H,d}|}{|\mathcal{E}_H|}$ is also non-negligible, $\frac{|T'|}{|\mathcal{E}_H|}$ is a non-negligible fraction δ . Since $\frac{|\Xi_{E,C,H,d}|}{|\mathcal{E}_H|}$ is non-negligible for all $(E, C) \in \mathcal{E}_H$, it follows that (F', D') has a non-negligible probability to be in $\Xi_{E',C',H,d}$. Hence, $\mathcal{B}$ has a non-negligible probability of succeeding. $\square$

In order to use the above theorem, we require a way to compute a lower bound for $\frac{|\Xi_{E,H,C,d}|}{|\mathcal{E}_H|}$. As shown later in this chapter, we can obtain a useful lower bound for some level structures when d is large enough. In practice, we need d to be larger than the degree used in the current parameters by a small factor. To obtain the desired lower bound, we present a variant of a theorem by Sardari [57] that proves such a lower bound for Ramanujan graphs. Our variant works with walks on multiple isogeny graphs, allowing us to represent isogenies whose degree is not a prime power.

Theorem 3.1.3 ([57, Thm. 1.4]). *Let G be a k -regular Ramanujan graph with n vertices and fix a vertex $x \in V(G)$. Let $\epsilon > 0$ and let R be an integer such that $R > (1+\epsilon) \log_{k-1}(n)$. Define $M(x, R)$ to be the set of all vertices $y \in G$ such that there is no non-backtracking walk from x to y with length R . Then*

$$|M(x, R)| \leq n^{1-\epsilon}(1+R)^2.$$

In order to simulate isogenies whose degree has multiple prime factors, we have to show that going from one isogeny graph to the other does not cause us to do so much backtracking that we lose the expander property. To do so, we require multiple properties from our graphs, which are held when $p \equiv 1 \pmod{12}$. In order to obtain results applicable to non-prime power isogenies, we use different inequalities than Sardari, leading to our result having a slightly different shape.

Theorem 3.1.4. *Let $G_1, \dots, G_k$ be a family of Ramanujan graphs with the same common vertex set V , of size n , such that the following properties hold:*

1. G_i is a $(\ell_i + 1)$ -regular graph with a diagonalizable adjacency matrix, for some positive integer ℓ_i . (While not necessary for this theorem, ℓ_i is prime for our applications.)
2. G_i is an undirected multigraph, so that its adjacency matrix is symmetric.
3. $\gcd(\ell_i, \ell_j) = 1$ for $i \neq j$.
4. The adjacency matrices A_i and A_j of each pair of graphs G_i and G_j commute.

Let $\mathbf{e} = (e_1, \dots, e_k)$ be a vector of non-negative integers. Let $x \in V$ and let $\Xi_{\mathbf{e}}(x)$ be the set of vertices we can arrive at using the following algorithm.

1. Let $x_0 = x$.
2. For $1 \leq i \leq k$, take a random non-backtracking walk in G_i of length e_i starting at x_{i-1} and let the final point be x_i .
3. Return x_k .

Then

$$|\Xi_{\mathbf{e}}(x)| \geq n \left(1 - \frac{n}{\prod_{i=1}^k \ell_i^{e_i}} \prod_{i=1}^k \left(e_i + 1 + \frac{e_i - 1}{\ell_i} \right)^2 \right).$$

Remark 3.1.5. When $p \equiv 1 \pmod{12}$, G_i is taken to be the supersingular isogeny graph $\mathcal{G}_{\ell_i}$ over $\mathbb{F}_{p^2}$. Properties 1 and 2 come from Theorem 2.4.5, Property 3 is due to the fact that distinct supersingular isogeny graphs use distinct prime values of ℓ and Property 4 comes from Lemma 2.4.13. As mentioned in Remark 2.4.12, when N is prime and the level structure is non-trivial ($H \neq \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$), we have more cases where the isogeny graph is undirected and where this theorem applies. Namely, if $p \equiv 7 \pmod{12}$ and ℓ belongs to H , Theorem 3.1.4 still applies if $N \equiv 3 \pmod{4}$. Similarly, if $p \equiv 5 \pmod{12}$ and ℓ belongs to H , then our theorem applies when $N \equiv 2 \pmod{3}$, and if $p \equiv 11 \pmod{12}$ then it applies when $N \equiv 11 \pmod{12}$.

Proof. Let $S_i(R)$ be the adjacency matrix of the graph whose edges are the walks of length R on G_i with no backtracking. Using this matrix, we have that

$$\Xi_{\mathbf{e}}(x) = \{y \in V : \prod_{i=1}^k S_i(e_i)(x, y) > 0\} = V \setminus \{y \in V : \prod_{i=1}^k S_i(e_i)(x, y) = 0\}.$$

Hence

$$|\Xi_{\mathbf{e}}(x)| = |V| - \left| \{y \in V : \prod_{i=1}^k S_i(e_i)(x, y) = 0\} \right| = n - \left| \{y \in V : \prod_{i=1}^k S_i(e_i)(x, y) = 0\} \right|.$$

We can therefore obtain the desired $|\Xi_{\mathbf{e}}(x)|$ lower bound by showing that

$$\left| \{y \in V : \prod_{i=1}^k S_i(e_i)(x, y) = 0\} \right| \leq \frac{n^2}{d} \prod_{i=1}^k \left(e_i + 1 + \frac{e_i - 1}{\ell_i} \right)^2.$$

To do so, we can fix x and let y be a random variable whose distribution is a uniformly random vertex in V . We have that

$$\left| \{y \in V : \prod_{i=1}^k S_i(e_i)(x, y) = 0\} \right| = n \Pr_y \left[\prod_{i=1}^k S_i(e_i)(x, y) = 0 \right]$$

which allows us to use techniques from probability theory.

This proof can be separated into 4 four steps:

1. Obtain a useful formula for $\prod_{i=1}^k S_i(e_i)(x, y)$.
2. Compute $E_y \left[\prod_{i=1}^k S_i(e_i)(x, y) \right]$.
3. Compute an upper bound for $\text{Var}_y \left[\prod_{i=1}^k S_i(e_i)(x, y) \right]$.
4. Use Chebyshev's inequality to obtain the desired upper bound on $\left| \{y \in V : \prod_{i=1}^k S_i(e_i)(x, y) = 0\} \right|$.

Step 1: Obtain a useful formula.

Using the results from Alon, Benjamini, Lubetzky and Sodin [2], we have that

$$S_i(R) = \begin{cases} I & \text{if } R = 0, \\ A_i & \text{if } R = 1, \\ A_i^2 - (\ell_i + 1)I & \text{if } R = 2, \\ A_i S_i(R-1) - \ell_i S_i(R-2) & \text{otherwise.} \end{cases}$$

From the above recurrence, we obtain the equation

$$S_i(R) = (\ell_i)^{\frac{R}{2}} U_R \left(\frac{A_i}{2\sqrt{\ell_i}} \right) - (\ell_i)^{\frac{R-2}{2}} U_{R-2} \left(\frac{A_i}{2\sqrt{\ell_i}} \right)$$

where U_R is the Chebyshev polynomial of the second kind defined in Definition 2.10.1. For the sake of notation, let

$$Y_{R,\ell}(x) = U_R(x) - \frac{1}{\ell} U_{R-2}(x)$$

so that

$$S_i(R) = (\ell_i)^{\frac{R}{2}} Y_{R,\ell_i} \left(\frac{A_i}{2\sqrt{\ell_i}} \right).$$

Since the adjacency matrices commute and are diagonalizable, Theorem 2.10.3 implies that there is a common basis of eigenvectors. Denote these (normalized) eigenvectors by ϕ_j , where ϕ_1 is the constant eigenvector (which exists since we are dealing with regular graphs).

Let $\lambda_{i,j}$ be the eigenvalue associated with ϕ_j for A_i . Observe that, since A_i is diagonalizable, we can write it as

$$\sum_{j=1}^n \lambda_{i,j} \phi_j^t \phi_j.$$

Since for any fixed R , $S_i(R)$ is a polynomial in A_i , we can use Theorem 2.10.5 to obtain

$$S_i(R) = \sum_{j=1}^n (\ell_i)^{\frac{R}{2}} Y_{R,\ell_i} \left(\frac{\lambda_{i,j}}{2\sqrt{\ell_i}} \right) \phi_j^t \phi_j.$$

With multiplicity, the set of non-backtracking **e**-walks is represented by

$$\begin{aligned}
\prod_{i=1}^k S_i(e_i) &= \prod_{i=1}^k \sum_{j=1}^n (\ell_i)^{\frac{e_i}{2}} Y_{e_i, \ell_i} \left(\frac{\lambda_{i,j}}{2\sqrt{\ell_i}} \right) \phi_j^t \phi_j \\
&= \sum_{1 \leq j_1, \dots, j_k \leq n} \prod_{i=1}^k \left((\ell_i)^{\frac{e_i}{2}} Y_{e_i, \ell_i} \left(\frac{\lambda_{i,j_i}}{2\sqrt{\ell_i}} \right) \phi_{j_i}^t \phi_{j_i} \right) \\
&= \sum_{1 \leq j_1, \dots, j_k \leq n} \prod_{i=1}^k \left((\ell_i)^{\frac{e_i}{2}} Y_{e_i, \ell_i} \left(\frac{\lambda_{i,j_i}}{2\sqrt{\ell_i}} \right) \right) \prod_{i=1}^k (\phi_{j_i}^t \phi_{j_i}).
\end{aligned}$$

Since the set of ϕ_j consists of eigenvectors of real symmetric matrices, said vectors are orthonormal. Hence, $\prod_{i=1}^k (\phi_{j_i}^t \phi_{j_i})$ is equal to 1 if all j_i 's are equal and 0 otherwise, so

$$\prod_{i=1}^k S_i(e_i) = \sum_{j=1}^n \phi_j^t \phi_j \prod_{i=1}^k (\ell_i)^{\frac{e_i}{2}} Y_{e_i, \ell_i} \left(\frac{\lambda_{i,j}}{2\sqrt{\ell_i}} \right).$$

For any two vertices x and y of V ,

$$\prod_{i=1}^k S_i(e_i)(x, y) = \sum_{j=1}^n \phi_j(x) \phi_j(y) \prod_{i=1}^k (\ell_i)^{\frac{e_i}{2}} Y_{e_i, \ell_i} \left(\frac{\lambda_{i,j}}{2\sqrt{\ell_i}} \right)$$

is equal to the number of **e**-walks from x to y .

Step 2: Compute the expected value.

Let x be a fixed vertex of V , and let y be a uniformly distributed random vertex of G . Since the total number of **e**-walks is constant regardless of the choice of x (because the graphs are regular), we have that

$$E_y \left[\prod_{i=1}^k S_i(e_i)(x, y) \right] = \frac{\prod_{i=1}^k (\ell_i^{e_i-1} (\ell_i + 1))}{n}.$$

Step 3: Compute an upper bound on the variance.

We also want to evaluate

$$\text{Var}_y \left[\prod_{i=1}^k S_i(e_i)(x, y) \right] = \text{Var}_y \left[\sum_{j=1}^n \phi_j(x) \phi_j(y) \prod_{i=1}^k (\ell_i)^{\frac{e_i}{2}} Y_{e_i, \ell_i} \left(\frac{\lambda_{i,j}}{2\sqrt{\ell_i}} \right) \right].$$

Note ϕ_1 is the constant eigenvector, which implies that

$$\phi_1(x)\phi_1(y) \prod_{i=1}^k (\ell_i)^{\frac{e_i}{2}} Y_{e_i, \ell_i} \left(\frac{\lambda_{i,1}}{2\sqrt{\ell_i}} \right)$$

does not depend on y and can be ignored when evaluating the variance.

$$\begin{aligned} \text{Var}_y \left[\prod_{i=1}^k S_i(e_i)(x, y) \right] &= \text{Var}_y \left[\sum_{j=2}^n \phi_j(x) \phi_j(y) \prod_{i=1}^k (\ell_i)^{\frac{e_i}{2}} Y_{e_i, \ell_i} \left(\frac{\lambda_{i,j}}{2\sqrt{\ell_i}} \right) \right] \\ &= E_y \left[\left(\sum_{j=2}^n \phi_j(x) \phi_j(y) \prod_{i=1}^k (\ell_i)^{\frac{e_i}{2}} Y_{e_i, \ell_i} \left(\frac{\lambda_{i,j}}{2\sqrt{\ell_i}} \right) \right)^2 \right] \\ &\quad - E_y \left[\sum_{j=2}^n \phi_j(x) \phi_j(y) \prod_{i=1}^k (\ell_i)^{\frac{e_i}{2}} Y_{e_i, \ell_i} \left(\frac{\lambda_{i,j}}{2\sqrt{\ell_i}} \right) \right]^2 \\ &\leq E_y \left[\left(\sum_{j=2}^n \phi_j(x) \phi_j(y) \prod_{i=1}^k (\ell_i)^{\frac{e_i}{2}} Y_{e_i, \ell_i} \left(\frac{\lambda_{i,j}}{2\sqrt{\ell_i}} \right) \right)^2 \right] \\ &= \frac{1}{n} \sum_{y \in V} \left(\sum_{j=2}^n \phi_j(x) \phi_j(y) \prod_{i=1}^k (\ell_i)^{\frac{e_i}{2}} Y_{e_i, \ell_i} \left(\frac{\lambda_{i,j}}{2\sqrt{\ell_i}} \right) \right)^2 \\ &= \frac{1}{n} \sum_{y \in V} \sum_{j_1=2}^n \sum_{j_2=2}^n \phi_{j_1}(x) \phi_{j_2}(x) \phi_{j_1}(y) \phi_{j_2}(y) \\ &\quad \left(\prod_{i=1}^k (\ell_i)^{\frac{e_i}{2}} Y_{e_i, \ell_i} \left(\frac{\lambda_{i,j_1}}{2\sqrt{\ell_i}} \right) \right) \left(\prod_{i=1}^k (\ell_i)^{\frac{e_i}{2}} Y_{e_i, \ell_i} \left(\frac{\lambda_{i,j_2}}{2\sqrt{\ell_i}} \right) \right). \end{aligned}$$

Since the ϕ_j 's are orthonormal, we have that

$$\begin{aligned} \text{Var}_y \left[\prod_{i=1}^k S_i(e_i)(x, y) \right] &\leq \frac{1}{n} \sum_{j_1=2}^n \sum_{j_2=2}^n \phi_{j_1}(x) \phi_{j_2}(x) \left(\prod_{i=1}^k (\ell_i)^{\frac{e_i}{2}} Y_{e_i, \ell_i} \left(\frac{\lambda_{i,j_1}}{2\sqrt{\ell_i}} \right) \right) \\ &\quad \left(\prod_{i=1}^k (\ell_i)^{\frac{e_i}{2}} Y_{e_i, \ell_i} \left(\frac{\lambda_{i,j_2}}{2\sqrt{\ell_i}} \right) \right) \sum_{y \in V} \phi_{j_1}(y) \phi_{j_2}(y) \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{n} \sum_{j_1=2}^n \sum_{j_2=2}^n \phi_{j_1}(x) \phi_{j_2}(x) \left(\prod_{i=1}^k (\ell_i)^{\frac{e_i}{2}} Y_{e_i, \ell_i} \left(\frac{\lambda_{i, j_1}}{2\sqrt{\ell_i}} \right) \right) \\
&\quad \left(\prod_{i=1}^k (\ell_i)^{\frac{e_i}{2}} Y_{e_i, \ell_i} \left(\frac{\lambda_{i, j_2}}{2\sqrt{\ell_i}} \right) \right) \langle \phi_{j_1}, \phi_{j_2} \rangle \\
&= \frac{1}{n} \sum_{j=2}^n \phi_j(x)^2 \left(\prod_{i=1}^k (\ell_i)^{e_i} Y_{e_i, \ell_i} \left(\frac{\lambda_{i, j}}{2\sqrt{\ell_i}} \right)^2 \right).
\end{aligned}$$

When $j \geq 2$, we have $\lambda_{i, j} \leq 2\sqrt{\ell_i}$ since G_i is Ramanujan. Hence $-1 \leq \frac{\lambda_{i, j}}{2\sqrt{\ell_i}} \leq 1$. Theorem 2.10.2 therefore implies that

$$-(e_i + 1) \leq U_{e_i} \left(\frac{\lambda_{i, j}}{2\sqrt{\ell_i}} \right) \leq e_i + 1.$$

Therefore

$$\begin{aligned}
Y_{e_i, \ell_i} \left(\frac{\lambda_{i, j}}{2\sqrt{\ell_i}} \right)^2 &= \left(U_{e_i} \left(\frac{\lambda_{i, j}}{2\sqrt{\ell_i}} \right) - \frac{1}{\ell_i} U_{e_i-2} \left(\frac{\lambda_{i, j}}{2\sqrt{\ell_i}} \right) \right)^2 \\
&\leq \left(\left| U_{e_i} \left(\frac{\lambda_{i, j}}{2\sqrt{\ell_i}} \right) \right| + \frac{1}{\ell_i} \left| U_{e_i-2} \left(\frac{\lambda_{i, j}}{2\sqrt{\ell_i}} \right) \right| \right)^2 \\
&\leq \left(e_i + 1 + \frac{e_i - 1}{\ell_i} \right)^2.
\end{aligned}$$

Hence

$$\begin{aligned}
\text{Var}_y \left[\prod_{i=1}^k S_i(e_i)(x, y) \right] &\leq \frac{1}{n} \sum_{j=2}^n \phi_j(x)^2 \left(\prod_{i=1}^k (\ell_i)^{e_i} Y_{e_i, \ell_i} \left(\frac{\lambda_{i,j}}{2\sqrt{\ell_i}} \right)^2 \right) \\
&= \frac{\prod_{i=1}^k (\ell_i)^{e_i}}{n} \sum_{j=2}^n \phi_j(x)^2 \prod_{i=1}^k Y_{e_i, \ell_i} \left(\frac{\lambda_{i,j}}{2\sqrt{\ell_i}} \right)^2 \\
&\leq \frac{\prod_{i=1}^k (\ell_i)^{e_i} \left(e_i + 1 + \frac{e_i-1}{\ell_i} \right)^2}{n} \sum_{j=2}^n \phi_j(x)^2 \\
&< \frac{\prod_{i=1}^k (\ell_i)^{e_i} \left(e_i + 1 + \frac{e_i-1}{\ell_i} \right)^2}{n}.
\end{aligned}$$

since $\sum_{j=2}^n \phi_j(x)^2 < 1$ as the ϕ_j 's form an orthonormal basis missing one vector.

Step 4: Complete the proof.

We now use Chebyshev's inequality:

$$\begin{aligned}
&\Pr \left[\prod_{i=1}^k S_i(e_i)(x, y) = 0 \right] \\
&\leq \Pr \left[\left| \prod_{i=1}^k S_i(e_i)(x, y) - E_y \left[\prod_{i=1}^k S_i(e_i)(x, y) \right] \right| \geq E_y \left[\prod_{i=1}^k S_i(e_i)(x, y) \right] \right] \\
&\leq \frac{\text{Var}_y \left[\prod_{i=1}^k S_i(e_i)(x, y) \right]}{E_y \left[\left(\prod_{i=1}^k S_i(e_i)(x, y) \right)^2 \right]} \\
&< n \frac{\prod_{i=1}^k (\ell_i)^{e_i} \left(e_i + 1 + \frac{e_i-1}{\ell_i} \right)^2}{\prod_{i=1}^k (\ell_i^{e_i-1} (\ell_i + 1))^2} \\
&< n \frac{\prod_{i=1}^k \left(e_i + 1 + \frac{e_i-1}{\ell_i} \right)^2}{\prod_{i=1}^k \ell_i^{e_i}}.
\end{aligned}$$

Therefore

$$\begin{aligned} \left| \{y \in V : \prod_{i=1}^k S_i(e_i)(x, y) = 0\} \right| &< n^2 \frac{\prod_{i=1}^k \left(e_i + 1 + \frac{e_i - 1}{\ell_i} \right)^2}{\prod_{i=1}^k \ell_i^{e_i}} \\ &= \frac{n^2}{\prod_{i=1}^k \ell_i^{e_i}} \prod_{i=1}^k \left(e_i + 1 + \frac{e_i - 1}{\ell_i} \right)^2, \end{aligned}$$

so that

$$\begin{aligned} |\Xi_{\mathbf{e}}(x)| &= \left| \{y \in V : \prod_{i=1}^k S_i(e_i)(x, y) > 0\} \right| \\ &= n - \left| \{y \in V : \prod_{i=1}^k S_i(e_i)(x, y) = 0\} \right| \\ &\geq n - \frac{n^2}{\prod_{i=1}^k \ell_i^{e_i}} \prod_{i=1}^k \left(e_i + 1 + \frac{e_i - 1}{\ell_i} \right)^2 \\ &\geq n \left(1 - \frac{n}{\prod_{i=1}^k \ell_i^{e_i}} \prod_{i=1}^k \left(e_i + 1 + \frac{e_i - 1}{\ell_i} \right)^2 \right). \end{aligned}$$

which is the desired inequality. $\square$

Combining Theorem 3.1.2 and Theorem 3.1.4 with Theorem 2.4.10, we obtain:

Corollary 3.1.6. *Let $d = \prod_{i=1}^k \ell_i^{e_i}$ be the factorization of the degree of the secret isogeny in Problem 2.5.8. If $p \equiv 1 \pmod{12}$, and*

$$\frac{d}{|\mathcal{E}_H| \prod_{i=1}^k \left(e_i + 1 + \frac{e_i - 1}{\ell_i} \right)^2}$$

is non-negligible, and H is CL-admissible, then Problem 2.5.8 is randomly self-reducible.

Proof. Let c be the smallest positive integer such that

$$\begin{aligned} 1 - \frac{n}{\ell_1^{e_1+c} \prod_{i=2}^k \ell_i^{e_i}} (e_1 + c) \prod_{i=2}^k \left(e_i + 1 + \frac{e_i - 1}{\ell_i} \right)^2 \\ = 1 - \left(\frac{d}{|\mathcal{E}_H| \prod_{i=1}^k \left(e_i + 1 + \frac{e_i - 1}{\ell_i} \right)^2} \right)^{-1} \frac{(e_1 + c)}{e_1 \ell_1^c} \end{aligned}$$

is positive and non-negligible. By hypothesis, $\left(\frac{d}{|\mathcal{E}_H| \prod_{i=1}^k \left(e_i + 1 + \frac{e_i - 1}{\ell_i}\right)^2}\right)^{-1}$ can be upper bounded by a polynomial in terms of the input size. Therefore, a value of c respecting the desired conditions exists and is also such that ℓ_1^c is of polynomial size. As a direct result of Theorems 3.1.2 and 3.1.4, when the conditions for Theorem 2.4.10 to apply are met, we have that Problem 2.5.8 is randomly self-reducible when the degree of the secret isogeny is $\ell_1^c d$.

We remark that Problem 2.5.8 for a secret isogeny of degree d is polynomially equivalent to the same problem for a secret isogeny of degree $\ell_1^c d$. This is because, in the latter case, we can write the secret isogeny as $\phi_1 \phi_2$ where ϕ_1 has degree d and ϕ_2 has degree ℓ_1^c . Since ℓ_1^c is polynomial, we can simply guess ϕ_2 and reduce the problem to an instance where the secret isogeny has degree d .

Hence, Problem 2.5.8 is also randomly self-reducible for the cases where the secret isogeny is of degree d . $\square$

Remark 3.1.7. *For the choice of what is non-negligible, as for every other result in this thesis, we define it as in Definition 2.5.1. For an actual implementation, the fraction in the above corollary can be seen as an estimation for a lower bound on the probability of the reduction algorithm from Theorem 3.1.2 succeeding in a single attempt. As such, when choosing parameter sets, we will want this fraction to be large enough for the expected number of required attempts to be feasible, depending on the computational resources available.*

Remark 3.1.8. *This result still applies when $p \not\equiv 1 \pmod{12}$, if N is prime and the conditions discussed in Remark 3.1.5 are respected. Namely, if $p \equiv 7 \pmod{12}$ and ℓ belongs to H , Corollary 3.1.6 still applies if $N \equiv 3 \pmod{4}$. Similarly, if $p \equiv 5 \pmod{12}$ and ℓ belongs to H , then our corollary applies when $N \equiv 2 \pmod{3}$, and if $p \equiv 11 \pmod{12}$ then it applies when $N \equiv 11 \pmod{12}$.*

If $p \not\equiv 1 \pmod{12}$, we can still prove random self-reducibility when the secret degree is a prime power (i.e. $k = 1$), by running through the proof of Corollary 3.1.6 and applying Theorem 3.1.3 instead of Theorem 3.1.4.

Corollary 3.1.9. *In the case where d is a prime power, let $d = \ell_1^{e_1}$ be the factorization of the degree of the secret isogeny in Problem 2.5.8. If*

$$\frac{\ell_1^{e_1}}{|\mathcal{E}_H| \left(e_1 + 1 + \frac{e_1 - 1}{\ell_1}\right)^2}$$

is non-negligible, and H is CL-admissible, then Problem 2.5.8 is randomly self-reducible.

3.2 Applications to Isogeny Problems

Using the results of the previous section, we can now explore which choices of H make Problem 2.5.8 potentially randomly self-reducible. In such cases, we give an upper bound for the required secret isogeny degree.

When studying these cases, we assume that N is a smooth number and that at least one of the following cases applies:

1. $p \equiv 1 \pmod{12}$,
2. the secret isogeny degree is a prime power,
3. N is prime and one of the following subconditions applies:
 - (a) $p \equiv 5 \pmod{12}$ and $N \equiv 2 \pmod{3}$,
 - (b) $p \equiv 7 \pmod{12}$ and $N \equiv 3 \pmod{4}$,
 - (c) $p \equiv 11 \pmod{12}$ and $N \equiv 11 \pmod{12}$.

Not that the third case asks for N to be both smooth and prime at the same time. This forces N to be small, which limits the applications of this case.

3.2.1 $H = GL_2(\mathbb{Z}/N\mathbb{Z})$

This version of Γ -SIDH is equivalent to not transmitting any information about the images of the torsion points, but having the degree be public. In the generic level structure case, the necessary conditions for Corollary 3.1.6 hold and $|\mathcal{E}_H| = |\mathcal{E}| \sim \frac{p}{12}$. Problem 2.5.8 is therefore randomly self-reducible as long as

$$\frac{12d}{p \prod_{i=1}^k (e_i + 1 + \frac{e_i - 1}{\ell_i})^2}$$

is non-negligible in terms of $\log(p)$.

Usually, we can assume that d contains a small number k of distinct prime factors. In such cases, $\prod_{i=1}^k (e_i + 1 + \frac{e_i - 1}{\ell_i})^2$ can be estimated as $\log(p)^{2k}$, which is small when compared to p . Therefore, in most cases, we have that the problem is randomly self-reducible so long as d is not vastly smaller than p .

$$\mathbf{3.2.2} \quad H = \left\{ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right\}$$

When $N^2 > d$, the full level structure case can be attacked in polynomial time, as shown by the SIDH attacks [14, 42, 55], making it trivially randomly self-reducible. This inequality holds for most key exchanges based on SIDH since the values of N and d for one party are flipped for the other, making the attack work against at least one party.

If one were to use parameters such that the SIDH attacks do not hold, for example, having $d > N^2$, our results would not apply since H does not contain every scalar matrix. According to Codogni and Lido [18], the isogeny graph in this case is connected if and only if ℓ generates $(\mathbb{Z}/N\mathbb{Z})^\times$, and walks on the graph will not reach every vertex when the graph is not connected. When the isogeny graph is not connected, we cannot apply Theorem 3.1.4 since the graph is not Ramanujan. Imagine a degenerate algorithm that could perfectly solve the problem on half the connected components, but would always fail on the others. A random walk could not be used to reduce a solution on a successful component to a solution on a failure component. It is worth noting that the graph components are known to be isomorphic [18, Cor. 2.3.7]. If the isomorphism is efficiently computable, then one could use the isomorphism to obtain an equivalence of problems across components. However, if an isomorphism is not efficiently computable, then a reduction is not guaranteed. Of course, this does not necessarily mean that the problem is not randomly self-reducible, only that if it were, the reduction would require different techniques. Another option would be to alter the Γ -SIDH problem to only be defined on a component and then study the expander properties of said component.

Even in the cases where the isogeny graph is connected, Corollary 3.1.6 would not apply as is, because, according to [18], the isogeny graph contains non-real eigenvalues. This does not respect Condition 2 of Theorem 3.1.4 since non-real eigenvalues would imply that the adjacency matrix is not symmetric. The proof of Theorem 3.1.4 relies on the fact that every eigenvalue is real, in order to use a known upper bound on Chebyshev polynomials of the second kind. It might be possible to generalize our results to apply here by first proving an upper bound for Chebyshev polynomials in the complex disk of radius 1.

$$\mathbf{3.2.3} \quad H = \left\{ \begin{pmatrix} 1 & * \\ 0 & 1 \end{pmatrix} \right\}$$

As shown by De Feo, Fouotsa and Panny [26], the single torsion point level structure case can be reduced to SIDH, making it also solvable in polynomial time when N is a square

and $N > d$. In these cases, the problem is trivially randomly self-reducible.

Similarly to the SIDH case, if the parameters are chosen so that no polynomial attack is known, our results do not apply since H does not contain every scalar matrix. According to Codogni and Lido [18], when $H = \left\{ \begin{pmatrix} 1 & * \\ 0 & 1 \end{pmatrix} \right\}$, the isogeny graph is connected if and only if ℓ generates $(\mathbb{Z}/N\mathbb{Z})^\times$, but the eigenvalues are not all real. This leads to the same issues as the previous case, with similar avenues for further research.

3.2.4 $H = \left\{ \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \right\}$

Unfortunately, the results in this thesis do not apply to M-SIDH (and MD-SIDH) since H does not contain matrices of non-square determinant. By [18, Thm. 1.6], if ℓ is a square in $(\mathbb{Z}/N\mathbb{Z})^\times$, then the isogeny graph is not connected, and we run into the same issues as in the previous subsections. On the other hand, if ℓ is relatively prime to N and not a square in $(\mathbb{Z}/N\mathbb{Z})^\times$, then the isogeny graph is connected and all of its eigenvalues are real. However, for these values of ℓ , the isogeny graphs have both $(\ell + 1)$ and $-(\ell + 1)$ as eigenvalues because the graph is bipartite. This does not respect the necessary conditions for Theorem 3.3 to apply since the isogeny graph is not Ramanujan when it has this extra trivial eigenvalue. For the proof of Theorem 3.1.4 to work, we require every eigenvalue, except for the trivial eigenvalue $(\ell + 1)$ having multiplicity one, to be at most $2\sqrt{\ell}$ in absolute value. This is because, when computing the variance in the proof of Theorem 3.1.4, the largest eigenvalue can be ignored as its eigenvector is the constant eigenvector and therefore has no influence on said variance. This trick does not work with any other eigenvalue, and we then have to use upper bounds on Chebyshev's polynomials of the second kind. For the eigenvalue $-(\ell + 1)$, we currently have no usable upper bound, and this causes us similar issues as when the eigenvalues are non-real. While this argument does not prove that the problem is not random self-reducible, it does provide intuition for why M-SIDH might not be randomly self-reducible. For further research, one could find a good upper bound to the Chebyshev polynomial when the eigenvalue is large, perhaps by studying the trigonometric definition of these polynomials. Alternatively, one could use the properties of the eigenvector associated to $-(\ell + 1)$, whose terms are $\frac{1}{\sqrt{n}}$ on one part and $\frac{-1}{\sqrt{n}}$ on the other, in order to find a trick similar to the one we use to eliminate $(\ell + 1)$.

$$\mathbf{3.2.5} \quad H = \left\{ \begin{pmatrix} * & * \\ 0 & * \end{pmatrix} \right\}$$

As mentioned in [26], this case is the Borel level structure case. We work with all upper (or lower) triangular matrices. This level structure mainly applies to protocols for zero-knowledge proofs using SIDH squares, for example [25, 40, 6]. Here H is CL-admissible. Therefore, our results apply.

We have that $|H| = N\varphi(N)^2$. Let $N = \prod_{k=1}^t \rho_k^{\alpha_k}$ and $d = \prod_{i=1}^k \ell_i^{e_i}$. Applying our results, we find that Problem 2.5.8 is randomly self-reducible as long as

$$\frac{12dN\varphi(N)^2}{p \left(\prod_{k=1}^t \rho_k^{4(\alpha_k-1)} (\rho_k^2 - 1)(\rho_k^2 - \rho_k) \right) \prod_{i=1}^k (e_i + 1 + \frac{e_i-1}{\ell_i})^2}$$

is not negligible.

To get an idea of the size of the above fraction for the current scheme, take, for example, the zero-knowledge proof found in the paper by Basso, Codogni, Connolly, De Feo, Fouotsa, Lido, Morrison, Panny, Patranabis and Wesolowski [6]. One of the parameter sets used in the paper is the one from SIKE434. In this case, we have that $p = 2^{216} \times 3^{137} - 1$, $N = 2^{216}$ and $d = 3^{137}$. The formula becomes:

$$\frac{12 \times 2^{216} \times 3^{137} 2^{2 \times 216 - 2}}{(2^{216} \times 3^{137} - 1) (2^{4(216-1)} (2^2 - 1)(2^2 - 2)) (137 + 1 + \frac{137-1}{3})^2} \approx 2.15 \times 10^{-134}.$$

Sadly, this is much too small to be useful. However, the zero-knowledge proof in [6] could be modified so that the d -isogeny is longer. For an example of parameters for which the formula is useful, let $p = 2^a 3^b - 1$, $d = 3^{3b}$ and $N = 2^a$. The formula becomes:

$$\frac{12 \times 3^{3b} 2^{3a-2}}{(2^a 3^b - 1) (2^{4(a-1)} (2^2 - 1)(2^2 - 2)) (b + 1 + \frac{b-1}{3})^2} \approx \frac{3^{2b}}{2^{2a}}$$

If $2^a \approx 3^b$, then the problem is randomly self-reducible. Using the same prime as before, a valid parameter set would be $p = 2^{216} \times 3^{137} - 1$, $N = 2^{216}$ and $d = 3^{411}$. In this case, the formula is equal to approximately 1.29×10^{-4} . As this is clearly non-negligible, this parameter set leads us to a randomly self-reducible problem. Of course, this is a lower bound on the probability of success of the reduction algorithm and not on the length of the algorithm itself. As such, it is possible for the reduction to be more efficient in practice,

in which case we could use smaller parameters. Still, with the above choice of parameters, we would need to compute isogeny chains of three times the length as those used in SIDH. While there is a required loss of efficiency if one wishes to use our results in order to obtain random self-reducibility, it is only by a small factor.

$$\mathbf{3.2.6} \quad H = \left\{ \begin{pmatrix} * & 0 \\ 0 & * \end{pmatrix} \right\}$$

As mentioned in [26], this case is the split Cartan level structure case. Here H is CL-admissible. This level structure applies to FESTA [9]. Therefore, we can apply our results, with a different choice of parameters than the one in the FESTA paper.

We have that $|H| = \varphi(N)^2$. Let $N = \prod_{k=1}^t \rho_k^{\alpha_k}$ and $d = \prod_{i=1}^k \ell_i^{e_i}$. Applying our results, we have that Problem 2.5.8 is randomly self-reducible as long as

$$\frac{12d\varphi(N)^2}{p \left(\prod_{k=1}^t \rho_k^{4(\alpha_k-1)} (\rho_k^2 - 1)(\rho_k^2 - \rho_k) \right) \prod_{i=1}^k (e_i + 1 + \frac{e_i-1}{\ell_i})^2}$$

is not negligible.

Similarly to the case in Subsection 3.2.5, the above formula is too small to apply to the current FESTA parameters. As a rough estimate, to get a random self-reduction with a feasible probability of succeeding, the degree d of the secret isogeny should be around four times larger than N . For an example of a parameter set for which the formula is useful, let $p = 2^a 3^b - 1$, $d = 3^{4b}$ and $N = 2^a$. The formula becomes:

$$\frac{12 \times 3^{4b} 2^{2a-2}}{(2^a 3^b - 1) (2^{4(a-1)} (2^2 - 1)(2^2 - 2)) (b + 1 + \frac{b-1}{3})^2} \approx \frac{3^{3b}}{2^{3a}}$$

If $2^a \approx 3^b$, then the problem is randomly self-reducible. This is similar to the previous case, with an efficiency loss factor of about 4 instead of 3. For example, we could choose $p = 2^{216} \times 3^{137} - 1$, $N = 2^{216}$ and $d = 3^{548}$. In this case, the formula above is approximately equal to 1.60×10^{-4} . Hence, the problem is randomly self-reducible for this parameter set.

3.3 Conclusion

Using the results in this chapter, we find that there are families of SIDH variants for which the problem of recovering the secret key can be shown to be randomly self-reducible,

providing support for the strategy of choosing a random starting curve for these schemes. In particular, our results hold for Borel and Cartan level structures, linked to zero-knowledge proofs (such as [25, 40, 6]) and FESTA respectively.

As seen in the previous section, in order for our results to guarantee random self-reducibility for Borel and Cartan level structures, the degree of the secret isogeny needs to be larger than those found in the current parameter sets by a small power. Assuming that the field and torsion sizes are the same as in current protocols, this translates to requiring the secret key to be longer by a small factor.

In cases where our theorems do not imply random self-reducibility, such as for M-SIDH, we have some plausibility arguments for why random self-reducibility might not hold, since the associated isogeny graphs are either not connected, or have an extra trivial eigenvalue and eigenvector. Of course, this conclusion does not rule out the possibility of finding a reduction using another method.

The most direct avenue for future work would be to remove the conditions on p and N modulo 12. In practice, for any supersingular isogeny graph, there are at most two vertices that do not act like an undirected graph. As it is an exponentially small fraction of the total number of vertices in the graph, it would be surprising if they alone could affect the random self-reducibility of the problem. On the other hand, small changes in a mathematical object can lead some of its properties to drastically change. For example, changing a single term of a large degree polynomial can completely change the values of its roots. Another direction for future work would be to improve the efficiency of our reduction. For our results, we only use the fact that the graphs we are working with are Ramanujan, giving us an upper bound on the second-largest eigenvalue. One could, in theory, obtain better reductions by finding smaller upper bounds on some eigenvalues and applying them in the proof of Theorem 3.1.4.

Chapter 4

Starting Curves

In Chapter 3, we have seen that if the parameter sizes and level structure are chosen appropriately, then [F-SIDH](#) is randomly self-reducible. A consequence of this fact is that, in such cases, having the starting elliptic curve be random, as opposed to a special curve, can only improve security.

The question now is: how do we generate a random supersingular elliptic curve without anybody knowing its endomorphism ring?

If the curve did not need to be supersingular, this would be an easy problem since there are known formulas to generate elliptic curves of a given j -invariant. Hence, we would only need to sample a random element of $\mathbb{F}_{p^2}$ to be the j -invariant. However, there are p^2 such curves while only about $\frac{p}{12}$ of them are supersingular. This technique, therefore, has an exponentially low probability of success.

The actual way we generate random supersingular curves in practice is to start with a known supersingular elliptic curve and then compute a random isogeny of large degree with it as the domain. Because of the expander properties given by [Theorem 2.4.6](#), we can obtain a supersingular elliptic curve that is indistinguishable from a uniformly random one.

However, the party that computes the random isogeny knows it and, because of [Theorem 2.5.6](#), they are able to compute the endomorphism ring of the new curve. There is currently no known other way to efficiently generate random supersingular elliptic curves. This means that the problem of a single party generating such a curve in a manner that can be trusted still has not been solved [\[46\]](#).

While the problem is open in a single-party setting, we can use multiparty protocols. The basic idea is simple. Given n parties that do not trust each other, they can, in turn,

generate a large degree isogeny and publish the codomain, with the next party using the new curve as their starting curve. This creates a long isogeny that no subset of $n - 1$ parties can compute, meaning that the final curve has an unknown endomorphism ring so long as no party cheats.

Of course, we have to assume that some parties will try to cheat. In order to avoid this scenario, we can have each party do zero-knowledge proofs in order to prove that they compute their isogeny honestly. This method has been used by Basso, Codogni, Connolly, De Feo, Fouotsa, Lido, Morrison, Panny, Patranabis and Wesolowski [6] to generate supersingular elliptic curves defined over $\mathbb{F}_{p^2}$ with unknown endomorphism rings.

However, some schemes, such as CSIDH, require the starting elliptic curve to be defined over $\mathbb{F}_p$ and, as we will discuss in the next section, the Basso, Codogni, Connolly, De Feo, Fouotsa, Lido, Morrison, Panny, Patranabis and Wesolowski signature scheme is not zero-knowledge when we restrict isogenies to be compatible with CSIDH. Hence, in this chapter, we present our work to create a multiparty scheme that works for curves over $\mathbb{F}_p$.

This work was published in [43] and also contains an algorithm to generate random curves of unknown endomorphism ring over either field.

4.1 Existing Solutions

We start by presenting a brief overview of possible ways to generate supersingular elliptic curves of unknown endomorphism ring that were either presented or mentioned in previous papers before ours.

4.1.1 Signature Schemes

In the model used by Basso, Codogni, Connolly, De Feo, Fouotsa, Lido, Morrison, Panny, Patranabis and Wesolowski [6], we have n parties $\mathcal{P}_1, \dots, \mathcal{P}_n$ whose goal is to generate a supersingular elliptic curve of unknown endomorphism ring. A starting elliptic curve E_0 is provided. Their idea is to have each party $\mathcal{P}_i$ in turn compute a random isogeny $\phi_i : E_{i-1} \rightarrow E_i$ and publish E_i . If all parties act honestly and do not share their secret isogenies, then the endomorphism ring is unknown to them all by Theorem 2.5.6.

Of course, the above is not sufficient against dishonest adversaries, as nothing stops $\mathcal{P}_n$ from choosing the curve of their choice as E_n and lying about their isogeny. To solve

this issue, [6] proposed having each party prove their knowledge of their claimed isogeny by publishing a Fiat–Shamir signature of a zero-knowledge proof.

Using a signature as proof of knowledge has the advantage of keeping the number of required interactions to a minimum. The one proposed in [6], in particular, can be used to efficiently generate curves over $\mathbb{F}_{p^2}$. However, the zero-knowledge proof proposed in [6] reveals the degree of the secret isogeny. While it does not create issues when working over $\mathbb{F}_{p^2}$, this leaks too much information when dealing with isogenies defined over $\mathbb{F}_p$. This is because isogenies defined over $\mathbb{F}_p$ are usually sampled using CSIDHSAMPLE. Therefore, the isogeny degree can be used to efficiently compute $|e_i|$, and this knowledge massively reduces the possible key space. Because of this, a different signature scheme would need to be used to prove the knowledge of the claimed isogenies.

Basso, Codogni, Connolly, De Feo, Fouotsa, Lido, Morrison, Panny, Patranabis and Wesolowski mention the possible use of either SeaSign [27] or CSI-FiSh [11] as possible replacement signatures. While both work in theory, they each come with issues limiting their practical applications. While SeaSign is a zero-knowledge signature, its current computation times are way too long to be used for everyday applications. However, it is worth noting that, in cases where a single secure curve needs to be generated by parties that can afford to wait multiple hours, for example, when generating secure parameters for a scheme, using SeaSign is a possible solution.

On the other hand, CSI-FiSh can potentially be both zero-knowledge and efficient. It is worth noting that such a construction would lead to a protocol similar to CSI-RAShi [10]. However, CSI-FiSh requires full knowledge of the ideal class group associated with the chosen parameter set. Currently, the parameter sets for which the ideal class group is known are pretty limited and, as discussed by Panny [48], computing new ones would require an extensive amount of computation even with access to a quantum computer. While recent results presented in the SCALLOP paper [31] have expanded the number of parameter sets that can be used today, the complexity of finding new ones is still super-polynomial.

Clapoti [47], a recent algorithm developed by Page and Robert, can compute the isogenies associated with any ideal class in polynomial time without requiring a specific form for the input. However, it is still unclear if the current version of the algorithm is fast enough for most practical uses.

4.1.2 Multiparty Key Generation

Two other possible ideas to generate secure curves defined over $\mathbb{F}_p$ were proposed by Moriya, Takashima and Tagaki [45]. However, the adversarial model in that paper is honest but curious, and this creates issues when trying to adapt their techniques to active adversaries.

The core idea of the protocol, given n parties $\mathcal{P}_1, \dots, \mathcal{P}_n$, is to have n chains of isogenies that all loop over the same set of ideal class group elements so that, for each party, there is a chain where they are the last to apply their group action. This then implies that each party can trust the security of one chain and that, since the commutativity of the ideal class group implies that all chains end at the same curve, they can all trust the security of the final curve.

While the security of the above scheme is not proven against active adversaries in [45], such a proof might be possible. However, another issue is that the number of required interactions grows quadratically in proportion to the number of parties, making the scheme inefficient when working with a large number of parties.

4.2 Assumptions

Since the goal of this chapter is to generate elliptic curves of unknown endomorphism ring, we have to assume that computing the endomorphism ring of a given elliptic curve is hard. Because of Theorem 2.5.6, we will often use the problems of computing an isogeny and the problem of computing an endomorphism ring interchangeably and therefore also assume that computing an isogeny between two curves is hard. To be more precise, the isogeny problem on which we base our protocol requires a stronger assumption.

Assumption 4.2.1 (CCSIDH[15]). *Let E be a supersingular elliptic curve defined over $\mathbb{F}_p$ with an unknown endomorphism ring and let $\mathcal{C}$ be the ideal class group of $\text{End}(E)$. Given $\mathfrak{a} \star E$ and $\mathfrak{b} \star E$ with unknown $\mathfrak{a}, \mathfrak{b} \in \mathcal{C}$, the Computational CSIDH problem (CCSIDH) is to compute $\mathfrak{a} \star \mathfrak{b} \star E$. We assume that this problem is hard, as defined in Definition 2.5.3.*

The above assumption is required for (and equivalent to) the one-way security of CSIDH, and is therefore already widely accepted for many protocols that work with isogenies over $\mathbb{F}_p$. We also note that Assumption 4.2.1 implies that computing isogenies between two curves over $\mathbb{F}_p$ is hard.

Our second assumption is an adaptation of the Knowledge-of-Exponent Assumption (KEA) that was first described by Damgård [23]. This assumption was then used to create

a classical Diffie–Hellman protocol by Wu and Stinson [65]. We denote this new assumption as the CSIDH Knowledge of Exponent assumption, or CSIDHKeE for short.

Assumption 4.2.2 (CSIDHKeE). *Let E be a supersingular elliptic curve defined over $\mathbb{F}_p$. For any probabilistic polynomial-time (PPT) $\mathbf{A}$ that takes as input E and $\mathbf{b} \star E$, where $\mathbf{b}$ is sampled using $\text{CSIDHSAMPLE}(\mathcal{S}, B)$, and which produces as output a pair of supersingular elliptic curves over $\mathbb{F}_p$ (F, F') , there exists probabilistic polynomial-time extractor $\mathbf{E}$ which takes the same input and outputs the same pair (F, F') , along with $\mathbf{a}$, such that the probability of $((F' = \mathbf{b} \star F) \wedge (\mathbf{a} \star E \neq F))$ being true is negligible, in terms of $\log(p)$.*

In other words, we assume that there is no pair of supersingular elliptic curves over $\mathbb{F}_p$ such that computing an isogeny between them is hard, but computing a random CSIDH exchange involving them is easy. The original KEA is used in the context of classical Diffie–Hellman, while this new version is applied to CSIDH. There is currently no known way to attack such an assumption.

Our third assumption simply states that we have access to a function H with some strong security properties. This same assumption is used by Basso, Codogni, Connolly, De Feo, Fouotsa, Lido, Morrison, Panny, Patranabis and Wesolowski [6] for generating a multiparty protocol for secure curves over $\mathbb{F}_{p^2}$.

Assumption 4.2.3 (Secure Commitment Scheme Assumption). *A commitment scheme is said to be statistically hiding if its output is indistinguishable from random. A commitment scheme is said to be computationally binding if it is infeasible to find a valid input for any commitment that was not generated honestly and it is infeasible to find two inputs leading to the same commitment.*

As in [6], we assume the existence of a function $H : \{0, 1\}^* \rightarrow \mathcal{H}$ which is a statistically hiding and computationally binding commitment scheme on the set of binary strings.

In the same way as W is treated in Assumption 2.9.4, we assume that H has a large codomain $\mathcal{H}$ and we implicitly assume that this data is encoded in the form of a binary string using a suitable encoding scheme in cases where we use H on arbitrary data.

The above three assumptions are enough to obtain an efficient protocol generating supersingular elliptic curves of unknown endomorphism rings. However, we can obtain a more efficient protocol using a random oracle function and Assumption 2.9.4.

The above assumptions are required for most of our protocols. These assumptions are enough for all our protocols except CSIDH-CR. Hence, if we only need to generate

supersingular elliptic curves with unknown endomorphism rings over $\mathbb{F}_p$, we do not need to invoke Assumption 4.2.4 below. On the other hand, for our results on random curve generation (Section 4.5), we require the following assumption, which was considered in the SeaSign paper [27, p.766].

Assumption 4.2.4. *The output distribution of $\text{CSIDHSAMPLE}(\mathcal{S}, B)$ is indistinguishable from the uniform distribution on the ideal class group $\mathcal{C}$ of $\text{End}(E)$.*

4.3 A New Zero-Knowledge Proof

As mentioned in the previous section, there is currently no known fast CSIDH-based signature scheme that works with any parameter set without heavy precomputation or the use of a quantum computer. To get around this issue, we propose replacing the signature part of the multiparty protocol from Basso, Codogni, Connolly, De Feo, Fouotsa, Lido, Morrison, Panny, Patranabis and Wesolowski [6] with an interactive zero-knowledge proof. The following proposal, CSIDH Random Oracle Interactive Proof (CSIDH-ROIP), requires Assumptions 4.2.1, 2.9.4 and 4.2.2 for its security proof. Note that, in contrast to Moriya, Takashima and Takagi [45], we allow for malicious parties.

Definition 4.3.1 (CSIDH-ROIP). *Let a prover P know a secret $\mathbf{a} \in \mathcal{C}$ with associated public data $(E, E_{\mathcal{P}} := \mathbf{a} \star E)$. The goal of the following protocol is for $\mathcal{P}$ to prove their knowledge of $\mathbf{a}$ to a verifier $\mathcal{V}$ without leaking any extra information. CSIDH-ROIP consists of the following steps:*

1. *Challenge:* $\mathcal{V}$ sends the challenge curve $E_{\mathcal{V}}$.
2. *Response:* $\mathcal{P}$ computes $\mathbf{a} \star E_{\mathcal{V}}$ and publishes a masked version of it using a random oracle function.
3. *Verification:* $\mathcal{V}$ verifies $\mathcal{P}$'s answer.

CHALLENGE($\mathcal{P}, \mathcal{V}, E, E_{\mathcal{P}}$)	RESPONSE($\mathcal{P}, \mathcal{V}, E, E_{\mathcal{P}}, \mathbf{a}, E_{\mathcal{V}}$)	VERIFICATION($\mathcal{P}, \mathcal{V}, E, E_{\mathcal{P}}, E_{\mathcal{V}}, \mathbf{b}, M_{\mathcal{P}, \mathcal{V}}$)
$\mathbf{b} \leftarrow \text{CSIDHSAMPLE}(\mathcal{S}, B)$	$E_{\mathcal{P}, \mathcal{V}} \leftarrow \mathbf{a} \star E_{\mathcal{V}}$	$M'_{\mathcal{P}, \mathcal{V}} \leftarrow W(\mathcal{P}, \text{JINV}(\mathbf{b} \star E_{\mathcal{P}}))$
$E_{\mathcal{V}} \leftarrow \mathbf{b} \star E$	$M_{\mathcal{P}, \mathcal{V}} \leftarrow W(\mathcal{P}, \text{JINV}(E_{\mathcal{P}, \mathcal{V}}))$	if $M'_{\mathcal{P}, \mathcal{V}} = M_{\mathcal{P}, \mathcal{V}}$: return true
return $E_{\mathcal{V}}$	return $M_{\mathcal{P}, \mathcal{V}}$	else : return false

The correctness of CSIDH-ROIP comes from the fact that $\mathbf{a} \star (\mathbf{b} \star E) = \mathbf{b} \star (\mathbf{a} \star E)$. Its soundness and zero-knowledge properties come from the following theorems. The proof of zero-knowledge only requires Assumption 2.9.4. The other assumptions are used in the soundness proof.

Theorem 4.3.2. *Given Assumption 2.9.4, CSIDH-ROIP is zero-knowledge.*

Proof. Recall that W is a random oracle function defined in Assumption 2.9.4 whose input is arbitrary data. We can separate the adversarial strategies as the verifier in two cases. Either they send a challenge E_V for which they know the associated $\mathbf{b}$, or they choose a curve for whose associated group element is unknown and not feasibly computable.

In the former case, for any chosen $\mathbf{b}$, the associated simulator is simple as an honest verifier knows the correct answer without needing to interact with the prover. In the latter case, we use the fact that W is a random oracle function to simulate transcripts indistinguishable from honest ones. This time, the adversary can use any method they desire to sample E_V , but they do not gain any usable information from the answer as it is masked by W .

As we have two cases, we present two simulators. SIMULATOR1 represents the first case while SIMULATOR2 represents the second. The simulator returns a challenge-response pair with the same distribution as that of an honest exchange.

SIMULATOR1($\mathcal{P}, E, E_{\mathcal{P}}, \mathbf{b}$)	SIMULATOR2($\mathcal{P}, E, E_{\mathcal{P}}, E_V$)
$E_V \leftarrow \mathbf{b} \star E$	Challenge $\leftarrow E_V$
Challenge $\leftarrow E_V$	$M_{\mathcal{P}, V} \leftarrow \mathcal{W}$
$M_{\mathcal{P}, V} \leftarrow W(\mathcal{P}, \text{JINV}(\mathbf{b} \star E_{\mathcal{P}}))$	Response $\leftarrow M_{\mathcal{P}, V}$
Response $\leftarrow M_{\mathcal{P}, V}$	return (Challenge, Response)
return (Challenge, Response)	

□

Theorem 4.3.3. *CSIDH-ROIP is computationally sound under Assumptions 4.2.1, 2.9.4 and 4.2.2.*

Proof. Let $\mathcal{A}$ be a probabilistic polynomial-time (PPT) algorithm able to generate valid responses $M_{\mathcal{P}, V}$ for random challenges E_V . Since W is a random oracle function, computing a valid $M_{\mathcal{P}, V}$ is equivalent to computing the correct $E_{\mathcal{P}, V}$ associated with the given challenge. This implies the existence of a PPT algorithm $\mathcal{A}'$ capable of generating $E_{\mathcal{P}, V}$ when given E_V . By Assumption 4.2.2, this then implies the existence of a PPT extractor capable of computing a valid witness $\mathbf{a}$. □

4.3.1 Avoiding the Random Oracle Model

As mentioned in Section 4.2, it is possible to generate supersingular elliptic curves over $\mathbb{F}_p$ with unknown endomorphism ring without having to rely on a random oracle function. The core idea of this trick is to remark that, without the use of W , CSIDH-ROIP is still honest verifier zero-knowledge. Therefore, we only need a way to deal with dishonest verifiers.

CSIDH-ROIP does so by publishing information unusable by dishonest verifiers, but this is not the only way to proceed. Another way is to have the verifiers prove that they were honest at the end of the proof. This can be done by requiring verifiers to publish a masked commitment for their challenge using a statistically hiding and computationally binding protocol H .

Of course, an adversary might still lie about their commitment and not care about being found cheating, as they still obtain information from the prover's answer. However, when it comes to using our proof scheme to generate secure curves, once the honest parties detect that someone cheated, the guilty party can then simply be removed from the protocol, and the honest parties can then restart the protocol using new random values. As long as the new values are independent of the previous ones, the adversary gains no information by cheating.

Definition 4.3.4 (CSIDH-AIP). *Let a prover $\mathcal{P}$ know a secret $\mathbf{a} \in \mathcal{C}$ with associated public data $(E, E_{\mathcal{P}} := \mathbf{a} \star E)$. The goal of the following protocol is for $\mathcal{P}$ to prove their knowledge of $\mathbf{a}$ to a verifier $\mathcal{V}$ without leaking any extra information. CSIDH-AIP consists of the following steps:*

1. *Challenge: $\mathcal{V}$ sends the challenge curve $E_{\mathcal{V}}$ and commits the associated $\mathbf{b}$ using H . A random value r , used for the commitment, is computed in this step and is kept in memory.*
2. *Response: $\mathcal{P}$ computes $\mathbf{a} \star E_{\mathcal{V}}$ and publishes it.*
3. *Verification1: $\mathcal{V}$ verifies $\mathcal{P}$'s answer and, if it is valid, publishes $(\mathbf{b}, r)$.*
4. *Verification2: $\mathcal{P}$ verifies that $\mathcal{V}$'s commitment coincides with the challenge.*

The details of the protocol are as follows:

CHALLENGE($\mathcal{V}, E$)	RESPONSE($\mathcal{P}, \mathbf{a}, E_{\mathcal{V}}$)
$\mathbf{b} \leftarrow \text{CSIDHSAMPLE}(\mathcal{S}, B)$	$E_{\mathcal{P}, \mathcal{V}} \leftarrow \mathbf{a} \star E_{\mathcal{V}}$
$E_{\mathcal{V}} \leftarrow \mathbf{b} \star E$	return $E_{\mathcal{P}, \mathcal{V}}$
$r \leftarrow \$_N$	
$C \leftarrow H(\mathbf{b}, r)$	
return $(E_{\mathcal{V}}, C)$	
VERIFICATION1($\mathcal{V}, E_{\mathcal{P}}, \mathbf{b}, r, E_{\mathcal{P}, \mathcal{V}}$)	VERIFICATION2($\mathcal{P}, E, E_{\mathcal{V}}, \mathbf{b}, r, C$)
$E'_{\mathcal{P}, \mathcal{V}} \leftarrow \mathbf{b} \star E_{\mathcal{P}}$	if $C \neq H(\mathbf{b}, r)$: return false
if $\text{JINV}(E'_{\mathcal{P}, \mathcal{V}}) = \text{JINV}(E_{\mathcal{P}, \mathcal{V}})$: return $(\mathbf{b}, r)$	$E'_{\mathcal{V}} \leftarrow \mathbf{b} \star E$
else : return false	if $E'_{\mathcal{V}} \neq E_{\mathcal{V}}$: return false
	return true

The proof fails if either Verification1 or Verification2 returns **false** and succeeds otherwise.

The correctness of CSIDH-AIP holds for the same reason as CSIDH-ROIP's. It is also sound and zero-knowledge, as the following theorems show.

Theorem 4.3.5. *Given Assumption 4.2.3, if it does not abort, CSIDH-AIP is zero-knowledge.*

Proof. Since H is computationally binding, $\mathcal{V}$ must compute $E_{\mathcal{V}}$ by first choosing a valid $\mathbf{b}$. By the same reasoning, they must also choose their nonce r in advance. The following simulator returns a Challenge-Response-Verification1 triple for any challenge constructed using $(\mathbf{b}, r)$.

SIMULATOR($E, E_{\mathcal{P}}, \mathbf{b}, r$)
$E_{\mathcal{V}} \leftarrow \mathbf{b} \star E$
$C \leftarrow H(\mathbf{b}, r)$
Challenge $\leftarrow (E_{\mathcal{V}}, C)$
$E_{\mathcal{P}, \mathcal{V}} \leftarrow \mathbf{b} \star E_{\mathcal{P}}$
Response $\leftarrow E_{\mathcal{P}, \mathcal{V}}$
Verification1 $\leftarrow (\mathbf{b}, r)$
return (Challenge, Response, verification1)

□

Theorem 4.3.6. *CSIDH-AIP is computationally sound under Assumptions 4.2.1, 4.2.2 and 4.2.3.*

Proof. Since H is statistically hiding, from $\mathcal{P}$'s point of view, C can be replaced with a random value and, therefore, gives no advantage when it comes to beating the soundness property. Without being able to make use of this extra information, successfully generating a valid CSIDH-AIP proof becomes equivalent to generating $\mathbf{a} \star \mathbf{b} \star E$ when given $\mathbf{b} \star E$ and $\mathbf{a} \star E$.

By Assumption 4.2.2, if there were a PPT algorithm capable of doing so, then there would be another PPT algorithm capable of solving the CCISDH problem, which we assume is hard. $\square$

4.4 Secure Curve Generation

With the help of CSIDH-ROIP, we can now present our multiparty protocol for generating supersingular elliptic curves defined over $\mathbb{F}_p$ with unknown endomorphism ring. As its security mostly relies on CSIDH-ROIP, this new protocol, which we call CSIDH Secure Curve Generator (CSIDH-SCG), requires Assumptions 4.2.1 4.2.2 and 2.9.4, but not Assumption 4.2.4.

Definition 4.4.1 (CSIDH-SCG). *Let $\mathcal{P}_1, \dots, \mathcal{P}_n$ be n parties that want to generate a supersingular elliptic curve over $\mathbb{F}_p$ with unknown endomorphism ring. Let E_0 be a known supersingular curve over the same field. CSIDH-SCG consists of the following steps.*

- **CurveGen:** For i from 1 to n , party $\mathcal{P}_i$ computes an ideal class group element $\mathbf{a}_i$, saves it, and publishes $E_i := \mathbf{a}_i \star E_{i-1}$.
- **Challenge:** For $j \in [n] \setminus \{i\}$, $\mathcal{P}_i$ sends a CSIDH-ROIP challenge $E_{i,j}$ to $\mathcal{P}_j$ and saves the associated $\mathbf{b}_{i,j}$.
- **Response:** After receiving a challenge from every $\mathcal{P}_j$ such that $j \in [n] \setminus \{i\}$, $\mathcal{P}_i$ publishes a CSIDH-ROIP response $M'_{i,j}$ for each of them.
- **Verification:** After receiving responses to all their challenges, $\mathcal{P}_i$ checks them for validity. If all the responses are correct, $\mathcal{P}_i$ publishes **true** and then accepts E_n as the final curve if every other party also publishes **true**. Otherwise, $\mathcal{P}_i$ publishes **false** and aborts the entire protocol.
- **Abort:** If at any point, a party published **false**, the protocol is aborted and every party must publish all their computed values. Any dishonest parties are thereby revealed.

The algorithms for each step are as follows:

CURVEGEN($\mathcal{P}_i, E_{i-1}$)	CHALLENGE($\mathcal{P}_i, \mathcal{P}_j, E_{j-1}$)
$\mathbf{a}_i \leftarrow \text{CSIDHSAMPLE}(\mathcal{S}, B)$	$\mathbf{b}_{i,j} \leftarrow \text{CSIDHSAMPLE}(\mathcal{S}, B)$
$E_i \leftarrow \mathbf{a}_i \star E_{i-1}$	$E_{i,j} \leftarrow \mathbf{b}_{i,j} \star E_{j-1}$
return E_i	return $E_{i,j}$
RESPONSE($\mathcal{P}_i, \mathcal{P}_j, \mathbf{a}_i, E_{j,i}$)	VERIFICATION($\mathcal{P}_i, \mathcal{P}_j, M'_{i,j}, E_j$)
$M'_{j,i} \leftarrow W(\mathcal{P}_i, \mathbf{a}_i \star E_{j,i})$	$M''_{i,j} \leftarrow W(\mathcal{P}_j, \mathbf{b}_{i,j} \star E_j)$
return $M'_{j,i}$	if $M'_{i,j} = M''_{i,j}$: return true
	else : return false

Notation 4.4.2 (Secure Curve Generation Adversary). For a secure curve generation multiparty protocol with n parties, the adversary is denoted $\mathcal{A}_{SCG}$. The goal of $\mathcal{A}_{SCG}$ is to compute the endomorphism ring of the final curve E_n . $\mathcal{A}_{SCG}$ is able to take control of all parties but one, say $\mathcal{P}_i$. They can try to be dishonest during the multiparty protocol. However, they fail if the protocol is aborted.

Theorem 4.4.3. Given Assumptions 4.2.1, 4.2.2, and 2.9.4, CSIDH-SCG is secure against $\mathcal{A}_{SCG}$ adversaries.

Proof. During CSIDH-SCG, every party must prove knowledge of their $\mathbf{a}_i$ using $n - 1$ parallel CSIDH-ROIP proofs. By Theorem 4.3.2, $\mathcal{A}_{SCG}$ gains no information about $\mathbf{a}_i$. By Theorem 4.3.3, $\mathcal{A}_{SCG}$ cannot lie about any of their $\mathbf{a}_j$. Since $\mathcal{A}_{SCG}$ knows every $\mathbf{a}_j$ except for $\mathbf{a}_i$, computing the endomorphism ring of E_n is equivalent to computing the endomorphism ring of E_i . However, by Theorem 2.5.6, doing so is equivalent to computing $\mathbf{a}_i$, which is hard given Assumption 4.2.1. $\square$

Remark 4.4.4. CSIDH-SCG's security does not depend on whether the endomorphism ring of the initial curve is known or not. As long as $\mathcal{P}_i$'s action is added to the chain and the following parties act on $\mathcal{P}_i$'s published curve, the final endomorphism ring will be unknown to all.

In addition to being secure, CSIDH-SCG is also efficient, even when considering large groups. This is true for both the computation time and the number of interactions. In practice, the CSIDH parameters are chosen so that both sampling and group actions are computed efficiently. W is also assumed to be efficiently computable. In CSIDH-SCG, the number of times each party must compute a group action grows linearly in terms of the

number of participants. The same is true for the number of times each party must call the functions W and CSIDHSAMPLE .

When it comes to the number of operations, CSIDH-SCG is constructed in a way that every step of the proof can be done in parallel with every other party. Because of this, each party is only required to publish once for each of the four steps of CSIDH-SCG, making the total number of interactions $4n$, which is linear in terms of the number of participants.

4.4.1 Generating Secure Curves Without a Random Oracle

In cases where we want to avoid needing a random oracle, we can use CSIDH-AIP instead of CSIDH-ROIP.

Definition 4.4.5 (CSIDH-ASCG). *Let $\mathcal{P}_1, \dots, \mathcal{P}_n$ be n parties that want to generate a supersingular elliptic curve over $\mathbb{F}_p$ with unknown endomorphism ring. Let E_0 be a known supersingular curve over the same field. CSIDH-ASCG consists of the following steps.*

- **CurveGen:** For i from 1 to n , party $\mathcal{P}_i$ computes an ideal class group element $\mathbf{a}_i$, saves it, and publishes $E_i := \mathbf{a}_i \star E_{i-1}$.
- **Challenge:** For $j \in [n] \setminus \{i\}$, $\mathcal{P}_i$ sends a CSIDH-AIP challenge $(E_{i,j}, C_{i,j})$ to $\mathcal{P}_j$ and saves the associated $(\mathbf{b}_{i,j}, r_{i,j})$.
- **Response:** After receiving a challenge from every $\mathcal{P}_j$ such that $j \in [n] \setminus \{i\}$, $\mathcal{P}_i$ publishes a CSIDH-AIP response $E'_{i,j}$ for each of them.
- **Verification1:** After receiving responses to all their challenges, $\mathcal{P}_i$ checks them for validity. If all the responses are correct, $\mathcal{P}_i$ publishes all their pairs $(\mathbf{b}_{i,j}, r_{i,j})$. Otherwise, $\mathcal{P}_i$ publishes **false** and aborts the entire protocol.
- **Verification2:** After receiving every pair $(\mathbf{b}_{j,i}, r_{j,i})$, $\mathcal{P}_i$ verifies that each pair agrees with their challenges. If that is the case for every pair, $\mathcal{P}_i$ publishes **true**. Otherwise, $\mathcal{P}_i$ publishes **false** and aborts the entire protocol. If every party publishes **true**, E_n is accepted as the curve of unknown endomorphism ring.
- **Abort:** If at any point, a party published **false**, the protocol is aborted and every party must publish all their computed values. Any dishonest parties are thereby revealed.

The algorithms for each step are as follows:

$\text{CURVEGEN}(\mathcal{P}_i, E_{i-1})$	$\text{CHALLENGE}(\mathcal{P}_i, \mathcal{P}_j, E_{j-1})$	$\text{RESPONSE}(\mathcal{P}_i, \mathcal{P}_j, \mathbf{a}_i, E_{j,i})$
$\mathbf{a}_i \leftarrow \text{CSIDHSAMPLE}(\mathcal{S}, B)$ $E_i \leftarrow \mathbf{a}_i \star E_{i-1}$ return E_i	$\mathbf{b}_{i,j} \leftarrow \text{CSIDHSAMPLE}(\mathcal{S}, B)$ $E_{i,j} \leftarrow \mathbf{b}_{i,j} \star E_{j-1}$ $r_{i,j} \leftarrow \$ N$ $C_{i,j} \leftarrow H(\mathbf{b}_{i,j}, r_{i,j})$ return $(E_{i,j}, C_{i,j})$	$E'_{j,i} \leftarrow \mathbf{a}_i \star E_{j,i}$ return $E'_{j,i}$
$\text{VERIFICATION1}(\mathcal{P}_i, \mathcal{P}_j, E'_{i,j}, E_j, \mathbf{b}_{i,j}, r_{i,j})$	$\text{VERIFICATION2}(\mathcal{P}_i, \mathcal{P}_j, C_{j,i}, \mathbf{b}_{j,i}, r_{j,i}, E_{i-1}, E_{j,i})$	
$E''_{i,j} \leftarrow \mathbf{b}_{i,j} \star E_j$ if $E''_{i,j} = E'_{i,j}$: return $(\mathbf{b}_{i,j}, r_{i,j})$ else : return false	if $C_{j,i} \neq H(\mathbf{b}_{j,i}, r_{j,i})$: return false $E'''_{j,i} \leftarrow \mathbf{b}_{j,i} \star E_{i-1}$ if $E'''_{j,i} \neq E_{j,i}$: return false return true	

Theorem 4.4.6. *Given Assumptions 4.2.1, 4.2.3, and 4.2.2, CSIDH-ASCG is secure against $\mathcal{A}_{SCG}$ adversaries.*

Proof. During CSIDH-ASCG, every party must prove knowledge of their $\mathbf{a}_i$ using $n - 1$ parallel CSIDH-AIP proofs. By Assumption 4.2.3, since H is statistically hiding, $\mathcal{A}_{SCG}$ gains no information from $\mathcal{P}_i$ publishing $C_{i,j}$. This implies that $\mathcal{A}_{SCG}$ must prove knowledge of its $\mathbf{a}_j$ using CSIDH-AIP without any extra information. By Theorem 4.3.5, since the adversary loses if they cause an abort, $\mathcal{A}_{SCG}$ gains no information about $\mathbf{a}_i$. By Theorem 4.3.6, $\mathcal{A}_{SCG}$ cannot lie about any of their $\mathbf{a}_j$.

Since $\mathcal{A}_{SCG}$ knows every $\mathbf{a}_j$ except for $\mathbf{a}_i$, computing the endomorphism ring of E_n is equivalent to computing the endomorphism ring of E_i . However, by Theorem 2.5.6, doing so is equivalent to computing $\mathbf{a}_i$, which is hard given Assumption 4.2.1. $\square$

While efficient and secure, CSIDH-ASCG comes with two disadvantages compared to CSIDH-SCG. First, since there are two verification steps in CSIDH-AIP, CSIDH-ASCG requires one more round of interaction than CSIDH-SCG. The other issue is that CSIDH-AIP is only zero-knowledge if there are no aborts. Because of this, every time a dishonest party is found, the entire process needs to be restarted with new random values. In practice, needing a random oracle function assumption is usually worth it for the efficiency gains of CSIDH-SCG.

4.5 Curve Randomizer

CSIDH-SCG allows us to generate a supersingular elliptic curve of unknown endomorphism ring. However, E_n is not uniformly random, as $\mathcal{P}_n$ has some control over what the final curve is. This limitation also appears in the protocol proposed by Basso, Codogni, Connolly, De Feo, Fouotsa, Lido, Morrison, Panny, Patranabis and Wesolowski [6]. However, some protocols (for example, Delay Encryption [13]) explicitly ask for a uniformly random (or at least nearly uniform) curve over $\mathbb{F}_p$. In order to generate curves that are compatible with the security proofs of such schemes, we require a way to convert a secure elliptic curve into a random and secure elliptic curve, which justifies the need for the following protocol.

The core idea of CSIDH Curve Randomizer (CSIDH-CR) is to use a multiparty commitment scheme to generate some random data and then convert that data into a random isogeny whose codomain is chosen as the random curve. CSIDH-CR is the only protocol in this thesis that requires Assumption 4.2.4. On the other hand, it does not require Assumption 4.2.1.

Definition 4.5.1 (CSIDH-CR). *Let $\mathcal{P}_1, \dots, \mathcal{P}_n$ be n parties that want to generate a random supersingular elliptic curve over $\mathbb{F}_p$. Let E_n be a supersingular elliptic curve of unknown endomorphism ring defined over the same field. CSIDH-SCG consists of the following steps.*

- **RandomSample:** Each $\mathcal{P}_i$ samples a random $\mathbf{a}'_i$ and nonce r'_i and commits $H(\mathbf{a}'_i, r'_i)$.
- **CurveComp:** Once every commitment has been published, each party publishes their pair $(\mathbf{a}'_i, r'_i)$. If every pair corresponds to their commitment, the final curve is chosen to be $(\prod_{i=1}^n \mathbf{a}'_i) \star E_n$.
- **Abort:** If any of the pairs do not correspond to their commitments, the protocol aborts, and any dishonest parties are thereby revealed.

The algorithms for each step are as follows:

RANDOMSAMPLE($\mathcal{P}_i$)	CURVECOMP($E_n, (C'_1, \dots, C'_n), ((\mathbf{a}'_1, r'_1), \dots, (\mathbf{a}'_1, r'_1))$)
$\mathbf{a}'_i \leftarrow \text{CSIDHSAMPLE}(\mathcal{S}, B)$ $r'_i \leftarrow \$N$ $C'_i \leftarrow H(\mathbf{a}'_i, r'_i)$ return C'_i	$F \leftarrow E_n$ for $j \in [n]$: if $C'_j \neq H(\mathbf{a}'_j, r'_j)$: return false $F \leftarrow \mathbf{a}'_j \star F$ return F

Notation 4.5.2 (Curve Randomizer Adversary). *Given a curve randomizer multiparty protocol with n parties and whose initial curve E_n has an unknown endomorphism ring, the adversary is denoted $\mathcal{A}_{CR}$. The goal of $\mathcal{A}_{CR}$ is either to guess the final curve F before starting the scheme or to compute its endomorphism ring. $\mathcal{A}_{CR}$ is able to take control of all parties but one, say $\mathcal{P}_i$. They can try to be dishonest during the multiparty protocol. However, they fail if the protocol is aborted.*

Theorem 4.5.3. *Given Assumptions 4.2.3 and 4.2.4, CSIDH-CR is secure against $\mathcal{A}_{CR}$ adversaries.*

Proof. By Assumption 4.2.3, H is statistically hiding and $\mathcal{A}_{CR}$ gains no information from C'_i . By the same assumption, since H is a binding commitment scheme, $\mathcal{A}_{CR}$ must choose their α'_j before knowing anything about α'_i . Once the α'_j have been chosen, set $F' := \left(\prod_{j \in [n] \setminus \{i\}} \alpha'_j\right) \star E_n$.

By Assumption 4.2.4, α'_i is indistinguishable from a uniformly random ideal class group element. Since $\mathcal{C}$ is an abelian group, we have that $F = \alpha'_i \star F'$. Therefore, α'_i is indistinguishable from random, and so is F . Also, since every α'_i is revealed at the end of the protocol, computing the endomorphism ring of F is equivalent to computing the endomorphism ring of E_n , which is hard. $\square$

4.6 Conclusion

CSIDH-SCG enables efficient generation of supersingular elliptic curves defined over $\mathbb{F}_p$ with unknown endomorphism ring. In analogy to the work of Basso, Codogni, Connolly, De Feo, Fouotsa, Lido, Morrison, Panny, Patranabis and Wesolowski for curves over $\mathbb{F}_{p^2}$ [6], the total number of interactions required for CSIDH-SCG grows linearly in terms of the number of participants. Currently, the greatest limitation of CSIDH-SCG is the need for a Knowledge of Exponent Assumption. While such an assumption is sometimes used in classical schemes, this new CSIDH variant requires further study on its security.

The other limitation of CSIDH-SCG is its need for a random oracle function. However, this can be dealt with by using CSIDH-ASCG. While this alternative is strictly less efficient, the increase in computation time is only a single additional round of interactions. In cases where a new secure curve must be generated often or some parties are slow to interact, one might prefer to use CSIDH-SCG. The decision on which of the two schemes to use depends on details of the application and one's comfort in using a random oracle assumption.

Given an additional assumption on the randomness of CSIDH samples, CSIDH-CR makes it possible for the generated curve to be random.

It is worth mentioning that the curve randomizer structure can also be adapted to generate random supersingular elliptic curves defined over $\mathbb{F}_{p^2}$. Using the Ramanujan property of supersingular isogeny graphs defined over $\mathbb{F}_{p^2}$, Jao, Miller and Venkatesan [41] showed that the codomain of a random isogeny of large enough degree is indistinguishable from random. It is therefore possible to use a multiparty protocol to generate random data, which can then be converted into a random isogeny in order to obtain a random supersingular elliptic curve. We leave the implementation of such a protocol for future work.

By itself, CSIDH-ROIP is a secure and efficient zero-knowledge proof that can be used with any CSIDH parameter set. While its structure makes it so that it cannot be used in a signature scheme, its large challenge space makes it so that a single run of CSIDH-ROIP is enough to achieve levels of security comparable to CSIDH, given the CSIDH KoE assumption.

Chapter 5

Signatures

In Chapter 4, we have seen that we can generate supersingular elliptic curves of unknown endomorphism rings by generating random curves. In Chapter 3, we saw that, when it comes to key recovery attacks against SIDH variants with well-chosen level structures and parameter sizes, using curves whose endomorphism ring is unknown can only increase the security of the associated protocol. However, passive attacks are not the only known attacks on isogeny-based schemes.

Even before SIDH was broken, there was a known adaptive attack against it [35]. The attack consists of publishing a torsion mapping that is slightly wrong. An attacker can then check if the exchange still works and, by doing so, gains one bit of information on the victim's secret. This can be repeated to obtain the entire secret key one bit at a time.

In order to defend against this attack, SIKE uses the Hofheinz–Hövelmanns–Kiltz variant of the Fujisaki–Okamoto transform as well as ephemeral keys in order to convert SIDH into a key encapsulation mechanism that is resistant to these adaptive attacks. However, having to use ephemeral keys and random oracles limits the applications of the scheme. For example, not every machine can produce the random values necessary for the Fujisaki–Okamoto transform.

Another way to protect SIDH can be found in the original SIDH paper [40]. The idea is to use a zero-knowledge proof to force every party to use valid public keys, making the adaptive attack impossible. If the implementation allows for random oracles, the zero-knowledge proof can be converted to a signature using a Fiat–Shamir transform in order to make the scheme not require interaction. While the original zero-knowledge proof was later shown to not be sound in some degenerate cases [25], the same paper [25] proposes a zero-knowledge proof that is 3-sound and relies on the Double DSSP assumption, which is

a variant of [DSSP](#) that generates two SIDH square intersecting at the secret isogeny ϕ and publish either the two isogenies parallel to ϕ or two random isogenies of the same degree.

When Fujisaki–Okamoto transforms cannot be used, we need zero-knowledge proofs for SIDH variants as they are vulnerable to the same adaptive attacks otherwise [\[33\]](#). Also, while the level structure involved in DSSP can be seen as transmitting the mapping of a single torsion point, the one in Double DSSP can be seen as transmitting two such mappings. Because of that, Double DSSP is broken using the same attacks as those on SIDH. Because of this, the level structure associated with the zero-knowledge proof needs to be as close to trivial as possible in order for it to be secure. In this chapter, we rely on the DSSP assumption instead of the double DSSP assumption, meaning that the level structure we are working with is of the Borel type.

In cases where the prover has access to the endomorphism ring of the domain curve, they can prove their knowledge of an isogeny between the claimed curves using SQISign [\[28\]](#) or SQISignHD [\[24\]](#). Since the isogeny revealed in the associated zero-knowledge proof is independent of the secret isogeny, including its degree, it stays zero-knowledge even when the degree of the secret isogeny is part of the secret. However, in the context of using it to prove honest public keys for SIDH variants, this technique cannot be used to prove that the claimed torsion point information is correct. Also, there are protocols where the endomorphism ring has to be kept unknown from all participants, in which case SQISign cannot be used.

There is already a zero-knowledge proof proposed for M-SIDH [\[5\]](#). However, this proof relies on the security of a decisional variant of M-SIDH. Since the level structure group for M -SIDH is a subgroup of the one used in the Borel level structure, this assumption is stronger than DSSP, which is a decisional variant of [\$\Gamma\$ -SIDH](#) with Borel level structure. In this chapter, we present zero-knowledge proofs able to show the same properties while only needing DSSP at the cost of a slight loss in efficiency.

5.1 Masking the Degree

Suppose that we want to prove knowledge of an isogeny between two supersingular elliptic curves without revealing its degree. When possible, the most efficient solution would be to use SQISign [\[28\]](#) or SQISignHD [\[24\]](#). However, these protocols both require the prover to know the endomorphism ring of the starting curve, which limits the possible applications.

In protocols where the degree of an isogeny is part of the secret, a multiple of said degree is usually publicly known. That is the case for both MD-SIDH [\[33\]](#) and terSIDH

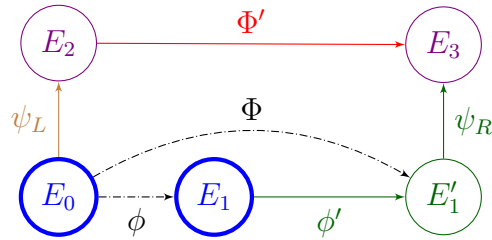
[8]. In such cases, we can use rejection sampling, first introduced in the context of isogenies in SeaSign [27], in order to mask the degree during a normal SIDH proof.

The core idea of **MDISOZKP**, which is presented later in this section, when trying to prove knowledge of an isogeny $\phi : E_0 \rightarrow E_1$, is to start by computing an isogeny $\phi' : E_1 \rightarrow E'_1$ of random and potentially large degree with the same prime factors as the degree of ϕ .

We can then remove the backtracking appearing in $\phi'\phi$ in order to obtain the cyclic isogeny $\Phi : E_0 \rightarrow E'_1$. The step is necessary to make sure that the adversary does not learn anything about the secret isogeny since the degree of backtracking is a non-trivial factor of the degree of ϕ .

We can then compute an isogeny $\psi_L : E_0 \rightarrow E_2$ of degree relatively prime to Φ and "compute the SIDH square" between the two to obtain the isogenies $\Phi' : E_2 \rightarrow E_3$ and $\psi_R : E'_1 \rightarrow E_3$. For the commitment, the prover can publish a hash of E_2 and E_3 and, depending on the challenge, the prover either reveals ψ_L , Φ' or $\psi_R\phi'$.

Remark 5.1.1. *Given two isogenies ϕ_1 and ϕ_2 with the same domain curve and relatively prime smooth degrees. The SIDH square of ϕ_1 and ϕ_2 is the tuple $(\phi_1, \phi_2, [\phi_1]_*\phi_2, [\phi_2]_*\phi_1)$. The name comes from the fact that the two pushforward isogenies have the same codomain up to isomorphism and this property is key for the correctness of SIDH. While SIDH is not secure, its revealing of the torsion mapping is only required because the two initial isogenies are known to distinct parties. When a single party knows both ϕ_1 and ϕ_2 , they can compute both pushforward isogenies easily without leaking any extra information to a potential attacker. It is also worth mentioning that, since the isogeny degrees are smooth and we know both isogenies, it is easy to construct $[\phi_1]_*\phi_2$ and $[\phi_2]_*\phi_1$ so that their mappings on the relevant torsion points coincide.*



Definition 5.1.2 (**Masked Degree ISOgeny Zero-Knowledge Proof (MDISOZKP)**). *Let $\mathcal{A} = \prod_{i=1}^s q_i^{f_i}$ be a large integer such that the q_i s are distinct primes dividing $p + 1$. Let B be a large positive integer relatively prime to A . Let $\phi : E_0 \rightarrow E_1$ be a secret cyclic isogeny of degree $A \mid \mathcal{A}$.*

n is a positive integer representing the number of times the following proof will be repeated. The challenge is a random $\text{chall} \in \{-1, 0, 1\}$.

COMMITMENT	RESPONSE
$A' \leftarrow \$ \text{FACSET}(\mathcal{A}^{sn})$	if $\text{chall} = -1$:
$\phi' \leftarrow \text{CYCLICISOGENY}(E_1, A')$	return (E_2, r_2, ψ_L)
$E'_1 \leftarrow \text{CODOMAIN}(\phi')$	if $\text{chall} = 0$:
$\Phi \leftarrow \text{CYCLIPHY}(\phi' \phi)$	return $(E_3, r_3, \psi_R \phi')$
$K_{\psi_L} \leftarrow \$ E_0[[B]]$	if $\text{chall} = 1$:
$\psi_L \leftarrow \text{ISOGENYFROMKERNEL}(K_{\psi_L})$	if $\deg(\Phi) \notin \text{FACSETTWO}(\mathcal{A}, \mathcal{A}^{sn})$: abort
$E_2 \leftarrow \text{CODOMAIN}(\psi_L)$	return $(E_2, E_3, r_2, r_3, \Phi')$
$\Phi' \leftarrow \text{PARALLELISOGENY}(\Phi, \psi_L)$	VERIFICATION
$\psi_R \leftarrow \text{PARALLELISOGENY}(\psi_L, \Phi)$	if $\text{chall} = -1$:
$E_3 \leftarrow \text{CODOMAIN}(\psi_R)$	if $C_2 \neq H(E_2, r_2)$: return $\perp$
$r_2, r_3 \leftarrow \$ N$	if $\neg \text{ISOVALID}(E_0, E_2, \psi_L)$: return $\perp$
$C_2 \leftarrow H(E_2, r_2)$	if $\text{chall} = 0$:
$C_3 \leftarrow H(E_3, r_3)$	if $C_3 \neq H(E_3, r_3)$: return $\perp$
return (C_2, C_3)	if $\neg \text{ISOVALID}(E_1, E_3, \psi_R \phi')$: return $\perp$
	if $\text{chall} = 1$:
	if $(C_2, C_3) \neq (H(E_2, r_2), H(E_3, r_3))$: return $\perp$
	if $\neg \text{ISOVALID}(E_2, E_3, \Phi')$: return $\perp$
	return true

Remark 5.1.3. In order to mask the degree of the secret isogeny, we use rejection sampling. In cases where the challenge asks for Φ' to be revealed, the prover needs to check that the degree is a multiple of $\mathcal{A}$ and a factor of $\mathcal{A}^{sn}$. Otherwise, the proof is aborted. Later in this section, we prove that the probability of requiring an abort is low enough for a Fiat–Shamir signature to be feasible. We abort in the response phase, rather than the commitment phase, because doing so lowers the probability of an abort happening.

Theorem 5.1.4. Given Assumption 4.2.3, MDISOZKP is 3-special sound for the knowledge of an isogeny from E_0 to E_1 .

Proof. Since H is computationally binding, the commitments are equivalent to obtaining E_2 and E_3 directly when it comes to soundness. Given valid $\psi_L, \psi_R \phi'$ and Φ' for the same commitment (E_2, E_3) , $\psi_R \phi' \Phi' \psi_L$ is an isogeny from E_0 to E_1 . $\square$

Theorem 5.1.5. If MDISOZKP does not abort when $\text{chall} = 1$, then the degree of Φ' is a uniformly random element of $\text{FACSETTWO}(\mathcal{A}, \mathcal{A}^{sn})$.

Proof. Let $d = \frac{\deg(\phi'\phi)}{\deg(\Phi)}$. For any value of d , $\frac{\deg(\phi)}{d}$ divides $\mathcal{A}$. If we fix the value of d , ϕ' can have any degree dividing $\mathcal{A}^{ns}$. We also have that the degree of ϕ' must divide $\mathcal{A}^{ns}$. Hence, since we are conditioning on the fact that MDISOZKP does not abort and $\text{chall} = 1$, for any possible degree of ϕ and value of d , there is a unique degree of ϕ' for every target degree of Φ . This makes the degree of Φ uniform, and since Φ and Φ' have the same degree, the same holds for Φ' . $\square$

Theorem 5.1.6. *Given Assumptions 2.5.11 and 4.2.3, if MDISOZKP does not abort, then it is honest verifier zero-knowledge.*

Proof. For the proof, we can create a simulator $\mathcal{S}$ for outputting a commitment-challenge-answer triple indistinguishable from that of honest parties. To do so, we describe the simulator for each possible challenge. When $\text{chall} = -1$, $\mathcal{S}$ can generate $(K_{\psi_L}, \psi_L, E_2, r_2)$ as would an honest party and generate an honest commitment C_2 . Since H is statistically hiding, randomly sampling C_3 from $\mathcal{H}$ is indistinguishable from an honest computation. If $\text{chall} = 0$, $\mathcal{S}$ can generate (A', ϕ', E'_1) as would an honest party. Since there is a bijection between $E_0[[B]]$ and $E'_1[[B]]$ in addition to the fact that K_{ψ_L} is never revealed, directly sampling K_{ψ_R} is indistinguishable from an honest output. Using this value, $\mathcal{S}$ can generate (ψ_R, E_3, r_3, C_3) as normal and randomly sample C_2 . When $\text{chall} = -1$, $\mathcal{S}$ can sample a random $A'' \in \text{FACSETTWO}(\mathcal{A}, \mathcal{A}^{sn})$, a random supersingular elliptic curve E_2 and then a random cyclic isogeny Φ' using $\text{CYCLICISOGENY}(E_2, A'')$. Given Assumption 2.5.11, this construction is indistinguishable from an honest one. $\mathcal{S}$ can then compute the other values as normal. $\square$

Theorem 5.1.7. *Given n rounds of MDISOZKP, the probability that no abort happens is at least $\frac{1}{e}$.*

Proof. For fixed ϕ and any value of $d = \frac{\deg(\phi'\phi)}{\deg(\Phi)}$, there is always at least $\prod_{i=1}^s (ns - 2)f_i$ possible degrees of ϕ' that do not cause an abort. Hence, the probability of the protocol not aborting in a given round can be lower bounded by

$$\frac{2}{3} + \frac{1}{3} \frac{\prod_{i=1}^s (ns - 2)f_i}{\prod_{i=1}^s (ns + 1)f_i} = \frac{2}{3} + \frac{1}{3} \left(\frac{ns - 2}{ns + 1} \right)^s$$

The probability of having no abort in any round can then be lower bounded by

$$P(n, s) := \left(\frac{2}{3} + \frac{1}{3} \left(\frac{ns - 2}{ns + 1} \right)^s \right)^n$$

When $s = 1$, we have that

$$P(n, 1) = \left(\frac{n}{n+1} \right)^n \geq \frac{1}{e}.$$

When $s = 2$, we have that

$$P(n, 2) = \left(\frac{2 + 4n^2}{(1 + 2n)^2} \right)^n \geq \frac{1}{e}.$$

For a variable $s \geq 2$ and some fixed value of $n \geq 1$, we have that

$$\frac{d}{ds}P(n, s) = \frac{3^{-n}n \left(\frac{ns-2}{ns+1} \right)^{s-1} \left(\left(\frac{ns-2}{ns+1} \right)^s + 2 \right)^{n-1} \left((n^2s^2 - ns - 2) \ln \left(\frac{ns-2}{ns+1} \right) + 3ns \right)}{(ns+1)^2}$$

Since $ns \geq 2$, we have that $(n^2s^2 - ns - 2) \ln \left(\frac{ns-2}{ns+1} \right) + 3ns \geq 0$. Since every other factoring term of $\frac{d}{ds}P(n, s)$ is also non-negative, we have that $\frac{d}{ds}P(n, s) \geq 0$.

Therefore, when $s \geq 2$, we have that $P(n, s) \geq P(n, 2) \geq \frac{1}{e}$. By combining all our cases, we obtain the fact that $P(n, s) \geq \frac{1}{e}$ for every positive values of n and s , which completes the proof.

□

5.2 Masked Torsion

The easiest non-trivial variant of Γ -SIDH to create a zero-knowledge proof for is M-SIDH. This is because the masking matrices for its level structure are all diagonal, which means that they commute with any isogeny. M-SIDH [33] has been proposed as a possible fix for the attacks on SIDH [14, 42, 55]. The main difference being that, for a secret isogeny $\phi : E_0 \rightarrow E_1$ of degree A between two publicly known supersingular elliptic curve, M-SIDH also reveals $([\alpha]\phi(P_0), [\alpha]\phi(Q_0))$ for an unknown random α where (P_0, Q_0) is a basis of $E_0[B]$.

This protocol creates the need of being to prove knowledge of an isogeny with the above properties without leaking extra information. Bassa [5] published a 3-sound zero-knowledge proof that does just that. However, that protocol relies on a stronger assumption

than DSSP. A 6-sound variation only relying on DSSP is mentioned in the same paper, but is dismissed for being too inefficient.

For the original SIDH protocol, De Feo, Dobson, Galbraith and Zobernig [25] proposed a 3-sound zero-knowledge proof that relied on the double-DSSP assumption. Such an assumption is too strong, as it can be broken by the same attacks as SIDH. However, we can modify the protocol to only rely on the DSSP assumption at the cost of no longer being 4-sound. Masking the torsion can be done easily by adding a random scalar in the protocol.

The core idea of **MTISOZKP**, which is presented later in this section, consists of generating two cyclic isogenies $\psi_{L,i} : E_0 \rightarrow E_{2,i}$ of degree B whose kernel generators form a basis of $E[B]$. These isogenies can be used with ϕ to construct two SIDH squares sharing an edge. In order not to leak information by doing so, we work with the dual of the isogenies of degree B as well as random torsion bases of $E_{2,i}$. Also, we use H in order to force the commitments to be honest without leaking information.

Definition 5.2.1 (**Masked Torsion ISOgeny Zero-Knowledge Proof (MTISOZKP)**). *Let A and B be two relatively prime positive integers. Let $\phi : E_0 \rightarrow E_1$ be a secret isogeny of degree A and let $\phi(P_0) = P_1$ and $\phi(Q_0) = Q_1$ where P_0 and Q_0 are a basis of $E_0[B]$. Let $\alpha \leftarrow_{\$} (\mathbb{Z}/B\mathbb{Z})^*$ be secret and $(E_0, E_1, P_0, Q_0, [\alpha]P_1, [\alpha]Q_1)$ be public. The challenge is a random $\text{chall} \in \{-1, 0, 1, 2\}$.*

COMMITMENT	RESPONSE
$(P'_1, Q'_1) \leftarrow ([\alpha]P_1, [\alpha]Q_1)$	$z_{L,0} \leftarrow (E_{2,0}, P_{2,0}, Q_{2,0}, r_{L_0})$
$K_\phi \leftarrow \text{GENERATINGPOINT}(\phi)$	$z_{L,1} \leftarrow (E_{2,1}, P_{2,1}, Q_{2,1}, r_{L_1})$
$(K_{\psi_{L,0}}, K_{\psi_{L,1}}) \leftarrow \text{RANDOMBASIS}(E_0, B)$	$z_{R,0} \leftarrow (E_{3,0}, P_{3,0}, Q_{3,0}, r_{R_0})$
$\beta \leftarrow \$ (\mathbb{Z}/B\mathbb{Z})^*$	$z_{R,1} \leftarrow (E_{3,1}, P_{3,1}, Q_{3,1}, r_{R_1})$
$(U, V) \leftarrow \text{RANDOMBASIS}(E_0, A)$	$z_{w,0} \leftarrow (U'_0, V'_0, r_{w,0})$
$(e, f) \leftarrow \text{DDLOG}(U, V, K_\phi)$	$z_{w,1} \leftarrow (U'_1, V'_1, r_{w,1})$
for $i \in \{0, 1\}$:	$z_{m,0} \leftarrow (c_0, d_0, c_1, d_1, c'_0, d'_0, a_0, b_0, r_{m,0})$
$\psi_{L,i} \leftarrow \text{ISOGENYFROMKERNEL}(K_{\psi_{L,i}})$	$z_{m,1} \leftarrow (c'_1, d'_1, a_1, b_1, r_{m,1})$
$E_{2,i} \leftarrow \text{CODOMAIN}(\psi_{L,i})$	if $\text{chall} = -1$:
$(P_{2,i}, Q_{2,i}) \leftarrow \text{RANDOMBASIS}(E_{2,i}, B)$	return $(z_{L,0}, z_{L,1}, z_{m,0}, z_{m,1}, z_{w,0}, z_{w,1})$
$\phi_i \leftarrow \text{PARALLELISOGENY}(\phi, \psi_{L,i})$	if $\text{chall} = 0$:
$\psi_{R,i} \leftarrow \text{PARALLELISOGENY}(\psi_{L,i}, \phi)$	return $(z_{R,0}, z_{R,1}, z_{m,0}, z_{m,1}, (\gamma, r_A))$
$E_{3,i} \leftarrow \text{CODOMAIN}(\phi_i)$	if $\text{chall} = 1$:
$(P_{3,i}, Q_{3,i}) \leftarrow ([\beta]\phi_i(P_{2,i}), [\beta]\phi_i(Q_{2,i}))$	return $(z_{w,0}, (e, f, r_E), z_{L,0}, z_{R,0}, (\beta, r_B))$
$K_{\hat{\psi}_{L,i}} \leftarrow \text{GENERATINGPOINT}(\hat{\psi}_{L,i})$	if $\text{chall} = 2$:
$(c_i, d_i) \leftarrow \text{DDLOG}(P_{2,i}, Q_{2,i}, K_{\hat{\psi}_{L,i}})$	return $(z_{w,1}, (e, f, r_E), z_{L,1}, z_{R,1}, (\beta, r_B))$
$(c'_i, d'_i) \leftarrow \text{INVERPAIR}(c_i, d_i, B)$	
$R_{0,i} \leftarrow \hat{\psi}_{L,i}([c'_i]P_{2,i} + [d'_i]Q_{2,i})$	
$(a_i, b_i) \leftarrow \text{DDLOG}(P_0, Q_0, R_{0,i})$	
$U'_i \leftarrow \psi_{L,i}(U)$	
$V'_i \leftarrow \psi_{L,i}(V)$	
$r_{L,i}, r_{R,i}, r_{m,i}, r_{w,i} \leftarrow \$ N$	
$C_{L,i} \leftarrow H(E_{2,i}, P_{2,i}, Q_{2,i}, r_{L,i})$	
$C_{R,i} \leftarrow H(E_{3,i}, P_{3,i}, Q_{3,i}, r_{R,i})$	
$C_{m,i} \leftarrow H(c_i, d_i, c'_i, d'_i, a_i, b_i, r_{m,i})$	
$C_{w,i} \leftarrow H(U'_i, V'_i, r_{w,i})$	
$r_A, r_B, r_E \leftarrow \$ N$	
$\gamma \leftarrow \alpha\beta^{-1}$	
$C_A \leftarrow H(\gamma, r_A)$	
$C_B \leftarrow H(\beta, r_B)$	
$C_E \leftarrow H(e, f, r_E)$	
$C_1 \leftarrow (C_{L,0}, C_{R,0}, C_{m,0}, C_{w,0})$	
$C_2 \leftarrow (C_{L,1}, C_{R,1}, C_{m,1}, C_{w,1})$	
$C_3 \leftarrow (C_A, C_B, C_E)$	
return (C_1, C_2, C_3)	

VERIFICATION

```

if  $chall = -1$  :
  for  $i \in \{0, 1\}$  :
    if  $C_{L,i} \neq H(E_{2,i}, P_{2,i}, Q_{2,i}, r_{L,i})$  : return  $\perp$ 
    if  $(C_{m,i}, C_{w,i}) \neq (H(c_i, d_i, c'_i, d'_i, a_i, b_i, r_{m,i}), H(U'_i, V'_i, r_{w,i}))$  : return  $\perp$ 
     $K_{\hat{\psi}_{L,i}} \leftarrow [c_i]P_{2,i} + [d_i]Q_{2,i}$ 
    if  $K_{\hat{\psi}_{L,i}} \notin E_{2,i}[[B]]$  : return  $\perp$ 
     $\hat{\psi}_{L,i} \leftarrow \text{ISOGENYFROMKERNEL}(K_{\hat{\psi}_{L,i}})$ 
     $E'_{0,i} \leftarrow \text{CODOMAIN}(\hat{\psi}_{L,i})$ 
    if  $E'_{0,i} \neq E_0$  : return  $\perp$ 
     $R'_{0,i} \leftarrow \hat{\psi}_{L,i}([c'_i]P_{2,i} + [d'_i]Q_{2,i})$ 
    if  $R'_{0,i} \neq [a_i]P_0 + [b_i]Q_0$  : return  $\perp$ 
    if  $(\text{GCD}(a_0b_1 - a_1b_0, B), \text{GCD}(c'_i d_i - d'_i c_i, B)) \neq (1, 1)$  : return  $\perp$ 
  if  $(\hat{\psi}_{L,0}(U'_0), \hat{\psi}_{L,0}(V'_0)) \neq (\hat{\psi}_{L,1}(U'_1), \hat{\psi}_{L,1}(V'_1))$  : return  $\perp$ 
if  $chall = 0$  :
  for  $i \in \{0, 1\}$  :
    if  $(C_{R,i}, C_A) \neq (H(E_{3,i}, P_{3,i}, Q_{3,i}, r_{R,i}), H(\gamma, r_A))$  : return  $\perp$ 
    if  $C_{m,i} \neq H(c_i, d_i, c'_i, d'_i, a_i, b_i, r_{m,i})$  : return  $\perp$ 
     $K_{\hat{\psi}_{R,i}} \leftarrow [c_i]P_{3,i} + [d_i]Q_{3,i}$ 
    if  $K_{\hat{\psi}_{R,i}} \notin E_{3,i}[[B]]$  : return  $\perp$ 
     $\hat{\psi}_{R,i} \leftarrow \text{ISOGENYFROMKERNEL}(K_{\hat{\psi}_{R,i}})$ 
     $E'_{1,i} \leftarrow \text{CODOMAIN}(\hat{\psi}_{R,i})$ 
    if  $E'_{1,i} \neq E'_1$  : return  $\perp$ 
     $R'_{1,i} \leftarrow \hat{\psi}_{R,i}([c'_i]P_{3,i} + [d'_i]Q_{3,i})$ 
    if  $[\gamma]R'_{1,i} \neq [a_i]P'_1 + [b_i]Q'_1$  : return  $\perp$ 
    if  $(\text{GCD}(a_0b_1 - a_1b_0, B), \text{GCD}(c'_i d_i - d'_i c_i, B)) \neq (1, 1)$  : return  $\perp$ 
else :
   $i \leftarrow chall - 1$  :
   $K_{\Phi'_i} \leftarrow [e]U'_i + [f]V'_i$ 
  if  $(C_{L,i}, C_{R,i}) \neq (H(E_{2,i}, P_{2,i}, Q_{2,i}, r_{L,i}), H(E_{3,i}, P_{3,i}, Q_{3,i}, r_{R,i}))$  : return  $\perp$ 
  if  $(C_{w,i}, C_E, C_B) \neq (H(U'_i, V'_i, r_{w,i}), H(e, f, r_E), H(\beta, r_B))$  : return  $\perp$ 
  if  $K_{\phi'_i} \notin E_{2,i}[[A]]$  : return  $\perp$ 
   $\phi'_i \leftarrow \text{ISOGENYFROMKERNEL}(K_{\phi'_i})$ 
   $E'_{3,i} \leftarrow \text{CODOMAIN}(\Phi'_i)$ 
  if  $(E'_{3,i}, [\beta]\phi'_i(P_{2,i}), [\beta]\phi'_i(Q_{2,i})) \neq (E_{3,i}, P_{3,i}, Q_{3,i})$  : return  $\perp$ 
return true

```

Theorem 5.2.2 (Correctness). *If the prover is honest, then the verification algorithm will always return **true**.*

Proof. If $\text{chall} = -1$, the properties checked by the verification algorithm were directly computed by the prover. Hence, this case will always be correct.

If $\text{chall} \in \{1, 2\}$, the properties checked by the verification algorithm are all respected by an honest $[\beta]\phi_{\text{chall}-1}$. Hence, this case will always be correct.

If $\text{chall} = 0$, we are working with almost the same SIDH square as in [25]. The main difference being that $(P_{3,i}, Q_{3,i})$ have an extra β factor and (P'_1, Q'_1) have an extra α factor. This is dealt with by multiplying $\psi_{R,i}$ by $[\alpha\beta^{-1}]$. $\square$

Theorem 5.2.3 (Soundness). *Given Assumption 4.2.3, MTISOZKP is 4-special sound for the knowledge of a cyclic isogeny of the claimed degree between the claimed curves with the claimed torsion point information.*

Proof. We show that for a fixed commitment, if one obtains valid answers to all 4 possible challenges, then they can compute an isogeny with the claimed properties. Since H is a computationally binding commitment scheme, we can assume that the four answers agree on the committed values.

The goal is to use the possible answers to compute an isogeny $\rho : E_0 \rightarrow E_1$ of degree A and an integer α such that $(P'_1, Q'_1) = ([\alpha]\rho(P_0), [\alpha]\rho(Q_0))$.

Looking at the isogeny square for each $i \in \{0, 1\}$, we are given the pair (c_i, d_i) which define the point $K_{\psi_{L,i}} = [c_i]P_{2,i} + [d_i]Q_{2,i}$ which in turn defines the isogeny $\psi_{L,i} : E_{2,i} \rightarrow E_0$ of degree B . We are also given $K_{\phi'_i} = [e]U'_i + [f]V'_i$ which defines an isogeny $\phi'_i : E_{2,i} \rightarrow E_{3,i}$ of degree A . We can then complete the $(\psi_{L,i}, \phi'_i)$ -isogeny square to obtain a ρ candidate of degree A that we name $\rho_i : E_0 \rightarrow E_1$.

Next, we show that ρ_0 and ρ_1 have the same kernel and are therefore equivalent. It is the case since

$$\begin{aligned} \ker(\rho_0) &= \psi_{L,0}(\ker(\phi'_0)) = \langle \psi_{L,0}([e]U'_0 + [f]V'_0) \rangle = \langle \psi_{L,1}([e]U'_1 + [f]V'_1) \rangle \\ &= \psi_{L,1}(\ker(\phi'_1)) = \ker(\rho_1) \end{aligned}$$

We also have an α candidate in $\gamma\beta$. All that remains is to show that $\rho = \rho_0 = \rho_1$ has the correct torsion point images.

Recall that, by using INVERPAIR, we are given pairs (a_i, b_i) such that $a_0b_1 - b_0a_1$ is invertible modulo B . This implies that the matrix $M := \begin{pmatrix} a_0 & b_0 \\ a_1 & b_1 \end{pmatrix}$ is also invertible. In addition, since $R_{0,i} = [a_i]P_0 + [b_i]Q_0$, we have that $\{R_{0,0}, R_{0,1}\}$ is a basis of $E_0[B]$.

Also recall that $R_{0,i} = \hat{\psi}_{L,i}([c'_i]P_{2,i} + [d'_i]Q_{2,i})$, $R_{1,i} = \hat{\psi}_{R,i}([c'_i]P_{3,i} + [d'_i]Q_{3,i})$ and $(P_{3,i}, Q_{3,i}) = ([\beta]\phi_i(P_{2,i}), [\beta]\phi_i(Q_{2,i}))$. Since $\rho\hat{\psi}_{L,i} = \hat{\psi}_{R,i}\phi_i$, we have that $\rho(R_{0,i}) = [\beta^{-1}]R_{1,i}$. Hence:

$$\begin{aligned} \begin{pmatrix} R_{0,0} \\ R_{0,1} \end{pmatrix} = M \begin{pmatrix} P_0 \\ Q_0 \end{pmatrix} &\implies \begin{pmatrix} \rho(R_{0,0}) \\ \rho(R_{0,1}) \end{pmatrix} = M \begin{pmatrix} \rho(P_0) \\ \rho(Q_0) \end{pmatrix} \\ \implies \begin{pmatrix} [\beta^{-1}]R_{1,0} \\ [\beta^{-1}]R_{1,1} \end{pmatrix} = M \begin{pmatrix} \rho(P_0) \\ \rho(Q_0) \end{pmatrix} &\implies M^{-1} \begin{pmatrix} [\beta^{-1}]R_{1,0} \\ [\beta^{-1}]R_{1,1} \end{pmatrix} = \begin{pmatrix} \rho(P_0) \\ \rho(Q_0) \end{pmatrix} \\ \implies \begin{pmatrix} [\gamma^{-1}\beta^{-1}]P'_1 \\ [\gamma^{-1}\beta^{-1}]Q'_1 \end{pmatrix} = \begin{pmatrix} \rho(P_0) \\ \rho(Q_0) \end{pmatrix} &\implies \begin{pmatrix} P'_1 \\ Q'_1 \end{pmatrix} = \begin{pmatrix} [\beta\gamma]\rho(P_0) \\ [\beta\gamma]\rho(Q_0) \end{pmatrix} \end{aligned}$$

and this completes the proof. $\square$

Theorem 5.2.4 (Zero-knowledge). *Given Assumptions 2.5.11 and 4.2.3, MTISOZKP is zero-knowledge.*

Proof. We prove it by showing a simulator $\mathcal{S}$ outputting a valid commitment-challenge-answer tuple with the same distribution as an honest prover for each possible challenge.

When the challenge is -1 , the value that is published without being masked by H can be computed honestly. $C_{R,0}$, $C_{R,1}$, C_A , and C_B can be randomly sampled from $\mathcal{H}$ while being indistinguishable from an honest output by Assumption 4.2.3.

When the challenge is 0 , the simulator can use the homomorphism property of isogenies to work on the right side of the SIDH squares instead of the left. The masked values can once again be sampled randomly.

When the challenge is 1 or 2 , the simulator can sample a random ϕ_i and β and compute the rest using these values. The masked values are, again, sampled randomly. Distinguishing this simulator from an honest output is equivalent to solving the DSSP, which we assume to be hard. $\square$

5.3 Double Masked Subgroup

Instead of multiplying both torsion points images by the same constant, binSIDH, terSIDH [8] and the diagonal variant of FESTA [9] multiply each point by independent random scalars. This has the consequence of making MTISOZKP hard to adapt in this case, as the correctness of the previous protocol relies on the fact that the isogeny multiplying every point by the same constant commutes with every isogeny. Needing to multiply each basis

point by a different scalar loses this commutative property, which means that we must look elsewhere for a zero-knowledge proof.

DMSISOZKP is a zero-knowledge proving that a party known an isogeny $\phi : E_0 \rightarrow E_1$ such that $([\alpha_P]\phi(P_0), [\alpha_Q]\phi(Q_0)) = (P_1, Q_1)$ for some unknown values of α_P and α_Q . Similarly to MTISOZKP, DMSISOZKP consists of building an SIDH square and using dual isogenies and hashed commitments to maintain zero-knowledge and soundness. The main difference being that, in DMSISOZKP, the generated isogenies are of degree C and the random basis is of order BC . This is so that we can prove information on the B torsion. While only requiring Assumptions 4.2.3 and 2.5.11 for its security, DMSISOZKP requires some additional conditions on the field for it to be efficient. We need a field such that isogenies of degree C can be efficiently computed while, at the same time, points of order BC can be efficiently computed and used in other computations. In practice, this requires the chosen prime to be about 50% larger, as isogenies of degree C must also be hard to attack.

Definition 5.3.1 (**Double Masked Scalar ISOgeny Zero-Knowledge Proof (DMSISOZKP)**). *Let A, B and C be two relatively prime positive integers. Let $\phi : E_0 \rightarrow E_1$ be a secret isogeny of degree A and let $(\phi(P_0), \phi(Q_0)) = (P_0, Q_0)$ where (P_1, Q_1) form a basis of $E_0[B]$. Let $\alpha_P, \alpha_Q \leftarrow_{\$} (\mathbb{Z}/(BC)\mathbb{Z})^*$ be secret and $(E_0, E_1, P_0, Q_0, [\alpha_P]P_1, [\alpha_Q]Q_1)$ be public. The challenge is a random $\text{chall} \in \{-1, 0, 1\}$.*

COMMITMENT

```

( $P'_1, Q'_1$ )  $\leftarrow ([\alpha_P]P_1, [\alpha_Q]Q_1)$ 
 $K_\phi \leftarrow \text{GENERATINGPOINT}(\phi)$ 
 $K_{\psi_L} \leftarrow \$E[[C]]$ 
 $\psi_L \leftarrow \text{ISOGENYFROMKERNEL}(K_{\psi_L})$ 
 $\psi_R \leftarrow \text{PARALLELISOGENY}(\psi_L, \phi)$ 
 $E_2 \leftarrow \text{CODOMAIN}(\psi_L)$ 
 $E_3 \leftarrow \text{CODOMAIN}(\psi_R)$ 
 $(P_{2,P}, Q_{2,P}) \leftarrow \text{RANDOMBASIS}(E_2, BC)$ 
 $(P_{2,Q}, Q_{2,Q}) \leftarrow \text{RANDOMBASIS}(E_2, BC)$ 
 $\beta_P, \beta_Q \leftarrow \$(\mathbb{Z}/(BC)\mathbb{Z})^*$ 
 $\phi' \leftarrow \text{PARALLELISOGENY}(\phi, \psi_L)$ 
 $(P_{3,P}, Q_{3,P}) \leftarrow ([\beta_P]\phi'(P_{2,P}), [\beta_P]\phi'(Q_{2,P}))$ 
 $(P_{3,Q}, Q_{3,Q}) \leftarrow ([\beta_Q]\phi'(P_{2,Q}), [\beta_Q]\phi'(Q_{2,Q}))$ 
 $K_{\hat{\psi}_L} \leftarrow \text{GENERATINGPOINT}(\hat{\psi}_L)$ 
 $(c, d) \leftarrow \text{DDLOG}(P_{2,P}, Q_{2,P}, K_{\hat{\psi}_L})$ 
 $(a_P, b_P) \leftarrow \text{DDLOG}(\hat{\psi}_L(P_{2,P}), \hat{\psi}_L(Q_{2,P}), P_0)$ 
 $(a_Q, b_Q) \leftarrow \text{DDLOG}(\hat{\psi}_L(P_{2,Q}), \hat{\psi}_L(Q_{2,Q}), Q_0)$ 
 $\gamma_P \leftarrow \alpha_P \beta_P^{-1}$ 
 $\gamma_Q \leftarrow \alpha_Q \beta_Q^{-1}$ 
 $r_L, r_R, r_m, r_A, r_B \leftarrow \$N$ 
 $C_L \leftarrow H(E_2, P_{2,P}, Q_{2,P}, P_{2,Q}, Q_{2,Q}, r_L)$ 
 $C_R \leftarrow H(E_3, P_{3,P}, Q_{3,P}, P_{3,Q}, Q_{3,Q}, r_R)$ 
 $C_m \leftarrow H(a_P, b_P, a_Q, b_Q, c, d, r_m)$ 
 $C_A \leftarrow H(\gamma_P, \gamma_Q, r_A)$ 
 $C_B \leftarrow H(\beta_P, \beta_Q, r_B)$ 
return ( $C_L, C_R, C_m, C_A, C_B$ )

```

RESPONSE

```

 $z_L \leftarrow (E_2, P_{2,P}, Q_{2,P}, P_{2,Q}, Q_{2,Q}, r_L)$ 
 $z_R \leftarrow (E_3, P_{3,P}, Q_{3,P}, P_{3,Q}, Q_{3,Q}, r_R)$ 
 $z_m \leftarrow (a_P, b_P, a_Q, b_Q, c, d, r_m)$ 
if  $chall = -1$  :
    return ( $z_L, z_m$ )
if  $chall = 0$  :
    return ( $z_R, z_m, \gamma_P, \gamma_Q, r_A$ )
if  $chall = 1$  :
    return ( $z_L, z_R, \beta_P, \beta_Q, r_B, K_{\phi'}$ )

```

VERIFICATION

```

if  $chall = -1$  :
    if  $C_L \neq H(E_2, P_{2,P}, Q_{2,P}, P_{2,Q}, Q_{2,Q}, r_L)$  : return  $\perp$ 
    if  $C_m \neq H(a_P, b_P, a_Q, b_Q, c, d, r_m)$  : return  $\perp$ 
    if  $\neg \text{ISBASIS}(P_{2,P}, Q_{2,P}, E_2, BC)$  : return  $\perp$ 
    if  $\neg \text{ISBASIS}(P_{2,Q}, Q_{2,Q}, E_2, BC)$  : return  $\perp$ 
     $K_{\hat{\psi}_L} \leftarrow [c]P_{2,P} + [d]Q_{2,P}$ 
    if  $K_{\hat{\psi}_L} \notin E_2[[C]]$  : return  $\perp$ 
     $\hat{\psi}_L \leftarrow \text{ISOGENYFROMKERNEL}(K_{\hat{\psi}_L})$ 
     $E'_0 \leftarrow \text{CODOMAIN}(\hat{\psi}_L)$ 
    if  $E'_0 \neq E_0$  : return  $\perp$ 
    if  $P_0 \neq [a_P]\hat{\psi}_L(P_{2,P}) + [b_P]\hat{\psi}_L(Q_{2,P})$  : return  $\perp$ 
    if  $Q_0 \neq [a_Q]\hat{\psi}_L(P_{2,Q}) + [b_Q]\hat{\psi}_L(Q_{2,Q})$  : return  $\perp$ 
if  $chall = 0$  :
    if  $C_R \neq H(E_3, P_{3,P}, Q_{3,P}, P_{3,Q}, Q_{3,Q}, r_R)$  : return  $\perp$ 
    if  $C_m \neq H(a_P, b_P, a_Q, b_Q, c, d, r_m)$  : return  $\perp$ 
    if  $C_A \neq H(\gamma_P, \gamma_Q, r_A)$  : return  $\perp$ 
    if  $\neg \text{ISBASIS}(P_{3,P}, Q_{3,P}, E_3, BC)$  : return  $\perp$ 
    if  $\neg \text{ISBASIS}(P_{3,Q}, Q_{3,Q}, E_3, BC)$  : return  $\perp$ 
     $K_{\hat{\psi}_R} \leftarrow [c]P_{3,P} + [d]Q_{3,P}$ 
    if  $K_{\hat{\psi}_R} \notin E_3[[C]]$  : return  $\perp$ 
     $\hat{\psi}_R \leftarrow \text{ISOGENYFROMKERNEL}(K_{\hat{\psi}_R})$ 
     $E'_1 \leftarrow \text{CODOMAIN}(\hat{\psi}_R)$ 
    if  $E'_1 \neq E_1$  : return  $\perp$ 
    if  $P'_1 \neq [a_P\gamma_P]\hat{\psi}_R(P_{3,P}) + [b_P\gamma_P]\hat{\psi}_R(Q_{3,P})$  : return  $\perp$ 
    if  $Q'_1 \neq [a_Q\gamma_Q]\hat{\psi}_R(P_{3,Q}) + [b_Q\gamma_Q]\hat{\psi}_R(Q_{3,Q})$  : return  $\perp$ 
if  $chall = 1$  :
    if  $C_L \neq H(E_2, P_{2,P}, Q_{2,P}, P_{2,Q}, Q_{2,Q}, r_L)$  : return  $\perp$ 
    if  $C_R \neq H(E_3, P_{3,P}, Q_{3,P}, P_{3,Q}, Q_{3,Q}, r_R)$  : return  $\perp$ 
    if  $C_B \neq H(\beta_P, \beta_Q, r_B)$  : return  $\perp$ 
    if  $K_{\phi'} \notin E_2[[A]]$  : return  $\perp$ 
     $\phi' \leftarrow \text{ISOGENYFROMKERNEL}(K_{\phi'})$ 
     $E'_3 \leftarrow \text{CODOMAIN}(\phi')$ 
    if  $E'_3 \neq E_3$  : return  $\perp$ 
    if  $P_{3,P} \neq [\beta_P]\phi'(P_{2,P})$  : return  $\perp$ 
    if  $Q_{3,P} \neq [\beta_P]\phi'(Q_{2,P})$  : return  $\perp$ 
    if  $P_{3,Q} \neq [\beta_Q]\phi'(P_{2,Q})$  : return  $\perp$ 
    if  $Q_{3,Q} \neq [\beta_Q]\phi'(Q_{2,Q})$  : return  $\perp$ 
return true

```

Theorem 5.3.2 (Correctness). *If the prover is honest, then the verification algorithm will always return **true**.*

Proof. If $\text{chall} = -1$, the properties checked by the verification algorithm were directly computed by the prover. Hence, this case will always be correct.

If $\text{chall} = 1$, the properties checked by the verification algorithm are all respected by honest $[\beta_P]\phi'$ and $[\beta_Q]\phi'$. Hence, this case will always be correct.

If $\text{chall} = 0$, we have that $\hat{\psi}_R\phi' = \phi\hat{\psi}_L$ since the four isogenies form an SIDH square. Hence:

$$\begin{aligned} P'_1 &= [\alpha_P]\phi(P_0) \\ &= [\alpha_P]\phi([a_P]\hat{\psi}_L(P_{2,P}) + [b_P]\hat{\psi}_L(Q_{2,P})) \\ &= [a_P\alpha_P]\phi\hat{\psi}_L(P_{2,P}) + [b_P\alpha_P]\phi\hat{\psi}_L(Q_{2,P}) \\ &= [a_P\alpha_P]\hat{\psi}_R\phi'(P_{2,P}) + [b_P\alpha_P]\hat{\psi}_R\phi'(Q_{2,P}) \\ &= [a_P\gamma_P]\hat{\psi}_R(P_{3,P}) + [b_P\gamma_P]\hat{\psi}_R(Q_{3,P}) \end{aligned}$$

which is the checked equation. The same argument holds for $Q'_1 = [a_Q\gamma_Q]\hat{\psi}_R(P_{3,Q}) + [b_Q\gamma_Q]\hat{\psi}_R(Q_{3,Q})$. $\square$

Theorem 5.3.3 (Zero-knowledge). *Given Assumptions 4.2.3 and 2.5.11, DMSISOZKP is zero-knowledge.*

Proof. We prove it by showing a simulator outputting a valid commitment-challenge-answer tuple with the same distribution as an honest prover for each possible challenge.

When the challenge is -1 , the simulator can compute the revealed values honestly and sample random values for the masked data, which is indistinguishable from random by Assumption 4.2.3.

When the challenge is 0, the simulator can use the homomorphism property of isogenies to work on the right side of the SIDH squares instead of the left. The masked values can once again be sampled randomly.

When the challenge is 1, the simulator can sample a random ϕ' and (β_P, β_Q) and compute the rest using these values. The masked values are, again, sampled randomly. Distinguishing this simulator from an honest output is equivalent to solving the DSSP, which we assume to be hard. $\square$

Theorem 5.3.4 (Soundness). *Given Assumption 4.2.3, DMSISOZKP is 3-special sound for the knowledge of a cyclic isogeny of the claimed degree between the claimed curves with the claimed torsion point information.*

Proof. We show that for a fixed commitment, if one obtains valid answers to all 3 possible challenges, then they can compute an isogeny with the claimed properties.

Since H is a computationally binding commitment scheme, we can assume that the three answers agree on the committed values.

The goal is to use the possible answers in order to compute an isogeny $\rho : E_0 \rightarrow E_1$ of degree A and a pair of integers (α_P, α_Q) such that $(P'_1, Q'_1) = ([\alpha_P]\rho(P_0), [\alpha_Q]\rho(Q_0))$.

We are given the pair (c, d) which define the point $K_{\hat{\psi}_L}$, which in turn defines the isogeny $\hat{\psi}_L : E_2 \rightarrow E_0$ of degree C . We are also given the point $K_{\phi'}$ which defines the isogeny $\phi' : E_2 \rightarrow E_3$. We can then complete the $(\hat{\psi}_L, \phi')$ -isogeny square to obtain our ρ candidate.

We also have a (α_P, α_Q) candidate in $(\gamma_P\beta_P, \gamma_Q\beta_Q)$.

We have that $\rho : E_0 \rightarrow E_1$ is of degree A , so we only need to check that it respects the claimed mapping. Since ρ is constructed by completing an SIDH square, we have that $\rho\hat{\psi}_L = \hat{\psi}_R\phi'$. Hence:

$$\begin{aligned} \rho(P_0) &= \rho([a_P]\hat{\psi}_L(P_{2,P}) + [b_P]\hat{\psi}_L(Q_{2,P})) \\ &= [a_P]\rho\hat{\psi}_L(P_{2,P}) + [b_P]\rho\hat{\psi}_L(Q_{2,P}) \\ &= [a_P]\hat{\psi}_R\phi'(P_{2,P}) + [b_P]\hat{\psi}_R\phi'(Q_{2,P}) \\ \rho(P_0) &= [\beta_P^{-1}\gamma_P^{-1}]P'_1 \\ [\alpha_P]\rho(P_0) &= P'_1 \end{aligned}$$

The same arguments holds for $[\alpha_Q]\rho(Q_0) = Q'_1$. Therefore, ρ is a valid secret isogeny and this completes the proof. $\square$

5.4 Masked Degree and Double Subgroup

Section 5.1's technique, used to prove knowledge of an isogeny while masking its degree, can be combined with any of the zero-knowledge proofs in this paper in order to prove the

desired torsion point information. Since the combination method and security proofs of every case are almost identical, we only explicitly present one of them in this paper.

For applications such as terSIDH [8], we require a zero-knowledge proof that can prove knowledge of an isogeny with the given subgroup images without leaking information about either the isogeny itself or its degree. In order to do this, we start DMSISOZKP and add the random sampling technique presented in MDISOZKP.

Definition 5.4.1 (**Masked Degree and Torsion ISOgeny Zero-Knowledge Proof (MDTI-SOZKP)**). *Let $\mathcal{A} = \prod_{i=1}^s q_i^{f_i}$ be a large integer such that the q_i s are distinct primes dividing $p + 1$. Let B and C be large positive integers relatively prime to $\mathcal{A}$. Let $\phi : E_0 \rightarrow E_1$ be a secret cyclic isogeny of degree $A \mid \mathcal{A}$ and let $(\phi(P_0), \phi(Q_0)) = (P_1, Q_1)$ where (P_1, Q_1) form a basis of $E_0[B]$. Let $\alpha_P, \alpha_Q \leftarrow_{\$} (\mathbb{Z}/(BC)\mathbb{Z})^*$ be secret and $(E_0, E_1, P_0, Q_0, [\alpha_P]P_1, [\alpha_Q]Q_1)$ be public.*

Let $(P'_1, Q'_1) := ([\alpha_P]P_1, [\alpha_Q]Q_1)$ and let n be a positive integer representing the number of times the following proof will be repeated. The challenge is a random $\text{chall} \in \{-1, 0, 1\}$.

COMMITMENT

```

 $A' \leftarrow \$ \text{FACSET}(\mathcal{A}^{sn})$ 
 $\phi' \leftarrow \text{CYCLICISOGENY}(E_1, A')$ 
 $E'_1 \leftarrow \text{CODOMAIN}(\phi')$ 
 $\Phi \leftarrow \text{CYCLIPHY}(\phi' \phi)$ 
 $A'' \leftarrow \text{deg}(\Phi)$ 
 $\xi \leftarrow A'' / (A \times A')$ 
 $K_{\psi_L} \leftarrow \$ E[[C]]$ 
 $K_{\psi_R} \leftarrow \Phi(K_{\psi_L})$ 
 $\psi_L \leftarrow \text{ISOGENYFROMKERNEL}(K_{\psi_L})$ 
 $\psi_R \leftarrow \text{ISOGENYFROMKERNEL}(K_{\psi_R})$ 
 $E_2 \leftarrow \text{CODOMAIN}(\psi_L)$ 
 $E_3 \leftarrow \text{CODOMAIN}(\psi_R)$ 
 $(P_{2,P}, Q_{2,P}) \leftarrow \text{RANDOMBASIS}(E_2, BC)$ 
 $(P_{2,Q}, Q_{2,Q}) \leftarrow \text{RANDOMBASIS}(E_2, BC)$ 
 $\beta_P, \beta_Q \leftarrow \$ (\mathbb{Z}/(BC)\mathbb{Z})^*$ 
 $\Phi' \leftarrow \text{PARALLELISOGENY}(\Phi, \psi_L)$ 
 $(P_{3,P}, Q_{3,P}) \leftarrow ([\beta_P]\Phi'(P_{2,P}), [\beta_P]\Phi'(Q_{2,P}))$ 
 $(P_{3,Q}, Q_{3,Q}) \leftarrow ([\beta_Q]\Phi'(P_{2,Q}), [\beta_Q]\Phi'(Q_{2,Q}))$ 
 $K_{\hat{\psi}_L} \leftarrow \text{GENERATINGPOINT}(\hat{\psi}_L)$ 
 $(c, d) \leftarrow \text{DDLOG}(P_{2,P}, Q_{2,P}, K_{\hat{\psi}_L})$ 
 $(a_P, b_P) \leftarrow \text{DDLOG}(\hat{\psi}_L(P_{2,P}), \hat{\psi}_L(Q_{2,P}), P_0)$ 
 $(a_Q, b_Q) \leftarrow \text{DDLOG}(\hat{\psi}_L(P_{2,Q}), \hat{\psi}_L(Q_{2,Q}), Q_0)$ 
 $\gamma_P \leftarrow \alpha_P \beta_P^{-1} \xi^{-1}$ 
 $\gamma_Q \leftarrow \alpha_Q \beta_Q^{-1} \xi^{-1}$ 
 $r_L, r_R, r_m, r_A, r_B \leftarrow \$ N$ 
 $C_L \leftarrow H(E_2, P_{2,P}, Q_{2,P}, P_{2,Q}, Q_{2,Q}, r_L)$ 
 $C_R \leftarrow H(E_3, P_{3,P}, Q_{3,P}, P_{3,Q}, Q_{3,Q}, r_R)$ 
 $C_m \leftarrow H(a_P, b_P, a_Q, b_Q, c, d, r_m)$ 
 $C_A \leftarrow H(\gamma_P, \gamma_Q, r_A)$ 
 $C_B \leftarrow H(\beta_P, \beta_Q, r_B)$ 
return  $(C_L, C_R, C_m, C_A, C_B)$ 

```

VERIFICATION

```

if  $chall = -1$  :
  if  $C_L \neq H(E_2, P_{2,P}, Q_{2,P}, P_{2,Q}, Q_{2,Q}, r_L)$  : return  $\perp$ 
  if  $C_m \neq H(a_P, b_P, a_Q, b_Q, c, d, r_m)$  : return  $\perp$ 
  if  $\neg \text{ISBASIS}(P_{2,P}, Q_{2,P}, E_2, BC)$  : return  $\perp$ 
  if  $\neg \text{ISBASIS}(P_{2,Q}, Q_{2,Q}, E_2, BC)$  : return  $\perp$ 
   $K_{\hat{\psi}_L} \leftarrow [c]P_{2,P} + [d]Q_{2,P}$ 
  if  $K_{\hat{\psi}_L} \notin E_2[[C]]$  : return  $\perp$ 
   $\hat{\psi}_L \leftarrow \text{ISOGENYFROMKERNEL}(K_{\hat{\psi}_L})$ 
   $E'_0 \leftarrow \text{CODOMAIN}(\hat{\psi}_L)$ 
  if  $E'_0 \neq E_0$  : return  $\perp$ 
  if  $P_0 \neq [a_P]\hat{\psi}_L(P_{2,P}) + [b_P]\hat{\psi}_L(Q_{2,P})$  : return  $\perp$ 
  if  $Q_0 \neq [a_Q]\hat{\psi}_L(P_{2,Q}) + [b_Q]\hat{\psi}_L(Q_{2,Q})$  : return  $\perp$ 
if  $chall = 0$  :
  if  $C_R \neq H(E_3, P_{3,P}, Q_{3,P}, P_{3,Q}, Q_{3,Q}, r_R)$  : return  $\perp$ 
  if  $C_m \neq H(a_P, b_P, a_Q, b_Q, c, d, r_m)$  : return  $\perp$ 
  if  $C_A \neq H(\gamma_P, \gamma_Q, r_A)$  : return  $\perp$ 
  if  $\neg \text{ISBASIS}(P_{3,P}, Q_{3,P}, E_3, BC)$  : return  $\perp$ 
  if  $\neg \text{ISBASIS}(P_{3,Q}, Q_{3,Q}, E_3, BC)$  : return  $\perp$ 
   $K_{\hat{\psi}_R} \leftarrow [c]P_{3,P} + [d]Q_{3,P}$ 
  if  $K_{\hat{\psi}_R} \notin E_3[[C]]$  : return  $\perp$ 
   $\hat{\psi}_R \leftarrow \text{ISOGENYFROMKERNEL}(K_{\hat{\psi}_R})$ 
   $E'_1 \leftarrow \text{CODOMAIN}(\hat{\psi}_R)$ 
  if  $\neg \text{ISOVALID}(E_1, E'_1, \phi')$  : return  $\perp$ 
  if  $\text{deg}(\phi') \nmid \mathcal{A}^{sn}$  : return  $\perp$ 
  if  $\phi'(P'_1) \neq [a_P\gamma_P]\hat{\psi}_R(P_{3,P}) + [b_P\gamma_P]\hat{\psi}_R(Q_{3,P})$  : return  $\perp$ 
  if  $\phi'(Q'_1) \neq [a_Q\gamma_Q]\hat{\psi}_R(P_{3,Q}) + [b_Q\gamma_Q]\hat{\psi}_R(Q_{3,Q})$  : return  $\perp$ 
if  $chall = 1$  :
  if  $C_L \neq H(E_2, P_{2,P}, Q_{2,P}, P_{2,Q}, Q_{2,Q}, r_L)$  : return  $\perp$ 
  if  $C_R \neq H(E_3, P_{3,P}, Q_{3,P}, P_{3,Q}, Q_{3,Q}, r_R)$  : return  $\perp$ 
  if  $C_B \neq H(\beta_P, \beta_Q, r_B)$  : return  $\perp$ 
  if  $\neg \text{ISOVALID}(E_2, E_3, \Phi')$  : return  $\perp$ 
  if  $\text{deg}(\Phi') \notin \text{FACSETTWO}(\mathcal{A}, \mathcal{A}^{sn})$  : return  $\perp$ 
  if  $(P_{3,P}, Q_{3,P}) \neq ([\beta_P]\phi'(P_{2,P}), [\beta_P]\phi'(Q_{2,P}))$  : return  $\perp$ 
  if  $(P_{3,Q}, Q_{3,Q}) \neq ([\beta_Q]\phi'(P_{2,Q}), [\beta_Q]\phi'(Q_{2,Q}))$  : return  $\perp$ 
return true

```

RESPONSE

```

if  $chall = -1$  :
    return  $(E_2, P_{2,P}, Q_{2,P}, P_{2,Q}, Q_{2,Q}, r_L, a_P, b_P, a_Q, b_Q, c, d, r_m)$ 
if  $chall = 0$  :
    return  $(E_3, P_{3,P}, Q_{3,P}, P_{3,Q}, Q_{3,Q}, r_R, a_P, b_P, a_Q, b_Q, c, d, r_m, \gamma_P, \gamma_Q, r_A, \phi')$ 
if  $chall = 1$  :
    if  $A'' \notin \text{FACSETTWO}(\mathcal{A}, \mathcal{A}^{sn})$  : abort
    return  $(E_2, P_{2,P}, Q_{2,P}, P_{2,Q}, Q_{2,Q}, r_L, E_3, P_{3,P}, Q_{3,P}, P_{3,Q}, Q_{3,Q}, r_R, \beta_P, \beta_Q, r_B, \Phi')$ 

```

Theorem 5.4.2 (Correctness). *If the prover is honest and does not abort, then the verification algorithm will always return **true**.*

Proof. If $chall = -1$, the properties checked by the verification algorithm were directly computed by the prover. Hence, this case will always be correct.

If $chall = 1$, the properties checked by the verification algorithm are all respected by honest $[\beta_P]\Phi'$ and $[\beta_Q]\Phi'$. Hence, this case will always be correct.

If $chall = 0$, we have that $\hat{\psi}_R\Phi' = \Phi\hat{\psi}_L$ since the four isogenies form an SIDH square. We also have that $\phi\phi' = [\xi]\Phi$. Hence:

$$\begin{aligned}
 \phi'(P'_1) &= [\alpha_P]\phi\phi'(P_0) \\
 &= [\alpha_P\xi]\Phi(P_0) \\
 &= [\alpha_P\xi]\Phi([a_P]\hat{\psi}_L(P_{2,P}) + [b_P]\hat{\psi}_L(Q_{2,P})) \\
 &= [a_P\alpha_P\xi]\Phi\hat{\psi}_L(P_{2,P}) + [b_P\alpha_P\xi]\Phi\hat{\psi}_L(Q_{2,P}) \\
 &= [a_P\alpha_P\xi]\hat{\psi}_R\Phi'(P_{2,P}) + [b_P\alpha_P\xi]\hat{\psi}_R\Phi'(Q_{2,P}) \\
 &= [a_P\gamma_P]\hat{\psi}_R(P_{3,P}) + [b_P\gamma_P]\hat{\psi}_R(Q_{3,P})
 \end{aligned}$$

which is the checked equation. The same argument hold for $\phi'(Q'_1) = [a_Q\gamma_Q]\hat{\psi}_R(P_{3,Q}) + [b_Q\gamma_Q]\hat{\psi}_R(Q_{3,Q})$. $\square$

Before proving the security of MDTISOZKP, it is important to remark that Theorems 5.1.5 and 5.1.7 also hold for MDTISOZKP as the proof is identical. Hence, the probability of the scheme not aborting during n rounds is at least $\frac{1}{e}$.

Theorem 5.4.3 (Zero-knowledge). *Given Assumptions 4.2.3 and 2.5.11, if MDTISOZKP does not abort, then it is zero-knowledge.*

Proof. We prove it by showing a simulator outputting a valid commitment-challenge-answer tuple with the same distribution as an honest prover for each possible challenge.

When the challenge is -1 , the simulator can compute the revealed values honestly and sample random values for the masked data, which is indistinguishable from an honest output by Assumption 4.2.3.

When the challenge is 0 , the simulator can compute ϕ' honestly. Then, we can use the homomorphism property of isogenies to work on the right side of the SIDH squares instead of the left. The masked values can once again be sampled randomly.

When the challenge is 1 , the simulator can sample a random Φ' and (β_P, β_Q) and compute the rest using these values. The masked values are, again, sampled randomly. The degree of Φ' is indistinguishable by Theorem 5.1.5. Hence, distinguishing this simulator from an honest output is equivalent to solving the DSSP, which we assume to be hard. $\square$

Theorem 5.4.4 (Soundness). *Given Assumption 4.2.3, MDTISOZKP is 3-special sound for the knowledge of a cyclic isogeny between the claimed curves with the claimed torsion point information.*

Proof. We show that for a fixed commitment, if one obtains valid answers to all 3 possible challenges, then they can compute an isogeny with the claimed properties.

Since H is a computationally binding commitment scheme, we can assume that the three answers agree on the committed values.

We are given an isogeny $\phi' : E_1 \rightarrow E'_1$ of degree A' . Let $P''_1 = \phi'(P'_1)$ and $Q''_1 = \phi'(Q'_1)$. Given an isogeny $\Phi : E_0 \rightarrow E'_1$ such that $\Phi(P_0) = [\delta_P]P''_1$ and $\Phi(Q_0) = [\delta_Q]Q''_1$, $\phi'\Phi$ is a valid extractor.

Therefore, the goal is to use the possible answers in order to compute an isogeny $\rho : E_0 \rightarrow E'_1$ of degree A'' and a pair of integers (δ_P, δ_Q) such that $(P''_1, Q''_1) = ([\delta_P]\rho(P_0), [\delta_Q]\rho(Q_0))$.

We are given the pair (c, d) which define the point $K_{\hat{\psi}_L}$, which in turn defines the isogeny $\hat{\psi}_L : E_2 \rightarrow E_0$ of degree C . We also have the isogeny $\Phi' : E_2 \rightarrow E_3$ of degree A'' . We can then complete the $(\hat{\psi}_L, \Phi')$ -isogeny square to obtain our ρ candidate.

We also chose (δ_P, δ_Q) as our candidate to be a valid value of $(\gamma_P\beta_P, \gamma_Q\beta_Q)$.

We have that $\rho : E_0 \rightarrow E'_1$ is of degree A'' , so we only need to check that it respects the claimed mapping. Since ρ is constructed by completing an SIDH square, we have that $\rho\hat{\psi}_L = \hat{\psi}_R\Phi'$. Hence:

$$\begin{aligned}
\rho(P_0) &= \rho([a_P]\hat{\psi}_L(P_{2,P}) + [b_P]\hat{\psi}_L(Q_{2,P})) \\
&= [a_P]\rho\hat{\psi}_L(P_{2,P}) + [b_P]\rho\hat{\psi}_L(Q_{2,P}) \\
&= [a_P]\hat{\psi}_R\Phi'(P_{2,P}) + [b_P]\hat{\psi}_R\Phi'(Q_{2,P}) \\
&= [a_P\beta_P^{-1}]\hat{\psi}_R P_{4,P} + [b_P\beta_P^{-1}]\hat{\psi}_R Q_{3,P} \\
\rho(P_0) &= [\beta_P^{-1}\gamma_P^{-1}]P_1'' \\
[\delta_P]\rho(P_0) &= P_1''
\end{aligned}$$

The same arguments hold for $[\delta_Q]\rho(Q_0) = Q_1''$. Therefore, ρ can be used to generate a valid secret isogeny and this completes the proof. □

5.5 Conclusion

Using the schemes in this chapter, we can prove knowledge of isogenies with masked torsion-point information while either proving the degree or masking it, as desired.

The fact that the security of our scheme relies mainly on DSSP allows us to obtain statistical zero-knowledge using large enough parameters as a consequence of Theorem [2.4.6](#).

For further research, it is worth mentioning that some variants of FESTA use non-diagonal matrices to mask their torsion point information. In those cases, the schemes in this paper do not apply. However, it is very likely that the technique in DMSISOZKP can be generalized for other abelian subgroups of $GL_2(\mathbb{Z}/B\mathbb{Z})$.

Chapter 6

Conclusion

During the research presented in this thesis, we have explored the feasibility of working with elliptic curves of unknown endomorphism rings for implementations of SIDH variants.

From Chapter 3, we have seen that if the level structure is chosen appropriately and the degree of the secret isogeny is large enough compared to the size of the revealed level-structure mapping, then the key recovery problem associated with SIDH variants is randomly self-reducible. These results allow one to be assured that using random starting curves can never create new vulnerabilities when our results apply. There are many possible avenues for improving our results. A direct way to do so is that our bounds for the success probability of a reduction attempt could easily be improved if one could find better bounds for either the eigenvalues of the isogeny graphs studied or Chebyshev's polynomials of the second kind. Any such improvement would lead to a lowering of the required parameter sizes to guarantee random self-reducibility. Another point to improve is that our results, as they currently are, require the isogeny graph to be an undirected multigraph, leading to restrictions on the chosen parameters. This restriction does not appear in papers such as [6] or [18] because they study walks on a single isogeny graph. Doing so works well when the desired application can allow for the degree of secret isogeny to always be a prime power. However, for more generic applications where the degree can be a product of prime powers, we have to study how multiple isogeny graphs interact with each other. In such a case, having the adjacency matrices of the different graphs not be symmetrical, and therefore possibly not commute with each other, makes working with them much harder. On the other hand, this lack of symmetry can be attributed to at most two vertices in a graph of cryptographic size. It is therefore likely that a similar result can be generalized to most parameters by studying in more detail the error term introduced by these vertices.

However, our results can still be used as they currently are. They are especially useful when one wants to chain isogenies in a multiparty setting, such as in [6] and Chapter 4.

On the subject of Chapter 4, we presented a multiparty method to generate supersingular elliptic curves over $\mathbb{F}_p$ with unknown endomorphism rings. In addition to [6] doing the same thing for supersingular curves over $\mathbb{F}_{p^2}$, it is now feasible to generate curves defined on the two common fields used for isogeny-based cryptography. Additionally, using the curve randomizer algorithm also allows a group to generate a random curve on either field without having to trust each other. The main limitation of our results is the reliance on a knowledge of exponent assumption, which is not as well studied as other isogeny primitives. Until then, a solution would be to find another way to sample elements of the ideal class group, in which leaking the degree of the associated isogeny does not reveal too much about it. The techniques introduced in Clapoti [47] seem like a promising start for this direction as they allow for the computation of the action of any ideal class in polynomial time in terms of the $\log(p)$, without requiring the ideal to be represented as the product of small norm ideals. If we can find a way to convert the output of $\mathbf{e}$ of CSIDHSAMPLE into an ideal whose norm does not give information about $\mathbf{e}$, then we could construct a signature scheme similar to SeaSign without the need for rejection sampling and potentially being more efficient.

Finally, in Chapter 5, we presented zero-knowledge proofs based on the masked-SIDH, Borel and Cartan level structures. These proofs only rely on the DSSP assumption and can be used to defend the associated SIDH variant from adaptive attacks similar to GPST [35]. We also show how we can use rejection sampling to mask the degree of the secret isogeny. When it comes to further research on the subject, while the level structures we focus on are by far the most common for implementations, they are not the only ones. For example, FESTA can be implemented using circulant matrices for which our results do not apply as they are. It would be useful to generalize our zero-knowledge proof in a way for them to apply to any level structure.

References

- [1] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York, ninth Dover printing, tenth GPO printing edition, 1964. <https://doi.org/10.2307/2282672>.
- [2] Noga Alon, Itai Benjamini, Eyal Lubetzky, and Sasha Sodin. Non-backtracking random walks mix faster. *Communications in Contemporary Mathematics*, 09(04):585–603, August 2007. <http://dx.doi.org/10.1142/S0219199707002551>.
- [3] Sarah Arpin. Adding level structure to supersingular elliptic curve isogeny graphs. *Journal de théorie des nombres de Bordeaux*, 36(2):405–443, 2024. <https://doi.org/10.5802/jtnb.1283>.
- [4] Shahla Atapoor, Karim Baghery, Daniele Cozzo, and Robi Pedersen. Practical robust DKG protocols for CSIDH. In Mehdi Tibouchi and XiaoFeng Wang, editors, *Applied Cryptography and Network Security*, pages 219–247, Cham, 2023. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-33491-7_9.
- [5] Andrea Basso. A post-quantum round-optimal oblivious PRF from isogenies. In Claude Carlet, Kalikinkar Mandal, and Vincent Rijmen, editors, *Selected Areas in Cryptography — SAC 2023*, pages 147–168, Cham, 2024. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-53368-6_8.
- [6] Andrea Basso, Giulio Codogni, Deirdre Connolly, Luca De Feo, Tako Boris Fouotsa, Guido Maria Lido, Travis Morrison, Lorenz Panny, Sikhar Patranabis, and Benjamin Wesolowski. Supersingular curves you can trust. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology — EUROCRYPT 2023*, pages 405–437, Cham, 2023. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-30617-4_14.
- [7] Andrea Basso, Pierrick Dartois, Luca De Feo, Antonin Leroux, Luciano Maino, Giacomo Pope, Damien Robert, and Benjamin Wesolowski. SQIsign2D–West. In Kai-Min

- Chung and Yu Sasaki, editors, *Advances in Cryptology — ASIACRYPT 2024*, pages 339–370, Singapore, 2025. Springer Nature Singapore. https://doi.org/10.1007/978-981-96-0891-1_11.
- [8] Andrea Basso and Tako Boris Fouotsa. New SIDH countermeasures for a more efficient key exchange. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology — ASIACRYPT 2023*, pages 208–233, Singapore, 2023. Springer Nature Singapore. https://doi.org/10.1007/978-981-99-8742-9_7.
 - [9] Andrea Basso, Luciano Maino, and Giacomo Pope. FESTA: Fast encryption from supersingular torsion attacks. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology — ASIACRYPT 2023*, pages 98–126, Singapore, 2023. Springer Nature Singapore. https://doi.org/10.1007/978-981-99-8739-9_4.
 - [10] Ward Beullens, Lucas Disson, Robi Pedersen, and Frederik Vercauteren. CSI-RASh: Distributed key generation for CSIDH. In Jung Hee Cheon and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography*, pages 257–276, Cham, 2021. Springer International Publishing. https://doi.org/10.1007/978-3-030-81293-5_14.
 - [11] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology — ASIACRYPT 2019*, pages 227–247, Cham, 2019. Springer International Publishing. https://doi.org/10.1007/978-3-030-34578-5_9.
 - [12] Jean-François Biasse, David Jao, and Anirudh Sankar. A quantum algorithm for computing isogenies between supersingular elliptic curves. In Willi Meier and Debdeep Mukhopadhyay, editors, *Progress in Cryptology — INDOCRYPT 2014*, pages 428–442, Cham, 2014. Springer International Publishing. https://doi.org/10.1007/978-3-319-13039-2_25.
 - [13] Jeffrey Burdges and Luca De Feo. Delay encryption. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology — EUROCRYPT 2021*, pages 302–326, Cham, 2021. Springer International Publishing. https://doi.org/10.1007/978-3-030-77870-5_11.
 - [14] Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology — EUROCRYPT 2023*, pages 423–447, Cham, 2023. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-30589-4_15.

- [15] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology — ASIACRYPT 2018*, pages 395–427, Cham, 2018. Springer International Publishing. https://doi.org/10.1007/978-3-030-03332-3_15.
- [16] Wouter Castryck and Frederik Vercauteren. A polynomial time attack on instances of M-SIDH and FESTA. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology — ASIACRYPT 2023*, pages 127–156, Singapore, 2023. Springer Nature Singapore. https://doi.org/10.1007/978-981-99-8739-9_5.
- [17] Andrew Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1–29, Jan 2014. <https://doi.org/10.1515/jmc-2012-0016>.
- [18] Giulio Codogni and Guido Lido. Spectral theory of isogeny graphs, 2024. <https://arxiv.org/abs/2308.13913>.
- [19] Craig Costello. The case for SIKE: A decade of the supersingular isogeny problem. Cryptology ePrint Archive, Paper 2021/543, 2021. <https://eprint.iacr.org/2021/543>.
- [20] Jean-Marc Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Paper 2006/291, 2006. <https://eprint.iacr.org/2006/291>.
- [21] Ronald Cramer. Modular design of secure yet practical cryptographic protocols. *Ph. D.-thesis, CWI and U. of Amsterdam*, 2, 1996. <https://hdl.handle.net/11245/1.135251>.
- [22] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo G. Desmedt, editor, *Advances in Cryptology — CRYPTO '94*, pages 174–187, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-48658-5_19.
- [23] Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In Joan Feigenbaum, editor, *Advances in Cryptology — CRYPTO '91*, pages 445–456, Berlin, Heidelberg, 1992. Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-46766-1_36.
- [24] Pierrick Dartois, Antonin Leroux, Damien Robert, and Benjamin Wesolowski. SQIsignHD: New dimensions in cryptography. In Marc Joye and Gregor Leander,

- editors, *Advances in Cryptology — EUROCRYPT 2024*, pages 3–32, Cham, 2024. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-58716-0_1.
- [25] Luca De Feo, Samuel Dobson, Steven D. Galbraith, and Lukas Zobernig. SIDH proof of knowledge. In *Advances in Cryptology – ASIACRYPT 2022*, page 310–339, Berlin, Heidelberg, 2023. Springer-Verlag. https://doi.org/10.1007/978-3-031-22966-4_11.
 - [26] Luca De Feo, Tako Boris Fouotsa, and Lorenz Panny. Isogeny problems with level structure. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology — EUROCRYPT 2024*, pages 181–204, Cham, 2024. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-58754-2_7.
 - [27] Luca De Feo and Steven D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology — EUROCRYPT 2019*, pages 759–789, Cham, 2019. Springer International Publishing. https://doi.org/10.1007/978-3-030-17659-4_26.
 - [28] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. SQISign: Compact post-quantum signatures from quaternions and isogenies. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology — ASIACRYPT 2020*, pages 64–93, Cham, 2020. Springer International Publishing. https://doi.org/10.1007/978-3-030-64837-4_3.
 - [29] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976. <https://doi.org/10.1109/TIT.1976.1055638>.
 - [30] Edward Eaton, David Jao, Chelsea Komlo, and Youcef Mokrani. Towards post-quantum key-updatable public-key encryption via supersingular isogenies. In Riham AlTawy and Andreas Hülsing, editors, *Selected Areas in Cryptography*, pages 461–482, Cham, 2022. Springer International Publishing. https://doi.org/10.1007/978-3-030-99277-4_22.
 - [31] Luca De Feo, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Simon-Philipp Merz, Lorenz Panny, and Benjamin Wesolowski. SCALLOP: Scaling the CSI-FiSh. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *Public-Key Cryptography — PKC 2023*, pages 345–375, Cham, 2023. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-31368-4_13.

- [32] Amos Fiat and Adi Shamir. How to prove yourself: practical solutions to identification and signature problems. In *Proceedings on Advances in Cryptology—CRYPTO '86*, page 186–194, Berlin, Heidelberg, 1987. Springer-Verlag. https://doi.org/10.1007/3-540-47721-7_12.
- [33] Tako Boris Fouotsa, Tomoki Moriya, and Christophe Petit. M-SIDH and MD-SIDH: Countering SIDH attacks by masking information. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology — EUROCRYPT 2023*, pages 282–309, Cham, 2023. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-30589-4_10.
- [34] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael Wiener, editor, *Advances in Cryptology — CRYPTO' 99*, pages 537–554, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-48405-1_34.
- [35] Steven D. Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology — ASIACRYPT 2016*, pages 63–91, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-53887-6_3.
- [36] Juncheol Han. The general linear group over a ring. *Bulletin of the Korean Mathematical Society*, 43, 08 2006. <https://doi.org/10.4134/BKMS.2006.43.3.619>.
- [37] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the fujisaki-okamoto transformation. In *Theory of Cryptography: 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part I*, page 341–371, Berlin, Heidelberg, 2017. Springer-Verlag. https://doi.org/10.1007/978-3-319-70500-2_12.
- [38] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985. <https://doi.org/10.1017/CB09780511810817>.
- [39] David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Geovandro Pereira, Joost Renes, Vladimir Soukharev, and David Urbanik. Supersingular Isogeny Key Encapsulation (NIST Round 2), 04 2019. <https://doi.org/10.13140/RG.2.2.26543.07847>.
- [40] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In Bo-Yin Yang, editor, *Post-Quantum Cryptog-*

- raphy, pages 19–34, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-25405-5_2.
- [41] David Jao, Stephen D. Miller, and Ramarathnam Venkatesan. Expander graphs based on GRH with an application to elliptic curve cryptography. *Journal of Number Theory*, 129(6):1491–1504, Jun 2009. <https://doi.org/10.1016/j.jnt.2008.11.006>.
 - [42] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology — EUROCRYPT 2023*, pages 448–471, Cham, 2023. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-30589-4_16.
 - [43] Youcef Mokrani and David Jao. Generating supersingular elliptic curves over $\mathbb{F}_p$ with unknown endomorphism ring. In Anupam Chattopadhyay, Shivam Bhasin, Stjepan Picek, and Chester Rebeiro, editors, *Progress in Cryptology — INDOCRYPT 2023*, pages 159–174, Cham, 2024. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-56232-7_8.
 - [44] Youcef Mokrani and David Jao. Commuting Ramanujan graphs and the random self-reducibility of isogeny problems. In Ruben Niederhagen and Markku-Juhani O. Saarinen, editors, *Post-Quantum Cryptography*, pages 38–60, Cham, 2025. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-86602-9_2.
 - [45] Tomoki Moriya, Katsuyuki Takashima, and Tsuyoshi Takagi. Group key exchange from CSIDH and its application to trusted setup in supersingular isogeny cryptosystems. In Zhe Liu and Moti Yung, editors, *Information Security and Cryptology*, pages 86–98, Cham, 2020. Springer International Publishing. https://doi.org/10.1007/978-3-030-42921-8_5.
 - [46] Marzio Mula, Nadir Murru, and Federico Pintore. On random sampling of supersingular elliptic curves. *Annali di Matematica Pura ed Applicata (1923 -)*, 2024. <https://doi.org/10.1007/s10231-024-01528-x>.
 - [47] Aurel Page and Damien Robert. Introducing Clapoti(s): Evaluating the isogeny class group action in polynomial time. Cryptology ePrint Archive, Paper 2023/1766, 2023. <https://eprint.iacr.org/2023/1766>.
 - [48] Lorenz Panny. CSI-FiSh really isn’t polynomial-time, 2023. <https://yx7.cc/blah/2023-04-14.html>.

- [49] Chris Peikert. He gives C-sieves on the CSIDH. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology — EUROCRYPT 2020*, pages 463–492, Cham, 2020. Springer International Publishing. https://doi.org/10.1007/978-3-030-45724-2_16.
- [50] Peter Petersen. *Linear Algebra*. Undergraduate Texts in Mathematics. Springer New York, 2012. <https://doi.org/10.1007/978-1-4614-3612-6>.
- [51] Christophe Petit. Faster algorithms for isogeny problems using torsion point images. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology — ASIACRYPT 2017*, pages 330–353, Cham, 2017. Springer International Publishing. https://doi.org/10.1007/978-3-319-70697-9_12.
- [52] Arnold K. Pizer. Ramanujan graphs and Hecke operators. *Bulletin (New Series) of the American Mathematical Society*, 23(1):127 – 137, 1990. <https://doi.org/10.1090/s0273-0979-1990-15918-x>.
- [53] Stephen Pohlig and Martin Hellman. *An Improved Algorithm for Computing Logarithms over $GF(p)$ and Its Cryptographic Significance*, pages 415–430. ACM Press, 08 2022. <https://doi.org/10.1145/3549993.3550009>.
- [54] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978. <https://doi.org/10.1145/359340.359342>.
- [55] Damien Robert. Breaking SIDH in polynomial time. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology — EUROCRYPT 2023*, pages 472–503, Cham, 2023. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-30589-4_17.
- [56] Damien Robert. On the efficient representation of isogenies (a survey). Cryptology ePrint Archive, Paper 2024/1071, 2024. <https://eprint.iacr.org/2024/1071>.
- [57] Naser T. Sardari. Diameter of Ramanujan graphs and random Cayley graphs. *Combinatorica*, 39:427–446, 2015. <https://doi.org/10.1007/s00493-017-3605-0>.
- [58] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, October 1997. <http://dx.doi.org/10.1137/S0097539795293172>.

- [59] Joseph H Silverman. *The Arithmetic of Elliptic Curves*. Graduate texts in mathematics. Springer, Dordrecht, 2009. <https://doi.org/10.1007/978-0-387-09494-6>.
- [60] Seiichiro Tani. Claw finding algorithms using quantum walk. *Theoretical Computer Science*, 410(50):5285–5297, November 2009. <http://dx.doi.org/10.1016/j.tcs.2009.08.030>.
- [61] Martin Tompa and Heather Woll. Random self-reducibility and zero knowledge interactive proofs of possession of information. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pages 472–482, 1987. <https://doi.org/10.1109/SFCS.1987.49>.
- [62] Jacques Vélu. Isogenies entre courbes elliptiques. *Comptes-Rendus de l'Academie des Sciences*, 273:238–241, 1971. <https://cir.nii.ac.jp/crid/1573387449093111296>.
- [63] Benjamin Wesolowski. The supersingular isogeny path and endomorphism ring problems are equivalent. *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1100–1111, 2021. <https://doi.org/10.1109/FOCS52979.2021.00109>.
- [64] Benjamin Wesolowski. Orientations and the supersingular endomorphism ring problem. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology — EUROCRYPT 2022*, pages 345–371, Cham, 2022. Springer International Publishing. https://doi.org/10.1007/978-3-031-07082-2_13.
- [65] Jiang Wu and Doug R. Stinson. An efficient identification protocol and the knowledge-of-exponent assumption. Cryptology ePrint Archive, Paper 2007/479, 2007. <https://eprint.iacr.org/2007/479>.